

Version 9 Release 1

*IBM InfoSphere Optim
Optim Designer verwenden*

IBM

Version 9 Release 1

*IBM InfoSphere Optim
Optim Designer verwenden*

IBM

Hinweis

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen unter „Bemerkungen“ auf Seite 131 gelesen werden.

Erste Ausgabe

Diese Ausgabe bezieht sich auf Version 2, Release 2, Modifikation 3 von Optim Designer und alle nachfolgenden Releases und Modifikationen, bis dieser Hinweis in einer Neuausgabe geändert wird.

Diese Veröffentlichung ist eine Übersetzung des Handbuchs
IBM InfoSphere Optim, Optim Designer User Guide,
herausgegeben von International Business Machines Corporation, USA

© Copyright International Business Machines Corporation 1996, 2012

Informationen, die nur für bestimmte Länder Gültigkeit haben und für Deutschland, Österreich und die Schweiz nicht zutreffen, wurden in dieser Veröffentlichung im Originaltext übernommen.

Möglicherweise sind nicht alle in dieser Übersetzung aufgeführten Produkte in Deutschland angekündigt und verfügbar; vor Entscheidungen empfiehlt sich der Kontakt mit der zuständigen IBM Geschäftsstelle.

Änderung des Textes bleibt vorbehalten.

Herausgegeben von:
TSC Germany
Kst. 2877
August 2012

Inhaltsverzeichnis

Kapitel 1. InfoSphere Optim Designer

verwenden	1
Erste Schritte	1
Optim-Perspektive	1
Optim Manager über Optim Designer verwenden	4
Beispieldaten	4
Datenbankunterstützung	13
Funktionen zur behindertengerechten Bedienung	13

Kapitel 2. Datenquellenverbindungen

verwalten	15
Mit einem Datenspeicheraliasnamen arbeiten	15
Datenspeicheraliasnamen definieren	15
Datenspeicheraliasnamen bearbeiten	15
Verbindung zu einem Datenspeicheraliasnamen herstellen	16
Mit Optim-Repository arbeiten	16
Verbindung zum Optim-Repository herstellen	16
Repository-Verbindung bearbeiten	16
Repository-Verbindung ändern	16
Optim-Verzeichnis dem Repository zuordnen	17
Anforderung in einen Service umsetzen	17

Kapitel 3. Datenmodelle verwalten

Mit Zugriffsdefinitionen arbeiten	19
Zugriffsdefinition erstellen	20
Tabellen in einer Zugriffsdefinition verwalten	20
Auswahlkriterien verwalten	22
Beziehungen in einer Zugriffsdefinition verwalten	23
Variablen in einer Zugriffsdefinition verwalten	25
Point-and-shoot-Listen verwalten	26
Mit Optim-Beziehungen arbeiten	26
Kompatibilitätsregeln für Beziehungen	28
Optim-Beziehung erstellen	30
Optim-Beziehung bearbeiten	30
Mit Optim-Primärschlüsseln arbeiten	32
Primärschlüssel erstellen	32
Primärschlüssel bearbeiten	33
Mit Tabellenzuordnungen arbeiten	33
Tabellenzuordnung erstellen	34
Tabellenzuordnung bearbeiten	34
Mit Spaltenzuordnungen arbeiten	36
Kompatibilitätsregeln für Spaltenzuordnungen	36
Spaltenzuordnung erstellen	38
Spaltenzuordnung bearbeiten	39

Kapitel 4. Datenverwaltungsservices

entwerfen	41
Mit Extraktionsservices arbeiten	41
Extraktionsservice erstellen	41
Extraktionsservice bearbeiten	42
Mit Konvertierungsservices arbeiten	44
Konvertierungsservice erstellen	45

Konvertierungsservice bearbeiten	45
Mit Einfügeservices arbeiten	46
Einfügeservice erstellen	46
Einfügeservice bearbeiten	46
Mit Ladeservices arbeiten	48
Ladeservice erstellen	50
Ladeservice bearbeiten	50
Datenverwaltungsservice testen	52

Kapitel 5. Daten maskieren

Datenmaskierungsfunktion anwenden	53
Datenmaskierungsfunktionen	53
Suchfunktionen	53
Mischfunktion	62
Funktion TRANS SSN	64
Funktion TRANS CCN	68
Funktion TRANS EML	71
Funktion TRANS COL	75
Funktion TRANS NID	77
Unterzeichenfolgefunktion	89
Zufallsfunktion	90
Sequenzielle Funktion	90
Identitäts- oder serielle Funktion	91
Oracle-Sequenzfunktion	92
Weitergabefunktion für Primärschlüssel- oder Fremdschlüsselwerte	92
Verkettete Ausdrücke	94
Numerische Ausdrücke	95
Literal- und Wertfunktionen	95
Alterungsfunktion	96
Währungsfunktion	99
Automatisch generierter E-Mail-Name	101
Formatierter E-Mail-Name	102
Funktion für Zufallszahlen	102
Funktion für fortlaufende Zahlen	102
Exitroutinen verwenden	102
Exitroutinen schreiben	103
Standardexitroutine	104
Quellenformatexit	106
Zielformatexit	108
Spaltenzuordnungsprozeduren mit Lua-Scripting schreiben	111
Lua-Funktionen für Spaltenzuordnungsprozeduren	111
Einschränkungen von Spaltenzuordnungsprozeduren	113
Beispiel für Spaltenzuordnungsprozedur: Generische Prozedur	113
Beispiel für Spaltenzuordnungsprozedur: Maskierungssuche	115
Beispiel für Spaltenzuordnungsprozedur: Maskieren länderspezifischer IDs	118

Kapitel 6. Daten verwalten

Daten durchsuchen	121
-------------------	-----

Daten bearbeiten	121
Daten vergleichen	122
Vergleichsanforderung definieren.	122
Vergleichsanforderung bearbeiten.	123
Vergleichsprozess ausführen	123
Tabellen erstellen	124

Kapitel 7. Daten mit Optim Designer	
verwalten	125
Daten mit Optim Designer extrahieren	125
Repository-Explorer-Ordner erstellen	125
Zugriffsdefinition erstellen	125

Auswahlkriterien definieren	126
Extraktionsservice erstellen	127
Daten mit Optim Designer maskieren	127
Tabellenzuordnung erstellen	128
Tabellenzuordnung bearbeiten.	128
Spaltenzuordnung erstellen.	129
Datenmaskierungsfunktion anwenden	129
Spaltenzuordnung einer Tabellenzuordnung hin-	
zufügen	130
Konvertierungsservice erstellen	130

Index	135
------------------------	------------

Kapitel 1. InfoSphere Optim Designer verwenden

Mit IBM® InfoSphere Optim Designer können Sie Datenmodelle, Datenschutzrichtlinien, Datenverwaltungsservices und Datenspeicher definieren. Sie können Objekte in Optim Designer definieren und sie im Optim-Repository freigeben. Verwenden Sie den Repository-Explorer, um Objekte im Repository zu verwalten. Mit dem Verzeichnisexplorer können Sie Objekte und Umsetzungsanforderungen in einem Optim-Verzeichnis anzeigen.

Datenverwaltungsservices

Mit einem Datenverwaltungsservice können Sie Daten extrahieren, konvertieren, laden oder einfügen. Ein Service verwendet Zugriffsdefinitionen zum Definieren der zu extrahierenden Daten. Ferner verwendet ein Service die Tabellenzuordnungen und Spaltenzuordnungen, um Quellen- und Zieldaten zuzuordnen. Sie können ebenfalls Prozessanforderungen in einem Optim-Verzeichnis in Datenverwaltungsservices umsetzen. Mit Optim Manager im integrierten Modus können Sie Services testen.

Datenmaskierung

Sie können von einem Datenverwaltungsservice verarbeitete Daten maskieren. Mit Datenmaskierung können Sie Daten wie länderspezifische ID-Nummern, Kreditkartennummern, Datumsangaben, numerische Werte und personenbezogene Daten umsetzen. Geben Sie eine Datenmaskierungsfunktion über eine Spaltenzuordnung ein oder erstellen Sie eine Spaltenzuordnungsprozedur über ein LUA-Script.

Zugriffsdefinitionen

Mit einer Zugriffsdefinition können Sie Datenmodelle und Untergruppen von Daten in übereinstimmenden oder in verschiedenen Schemas definieren. Eine Zugriffsdefinition gibt Auswahlkriterien und Beziehungen an, die in einem Datenverwaltungsservice verwendet werden. Sie können Beziehungen außerhalb eines Datenbankverwaltungssystems definieren und SQL zum Definieren der Auswahlkriterien verwenden.

Optim Manager

Sie können Optim Manager in Optim Designer (integrierter Modus) öffnen, was es Ihnen ermöglicht, Datenverwaltungsservices zu testen.

Erste Schritte

Zuerst müssen Sie mit dem Repository-Explorer eine Verbindung zu einem Optim-Repository herstellen.

Optim-Perspektive

In Optim Designer stellt die Optim-Perspektive Tools bereit, die Sie zum Definieren von Datenmodellen und Datenverwaltungsservices und zur Herstellung einer Verbindung zu einem Optim-Verzeichnis benötigen. Wenn Sie Optim Designer zum ersten Mal öffnen, ist die Optim-Perspektive die Standardperspektive.

Die Optim-Perspektive enthält folgende Sichten:

Repository-Explorer

Definieren Sie über den Repository-Explorer Datenspeicher-Aliasnamen, Zugriffsdefinitionen, Datenverwaltungsservices und andere Objekte, die in einem Optim-Repository gespeichert werden.

Verzeichnisexplorer

Stellen Sie über den Verzeichnisexplorer die Verbindung zu einem Optim-Verzeichnis her.

Klicken Sie auf **Fenster > Perspektive öffnen > Andere**, um die Optim-Perspektive zu öffnen. Wählen Sie **Optim** im Fenster **Perspektive öffnen** aus.

Repository-Explorer

Definieren Sie über den Repository-Explorer Datenspeicher-Aliasnamen, Zugriffsdefinitionen, Datenverwaltungsservices und andere Objekte, die in einem Optim-Repository gespeichert werden.

Wenn Optim Designer mit einem Optim-Repository verbunden ist, schließt der Repository-Explorer folgende Objekte ein:

Datenspeicheraliasnamen

Ein Datenspeicheraliasname ist ein benutzerdefiniertes Objekt, das einer Datenbank zugeordnet ist. Wenn Sie einen Datenspeicheraliasnamen definieren, stellen Sie Parameter bereit, anhand derer Optim mit dieser Datenbank kommuniziert. Zu diesen Parametern zählen Typ und Version des Datenbankverwaltungssystems und die Eigenschaften für eine Clientverbindung und eine JDBC-Verbindung.

Ein Datenspeicheraliasname dient als übergeordnetes Qualifikationsmerkmal, mit dem Sie für die Ausführung angeforderter Funktionen auf eine bestimmte Datenbank zugreifen können. In einer Zugriffsdefinition müssen Sie z. B. den Name einer Tabelle mit dem Datenspeicheraliasnamen qualifizieren. Der referenzierte Datenspeicheraliasname stellt die Parameter bereit, die zum Herstellen einer Verbindung zur Datenbank erforderlich sind, in der die Tabelle gespeichert ist.

Optim-Primärschlüssel

Ein Primärschlüssel ist die Spalte(n) mit Werten, die jede Zeile in einer Tabelle eindeutig identifizieren. Eine Datenbanktabelle muss einen Primärschlüssel haben, damit Optim Daten in ihr einfügen, aktualisieren, wiederherstellen oder löschen kann. Optim verwendet Primärschlüssel, die für die Datenbank definiert sind. Sie können allerdings auch Optim-Primärschlüssel definieren, um die Primärschlüssel in der Datenbank zu ergänzen.

Optim-Beziehungen

Optim verwendet Beziehungen, um die Daten zu ermitteln, die aus zugehörigen Tabellen abgerufen werden sollen, und basiert auf in der Datenbank definierten Beziehungen, soweit sie vorhanden sind. Sie können allerdings auch Optim-Beziehungen definieren, um Beziehungen in der Datenbank zu ergänzen.

Zugriffsdefinitionen

Über Zugriffsdefinitionen können Sie die Tabellen, Beziehungstraversierung und Auswahlkriterien für die zu verarbeitenden Daten angeben.

Spaltenzuordnungen

Eine Spaltenzuordnung stellt Spezifikationen bereit, die zum Abgleichen von Spalten oder zum Ausschließen von Spalten aus der Verarbeitung in einem Datenverwaltungsservice erforderlich sind. Konvertierungs-, Einfüge- und Ladeservices müssen eine Tabellenzuordnung referenzieren, die mindestens eine Spaltenzuordnung referenzieren kann. Sie können mit einer Spaltenzuordnung Datenumsetzungen mit Funktionen oder Spaltenzuordnungsprozeduren definieren.

Tabellenzuordnungen

Mit einer Tabellenzuordnung können Sie Spezifikationen für das Korrelieren von Quellen- und Zieltabellen kompatibler Daten definieren. Sie können Tabellen zuordnen, die verschiedene Namen haben, Tabellennamen ändern, Tabellen aus einem Prozess ausschließen oder Spaltenzuordnungen für eine bessere Kontrolle über die Daten einschließen.

Services

Mit einem Datenverwaltungsservice können Sie Daten extrahieren, konvertieren, laden oder einfügen. Ein Service verwendet Zugriffsdefinitionen zum Definieren der zu extrahierenden Daten. Ferner verwendet ein Service die Tabellenzuordnungen und Spaltenzuordnungen, um Quellen- und Zieldaten zuzuordnen. Sie können Daten maskieren, indem Sie eine Datenschutzrichtlinie auf eine Entität anwenden, die von einem Service verarbeitet wird. Sie können ebenfalls Prozessanforderungen in einem Optim-Verzeichnis in Datenverwaltungsservices umsetzen. Mit Optim Manager im integrierten Modus können Sie Services testen.

Ordner

Ein Repository-Explorer-Ordner enthält Services, Zugriffsdefinitionen, Spaltenzuordnungen und Tabellenzuordnungen. Verwenden Sie die Ordner, um diese Objekte zu organisieren.

Repository-Explorer-Ordner erstellen:

Mit dem Fenster **Neuer Ordner** können Sie einen Ordner im Repository-Explorer erstellen.

Ein Repository-Explorer-Ordner enthält Services, Zugriffsdefinitionen, Spaltenzuordnungen und Tabellenzuordnungen. Verwenden Sie die Ordner, um diese Objekte zu organisieren.

Gehen Sie wie folgt vor, um einen Repository-Explorer-Ordner zu erstellen.

1. Führen Sie abhängig davon, ob ein Ordner vorhanden ist, einen der folgenden Schritte aus:
 - a. Wenn Sie einen ersten Ordner erstellen wollen, klicken Sie auf **Zum Erstellen eines neuen Ordners hier klicken** im Repository-Explorer.
 - b. Wenn Sie einen zusätzlichen Ordner erstellen wollen, klicken Sie mit der rechten Maustaste auf die Repository-Explorer-Ansicht und klicken Sie auf **Neu > Ordner**.

Das Fenster **Neuer Ordner** wird geöffnet.

2. Geben Sie einen Ordnernamen ein und klicken Sie auf **OK**.

Der neue Ordner wird im Repository-Explorer angezeigt.

Verzeichnisexplorer

Stellen Sie über den Verzeichnisexplorer die Verbindung zu einem Optim-Verzeichnis her. Einem Repository kann nur ein Optim-Verzeichnis zugeordnet werden.

Verwenden Sie den Verzeichnisexplorer um Prozessanforderungen in Datenverwaltungsservices umzusetzen.

Die folgenden Definitionen sind verfügbar:

Symbol	Definition
	Zugriffsdefinition
	Archivierungsanforderung
	Spalte
	Spaltenzuordnung
	Konvertierungsanforderung
	Ersteller-ID
	Datenbankaliasname
	Löschanforderung
	Extraktionsanforderung
	Einfügeanforderung
	Ladeanforderung

Symbol	Definition
	Primärschlüssel
	Beziehung
	Wiederherstellungsanforderung
	Tabelle
	Tabellenzuordnung
	Variable

Optim Manager über Optim Designer verwenden

Mit Optim Manager können Sie über Optim Designer Datenverwaltungsservices und Optim-Interoperabilitätsservices testen. Das Aufrufen von Optim Manager über Optim Designer wird auch als Verwendung von Optim Manager im integrierten Modus bezeichnet.

Optim Manager ist eine Webanwendung, mit der Sie Services testen können. Optim Manager wird im von Optim Designer bereitgestellten internen Browser oder in einem externen Browser angezeigt. Sie können den Browser auswählen, der von Optim Designer für Optim Manager verwendet werden soll, indem Sie in Optim Designer auf **Fenster > Vorgaben > Allgemein > Web-Browser** klicken.

Optim Manager in Optim Designer öffnen

Sie können Optim Manager öffnen, indem Sie einen Service testen. Außerdem können Sie Optim Manager öffnen, indem Sie die folgende URL in einen Web-Browser eingeben: `http://localhost:portnumber/console`. Dabei ist *portnumber* die Portnummer, die Optim Manager zugeordnet ist. Die Standardportnummer ist 60000.

Wenn der Browser beim Öffnen von Optim Manager eine Nachricht anzeigt, dass eine Seite nicht gefunden wurde, besteht möglicherweise ein Portkonflikt und Sie müssen die Portnummer für Optim Manager ändern.

Sie können Optim Manager auch über die Optim Designer-Symbolleiste öffnen.

Portnummer von Optim Manager ändern

Wenn Sie die Portnummer ändern wollen, müssen Sie die folgende Eigenschaft in der Datei `eclipse.ini` im Optim Designer-Installationsverzeichnis bearbeiten:

```
-Dorg.eclipse.equinox.http.jetty.http.port=portnumber
```

Dabei ist *portnumber* die neue Portnummer von Optim Manager. Wenn Optim Designer geöffnet ist, müssen Sie die Anwendung erneut starten, um die neue Portnummer anzuwenden.

Beispieldaten

Optim stellt Beispieldaten zur Datenmaskierung sowie vordefinierte Quellen- und Zieldatenquellen bereit.

Beispieldatenbanktabellen und Struktur

Als Gruppe enthalten die Datenbanktabellen Informationen zu Kunden und Aufträgen mit Versandanweisungen. Die Beispieldatenbanken enthalten außerdem Informationen zu Verkauf und Lagerbestand. Geringfügig

gige Unterschiede in den Datentypen können je nach Datenbankverwaltungssystem vorliegen, das Sie zum Installieren der Beispieltabellen verwenden. Das folgende Diagramm zeigt die Basisstruktur der Beispieldatenbank.

Als Gruppe enthalten die Datenbanktabellen Informationen zu Kunden und Aufträgen mit Versandanweisungen. Die Beispieltabellen enthalten außerdem Informationen zu Verkauf und Lagerbestand. Geringfügige Unterschiede in den Datentypen können je nach Datenbankverwaltungssystem vorliegen, das Sie zum Installieren der Beispieltabellen verwenden.

Zusammen mit den Beispieldatenbanken wird eine zusätzliche Gruppe von Tabellen installiert. Tabellen in dieser zusätzlichen Gruppe haben dieselben Namen wie die Tabellen in der ersten Gruppe, nur mit dem angehängten Suffix "2". Die vier Tabellen in dieser Zusatzgruppe sind:

- OPTIM_CUSTOMERS2
- OPTIM_ORDERS2
- OPTIM_DETAILS2
- OPTIM_ITEMS2

Tabellen in der Zusatzgruppe enthalten keine Daten. Sie dienen lediglich zur Veranschaulichung der Funktionen in Optim.

Tabelle OPTIM_SALES:

Die Tabelle OPTIM_SALES gibt jeden Vertriebsmitarbeiter mit Name, ID-Number und Führungskraft an.

Die Tabelle OPTIM_SALES hat folgende Spalten:

SALESMAN_ID

CHAR; bis zu 6 Zeichen; darf nicht NULL enthalten.

FIRST_NAME

VARCHAR; bis zu 15 Zeichen; darf nicht NULL enthalten.

LAST_NAME

VARCHAR; bis zu 15 Zeichen; darf nicht NULL enthalten.

NATIONALITY

VARCHAR; bis zu 30 Zeichen

NATIONAL_ID

VARCHAR; bis zu 30 Zeichen

PHONE_NUMBER

VARCHAR; bis zu 20 Zeichen; darf keinen Nullwert enthalten.

EMAIL_ADDRESS

VARCHAR; bis zu 70 Zeichen; darf nicht NULL enthalten.

AGE SMALLINT; darf nicht NULL enthalten; hat einen Standardwert.

SEX CHAR; ein Zeichen; darf nicht NULL enthalten; hat einen Standardwert. value.

TERRITORY

VARCHAR; bis zu 14 Zeichen; darf nicht NULL enthalten.

MANAGER_ID

VARCHAR; bis zu 6 Zeichen

Primärschlüssel

Die Spalte SALESMAN_ID ist die Primärschlüsselspalte.

Beziehungen zu anderen Tabellen

Die Tabelle OPTIM_SALES ist übergeordnet

- der Tabelle OPTIM_CUSTOMERS (über einen Fremdschlüssel für Spalte SALESMAN_ID).
- der Tabelle OPTIM_MALE_RATES über eine Optim-datengesteuerte Beziehung für Spalte AGE, wenn SEX = 'M'.
- der Tabelle OPTIM_FEMALE_RATES über eine Optim-datengesteuerte Beziehung für Spalte AGE, wenn SEX = 'F'.
- der Tabelle OPTIM_STATE_LOOKUP über eine Optim-Unterzeichenfolgenbeziehung mithilfe von SUBSTR(SALESMAN_ID,1,2).

Tabelle OPTIM_CUSTOMERS:

Die Tabelle OPTIM_CUSTOMERS enthält Kundennamen, ID-Nummern und Adressen.

Die Tabelle OPTIM_CUSTOMERS hat folgende Spalten:

CUST_ID

CHAR; bis zu 5 Zeichen; darf nicht NULL enthalten; enthält eine Prüfbedingung.

CUSTNAME

CHAR; bis zu 20 Zeichen; darf nicht NULL enthalten.

ADDRESS1

VARCHAR; bis zu 100 Zeichen; darf nicht NULL enthalten.

ADDRESS2

VARCHAR; bis zu 100 Zeichen; darf nicht NULL enthalten.

LOCALITY

VARCHAR; bis zu 56 Zeichen

CITY VARCHAR; bis zu 60 Zeichen

STATE

VARCHAR; bis zu 30 Zeichen

COUNTRY_CODE

CHAR; bis zu 2 Zeichen

POSTAL_CODE

VARCHAR; bis zu 15 Zeichen

POSTAL_CODE_PLUS4

CHAR; bis zu 4 Zeichen; darf keinen Nullwert enthalten.

EMAIL_ADDRESS

VARCHAR; bis zu 70 Zeichen

PHONE_NUMBER

VARCHAR; bis zu 20 Zeichen

YTD_SALES

DECIMAL; Dollarbetrag; darf nicht NULL enthalten; hat einen Standardwert.

SALESMAN_ID

CHAR; bis zu 6 Zeichen

NATIONALITY

VARCHAR; bis zu 30 Zeichen

NATIONAL_ID

VARCHAR; bis zu 30 Zeichen

CREDITCARD_NUMBER
VARCHAR; 19 Zeichen

CREDITCARD_TYPE
VARCHAR; bis zu 30 Zeichen

CREDITCARD_EXP
CHAR; 4 Zeichen

CREDITCARD_CVV
VARCHAR; bis zu 4 Zeichen

DRIVER_LICENSE
VARCHAR; bis zu 30 Zeichen

CUSTOMER_INFO
XMLTYPE

Primärschlüssel

Die Spalte CUST_ID ist die Primärschlüsselspalte.

Beziehungen zu anderen Tabellen

Die Tabelle OPTIM_CUSTOMERS ist übergeordnet

- der Tabelle OPTIM_ORDERS (über einen Fremdschlüssel für Spalte CUST_ID).
- der Tabelle OPTIM_SHIP_TO (über eine Optim-Beziehung für Spalte CUST_ID).

Die Tabelle OPTIM_CUSTOMERS ist untergeordnet:

- der Tabelle OPTIM_SALES (über ihren Fremdschlüssel für Spalte SALESMAN_ID).

Tabelle OPTIM_ORDERS:

Die Tabelle OPTIM_ORDERS enthält Informationen zu Aufträgen einschließlich Auftragsnummer, Kunden-ID und Vertriebsmitarbeiter.

Die Tabelle OPTIM_ORDERS hat folgende Spalten:

ORDER_ID
DECIMAL; Auftrags-ID-Nummer; darf nicht NULL enthalten.

CUST_ID
CHAR; Kunden-ID-Nummer; bis zu 5 Zeichen; darf nicht NULL enthalten.

ORDER_DATE
TIMESTAMP; Auftragsdatum; darf nicht NULL enthalten; hat einen Standardwert.

ORDER_TIME
TIMESTAMP; Uhrzeit; darf nicht NULL enthalten; hat einen Standardwert.

FREIGHT_CHARGES
DECIMAL; Dollarbetrag

ORDER_SALESMAN
CHAR; bis zu 6 Zeichen

ORDER_POSTED_DATE
TIMESTAMP; darf nicht NULL enthalten; hat einen Standardwert.

ORDER_SHIP_DATE
CHAR; Datum der Auslieferung des Auftrags; bis zu 8 Zeichen; darf nicht NULL enthalten; hat einen Standardwert.

Primärschlüssel

Die Spalte ORDER_ID ist die Primärschlüsselspalte.

Beziehungen zu anderen Tabellen

Die Tabelle OPTIM_ORDERS ist der Tabelle OPTIM_DETAILS über einen Fremdschlüssel für Spalte ORDER_ID übergeordnet.

Die Tabelle OPTIM_ORDERS ist der Tabelle OPTIM_CUSTOMERS über ihren Fremdschlüssel für Spalte CUST_ID untergeordnet.

Tabelle OPTIM_DETAILS:

Die Tabelle OPTIM_DETAILS enthält zusätzliche Informationen für jeden Auftrag in der Tabelle OPTIM_ORDERS.

Die Tabelle OPTIM_DETAILS hat folgende Spalten:

ORDER_ID

DECIMAL; Auftrags-ID-Nummer; darf nicht NULL enthalten.

ITEM_ID

CHAR; max. 5 Zeichen; Element-ID-Nummer; darf nicht NULL enthalten.

ITEM_QUANTITY

DECIMAL; Anzahl Elemente; darf nicht NULL enthalten.

DETAIL_UNIT_PRICE

DECIMAL; Preis pro Einheit; Dollarbetrag; darf nicht NULL enthalten.

Primärschlüssel

Die Spalten ORDER_ID und ITEM_ID sind der Primärschlüssel.

Beziehungen zu anderen Tabellen

Die Tabelle OPTIM_DETAILS ist untergeordnet:

- der Tabelle OPTIM_ORDERS (über ihren Fremdschlüssel für Spalte ORDER_ID).
- der Tabelle OPTIM_ITEMS (über ihren Fremdschlüssel für Spalte ITEM_ID).

Tabelle OPTIM_ITEMS:

Die Tabelle OPTIM_ITEMS enthält Informationen zu jedem Element für einen Auftrag einschließlich Beschreibung, Preis und Lagerbestand.

Die Tabelle OPTIM_ITEMS hat folgende Spalten:

ITEM_ID

CHAR; bis zu 5 Zeichen; darf nicht NULL enthalten.

ITEM_DESCRIPTION

VARCHAR; bis zu 72 Zeichen; darf nicht NULL enthalten.

CATEGORY

VARCHAR; bis zu 14 Zeichen; darf nicht NULL enthalten.

RATING

CHAR; bis zu 4 Zeichen; darf nicht NULL enthalten.

UNIT_PRICE
DECIMAL; Dollarbetrag; darf nicht NULL enthalten.

ON_HAND_INVENTORY
INTEGER; darf nicht NULL enthalten.

Primärschlüssel

Die Spalte ITEM_ID ist die Primärschlüsselspalte.

Beziehungen zu anderen Tabellen

Die Tabelle OPTIM_ITEMS ist der Tabelle OPTIM_DETAILS über einen Fremdschlüssel für Spalte ITEM_ID übergeordnet.

Tabelle OPTIM_SHIP_TO:

Die Tabelle OPTIM_SHIP_TO enthält Auftragsversandinformationen.

Die Tabelle OPTIM_SHIP_TO hat folgende Spalten:

CUST_ID
CHAR; bis zu 5 Zeichen; darf nicht NULL enthalten.

SHIP_ID
DECIMAL; darf nicht NULL enthalten.

ADDRESS1
VARCHAR; bis zu 100 Zeichen

ADDRESS2
VARCHAR; bis zu 100 Zeichen

LOCALITY
VARCHAR; bis zu 56 Zeichen

CITY VARCHAR; bis zu 30 Zeichen

STATE
VARCHAR; bis zu 30 Zeichen

COUNTRY_CODE
CHAR; aus 2 Zeichen bestehende Abkürzung

POSTAL_CODE
VARCHAR; bis zu 15 Zeichen

POSTAL_CODE_PLUS4
CHAR; 4 Zeichen

IN_CARE_OF
VARCHAR; bis zu 31 Zeichen

SHIPPING_CHANGE_DT
TIMESTAMP; darf keine Nullwerte enthalten; hat einen Standardwert.

Primärschlüssel

Die Spalte SHIP_ID ist die Primärschlüsselspalte.

Beziehungen zu anderen Tabellen

Die Tabelle OPTIM_SHIP_TO ist der Tabelle OPTIM_SHIP_INSTR über eine Optim-Beziehung für Spalte SHIP_ID übergeordnet.

Die Tabelle OPTIM_SHIP_TO ist der Tabelle OPTIM_CUSTOMERS über ihre Optim-Beziehung für Spalte CUST_ID untergeordnet.

Tabelle OPTIM_SHIP_INSTR:

Die Tabelle OPTIM_SHIP_INSTR enthält detaillierte Informationen zum Auftragsversand.

Die Tabelle OPTIM_SHIP_INSTR hat folgende Spalten:

SHIP_ID

DECIMAL

SHIP_INSTR_ID

INTEGER

ORDER_SHIP_INSTR

VARCHAR; bis zu 254 Zeichen

SHIP_UPDATED

TIMESTAMP; darf nicht NULL enthalten; hat einen Standardwert.

Primärschlüssel

Die Spalte SHIP_INSTR_ID ist die Primärschlüsselspalte.

Beziehungen zu anderen Tabellen

Die Tabelle OPTIM_SHIP_INSTR ist der Tabelle OPTIM_SHIP_TO über ihre Optim-Beziehung für Spalte SHIP_ID untergeordnet.

Tabelle OPTIM_MALE_RATES:

Die Tabelle OPTIM_MALE_RATES enthält Versicherungstarife auf der Basis des Alters.

Die Tabelle OPTIM_MALE_RATES hat folgende Spalten:

AGE SMALLINT

RATE_PER_1000

DECIMAL; Tarif als Dollarbetrag

Primärschlüssel

Die Spalte RATE_PER_1000 ist die Primärschlüsselspalte.

Beziehungen zu anderen Tabellen

Die Tabelle OPTIM_MALE_RATES ist der Tabelle OPTIM_SALES über ihre Optim-datengesteuerte Beziehung für Spalte AGE untergeordnet.

Tabelle OPTIM_FEMALE_RATES:

Die Tabelle OPTIM_FEMALE_RATES enthält Versicherungstarife auf der Basis des Alters.

Die Tabelle OPTIM_FEMALE_RATES hat folgende Spalten:

AGE SMALLINT

RATE_PER_1000

DECIMAL; Tarif als Dollarbetrag

Primärschlüssel

Die Spalte RATE_PER_1000 ist die Primärschlüsselspalte.

Beziehungen zu anderen Tabellen

Die Tabelle OPTIM_FEMALE_RATES ist der Tabelle OPTIM_SALES über ihre Optim-datengesteuerte Beziehung für Spalte AGE untergeordnet.

Tabelle OPTIM_STATE_LOOKUP:

Die Tabelle OPTIM_STATE_LOOKUP enthält Codes für Staaten und entsprechende Abkürzungen.

Die Tabelle OPTIM_STATE_LOOKUP hat folgende Spalten:

DIST_CODE

CHAR; 3 Zeichen; darf keinen Nullwert enthalten.

DISTRICT

CHAR; 2 Zeichen; darf keinen Nullwert enthalten.

Primärschlüssel

Die Tabelle OPTIM_STATE_LOOKUP hat keinen Primärschlüssel.

Beziehungen zu anderen Tabellen

Die Tabelle OPTIM_STATE_LOOKUP ist der Tabelle OPTIM_SALES über eine Unterzeichenfolgenbeziehung für Spalte DISTRICT mithilfe von SUBSTR(SALESMAN_ID,1,2) untergeordnet.

Datenschutztabellen

Datenschutztabellen sind für Clients verfügbar, die über eine Optim-Datenschutzlizenz verfügen. Mit den Tabellen können Sie Unternehmensdaten und persönliche Daten wie Mitarbeiternamen, Kundennamen, Sozialversicherungsnummern, Kreditkartennummern und E-Mail-Adressen maskieren. Mit den Tabellen können Sie umgesetzte Daten generieren, die im Kontext der Anwendung eindeutig und gültig sind.

Sie können Datenschutztabellen für folgende Zwecke verwenden:

- Interne Datenschutzverstöße verhindern, indem die Entwicklern, Qualitätssicherungstestern und anderen Mitarbeitern zur Verfügung gestellten Daten anonymisiert oder maskiert werden.
- Initiativen zur Einhaltung von Datenschutzvorgaben dadurch unterstützen, dass Kundendaten durch dem Kontext entsprechende, aber fiktive Daten ersetzt werden.
- Vertrauliche Kundeninformationen und Mitarbeiterdaten in Ihren Umgebungen für Anwendungsentwicklung und -test schützen.
- Gültige Testergebnisse sicherstellen, indem maskierte Elemente über zugehörige Tabellen hinweg weitergegeben werden, um die referenzielle Integrität der Datenbank sicherzustellen.

Jede Kategorie für personenbezogene Daten wird in einer separaten Tabelle für folgende Länder bereitgestellt (Abkürzungen in Klammern): Australien (AU), Kanada (CA), Frankreich (FR), Deutschland (DE),

Italien (IT), Japan (JP), Spanien (ES), Großbritannien (UK) und Vereinigte Staaten (US). Jede Tabelle enthält eine Spalte mit fortlaufenden Zahlen, die für Suchrichtlinien verwendet wird, die eine Zeile in der Suchtabelle mithilfe von Hashwerten auswählen.

Jeder Tabellename setzt sich aus einem Präfix mit einer Abkürzung für das Land und der Kategorie (*Länderabkürzung_Kategorie*) zusammen. Die Adresstabelle für Kanada heißt beispielsweise CA_ADDRESSES und die Adresstabelle für Deutschland DE_ADDRESSES.

Das Schema verfügt über folgende Kategorien:

ADDRESSES

Tabellen, die Spalten für Anschrift, Ort, Standort (Staat oder Provinz) und Postleitzahl enthalten.

FIRSTNAME

Tabellen, die eine Spalte mit männlichen und weiblichen Vornamen enthalten.

FIRSTNAME_F

Tabellen, die eine Spalte mit weiblichen Vornamen enthalten.

FIRSTNAME_M

Tabellen, die eine Spalte mit männlichen Vornamen enthalten.

LASTNAME

Tabellen, die eine Spalte mit Familiennamen enthalten.

PERSON

Tabellen, die Spalten für Geburtsdatum, Vorname, Nachname, Geschlecht, Telefonnummer, länderspezifische ID, Firmenname und E-Mail-Adresse enthalten.

CCN Tabellen, die Kreditkartennummern für den zugehörigen Aussteller (MasterCard, VISA, usw.) enthalten.

DOMAIN_NAMES

Tabelle, die Domännennamen zum Maskieren von E-Mail-Adressen enthält.

Beispieldatentabellen erstellen

Mit dem Dienstprogramm 'Configuration' können Sie Beispieldatentabellen erstellen.

Gehen Sie wie folgt vor, um Beispieldatentabellen zu erstellen:

1. Klicken Sie in Optim Designer auf **Dienstprogramme > Konfigurieren**. Das Dienstprogramm 'Configuration' wird geöffnet.
2. Klicken Sie auf **Tasks > Load/Drop Sample Data** Der Assistent **Load/Drop Sample Data** wird geöffnet.
3. Führen Sie die Schritte im Assistenten aus.
Sie müssen ein Optim-Verzeichnisprojekt erstellen. Der Optim-Standardverzeichnisname für das Optim-Repository ist PODREPO.
Wenn Sie zur Auswahl eines Datenbankaliasnamens aufgefordert werden, wählen Sie einen Datenspeicheraliasnamen aus.

Datenschutztabellen erstellen

Mit dem Dienstprogramm 'Configuration' können Sie Datenschutztabellen erstellen.

Gehen Sie wie folgt vor, um Datenschutztabellen zu erstellen:

1. Klicken Sie in Optim Designer auf **Dienstprogramme > Konfigurieren**. Das Dienstprogramm 'Configuration' wird geöffnet.
2. Klicken Sie auf **Tasks > Load/Drop Data Privacy Data** Der Assistent **Load/Drop Data Privacy Data** wird geöffnet.
3. Führen Sie die Schritte im Assistenten aus.

Sie müssen ein Optim-Verzeichnisprojekt erstellen. Der Optim-Standardverzeichnisname für das Optim-Repository ist PODREPO.

Wenn Sie zur Auswahl eines Datenbankaliasnamens aufgefordert werden, wählen Sie einen Datenspeicheraliasnamen aus.

Datenbankunterstützung

Optim Designer bietet Unterstützung für mehrere Datenbankverwaltungssysteme.

Optim Designer unterstützt folgende Datenbanken:

- DB2 for z/OS Version 8.1, Version 9.1, Version 10.1
- DB2 for Linux, UNIX and Windows Version 8.2, Version 9.1, Version 9.5, Version 9.7
- DB2 for i Version 5.4
- Informix Version 10
- Microsoft SQL Server 2005, 2008
- Oracle Version 10.2, Version 11, Version 11.2
- Sybase Version 12.5, Version 15, Version 15.5
- Teradata Version 2.6, Version 12, Version 13
- IBM Informix Version 11.5, Version 11.7

DB2-Voraussetzungen

Damit Optim vollständige JDBC-Metadaten von einer DB2 z/OS-Instanz abrufen kann, muss der Wert DESCSTAT in ZPARMS auf YES gesetzt sein. Sie müssen außerdem den Job DSNTIJMS ausführen, um die von JDBC benötigten gespeicherten Prozeduren zu installieren, die erforderlichen Pakete zu binden und die Sicherheitsberechtigungen festzulegen. Darüber hinaus sind Workload Manager-Definitionen erforderlich, um sicherzustellen, dass Workload Manager (WML) den Adressraum der gespeicherten Prozeduren starten kann, wenn dies von DB2 angefordert wird.

Funktionen zur behindertengerechten Bedienung

Funktionen zur behindertengerechten Bedienung helfen Menschen mit körperlichen Behinderungen wie eingeschränktem Bewegungs- oder Sehvermögen oder mit anderen Sonderanforderungen erfolgreich mit Softwareprodukten zu arbeiten.

Optim Designer verwendet die in der Eclipse-Umgebung verfügbaren Funktionen zur behindertengerechten Bedienung.

Funktionen zur behindertengerechten Bedienung helfen Menschen mit körperlichen Behinderungen wie eingeschränktem Bewegungs- oder Sehvermögen oder mit Sonderanforderungen, erfolgreich mit Softwareprodukten zu arbeiten.

Kapitel 2. Datenquellenverbindungen verwalten

Mit dem Repository-Explorer können Sie Aliasnamen für Dateidatenspeicher erstellen. Stellen Sie über den Verzeichnisexplorer die Verbindung zu einem Optim-Verzeichnis her.

Mit einem Datenspeicheraliasnamen arbeiten

Ein Datenspeicheraliasname ist ein benutzerdefiniertes Objekt, das einer Datenbank zugeordnet ist. Wenn Sie einen Datenspeicheraliasnamen definieren, stellen Sie Parameter bereit, anhand derer Optim mit dieser Datenbank kommuniziert. Zu diesen Parametern zählen Typ und Version des Datenbankverwaltungssystems und die Eigenschaften für eine Clientverbindung und eine JDBC-Verbindung.

Ein Datenspeicheraliasname dient als übergeordnetes Qualifikationsmerkmal, mit dem Sie für die Ausführung angeforderter Funktionen auf eine bestimmte Datenbank zugreifen können. In einer Zugriffsdefinition müssen Sie z. B. den Name einer Tabelle mit dem Datenspeicheraliasnamen qualifizieren. Der referenzierte Datenspeicheraliasname stellt die Parameter bereit, die zum Herstellen einer Verbindung zur Datenbank erforderlich sind, in der die Tabelle gespeichert ist.

Ein Datenspeicheraliasname gibt eine bestimmte Datenbank an und dient als Präfix in den vollständig qualifizierten Namen von Primärschlüsseln, Datenbanktabellen und Beziehungen. Datenspeicheraliasnamen sind wichtige Elemente beim Verwalten Ihrer Datenbanken. Die folgenden Regeln gelten für Datenspeicheraliasnamen:

- Jede Datenbank kann nur einen einzigen Datenspeicheraliasnamen haben.
- Jeder Datenspeicheraliasname muss eindeutig sein.
- Objekte im Repository können nicht denselben Namen wie ein Datenspeicheraliasname haben.

Datenspeicheraliasnamen definieren

Mit dem Assistenten **Neuer Datenspeicheraliasname** können Sie einen neuen Datenspeicheraliasnamen definieren.

Zur Definition eines Datenspeicheraliasnamens benötigen Sie die Verbindungszeichenfolge für eine Clientverbindung und die .jar-Datei für eine JDBC-Verbindung.

Gehen Sie wie folgt vor, um einen Datenspeicheraliasnamen zu definieren:

1. Klicken Sie im Repository-Explorer mit der rechten Maustaste auf **Datenspeicheraliasnamen** und klicken Sie auf **Neuer Datenspeicheraliasname**. Der Assistent **Neuer Datenspeicheraliasname** wird geöffnet.
2. Führen Sie die Schritte im Assistenten aus.

Sie müssen Eigenschaften für eine Clientverbindung und eine JDBC-Verbindung eingeben. Sie müssen ebenfalls den Zeichensatz angeben, der von der Datenbank verwendet wird, und festlegen, wie Pakete zum Zugreifen auf Optim-Objekte verwaltet werden.

Datenspeicheraliasnamen bearbeiten

Mit dem Assistenten für Verbindungseigenschaften können Sie einen Datenspeicheraliasnamen bearbeiten.

Gehen Sie wie folgt vor, um einen Datenspeicheraliasnamen zu bearbeiten:

1. Erweitern Sie im Repository-Explorer den Knoten **Datenspeicheraliasnamen**, klicken Sie mit der rechten Maustaste auf den Datenspeicheraliasnamen und klicken Sie auf **Öffnen**. Der Assistent für Verbindungseigenschaften wird geöffnet.
2. Führen Sie die Schritte im Assistenten aus.

Verbindung zu einem Datenspeicheraliasnamen herstellen

Mit dem Repository-Explorer können Sie eine Verbindung zu einem Datenspeicheraliasnamen herstellen.

Gehen Sie wie folgt vor, um eine Verbindung zu einem Datenspeicheraliasnamen herzustellen:

Erweitern Sie im Repository-Explorer den Knoten **Datenspeicheraliasnamen**, klicken Sie mit der rechten Maustaste auf den Datenspeicheraliasnamen und klicken Sie auf **Verbindung herstellen**.

Mit Optim-Repository arbeiten

Das Optim-Repository enthält die Definitionen wie Datenspeicheraliasnamen, Services und Zugriffsdefinitionen, die von den Optim-Komponenten gemeinsam verwendet werden.

Einem Repository kann nur ein Optim-Verzeichnis zugeordnet werden.

Verbindung zum Optim-Repository herstellen

Mit dem Fenster **Neue Repository-Verbindung** können Sie eine Verbindung zum Optim-Repository definieren.

Das Optim-Repository muss aktiv sein, damit eine Verbindung erstellt werden kann.

Gehen Sie wie folgt vor, um eine Verbindung zum Optim-Repository herzustellen:

1. Klicken Sie im Repository-Explorer mit der rechten Maustaste auf **Optim Repository** und klicken Sie auf **Neu > Repository-Verbindung**. Das Fenster **Neue Repository-Verbindung** wird geöffnet.
2. Geben Sie Verbindungsinformationen und die Benutzerberechtigungsnaachweise für das Repository ein.
3. Klicken Sie auf **Testverbindung**, um die Verbindung zu prüfen.
4. Klicken Sie auf **OK**.

Wenn Optim Designer eine Verbindung zum Repository herstellt, wird im Repository-Explorer der Inhalt des Repositorys angezeigt.

Repository-Verbindung bearbeiten

Mit dem Fenster **Repository-Verbindung bearbeiten** können Sie eine Verbindung zum Optim-Repository definieren.

Gehen Sie wie folgt vor, um eine Verbindung zum Optim-Repository herzustellen:

1. Klicken Sie im Repository-Explorer mit der rechten Maustaste auf **Optim Repository** und klicken Sie auf **Öffnen**. Das Fenster **Repository-Verbindung bearbeiten** wird geöffnet.
2. Geben Sie Verbindungsinformationen und die Benutzerberechtigungsnaachweise für das Repository ein.
3. Klicken Sie auf **Testverbindung**, um die Verbindung zu prüfen.
4. Klicken Sie auf **OK**.

Wenn Optim Designer eine Verbindung zum Repository herstellt, wird im Repository-Explorer der Inhalt des Repositorys angezeigt.

Repository-Verbindung ändern

Im Fenster **Repository wechseln** können Sie eine Verbindung zu einem anderen Optim-Repository herstellen.

Das Optim-Repository muss aktiv sein, damit eine Verbindung geändert werden kann.

Gehen Sie wie folgt vor, um eine Verbindung zum Optim-Repository zu ändern:

1. Klicken Sie im Repository-Explorer mit der rechten Maustaste auf **Optim Repository** und klicken Sie auf **Repository wechseln**. Das Fenster **Repository wechseln** wird geöffnet.
2. Wählen Sie eine Repository-Verbindung aus.
3. Klicken Sie auf **OK**.

Wenn Optim Designer eine Verbindung zum Repository herstellt, wird im Repository-Explorer der Inhalt des Repositorys angezeigt.

Optim-Verzeichnis dem Repository zuordnen

Mit dem Fenster **Verzeichnis zuordnen** können Sie dem Repository ein Optim-Verzeichnis zuordnen.

Es muss eine Optim-Verzeichnisverbindung in der Registry des Optim Designer-Computers definiert sein. Mit dem Optim-Konfigurationsprogramm können Sie eine Verbindung in der Registry definieren.

Einem Repository kann nur ein Optim-Verzeichnis zugeordnet werden.

Gehen Sie wie folgt vor, um dem Repository ein Optim-Verzeichnis zuzuordnen:

1. Klicken Sie im Verzeichnisexplorer auf **Klicken Sie, um der Repository-Verbindung ein Optim-Verzeichnis zuzuordnen**. Das Fenster **Verzeichnis zuordnen** wird geöffnet.
2. Wählen Sie ein Optim-Verzeichnis aus.
3. Klicken Sie auf **Fertig stellen**.

Wenn Optim Designer eine Verbindung zum Optim-Verzeichnis herstellt, zeigt der Verzeichnisexplorer den Inhalt des Optim-Verzeichnisses an.

Anforderung in einen Service umsetzen

Mit der Umsetzungsanforderung an den Optim-Serviceassistenten kann eine Anforderung in einem Optim-Verzeichnis in einen Datenverwaltungsservice umgesetzt werden.

Ein Optim-Verzeichnis muss dem Repository zugeordnet sein.

Gehen Sie wie folgt vor, um eine Anforderung in einem Optim-Verzeichnis in einen Datenverwaltungsservice umzusetzen:

1. Klicken Sie im Verzeichnisexplorer auf **Umsetzen**. Die Umsetzungsanforderung an den Optim-Serviceassistenten wird geöffnet.
2. Überprüfen Sie auf der Seite **Datenbankaliasname und Datenbankverbindungen** die Verbindungsinformationen. Wenn kein Datenspeicheraliasname für einen Datenbankaliasnamen definiert ist, der der Anforderung zugeordnet ist, klicken Sie auf **Verbindung bearbeiten**, um einen Datenspeicheraliasnamen zu definieren. Klicken Sie auf **Weiter**.
3. Wenn die Umsetzung erfolgreich ist, klicken Sie auf **Fertig stellen**. Wenn die Umsetzung nicht erfolgreich ist, klicken Sie auf **Weiter**, um die Fehlerinformationen anzuzeigen.

Ein Ordner mit dem Namen des Optim-Verzeichnisses, das die Anforderung enthielt, wird im Repository-Explorer erstellt. Die umgesetzte Anforderung wird als Service unter dem Ordner aufgelistet.

Kapitel 3. Datenmodelle verwalten

Sie können Datenmodelle erstellen, die Daten zum Extrahieren und Zuordnen von Zieldaten für Datenverwaltungsservices definieren. Sie können auch Optim-Beziehungen und -Primärschlüssel zum Ergänzen von Beziehungen und Primärschlüsseln in der Datenbank erstellen.

Mit Zugriffsdefinitionen arbeiten

Über Zugriffsdefinitionen können Sie die Tabellen, Beziehungstraversierung und Auswahlkriterien für die zu verarbeitenden Daten angeben.

Eine Zugriffsdefinition beinhaltet die folgenden Elemente.

Tabellen

Eine Zugriffsdefinition muss auf mindestens eine Tabelle, eine Sicht, einen Aliasnamen oder ein Synonym verweisen. Die Tabelle, die Sicht, der Aliasname oder das Synonym, aus der oder dem Zeilen zuerst ausgewählt werden, wird als Starttabelle bezeichnet. Sie können den Namen einer Starttabelle eingeben und problemlos die Namen aller Tabellen einfügen, die zur Starttabelle gehören (max. 24.000 Tabellen).

Starttabelle

Die Starttabelle ist die erste Tabelle, die beim Extrahieren von Daten verwendet werden soll. Sie können jede Tabelle in der Zugriffsdefinition außer einer Referenztable als Starttabelle angeben. Wenn Sie nicht explizit eine Starttabelle angeben, ist die erste Tabelle in der Tabellenliste die Starttabelle.

Referenztable

Tabellen, aus denen alle Zeilen ausgewählt sind, sofern keine Auswahlkriterien für die Referenztable angegeben werden. Es kann jede Tabelle außer der Starttabelle als Referenztable angegeben werden.

Beziehungen

Beziehungen legen den Traversierungspfad für die Auswahl von Daten aus Tabellen fest. Standardmäßig werden Beziehungen vom übergeordneten zum untergeordneten Element traversiert, aber Sie können die Traversierungsrichtung mit Einstellungen in der Zugriffsdefinition steuern. Beziehungen zwischen Tabellen, die über die Zugriffsdefinition referenziert werden, werden in der Registerkarte für Beziehungen (bis zu 24.000 Tabellen) aufgelistet). Sie können in der Verarbeitung zu verwendende Beziehungen und die Richtung, in der sie traversiert werden, auswählen.

Auswahlkriterien

Auswahlkriterien definieren eine bestimmte Gruppe von Daten in den Tabellen, die in einer Zugriffsdefinition verwendet werden soll. Sie können SQL-Operatoren und -werte angeben und Substitutionsvariablen mit Standardwerten verwenden.

Point-and-shoot

Verwenden Sie eine Point-and-shoot-Liste zum Auswählen bestimmter Zeilen aus einer Starttabelle, die in einen Service aufgenommen werden sollen.

Variablen

Variablen sind benutzerdefinierte Standardwerte, die in einer Zugriffsdefinition angegeben werden. Sie können diese Substitutionsvariablen zum Angeben von Spaltenauswahlkriterien oder zum Erstellen einer SQL-Klausel WHERE verwenden.

Zusätzliche Parameter

Verwenden Sie zusätzliche Parameter zum Extrahieren von Zeilen, die einem bestimmten Spaltenwert in der Starttabelle entsprechen, oder zum Verwenden einer angegebenen Stichprobenrate (jede nte Zeile).

Namenskonventionen

Ein vollständig qualifizierter Zugriffsdefinitionsname hat die Form *identifizier.name*. Dabei gilt Folgendes:
identifizier

Der Zugriffsdefinition zugeordnetes Qualifikationsmerkmal (1 bis 8 Zeichen).

name Der Zugriffsdefinition zugeordneter Basisname (1 bis 12 Zeichen).

Eine logische Gruppe von Namenskonventionen kann die Verwendung für jede Zugriffsdefinition angeben und ihre Organisation für einen einfachen Zugriff ermöglichen.

Zugriffsdefinition erstellen

Mit dem Assistenten **Neue Zugriffsdefinition** können Sie eine Zugriffsdefinition erstellen.

Bevor Sie eine Zugriffsdefinition erstellen können, muss ein Datenspeicheraliasname für die Datenbank vorhanden sein, die die Starttabelle enthält.

Gehen Sie wie folgt vor, um eine Zugriffsdefinition zu erstellen:

1. Erweitern Sie einen Ordner im Repository-Explorer, der die Zugriffsdefinition enthalten soll, klicken Sie mit der rechten Maustaste auf **Zugriffsdefinitionen** und klicken Sie auf **Neue Zugriffsdefinition**. Der Assistent **Neue Zugriffsdefinition** wird geöffnet.
2. Führen Sie die Schritte im Assistenten aus.
Sie müssen einen Datenspeicheraliasnamen und eine Starttabelle aus dem Aliasnamen auswählen. Wählen Sie aus, dass Tabellen zur zugehörigen Starttabelle hinzugefügt werden sollen.

Die neue Zugriffsdefinition wird im Zugriffsdefinitionseditor angezeigt.

Tabellen in einer Zugriffsdefinition verwalten

Mit der Registerkarte **Tabellen** im Zugriffsdefinitionseditor können Sie Tabellen in einer Zugriffsdefinition hinzufügen oder entfernen. Außerdem können Sie Tabellen in zugehörige oder Referenztablelle ändern.

Tabellen einer Zugriffsdefinition hinzufügen

Mit dem Assistenten **Tabellen hinzufügen** können Sie Tabellen in einem Datenspeicheraliasnamen einer Zugriffsdefinition hinzufügen.

Gehen Sie wie folgt vor, um einer Zugriffsdefinition Tabellen hinzuzufügen:

1. Erweitern Sie einen Ordner im Repository-Explorer, der die Zugriffsdefinition enthält, erweitern Sie den Knoten **Zugriffsdefinitionen** und doppelklicken Sie auf die Zugriffsdefinition. Der Zugriffsdefinitionseditor wird geöffnet.
2. Wählen Sie die Registerkarte **Tabellen** aus.
3. Klicken Sie auf **Tabellen hinzufügen**. Der Assistent **Tabellen hinzufügen** wird geöffnet.
4. Führen Sie die Schritte im Assistenten aus.
Sie müssen einen Datenspeicheraliasnamen auswählen, der die hinzuzufügenden Tabellen enthält. Sie können Referenztabellen oder Tabellen, die zu einer ausgewählten Tabelle gehören, hinzufügen.
5. Speichern Sie die Zugriffsdefinition.

Tabellen in Referenz- oder zugehörige Tabellen ändern

Mit dem Zugriffsdefinitionseditor können Sie eine Tabelle in einer Zugriffsdefinition in eine zugehörige oder Referenztabelle ändern.

Gehen Sie wie folgt vor, um Tabellen in Referenz- oder zugehörige Tabellen zu ändern:

1. Erweitern Sie den Ordner im Repository-Explorer, der die Zugriffsdefinition enthält, erweitern Sie den Knoten **Zugriffsdefinitionen** und doppelklicken Sie auf die Zugriffsdefinition. Der Zugriffsdefinitionseditor wird geöffnet.
2. Wählen Sie die Registerkarte **Tabellen** aus.
3. Wählen Sie eine Tabelle aus, die in eine zugehörige oder Referenztabelle geändert werden soll.
4. Klicken Sie auf **In Referenzentität ändern** oder **In zusammengehörige Entität ändern**. In der Spalte **Typ** wird der neue Tabellentyp angezeigt.
5. Speichern Sie die Zugriffsdefinition.

Tabellen aus Zugriffsdefinition entfernen

Mit dem Zugriffsdefinitionseditor können Sie Tabellen aus einer Zugriffsdefinition entfernen.

Gehen Sie wie folgt vor, um Tabellen aus einer Zugriffsdefinition zu entfernen:

1. Erweitern Sie den Ordner im Repository-Explorer, der die Zugriffsdefinition enthält, erweitern Sie den Knoten **Zugriffsdefinitionen** und doppelklicken Sie auf die Zugriffsdefinition. Der Zugriffsdefinitionseditor wird geöffnet.
2. Wählen Sie die Registerkarte **Tabellen** aus.
3. Wählen Sie eine zu entfernende Tabelle aus.
4. Klicken Sie auf **Tabelle entfernen**.
5. Speichern Sie die Zugriffsdefinition.

Traversierungsschritte anzeigen und bearbeiten

Mit dem Fenster **Traversierungsschritte** können Sie die Traversierungsschritte einer Zugriffsdefinition anzeigen und bearbeiten.

Sie können die Tabellen, Beziehungen und die Schritte zum Auswählen von Daten auswerten. Durch diese Auswertung kann sichergestellt werden, dass Sie die gewünschte Gruppe von Daten abrufen.

Die Schritte in einem Prozess können entsprechend den von Ihnen ausgewählten Beziehungen und den von Ihnen angegebenen Kriterien beliebig oft wiederholt werden.

Jede Tabelle kann in aufeinanderfolgenden Schritten noch mehrmals bearbeitet werden. Es können auch Zyklen verwendet werden. Ein Zyklus bewirkt, dass eine Gruppe von Tabellen wiederholt traversiert wird, bis ein vollständiges Durchlaufen des Zyklus nicht mehr zur Auswahl zusätzlicher Zeilen führt.

Wenn eine Beziehung traversiert wird, um übergeordnete Zeilen für bereits ausgewählte untergeordnete Zeilen auszuwählen, werden alle Auswahlkriterien für die übergeordnete Tabelle ignoriert.

Gehen Sie wie folgt vor, um Traversierungsschritte anzuzeigen und zu bearbeiten:

1. Erweitern Sie einen Ordner im Repository-Explorer, der die Zugriffsdefinition enthält, erweitern Sie den Knoten **Zugriffsdefinitionen** und doppelklicken Sie auf die Zugriffsdefinition. Der Zugriffsdefinitionseditor wird geöffnet.
2. Wählen Sie die Registerkarte **Tabellen** aus.
3. Klicken Sie auf **Schritte anzeigen**. Das Fenster **Traversierungsschritte** wird geöffnet.
4. Überprüfen Sie die Schritte und bearbeiten Sie die Reihenfolge, in der Tabellen ausgewählt werden.
5. Klicken Sie auf **OK**.
6. Speichern Sie die Zugriffsdefinition.

Auswahlkriterien verwalten

Nutzen Sie Auswahlkriterien, um den Fokus auf eine bestimmte Gruppe zusammengehöriger Daten zu richten, indem Sie eine SQL-Klausel WHERE definieren und Substitutionsvariablen mit Standardwerten verwenden.

Auswahlkriterien müssen der SQL-Syntax entsprechen und relationale oder logische Operatoren enthalten. Logische Operatoren und die Syntax variieren je nach Datenbankverwaltungssystem. Nähere Informationen finden Sie in der Dokumentation des entsprechenden Datenbankverwaltungssystems.

Zum Auswählen der gewünschten Datengruppe für eine Tabelle benötigen Sie möglicherweise eine Kombination aus den logischen Operatoren AND und OR.

Wenn ebenfalls eine Point-and-shoot-Liste verwendet wird, wird sie logisch in die anderen Kriterien mit einem logischen Operator OR aufgenommen.

Wenn eine Beziehung vom untergeordneten zum übergeordneten Element traversiert wird, werden alle Auswahlkriterien für die übergeordnete Tabelle ignoriert.

Datumskriterien

Ein eindeutiger Operator, **BEFORE**, ermöglicht das Auswählen von Daten auf Basis von Werten in einer DATE-Spalte. Dieser Operator hat folgende Syntax:

BEFORE (*nD nW nM nY*)

Mit den Argumenten **D**, **W**, **M** und **Y** in einer beliebigen Kombination können Sie die Anzahl Tage, Wochen, Monate oder Jahre angeben, die vom Datum zur Ausführungszeit abgezogen werden. Wenn keine Argumente angegeben werden, wird das aktuelle Datum verwendet. Zeilen mit einem Datum, das älter als das berechnete Datum ist, werden extrahiert oder archiviert. Der Multiplikator *n* ist eine ganze Zahl, der wahlweise + oder - vorangestellt werden kann.

Auswahlkriterien definieren

Verwenden Sie das Fenster **Auswahlkriterien bearbeiten** zum Definieren von Auswahlkriterien mit einer SQL-Klausel WHERE.

Gehen Sie wie folgt vor, um Auswahlkriterien zu definieren:

1. Erweitern Sie einen Ordner im Repository-Explorer, der die Zugriffsdefinition enthält, erweitern Sie den Knoten **Zugriffsdefinitionen** und doppelklicken Sie auf die Zugriffsdefinition. Der Zugriffsdefinitionseditor wird geöffnet.
2. Wählen Sie die Registerkarte **Tabellen** aus.
3. Wählen Sie die Tabelle aus, der Sie Auswahlkriterien hinzufügen wollen.
4. Klicken Sie auf **Auswahlkriterien bearbeiten**. Das Fenster **Auswahlkriterien bearbeiten** wird geöffnet und zeigt eine Klausel WHERE für die Tabelle.
5. Geben Sie Auswahlkriterien ein.
Zur Anzeige einer Liste der Spalten und Operatoren klicken Sie mit der rechten Maustaste auf die Klausel WHERE und klicken Sie auf die Inhaltshilfe.
Wählen Sie zum Hinzufügen einer Variablen einen Variablenbegrenzer aus und klicken Sie auf **Variablen einfügen**.
6. Klicken Sie auf **Syntax prüfen**, um die Syntax zu prüfen und Fehler zu ermitteln.
6. Klicken Sie auf **OK**.
Wenn die Syntax nicht gültig ist, wird eine entsprechende Nachricht angezeigt. Sie können die Kriterien nicht speichern, wenn sie Fehler enthalten.
7. Speichern Sie die Zugriffsdefinition.

Optionen für die Datengruppierung angeben

Mit der Registerkarte **Datengruppeneigenschaften** im Zugriffsdefinitionseditor können Sie eine Gruppenauswahl angeben, bei der eine Anzahl von Zeilen auf der Basis von Werten in einer bestimmten Spalte in der Starttabelle extrahiert wird. Zeilen in jeder Gruppe haben denselben Wert in der ausgewählten Spalte.

Wenn Kriterien in Kombination mit Gruppenauswahlparametern verwendet werden, werden die Kriterien zuerst angewendet, wobei Gruppenauswahlparameter auf das Ergebnis angewendet werden.

Gehen Sie wie folgt vor, um Datengruppierungsoptionen anzugeben:

1. Erweitern Sie einen Ordner im Repository-Explorer, der die Zugriffsdefinition enthält, erweitern Sie den Knoten **Zugriffsdefinitionen** und doppelklicken Sie auf die Zugriffsdefinition. Der Zugriffsdefinitionseditor wird geöffnet.
2. Wählen Sie die Registerkarte **Datengruppeneigenschaften** aus.
3. Wählen Sie eine Spalte aus und geben Sie Gruppierungsoptionen an.
4. Speichern Sie die Zugriffsdefinition.

Optionen für Datenstichproben und Zeilenbeschränkungen angeben

Mit der Registerkarte **Tabellen** im Zugriffsdefinitionseditor können Sie Optionen zum Extrahieren von Zeilenstichproben oder zum Begrenzen der Anzahl zu extrahierender Zeilen angeben.

Wenn mit einer Point-and-shoot-Liste Starttabellenzeilen ausgewählt werden, ignoriert der Extraktionservice Parameter für Datenstichproben oder Zeilenbeschränkungen für die Starttabelle.

Gehen Sie wie folgt vor, um Optionen für Datenstichproben und Zeilenbeschränkungen anzugeben:

1. Erweitern Sie einen Ordner im Repository-Explorer, der die Zugriffsdefinition enthält, erweitern Sie den Knoten **Zugriffsdefinitionen** und doppelklicken Sie auf die Zugriffsdefinition. Der Zugriffsdefinitionseditor wird geöffnet.
2. Wählen Sie die Registerkarte **Tabellen** aus.
3. Geben Sie folgende Optionen ein:

Alle n Zeilen

Geben Sie einen numerischen Wert zum Festlegen eines Stichprobenfaktors für eine Tabelle ein. Wenn Sie beispielsweise 5 eingeben, extrahiert der Prozess jede fünfte Zeile in der Tabelle beginnend mit der fünften Zeile. Gültige Werte sind 1 bis 9999.

Zeilenbegrenzung

Geben Sie einen numerischen Wert zum Begrenzen der Anzahl von Zeilen, die aus einer Tabelle extrahiert werden, ein. Gültige Werte sind 1 bis 999999999999.

4. Speichern Sie die Zugriffsdefinition.

Beziehungen in einer Zugriffsdefinition verwalten

Mit der Registerkarte **Beziehungen** im Zugriffsdefinitionseditor können Sie Beziehungen in einer Zugriffsdefinition hinzufügen oder entfernen. Sie können außerdem Beziehungstraversierung und Tabellenzugriffsoptionen angeben.

Beziehungstraversierungsoptionen auswählen

Mit dem Zugriffsdefinitionseditor können Sie Beziehungstraversierungsoptionen in einer Zugriffsdefinition verwalten.

Sie können die folgenden Optionen angeben:

- Die maximale für eine Beziehung auszuwählende Anzahl von Zeilen aus der untergeordneten Tabelle.
- Die maximale Anzahl in einem Mal ausgeführter Schlüsselsuchen für eine Tabelle.

Durch die Erhöhung des Grenzwerts für die Schlüsselsuche kann sich die Leistung erheblich verbessern. Wenn Sie z. B. 5 als Grenzwert für die Schlüsselsuche angeben und der Schlüssel eine einzige Spalte hat, werden 5 Schlüsselwerte in einer einzigen Anforderung an das Datenbankverwaltungssystem gesucht.

- Die Traversierungspfade:
 - Die Beziehung vom untergeordneten zum übergeordneten Element traversieren, um eine übergeordnete Zeile für jede untergeordnete Zeile auszuwählen, damit die relationale Integrität sichergestellt ist. (Option 1)
 - Die Beziehung traversieren, um zusätzliche untergeordnete Zeilen für jede übergeordnete Zeile auszuwählen, die als Ergebnis einer Traversierung von untergeordneter zu übergeordneter Zeile ausgewählt wurde. (Option 2)

Optionen 1 und 2 sind relevant, wenn die Starttabelle eine untergeordnete Tabelle ist oder wenn eine Tabelle mehrere übergeordnete Tabellen hat, die von der Zugriffsdefinition referenziert werden.

Option 2 ist nur relevant, wenn eine Beziehung vom untergeordneten zum übergeordneten Element traversiert wird. Wenn z. B. ein Prozess vom untergeordneten zum übergeordneten Element traversiert (Option 1) und eine übergeordnete Zeile ausgewählt wird, bewirkt Option 2, dass der Prozess zusätzliche untergeordnete Zeilen für diese übergeordnete Zeile auswählt.

Wenn Sie Option 2 für eine Beziehung auswählen, erwägen Sie die Angabe eines Grenzwerts für untergeordnete Elemente bei der Anzahl zu extrahierender untergeordneter Zeilen.

Gehen Sie wie folgt vor, um Beziehungstraversierungsoptionen in einer Zugriffsdefinition zu verwalten:

1. Erweitern Sie den Ordner im Repository-Explorer, der die Zugriffsdefinition enthält, erweitern Sie den Knoten **Zugriffsdefinitionen** und doppelklicken Sie auf die Zugriffsdefinition. Der Zugriffsdefinitionseditor wird geöffnet.
2. Wählen Sie die Registerkarte **Beziehungen** und dann die Registerkarte **Traversierungsoptionen** aus.
3. Geben Sie Traversierungsoptionen ein.
4. Speichern Sie die Zugriffsdefinition.

Tabellenzugriffsoptionen auswählen

Mit dem Zugriffsdefinitionseditor können Sie die Methode für den Zugriff auf die übergeordnete oder untergeordnete Tabelle für jede Beziehung auswählen.

Sie können die folgenden Optionen angeben:

Standard

Optim ermittelt die beste Methode. Es wird eine Schlüsselsuche verwendet, wenn ein Index für das Datenbankverwaltungssystem verfügbar ist, und ein Suchlauf, wenn kein Index verfügbar ist. Wenn allerdings auf einen großen Teil der Tabelle zugegriffen wird, ist ein Suchlauf der Standardwert, auch wenn ein Index vorhanden ist.

Überprüfung erzwingen

Alle Zeilen in einer Tabelle auf einmal lesen.

Schlüsselsuche erzwingen

Zeilen mit einer WHERE-Klausel suchen, um Primär- oder Fremdschlüsselwerte zu finden.

Anmerkung: Überschreiben Sie die Standardmethode nur, wenn die statistischen Informationen im Prozessbericht darauf hinweisen, dass die Standardmethode weniger effizient ist.

Gehen Sie wie folgt vor, um Tabellenzugriffsoptionen zu verwalten:

1. Erweitern Sie den Ordner im Repository-Explorer, der die Zugriffsdefinition enthält, erweitern Sie den Knoten **Zugriffsdefinitionen** und doppelklicken Sie auf die Zugriffsdefinition. Der Zugriffsdefinitionseditor wird geöffnet.
2. Wählen Sie die Registerkarte **Beziehungen** und dann die Registerkarte **Tabellenzugriff** aus.

3. Geben Sie Tabellenzugriffsoptionen ein.
4. Speichern Sie die Zugriffsdefinition.

Variablen in einer Zugriffsdefinition verwalten

Variablen sind benutzerdefinierte Standardwerte, die in einer Zugriffsdefinition angegeben werden. Sie können mit diesen Substitutionsvariablen eine SQL-Klausel WHERE erstellen.

Durch Zuweisen von Variablen können Sie jedes Mal, wenn die Zugriffsdefinition verarbeitet wird, Werte für die Variablen bereitstellen. Als Option können Sie Standardwerte für Substitutionsvariablen bereitstellen. Die Variablen werden mit der Zugriffsdefinition gespeichert.

Standardwerte

Wenn Sie eine Variable erstellen, können Sie einen optionalen Standardwert eingeben, der verwendet werden soll, wenn zur Ausführungszeit kein Wert für die Variable angegeben wird.

Standardwerte müssen den entsprechenden Datentyp und die geeignete Größe für die Spalte haben und der SQL-Syntax entsprechen. Nehmen Sie beispielsweise an, dass ein Variablenname **ST** (Staat) ist, der Variablenbegrenzer ein Doppelpunkt (:) ist und die Spalte Zeichendaten erfordert.

- Wenn Sie die Variable mit einfachen Anführungszeichen in den Auswahlkriterien verwenden, müssen Sie den Wert ohne einfache Anführungszeichen angeben:

Auswahlkriterien	Wert
= 'ST'	CA

- Wenn Sie die Variable ohne einfache Anführungszeichen in den Auswahlkriterien verwenden, müssen Sie den Wert mit einfachen Anführungszeichen angeben:

Auswahlkriterien	Wert
= :ST	'CA'

Anmerkung: Standardwerte werden erst zur Ausführungszeit überprüft. Wenn ein Wert einen ungültigen Datentyp oder eine ungültige Größe für die Spalte hat oder nicht der SQL-Syntax entspricht, können Verarbeitungsfehler auftreten.

Eingabeaufforderung

Sie müssen den Text eingeben, der während der Ausführung zur Eingabe des Variablenwerts auffordert. Geben Sie die Eingabeaufforderung genau so ein, wie sie im Prozessanforderungsdialog angezeigt werden soll (max. 50 Zeichen). Diese Eingabeaufforderung wird angezeigt, bevor Sie den Prozess ausführen.

Variable erstellen

Mit dem Fenster **Variable hinzufügen** können Sie eine Variable in einer Zugriffsdefinition erstellen.

Gehen Sie wie folgt vor, um eine Variable zu erstellen:

1. Erweitern Sie den Ordner im Repository-Explorer, der die Zugriffsdefinition enthält, erweitern Sie den Knoten **Zugriffsdefinitionen** und doppelklicken Sie auf die Zugriffsdefinition. Der Zugriffsdefinitionseditor wird geöffnet.
2. Wählen Sie die Registerkarte **Variablen** aus.
3. Klicken Sie auf **Hinzufügen**. Das Fenster **Variable hinzufügen** wird geöffnet.
4. Geben Sie Informationen für die Variable ein. Sie müssen einen Variablennamen und Eingabeaufforderungstext eingeben. Klicken Sie auf **OK**.
5. Speichern Sie die Zugriffsdefinition.

Point-and-shoot-Listen verwalten

Verwenden Sie eine Point-and-shoot-Liste zum Auswählen bestimmter Zeilen aus einer Starttabelle, die in einen Service aufgenommen werden sollen.

Eine Point-and-shoot-Liste ist in einer Zugriffsdefinition enthalten. Wenn Sie mit Point-and-shoot Zeilen aus der Starttabelle auswählen, werden die Primärschlüssel für diese Zeilen in einer Point-and-shoot-Datei gespeichert. Die Serviceanforderung verwendet die Primärschlüssel zum Ermitteln der Zeilen, die zuerst verarbeitet werden sollen.

Point-and-shoot-Listendatei erstellen

Mit dem Fenster **Neue Point-and-shoot-Datei** können Sie eine Point-and-shoot-Listendatei für eine Tabelle in einem Datenspeicheraliasnamen erstellen.

Gehen Sie wie folgt vor, um eine Point-and-shoot-Listendatei zu erstellen:

1. Erweitern Sie im Repository-Explorer den Knoten **Datenspeicheraliasnamen** und den Aliasnamen des Zieldatenspeichers, bis die Tabelle angezeigt wird, der Sie die Point-and-shoot-Datei hinzufügen wollen.
2. Erweitern Sie die Tabelle, klicken Sie mit der rechten Maustaste auf **Point-and-shoot-Listen** und klicken Sie auf **Neuer Point-and-shoot**. Das Fenster **Neue Point-and-shoot-Datei** wird geöffnet.
3. Geben Sie einen Dateinamen ein und klicken Sie auf **OK**.

Die Point-and-shoot-Listendatei wird unter **Point-and-shoot-Listen** angezeigt.

Sie müssen Zeilen oder die Point-and-shoot-Liste mit dem Point-and-shoot-Editor auswählen.

Point-and-shoot-Liste auswählen

Mit dem Zugriffsdefinitionseditor können Sie eine Point-and-shoot-Liste für eine Zugriffsdefinition auswählen.

Damit eine Point-and-shoot-Liste ausgewählt werden kann, muss eine Point-and-shoot-Datei für die Starttabelle verfügbar sein.

Gehen Sie wie folgt vor, um eine Point-and-shoot-Liste auszuwählen:

1. Erweitern Sie den Ordner im Repository-Explorer, der die Zugriffsdefinition enthält, erweitern Sie den Knoten **Zugriffsdefinitionen** und doppelklicken Sie auf die Zugriffsdefinition. Der Zugriffsdefinitionseditor wird geöffnet.
2. Wählen Sie die Registerkarte **Point-and-shoot** aus.
3. Wählen Sie eine Point-and-shoot-Liste aus.
4. Speichern Sie die Zugriffsdefinition.

Mit Optim-Beziehungen arbeiten

Optim verwendet Beziehungen, um die Daten zu ermitteln, die aus zugehörigen Tabellen abgerufen werden sollen, und basiert auf in der Datenbank definierten Beziehungen, soweit sie vorhanden sind. Sie können allerdings auch Optim-Beziehungen definieren, um Beziehungen in der Datenbank zu ergänzen.

Mit Optim-Beziehungen wird eine Anzahl von Datenbankbeschränkungen gelockert. Beispiel:

- Primärschlüssel und Fremdschlüssel sind nicht erforderlich.
- Einander entsprechende Spalten müssen nicht gleich, nur noch kompatibel sein.
- Mindestens eine Spalte eines Paares einander entsprechender Spalten muss durch den Spaltennamen angegeben werden. Allerdings kann der Wert in der zweiten Spalte über einen Ausdruck berechnet oder definiert werden. Ausdrücke können Zeichenfolgeliterale, numerische Konstanten, NULL, Verkettung und Unterzeichenfolgen enthalten.

Die flexibleren Optim-Beziehungen werden als erweiterte Beziehungen bezeichnet. Erweiterte Beziehungen können implizite oder anwendungsverwaltete Beziehungen in Ihrer Datenbank replizieren und es Ihnen dadurch ermöglichen, Gruppen relationaler Daten auf dieselbe Weise wie in Ihrer Produktionsumgebung zu bearbeiten.

Zusätzlich kann eine Optim-Beziehung im Optim-Verzeichnis gespeichert werden als:

- **explizite** Beziehung, die für ein einzelnes Paar von Tabellen verwendet wird.
- **generische** Beziehung, die für mindestens ein Paar von Tabellen verwendet wird, die denselben Basisnamen, dieselben Spaltennamen und -attribute, aber unterschiedliche Ersteller-IDs haben.

Generische Beziehungen sind nützlich, wenn mehrere Gruppen von Tabellen sich nur in der Ersteller-ID unterscheiden. (Beispielsweise kann in einer Testumgebung jeder Programmierer eine separate Kopie derselben Produktionstabellen verwenden. Jede Tabellengruppe kann durch die Ersteller-ID unterschieden werden.) Über generische Beziehungen können Sie eine Gruppe von Beziehungen definieren, die für alle Tabellengruppen gilt. Außerdem treten die generischen Beziehungen automatisch in Kraft, wenn eine Gruppe dieser Tabellen hinzugefügt wird.

Einschränkungen

Obwohl die Regeln zur Erstellung von Optim-Beziehungen flexibler als die Regeln zur Erstellung durch Datenbanken definierter Beziehungen sind, bestehen einige Einschränkungen:

- Sie müssen mindestens eine Spalte für jede Tabelle in der Beziehung referenzieren.
- Sie können maximal 64 Spalten für jede Tabelle in der Beziehung referenzieren.
- Sie können kein Literal oder keine Konstante mit einem Literal oder einer Konstante abgleichen.
- Sie können keine Spalte mit einem großen Objekt (LOB - Large Object) oder einer SQL-Variante verwenden.
- Die Gesamtlänge aller Werte, die in der übergeordneten oder der untergeordneten Tabelle verwendet werden, darf 3584 Byte nicht überschreiten.
- Sie können keine Beziehung mithilfe einer SQL-Variantspalte erstellen.

In einer Beziehungsdefinition für eine Mehrbyte- oder Unicode-Datenbank gelten folgende Einschränkungen:

- Sie können die Unterzeichenfolgefunktion nicht verwenden.
- Sie können Zeichendaten (CHAR oder NCHAR) nicht mit Binärdaten (RAW) verketten.
- Wenn Oracle-Zeichensemantik für eine CHAR-Spalte verwendet wird, müssen alle CHAR-Spalten in der Beziehung Zeichensemantik oder den Datentyp NCHAR haben.

BEISPIELE:

Übergeordnetes Objekt	Unterstützt/ Nicht unterstützt	Untergeordnetes Objekt	Beschreibung
CHAR	→	CHAR	Unterstützt, Semantik muss stimmen
NCHAR	→	NCHAR	Unterstützt, Semantik irrelevant.
CHAR	↗	NCHAR	Nicht unterstützt
CHAR	→	VARCHAR	Unterstützt, Semantik muss stimmen

Übergeordnetes Objekt	Unterstützt/ xy Nicht unter- stützt	Untergeordnetes Objekt	Beschreibung
NCHAR	⇒	NVARCHAR	Unterstützt, Semantik irrelevant.
CHAR NCHAR	⇒	NCHAR CHAR	Unterstützt bei Zeichensemantik; nicht unterstützt bei Bytesemantik.
CHAR NCHAR	⇒	NCHAR NCHAR	Unterstützt bei Zeichensemantik; nicht unterstützt bei Bytesemantik.

Kompatibilitätsregeln für Beziehungen

Wenn Sie eine Optim-Beziehung definieren, müssen die entsprechenden Werte kompatibel sein.

Spaltentyp	Ist kompatibel mit
Zeichenspalte	<ul style="list-style-type: none"> • Zeichenspalte • Numerische Spalte • Zeichenfolgeliteral • Zeichenausdruck
Numerische Spalte	<ul style="list-style-type: none"> • Numerische Spalte • Numerische Konstante • Zeichenspalte
Binärspalte	<ul style="list-style-type: none"> • Binärspalte • Hexadezimaler Literal • Binärausdruck
Boolesche Spalte	<ul style="list-style-type: none"> • Boolesche Spalte • Boolesche Konstante (True oder False).
Datums-/Uhrzeitspalte	Datums-/Uhrzeitspalte
Datumsspalte	Datumsspalte
Uhrzeitspalte	Uhrzeitspalte
Intervallspalte	Intervallspalte

Anmerkung:

- Bei der Verarbeitung wird ein Wert in den Datentyp konvertiert, der zum Auswählen verwandter Zeilen erforderlich ist. Standardmäßig ergibt die Konvertierung eines numerischen Werts in einen Zeichendatentyp rechtsbündig ausgerichtete Daten mit führenden Nullen. Spezielle Registry-Einstellungen ermöglichen Ihnen das Ändern des Standardwerts auf linksbündige Ausrichtung mit führenden oder nachfolgenden Leerzeichen. Außerdem erfordert eine Zuordnung zwischen Zeichenwerten und numerischen Spalten eine Skalierung ungleich 0 für die numerische Spalte.
- Sie können NULL für jede für Nullwerte auswählbare Spalte verwenden.
- Unicode- oder Mehrbytespalten müssen denselben Zeichensatz haben.

Datentypen

Die folgenden Datenklassen und zugehörigen Datentypen werden unterstützt. Diese Datenklassen sind für die Datenkompatibilität von Bedeutung, wenn Sie Spaltenwerte in Beziehungen verwenden.

Zeichenklasse

Datenbankverwaltungssystem	Datentypen
DB2	CHAR, VARCHAR, CLOB
Oracle	CHAR, VARCHAR2, LONG, CLOB, NCLOB, NCHAR, NVARCHAR
Sybase ASE	CHAR, VARCHAR, TXT
SQL Server	CHAR, VARCHAR, TXT
Informix	CHAR, VARCHAR, TXT

Anmerkung: Spalten mit Einzelbytezeichen sind nicht kompatibel mit Spalten mit Mehrbyte- oder Unicode-Zeichen.

Numerische Klasse

Datenbankverwaltungssystem	Datentypen
DB2	INTEGER, SMALLINT, DECIMAL, FLOAT, DOUBLE
Oracle	NUMBER, FLOAT
Sybase ASE	TINYINT, INT, SMALLINT, DECIMAL, FLOAT, REAL, MONEY, SMALL MONEY
SQL Server	TINYINT, INT, SMALLINT, DECIMAL, FLOAT, REAL, MONEY, SMALL MONEY
Informix	INTEGER, SMALLINT, DECIMAL, FLOAT, REAL, DOUBLE PRECISION, SMALLFLOAT, SERIAL, MONEY, NUMERIC

Binärklasse

Datenbankverwaltungssystem	Datentypen
DB2	CHAR (für Bitdaten), VARCHAR (for Bit Data), BLOB
Oracle	RAW, LONG RAW
Sybase ASE	BINARY, VARBINARY, IMAGE
SQL Server	BINARY, VARBINARY, IMAGE
Informix	BYTE

Boolesche Klasse

Datenbankverwaltungssystem	Datentypen
Sybase ASE	BOOLEAN (TRUE oder FALSE)

Datum/Uhrzeit

Datenbankverwaltungssystem	Datentypen
DB2	TIMESTAMP
Oracle	DATE, TIMESTAMP, TIMESTAMP WITH LOCAL TIME ZONE, TIMESTAMP WITH TIME ZONE
Sybase ASE	DATETIME, SMALL DATE TIME
SQL Server	DATETIME, SMALL DATE TIME
Informix	DATE, DATETIME

Datumsklasse

Datenbankverwaltungssystem	Datentypen
DB2	DATE
Oracle	DATE
Informix	DATE

Zeitklasse

Datenbankverwaltungssystem	Datentypen
DB2	TIME

Intervallklasse

Datenbankverwaltungssystem	Datentypen
Oracle	YEAR/MONTH INTERVAL, DAY/SECOND INTERVAL
Informix	YEAR/MONTH INTERVAL, DAY/TIME INTERVAL

Optim-Beziehung erstellen

Mit dem Assistenten **Neue Optim-Beziehung** können Sie eine neue Optim-Beziehung erstellen.

Bevor Sie eine Optim-Beziehung erstellen können, muss ein Datenspeicheraliasname für die Tabellen in der Beziehung vorhanden sein.

Gehen Sie wie folgt vor, um eine Optim-Beziehung zu erstellen:

1. Klicken Sie im Repository-Explorer mit der rechten Maustaste auf **Optim-Beziehungen** und klicken Sie auf **Neue Optim-Beziehung**. Der Assistent **Neue Optim-Beziehung** wird geöffnet.
2. Führen Sie die Schritte im Assistenten aus.

Sie müssen die übergeordnete und die untergeordnete Tabelle in der Beziehung auswählen.

Die neue Optim-Beziehung wird im Beziehungseditor angezeigt.

Sie müssen mit dem Beziehungseditor einen Spaltenausdruck definieren, der die über- und untergeordneten Spalten in der Beziehung angibt.

Optim-Beziehung bearbeiten

Mit dem Beziehungseditor können Sie einen Spaltenausdruck definieren, der die über- und untergeordneten Spalten in einer Optim-Beziehung angibt. Sie können mit dem Editor auch eine generische Beziehung erstellen und Spalten bearbeiten.

Spaltenausdruck erstellen

Im Fenster **Spaltenausdruck hinzufügen** Sie einen Spaltenausdruck erstellen, der die über- und untergeordneten Spalten in einer Optim-Beziehung angibt.

Gehen Sie wie folgt vor, um einen Spaltenausdruck in einer Optim-Beziehung zu erstellen:

1. Erweitern Sie im Repository-Explorer den Knoten **Optim-Beziehungen** und doppelklicken Sie auf die Beziehung, die Sie bearbeiten wollen. Der Beziehungseditor wird geöffnet.
2. Klicken Sie auf **Spaltenausdruck hinzufügen....** Das Fenster **Spaltenausdruck hinzufügen** wird geöffnet.
3. Klicken Sie auf **Übergeordnete Spalte auswählen**. Das Fenster **Übergeordnete Spalte auswählen** wird geöffnet.

4. Wählen Sie eine übergeordnete Spalte aus und klicken Sie auf **OK**, um zum Fester **Spaltenausdruck hinzufügen** zurückzukehren.
5. Klicken Sie auf **Untergeordnete Spalte auswählen**. Das Fester **Untergeordnete Spalte auswählen** wird geöffnet.
6. Wählen Sie eine untergeordnete Spalte aus und klicken Sie auf **OK**, um zum Fester **Spaltenausdruck hinzufügen** zurückzukehren.
7. Klicken Sie auf **OK**, um zum Editor zurückzukehren. Die übergeordnete Spalte und die untergeordnete Spalte werden jetzt im Editor aufgelistet.
8. Speichern Sie die Beziehung.

Spalten in einer Beziehung bearbeiten

Mit den Fenstern **Übergeordnete Spalte auswählen** und **Untergeordnete Spalte auswählen** können Sie die Spalten in einer Optim-Beziehung bearbeiten.

Gehen Sie wie folgt vor, um die Spalten in einer Optim-Beziehung zu bearbeiten:

1. Erweitern Sie im Repository-Explorer den Knoten **Optim-Beziehungen** und doppelklicken Sie auf die Beziehung, die Sie bearbeiten wollen. Der Beziehungseditor wird geöffnet.
2. Wählen Sie die Zeile aus, die eine zu bearbeitende Spalte enthält. Gehen Sie wie folgt vor, um eine übergeordnete Spalte zu bearbeiten:
 - a. Klicken Sie auf **Übergeordnete Spalte auswählen**. Das Fenster **Übergeordnete Spalte auswählen** wird geöffnet.
 - b. Wählen Sie eine übergeordnete Spalte aus und klicken Sie auf **OK**, um zum Editor zurückzukehren.

Gehen Sie wie folgt vor, um eine untergeordnete Spalte zu bearbeiten:

- a. Klicken Sie auf **Untergeordnete Spalte auswählen**. Das Fester **Untergeordnete Spalte auswählen** wird geöffnet.
- b. Wählen Sie eine untergeordnete Spalte aus und klicken Sie auf **OK**, um zum Editor zurückzukehren.

Die ausgewählten Elemente werden im Editor aufgelistet.

3. Speichern Sie die Beziehung.

Spaltenreihenfolge in einer Beziehung ändern

Mit dem Beziehungseditor können Sie die Reihenfolge von Spalten in einer Optim-Beziehung ändern.

Gehen Sie wie folgt vor, um die Reihenfolge von Spalten in einer Optim/Beziehung zu ändern:

1. Erweitern Sie im Repository-Explorer den Knoten **Optim-Beziehungen** und doppelklicken Sie auf die Beziehung, die Sie bearbeiten wollen. Der Beziehungseditor wird geöffnet.
2. Wählen Sie die Zeile aus, die Spalten enthält, deren Reihenfolge geändert werden soll.
3. Klicken Sie auf **Zeile nach oben versetzen** oder **Zeile nach unten versetzen**, um die Reihenfolge zu ändern.
4. Speichern Sie die Beziehung.

Generische Beziehung erstellen

Mit dem Beziehungseditor können Sie eine generische Optim-Beziehung erstellen.

Einige Datenbanken enthalten Gruppen von Tabellen, die mit Ausnahme der Ersteller-ID identisch sind. Anstatt eine Beziehung für jede Tabellengruppe zu definieren, können Sie auch eine generische Beziehung definieren, die unabhängig von der Ersteller-ID für alle Gruppen von Tabellen mit dem gleichen Basisnamen gilt. Sie können die Basistabellen in einer generischen Beziehung modifizieren.

Gehen Sie wie folgt vor, um eine generische Optim-Beziehung zu erstellen:

1. Erweitern Sie im Repository-Explorer den Knoten **Optim-Beziehungen** und doppelklicken Sie auf die Beziehung, die Sie bearbeiten wollen. Der Beziehungseditor wird geöffnet.
2. Klicken Sie auf **Generisch**.
3. Speichern Sie die Beziehung.

Mit Optim-Primärschlüsseln arbeiten

Ein Primärschlüssel ist die Spalte(n) mit Werten, die jede Zeile in einer Tabelle eindeutig identifizieren. Eine Datenbanktabelle muss einen Primärschlüssel haben, damit Optim Daten in ihr einfügen, aktualisieren, wiederherstellen oder löschen kann. Optim verwendet Primärschlüssel, die für die Datenbank definiert sind. Sie können allerdings auch Optim-Primärschlüssel definieren, um die Primärschlüssel in der Datenbank zu ergänzen.

Ein Primärschlüssel wird benötigt

- in jeder Tabelle, die in einem Prozess mehrmals bearbeitet wird, z. B. eine untergeordnete Tabelle, die mindestens zwei in der Zugriffsdefinition referenzierte übergeordnete Tabellen hat.
- zur Aktivierung der Point-and-shoot-Funktion für eine Starttabelle.

Anmerkung: Wenn ein Primärschlüssel nicht definiert, aber zum Ausführen einer bestimmten Task erforderlich ist, wird eine Fehlermeldung angezeigt.

Typen von Optim-Primärschlüsseln

Sie können zwei Typen von Primärschlüsseln definieren:

- Ein **expliziter** Primärschlüssel wird für eine einzelne Tabelle angewendet.
- Ein **generischer** Primärschlüssel wird für alle Tabellen angewendet, die denselben Basisnamen sowie dieselben Spaltennamen und Attributspezifikationen, aber unterschiedliche Ersteller-IDs haben.

Es gibt keinen Unterschied in Funktion oder Darstellung zwischen generischen und expliziten Primärschlüsseln. Wenn eine Tabelle allerdings über Schlüssel beider Typen verfügt, wird der explizite Primärschlüssel verwendet.

Namenskonventionen

Der vollständig qualifizierte Name eines Primärschlüssels ist identisch mit dem vollständig qualifizierten Namen der Datenbanktabelle, für die er definiert ist. Dieser Name besteht aus folgenden Komponenten: *alias.creatorid.tablename*.

alias Aliasname, der die Datenbank angibt, in der die Tabelle gespeichert wird (1 bis 12 Zeichen).

creatorid

Der Tabelle zugeordnete Ersteller-ID (1 bis 64 Zeichen).

tablename

Basistabellenname (1 bis 64 Zeichen).

Anmerkung:

- Die kombinierte Gesamtlänge der Spalten für einen Primärschlüssel ist auf 3584 Byte beschränkt.

Primärschlüssel erstellen

Mit dem Assistent **Neuer Primärschlüssel** können Sie einen neuen Primärschlüssel für eine Tabelle erstellen.

Bevor Sie einen Primärschlüssel erstellen können, muss ein Datenspeicheraliasname für die Tabelle vorhanden sein.

Gehen Sie wie folgt vor, um einen Primärschlüssel zu erstellen:

1. Klicken Sie im Repository-Explorer mit der rechten Maustaste auf **Optim-Primärschlüssel** und klicken Sie auf **Neuer Primärschlüssel**. Der Assistent **Neuer Primärschlüssel** wird geöffnet.
2. Führen Sie die Schritte im Assistenten aus.
Sie müssen eine Tabelle auswählen.

Der neue Primärschlüssel wird im Primärschlüsseeditor angezeigt.

Sie müssen mit dem Primärschlüsseeditor Primärschlüsselspalten für die Tabelle auswählen.

Primärschlüssel bearbeiten

Mit dem Primärschlüsseeditor können Sie einen Primärschlüssel für eine Tabelle bearbeiten. Sie können Schlüsselspalten auswählen und einen generischen Primärschlüssel definieren.

Primärschlüsselspalten auswählen

Mit dem Primärschlüsseeditor können Sie Primärschlüsselspalten für eine Tabelle auswählen. Sie können auch die Reihenfolge der Schlüsselspalten festlegen.

Gehen Sie wie folgt vor, um Primärschlüsselspalten für eine Tabelle auszuwählen:

1. Erweitern Sie im Repository-Explorer den Knoten **Optim-Primärschlüssel** und doppelklicken Sie auf die Beziehung, die Sie bearbeiten wollen. Der Primärschlüsseeditor wird geöffnet.
2. Wählen Sie in der Liste **Verfügbare Spalten** die Schlüsselspalten aus.
3. Klicken Sie auf >>, um die Spalten in der Liste **Schlüsselspalten** zu verschieben.
Gehen Sie wie folgt vor, um die Reihenfolge der Schlüsselspalten festzulegen:
 - a. Wählen Sie in der Liste **Schlüsselspalten** eine zu verschiebende Spalte aus.
 - b. Klicken Sie auf **Nach oben versetzen**, um eine Spalte in der Reihenfolge nach oben zu versetzen, oder auf **Nach unten versetzen**, um eine Spalte in der Reihenfolge nach unten zu versetzen.
4. Speichern Sie den Primärschlüssel.

Generischen Primärschlüssel erstellen

Mit dem Primärschlüsseeditor können Sie einen generischen Optim-Primärschlüssel erstellen.

Einige Datenbanken enthalten Gruppen von Tabellen, die mit Ausnahme der Ersteller-ID identisch sind. Anstatt einen expliziten Primärschlüssel für jede Tabelle zu definieren, können Sie einen generischen Primärschlüssel unabhängig von der Ersteller-ID für alle Tabellen definieren, die denselben Basisnamen haben.

Gehen Sie wie folgt vor, um einen generischen Optim-Primärschlüssel zu erstellen:

1. Erweitern Sie im Repository-Explorer den Knoten **Optim-Primärschlüssel** und doppelklicken Sie auf die Beziehung, die Sie bearbeiten wollen. Der Primärschlüsseeditor wird geöffnet.
2. Klicken Sie auf **Generisch**.
3. Speichern Sie den Primärschlüssel.

Mit Tabellenzuordnungen arbeiten

Mit einer Tabellenzuordnung können Sie Spezifikationen für das Korrelieren von Quellen- und Zieltabellen kompatibler Daten definieren. Sie können Tabellen zuordnen, die verschiedene Namen haben, Tabellennamen ändern, Tabellen aus einem Prozess ausschließen oder Spaltenzuordnungen für eine bessere Kontrolle über die Daten einschließen.

Mit einer Tabellenzuordnung können Sie Folgendes ausführen:

- Die Platzierung von Daten in einem Konvertierungs-, Einfüge- oder Ladeservice steuern.

- Mindestens eine Tabelle aus einem Konvertierungs-, Einfüge- oder Ladeservice ausschließen.
- Eine Spaltenzuordnung einbeziehen.

Je nach Prozess, für den Sie eine Tabellenzuordnung verwenden, werden die beiden Gruppen von Tabellen als Quellentabellen und Zieltabellen bezeichnet:

- **Quellentabellen** sind extrahierte Tabellen, die in einem Konvertierungs-, Einfüge- oder Ladeprozess zu verwendende Daten enthalten.
- **Zieltabellen** sind die Tabellen, in die Daten konvertiert, eingefügt oder geladen werden.

Anmerkung: Einander zugeordnete Tabellen können unterschiedliche Ersteller-IDs oder Namen haben.

Tabellenzuordnungen werden im Optim-Repository gespeichert oder sind in einen Datenverwaltungsservice integriert.

Optional können Sie eine Spaltenzuordnung für ein beliebiges Paar von Tabellen in einer Tabellenzuordnung verwenden. Spaltenzuordnungen geben Spalten in einem Paar von Tabellen an und weisen sie zu. Eine Spaltenzuordnung muss verwendet werden, wenn Spaltennamen oder -attribute unterschiedlich sind oder Datenumsetzungen erforderlich sind.

Namenskonventionen

Der vollständig qualifizierte Name einer Tabellenzuordnung hat folgende Form: *identifizier.name*.

identifizier

Die der Tabellenzuordnung zugewiesene ID (1 bis 8 Zeichen).

name

Der der Tabellenzuordnung zugewiesene Name (1 bis 12 Zeichen).

Die Verwendung eines logischen Satzes von Namenskonventionen ist sinnvoll, um die Verwendung jedes Elements anzugeben und Definitionen für einen einfachen Zugriff zu organisieren.

Tabellenzuordnung erstellen

Mit dem Assistenten **Neue Tabellenzuordnung** können Sie eine neue Tabellenzuordnung erstellen.

Bevor Sie eine Tabellenzuordnung erstellen können, muss ein Dateidatenspeicheraliasname für die Quellentabelle vorhanden sein.

Gehen Sie wie folgt vor, um eine Tabellenzuordnung zu erstellen:

1. Erweitern Sie einen Ordner im Repository-Explorer, der die Tabellenzuordnung enthalten soll, klicken Sie mit der rechten Maustaste auf **Tabellenzuordnungen** und klicken Sie auf **Neue Tabellenzuordnung**. Der Assistent **Neue Tabellenzuordnung** wird geöffnet.
2. Führen Sie die Schritte im Assistenten aus.
Sie müssen einen Aliasnamen für Dateidatenspeicher für die Quellentabelle auswählen.

Die neue Tabellenzuordnung wird im Tabellenzuordnungseditor angezeigt.

Sie müssen den Editor zum Definieren von Zieldatenspeichern und Schemata verwenden, bevor Sie die Tabellenzuordnung speichern.

Tabellenzuordnung bearbeiten

Verwenden Sie den Tabellenzuordnungseditor, um Zieldaten zu bearbeiten und eine Spaltenzuordnung hinzuzufügen.

Standardzieldaten in einer Tabellenzuordnung definieren

Mit dem Tabellenzuordnungseditor können Sie Standardzieldatenspeicher und -schemata in einer Tabellenzuordnung definieren.

Bevor Sie Zieldaten für Standardzieldaten für Tabellenzuordnungen definieren, muss ein Datenspeicheraliasname für die Zieldaten vorhanden sein.

Gehen Sie wie folgt vor, um Standardzieldaten in einer Tabellenzuordnung zu definieren:

1. Erweitern Sie einen Ordner im Repository-Explorer, der die Tabellenzuordnung enthält, erweitern Sie den Knoten **Tabellenzuordnungen** und doppelklicken Sie auf die Tabellenzuordnung. Der Tabellenzuordnungseditor wird geöffnet.
2. Wählen Sie die Registerkarte **Tabellenzuordnung** aus.
3. Wählen Sie im Bereich **Datenspeicheraliasname und Schemazuordnung** die Zeile aus, die den Quelldatenspeicher enthält.
4. Klicken Sie zum Definieren eines Aliasnamens für einen Zieldatenspeicher auf die Zelle **Aliasname für Zieldatenspeicher** und wählen Sie den Aliasnamen für einen Zieldatenspeicher aus der Liste aus.
5. Klicken Sie zum Definieren eines Zielschemas auf die Zelle **Zielschema** und wählen Sie das Zielschema aus der Liste aus.
6. Speichern Sie die Tabellenzuordnung.

Zieldaten in einer Tabellenzuordnung bearbeiten

Mit dem Tabellenzuordnungseditor können Sie Zieldatenspeicher, Schemata und Tabellen für Quellentabellen in einer Tabellenzuordnung bearbeiten.

Bevor Sie Zieldaten für Tabellenzuordnungen bearbeiten, muss ein Aliasname für den Standardzieldatenspeicher für die Zieldaten vorhanden sein.

Gehen Sie wie folgt vor, um Zieldaten in einer Tabellenzuordnung zu bearbeiten:

1. Erweitern Sie einen Ordner im Repository-Explorer, der die Tabellenzuordnung enthält, erweitern Sie den Knoten **Tabellenzuordnungen** und doppelklicken Sie auf die Tabellenzuordnung. Der Tabellenzuordnungseditor wird geöffnet.
2. Wählen Sie die Registerkarte **Tabellenzuordnung** aus.
3. Wählen Sie im Bereich **Tabellenzuordnung** die Zeile aus, die die Quellentabelle enthält.
4. Klicken Sie zum Definieren eines Aliasnamens für einen Zieldatenspeicher auf die Zelle **Zieldatenspeicher** und geben Sie den Aliasnamen für einen Zieldatenspeicher ein.
5. Klicken Sie zum Definieren eines Zielschemas auf die Zelle **Zielschema** und wählen Sie das Zielschema aus der Liste aus.
6. Klicken Sie zum Definieren einer Zieltabelle auf die Zelle **Zieltabelle** und wählen Sie die Zieltabelle aus der Liste aus.
7. Speichern Sie die Tabellenzuordnung.

Neue Spaltenzuordnung einer Tabellenzuordnung hinzufügen

Mit dem Tabellenzuordnungseditor können Sie einem Paar zugeordneter Tabellen eine Spaltenzuordnung hinzufügen.

Gehen Sie wie folgt vor, um eine Spaltenzuordnung einer Tabellenzuordnung hinzuzufügen:

1. Erweitern Sie einen Ordner im Repository-Explorer, der die Tabellenzuordnung enthält, erweitern Sie den Knoten **Tabellenzuordnungen** und doppelklicken Sie auf die Tabellenzuordnung. Der Tabellenzuordnungseditor wird geöffnet.
2. Wählen Sie die Registerkarte **Tabellenzuordnung** aus.
3. Wählen Sie im Bereich **Tabellenzuordnung** die Zeile aus, die die Tabellen für die Spaltenzuordnung enthält.

4. Klicken Sie auf **Spaltenzuordnung hinzufügen**. Das Fenster **Neue Spaltenzuordnung** wird mit einer Liste der Spaltenzuordnungen geöffnet, die die ausgewählten Tabellen enthalten.
5. Wählen Sie eine Spaltenzuordnung oder **Neue Spaltenzuordnung erstellen** aus, um eine Spaltenzuordnung auf der Basis der ausgewählten Tabellen zu erstellen. Klicken Sie auf **OK**.
 - a. Wenn Sie auswählen, dass eine neue Spaltenzuordnung erstellt werden soll, wird der Spaltenzuordnungseditor geöffnet und mit den Spalten aus den ausgewählten Tabellen aufgefüllt.
 - b. Bearbeiten Sie die neue Spaltenzuordnung und speichern Sie sie.
6. Speichern Sie die Tabellenzuordnung.

Mit Spaltenzuordnungen arbeiten

Eine Spaltenzuordnung stellt Spezifikationen bereit, die zum Abgleichen von Spalten oder zum Ausschließen von Spalten aus der Verarbeitung in einem Datenverwaltungsservice erforderlich sind. Konvertierungs-, Einfüge- und Ladeservices müssen eine Tabellenzuordnung referenzieren, die mindestens eine Spaltenzuordnung referenzieren kann. Sie können mit einer Spaltenzuordnung Datenumsetzungen mit Funktionen oder Spaltenzuordnungsprozeduren definieren.

Eine Spaltenzuordnung muss verwendet werden, wenn Spaltennamen oder -attribute unterschiedlich sind, wenn Datenumsetzungen erforderlich sind oder wenn mindestens eine Spalte von der Verarbeitung ausgeschlossen werden soll. Eine Spaltenzuordnung, die in einem Konvertierungs-, Einfüge- und Ladeservice referenziert wird, kann Daten ändern, Datumswerte nach Verweildauer steuern oder Währungen konvertieren. Sie können Datenumsetzungen mithilfe von Funktionen oder Spaltenzuordnungsprozeduren definieren.

Wenn Sie eine neue Spaltenzuordnung erstellen, müssen Sie einen Aliasnamen für Dateidatenspeicher als Quelle für die Spalten, die Sie zuordnen wollen, auswählen. In ähnlicher Weise müssen Sie einen Daten-speicheraliasnamen für die Zieldaten angeben.

Spaltenzuordnungen, die im Optim-Repository gespeichert werden, sind für die erneute Verwendung oder für die gemeinsame Nutzung mit anderen Benutzern verfügbar. Eine lokale Spaltenzuordnung wird als Teil eines Datenverwaltungsservice gespeichert und ist nicht für andere Services verfügbar. Wenn die zugehörige Tabellenzuordnung für einen Service lokal ist, sind die Tabellenzuordnung und die Spaltenzuordnung nur für den jeweiligen Service verfügbar.

Namenskonventionen

Der vollständig qualifizierte Name einer Spaltenzuordnung hat zwei Teile: *identifizier.name*.

identifizier

Die der Spaltenzuordnung zugewiesene ID (1 bis 8 Zeichen).

name Der der Spaltenzuordnung zugewiesene Name (1 bis 12 Zeichen).

Beim Erstellen von Spaltenzuordnungen ist die Verwendung eines logischen Satzes von Namenskonventionen sinnvoll, um Definitionen für einen einfachen Zugriff zu benennen und zu organisieren.

Kompatibilitätsregeln für Spaltenzuordnungen

Die folgenden Datenklassen und zugehörigen Datentypen werden unterstützt. Diese Datenklassen sind für die Datenkompatibilität von Bedeutung, wenn Sie Spaltenwerte in Beziehungen und Spaltenzuordnungen angeben.

Zeichen

Datenbank- verwaltungs- system

Datentypen

DB2	CHAR, VARCHAR, CLOB
Oracle	CHAR, VARCHAR2, LONG, CLOB, NCLOB, NCHAR, NVARCHAR
Sybase ASE	CHAR, VARCHAR, TXT
SQL Server	CHAR, VARCHAR, TXT
Informix	CHAR, VARCHAR, TXT

Anmerkung: Spalten mit Einzelbytezeichen sind nicht kompatibel mit Spalten mit Mehrbyte- oder Unicode-Zeichen.

Numerisch

Datenbank- verwaltungs- system

Datentypen

DB2	INTEGER, SMALLINT, DECIMAL, FLOAT, DOUBLE
Oracle	NUMBER, FLOAT
Sybase ASE	TINYINT, INT, SMALLINT, DECIMAL, FLOAT, REAL, MONEY, SMALL MONEY
SQL Server	TINYINT, INT, SMALLINT, DECIMAL, FLOAT, REAL, MONEY, SMALL MONEY
Informix	INTEGER, SMALLINT, DECIMAL, FLOAT, REAL, DOUBLE PRECISION, SMALLFLOAT, SERIAL, MONEY, NUMERIC

Binär

Datenbank- verwaltungs- system

Datentypen

DB2	CHAR (für Bitdaten), VARCHAR (for Bit Data), BLOB
Oracle	RAW, LONG RAW
Sybase ASE	BINARY, VARBINARY, IMAGE
SQL Server	BINARY, VARBINARY, IMAGE
Informix	BYTE

Boolesch

Datenbank- verwaltungs- system

Datentypen

Sybase ASE	BOOLEAN (TRUE oder FALSE)
------------	---------------------------

Datum/Uhrzeit

Datenbank- verwaltungs- system

Datentypen

DB2	TIMESTAMP
Oracle	DATE, TIMESTAMP, TIMESTAMP WITH LOCAL TIME ZONE, TIMESTAMP WITH TIME ZONE
Sybase ASE	DATETIME, SMALL DATE TIME
SQL Server	DATETIME, SMALL DATE TIME
Informix	DATE, DATETIME

Datum

Datenbank- verwaltungs- system

Datentypen

DB2	DATE
Oracle	DATE
Informix	DATE

Zeit

Datenbank- verwaltungs- system

Datentypen

DB2	TIME
-----	------

Intervall

Datenbank- verwaltungs- system

Datentypen

Oracle	YEAR/MONTH INTERVAL, DAY/SECOND INTERVAL
Informix	YEAR/MONTH INTERVAL, DAY/TIME INTERVAL

Spaltenzuordnung erstellen

Mit dem Assistenten 'Neue Spaltenzuordnung' können Sie eine Spaltenzuordnung erstellen.

Bevor Sie eine Spaltenzuordnung erstellen können, muss ein Dateidatenspeicheraliasname für die Quellen- und Zieldaten vorhanden sein.

Gehen Sie wie folgt vor, um eine Spaltenzuordnung zu erstellen:

1. Erweitern Sie einen Ordner im Repository-Explorer, der die Spaltenzuordnung enthalten soll, klicken Sie mit der rechten Maustaste auf **Spaltenzuordnungen** und klicken Sie auf **Neue Spaltenzuordnung**. Der Assistent **Neue Spaltenzuordnung** wird geöffnet.

2. Führen Sie die Schritte im Assistenten aus.

Sie müssen einen Aliasnamen für Dateidatenspeicher und eine Tabelle für die Quellentabelle auswählen. Außerdem müssen Sie einen Datenspeicheraliasnamen und eine Tabelle für die Zieldaten auswählen.

Die neue Spaltenzuordnung wird im Spaltenzuordnungseditor angezeigt.

Spaltenzuordnung bearbeiten

Mit dem Spaltenzuordnungseditor können Sie eine Spaltenzuordnung bearbeiten und Datenmaskierungsfunktionen anwenden.

Datenmaskierungsfunktion anwenden

Mit dem Spaltenzuordnungseditor können Sie eine Funktion für eine Spalte anwenden und bearbeiten.

Gehen Sie wie folgt vor, um eine Datenmaskierungsfunktion anzuwenden:

1. Erweitern Sie einen Ordner im Repository-Explorer, der die Spaltenzuordnung enthält, erweitern Sie den Knoten **Spaltenzuordnungen** und doppelklicken Sie auf die Spaltenzuordnung. Der Spaltenzuordnungseditor wird geöffnet.
2. Wählen Sie die Spalte für die Richtlinie aus.
3. Klicken Sie auf **Funktion anwenden**. Das Fenster **Funktion anwenden** wird geöffnet.
4. Wählen Sie eine Funktion aus, die Sie anwenden wollen. Klicken Sie auf **OK**. Der Funktionsname wird in der zugehörigen Spalte angezeigt und der Funktionseditor wird im Spaltenzuordnungseditor geöffnet.
5. Führen Sie je nach Funktion einen der folgenden Schritte aus:
Wählen Sie die Registerkarte für Funktionsausdrücke aus und bearbeiten Sie den Funktionsausdruck.
Wählen Sie eine Optionsregisterkarte aus und wählen Sie Optionen für die Funktion aus.
6. Speichern Sie die Spaltenzuordnung.

Quellenspalte zuordnen

Mit dem Spaltenzuordnungseditor können Sie eine Quellenspalte einem Ziel zuordnen. Quellen- und Zielspalten mit demselben Namen und kompatiblen Datentypen werden einander automatisch zugeordnet.

Gehen Sie wie folgt vor, um eine Quellenspalte zuzuordnen:

1. Erweitern Sie einen Ordner im Repository-Explorer, der die Spaltenzuordnung enthält, erweitern Sie den Knoten **Spaltenzuordnungen** und doppelklicken Sie auf die Spaltenzuordnung. Der Spaltenzuordnungseditor wird geöffnet.
2. Klicken Sie auf eine Quellenspalte und wählen Sie einen Spaltennamen aus der Liste aus.
3. Speichern Sie die Spaltenzuordnung.

Mit Spaltenzuordnungsprozeduren arbeiten

Eine Spaltenzuordnungsprozedur ist eine Prozedur, mit der Daten in einer Spalte beim Ausführen eines Service maskiert oder umgesetzt werden. Wie der Name sagt, müssen Sie Spaltenzuordnungsprozeduren einer Spaltenzuordnung hinzufügen. Sie können Spaltenzuordnungsprozeduren mithilfe der Lua-Scripting-Sprache schreiben.

Spaltenzuordnungsprozedur erstellen:

Mit dem Lua-Scripteditor können Sie eine Spaltenzuordnungsprozedur erstellen.

Gehen Sie wie folgt vor, um eine Spaltenzuordnungsprozedur zu erstellen:

1. Erweitern Sie einen Ordner im Repository-Explorer, der die Spaltenzuordnung enthält, erweitern Sie den Knoten **Spaltenzuordnungen** und doppelklicken Sie auf die Spaltenzuordnung. Der Spaltenzuordnungseditor wird geöffnet.
2. Wählen Sie die Spalte für die Spaltenzuordnungsprozedur aus.
3. Klicken Sie auf **Prozedur hinzufügen**. Der Lua-Scripteditor wird geöffnet.
4. Erstellen und speichern Sie die Prozedur. In der der Prozedur zugeordneten Spalte wird **Prozedur** angezeigt.
5. Speichern Sie die Spaltenzuordnung.

Spaltenzuordnungsprozedur bearbeiten:

Mit dem Lua-Scripteditor können Sie eine Spaltenzuordnungsprozedur bearbeiten.

Gehen Sie wie folgt vor, um eine Spaltenzuordnungsprozedur zu bearbeiten:

1. Erweitern Sie einen Ordner im Repository-Explorer, der die Spaltenzuordnung enthält, erweitern Sie den Knoten **Spaltenzuordnungen** und doppelklicken Sie auf die Spaltenzuordnung. Der Spaltenzuordnungseditor wird geöffnet.
2. Wählen Sie die Spalte aus, die der Spaltenzuordnungsprozedur zugeordnet ist.
3. Klicken Sie auf **Prozedur bearbeiten**. Der Lua-Scripteditor wird geöffnet.
4. Bearbeiten und speichern Sie die Prozedur.
5. Speichern Sie die Spaltenzuordnung.

Eingeben von Werten für Parameter eine Spaltenzuordnungsprozedur:

Mit dem Spaltenzuordnungseditor können Sie Werte für Parameter in einer Spaltenzuordnungsprozedur eingeben.

Gehen Sie wie folgt vor, um Werte für Parameter in eine Spaltenzuordnungsprozedur einzugeben:

1. Erweitern Sie einen Ordner im Repository-Explorer, der die Spaltenzuordnung enthält, erweitern Sie den Knoten **Spaltenzuordnungen** und doppelklicken Sie auf die Spaltenzuordnung. Der Spaltenzuordnungseditor wird geöffnet.
2. Wählen Sie die Spalte aus, die die Prozedur enthält. Der Prozedureditor wird im Spaltenzuordnungseditor geöffnet.
3. Geben Sie in das Feld für die Prozedurparameter durch Kommas getrennte Werte für die Parameter ein. Beispiel: abc, def.
4. Speichern Sie die Spaltenzuordnung.

Kapitel 4. Datenverwaltungsservices entwerfen

Mit einem Datenverwaltungsservice können Sie Daten extrahieren, konvertieren, laden oder einfügen. Ein Service verwendet Zugriffsdefinitionen zum Definieren der zu extrahierenden Daten. Ferner verwendet ein Service die Tabellenzuordnungen und Spaltenzuordnungen, um Quellen- und Zieldaten zuzuordnen. Sie können Daten maskieren, indem Sie eine Datenschutzrichtlinie auf eine Entität anwenden, die von einem Service verarbeitet wird. Sie können ebenfalls Prozessanforderungen in einem Optim-Verzeichnis in Datenverwaltungsservices umsetzen. Mit Optim Manager im integrierten Modus können Sie Services testen.

Mit Extraktionsservices arbeiten

Sie können einen Extraktionsservice verwenden, um eine Gruppe zusammengehöriger Zeilen aus mindestens einer Tabelle zu kopieren und die Zeilen in einem Dateidateispeicher zu speichern.

Der Extraktionsservice gibt die Gruppe von Parametern, die zum Extrahieren von Daten und Objektdefinitionen aus Quellentabellen erforderlich sind, und den Dateidateispeicher, in dem die extrahierten Informationen gespeichert werden sollen, an.

Der Extraktionsservice extrahiert stets Definitionen für Tabellen und Spalten. Diese Definitionen werden erforderlichenfalls verwendet, um die Zieltabellen zu erstellen. Sie können auch auswählen, dass andere Objektdefinitionen wie Primärschlüssel, Beziehungen und Indizes extrahiert werden sollen.

Namenskonventionen

Der vollständig qualifizierte Name eines Extraktionsservice besteht aus folgenden Komponenten: *identifizier.name*.

identifizier

ID, die als Präfix für den Namen des Extraktionsservice dient (1 bis 8 Zeichen).

name Der dem Extraktionsservice zugewiesene Name (1 bis 12 Zeichen).

Beim Erstellen von Extraktionsservices ist die Verwendung eines logischen Satzes von Namenskonventionen sinnvoll, um die Verwendung jedes Service anzugeben und Definitionen für einen einfachen Zugriff zu organisieren.

Extraktionsservice erstellen

Mit dem Assistenten **Neuer Service** können Sie einen Extraktionsservice erstellen.

Für einen Extraktionsservice ist eine Zugriffsdefinition erforderlich. Sie können eine Zugriffsdefinition während der Erstellung eines Service auswählen oder erstellen.

Gehen Sie wie folgt vor, um einen Extraktionsservice zu erstellen:

1. Erweitern Sie einen Ordner im Repository-Explorer, der den Extraktionsservice enthalten soll, klicken Sie mit der rechten Maustaste auf **Services** und klicken Sie auf **Neuer Service**. Der Assistent **Neuer Service** wird geöffnet.
2. Führen Sie die Schritte im Assistenten aus.
Sie müssen eine Zugriffsdefinition auswählen oder erstellen. Sie müssen auch einen Namen für den Zieldateidatenspeicher eingeben.

Der neue Extraktionsservice wird im Extraktionsserviceeditor angezeigt.

Extraktionservice bearbeiten

Mit dem Extraktionserviceeditor können Sie Verarbeitungsoptionen für Extraktionservices angeben einschließlich der zu extrahierenden Datenobjekte, der Konvertierung, der Dateikomprimierung und der Variablenüberschreibungen.

Extraktionsverarbeitungsoptionen angeben

Mit der Registerkarte **Serviceeigenschaften** im Extraktionserviceeditor können Sie Verarbeitungsoptionen für Extraktionservices angeben.

Sie können folgende Verarbeitungsoptionen für Extraktionservices angeben:

- Anzahl von Datenbankverbindungen verwalten. Die Steigerung der Anzahl von Datenbankverbindungen verbessert die Leistung bei der Verarbeitung großer Datenmengen, weil mehrere Threads gleichzeitig Zeilen extrahieren können.
- Anzahl der extrahierten Zeilen beschränken.
- Dateianhänge ermöglichen.
- Statistische Informationen im Extraktionservicebericht einschließen.

Gehen Sie wie folgt vor, um Verarbeitungsoptionen für Extraktionservices anzugeben:

1. Erweitern Sie einen Ordner im Repository-Explorer, der den Extraktionservice enthält, erweitern Sie den Knoten **Services** und doppelklicken Sie auf den zu bearbeitenden Extraktionservice. Der Extraktionserviceeditor wird geöffnet.
2. Wählen Sie die Registerkarte **Serviceeigenschaften** aus.
3. Bearbeiten Sie die Serviceverarbeitungsoptionen.
4. Speichern Sie den Extraktionservice.

Objekte und Optionen für die Gruppierung angeben

Mit der Registerkarte **Daten und Objekte** im Extraktionserviceeditor können Sie zu extrahierende Datenobjekte angeben.

Sie können die folgenden Optionen angeben:

- Geben Sie an, ob der Service nur Daten, nur Objekte oder beides extrahieren soll.
- Wählen Sie die zu extrahierenden Objekte aus.

Gehen Sie wie folgt vor, um zu extrahierende Datenobjekte anzugeben:

1. Erweitern Sie einen Ordner im Repository-Explorer, der den Extraktionservice enthält, erweitern Sie den Knoten **Services** und doppelklicken Sie auf den zu bearbeitenden Extraktionservice. Der Extraktionserviceeditor wird geöffnet.
2. Wählen Sie die Registerkarte **Daten und Objekte** aus.
3. Bearbeiten Sie die Serviceverarbeitungsoptionen.
4. Speichern Sie den Extraktionservice.

Konvertierungsoptionen angeben

Mit der Registerkarte **Konvertierung** im Extraktionserviceeditor können Sie Konvertierungsoptionen angeben. Sie können Daten konvertieren, die im Service extrahiert wurden.

Sie können die folgenden Optionen angeben:

- Geben Sie an, ob der Service extrahierte Daten konvertieren soll.
- Geben Sie die maximal zulässige Anzahl gelöschter Zeilen an, die nicht konvertiert werden können, bevor der Service gestoppt wird.

Gehen Sie wie folgt vor, um Konvertierungsoptionen anzugeben:

1. Erweitern Sie einen Ordner im Repository-Explorer, der den Extraktionsservice enthält, erweitern Sie den Knoten **Services** und doppelklicken Sie auf den zu bearbeitenden Extraktionsservice. Der Extraktionsserviceeditor wird geöffnet.
2. Wählen Sie die Registerkarte **Konvertierung** aus.
3. Bearbeiten Sie die Serviceverarbeitungsoptionen.
4. Speichern Sie den Extraktionsservice.

Dateikomprimierungsoptionen angeben

Mit der Registerkarte **Dateikomprimierungsoptionen** im Extraktionsserviceeditor können Sie Optionen zum Komprimieren des Dateidatenspeichers oder extrahierter Tabellen angeben.

Sie können die folgenden Optionen angeben:

- Geben Sie an, ob die Datei oder bestimmte Tabellen komprimiert werden.
- Für jede Tabelle können Sie auswählen, ob ein Komprimierungsschwellenwert verwendet oder eine maximale Komprimierung ausgeführt werden soll. Der Schwellenwert ist die minimale Größenreduktion, die Sie beim Komprimieren der Tabelle erwarten. Geben Sie einen Wert im Bereich von 1 bis 99 ein, um einen Schwellenwert für diese Tabelle festzulegen.

Die folgenden Komprimierungsmethoden sind für einen Dateidatenspeicher verfügbar:

Inlinekomprimierung

Die Daten werden beim Extrahieren und vor dem Schreiben in den Dateidatenspeicher komprimiert. Die Inlinekomprimierung hat weniger Ein-/Ausgabeoperationen als die nachträgliche Komprimierung, verwendet aber während der Ausführung des Extraktionsservice Datenbankressourcen.

Die Inlinekomprimierung benötigt während des Extraktionsprozesses weniger Speicherressourcen als die nachträgliche Komprimierung.

Nachträgliche Komprimierung

Die Daten werden extrahiert und in einen unkomprimierten Dateidatenspeicher geschrieben. In einem zweiten Schritt liest Optim den unkomprimierten Dateidatenspeicher und schreibt eine komprimierte Version des Dateidatenspeichers. Der Vorteil der nachträglichen Komprimierung ist, dass Datenbankverbindungen schneller als bei der Inlinekomprimierung geschlossen werden. Allerdings ist bei der nachträglichen Komprimierung die abgelaufene Gesamtzeit länger, weil der unkomprimierte Dateidatenspeicher geschlossen und gelesen werden muss und dann eine komprimierte Version erstellt werden muss.

An Standorten, bei denen Bedenken wegen Datenbankressourcenkonflikten herrscht, ist die nachträgliche Komprimierung vermutlich nützlich, da sie die Verwendung der Datenbankressourcen verkürzt. Allerdings steigert die nachträgliche Komprimierung die Anforderungen hinsichtlich abgelaufener Zeit und Speicher für die Verarbeitung des Extraktionsservice. Während der Komprimierung ist zwar mehr temporärer Speicher erforderlich, dieser wird aber nach Beendigung der Komprimierung wieder freigegeben.

Gehen Sie wie folgt vor, um Dateikomprimierungsoptionen anzugeben:

1. Erweitern Sie einen Ordner im Repository-Explorer, der den Extraktionsservice enthält, erweitern Sie den Knoten **Services** und doppelklicken Sie auf den zu bearbeitenden Extraktionsservice. Der Extraktionsserviceeditor wird geöffnet.
2. Wählen Sie die Registerkarte **Dateikomprimierungsoptionen** aus.
3. Bearbeiten Sie die Serviceverarbeitungsoptionen.
4. Speichern Sie den Extraktionsservice.

Variablenoptionen angeben

Mit der Registerkarte **Variablen** im Extraktionsserviceeditor können Sie Standardwerte für Variablen überschreiben.

Gehen Sie wie folgt vor, um Variablenoptionen anzugeben:

1. Erweitern Sie einen Ordner im Repository-Explorer, der den Extraktionsservice enthält, erweitern Sie den Knoten **Services** und doppelklicken Sie auf den zu bearbeitenden Extraktionsservice. Der Extraktionsserviceeditor wird geöffnet.
2. Wählen Sie die Registerkarte **Variablen** aus.
3. Überschreiben Sie die Standardwerte durch geeignete Werte.
4. Speichern Sie den Extraktionsservice.

Zugriffsdefinition für einen Extraktionsservice ändern

Mit dem Assistenten **Zugriffsdefinition ändern** können Sie eine Zugriffsdefinition ändern, die einem Extraktionsservice zugeordnet ist.

Gehen Sie wie folgt vor, um eine Zugriffsdefinition zu ändern, die einem Extraktionsservice zugeordnet ist:

1. Erweitern Sie einen Ordner im Repository-Explorer, der den Extraktionsservice enthält, erweitern Sie den Knoten **Services** und doppelklicken Sie auf den zu bearbeitenden Extraktionsservice. Der Extraktionsserviceeditor wird geöffnet.
2. Klicken Sie auf **Ändern**. Der Assistent **Zugriffsdefinition ändern** wird geöffnet.
3. Führen Sie die Schritte im Assistenten aus.
Wählen Sie eine Zugriffsdefinition aus oder wählen Sie aus, dass eine lokale Zugriffsdefinition erstellt werden soll.
4. Speichern Sie den Extraktionsservice.

Mit Konvertierungsservices arbeiten

Mit einem Konvertierungsservice können Sie Daten in einem Dateidatenspeicher umsetzen. Sie können Daten umsetzen, um Datenschutz zu gewährleisten oder um Daten systematisch umzusetzen, damit sie Ihren Anforderungen für Anwendungstests entsprechen.

Geben Sie eine Tabellenzuordnung für den Abgleich von Tabellen im Quelldateidatenspeicher mit Tabellen im Zieldateidatenspeicher oder zum Ausschließen von Tabellen aus dem Konvertierungsservice an.

Mithilfe von Spaltenzuordnungen in der Tabellenzuordnung können Sie angeben, welche Daten wie konvertiert werden sollen.

Namenskonventionen

Anmerkung: Der vollständig qualifizierte Name eines Konvertierungsservice besteht aus folgenden Komponenten: *identifizier.name*.

identifizier

ID, die als Präfix für den Namen des Konvertierungsservice dient (1 bis 8 Zeichen).

name Der dem Konvertierungsservice zugewiesene Name (1 bis 12 Zeichen).

Beim Erstellen von Konvertierungsservices ist die Verwendung eines logischen Satzes von Namenskonventionen sinnvoll, um die Verwendung aller Services anzugeben und sie für einen einfachen Zugriff zu organisieren.

Konvertierungsservice erstellen

Mit dem Assistenten **Neuer Service** können Sie einen Konvertierungsservice erstellen.

Für einen Konvertierungsservice ist eine Tabellenzuordnung erforderlich. Sie können eine Tabellenzuordnung während der Erstellung eines Service auswählen oder erstellen.

Gehen Sie wie folgt vor, um einen Konvertierungsservice zu erstellen:

1. Erweitern Sie einen Ordner im Repository-Explorer, der den Service enthalten soll, klicken Sie mit der rechten Maustaste auf **Services** und klicken Sie auf **Neuer Service**. Der Assistent **Neuer Service** wird geöffnet.
2. Führen Sie die Schritte im Assistenten aus.
Sie müssen eine Tabellenzuordnung auswählen oder erstellen. Sie können einen Namen für den Zieldateidatenspeicher eingeben oder auswählen, dass der Quelldateidatenspeicher überschrieben werden soll.

Der neue Konvertierungsservice wird im Editor für den Konvertierungsservice angezeigt.

Konvertierungsservice bearbeiten

Mit dem Editor für Konvertierungsservices können Sie Verarbeitungsoptionen für Konvertierungsservices angeben.

Konvertierungsverarbeitungsoptionen angeben

Mit der Registerkarte **Verarbeitungsoptionen** können Sie im Konvertierungsserviceeditor Verarbeitungsoptionen für Konvertierungsservices angeben.

Sie können folgende Verarbeitungsoptionen für Konvertierungsservices angeben:

- Begrenzung für die zu verwerfenden Zeilen festlegen.
- Zieldatei komprimieren.
- Dateianhänge ermöglichen.

Gehen Sie wie folgt vor, um Verarbeitungsoptionen für Konvertierungsservices anzugeben:

1. Erweitern Sie einen Ordner im Repository-Explorer, der den Konvertierungsservice enthält, erweitern Sie den Knoten **Services** und doppelklicken Sie auf den zu bearbeitenden Konvertierungsservice. Der Editor für Konvertierungsservices wird geöffnet.
2. Wählen Sie die Registerkarte **Verarbeitungsoptionen** aus.
3. Bearbeiten Sie die Serviceverarbeitungsoptionen.
4. Speichern Sie den Konvertierungsservice.

Tabellenzuordnung für einen Konvertierungsservice ändern

Mit dem Assistenten **Tabellenzuordnung ändern** können Sie eine Tabellenzuordnung ändern, die einem Konvertierungsservice zugeordnet ist.

Gehen Sie wie folgt vor, um eine Tabellenzuordnung zu ändern, die einem Konvertierungsservice zugeordnet ist:

1. Erweitern Sie einen Ordner im Repository-Explorer, der den Konvertierungsservice enthält, erweitern Sie den Knoten **Services** und doppelklicken Sie auf den zu bearbeitenden Konvertierungsservice. Der Editor für Konvertierungsservices wird geöffnet.
2. Klicken Sie auf **Ändern**. Der Assistent **Tabellenzuordnung ändern** wird geöffnet.
3. Führen Sie die Schritte im Assistenten aus.
Wählen Sie eine Tabellenzuordnung aus oder wählen Sie aus, dass eine lokale Tabellenzuordnung erstellt werden soll.
4. Speichern Sie den Konvertierungsservice.

Mit Einfügeservices arbeiten

Mit einem Einfügeservice können Sie in einem Dateidatenspeicher gespeicherte Daten in eine Zieldatenbank einfügen.

Mithilfe von Tabellenzuordnungen und Spaltenzuordnungen können Daten aus der Quelle dem Ziel zugeordnet werden. Sie müssen eine Tabellenzuordnung verwenden, um das Ziel für die Tabellen im Dateidatenspeicher anzugeben. Sie können auch Spaltenzuordnungen für mindestens eine Zieltabelle verwenden. Spaltenzuordnungen ermöglichen das Angeben der Quelldaten für jede Spalte und optional das Umsetzen der Daten, bevor sie eingefügt werden.

Wenn Tabellen im Dateidatenspeicher am Ziel nicht vorhanden sind, verwenden Sie das Dienstprogramm 'Create', um sie zu erstellen.

Namenskonventionen

Der vollständig qualifizierte Name eines Einfügeservice besteht aus folgenden Komponenten: *identifizier.name*.

identifizier

ID, die als Präfix für den Namen des Einfügeservice dient (1 bis 8 Zeichen).

name Der dem Einfügeservice zugewiesene Name (1 bis 12 Zeichen).

Beim Erstellen von Einfügeservices ist die Verwendung eines logischen Satzes von Namenskonventionen sinnvoll, um die Verwendung der einzelnen Services anzugeben und sie für einen einfachen Zugriff zu organisieren.

Einfügeservice erstellen

Mit dem Assistenten **Neuer Service** können Sie einen Einfügeservice erstellen.

Für einen Einfügeservice ist eine Tabellenzuordnung erforderlich. Sie können eine Tabellenzuordnung während der Erstellung eines Service auswählen oder erstellen.

Gehen Sie wie folgt vor, um einen Einfügeservice zu erstellen:

1. Erweitern Sie einen Ordner im Repository-Explorer, der den Service enthalten soll, klicken Sie mit der rechten Maustaste auf **Services** und klicken Sie auf **Neuer Service**. Der Assistent **Neuer Service** wird geöffnet.
2. Führen Sie die Schritte im Assistenten aus.
Sie müssen eine Tabellenzuordnung auswählen oder erstellen.

Der neue Einfügeservice wird im Editor für den Einfügeservice angezeigt.

Einfügeservice bearbeiten

Mit dem Editor für Einfügeservices können Sie Verarbeitungsoptionen für Einfügeservices angeben. Sie können Optionen zum Einfügen und Löschen von Zeilen und zur Handhabung von Triggern und Integritätsbedingung angeben.

ür die Einfügeverarbeitung angeben

Mit der Registerkarte **Verarbeitungsoptionen** können Sie im Editor für Einfügeservices Verarbeitungsoptionen für Einfügeservices angeben.

Sie können die folgenden Verarbeitungsoptionen angeben:

- Typ der durchzuführenden Verarbeitung auswählen. Geben Sie Parameter zum Sperren von Tabellen, zum Festlegen einer Festschreibungsfrequenz und zum Festlegen einer Begrenzung für zu verwerfende Zeilen an.
- Zeilen aus allen oder aus angegebenen Tabellen löschen. Das Löschen ist für das Wiederherstellen von Daten während Tests nützlich. Wenn eine Zeile aus irgendeinem Grund nicht gelöscht werden kann, werden alle gelöschten Zeilen bis zur letzten Festschreibung wiederhergestellt und die Einfügeverarbeitung wird gestoppt.
- Datenbanktrigger inaktivieren.
- Datenbankintegritätsbedingung inaktivieren.

Die Zeilenverarbeitung schließt folgende Optionen ein:

Einfügen

Fügt neue Zeilen in die Tabellen ein.

- Wenn der Primärschlüssel einer Zeile in den Quelldaten *nicht* mit dem Primärschlüssel einer Zeile in der Zieltabelle übereinstimmt, wird die Zeile eingefügt.
- Wenn der Primärschlüssel einer Zeile in den Quelldaten mit dem Primärschlüssel einer Zeile in der Zieltabelle *übereinstimmt*, wird die Zeile übergangen und als gelöscht gekennzeichnet.

Gemischt

Aktualisierungen, Einfügungen oder Aktualisierungen/Einfügungen gemäß Ihrer Auswahl für jede Tabelle im Fenster **Nach Tabelle angeben**. Wenn Sie das Fenster **Nach Tabelle angeben** verwenden wollen, klicken Sie auf **Nach Tabelle angeben** und wählen Sie eine Prozessoption für jede Tabelle aus.

- Wenn Sie **Gemischt** auswählen und keine Auswahlen im Fenster **Nach Tabelle angeben** angeben oder wenn Sie alle Tabellen auf dieselbe Auswahl setzen, ändert die Prozessoption sich so, dass der für alle Tabellen verwendete Prozess angezeigt wird.

Anmerkung: Sie können **Gemischt** nicht auswählen, wenn **Alle Zeilen werden gelöscht** im Bereich **ür Zeilenlöschung** ausgewählt wird.

Aktualisieren

Aktualisiert Zeilen in den Tabellen. Tabellen müssen einen Primärschlüssel haben.

- Stimmt der Primärschlüssel einer Zeile in den Quelldaten mit dem Primärschlüssel einer Zeile in der Zieltabelle überein, wird die Zeile aktualisiert.
- Wenn der Primärschlüssel einer Zeile in den Quelldaten *nicht* mit dem Primärschlüssel einer Zeile in der Zieltabelle übereinstimmt, wird die Zeile als 'fehlgeschlagen' zurückgegeben.

Anmerkung: Sie können **Aktualisieren** nicht auswählen, wenn **Alle Zeilen werden gelöscht** im Bereich **ür Zeilenlöschung** ausgewählt wird.

Aktualisieren/Einfügen

Aktualisierungen und Einfügungen von Zeilen in Tabellen. Tabellen müssen einen Primärschlüssel haben.

- Wenn der Primärschlüssel einer Zeile in den Quelldaten *nicht* mit dem Primärschlüssel einer Zeile in der Zieltabelle übereinstimmt, wird die Zeile eingefügt.
- Stimmt der Primärschlüssel einer Zeile in den Quelldaten mit dem Primärschlüssel einer Zeile in der Zieltabelle überein, wird die Zeile aktualisiert.

Anmerkung: Sie können **Aktualisieren/Einfügen** nicht auswählen, wenn **Alle Zeilen werden gelöscht** im Bereich **ür Zeilenlöschung** ausgewählt wird.

Gehen Sie wie folgt vor, um Einfügeverarbeitungsoptionen anzugeben:

1. Erweitern Sie einen Ordner im Repository-Explorer, der den Einfügeservice enthält, erweitern Sie den Knoten **Services** und doppelklicken Sie auf den zu bearbeitenden Einfügeservice. Der Editor für Einfügeservices wird geöffnet.
2. Wählen Sie die Registerkarte **Verarbeitungsoptionen** aus.
3. Bearbeiten Sie die Serviceverarbeitungsoptionen.
4. Speichern Sie den Einfügeservice.

Tabellenzuordnung für einen Einfügeservice ändern

Mit dem Assistenten **Tabellenzuordnung ändern** können Sie eine Tabellenzuordnung ändern, die einem Einfügeservice zugeordnet ist.

Gehen Sie wie folgt vor, um eine Tabellenzuordnung zu ändern, die einem Einfügeservice zugeordnet ist:

1. Erweitern Sie einen Ordner im Repository-Explorer, der den Einfügeservice enthält, erweitern Sie den Knoten **Services** und doppelklicken Sie auf den zu bearbeitenden Einfügeservice. Der Editor für Einfügeservices wird geöffnet.
2. Klicken Sie auf **Ändern**. Der Assistent **Tabellenzuordnung ändern** wird geöffnet.
3. Führen Sie die Schritte im Assistenten aus.
Wählen Sie eine Tabellenzuordnung aus oder wählen Sie aus, dass eine lokale Tabellenzuordnung erstellt werden soll.
4. Speichern Sie den Einfügeservice.

Mit Ladeservices arbeiten

Mit einem Ladeservice können Sie den Inhalt eines Dateidatenspeichers in ein Format umsetzen, das für ein bestimmtes Ladeprogramm für das Datenbankverwaltungssystem geeignet ist, und dann, falls dies angegeben wurde, das entsprechende Datenbankladedienstprogramm starten.

Der Ladeservice generiert eine Datendatei im richtigen Format für jede Tabelle im Dateidatenspeicher und, je nach Datenbankverwaltungssystem, eine SQL-Datei oder eine BAT-Datei (Stapelausführungsdatei) mit der zum Starten des Datenbankladedienstprogramms erforderlichen Syntax.

Ein Ladeservice enthält die zum Vorbereiten von Daten für ein Ladeprogramm für das Datenbankverwaltungssystem verwendeten Parameter und die Anweisungen, die zum Verarbeiten des Ladevorgangs erforderlich sind. Geben Sie eine Tabellenzuordnung im Ladeservice an, um das Ziel für die zu ladenden Daten zuzuordnen. Mithilfe optionaler Spaltenzuordnungen im Ladeservice können Sie Daten vor dem Laden umsetzen.

Zu ladende Daten müssen in sich einem Dateidatenspeicher befinden.

Laden im Vergleich mit Einfügen

Optim kann über einen Lade- oder Einfügeservice Daten in eine Datenbank verschieben. Bedenken Sie folgende Aspekte, wenn Sie entscheiden, welche Methode Sie verwenden wollen:

- Das Datenvolumen und die Geschwindigkeit der Verwendung des Datenbankladedienstprogramms können die Vorteile des Einfügeservice aufwiegen.
- Die Daten können Zyklen für die referenzielle Integrität (RI) enthalten, die die Fähigkeit des Einfügeservice, alle Daten erfolgreich einzufügen, überschreiten.
- Das Datenbankladedienstprogramm erfordert eine exklusive Steuerung der Datenbank und verhindert den Benutzerzugriff während des Ladeservice. Die Datenbank ist für andere Benutzer verfügbar, während der Einfügeservice ausgeführt wird.
- Das Datenbankladedienstprogramm fügt neue Daten ein oder ersetzt vorhandene Daten. Der Einfügeservice ermöglicht die Aktualisierungs-/Einfügeverarbeitung in einem einzigen Schritt.

Ausgabedateinamen

Ein Ladeservice generiert folgende Typen von Dateien, um den Ladeprozess des Datenbankdienstprogramms zu unterstützen:

Datendateien

Datendateien enthalten die Daten, die Sie laden wollen. Sie liegen in dem Format vor, das für das von Ihnen verwendete Datenbankverwaltungssystem geeignet ist. Optim generiert eine Datendatei für jede Tabelle im Dateidatenspeicher. Datendateien haben denselben Namen wie der Dateidatenspeicher, aber sie haben sequenziell nummerierte Dateinamenerweiterungen. Ein Dateidatenspeicher mit dem Namen *demo.xf*, der drei Tabellen enthält, generiert z. B. drei Datendateien mit den Namen *demo.001*, *demo.002* und *demo.003*.

Nachrichtendateien

Nachrichtendateien enthalten Informationen, die das Datenbankladedienstprogramm während des Ladeservice generiert. In der Regel ist eine Nachrichtendatei für den gesamten Ladeservice vorhanden. Die Nachrichtendatei hat denselben Namen wie der Dateidatenspeicher, aber die Erweiterung *.msg*. Wenn z. B. der Dateidatenspeicher den Namen *demo.xf* hat, hat die Nachrichtendatei den Namen *demo.msg*.

SQL-Datei

Für DB2 wird eine SQL-Datei mit einer Anweisung für jede Zieltabelle generiert, die die Ladeprogramm syntax zum manuellen Ausführen des Ladeprogramms enthält. Die SQL-Datei hat denselben Namen wie der Dateidatenspeicher, aber die Erweiterung *.sql*.

BAT-Datei

Für Oracle, Sybase ASE, SQL Server und Informix wird eine BAT-Datei generiert, die die Syntax zum manuellen Ausführen des Ladeprogramms für jede Tabelle enthält. Es wird eine BAT-Datei für jeden Datenspeicher aliasnamen generiert, der in der Tabellenzuordnung angegeben ist. Jede BAT-Datei befindet sich im Verzeichnis mit dem entsprechenden konvertierten Dateidatenspeicher. Wenn Sie das Ladeprogramm manuell ausführen wollen, muss die BAT-Datei durch Ersetzen einer Zeichenfolge aus acht Fragezeichen durch spezielle Kennwortinformationen (mit Ausnahme von Informix) z. B. im Editor bearbeitet werden.

Zusätzlich wird eine Formatdatei für jede Datendatei generiert. Eine Formatdatei hat denselben Namen wie die entsprechende Datendatei, aber die Dateinamenerweiterung ist unterschiedlich. Wenn weniger als 500 zu ladende Tabellen vorhanden sind, ist die Formatdateinamenerweiterung numerisch um 500 größer als die Datendateinamenerweiterung. (Wenn z. B. drei Datendateien mit den Namen *demo.001*, *demo.002* und *demo.003* vorhanden sind, heißen die entsprechenden Formatdateien *demo.501*, *demo.502* bzw. *demo.503*.) Wenn mehr als 500 Tabellen vorhanden sind, wird ein komplexerer Algorithmus für die Generierung von Dateierweiterungen verwendet.

Anmerkung: Wenn Ihr Dateiserver keine langen Dateinamen mit über acht Zeichen zulässt und der Dateidatenspeicher einen langen Namen hat, schlägt der Ladeservice fehl. Die beste Lösung ist, lange Dateinamen für Dateidatenspeicher zu vermeiden. Erforderlichenfalls können Sie eine Datei kopieren und umbenennen, bevor Sie sie für einen Ladeservice verwenden.

Namenskonventionen

Der vollständig qualifizierte Name eines Ladeservice besteht aus folgenden Komponenten: *identifizier.name*.

identifizier

ID, die als Präfix für den Servicenamen dient (1 bis 8 Zeichen).

name Der dem Service zugewiesene Name (1 bis 12 Zeichen).

Beim Erstellen von Ladeservices ist die Verwendung eines logischen Satzes von Namenskonventionen sinnvoll, um die Verwendung der einzelnen Services anzugeben und sie für einen einfachen Zugriff zu organisieren.

Ladeservice erstellen

Mit dem Assistenten **Neuer Service** können Sie einen Ladeservice erstellen.

Für einen Ladeservice ist eine Tabellenzuordnung erforderlich. Sie können eine Tabellenzuordnung während der Erstellung eines Service auswählen oder erstellen.

Gehen Sie wie folgt vor, um einen Ladeservice zu erstellen:

1. Erweitern Sie einen Ordner im Repository-Explorer, der den Extraktionsservice enthalten soll, klicken Sie mit der rechten Maustaste auf **Services** und klicken Sie auf **Neuer Service**. Der Assistent **Neuer Service** wird geöffnet.
2. Führen Sie die Schritte im Assistenten aus.
Sie müssen eine Tabellenzuordnung auswählen oder erstellen.

Der neue Ladeservice wird im Ladeserviceeditor angezeigt.

Ladeservice bearbeiten

Mit dem Ladeserviceeditor können Sie Ladeverarbeitungsoptionen angeben. Sie können für den Ladeprozess und für das DBMS-Ladeprogramm angeben, das jedem Zieldatenspeicher zugeordnet ist.

Ladeverarbeitungsoptionen angeben

Mit der Registerkarte **Verarbeitungsoptionen** können Sie im Ladeserviceeditor Verarbeitungs- für Ladeservices angeben.

Sie können die folgenden Optionen angeben:

- Wählen Sie aus, wie mehrere Ladeprogramme für das Datenbankverwaltungssystem ausgeführt werden sollen, wenn mehrere Datenspeicheraliasnamen verwendet werden: parallel (gleichzeitiges Ausführen mehrerer Ladeprogramme für das Datenbankverwaltungssystem) oder in Folge (Ausführen mehrerer Ladeprogramme für das Datenbankverwaltungssystem nacheinander).
- Ein Ladeprogramm für das Datenbankverwaltungssystem stoppen, wenn ein Fehler auftritt. Wenn mehrere Ladeprogramme für das Datenbankverwaltungssystem nacheinander ausgeführt werden, wird die Verarbeitung für das Ladeprogramm für das Datenbankverwaltungssystem und alle nachfolgenden Ladeprogramme für das Datenbankverwaltungssystem mit einem Fehler gestoppt.
- Datenbanktrigger inaktivieren.
- Datenbankintegritätsbedingung inaktivieren.

Gehen Sie wie folgt vor, um Verarbeitungs- für Ladeservices anzugeben:

1. Erweitern Sie einen Ordner im Repository-Explorer, der den Ladeservice enthält, erweitern Sie den Knoten **Services** und doppelklicken Sie auf den zu bearbeitenden Ladeservice. Der Ladeserviceeditor wird geöffnet.
2. Wählen Sie die Registerkarte **Verarbeitungsoptionen** aus.
3. Bearbeiten Sie die Serviceverarbeitungsoptionen.
4. Speichern Sie den Ladeservice.

für das Ladeprogramm für das Datenbankverwaltungssystem angeben

Mit der Registerkarte **Ladeoptionen** im Ladeserviceeditor können Sie für jedes Ladeprogramm für das Datenbankverwaltungssystem angeben.

Sie können die folgenden Optionen angeben:

- Einen Verarbeitungsmodus.
- Optionen, die für jeden Aliasnamen für einen Zieldatenspeicher spezifisch sind.

Je nach Datenbankverwaltungssystem des ausgewählten Zieldateidatenspeichers sind folgende Verarbeitungsmodi verfügbar:

Einfügen

Fügt Zeilen aus dem Quelldateidatenspeicher in leere Zieltabellen ein. Wenn Zieltabellen Daten enthalten, gibt das Ladeprogramm einen Fehler zurück.

Ersetzen

Löscht und ersetzt alle vorhandenen Zeilen in den Zieltabellen mit den Zeilen aus dem Quelldateidatenspeicher. (**Ersetzen** kann erheblich ressourcenintensiver als **Abschneiden** sein, weil keine Protokollierung ausgeführt wird.)

Anhängen

Fügt die Zeilen aus dem Quelldateidatenspeicher in die Zieltabellen ein. Wenn die Primärschlüsselwerte übereinstimmen, werden doppelte Zeilen gelöscht oder in die Ausnahmetabelle eingefügt (falls eine solche angegeben wurde).

Abschneiden

Abschneiden ist gleich wie **Ersetzen**, aber die Datenbank protokolliert die gelöschten Zeilen nicht und **Abschneiden** erfordert, dass referenzielle Integritätsbedingungen inaktiviert sind.

Gehen Sie wie folgt vor, um für das Ladeprogramm für das Datenbankverwaltungssystem anzugeben:

1. Erweitern Sie einen Ordner im Repository-Explorer, der den Ladeservice enthält, erweitern Sie den Knoten **Services** und doppelklicken Sie auf den zu bearbeitenden Ladeservice. Der Ladeserviceeditor wird geöffnet.
2. Wählen Sie die Registerkarte **Ladeoptionen** aus.
3. Wählen Sie den Zieldatenspeicher aliasnamen aus.
4. Bearbeiten Sie die Serviceverarbeitungsoptionen.
5. Speichern Sie den Ladeservice.

Tabellenzuordnung für einen Ladeservice ändern

Mit dem Assistenten **Tabellenzuordnung ändern** können Sie eine Tabellenzuordnung ändern, die einem Ladeservice zugeordnet ist.

Gehen Sie wie folgt vor, um eine Tabellenzuordnung zu ändern, die einem Ladeservice zugeordnet ist:

1. Erweitern Sie einen Ordner im Repository-Explorer, der den Ladeservice enthält, erweitern Sie den Knoten **Services** und doppelklicken Sie auf den zu bearbeitenden Ladeservice. Der Ladeserviceeditor wird geöffnet.
2. Klicken Sie auf **Ändern**. Der Assistent **Tabellenzuordnung ändern** wird geöffnet.
3. Führen Sie die Schritte im Assistenten aus.
Wählen Sie eine Tabellenzuordnung aus oder wählen Sie aus, dass eine lokale Tabellenzuordnung erstellt werden soll.
4. Speichern Sie den Ladeservice.

Datenverwaltungsservice testen

Mit Optim Manager im integrierten Modus kann ein Datenverwaltungsservice getestet werden.

Gehen Sie wie folgt vor, um einen Datenverwaltungsservice zu testen:

1. Erweitern Sie einen Ordner im Repository-Explorer, der den Extraktionsservice enthält, und erweitern Sie den Knoten **Services**.
2. Klicken Sie mit der rechten Maustaste auf den Service und klicken Sie auf **Service ausführen**. Optim Manager wird geöffnet und das Fenster **Service ausführen** wird angezeigt.
3. Klicken Sie auf **Ausführen**. Mithilfe der Registerkarte **Serviceüberwachung** können Sie den Fortschritt des Service überwachen.

Kapitel 5. Daten maskieren

Sie können Daten wie länderspezifische ID-Nummern, Kreditkartennummern, Datumsangaben, numerische Werte und personenbezogene Daten maskieren. Geben Sie eine Datenmaskierungsfunktion über eine Spaltenzuordnung ein oder erstellen Sie eine Spaltenzuordnungsprozedur über ein LUA-Script. Verwenden Sie einen Konvertierungsservice, um die Daten umzusetzen.

Datenmaskierungsfunktion anwenden

Mit dem Spaltenzuordnungsektor können Sie eine Funktion für eine Spalte anwenden und bearbeiten.

Gehen Sie wie folgt vor, um eine Datenmaskierungsfunktion anzuwenden:

1. Erweitern Sie einen Ordner im Repository-Explorer, der die Spaltenzuordnung enthält, erweitern Sie den Knoten **Spaltenzuordnungen** und doppelklicken Sie auf die Spaltenzuordnung. Der Spaltenzuordnungsektor wird geöffnet.
2. Wählen Sie die Spalte für die Richtlinie aus.
3. Klicken Sie auf **Funktion anwenden**. Das Fenster **Funktion anwenden** wird geöffnet.
4. Wählen Sie eine Funktion aus, die Sie anwenden wollen. Klicken Sie auf **OK**. Der Funktionsname wird in der zugehörigen Spalte angezeigt und der Funktionseditor wird im Spaltenzuordnungsektor geöffnet.
5. Führen Sie je nach Funktion einen der folgenden Schritte aus:
Wählen Sie die Registerkarte für Funktionsausdrücke aus und bearbeiten Sie den Funktionsausdruck.
Wählen Sie eine Optionsregisterkarte aus und wählen Sie für die Funktion aus.
6. Speichern Sie die Spaltenzuordnung.

Datenmaskierungsfunktionen

Datenmaskierungsfunktionen bieten verschiedene Methoden zum Umsetzen oder Maskieren vertraulicher Daten.

Suchfunktionen

Über die Suchfunktionen können Sie Werte in einer Suchtabelle auswählen, die verwendet werden, um die Zieltabelle zu füllen. Mit den Funktionen für die Suche und für die Hashsuche können Sie Werte auf der Basis des Quellenwerts auswählen. Alternativ hierzu können Sie mit der Funktion für die Zufallssuche Werte aus einer Suchtabelle ohne Berücksichtigung des Quellenwerts auswählen.

Suchfunktion

Die Suchfunktion ruft den Wert für eine Zielspalte aus einer Suchtabelle entsprechend dem Wert in einer Quellenspalte ab. Es gibt zwei Formate der Suchfunktion, das Einzelspaltenformat und das Mehrspaltenformat.

Beim Einzelspaltenformat wird ein Wert in eine einzelne Zielspalte eingefügt. Beim Mehrspaltenformat werden Werte aus den Suchtabellenspalten in die entsprechenden Zielspalten eingefügt.

Sie können die Funktion zur Suche mit mehreren Spalten für jede beliebige Quellenspalte eingeben, die durch einen Suchtabellenwert ersetzt wird, aber Sie müssen die Spaltenzuordnung so bearbeiten, dass die Namen verbleibender Quellenspalten, die ebenfalls ersetzt werden, entfernt werden.

Mit dem Parameter *ignore* können Sie die Suchtabelle ignorieren und einen Quellenwert verwenden, wenn eine Zeile in einer angegebenen Quellenspalte einen angegebenen Wert (NULL, SPACES (für CHAR-Spalten) oder VARCHAR mit Nulllänge) enthält.

Sie können den Parameter *preserve* verwenden, um die Suchtabelle zu ignorieren und einen Quellenwert zu verwenden, wenn eine Zeile in einer angegebenen Quellenspalte einen angegebenen Wert (NULL, SPACES (für CHAR-Spalten) oder VARCHAR mit Nulllänge) enthält. *preserve* kann auch verwendet werden, um die Suchtabelle zu ignorieren, wenn eine Quellenspalte keinen Wert enthält.

Wenn keine Übereinstimmung in der Suchtabelle gefunden wird, wird ein Konvertierungsfehler ausgegeben.

Die Syntax ist:

```
LOOKUP ( [sourcesearchcol, | SRCSEARCH=(sourcecol1,...,sourcecoln)]
        [dest=(col1, coln) , ]
        [lktablename ( {LookupTableSearchcol
        | LKPSEARCH=(LookupTableSearchCol1,...,LookupTableSearchColn},
        {value | values=(col1, coln) }
        [,cache | ,nocache ] )
        [,ignore=(colname ( spaces, null, zero_len ), )
        | PRESERVE=( [ NOT_FOUND, ] colname (spaces, null, zero_len),... ) ] )
```

sourcesearchcol

Für die Einzelspaltensuche der Name der Quellentabellenspalte, die den Suchwert enthält (optional). Wird dieser Parameter nicht angegeben, wird der Name der Zielspalte verwendet.

SRCSEARCH=

Für die Mehrspaltensuche die Namen der Quellentabellenspalten, die den Suchwerte enthalten. Spaltennamen müssen in runde Klammern eingeschlossen und durch Kommas getrennt werden.

dest= Namen von Zieltabellenspalten, in die Werte aus der Suchtabelle eingefügt werden. (Für die Mehrspaltensuche erforderlich.)

col1, *coln*

Namen der Zieltabellenspalten. Die Reihenfolge der Spaltennamen muss den Suchtabellenspalten im Parameter *values=* entsprechen.

lktablename

Name der Suchtabelle. Sie können den Namen der Suchtabelle als *dbalias.creatorid.tablename*, *creatorid.tablename*, oder *tablename* angeben. Wenn Sie den Tabellennamen nicht vollständig qualifizieren, werden die Qualifikationsmerkmale für die Zieltabelle verwendet.

LookupTableSearchcol

Für die Einzelspaltensuche der Name der Spalte in der Suchtabelle, die einen Wert enthält, der mit dem Suchwert aus der Quellenspalte abgeglichen werden soll.

value Name der Spalte in der Suchtabelle, die den umgesetzten Suchwert enthält, der am Ziel eingefügt werden soll. (Für die Einzelspaltensuche erforderlich.)

values=

Namen der Suchtabellenspalten, die Werte enthalten, die am Ziel eingefügt werden sollen. (Für die Mehrspaltensuche erforderlich.)

col1, *coln*

Namen der Suchtabellenspalten. Die Reihenfolge der Spaltennamen muss den Zieltabellenspalten im Parameter *dest=* entsprechen.

cache | *nocache*

Geben Sie *cache* (Standardwert) an, um eine Tabelle mit gefundenen Suchwerten im Speicher zu behalten, oder geben Sie *nocache* an, um gefundene Werte zu löschen. Die Verwendung von *cache* ist beim häufigen Abrufen eines Werts zwar schneller, benötigt aber mehr Speicher.

ignore=

Liste der Quellenspalten mit Werten, die am Ziel anstelle des Suchwerts eingefügt werden, wenn die Spalte eine Zeile mit einem angegebenen Wert (NULL, SPACES, ZERO oder VARCHAR mit Nulllänge) hat.

col Der Name der Quellenspalte.

Geben Sie für die Einzelspaltsuche nur einen einzigen Spaltennamen ein.

Bei der Mehrspaltsuche muss die Reihenfolge der Spaltennamen muss den Zieltabellenspalten im Parameter *dest=* entsprechen. Die Anzahl der Spalten muss den Spalten im Parameter *dest=* entsprechen und mindestens eine Spalte muss einen Wert enthalten. Wenn Sie keine Werte für eine Spalte angeben wollen, geben Sie keinen Wert ein. Beispiel: *coln()*.

null Die Suchtabelle ignorieren, wenn die Quellenspaltenzeile einen Nullwert hat.

SPACES

Die Suchtabelle ignorieren, wenn die Quellenspaltenzeile den Wert SPACES hat. Nur für CHAR-Spalten.

ZERO_LEN

Die Suchtabelle ignorieren, wenn die Quellenspaltenzeile einen VARCHAR-Wert mit Nulllänge hat.

preserve=

Liste der Quellenspalten mit Werten, die am Ziel anstelle des Suchwerts eingefügt werden, wenn die Quellenspalte einen angegebenen Wert (NOT_FOUND, NULL, SPACES oder VARCHAR mit Nulllänge) hat.

NOT_FOUND

Die Suchtabelle ignorieren, wenn keine Übereinstimmung für die Quellenspaltenzeile gefunden wird.

Anmerkung:

preserve= und *ignore=* schließen sich gegenseitig aus. *ignore=* wird in einem späteren Release nicht weiter unterstützt.

Die Operanden *col*, *null*, *spaces* und *zero_len* haben den gleichen Effekt, wenn sie mit *preserve=* oder *ignore=* verwendet werden.

Einzelspaltenbeispiel

Mit der Suchfunktion können Sie den Quellenwert in einer Suchtabelle in einen entsprechenden Wert in einer anderen Tabelle umsetzen.

Es soll z. B. angenommen werden, dass die Quellenspalte, FSTATE, Abkürzungen von US-Bundesstaaten enthält (z. B. NJ) und dass die Zielspalte die kompletten Namen der Bundesstaaten enthält (in diesem Beispiel **New Jersey**). Eine Suchtabelle mit dem Namen STATE_LOOKUP enthält eine Spalte (CODE) für Bundesstaatsabkürzungen oder -codes und eine Spalte (NAME) für die entsprechenden Namen.

Wenn Sie den Wert für die Zielspalte mit der Tabelle STATE_LOOKUP abrufen wollen, geben Sie Folgendes an:

```
LOOKUP(STATE, STATE_LOOKUP(CODE, NAME))
```

Die Suchfunktion sucht nach einem Wert in der Spalte CODE der Tabelle STATE_LOOKUP, der dem Wert (NJ) in der Spalte STATE der Quellentabelle entspricht. Wenn eine Übereinstimmung gefunden wird, fügt die Funktion den entsprechenden Wert aus der Spalte NAME (**New Jersey**) in die Zielspalte ein.

Mehrspaltenbeispiel

Mit der Suchfunktion können Sie Werte aus Spalten in einer Suchtabellenzeile in Spalten in einer Zieltabellenzeile auf der Basis eines Werts in einer Quellenspalte einfügen.

Sie können z. B. auf der Basis einer Quellenspalte (SOC_SEC), die Sozialversicherungsnummern enthält, Werte in Zielspalten (FIRST_NAME und LAST_NAME) mit Vor- und Nachnamen aus einer Suchtabelle ersetzen. Eine Tabelle mit dem Namen NAME_LOOKUP enthält eine Spalte (SSN) mit den Sozialversicherungsnummern aus der Quellentabelle und Spalten (FIRST_MASK und LAST_MASK) zum Maskieren entsprechender Namen im Ziel.

Geben Sie zum Ersetzen von Namen in der Zieltabelle auf der Basis einer Sozialversicherungsnummer Folgendes an:

```
LOOKUP(SOC_SEC,DEST=(FIRST_NAME, LAST_NAME),  
NAME_LOOKUP(SSN,VALUES=(FIRST_MASK, LAST_MASK)))
```

Die Suchfunktion sucht nach einem Wert in der Spalte SSN der Tabelle NAME_LOOKUP, der dem Wert in der Spalte SOC_SEC der Quellentabelle entspricht. Wenn eine Übereinstimmung festgestellt wird, fügt die Funktion die entsprechenden Werte aus den Spalten FIRST_MASK und LAST_MASK der Suchtabelle in die entsprechenden Zielspalten ein.

Ignore-Beispiel

Verwenden Sie die folgende Anweisung, um das Einzelspaltenbeispiel zu erweitern, wenn Sie die Quellenwerte *NULL* und *SPACES* anstelle der Suchtabellenwerte verwenden wollen:

```
LOOKUP(STATE,STATE_LOOKUP(CODE,NAME),  
IGNORE=(STATE(NULL,SPACES)))
```

NoCache-Beispiel

Verwenden Sie die folgende Anweisung, um das Einzelspaltenbeispiel zu erweitern, wenn Sie eine Tabelle mit gefundenen Suchwerten nicht im Speicher beibehalten wollen.

```
LOOKUP(STATE,STATE_LOOKUP(CODE,NAME),NOCACHE)
```

Funktion für Hashsuche

Die Funktion für die Hashsuche ruft den Wert für eine Zielspalte aus einer Suchtabelle entsprechend dem aus einer Quellenspalte abgeleiteten Hashwert ab. Mithilfe der Funktion für die Hashsuche können Sie Daten konsistent maskieren, wenn Sie dieselben Quellen- und Suchtabellen in einer Umgebung verwenden.

Die Quellenspalte, für die das Hashing ausgeführt wird, muss keine Spalte sein, die durch Suchtabellenwerte ersetzt wird.

Bei der Funktion für Hashsuche muss Groß-/Kleinschreibung beachtet werden. Die Quellenwerte John und JOHN werden z. B. zu unterschiedlichen Werten hashverschlüsselt. Sie können mit dem Parameter TRIM den Quellenwert vor der Hashverschlüsselung in Großbuchstaben konvertieren.

Es gibt zwei Formate der Funktion für die Hashsuche, das Einzelspaltenformat und das Mehrspaltenformat. Beim Einzelspaltenformat wird ein Wert in eine einzelne Zielspalte eingefügt. Beim Mehrspaltenformat werden Werte aus mehreren Suchtabellenspalten in die entsprechenden Zielspalten auf der Basis eines einzelnen Hashwerts aus einer Quellenspalte eingefügt.

Sie können die Funktion zur Hashsuche mit mehreren Spalten für jede beliebige Quellenspalte eingeben, die durch Suchtabellenwerte ersetzt werden, aber Sie müssen die Spaltenzuordnung so bearbeiten, dass die Namen verbleibender Quellenspalten entfernt werden, die ebenfalls ersetzt werden.

Die Suchtabelle muss eine Schlüsselspalte enthalten, die fortlaufende Zahlenwerte ohne Lücken enthält, und die übrigen Spalten müssen Ersatzwerte enthalten. Die Schlüsselspalte muss einen numerischen Datentyp aufweisen. Die Suchtabelle ist in der Regel indexiert. Die Funktion führt eine Hashverschlüsselung für eine Quellenspalte aus, um in der Schlüsselspalte der Suchtabelle fortlaufende Zahlen von 1 bis zum Maximalwert abzuleiten. Der hashverschlüsselte Wert aus der Quellentabelle wird mit den fortlaufenden Zahlen in der Suchtabelle abgeglichen und Werte aus der entsprechenden Suchtabellenzeile werden am Ziel eingefügt.

Enthält die Quellenspalte, anhand derer der Hashwert abgeleitet wird, bestimmte Werte (NULL, Leerzeichen (für CHAR-Spalten), VARCHAR mit Nulllänge), wird für den Wert kein Hashing ausgeführt und die folgenden reservierten Werte werden als Schlüssel für die Suchtabelle verwendet:

Quellenwert	Suchtabellenschlüssel
NULL	-1
Leerzeichen (CHAR oder VARCHAR)	-2
VARCHAR mit Nulllänge	-3

Die Suchtabelle muss eine Zeile für jede dieser Zahlen enthalten, was es Ihnen ermöglicht, einen Suchwert für jeden dieser Quellenwerte einzufügen. Wenn einer dieser Quellenwerte gefunden wird und keine entsprechende Zahl in der Suchtabelle vorhanden ist, wird ein Konvertierungsfehler ausgegeben.

Mit dem Parameter *ignore* können Sie die Suchtabelle ignorieren und einen Quellenwert verwenden, wenn eine Zeile in einer angegebenen Quellenspalte einen angegebenen Wert (NULL, SPACES (für CHAR-Spalten) oder VARCHAR mit Nulllänge) enthält.

Sie können den Parameter *preserve* verwenden, um die Suchtabelle zu ignorieren und einen Quellenwert zu verwenden, wenn eine Zeile in einer angegebenen Quellenspalte einen angegebenen Wert (NULL, SPACES (für CHAR-Spalten) oder VARCHAR mit Nulllänge) enthält. *preserve* kann auch verwendet werden, um die Suchtabelle zu ignorieren, wenn eine Quellenspalte keinen Wert enthält.

Mit dem Parameter *trim* können Sie Zeichen angeben, die aus dem Quellenwert gelöscht werden sollen, bevor er hashverschlüsselt wird. Wenn Sie z. B. beschließen, dass Kommas aus einem Quellenwert gelöscht werden sollen, werden die Werte 'Smith, John' und 'Smith John' zu demselben Wert hashverschlüsselt. Sie können mit diesem Parameter auch den Quellenwert vor der Hashverschlüsselung in Großbuchstaben konvertieren.

Wenn der Quellenwert in Großbuchstaben konvertiert wird, werden die Abschneidezeichen ebenfalls in Großbuchstaben konvertiert.

Sie können mit dem Parameter *seed* die vom Hashalgorithmus ausgeführte Berechnung ändern. Der Hashwert aus der Quellenspalte und der *seed*-Wert werden mit einer fortlaufenden Zahl aus der Suchtabelle abgeglichen, um den Ersatzwert für die Zielspalte zu erhalten.

Die Syntax ist:

```
HASH_LOOKUP( [sourcecol,] [trim=( [char1char2 ] [\u]),]
  dest=(coll, coln), ltablename (search,
  { value | values=(coll, coln) } ) [ ,cache | ,nocache ]
  [,ignore=(col (spaces, null, zero_len), )
  | PRESERVE=( [ NOT_FOUND, ] colname (spaces, null, zero_len), ) )][,seed=n])
```

sourcecol

Name der Quellentabellenspalte, aus der Hashwerte abgeleitet werden (optional). Wenn dieser Wert nicht angegeben wird, wird der Name der Zielspalte verwendet.

trim= Liste der Zeichen, die vor der Hashverschlüsselung aus dem Quellenwert gelöscht werden, sowie eine Option, um den Quellenwert vor der Hashverschlüsselung in Großbuchstaben zu konvertieren. Wenn der hieraus resultierende Quellenwert NULL ist oder alle Leerzeichen nach Zeichen gelöscht wurden, erfolgt keine Hashverschlüsselung des Quellenwerts und dem Wert wird der entsprechende reservierte Wert (-1 oder -2) zugeordnet.

char1char2...

Zeichen, die aus dem Quellenwert gelöscht werden sollen, bevor er hashverschlüsselt wird. Bei der Liste wird Groß-/Kleinschreibung beachtet. Sie können ein Leerzeichen oder ein Komma als Zeichen angeben. Nach dem anfänglichen Vorkommen eines Zeichens werden zusätzliche Vorkommen in der Liste ignoriert.

Wenn Sie einen Backslash ("\") oder eine rechte Klammer (") angeben wollen, müssen Sie dem Zeichen ein Backslash als Escapezeichen voranstellen. Wenn Sie z. B. eine rechte Klammer angeben wollen, geben Sie Folgendes ein: *trim=(\)*.

Sie können das Escapezeichen nur mit einem Backslash, mit einer echten Klammer oder als Teil des Anzeigers für die Großschreibung verwenden.

\u Gibt an, dass der Quellenwert in Großbuchstaben konvertiert werden soll, bevor er hashverschlüsselt wird. Die zu löschenden Zeichen werden ebenfalls in Großbuchstaben konvertiert.

dest= Namen von Zieltabellenspalten, in die Werte aus der Suchtabelle eingefügt werden. (Für die Mehrspaltensuche erforderlich.)

col1,coln

Namen der Zieltabellenspalten. Die Reihenfolge der Spaltennamen muss den Suchtabellenspalten im Parameter *values=* entsprechen.

lktablename

Name der Suchtabelle. Sie können den Namen der Suchtabelle als **dbalias.creatorid.tablename**, **creatorid.tablename**, oder **tablename** angeben. Wenn Sie den Tabellennamen nicht vollständig qualifizieren, werden die Qualifikationsmerkmale für die Zieltabelle verwendet.

search Name der Spalte in der Suchtabelle, die sequenzielle Werte enthält, der mit den Hashwerten aus der Quellenspalte abgeglichen werden sollen.

value Name der Spalte in der Suchtabelle, die den umgesetzten Suchwert enthält, der am Ziel eingefügt werden soll. (Für die Einzelspaltensuche erforderlich.)

values=

Namen der Spalten in der Suchtabelle, die Werte enthalten, die am Ziel eingefügt werden sollen. (Für die Mehrspaltensuche erforderlich.)

col1,coln

Namen der Suchtabellenspalten. Die Reihenfolge der Spaltennamen muss den Zieltabellenspalten im Parameter *dest=* entsprechen.

cache | *nocache*

Geben Sie *cache* (Standardwert) an, um eine Tabelle mit gefundenen Suchwerten im Speicher zu behalten, oder geben Sie *nocache* an, um gefundene Werte zu löschen. Die Verwendung von *cache* ist beim häufigen Abrufen eines Werts zwar schneller, benötigt aber mehr Speicher.

ignore=

Liste der Quellenspalten mit Werten, die am Ziel anstelle des Suchwerts eingefügt werden, wenn die Spalte eine Zeile mit einem angegebenen Wert (NULL, SPACES, ZERO oder VARCHAR mit Nulllänge) hat.

col Der Name der Quellenspalte.

Geben Sie für die Einzelspaltsuche nur einen einzigen Spaltennamen ein.

Bei der Mehrspaltsuche muss die Reihenfolge der Spaltennamen muss den Zieltabellenspalten im Parameter *dest=* entsprechen. Die Anzahl der Spalten muss den Spalten im Parameter *dest=* entsprechen und mindestens eine Spalte muss einen Wert enthalten. Wenn Sie keine Werte für eine Spalte angeben wollen, geben Sie keinen Wert ein. Beispiel: *coln()*.

NULL Die Suchtabelle ignorieren, wenn die Quellenspaltenzeile den Wert NULL hat.

SPACES

Die Suchtabelle ignorieren, wenn die Quellenspaltenzeile den Wert SPACES hat. Nur für CHAR-Spalten.

ZERO_LEN

Die Suchtabelle ignorieren, wenn die Quellenspaltenzeile einen VARCHAR-Wert mit Nulllänge hat.

preserve=

Liste der Quellenspalten mit Werten, die am Ziel anstelle des Suchwerts eingefügt werden, wenn die Spalte einen angegebenen Wert (NOT_FOUND, NULL, SPACES oder VARCHAR mit Nulllänge) enthält.

NOT_FOUND

Die Suchtabelle ignorieren, wenn keine Übereinstimmung für die Quellenspaltenzeile gefunden wird.

Anmerkung:

preserve= und *ignore=* schließen sich gegenseitig aus. *ignore=* wird in einem späteren Release nicht weiter unterstützt.

Die Operanden *col*, *null*, *spaces* und *zero_len* haben den gleichen Effekt, wenn sie mit *preserve=* oder *ignore=* verwendet werden.

seed= Mit *seed=* können Sie die Berechnung des Hashalgorithmus ändern. Werte von 1 bis 2.000.000.000 können verwendet werden. Wenn Sie den Wert 0 verwenden, wird der Parameter *seed=* ignoriert.

Einzelspaltenbeispiel

Mit der Funktion für die Hashsuche können Sie Werte aus einer Spalte in einer Suchtabelle auf der Basis eines Werts, der aus einer Quellenspalte hashverschlüsselt wurde, in eine Zieltabellenspalte einfügen.

Es soll z. B. angenommen werden, dass die Quellenspalte, *FIRST_NAME*, Vornamen enthält und die Zielspalte Ersatzvornamen aus der Suchtabelle enthält. Eine Suchtabelle, *NAME_LOOKUP*, enthält eine Spalte (*FIRST*) mit Vornamen und eine Spalte (*SEQ*) mit sequenziellen Werten.

Wenn Sie Werte für die Zielspalte mithilfe der Tabelle *NAME_LOOKUP* abrufen wollen, geben Sie Folgendes an:

```
HASH_LOOKUP(FIRST_NAME,NAME_LOOKUP(SEQ, FIRST))
```

Die Funktion für die Hashsuche gleicht die Hashwerte aus der Quellenspalte mit Werten in der Spalte *SEQ* der Tabelle *NAME_LOOKUP* ab. Wenn eine Übereinstimmung festgestellt wird, fügt die Funktion den entsprechenden Wert aus der Spalte *FIRST* in die Zielspalte ein.

Mehrspaltenbeispiel

Mit der Funktion für die Hashsuche können Sie Werte aus Spalten in einer Suchtabellenzeile in Spalten in einer Zieltabellenzeile auf der Basis eines Werts, der aus einer Quellenspalte hashverschlüsselt wurde, einfügen.

Sie können z. B. auf der Basis von Werten, die aus einer Quellenspalte (FIRST_NAME) hashverschlüsselt wurden, die Vornamen enthält, Werte in Zielspalten (FIRST und LAST) mit Vor- und Nachnamen aus einer Suchtabelle ersetzen. Eine Suchtabelle mit dem Namen NAME_LOOKUP enthält eine Spalte (SEQ) mit sequenziellen Werten und Spalten (FIRST_MASK und LAST_MASK) zum Maskieren von Werten im Ziel.

Geben Sie zum Ersetzen von Namen in der Zieltabelle auf der Basis von Werten, die aus einer Quellenspalte hashverschlüsselt wurden, Folgendes an:

```
HASH_LOOKUP(FIRST_NAME,DEST=(FIRST, LAST), NAME_LOOKUP(SEQ,VALUES=(FIRST_MASK, LAST_MASK)))
```

Die Funktion für die Hashsuche gleicht die Hashwerte aus der Quellenspalte FIRST_NAME mit Werten in der Spalte SEQ der Tabelle NAME_LOOKUP ab. Wenn eine Übereinstimmung festgestellt wird, fügt die Funktion die entsprechenden Werte aus der Suchtabelle FIRST_MASK und LAST_MASK in die entsprechenden Zielspalten ein.

Ignore-Beispiel

Verwenden Sie die folgende Anweisung, um das Einzelspaltenbeispiel zu erweitern, wenn Sie die Quellenwerte *NULL* und *SPACES* anstelle von Suchtabellenwerten verwenden wollen:

```
HASH_LOOKUP(FIRST_NAME,NAME_LOOKUP(SEQ, FIRST),IGNORE=(FIRST_NAME(NULL,SPACES)))
```

NoCache-Beispiel

Verwenden Sie die folgende Anweisung, um das Einzelspaltenbeispiel zu erweitern, wenn Sie eine Tabelle mit gefundenen Suchwerten nicht im Speicher beibehalten wollen.

```
HASH_LOOKUP(FIRST_NAME,NAME_LOOKUP(SEQ, FIRST),NOCACHE)
```

Beispiel zum Löschen

Verwenden Sie die folgende Anweisung, um das Einzelspaltenbeispiel zu erweitern, wenn Sie vor der Hashverschlüsselung die Leerzeichen und Kommas aus dem Quellenwert löschen und den Quellenwert in Großbuchstaben konvertieren wollen:

```
HASH_LOOKUP(FIRST_NAME, TRIM=( ,\u),NAME_LOOKUP(SEQ,FIRST))
```

Funktion für Zufallssuche

Die Funktion für die Zufallssuche wählt zufällig einen Wert aus einer angegebenen Suchtabelle aus, der in eine Zielspalte eingefügt wird. Die Funktion generiert eine Zufallszahl zwischen 1 und dem Grenzwert oder der Anzahl der Zeilen in der Suchtabelle, die als Index in der Tabelle verwendet werden soll. Der Spaltenwert oder die Spaltenwerte aus der Zeile, der dem Index entspricht/entsprechen, wird/werden in die Zielspalte eingefügt.

Es gibt zwei Formate der Funktion für die Zufallssuche, das Einzelspaltenformat und das Mehrspaltenformat. Beim Einzelspaltenformat wird ein Wert in eine einzelne Zielspalte eingefügt. Beim Mehrspaltenformat werden Werte aus mehreren Suchtabellenspalten in die entsprechenden Zielspalten eingefügt.

Sie können die Funktion zur Zufallssuche mit mehreren Spalten für jede beliebige Quellenspalte eingeben, die durch einen Suchtabellenwert ersetzt wird, aber Sie müssen die Spaltenzuordnung so bearbeiten, dass die Namen verbleibender Quellenspalten, die ebenfalls ersetzt werden, entfernt werden.

Mit dem Parameter *ignore* können Sie die Suchtabelle ignorieren und einen Quellenwert verwenden, wenn eine Zeile in einer angegebenen Quellenspalte einen angegebenen Wert (NULL, SPACES (für CHAR-Spalten) oder VARCHAR mit Nulllänge) enthält.

Sie können den Parameter *preserve* verwenden, um die Suchtabelle zu ignorieren und einen Quellenwert zu verwenden, wenn eine Zeile in einer angegebenen Quellenspalte einen angegebenen Wert (NULL, SPACES (für CHAR-Spalten) oder VARCHAR mit Nulllänge) enthält.

Die Syntax ist:

```
RAND_LOOKUP(lktablename, { columnname | dest=(col1,coln) },values=(col1,coln) }  
[,limit] [,ignore=(col(spaces, null, zero_len), ) | PRESERVE=( colname (spaces, null, zero_len), ) ] )
```

lktablename

Name der Suchtabelle. Sie können den Namen der Suchtabelle als *dbalias.creatorid.tablename*, *creatorid.tablename*, oder *tablename* angeben. Wenn der Tabellename nicht vollständig qualifiziert ist, werden Zieltabellenqualifikationsmerkmale verwendet.

columnname

Name der Spalte in der Suchtabelle, die die zufällig auszuwählenden Werte zum Einfügen am Ziel enthält. (Für die Einzelspaltensuche erforderlich.)

dest= Namen von Zieltabellenspalten, in die Werte aus der Suchtabelle eingefügt werden. (Für die Mehrspaltensuche erforderlich.)

col1,coln

Namen der Zieltabellenspalten. Die Reihenfolge der Spaltennamen muss den Suchtabellenspalten im Parameter *values*= entsprechen.

values=

Namen der Spalten in der Suchtabelle, die Werte enthalten, die am Ziel eingefügt werden sollen. (Für die Mehrspaltensuche erforderlich.)

col1,coln

Namen der Suchtabellenspalten. Die Reihenfolge der Spaltennamen muss den Zieltabellenspalten im Parameter *dest*= entsprechen.

limit Optionaler Grenzwert für die Anzahl der Zeilen aus der Suchtabelle, der für die Auswahl von Spaltenwerten verwendet wird. Geben Sie eine Ganzzahl mit einem Maximalwert von 2.000.000.000 an. Wenn kein Grenzwert angegeben ist, werden alle Zeilen verwendet.

Anmerkung: Im Speicher wird eine Tabelle von Spaltenwerten erstellt. Die Größe dieser Tabelle kann durch Systemressourcen beschränkt sein.

ignore=

Liste der Quellenspalten mit Werten, die am Ziel anstelle des Suchwerts eingefügt werden, wenn die Spalte eine Zeile mit einem angegebenen Wert (NULL, SPACES oder VARCHAR mit Nulllänge) hat.

col Der Name der Quellenspalte.

Geben Sie für die Einzelspaltensuche nur einen einzigen Spaltennamen ein.

Bei der Mehrspaltensuche muss die Reihenfolge der Spaltennamen muss den Zieltabellenspalten im Parameter *dest*= entsprechen. Die Anzahl der Spalten muss den Spalten im Parameter *dest*= entsprechen und mindestens eine Spalte muss einen Wert enthalten. Wenn Sie keine Werte für eine Spalte angeben wollen, geben Sie keinen Wert ein. Beispiel: *coln*().

null Die Suchtabelle ignorieren, wenn die Quellenspaltenzeile den Wert NULL hat.

spaces Die Suchtabelle ignorieren, wenn die Quellenspaltenzeile den Wert SPACES hat. Nur für CHAR-Spalten.

zero_len Die Suchtabelle ignorieren, wenn die Quellenspaltenzeile einen VARCHAR-Wert mit Nulllänge hat.

preserve= Liste der Quellenspalten mit Werten, die am Ziel anstelle des Suchwerts eingefügt werden, wenn die Spalte einen angegebenen Wert (NOT_FOUND, NULL, SPACES oder VARCHAR mit Nulllänge) hat.

NOT_FOUND Die Suchtabelle ignorieren, wenn keine Übereinstimmung für die Quellenspaltenzeile gefunden wird.

Anmerkung:

preserve= und *ignore=* schließen sich gegenseitig aus. *ignore=* wird in einem späteren Release nicht weiter unterstützt.

Die Operanden *col*, *null*, *spaces* und *zero_len* haben den gleichen Effekt, wenn sie mit *preserve=* oder *ignore=* verwendet werden.

Einzelspaltenbeispiel

Wenn Sie einen Wert zufällig aus der Spalte STATE in den ersten 50 Zeilen einer Tabelle mit dem Namen STATE_LOOKUP auswählen wollen und ihn in die Zielspalte einfügen wollen, geben Sie Folgendes ein:

```
RAND_LOOKUP(STATE_LOOKUP,STATE,50)
```

Beispiel mit mehreren Spalten

Wenn Sie Werte aus den Spalten CITY, STATE und ZIPCODE in einer zufälligen Zeile einer Tabelle mit dem Namen STATE_LOOKUP auswählen wollen und sie in die entsprechenden Zielspalten einfügen wollen, geben Sie Folgendes ein:

```
RAND_LOOKUP(STATE_LOOKUP,  
DEST=(CITY,STATE,ZIPCODE),  
VALUES=(CITY,STATE,ZIP))
```

Ignore-Beispiel

Verwenden Sie die folgende Anweisung, um das Einzelspaltenbeispiel zu erweitern, wenn die Quellenspalte den Namen STATES hat und Sie die Quellenwerte NULL und SPACES anstelle Suchtabellenwerten verwenden wollen:

```
RAND_LOOKUP(STATE_LOOKUP,STATE,50, IGNORE=(STATES(NULL,SPACES)))
```

Mischfunktion

Die Mischfunktion ersetzt einen Quellenwert durch einen anderen Wert aus der Spalte, der dann in eine Zielspalte eingefügt wird. Die Quellenzeile und die Zeile, die den Ersatzwert enthält, sind nie identisch, aber abhängig von Ihren Daten können Quellenwert und Ersatzwert identisch sein.

Sie können angeben, wie oft die Funktion einen Ersatzwert aktualisiert, bis ein Wert gefunden wird, der nicht mit dem Quellenwert identisch ist ("Wiederholung"), oder Sie lassen zu, dass ein Ersatzwert mit dem Quellenwert übereinstimmt. Jede Mischfunktion funktioniert unabhängig von anderen Mischfunktionen, die in einer Spaltenzuordnung verwendet werden.

Es gibt zwei Formate der Mischfunktion, das Einzelspaltenformat und das Mehrspaltenformat. Beim Einzelspaltenformat wird ein Ersatzwert in eine einzelne Zielspalte eingefügt. Beim Mehrspaltenformat werden Ersatzwerte aus mehreren Spalten in einer Zeile in die entsprechenden Zielspalten eingefügt. Eine Spalte kann nicht in mehrere Mischfunktionen in einer Spaltenzuordnung aufgenommen werden. Wenn die Wiederholungsfunktion mit einem Mehrspaltenmischen verwendet wird, ruft die Funktion eine andere Ersatzzeile erneut ab, wenn ein beliebiger Wert in der Quellenzeilenspalte mit dem Wert in einer entsprechenden Ersatzzeilenspalte übereinstimmt. (Das Mehrspaltenformat kann nicht in einer Weitergabefunktion verwendet werden.)

Wenn Sie eine Mehrspaltenmischfunktion erstellen wollen, geben Sie die Funktion für eine Quellenspalte ein, die durch gemischte Werte ersetzt wird, und bearbeiten Sie die Spaltenzuordnung, um die Namen beliebiger anderer Quellenspalten mit Werten, die ebenfalls ersetzt werden, zu entfernen.

Der Parameter *ignore* verhindert, dass die Funktion eine Quellenspalte ersetzt oder eine Ersatzzeile verwendet, wenn eine davon einen angegebenen Wert (NULL, SPACES (für CHAR-Spalten) oder VARCHAR mit Nulllänge) enthält. Wenn keine Wiederholungen zulässig sind, gilt der Parameter *ignore* nicht für die Ersatzzeile.

Die Syntax ist:

```
SHUFFLE [ ( dest=(col1,coln) ) ] |
[ ( dest=(col1,coln) , retry[=number] ) ] |
[ ( dest=(col1,coln) [ , retry[=number] ] , ignore=( col1 ( [spaces] | [spaces,null]
| [spaces,null,zero_len] | [null] | [null,zero_len] | [zero_len] ) , coln (...) ) ) ] |
[ ( retry[=number] ) ] |
[ ( retry[=number] , ignore=( col ( [spaces] | [spaces,null] | [spaces,null,zero_len]
| [null] | [null,zero_len] | [zero_len] ) ) ) ] |
[ ( ignore=( col ( [spaces] | [spaces,null] | [spaces,null,zero_len]
| [null] | [null,zero_len] | [zero_len] ) ) ) ]
```

Dabei gilt Folgendes:

dest= Namen der Zieltabellenspalten, in die Ersatzwerte eingefügt werden. (Für die Mehrspaltenmischung erforderlich.)

col1, coln, ...

Namen der Zieltabellenspalten.

retry Anzahl der Male, die ein Ersatzwert erneut abgerufen werden soll, um einen Wert zu finden, der nicht mit der Quellenspalte übereinstimmt. Geben Sie Null ein, wenn ein Ersatzwert mit der Quelle übereinstimmen darf.

Anmerkung: Wenn eine hohe Anzahl Wiederholungen mit Spalten verwendet wird, die viele doppelte Werte enthalten, verlängert sich die Verarbeitungszeit. Für diese Spalten sollte der Wiederholungswert 0 verwendet werden.

=number

Geben Sie einen Wert im Bereich von 0 bis 1000 ein. Geben Sie 0 ein, um zu ermöglichen, dass ein Ersatzwert mit der Quelle übereinstimmt.

ignore=

Liste von Spalten, für die die Funktion keinen Quellenwert ersetzt oder keinen Ersatzwert verwendet, wenn einer davon ein angegebener Wert ist (NULL, SPACES (für CHAR-Spalten) oder VARCHAR mit Nulllänge). Wenn ein Ersatzwert ignoriert wird, ruft die Funktion einen anderen Ersatzwert ab. Wenn keine Wiederholungen zulässig sind, gilt der Parameter *ignore=* nicht für Ersatzwerte.

col Der Name der Quellenspalte.

Geben Sie für die Einzelspaltenmischung nur einen einzigen Spaltennamen ein.

Bei der Mehrspaltenmischung muss die Reihenfolge der Spaltennamen den Zieltabellenspalten im Parameter *dest=* entsprechen. Die Anzahl der Spalten muss den Spalten im Parameter *dest=* entsprechen und mindestens eine Spalte muss einen Wert enthalten. Wenn Sie keine Werte für eine Spalte angeben wollen, geben Sie keinen Wert ein. Beispiel: *coln()*.

null Nicht den Quellenwert ersetzen oder einen Ersatzwert verwenden, wenn einer davon ein NULL-Wert ist.

spaces Nicht den Quellenwert ersetzen oder einen Ersatzwert verwenden, wenn einer davon ein SPACES-Wert ist. Nur für CHAR-Spalten.

zero_len Nicht den Quellenwert ersetzen oder einen Ersatzwert verwenden, wenn einer davon ein VARCHAR-Wert mit Nulllänge ist.

Einzelspaltenstandardbeispiel

Im folgenden Beispiel werden gemischte Werte in einer Einzelspalte eingefügt.

```
SHUFFLE
```

Einzelspaltenwiederholungsbeispiel

Im folgenden Beispiel werden gemischte Werte in einer einzelnen Spalte eingefügt und bis zu 12 Mal ein Ersatzwert erneut abgerufen, der nicht mit der Quelle übereinstimmt.

```
SHUFFLE(RETRY=12)
```

Mehrspaltenbeispiel

Im folgenden Beispiel werden gemischte Werte in die Spalten STATE und ZIP eingefügt und bis zu 12 Mal ein Ersatzwert erneut abgerufen, der nicht mit der Quelle übereinstimmt.

```
SHUFFLE(DEST=(STATE,ZIP),RETRY=12)
```

Ignore-Beispiel

Im folgenden Beispiel werden gemischte Werte in die Spalten STATE und ZIP eingefügt und bis zu 12 Mal ein Ersatzwert erneut abgerufen, der nicht mit der Quelle übereinstimmt. Im Beispiel wird auch für die Spalte STATE kein Quellenwert ersetzt oder kein Ersatzwert verwendet, wenn eine Quelle oder eine Ersatzzeile einen NULL- oder SPACES-Wert enthält, aber es werden nicht irgendwelche Quellen- oder Ersatzzeilen für die Spalte ZIP ignoriert.

```
SHUFFLE(DEST=(STATE,ZIP),RETRY=12,  
IGNORE=(STATE(NULL,SPACES),ZIP()))
```

Funktion TRANS SSN

Mit der Funktion TRANS SSN können Sie eine gültige und eindeutige US-amerikanische Sozialversicherungsnummer (SSN-Nummer) generieren. Standardmäßig generiert TRANS SSN auf der Basis der angegebenen Algorithmen eine konsistent geänderte SSN-Nummer auf der Basis der Quellen-SSN-Nummer. TRANS SSN kann auch eine Zufalls-SSN-Nummer generieren, wenn in den Quelldaten keinen SSN-Nummernwert vorhanden ist oder wenn die Quellen-SSN-Nummer nicht konsistent umgesetzt werden muss.

Eine Sozialversicherungsnummer (SSN-Nummer) besteht aus 3 Unterfeldern. Die ersten drei Ziffern (Bereich) stellen einen Bereich dar, der normalerweise durch den Bundesstaat bestimmt wird, in dem die SSN-Nummer ausgegeben wird. Die beiden nächsten Ziffern (Gruppe) definieren eine Gruppennummer,

die der Bereichsnummer entspricht. Die letzten vier Ziffern (fortlaufende Nummer) sind eine fortlaufende Nummer. Unabhängig vom Typ der Verarbeitung (Standard oder Zufall) generiert die Funktion TRANS SSN eine SSN-Nummer mit einer für die Bereichsnummer geeigneten Gruppennummer.

Mit der Standardverarbeitungsmethode wird eine SSN-Nummer generiert, die die Quellenbereichsnummer sowie geänderte Gruppen- und Seriennummern auf der Basis der Quellen-SSN-Nummer enthält.

Mit der Zufallsverarbeitungsmethode wird eine SSN-Nummer generiert, die eine Quellenbereichsnummer enthalten kann und eine Gruppennummer verwendet, die zuletzt von der Social Security Administration (SSA) für die Zielbereichsnummer ausgegeben wurde. Seriennummern beginnen mit 0001 und werden für jede weitere Sozialversicherungsnummer, die für die Bereichsnummer generiert wird, jeweils um 1 erhöht. Wenn die Seriennummer 9999 überschreitet, wird sie auf 0001 zurückgesetzt und die Gruppennummer vor der Nummer, die zuletzt für die Bereichsnummer ausgegeben wurde, wird verwendet.

Die Syntax von TRANS SSN ist:

```
TRANS SSN [( '[=flags] [sourcecol [preserve=invalid]' )]
```

flags Sie können mindestens ein Verarbeitungsoptionsflag ohne Berücksichtigung von Groß-/ Kleinschreibung angeben.

n Eine Zufalls-SSN-Nummer generieren, die nicht auf einem Quellenwert basiert.

m Die maximale Gruppe aller SSN-Nummernbereichswerte einschließlich Werten von 773 bis 899 und ausschließlich ungültiger Bereichsnummern verwenden.

r Eine SSN-Nummer mit einer Zufallsbereichsnummer generieren, die demselben Staat wie die Eingabe-SSN-Nummer entspricht.

v Die Quellengruppennummer durch Vergleich mit Nummern, die von der Social Security Administration verwendet werden, überprüfen.

- Die Ziel-SSN-Nummer soll Gedankenstriche als Trennzeichen für die Felder enthalten (z. B. 123-45-6789). Erfordert eine alphanumerische Zielspalte mit einer Länge von mindestens 11 Zeichen.

sourcecol

Der Name der Quellenspalte. Wenn kein Name für eine Quellenspalte angegeben ist, wird der Name der Zielspalte verwendet. Wenn kein Name für die Quellenspalte angegeben wird und der Name der Zielspalte keinem Spaltennamen in der Quellentabelle entspricht, tritt ein Fehler während der Verarbeitung aus.

preserve=invalid

Wenn die Quellenspalte eine ungültige SSN-Nummer enthält, diese nicht durch einen generierten Wert ersetzen. Der Wert der Quellenspalte wird in der Zielspalte verwendet.

Zulässige Datentypen

Die folgenden Quellen- und Zieldatentypen werden unterstützt:

CHAR

Die Länge der Daten in der Spalte muss 9 bis 256 Zeichen betragen.

DECIMAL

Die Genauigkeit der Spalte muss von 9 bis 20 liegen und die Anzahl der Kommastellen muss 0 sein.

INTEGER

Keine Einschränkungen.

VARCHAR

Die Länge der Daten in der Spalte muss 9 bis 254 Zeichen betragen.

Wenn eine Quellen- oder Zielspalte diese Beschränkungen nicht erfüllt, tritt ein Fehler bei der Verarbeitung auf.

Verarbeitungsregeln für Ziel

Die folgenden Regeln gelten je nach Zieldatentyp oder -wert für den Wert der Ziel-SSN-Nummer:

CHAR

Bei einem Quellenwert von 0, Leerzeichen (SPACES) oder einem VARCHAR-Typ mit Nulllänge wird der Zielwert auf SPACES gesetzt.

Bei einem Quellenwert mit 11 oder mehr Zeichen und eingebetteten Gedankenstrichen (-) oder wenn das Flag '-' angegeben ist, enthält auch der Zielwert Gedankenstriche, wenn die Länge der Zielspalte 11 oder mehr Zeichen aufweist.

DECIMAL, INTEGER

Bei einem Quellenwert von 0, Leerzeichen (SPACES) oder einem VARCHAR-Typ mit Nulllänge ist der Zielwert 0.

VARCHAR

Bei einem Quellenwert von 0, Leerzeichen (SPACES) oder einem VARCHAR-Typ mit Nulllänge ist die Ziellänge 0.

Bei einem Quellenwert mit 11 oder mehr Zeichen und eingebetteten Gedankenstrichen (-) oder wenn das Flag '-' angegeben ist, enthält auch der Zielwert Gedankenstriche, wenn die Länge der Zielspalte 11 oder mehr Zeichen aufweist.

NULL Wenn der Quellenwert NULL ist, ist auch der Zielwert NULL.

Übersprungene Zeilen

Die folgenden Bedingungen können dazu führen, dass eine Quellenzeile übersprungen und nicht in die Zieltabelle geschrieben wird:

- Der Quellenwert ist NULL und die Zielspalte lässt keinen Wert NULL zu.
- Die Quellenspalte ist eine CHAR- oder VARCHAR-Spalte und der Quellenwert hat weniger als 9 Zeichen, enthält ein nicht numerisches Zeichen (abgesehen von den Gedankenstrichen zwischen den drei Unterfeldern) oder ist zu groß.
- Die Quellenbereichsnummer wurde nicht von der Social Security Administration (SSA) verwendet.
- Die Quellengruppennummer wurde nicht mit der Bereichsnummer von der Social Security Administration (SSA) verwendet (nur bei Angabe des Flags 'v').
- Die Quellenseriennummer ist 0000 oder die SSN-Nummer ist ein reservierter Wert, der nicht von der Social Security Administration ausgegeben wurde (z. B. 078-05-1120).
- Der Quellenwert kann nicht in ein Format konvertiert werden, das TRANS SSN unterstützt.

Fehlernachrichten

Die folgenden Fehlernachrichten können ausgegeben werden:

SSN01

Parm on Col ccccc ("ppp") is invalid

Erläuterung

Die Spalte enthält eine Funktion TRANS mit einem ungültigen Verarbeitungsoptionsflag.

Benutzeraktion

Stellen Sie sicher, dass die für die Spalte angegebene Funktion TRANS ein gültiges Verarbeitungsoptionsflag (n, r, v, -) verwendet.

SSN02

Col ccccc not on source

Erläuterung

Die Spalte, die als Parameter *sourcecol* eingegeben wurde, oder der Name der Zielspalte (wenn der Parameter *sourcecol* nicht angegeben wurde) wurde in der Quellentabelle nicht gefunden.

Benutzeraktion

Prüfen Sie die Quellentabelle und korrigieren Sie eventuelle Abweichungen oder geben Sie fehlende Spalten an.

SSN03

Source Col ccccc-aaa invalid

Erläuterung

Das Format der Quellenspalte wird nicht unterstützt, da das angegebene Attribut nicht gültig ist.

Benutzeraktion

Prüfen Sie die Quellenspalte und stellen Sie sicher, dass die Werte für Typ, Länge, Genauigkeit und Anzahl der Kommastellen geeignet sind.

SSN04

Dest Col ccccc-aaa invalid

Erläuterung

Das Format der Zielspalte wird nicht unterstützt, da das angegebene Attribut nicht gültig ist.

Benutzeraktion

Prüfen Sie die Zielspalte und stellen Sie sicher, dass die Werte für Typ, Länge, Genauigkeit und Anzahl der Kommastellen geeignet sind.

SSN05

Get col ccccc data-rc=nnn

Erläuterung

Ein nicht erwarteter, interner Fehler ist beim Abrufen des Werts aus der Quellenspalte aufgetreten.

Benutzeraktion

Prüfen Sie die Werte der Quellen- und Zielspalten und stellen Sie sicher, dass die Werte für Typ, Länge, Genauigkeit und Anzahl der Kommastellen geeignet sind. Falls das Problem bestehen bleibt, wenden Sie sich an den IBM Software Support.

SSN08

Put col ccccc data-rc=nnn

Erläuterung

Ein nicht erwarteter, interner Fehler ist beim Festlegen des Werts für die Zielspalte aufgetreten.

Benutzeraktion

Prüfen Sie die Werte der Quellen- und Zielspalten und stellen Sie sicher, dass die Werte für Typ, Länge, Genauigkeit und Anzahl der Kommastellen geeignet sind. Falls das Problem bestehen bleibt, wenden Sie sich an den IBM Software Support.

Falls andere Fehler auftreten, wenden Sie sich an den Technical Support.

Beispiel 1

Im folgenden Beispiel wird ein Quellenspaltenname verwendet, der mit der Zielspalte übereinstimmt, und eine Zufalls-SSN-Nummer generiert, die nicht auf dem Quellenwert basiert:

```
TRANS SSN ('=n')
```

Beispiel 2

Im folgenden Beispiel wird ein Quellenspaltenname (NATIONAL_ID) verwendet, der von der Zielspalte abweicht, und über die Standardverarbeitungsmethode eine SSN-Nummer mit Gedankenstrichen generiert:

```
TRANS SSN ('=- NATIONAL_ID')
```

Funktion TRANS CCN

Verwenden Sie die Funktion TRANS CCN zum Generieren einer gültigen und eindeutigen Kreditkartennummer (CCN). Standardmäßig generiert TRANS CCN auf der Basis der angegebenen Algorithmen eine konsistent geänderte CCN auf der Basis der Quellen-CCN. TRANS CCN kann ebenfalls einen Zufallswert generieren, wenn die Quelldaten keinen CCN-Wert haben oder wenn das Umsetzen der Quellen-CCN auf konsistente Weise nicht erforderlich ist.

Eine Kreditkartennummer besteht laut Definition in ISO 7812 aus einer 6-stelligen Aussteller-ID, an die sich eine Kontonummer mit variabler Länge und eine abschließende einzelne Prüfziffer anschließen. Die Prüfziffer bestätigt die Richtigkeit der Kreditkartennummer und wird generiert, indem die ID-Nummer und die Kontonummer des Kreditkartenausstellers dem Luhn-Algorithmus unterzogen werden. Die maximale Länge einer Kreditkartennummer beträgt 19 Ziffern.

Die Standardverarbeitungsmethode generiert eine CCN durch Einbeziehen der ersten vier Ziffern der Aussteller-ID aus der Quellen-CCN und Ändern der verbleibenden zwei Ziffern der Aussteller-ID-Nummer und der Kontonummer auf der Basis der Quellen-CCN. Außerdem wird eine gültige Prüfziffer zugeordnet.

Die Zufallsverarbeitungsmethode generiert eine CCN, die die ersten vier Ziffern der Quellenaussteller-ID-Nummer oder eine Aussteller-ID-Nummer, die American Express, Discover, MasterCard oder VISA zugeordnet ist, enthalten kann. Außerdem wird eine gültige Prüfziffer zugeordnet. Wenn die ersten vier Ziffern einer Quellenaussteller-ID-Nummer enthalten sind, beginnt die erste Kontonummer auf der Basis dieser Ziffern mit 1 und für jede zusätzliche CCN, die diese Ziffern verwendet, wird die Kontonummer um 1 erhöht.

Die Syntax von TRANS CCN ist:

```
TRANS CCN [( 'flag' [sourcecol] [preserve=invalid]' )]
```

flag Geben Sie ein Optionsflag zum Generieren einer Zufalls-CCN an.

n Generieren Sie eine Zufalls-CCN, die nicht auf einem Quellenwert basiert und eine Aussteller-ID-Nummer enthält, die American Express, Discover, MasterCard oder VISA zugeordnet ist.

r Generieren Sie eine Zufalls-CCN, die die ersten vier Ziffern der Quellenaussteller-ID-Nummer enthält.

sourcecol

Der Name der Quellenspalte. Wenn kein Name für die Quellenspalte angegeben wird, wird der Name der Zielspalte verwendet.

Wenn kein Name für die Quellenspalte angegeben wird und der Name der Zielspalte keinem Spaltennamen in der Quellentabelle entspricht, tritt ein Fehler während der Verarbeitung aus.

preserve=invalid

Wenn die Quellenspalte eine ungültige CCN enthält, ersetzen Sie diese nicht durch einen generierten Wert. Der Wert der Quellenspalte wird in der Zielspalte verwendet.

Zulässige Datentypen

Die folgenden Quellen- und Zieldatentypen werden unterstützt:

CHAR

Die Spaltenlänge muss zwischen 13 und 256 Zeichen liegen.

VARCHAR

Die Spaltenlänge muss zwischen 13 und 254 Zeichen liegen.

DECIMAL

Die Genauigkeit der Spalte muss zwischen 13 und 254 liegen und die Anzahl der Kommastellen muss 0 sein.

Wenn diese Einschränkungen für eine Quellen- oder Zielspalte nicht beachtet werden, wird eine Fehlermeldung ausgegeben.

Verarbeitungsregeln für Ziel

Die folgenden Regeln gelten für den Ziel-CNN-Wert entsprechend des Zieldatentyps und Zielwerts:

CHAR

Bei einem Quellenwert von Leerzeichen (SPACES) oder einem VARCHAR-Typ mit Nulllänge wird der Zielwert auf Leerzeichen gesetzt.

VARCHAR

Bei einem Quellenwert von Leerzeichen (SPACES) oder einem VARCHAR-Typ mit Nulllänge ist die Ziellänge 0.

DECIMAL

Wenn der Quellenwert 0 ist, ist auch der Zielwert 0.

NULL Wenn der Quellenwert NULL ist, ist auch der Zielwert NULL.

Übersprungene Zeilen

Die folgenden Bedingungen können dazu führen, dass eine Quellenzeile übersprungen und nicht in die Zieltabelle geschrieben wird:

- Der Quellenwert ist NULL und die Zielspalte lässt keinen Wert NULL zu.
- Der Quellenwert weist weniger als 13 Zeichen auf, enthält ein nicht numerisches Zeichen, ist zu groß oder hat eine falsche Prüfziffer.
- Der Quellenwert ist für den Kreditkartenaussteller nicht gültig.
- Der Quellenwert kann nicht in ein Format konvertiert werden, das von TRANS CCN unterstützt wird.

Fehlernachrichten

Die folgenden Fehlernachrichten können ausgegeben werden:

CCN01

Parm on Col cccc ("ppp") is invalid

Erläuterung

Die angegebene Spalte enthält eine Funktion TRANS mit einem ungültigen Verarbeitungsoptionsflag.

Benutzeraktion

Stellen Sie sicher, dass die für die Spalte angegebene Funktion TRANS ein gültiges Verarbeitungsoptionsflag (n, r, 6) verwendet.

CCN02

Col cccc not on source

Erläuterung

Die Spalte, die als Parameter *sourcecol* eingegeben wurde, oder der Name der Zielspalte (wenn der Parameter *sourcecol* nicht angegeben wurde) wurde in der Quellentabelle nicht gefunden.

Benutzeraktion

Prüfen Sie die Quellentabelle und korrigieren Sie eventuelle Abweichungen oder geben Sie fehlende Spalten an.

CCN03

Source Col cccc-aaa invalid

Erläuterung

Das Format der Quellenspalte wird nicht unterstützt, da das angegebene Attribut nicht gültig ist.

Benutzeraktion

Prüfen Sie die Quellenspalte und stellen Sie sicher, dass die Werte für Typ, Länge, Genauigkeit und Anzahl der Kommastellen geeignet sind.

CCN04

Dest Col cccc-aaa invalid

Erläuterung

Das Format der Zielspalte wird nicht unterstützt, da das angegebene Attribut nicht gültig ist.

Benutzeraktion

Prüfen Sie die Zielspalte und stellen Sie sicher, dass die Werte für Typ, Länge, Genauigkeit und Anzahl der Kommastellen geeignet sind.

CCN05

Get col cccc data-rc=nnn

Erläuterung

Ein nicht erwarteter, interner Fehler ist beim Abrufen des Werts aus der Quellenspalte aufgetreten.

Benutzeraktion

Prüfen Sie die Werte der Quellen- und Zielspalten und stellen Sie sicher, dass die Werte für Typ, Länge, Genauigkeit und Anzahl der Kommastellen geeignet sind. Falls das Problem bestehen bleibt, wenden Sie sich an den IBM Software Support.

CCN08

Put col cccc data-rc=nnn

Erläuterung

Ein nicht erwarteter, interner Fehler ist beim Festlegen des Werts für die Zielspalte aufgetreten.

Benutzeraktion

Prüfen Sie die Werte der Quellen- und Zielspalten und stellen Sie sicher, dass die Werte für Typ, Länge, Genauigkeit und Anzahl der Kommastellen geeignet sind. Falls das Problem bestehen bleibt, wenden Sie sich an den IBM Software Support.

Falls andere Fehler auftreten, wenden Sie sich an den Technical Support.

Beispiel 1

Im folgenden Beispiel wird ein Quellenspaltenname (CREDITCARD) verwendet, der von der Zielspalte abweicht und eine Zufalls-CCN generiert, die nicht auf dem Quellenwert basiert:

```
TRANS CCN ('=n CREDITCARD')
```

Beispiel 2

Im folgenden Beispiel wird ein Quellenspaltenname (CREDITCARD) verwendet, der von der Zielspalte abweicht und mit der Standardverarbeitungsmethode eine CCN generiert:

```
TRANS CCN ('CREDITCARD')
```

Funktion TRANS EML

Mit der Funktion TRANS EML kann eine E-Mail-Adresse generiert werden. Eine E-Mail-Adresse besteht aus zwei Teilen, einem Benutzernamen und einem nachfolgenden Domännennamen, getrennt durch '@'. Beispiel: benutzer@domäne.com.

TRANS EML generiert eine E-Mail-Adresse mit einem Benutzernamen auf der Basis von Zieldaten oder auf der Basis eines Literals, das mit einer fortlaufenden Zahl verkettet ist. Der Domänenname kann auf einer E-Mail-Adresse in den Quelldaten oder einem Literal basieren oder zufällig aus einer Liste großer E-Mail-Service-Provider ausgewählt sein. Die E-Mail-Adresse kann auch in Groß- oder Kleinschreibung konvertiert werden.

TRANS EML kann einen Benutzernamen auf der Basis der Werte in einer Zieltabellenspalte oder zwei Zieltabellenspalten (in der Regel mit dem Namen eines Benutzers) generieren. Über Verarbeitungsoptionen können Sie bewirken, dass nur das erste Zeichen des Werts in der ersten Spalte verwendet wird (z. B. der Anfangsbuchstabe eines Vornamens), und dass die Werte aus beiden Spalten durch einen Punkt oder ein Unterstreichungszeichen getrennt werden.

Wenn der Benutzername auf einem einzelnen Zielspaltenwert oder einem Literal basiert, wird er mit einer fortlaufenden Zahl verkettet. Wenn ein Benutzername auf Werten in zwei Zieltabellenspalten basiert und kein Punkt oder Unterstreichungszeichen für die Trennung verwendet wird, werden die Werte verkettet. Wenn kein Parameter für den Benutzernamen bereitgestellt wird, wird der Name durch das Literal 'email' gebildet, das mit einer fortlaufenden Zahl verkettet wird. Fortlaufende Zahlen für Benutzernamen sind Suffixe, die mit 1 beginnen und um 1 erhöht werden.

TRANS EML hat folgende Syntax:

```
TRANS EML [( '[=flags] , [{sourcecol | "domain" | , }  
[ {name1col[name2col] | "userpfx"} ] [preserve=invalid]' )]
```

flags Sie können mindestens ein Verarbeitungsoptionsflag ohne Berücksichtigung von Groß-/Kleinschreibung angeben.

- n* Generieren Sie aus einer Liste großer E-Mail-Serviceanbieter einen Zufallsdomännennamen.
- .* Trennen Sie *name1col*- und *name2col*-Werte mit einem Punkt.
- _* Trennen Sie *name1col*- und *name2col*-Werte mit einem Unterstreichungszeichen.
- i* Verwenden Sie nur das erste Zeichen des *name1col*-Werts.
- l* Konvertieren Sie die E-Mail-Adresse in Kleinbuchstaben.
- u* Konvertieren Sie die E-Mail-Adresse in Großbuchstaben.

sourcecol

Der Quellenspaltenname mit E-Mail-Adressen, der zum Bereitstellen des Domänennamens verwendet wird.

Wenn weder das Flag 'n' noch der Parameter *domain* definiert sind, wird der Domänenname in der Quellenspalte verwendet. (Wenn *sourcecol* nicht definiert ist, basiert der Quellenspaltenname auf dem Zielspaltennamen.)

Wenn kein Name für die Quellenspalte angegeben wird und der Name der Zielspalte keinem Spaltennamen in der Quellentabelle entspricht, tritt ein Fehler während der Verarbeitung aus.

domain Ein Literal mit bis zu 31 Zeichen, das den Domänennamen bildet.

,

Ein Komma ist erforderlich, wenn weder ein Parameter *sourcecol* noch ein Parameter *domain* definiert ist und Sie entweder ein Literal oder einen oder mehrere Spaltennamen für den Domänennamen definieren.

name1col

Ein Zieltabellenspaltenname mit Werten, die zum Bilden des ersten (oder einzigen) Teils des Benutzernamens verwendet werden.

name2col

Ein Zieltabellenspaltenname mit Werten, die zum Bilden des zweiten Teils des Benutzernamens verwendet werden.

userpfx Ein Literal mit bis zu 31 Zeichen, das zur Bildung des Benutzernamens mit einer fortlaufenden Zahl verkettet wird.

preserve=invalid

Wenn die Quellenspalte eine ungültige E-Mail-Adresse enthält, diese nicht durch einen generierten Wert ersetzen. Der Wert der Quellenspalte wird in der Zielspalte verwendet.

Zulässige Datentypen

Die folgenden Quellen- und Zieldatentypen werden unterstützt:

CHAR

Die Spaltenlänge muss zwischen 3 und 4096 Zeichen liegen.

VARCHAR

Die Spaltenlänge muss zwischen 3 und 4094 Zeichen liegen.

Wenn diese Einschränkungen für eine Quellen- oder Zielspalte nicht beachtet werden, wird eine Fehlermeldung ausgegeben.

Verarbeitungsregeln für Ziel

Folgende Regeln gelten für den E-Mail-Zielwert je nach Zieldatentyp oder -wert:

CHAR

Bei einem Quellenwert von Leerzeichen (SPACES) oder einem VARCHAR-Typ mit Nulllänge wird der Zielwert auf Leerzeichen gesetzt.

VARCHAR

Bei einem Quellenwert von Leerzeichen (SPACES) oder einem VARCHAR-Typ mit Nulllänge ist die Ziellänge 0.

NULL Wenn der Quellenwert NULL ist, ist auch der Zielwert NULL.

Übersprungene Zeilen

Die folgenden Bedingungen können dazu führen, dass eine Quellenzeile übersprungen und nicht in die Zieltabelle geschrieben wird:

- Der Quellenwert ist NULL und die Zielspalte lässt keinen Wert NULL zu.
- Der Quellenwert ist ein VARCHAR-Wert, der kürzer als drei Zeichen ist.
- Der Quellenwert für die E-Mailadresse enthält das Zeichen '@' nicht.
- Der Quellenwert kann nicht in ein Format konvertiert werden, das von TRANS EML unterstützt wird.

Fehlernachrichten

Die folgenden Fehlernachrichten können ausgegeben werden:

EML01

Parm on Col cccc ("ppp") is invalid

Erläuterung

Die angegebene Spalte enthält eine Funktion TRANS mit einem ungültigen Verarbeitungsoptionsflag.

Benutzeraktion

Stellen Sie sicher, dass die für die Spalte angegebene Funktion TRANS ein gültiges Verarbeitungsoptionsflag (n, ., _, i, l, u) verwendet.

EML02

Col cccc not on source

Erläuterung

Die Spalte, die als Parameter *sourcecol* eingegeben wurde, oder der Name der Zielspalte (wenn der Parameter *sourcecol* nicht angegeben wurde) wurde in der Quellentabelle nicht gefunden.

Benutzeraktion

Prüfen Sie die Quellentabelle und korrigieren Sie eventuelle Abweichungen oder geben Sie fehlende Spalten an.

EML03

Source Col cccc-aaa invalid

Erläuterung

Das Format der Quellenspalte wird nicht unterstützt, da das angegebene Attribut nicht gültig ist.

Benutzeraktion

Prüfen Sie die Quellenspalte und stellen Sie sicher, dass die Werte für Typ, Länge, Genauigkeit und Anzahl der Kommastellen geeignet sind.

EML04

Dest Col cccc-aaa invalid

Erläuterung

Das Format der Zielspalte wird nicht unterstützt, da das angegebene Attribut nicht gültig ist.

Benutzeraktion

Prüfen Sie die Zielspalte und stellen Sie sicher, dass die Werte für Typ, Länge, Genauigkeit und Anzahl der Kommastellen geeignet sind.

EML05

Get col cccc data-rc=nnn

Erläuterung

Ein nicht erwarteter, interner Fehler ist beim Abrufen des Werts aus der Quellenspalte aufgetreten.

Benutzeraktion

Prüfen Sie die Werte der Quellen- und Zielspalten und stellen Sie sicher, dass die Werte für Typ, Länge, Genauigkeit und Anzahl der Kommastellen geeignet sind. Falls das Problem bestehen bleibt, wenden Sie sich an den IBM Software Support.

EML08

Put col cccc data-rc=nnn

Erläuterung

Ein nicht erwarteter, interner Fehler ist beim Festlegen des Werts für die Zielspalte aufgetreten.

Benutzeraktion

Prüfen Sie die Werte der Quellen- und Zielspalten und stellen Sie sicher, dass die Werte für Typ, Länge, Genauigkeit und Anzahl der Kommastellen geeignet sind. Falls das Problem bestehen bleibt, wenden Sie sich an den IBM Software Support.

EML09

Domain literal ssss too long

Erläuterung

Die als Domänennamensliteral (Domäne) angegebene Zeichenfolge überschreitet den Maximalwert von 31 Zeichen.

Benutzeraktion

Geben Sie einen Domänennamen an, der aus maximal 31 Zeichen besteht.

EML10

User literal ssss too long

Erläuterung

Die als Benutzernamensliteral (*userpfx*) angegebene Zeichenfolge überschreitet den Maximalwert von 31 Zeichen.

Benutzeraktion

Geben Sie einen Benutzernamen an, der aus maximal 31 Zeichen besteht.

EML11

Name1 Col cccc not on dest

Erläuterung

Damit die angegebene Funktion TRANS ausgeführt werden kann, muss eine Spalte *name1col* in der Zieltabelle angegeben werden.

Benutzeraktion

Prüfen Sie, ob die angegebene Spalte *name1col* mit der Spalte *name1col* übereinstimmt, die in der Funktion TRANS angegeben wurde.

Der Spaltenname *name1col* wurde nicht in der Zieltabelle gefunden.

EML12

Name1 Col cccc-aaa invalid

Erläuterung

Damit die angegebene Funktion TRANS ausgeführt werden kann, muss die angegebene Spalte *name1col* in einem gültigen Format vorliegen.

Benutzeraktion

Überprüfen Sie die Verfügbarkeit, den Typ und die Länge der Spalte *name1col*.

EML13

Name2 Col cccc not on dest

Erläuterung

Damit die angegebene Funktion TRANS ausgeführt werden kann, muss eine Spalte *name2col* in der Zieltabelle angegeben werden.

Benutzeraktion

Prüfen Sie, ob die angegebene Spalte *name2col* mit der Spalte *name2col* übereinstimmt, die in der Funktion TRANS angegeben wurde.

Der Name der Spalte *name2col* wurde nicht in der Zieltabelle gefunden.

EML14

Name2 Col cccc-aaa invalid

Erläuterung

Damit die angegebene Funktion TRANS ausgeführt werden kann, muss die angegebene Spalte *name2col* in einem gültigen Format vorliegen.

Benutzeraktion

Überprüfen Sie die Verfügbarkeit, den Typ und die Länge der Spalte *name2col*.

Falls andere Fehler auftreten, wenden Sie sich an den Technical Support.

Beispiel 1

Das folgende Beispiel verwendet ein Literal (optim.com) zum Bilden des Domänennamens und zwei Zieltabellenspalten (NAME_FIRST und NAME_LAST) zum Bilden eines Benutzernamens, der ein Unterstrichungszeichen enthält:

```
TRANS EML ('=_ "optim.com" NAME_FIRST NAME_LAST')
```

Beispiel 2

Im folgenden Beispiel wird ein Domänenname aus einer Quellenspalte und ein Literal (OptimUser) zum Bilden eines Benutzernamens verwendet, dem als Suffix eine fortlaufende Zahl hinzugefügt wird:

```
TRANS EML (' , "OptimUser" ')
```

Funktion TRANS COL

Mithilfe der Funktion TRANS COL können Sie Daten maskieren, die kein inhärentes Format oder ein relativ unbekanntes Format haben. Mit TRANS COL werden das Format und der Zeichentyp der Quelldaten am Ziel beibehalten.

Wenn die Quelldaten als alphabetische Zeichen in Großbuchstaben vorliegen, generiert TRANS COL alphabetische Zeichen in Großbuchstaben am Ziel. Diese Funktion maskiert alphabetische und numerische Zeichen, aber andere Zeichen in den Quelldaten werden zum Ziel kopiert, ohne dass sie geändert werden. Sie können TRANS COL verwenden, um CHAR, VARCHAR numerische Datentypen, die kein Gleitkomma verwenden, zu maskieren. Sie können eindeutige Werte, einen anderen Wert für jedes Auftreten derselben Quelle und Werte mit einer anderen Länge als die der Quelle generieren.

Die Syntax von TRANS COL lautet:

```
TRANS COL ( '{ unique | hash }[ source=colname ]  
[ copy=( start, len [, "lit" ] )... ]  
[ seed= {"lit" | var (variable )| RANDOM} ]
```

[length=*n* | max] [preserve=([null] [spaces] [zero_len])]
[TRIM=(*char1* [*charn*,...] [\u] [\r])] [num]')

unique Einen eindeutigen Zielwert generieren. Die Länge des Zielwerts stimmt mit der Quellenwertlänge überein.

hash Einen Zielwert durch Hashing des Quellenwerts generieren. Wenn Hashing verwendet wird, können verschiedene Quellenwerte jedes Mal, wenn der Prozess ausgeführt wird, dieselben Zielwerte erzeugen.

Anmerkung:

Für denselben Quellenwert kann derselbe Zielwert abgerufen werden, wenn der Parameter 'unique' oder der Parameter 'hash' verwendet wird. Wenn Sie 'hash' mit dem Parameter 'seed' verwenden, erzeugen Sie bei jedem Ausführen des Prozesses verschiedene Zielwerte.

source=colname

Mit diesem Parameter können Sie den Namen der Quellenspalte angeben, wenn die Zielspalte nicht mit der Quellenspalte identisch ist. Der von Ihnen angegebene Wert wird in Großbuchstaben konvertiert; wenn Sie die Konvertierung des Werts in Großbuchstaben verhindern wollen, schließen Sie ihn in Anführungszeichen ein.

copy= Mindestens ein Paar Unterzeichenfolgen, die ohne Maskierung ans Ziel kopiert werden sollen. Wenn Sie eine Literalzeichenfolge angeben, werden die Quellenzeichen an den angegebenen Stellen ersetzt. Der Parameter *copy=* gilt nur für eine Spalte des Zeichendatentyps.

seed= Zum Ändern des Verhaltens der Maskierungsalgorithmen verwendeter Wert. Geben Sie eine Literalzeichenfolge, einen Verweis auf eine Umgebungsvariable oder RANDOM an.

"lit" Wenn Sie eine Literalzeichenfolge angeben wollen, schließen Sie diese in Anführungszeichen ein.

var (Variable)

Eine Umgebungsvariable in Klammern angeben. Der Variablenname und sein Wert dürfen keine Anführungszeichen enthalten.

RANDOM

Einen Zufallsseedwert aus dem aktuellen Systemdatum und der aktuellen Systemzeit generieren.

length={n | max }

Generieren Sie einen Zielwert mit einer Länge, die von der Quellenwertlänge abweicht. Mit *length=max* können Sie einen Zielwert generieren, der die Spalte komplett ausfüllt. Bei Angabe einer Länge, die kürzer als der Quellenwert ist, wird der Quellenwert abgeschnitten, wenn er in das Ziel geschrieben wird. Der Wert, den Sie für *n* auswählen, darf die definierte Länge der Zielspalte nicht überschreiten. Der Parameter *length=* ist nur mit *hash=* gültig.

preserve=

Mindestens einen Quellenwert auflisten, der am Ziel nicht ersetzt werden soll.

null Wenn die Quellenspalte einen Nullwert enthält, den Wert am Ziel nicht ersetzen.

spaces Wenn die Quellenspalte einen Leerzeichenwert enthält, den Wert am Ziel nicht ersetzen. Nur für CHAR-Spalten.

zero_len

Wenn die Quellenspalte einen VARCHAR-Wert mit Nulllänge hat, den Wert am Ziel nicht ersetzen.

TRIM=(char1 [charn,...])

Das angegebene/die angegebenen Quellenspaltenzeichen ist/sind nicht maskiert und wird/werden nicht in das Ziel geschrieben. Wenn Sie z. B. *TRIM=(x,y,z,1,2,3)* angeben und beliebige der Zeichen *x, y, z, 1, 2, 3* irgendwo in der Quellenspalte vorhanden sind, werden sie nicht maskiert oder in das Ziel geschrieben.

- [\u]* Mit diesem Parameter können Sie das/die Zeichen vor dem Maskieren in Großbuchstaben konvertieren. Wenn ein Zeichen keine Darstellung in Großbuchstaben hat, wird es nicht verändert. Bei Angabe von *TRIM=(x,y \u)* werden z. B. die Zeichen x und y nicht maskiert, wenn sie irgendwo in der Quellenspalte enthalten sind. Außerdem werden alle anderen Quellenspaltenzeichen in Großbuchstaben geändert, bevor sie maskiert werden.
- [\r]* Verwenden Sie diesen Operanden, wenn Sie nachfolgende Leerzeichen entfernen wollen. Bei Angabe von *TRIM=(x,y \u \r)* werden z. B. die Zeichen x und y nicht maskiert, wenn sie irgendwo in der Quellenspalte enthalten sind, alle anderen Quellenspaltenzeichen in Großbuchstaben geändert, bevor sie maskiert werden. Außerdem werden vor dem Maskieren alle nachfolgenden Leerzeichen entfernt.
- num* Mit diesem Parameter wird bewirkt, dass die Umsetzung ganzer Zahlen in einer Zeichendatentypspalte identisch mit der einer numerischen Datentypspalte ist. Der Parameter 'num' ist nur bei numerischen Werten in einer Zeichendatentypspalte gültig. Bei der Verwendung auf diese Art ist Fremdschlüsselintegrität über abweichende Datentypen hinweg gewährleistet. Wenn Sie diesen Parameter verwenden, geben Sie *copy=* oder *length=* nicht an.

Beispiele:

- TRANS COL UNIQUE für den Quellenwert **CDE-7834** könnte folgenden Zielwert erzeugen: **ZWQ-4598**
- TRANS COL UNIQUE SEED=RANDOM für den Quellenwert **CDE-7834-2008** könnte folgenden Zielwert erzeugen: **SWX-3162-8451**
- TRANS COL UNIQUE für den Quellenwert **Smith, John** könnte folgenden Zielwert erzeugen: **Fnxwq, Lrzp**
- TRANS COL UNIQUE für den Quellenwert **SMITH JOHN** könnte folgenden Zielwert erzeugen: **FNXWQ LRZP**
- TRANS COL UNIQUE (COPY=((1,3)(10,4))) für den Quellenwert **CDE-7834-2008** könnte folgenden Zielwert erzeugen: **CDE-4032-2008**
- TRANS COL HASH (LENGTH=13) für den Quellenwert **CDE-7834** könnte folgenden Zielwert erzeugen: **ZWQ-4598RN7A0**
- TRANS COL UNIQUE PRESERVE=(spaces) für den Quellenwert **XYZ 477 6835** könnte folgenden Zielwert erzeugen: **LMN 623 0972**
- TRANS COL HASH TRIM=(e \u) für den Quellenwert **InfoSphere** könnte folgenden Zielwert erzeugen: **RBIWACRL**

Funktion TRANS NID

Mit der Funktion TRANS NID können Sie länderspezifische ID-Nummern maskieren.

Sie können länderspezifische ID-Nummern entweder mit einer reproduzierbaren Methode maskieren, bei der ein Teil des Quellenwerts erhalten bleibt, oder mit einer Zufallsmethode, bei der kein Teil des Quellenwerts erhalten bleibt.

Sie können auch den Typ von Trennzeichen angeben, der in den Ausgabewerten verwendet werden soll (Gedankenstriche, Punkte, Leerzeichen oder keine Trennzeichen).

Die Funktion TRANS NID verwendet folgende Syntax:

```
[TRANS] NID ('SWI=country_code, [FMT=(output_format)], [MTD={mask|random}],
            [SRC=column_name], [VAL={Y|N}], [PRE=INV]')
```

SWITCH oder *SWI*

Ein aus zwei Zeichen bestehender Wert, der den Typ der zu maskierenden länderspezifischen ID angibt. Dieser Parameter ist erforderlich. Nur ein einziger Switchwert ist zulässig.

Folgende aus zwei Zeichen bestehende Werte sind gültig:

Kanada: Kanadische Sozialversicherungsnummer (SIN)

CA

Frankreich: Nummer des französischen nationalen Instituts für Statistik und Wirtschaftsstudien (INSEE)

FR

Italien: Italienische Steuernummer (CF)

IT

Spanien: Spanische Steuernummer (NIF)

ES

Großbritannien: Britische Sozialversicherungsnummer (NINO)

UK

Vereinigte Staaten von Amerika: US-Sozialversicherungsnummern (SSN)

US

FMT oder *FORMAT*

Legt das Ausgabeformat und die Teile des Quellenwerts, die maskiert werden sollen, fest. Die Syntax für diesen Parameter wird durch die länderspezifische ID festgelegt, die maskiert wird:

- Kanadische Sozialversicherungsnummer (SIN)
- Nummer des französischen nationalen Instituts für Statistik und Wirtschaftsstudien (INSEE)
- Italienische Steuernummer (CF)
- Spanische Steuernummer (NIF)/Steuernummer für Ausländer (NIE)
- Britische Sozialversicherungsnummer (NINO)
- US-Sozialversicherungsnummer (SSN)

Wenn dieser Parameter nicht angegeben wird und *MTD=mask* zutrifft, wird das Quellenformat verwendet und die Standardfelder werden maskiert.

Wenn dieser Parameter nicht angegeben wird und *MTD=random* zutrifft, enthalten die Ausgabewerte keine Trennzeichen.

Wenn die Zielspalte nicht groß genug für ein Ausgabeformat ist, das Trennzeichen verwendet, werden keine Trennzeichen mit aufgenommen.

MTD oder *METHOD*

Gibt an, welche Maskierungsmethode verwendet werden soll (reproduzierbar oder Zufallsverfahren). Wenn dieser Parameter nicht angegeben wird, wird eine reproduzierbare Maskierung ausgeführt (*MTD=mask*).

mask Die Quellenwerte werden reproduzierbar maskiert. Die Ausgabewerte basieren auf den Quellenwerten. Standardeinstellung.

random

Die Ausgabewerte werden von einem Zufallsmaskierungsalgorithmus generiert. Die Ausgabewerte basieren nicht auf den Quellenwerten.

Anmerkung:

- *MTD=random* ist nicht mit folgenden Parametern kompatibel: *VAL=Y* und *PRE=INV*.
- Wenn *MTD=random* zutrifft und der Parameter *FMT* angibt, dass ein Teil des Quellenwerts in den Ausgabewert kopiert werden soll, enthält der Ausgabewert keine Quellenwerte; allerdings werden alle Trennzeichen, die im Parameter *FMT* angegeben sind, in den Ausgabewert aufgenommen.

SRC oder *SOURCE*

Der Name der Spalte, die die Quellenwerte enthält. Verwenden Sie diese Parameter nur, wenn die Namen der Quellen- und Zielspalten nicht übereinstimmen.

VAL oder **VALIDATE**

Gibt an, ob eine landesspezifische Überprüfung der Quellenwerte ausgeführt werden soll. Wenn dieser Parameter nicht angegeben wird, wird keine Überprüfung ausgeführt (*VAL=N*).

Y Die Quellenwerte überprüfen.

Anmerkung: Die Option kann nicht verwendet werden, wenn *MTD=random* zutrifft.

N Die Quellenwerte nicht überprüfen. Standardeinstellung.

PRE oder **PRESERVE**

Legt fest, ob ungültige Quellenwerte in die Zielspalte kopiert werden. Wenn dieser Parameter nicht angegeben wird, werden ungültige Quellenwerte nicht in die Zielspalte kopiert und Zeilen, die diese Werte enthalten, werden übersprungen. Dieser Parameter enthält nur eine Option, *INV* oder *INVALID*.

INV oder **INVALID**

Ungültige Quellenwerte in die Zielspalte kopieren.

Anmerkung: Die Option kann nicht verwendet werden, wenn *MTD=random* zutrifft.

Beispiele

Durch die folgende Syntax werden US-Sozialversicherungsnummern (SSN) mit einer reproduzierbaren Methode maskiert. Die Funktion kopiert die ersten drei Ziffern der Quellen-SSN und fügt einen Gedankenstrich als Trennzeichen für den Ausgabewert ein. Es wird eine Überprüfung für den Ausgabewert ausgeführt.

```
NID('SWITCH=US, FMT=(US=3C-2X-4X), MTD=MASK, VAL=Y')
```

Durch die folgende Syntax werden Nummern des französischen nationalen Instituts für Statistik und Wirtschaftsstudien (INSEE) mit einer Zufallsmethode maskiert. Die Funktion verwendet das Standardausgabeformat, das keine Trennzeichen enthält.

```
NID('SWITCH=FR, MTD=RANDOM')
```

Durch die folgende Syntax werden spanische Steuernummern (NIF) mit einer reproduzierbaren Methode maskiert. Die Funktion behält ungültige Quellenwerte bei und verwendet das Standardausgabeformat.

```
NID('SWITCH=ES, MTD=MASK, PRE=INV')
```

Kanadische Sozialversicherungsnummern maskieren

Mit der Funktion **TRANS NID** können Sie kanadische Sozialversicherungsnummern (SIN) maskieren.

Eine SIN ist eine 9-stellige Nummer, die aus einem einstelligen Regionscode gefolgt von einer 8-stelligen Seriennummer besteht. Die ersten drei Stellen werden als Header bezeichnet. Die letzte Ziffer der Seriennummer ist eine Prüfziffer.

Die Funktion **TRANS NID** generiert eine maskierte Sozialversicherungsnummer mit einer Prüfziffer, die auf der Basis der vorherigen maskierten acht Ziffern des Ausgabewerts errechnet wird.

Ausgabeformate (FMT=)

Die folgenden Ausgabeformate sind für eine Sozialversicherungsnummer (SIN) verfügbar.

C gibt Werte an, die kopiert werden. *X* gibt Werte an, die maskiert werden. *3C4X* gibt zum Beispiel an, dass die ersten drei Zeichen kopiert und die nächsten vier Zeichen maskiert werden.

Zu maskierende Felder	Format ohne Trennzeichen	Format mit Gedankenstrich als Trennzeichen	Format mit Leerzeichen als Trennzeichen	Format mit Punkt als Trennzeichen
Seriennummer ohne Headerziffern (Standardwert <i>MTD=mask</i>)	CA=3C6X	CA=3C-3X-3X	CA=3C 3X 3X	CA=3C.3X.3X
Seriennummer und Headerziffern	CA=9X	CA=3X-3X-3X	CA=3X 3X 3X	CA=3X.3X.3X

Datentypen

Folgende Datentypen sind für Quellen- und Zielspalten zulässig. Einschränkungen für die einzelnen Datentypen sind angegeben.

CHAR und NCHAR

Die Länge der Spalte muss mindestens 9 Zeichen aufweisen.

DECIMAL

Die Genauigkeit der Spalte muss zwischen 9 und 20 liegen und die Anzahl der Kommastellen muss 0 sein.

INTEGER

Keine Einschränkungen.

VARCHAR und NVARCHAR

Die Länge der Spalte muss mindestens 9 Zeichen aufweisen.

Überprüfung (VAL=Y)

Mit dem Parameter (VAL=Y) wird eine Quellenzeile übersprungen, wenn eine der folgenden Bedingungen zutrifft:

- Die erste Ziffer ist 8.
- In den Positionen 1-3, 4-6 oder 7-9 gibt es drei aufeinanderfolgende Nullen.
- Die Prüfziffer ist ungültig.

Sonderverarbeitung

Die folgenden Prüfungen werden während der Verarbeitung vorgenommen:

- Wenn bei einem Quellenwert von 0 (Datentyp INTEGER oder DECIMAL), Leerzeichen oder entweder einem VARCHAR- oder NVARCHAR-Zeichen mit Nulllänge
 - die Zielspalte den Datentyp INTEGER oder DECIMAL aufweist, ist der Ausgabewert 0.
 - die Zielspalte den Datentyp CHAR oder NCHAR aufweist, werden als Ausgabewert Leerzeichen ausgegeben.
 - die Zielspalte den Datentyp VARCHAR oder NVARCHAR aufweist, ist die Ziellänge 0.
- Wenn die Länge einer Zielspalte 11 Zeichen oder mehr aufweist und Trennzeichen für den Ausgabewert angegeben wurden, werden die Trennzeichen mit eingeschlossen, falls die Zielspalte einen Datentyp CHAR, NCHAR, VARCHAR oder NVARCHAR aufweist.
- Wenn der Quellenwert NULL ist, ist auch der Ausgabewert NULL.

Übersprungene Zeilen

Eine Quellenzeile wird übersprungen und nicht in die Zieltabelle geschrieben, wenn Folgendes zutrifft:

Nullwert

Der Quellenwert ist NULL, die Zielspalte lässt jedoch keinen Wert NULL zu.

Ungültige SIN

- Die Quellenspalte ist eine CHAR-, NCHAR-, VARCHAR- oder NVARCHAR-Spalte, der Quellenwert hat jedoch weniger als 9 Zeichen (Trennzeichen nicht eingeschlossen).
- Die Quellenspalte ist eine CHAR-, NCHAR-, VARCHAR- oder NVARCHAR-Spalte, der Quellenwert hat jedoch mehr als 9 Zeichen (Trennzeichen nicht eingeschlossen).
- Der Quellenwert enthält ein nicht numerisches Zeichen.

Nummern des französischen nationalen Instituts für Statistik und Wirtschaftsstudien (INSEE) maskieren

Mit der Funktion TRANS NID können Sie Nummern des französischen nationalen Instituts für Statistik und Wirtschaftsstudien (INSEE) maskieren.

Eine INSEE-Nummer ist eine 15-stellige Nummer mit folgendem Format: *SYMMDDCCCOOKK*.

S Informationen zu Geschlecht und Staatsangehörigkeit.

YY Die letzten beiden Ziffern des Geburtsjahrs.

MM Geburtsmonat.

DD Herkunftsdepartement.

CCC Herkunftskommune.

OOO Auftragsnummer.

KK Steuerungsschlüssel oder Prüfziffern.

Die Funktion TRANS NID generiert eine maskierte INSEE-Nummer gemäß folgenden Regeln:

- Wenn das Départementfeld maskiert ist, ist das Kommunenfeld ebenfalls mit einem kompatiblen Wert maskiert.
- Das Auftragsfeld ist immer maskiert.
- Das Prüfziffernfeld wird auf der Basis der vorhergehenden maskierten 13 Ziffern des Ausgabewerts errechnet.

Ausgabeformate (FMT=)

Die folgenden Ausgabeformate sind für eine INSEE-Nummer verfügbar.

Alle Formate maskieren die Auftrags- und Prüfziffernfelder. Wenn das Départementfeld maskiert ist, wird das Kommunenfeld ebenfalls mit einem kompatiblen Wert maskiert.

C gibt Werte an, die kopiert werden. X gibt Werte an, die maskiert werden. 3C4X gibt zum Beispiel an, dass die ersten drei Zeichen kopiert und die nächsten vier Zeichen maskiert werden.

Zu maskierende Felder (zusätzlich zu Auftrag und Prüfziffer)	Format ohne Trennzeichen	Format mit Gedankenstrich als Trennzeichen	Format mit Leerzeichen als Trennzeichen
Geschlecht, Jahr, Monat, Kommune (Standardwert <i>MTD=mask</i>)	FR=5X2C8X	FR=5X2C6X-2X	FR=5X2C6X 2X
Geschlecht	FR=1X9C5X	FR=1X9C3X-2X	FR=1X9C3X 2X
Geschlecht, Jahr	FR=3X7C5X	FR=3X7C3X-2X	FR=3X7C3X 2X
Geschlecht, Monat	FR=1X2C2X5C5X	FR=1X2C2X5C3X-2X	FR=1X2C2X5C3X 2X
Geschlecht, Kommune	FR=1X6C8X	FR=1X6C6X-2X	FR=1X6C6X 2X
Geschlecht, Département	FR=1X4C8X	FR=1X4C6X-2X	FR=1X4C6X 2X

Zu maskierende Felder (zusätzlich zu Auftrag und Prüfziffer)	Format ohne Trennzeichen	Format mit Gedanken- strich als Trennzeichen	Format mit Leerzeichen als Trennzeichen
Geschlecht, Jahr, Monat	FR=5X5C5X	FR=5X5C3X-2X	FR=5X5C3X 2X
Geschlecht, Jahr, Kommune	FR=3X4C8X	FR=3X4C6X-2X	FR=3X4C6X 2X
Geschlecht, Jahr, Département, Kommune	FR=3X2C10X	FR=3X2C8X-2X	FR=3X2C8X 2X
Geschlecht, Monat, Kom- mune	FR=1X2C2X2C8X	FR=1X2C2X2C6X-2X	FR=1X2C2X2C6X 2X
Geschlecht, Monat, Département, Kommune	FR=1X2C12X	FR=1X2C10X-2X	FR=1X2C10X 2X
Geschlecht, Jahr, Monat, Département, Kommune	FR=15X	FR=13X-2X 13X	FR=13X 2X
Jahr	FR=1C2X7C5X	FR=1C2X7C3X-2X	FR=1C2X7C3X 2X
Jahr, Monat	FR=1C4X5C5X	FR=1C4X5C3X-2X	FR=1C4X5C3X 2X
Jahr, Kommune	FR=1C2X4C8X	FR=1C2X4C6X-2X	FR=1C2X4C6X 2X
Jahr, Département	FR=1C2X2C10X	FR=1C2X2C8X-2X	FR=1C2X2C8X 2X
Jahr, Monat, Kommune	FR=1C4X2C8X	FR=1C4X2C6X-2X	FR=1C4X2C6X 2X
Jahr, Monat, Département	FR=1C14X	FR=1C12X-2X	FR=1C12X 2X
Monat	FR=3C2X5C5X	FR=3C2X5C3X-2X	FR=3C2X5C3X 2X
Monat, Kommune	FR=3C2X2C8X	FR=3C2X2C6X-2X	FR=3C2X2C6X 2X
Monat, Département	FR=3C12X	FR=3C10X-2X	FR=3C10X 2X
Kommune	FR=7C8X	FR=7C6X-2X	FR=7C6X 2X
Département	FR=5C10X	FR=5C8X-2X	FR=5C8X 2X

Datentypen

Folgende Datentypen sind für Quellen- und Zielspalten zulässig. Einschränkungen für die einzelnen Datentypen sind angegeben.

CHAR und NCHAR

Die Länge der Spalte muss mindestens 15 Zeichen aufweisen.

VARCHAR und NVARCHAR

Die Länge der Spalte muss mindestens 15 Zeichen aufweisen.

Überprüfung (VAL=Y)

Mit dem Parameter (VAL=Y) wird eine Quellenzeile übersprungen, wenn eine der folgenden Bedingungen zutrifft:

- Der Wert für das Quellenkommunenfeld ist ungültig.
- Das Feld für die Quellenprüfziffer ist ungültig.

Sonderverarbeitung

Die folgenden Prüfungen werden während der Verarbeitung vorgenommen:

- Wenn bei einem Quellenwert von Leerzeichen (SPACES) oder entweder einem VARCHAR- oder NVARCHAR-Zeichen mit Nulllänge
 - die Zielspalte den Datentyp CHAR oder NCHAR aufweist, werden als Ausgabewert Leerzeichen ausgegeben.

- die Zielspalte den Datentyp VARCHAR oder NVARCHAR aufweist, ist die Ziellänge 0.
- Wenn der Quellenwert NULL ist, ist auch der Ausgabewert NULL.
- Wenn die Länge einer Zielspalte 16 Zeichen oder mehr aufweist und ein Trennzeichen für den Ausgabewert angegeben wurde, wird das Trennzeichen mit eingeschlossen.

Übersprungene Zeilen

Eine Quellenzeile wird übersprungen und nicht in die Zieltabelle geschrieben, wenn der Quellenwert folgende Fehler aufweist:

Nullwert

Der Quellenwert ist NULL, die Zielspalte lässt jedoch keinen Wert NULL zu.

Ungültige INSEE-Nummer

- Der INSEE-Eingabewert ist zu groß oder zu klein.
- Der INSEE-Eingabewert enthält Trennzeichen, die ungültig sind oder sich an der falschen Stelle befinden.
- Als Geschlecht ist im betreffenden Feld nicht einer der folgenden Werte angegeben: 1, 2, 7 oder 8.

Italienische Steuernummern maskieren

Sie können mit der Funktion TRANS NID italienische Steuernummern (CF) maskieren.

Eine italienische Steuernummer ist ein aus 16 Zeichen bestehender alphanumerischer Wert mit folgendem Format: *FFF-NNN-YYMDD-RRRRC*.

FFF Codierter Nachname.

NNN Codierter Vorname.

YY Geburtsjahr.

M Geburtsmonat.

DD Tag der Geburt.

RRRR Regionscode.

C Steuerzeichen.

Die Funktion TRANS NID generierte eine maskierte Steuernummer gemäß folgenden Regeln:

- Jeder Konsonant, der in den angegebenen Feldern für Vorname oder Nachname erscheint, wird als Konsonant maskiert, und jeder Vokal wird als Vokal maskiert. Wenn ein X nach einem Vokal erscheint, muss es in den Ausgabewert kopiert werden.
- Das Steuerzeichenfeld wird auf der Basis der vorhergehenden maskierten 15 Ziffern des Ausgabewerts errechnet.

Ausgabeformate (FMT=)

Folgende Ausgabeformate sind für eine italienische Steuernummer verfügbar.

C gibt Werte an, die kopiert werden. X gibt Werte an, die maskiert werden. 3C4X gibt zum Beispiel an, dass die ersten drei Zeichen kopiert und die nächsten vier Zeichen maskiert werden.

Zu maskierende Felder	Format ohne Trennzeichen	Format mit Gedankenstrich als Trennzeichen	Format mit Leerzeichen als Trennzeichen
Geburtsdatum, Region (Standardwert <i>MTD=mask</i>)	IT=6C10X	IT=3C-3C-5X-5X	IT=3C 3C 5X 5X

Zu maskierende Felder	Format ohne Trennzeichen	Format mit Gedankenstrich als Trennzeichen	Format mit Leerzeichen als Trennzeichen
Nachname, Vorname, Region	IT=6X5C5X	IT=3X-3X-5C-5X	IT=3X 3X 5C 5X
Nachname, Vorname, Geburtsdatum	IT=11X4C1X	IT=3X-3X-5X-4C1X	IT=3X 3X 5X 4C1X
Nachname, Vorname	IT=6X9C1X	IT=3X-3X-5C-4C1X	IT=3X 3X 5C 4C1X
Geburtsdatum	IT=6C5X4C1X	IT=3C-3C-5X-4C1X	IT=3C 3C 5X 4C1X
Region	IT=11C5X	IT=3C-3C-5C-5X	IT=3C 3C 5C 5X
Nachname, Vorname, Geburtsdatum, Region	IT=16X	IT=3X-3X-5X-5X	IT=3X 3X 5X 5X

Datentypen

Folgende Datentypen sind für Quellen- und Zielspalten zulässig. Einschränkungen für die einzelnen Datentypen sind angegeben.

CHAR und NCHAR

Die Länge der Spalte muss mindestens 16 Zeichen aufweisen.

VARCHAR und NVARCHAR

Die Länge der Spalte muss mindestens 16 Zeichen aufweisen.

Überprüfung (VAL=Y)

Mit dem Parameter (VAL=Y) wird eine Quellenzeile übersprungen, wenn sie ein ungültiges Steuerzeichen enthält.

Sonderverarbeitung

Die folgenden Prüfungen werden während der Verarbeitung vorgenommen:

- Wenn bei einem Quellenwert von Leerzeichen (SPACES) oder entweder einem VARCHAR- oder NVARCHAR-Zeichen mit Nulllänge
 - die Zielspalte den Datentyp CHAR oder NCHAR aufweist, werden als Ausgabewert Leerzeichen ausgegeben.
 - die Zielspalte den Datentyp VARCHAR oder NVARCHAR aufweist, ist die Ziellänge 0.
- Wenn der Quellenwert NULL ist, ist auch der Ausgabewert NULL.
- Wenn die Länge einer Zielspalte 19 Zeichen oder mehr aufweist und Trennzeichen für den Ausgabewert angegeben wurden, werden die Trennzeichen mit eingeschlossen.

Übersprungene Zeilen

Eine Quellenzeile wird übersprungen und nicht in die Zieltabelle geschrieben, wenn Folgendes zutrifft:

Nullwert

Der Quellenwert ist NULL, die Zielspalte lässt jedoch keinen Wert NULL zu.

Ungültige Steuernummer

In der Quellenspalte befindet sich ein ungültiger Quellenwert oder der Quellenwert ist zu groß.

Der Quellenwert ist kürzer als 16 Zeichen.

Spanische Steuernummern und Steuernummern für Ausländer maskieren

Mit der Funktion TRANS NID können Sie spanische Steuernummern (NIF) und Steuernummern für Ausländer (NIE) maskieren.

Eine spanische Steuernummer ist ein aus acht Zeichen bestehender Wert im folgenden Format: NNNNNNN-A. Hierbei stellen die ersten sieben Zeichen eine Seriennummer dar und das letzte Zeichen ist ein alphabetisches Suffix. Das Suffix ist eine Prüfziffer.

Ausländische Staatsangehörige in Spanien verwenden eine Steuernummer für Ausländer (NIE), die ein aus neun Zeichen bestehender Wert ist, der dasselbe Format wie eine Steuernummer (NIF) verwendet, dem aber 'X' vorangeht. Ein NIE-Wert verwendet folgendes Format: X-NNNNNNN-A.

Die Funktion TRANS NID generiert eine maskierte Steuernummer oder Steuernummer für Ausländer mit einer Prüfziffer, die auf der Basis der vorherigen maskierten sieben Ziffern des Ausgabewerts errechnet wird.

Ausgabeformate (FMT=)

Für spanische Steuernummern oder Steuernummern für Ausländer sind folgende Ausgabeformate verfügbar.

Für jedes Format sind alle Zeichen maskiert. Spanische Steuernummern und Steuernummern für Ausländer verwenden dieselben Formatoptionen. Ein Quellenwert für eine Steuernummer für Ausländer enthält immer das Präfix X im Ausgabewert.

Zu maskierende Felder	Format ohne Trennzeichen	Format mit Gedankenstrich als Trennzeichen	Format mit Leerzeichen als Trennzeichen
Serienwert, Suffix (Standardwert <i>MTD=mask</i>)	ES=8X	ES=7X-1X	ES=7X 1X

Datentypen

Folgende Datentypen sind für Quellen- und Zielspalten zulässig. Einschränkungen für die einzelnen Datentypen sind angegeben.

CHAR und NCHAR

Die Länge der Spalte muss mindestens 8 Zeichen aufweisen.

VARCHAR und NVARCHAR

Die Länge der Spalte muss mindestens 8 Zeichen aufweisen.

Überprüfung (VAL=Y)

Mit dem Parameter (VAL=Y) wird eine Quellenzeile übersprungen, wenn sie ein ungültiges Suffix enthält.

Sonderverarbeitung

Die folgenden Prüfungen werden während der Verarbeitung vorgenommen:

- Wenn bei einem Quellenwert von Leerzeichen (SPACES) oder entweder einem VARCHAR- oder NVARCHAR-Zeichen mit Nulllänge
 - die Zielspalte den Datentyp CHAR oder NCHAR aufweist, werden als Ausgabewert Leerzeichen ausgegeben.
 - die Zielspalte den Datentyp VARCHAR oder NVARCHAR aufweist, ist die Ziellänge 0.
- Wenn der Quellenwert NULL ist, ist auch der Ausgabewert NULL.

- Wenn die Länge einer Zielspalte mindestens 11 Zeichen aufweist und ein Trennzeichen für den Ausgabewert angegeben wurde, wird das Trennzeichen mit eingeschlossen.

Übersprungene Zeilen

Eine Quellenzeile wird übersprungen und nicht in die Zieltabelle geschrieben, wenn Folgendes zutrifft:

Ungültige Länge

Der Eingabewert enthält keine Trennzeichen und ist kürzer als 8 Zeichen (MTD=NIF) oder kürzer als 9 Zeichen (NIE).

Der Eingabewert enthält Trennzeichen und ist kürzer als 9 Zeichen (NIF) oder kürzer als 11 Zeichen (NIE).

Ungültiges Muster

Die Länge des Quellenwerts ist gültig, aber der Wert stimmt nicht mit dem Muster einer Steuernummer oder Steuernummer für Ausländer überein.

Trennzeichen

Es wurden verschiedene Trennzeichen in den Positionen 2 und 10 in einem Quellenwert für eine Steuernummer für Ausländer gefunden.

Der Quellenwert enthält ein ungültiges Trennzeichen.

Nullwert

Der Quellenwert ist NULL, die Zielspalte lässt jedoch keinen Wert NULL zu.

Britische Sozialversicherungsnummern maskieren

Mit der Funktion TRANS NID können Sie britische Sozialversicherungsnummern (NINO) maskieren.

Eine britische Sozialversicherungsnummer (NINO) besteht aus drei Teilen: zwei Buchstaben (Präfix), sechs Ziffern (Nummer) und einem optionalen Buchstaben (Suffix).

Ausgabeformate (FMT=)

Die folgenden Ausgabeformate sind für eine britische Sozialversicherungsnummer verfügbar.

Eine britische Sozialversicherungsnummer kann ohne ein Trennzeichen oder mit einem Trennzeichen in einem dreiteiligen oder fünfteiligen Format maskiert werden.

C gibt Werte an, die kopiert werden. X gibt Werte an, die maskiert werden. 3C4X gibt zum Beispiel an, dass die ersten drei Zeichen kopiert und die nächsten vier Zeichen maskiert werden.

Verwenden Sie folgende Parameter, um eine britische Sozialversicherungsnummer ohne Trennzeichen zu erstellen:

Zu maskierende Felder	Format ohne Trennzeichen
Präfix, Nummer	UK=8X1C
Nummer (Standardwert MTD=Mask)	UK=2C6X1C
Präfix, Nummer, Suffix (nur MTD=random)	UK=9X

Wenn Sie eine britische Sozialversicherungsnummer in einem drei- oder fünfteiligen Format erstellen wollen, verwenden Sie folgende Parameter:

Zu maskierende Felder	Format mit Gedankenstrich als Trennzeichen	Format mit Leerzeichen als Trennzeichen	Format mit Punkt als Trennzeichen
Präfix, Nummer (dreiteilig)	UK=2X-6X-1C	UK=2X 6X 1C	UK=2X.6X.1C

Zu maskierende Felder	Format mit Gedankenstrich als Trennzeichen	Format mit Leerzeichen als Trennzeichen	Format mit Punkt als Trennzeichen
Präfix, Nummer (fünfteilig)	UK=2X-2X-2X-2X-1C	UK=2X 2X 2X 2X 1C	UK=2X.2X.2X.2X.1C
Nummer (dreiteilig)	UK=2C-6X-1C	UK=2C 6X 1C	UK=2C.6X.1C
Nummer (fünfteilig)	UK=2C-2X-2X-2X-1C	UK=2C 2X 2X 2X 1C	UK=2C.2X.2X.2X.1C
Präfix, Nummer, Suffix (dreiteilig) (nur <i>MTD=random</i>)	UK=2X-6X-1X	UK=2X 6X 1X	UK=2X.6X.1X
Präfix, Nummer, Suffix (fünfteilig) (nur <i>MTD=random</i>)	UK=2X-2X-2X-2X-1X	UK=2X 2X 2X 2X 1X	UK=2X.2X.2X.2X.1X

Datentypen

Folgende Datentypen sind für Quellen- und Zielspalten zulässig. Einschränkungen für die einzelnen Datentypen sind angegeben.

CHAR und NCHAR

Die Länge der Spalte muss mindestens 9 Zeichen aufweisen.

VARCHAR und NVARCHAR

Die Länge der Spalte muss mindestens 9 Zeichen aufweisen.

Überprüfung (*VAL=Y*)

Der Parameter *VAL=Y* ist für das Maskieren britischer Sozialversicherungsnummern nicht gültig und wird ignoriert.

Sonderverarbeitung

Die folgenden Prüfungen werden während der Verarbeitung vorgenommen:

- Wenn bei einem Quellenwert von Leerzeichen (SPACES) oder entweder einem VARCHAR- oder NVARCHAR-Zeichen mit Nulllänge
 - die Zielspalte den Datentyp CHAR oder NCHAR aufweist, werden als Ausgabewert Leerzeichen ausgegeben.
 - die Zielspalte den Datentyp VARCHAR oder NVARCHAR aufweist, ist die Ziellänge 0.
- Wenn der Quellenwert NULL ist, ist auch der Ausgabewert NULL.
- Wenn die Zielspalte kein Trennzeichen aufnehmen kann, enthält der Ausgabewert kein Trennzeichen.

Übersprungene Zeilen

Eine Quellenzeile wird übersprungen und nicht in die Zieltabelle geschrieben, wenn Folgendes zutrifft:

Nullwert

Der Quellenwert ist NULL, die Zielspalte lässt jedoch keinen Wert NULL zu.

Ungültige britische Sozialversicherungsnummer

- Die Größe des Quellenwerts ist kleiner oder größer als die Größe einer gültigen britischen Sozialversicherungsnummer.
- Der Quellenwert enthält Trennzeichen an den falschen Stellen.
- Der Quellenwert enthält ein ungültiges Trennzeichen.
- Der Quellenwert enthält ein ungültiges Präfix.
- Der Quellenwert enthält ein anderes Suffix als A, B, C oder D.

- Der Quellenwert enthält ein Zahlenfeld, dessen Wert nicht zwischen 000001 und 999999 liegt.

US-amerikanische Sozialversicherungsnummern maskieren

Mit der Funktion TRANS NID können Sie Sozialversicherungsnummern (SSN-Nummern) der Vereinigten Staaten maskieren.

Eine SSN-Nummer besteht aus drei Unterfeldern mit folgendem Format: AAAGSSSS.

- AAA** Bereichsnummer. Der Bereich wird in der Regel durch den Staat festgelegt, in dem die SSN-Nummer ausgegeben wird.
- GG** Gruppennummer. Eine Gruppennummer wird auf der Basis der Bereichsnummer zugeordnet.
- SSSS** Seriennummer.

Die Funktion TRANS NID generiert eine maskierte SSN-Nummer gemäß folgenden Regeln:

- Es wird eine Gruppennummer generiert, die für die Bereichsnummer geeignet ist. Die Gruppennummer ist die neueste Gruppe, die von der Social Security Administration für den Bereich verwendet wurde.
- Seriennummern beginnen mit 0001 und werden für jede weitere Sozialversicherungsnummer, die für die Bereichsnummer generiert wird, jeweils um 1 erhöht. Wenn die Seriennummer 9999 überschreitet, wird sie auf 0001 zurückgesetzt und die Gruppennummer vor der Nummer, die zuletzt für die Bereichsnummer ausgegeben wurde, wird verwendet.
- Wenn *MTD=mask* zutrifft, enthält der Ausgabewert eine Bereichsnummer, die demselben Staat entspricht wie die Quellenbereichsnummer.

Ausgabeformate (FMT=)

Folgende Ausgabeformate sind für eine SSN-Nummer verfügbar.

C gibt Werte an, die kopiert werden. X gibt Werte an, die maskiert werden. 3C4X gibt zum Beispiel an, dass die ersten drei Zeichen kopiert und die nächsten vier Zeichen maskiert werden.

Zu maskierende Felder	Format ohne Trennzeichen	Format mit Gedankenstrich als Trennzeichen	Format mit Leerzeichen als Trennzeichen	Format mit Punkt als Trennzeichen
Gruppe, Seriennummer (Standardwert <i>MTD=mask</i>)	US=3C6X	US=3C-2X-4X	US=3C 2X 4X	US=3C.2X.4X
Bereich, Gruppe, Seriennummer	US=9X	US=3X-2X-4X	US=3X 2X 4X	US=3X.2X.4X

Datentypen

Folgende Datentypen sind für Quellen- und Zielspalten zulässig. Einschränkungen für die einzelnen Datentypen sind angegeben.

CHAR und NCHAR

Die Länge der Spalte muss mindestens 9 Zeichen aufweisen.

DECIMAL

Die Genauigkeit der Spalte muss zwischen 9 und 20 liegen und die Anzahl der Kommastellen muss 0 sein.

INTEGER

Keine Einschränkungen.

VARCHAR und NVARCHAR

Die Länge der Spalte muss mindestens 9 Zeichen aufweisen.

Überprüfung (VAL=Y)

Mit dem Parameter (VAL=Y) wird eine Quellenzeile übersprungen, wenn eine der folgenden Bedingungen zutrifft:

- Die Quellenbereichsnummer überschreitet den Maximalwert.
- Die Quellenbereichsnummer wurde nicht von der Social Security Administration (SSA) verwendet.
- Die Quellengruppennummer wurde nicht für die Quellenbereichsnummer verwendet.

Sonderverarbeitung

Die folgenden Prüfungen werden während der Verarbeitung vorgenommen:

- Wenn bei einem Quellenwert von 0 (Datentyp INTEGER oder DECIMAL), Leerzeichen oder entweder einem VARCHAR- oder NVARCHAR-Zeichen mit Nulllänge
 - die Zielspalte den Datentyp INTEGER oder DECIMAL aufweist, ist der Ausgabewert 0.
 - die Zielspalte den Datentyp CHAR oder NCHAR aufweist, werden als Ausgabewert Leerzeichen ausgegeben.
 - die Zielspalte den Datentyp VARCHAR oder NVARCHAR aufweist, ist die Ziellänge 0.
- Wenn die Länge einer Zielspalte 11 Zeichen oder mehr aufweist und Trennzeichen für den Ausgabewert angegeben wurden, werden die Trennzeichen mit eingeschlossen.
- Wenn der Quellenwert NULL ist, ist auch der Ausgabewert NULL.

Übersprungene Zeilen

Eine Quellenzeile wird übersprungen und nicht in die Zieltabelle geschrieben, wenn Folgendes zutrifft:

Nullwert

Der Quellenwert ist NULL, die Zielspalte lässt jedoch keinen Wert NULL zu.

Ungültige SSN-Nummer

- Die Quellenspalte ist eine CHAR-, NCHAR-, VARCHAR- oder NVARCHAR-Spalte, der Quellenwert hat jedoch nicht 9 Zeichen (Trennzeichen nicht eingeschlossen).
- Der Quellenwert enthält ein nicht numerisches Zeichen.
- Der Quellenwert ist 0 oder ein reservierter Wert wie 078-05-1120 und 457-55-5462.

Unterzeichenfolgefunktion

Die Unterzeichenfolgefunktion gibt eine Unterzeichenfolge des Inhalts der angegebenen Spalte zurück.

`SUBSTR(columnname, start, [length])`

columnname

Name einer Zeichen- oder Binärspalte.

start Die Position des ersten Zeichens in der Zeichenfolge.

length Die zu verwendende Anzahl von Zeichen.

- Wenn die Ländereinstellung ein Komma als Dezimaltrennzeichen verwendet, müssen Sie nach jedem Komma ein Leerzeichen lassen, das numerische Parameter voneinander trennt (zum Beispiel nach dem Komma zwischen *start* und *length*).
- *start* und *length* sind Ganzzahlen größer-gleich 1.
- *start* plus *length* dürfen die Gesamtdatenlänge plus 1 nicht überschreiten.

- *column-name* und *start* sind erforderliche Werte. Wenn Sie nur eine ganze Zahl angeben, wird sie als *start*-Wert verwendet. Die Unterzeichenfolge beginnt bei *start* und enthält den Rest des Spaltenwerts.

Beispiel

Wenn die Spalte PHONE_NUMBER als CHAR(10) definiert ist, können Sie über die Unterzeichenfolgefunktion die Vorwahl ermitteln. Wenn Sie eine Unterzeichenfolge mit den ersten drei Stellen der Telefonnummer (US-amerikanische Vorwahl) für die Zielspalte abrufen wollen, geben Sie Folgendes an:

```
SUBSTR(PHONE_NUMBER, 1, 3)
```

Zufallsfunktion

Die Zufallsfunktion gibt eine Zufallszahl zurück, die innerhalb des durch die Werte für 'low' und 'high' angegebenen Bereichs ausgewählt wird.

```
RAND(low, high)
```

low Der niedrigste mögliche Zufallswert.

high Der höchste mögliche Zufallswert.

- Sie können die Zufallsfunktion mit Zeichendaten oder numerischen Daten verwenden.
- Wenn die Ländereinstellung ein Komma als Dezimaltrennzeichen verwendet, müssen Sie nach dem Komma ein Leerzeichen lassen.
- *low* und *high* sind ganze Zahlen im Bereich von **-2.147.483.648** bis **2.147.483.647**.
- *low* und *high* werden durch den Datentyp und die Länge für die Zielspalte zusätzlich begrenzt.
- *low* muss kleiner als *high* sein.
- Wenn Sie die Zufallsfunktion in einem verketteten Ausdruck verwenden, wird eine Zeichenfolge variabler Länge zurückgegeben.

Beispiel

Sie können mit der Zufallsfunktion Verkaufsdaten für eine Testdatenbank maskieren oder ändern. Angenommen, die Spalte YTD_SALES ist als DECIMAL(7,2) definiert. Die maximale Anzahl von Ziffern vor dem Dezimalzeichen ist 5; der mögliche Bereich für diese Spalte ist -99999 bis 99999. Wenn Sie Testdaten in einem Bereich von 1000 (niedrig) bis 89999 (hoch) erstellen wollen, geben Sie Folgendes an:

```
RAND(1000, 89999)
```

In diesem Beispiel gibt die Funktion Zufallsverkaufswerte im von Ihnen angegebenen Bereich von 1000,00 bis 89999,99 zurück.

Sequenzielle Funktion

Die sequenzielle Funktion gibt eine Zahl zurück, die sequenziell erhöht ist. Die Syntax ist:

```
SEQ(start, step)
```

start Startwert.

step Inkrementeller Wert.

- Sie können die sequenzielle Funktion mit Zeichendaten und numerischen Daten verwenden.
- Wenn die Ländereinstellung ein Komma als Dezimaltrennzeichen verwendet, müssen Sie nach dem Komma ein Leerzeichen lassen.
- *start* und *step* sind ganze Zahlen im Bereich von **-2.147.483.648** bis **2.147.483.647**.
- *start* und *step* werden durch den Datentyp und die Länge der Zielspalte zusätzlich begrenzt.

- Übersteigt der errechnete Wert die Länge der Zielspalte, wird die Funktion automatisch auf den *start*-Wert gesetzt.
- Wenn Sie die sequenzielle Funktion in einem verketteten Ausdruck verwenden, wird eine Zeichenfolge variabler Länge zurückgegeben.

Beispiel 1

Sie können mit der sequenziellen Funktion Kundendaten für eine Testdatenbank ändern. Angenommen, die Spalte CUST_ID ist als CHAR(5) definiert. Wenn Sie den Wert ausgehend von 1 um 50 erhöhen wollen, geben Sie Folgendes an:

```
SEQ(1, 50)
```

In diesem Beispiel gibt die Funktion CUST_ID-Werte ab 00001 zurück und generiert 00051, 00101 usw. durch Erhöhung um 50. Wenn das Ergebnis 99951 überschreitet, wird die Funktion auf den *start*-Wert 1 zurückgesetzt.

Beispiel 2

Sie können mit der sequenziellen Funktion in einer Spaltenzuordnung Verkaufsdaten für eine Testdatenbank maskieren. Angenommen, die Spalte YTD_SALES ist als DECIMAL(7,2) definiert. Wenn Sie den Wert ausgehend von 1000 um 100 erhöhen wollen, geben Sie Folgendes an:

```
SEQ(1000, 100)
```

In diesem Beispiel gibt die Funktion YTD_ID-Werte ab 1000 zurück und generiert die Werte 1100, 1200 usw. durch Erhöhung um 100. Wenn das Ergebnis 99999 überschreitet, wird die Funktion auf den *start*-Wert 1000 zurückgesetzt.

Beispiel 3

Angenommen, die Spalte SALESMAN_ID ist als CHAR(6) definiert. Wenn Werte, die mit 'NJ' beginnen, gefolgt von einer Anzahl ab 50 mit einem Erhöhungswert von 10 eingefügt werden sollen, verwenden Sie die Funktion in einem verketteten Ausdruck:

```
'NJ' || SEQ(50, 10)
```

In diesem Beispiel gibt die Funktion SALESMAN_ID-Werte ab 'NJ50 ' zurück und generiert die Werte 'NJ60 ', 'NJ70 ' usw. durch Erhöhung um 10. Wenn das Ergebnis 'NJ9990' überschreitet, wird die Funktion auf den *start*-Wert zurückgesetzt.

Identitäts- oder serielle Funktion

Die Funktionen IDENTITY und SERIAL weisen das Datenbankverwaltungssystem an, einen sequenziellen Wert (ganze Zahl) für eine Zielspalte bereitzustellen. Diese Funktionen haben folgende Syntax:

```
IDENTITY( )
```

```
SERIAL ( )
```

- Die Funktion IDENTITY wird für Identitätsspalten in DB2-, Sybase ASE- und SQL Server-Datenbanken verwendet.
- Die Funktion SERIAL wird für serielle Spalten in Informix-Datenbanken verwendet.
- Beide Funktionen sind für die Einfügeverarbeitung (Aktualisieren/Einfügen) und die Ladeverarbeitung, aber nicht für die Konvertierungsverarbeitung gültig.

- Wenn Zeilen in einem Einfügeprozess (Aktualisieren/Einfügen) aktualisiert werden, behält die von der Funktion IDENTITY oder SERIAL verwendete Zielspalte den ursprünglichen Wert bei. Falls die Zielspalte zusätzlich Teil des Primärschlüssels ist, bleibt der Spaltenwert unverändert, wenn die Zeile aktualisiert wird.
- Sie können die Funktion IDENTITY oder SERIAL mit der Weitergabefunktion für die Einfügeverarbeitung verwenden. Allerdings können Sie IDENTITY- oder SERIAL-Spalten nicht in einen Ladeprozess weitergeben.

Oracle-Sequenzfunktion

Die Oracle-Sequenzfunktion weist der Zielspalte durch Verwendung einer Oracle-Sequenz einen Wert zu.

`schema.seqname.NEXTVAL [INCL_UPD]`

schema Qualifikationsmerkmal für den Namen der Oracle-Sequenz.

seqname

Name der Oracle-Sequenz, die sequenzielle Werte zuordnet.

NEXTVAL

Schlüsselwort, das den nächsten Oracle-Wert in die Zielspalte einfügt.

INCL_UPD

Optionales Schlüsselwort, das einen Sequenzwert aktualisiert, der einer Spalte zuordnet ist, wenn Zeilen während eines Einfügeprozesses aktualisiert werden. Wenn der Wert nicht angegeben ist (Standardwert), bleibt der Spaltenwert unverändert, wenn die Zeile aktualisiert wird.

- Sie können mit der Oracle-Sequenzfunktion Zeilen, die in eine Oracle-Datenbank eingefügt werden sollen, eindeutige sequenzielle Werte zuordnen.
- Die Oracle-Sequenzfunktion ist gültig, wenn sie in einer Spaltenzuordnung für die Einfüge- oder Ladeverarbeitung verwendet wird, aber sie ist nicht für die Konvertierungsverarbeitung gültig.
- Wenn Zeilen in einem Einfügeprozess (Aktualisierungs-/Einfügeprozess) aktualisiert werden und die Zielspalte Teil des Primärschlüssels ist, bleibt der Spaltenwert unverändert, wenn die Zeile in der Zieltabelle aktualisiert wird. Wenn die Oracle-Sequenz bei der Durchführung einer Aktualisierung/Einfügung verwendet werden soll, geben Sie *INCL_UPD* in die Funktion an.
- Während eines Ladeprozesses verwendet der Prozess die Oracle-Sequenzfunktion zum Zuordnen eines neuen Werts zu jeder Zielzeile. Die Ladefunktion fordert das Abrufen dieser Werte durch das Datenbankverwaltungssystem an. Wenn Sie auswählen, dass das Oracle-Ladeprogramm nicht ausgeführt werden soll, werden diese Sequenzwerte nie verwendet.

Beispiel 1

Wenn Sie einen sequenziellen Wert zum Erhöhen von Kundennummern zuordnen wollen und der Name der Oracle-Sequenz **schema.numeven** ist, geben Sie Folgendes an:

`schema.numeven.NEXTVAL`

Beispiel 2

Wenn Sie das erste Beispiel erweitern wollen und einen vorhandenen Sequenzwert aktualisieren wollen, geben Sie Folgendes an:

`schema.numeven.NEXTVAL (INCL_UPD)`

Weitergabefunktion für Primärschlüssel- oder Fremdschlüsselwerte

Die Weitergabefunktion ordnet einer Primärschlüssel- oder Fremdschlüsselspalte einen Wert zu und gibt diesen Wert an alle zusammengehörigen Tabellen weiter.

Die Syntax ist:

```
PROP( { value [, columnname ] EXIT exitname |  
PROC { LOCAL | identifizier.name } } )
```

value Wert, der einer Spalte zugewiesen werden soll. Geben Sie einen beliebigen gültigen Quellenwert für Spaltenzuordnungen an (z. B. einen Spaltennamen, ein Zeichenfolgeliteral, einen Ausdruck oder eine Funktion) an. Der Wert muss für die Spalte geeignet sein.

columnname

Name der Quellenspalte, die den Wert enthält, der Gegenstand der Funktion ist. Der Ergebniswert wird in die Zielspalte der zugeordneten Tabelle und die entsprechende Zielspalte der mitwirkenden zugehörigen Tabellen eingefügt.

Der Spaltenname ist nur dann erforderlich, wenn keine Quellenspalte mit der Zielspalte bei Name und Datentyp übereinstimmt. Wenn kein Wert angegeben wurde, wird der Name der Zielspalte verwendet.

exitname

Exitname einer Spaltenzuordnung.

identifizier.name

Prozedurname einer Spaltenzuordnung.

- Wenn die Ländereinstellung ein Komma als Dezimaltrennzeichen verwendet, müssen Sie nach jedem Komma, das numerische Parameter trennt, ein Leerzeichen lassen.
- Die Weitergabefunktion ist in einer Spaltenzuordnung für die Einfügungsverarbeitung (aber nicht Aktualisierungs- oder Aktualisierungs-/Einfügungsverarbeitung), Ladeverarbeitung oder Konvertierungsverarbeitung gültig.
- Wenn Sie die Weitergabefunktion verwenden, muss mindestens eine zugehörige Tabelle im Prozess enthalten sein. Sie können die Weitergabefunktion mehrmals für denselben Prozess verwenden.
- Sie können die Weitergabefunktion für eine Primärschlüsselspalte oder die dazugehörige Fremdschlüsselspalte, aber nicht für beide verwenden.
- Wenn mehrere Spalten eine Beziehung definieren, können Sie die Weitergabefunktion für mindestens eine dieser Spalten verwenden. Allerdings können Sie in einer erweiterten Optim-Beziehung die Weitergabefunktion nur in Spalten-zu-Spalten-Beziehungen angeben.
- Sie können die Identitätsfunktion oder die serielle Funktion in der Weitergabefunktion zur Einfügeverarbeitung verwenden. Sie können jedoch die Identitätsfunktion nicht in einem Ladeprozess weitergeben.
- Die in der Weitergabefunktion angegebenen Parameter werden erst zur Ausführungszeit überprüft. Wenn Konflikte bestehen, wird der Prozess nicht ausgeführt.
- Einfügungen können Weitergabezyklen haben. Allerdings können die Lade- und Konvertierungsverarbeitung nicht zu Weitergabezyklen führen. Zyklen werden entdeckt, wenn der Prozess zur Ausführungszeit überprüft wird. Wenn eine Lade- oder Konvertierungsanforderung einen Weitergabezyklus generiert, wird der Prozess nicht ausgeführt.
- Optim speichert die Quellenwerte und die Werte, die den entsprechenden Zielspalten zugeordnet sind. Daher können Sie Daten in Zielspalten weitergeben, wo die Quelle ein Ausdruck ist. Wenn der ausgewertete Ausdruck mit einem Quellenwert übereinstimmt, ordnet Optim den entsprechenden Zielwert zu. Wenn der ausgewertete Ausdruck nicht mit irgendwelchen Quellenwerten übereinstimmt, tritt ein Konvertierungsfehler auf.

Bevor Sie einen Einfüge-, Lade- oder Konvertierungsprozess ausführen, können Sie die Spaltenzuordnung überprüfen, um zu sehen, wie die Weitergabefunktion im Prozess verwendet wird.

Beispiel 1

Sie können eine Zufallszahl generieren, sie der Standardzielspalte zuordnen und die Zahl in die Zielspalten zugehöriger Tabellen weitergeben. Wenn Sie einen Wert zwischen 10000 und 99999 generieren, in die zugeordnete Zielspalte einfügen und an die Zielspalten zugehöriger Tabellen weitergeben wollen, geben Sie Folgendes an:

```
PROP(RAND(10000, 99999))
```

Beispiel 2

Sie können dieselbe Funktion wie in Beispiel 1 ausführen, wenn die Quellen- und Zielspaltennamen nicht übereinstimmen. Wenn Sie den Namen der Quellspalte (CUST_NUMBER) in die Weitergabefunktion aufnehmen wollen, geben Sie Folgendes an:

```
PROP(RAND(10000, 99999), CUST_NUMBER)
```

Beispiel 3

Sie können die Oracle-Sequenzfunktion zum Generieren des Werts für die Zielspalte verwenden und diesen Wert in Zielspalten der zugehörigen Tabellen weitergeben. Geben Sie Folgendes ein, um die Oracle-Sequenz mit dem Namen **schema.numeven** weiterzugeben:

```
PROP(schema.numeven.NEXTVAL)
```

Verkettete Ausdrücke

Mithilfe einer Verkettung können Sie Spaltenwerte oder einen Spaltenwert mit einem anderen Wert unter Verwendung eines Verkettungsoperators (CONCAT, || oder +) kombinieren. Ein verketteter Ausdruck kann Zeichenwerte oder binäre Werte, jedoch nicht beide enthalten:

Zeichenwerte

Verkettete Zeichenwerte können Zeichenspalten, Zeichenfolgeliterale, Unterzeichenfolgen von Werten in Zeichenspalten, die sequenzielle Funktion oder die Zufallsfunktion sein.

Binärwerte

Verkettete Zeichenwerte können Binärspalten, hexadezimale Literale, Unterzeichenfolgen von Binärspalten, die sequenzielle Funktion oder die Zufallsfunktion sein.

- Ein verketteter Ausdruck kann kein Zeichenfolgeliteral der Länge null (' '), kein Sonderregister oder keine Alterungsfunktion enthalten.

Beispiel

Angenommen, in der Tabelle CUSTOMERS wird eine Adresse in zwei Spalten gespeichert: ADDRESS1 und ADDRESS2. In der Tabelle SHIP_TO wird eine Adresse in einer Spalte gespeichert: ADDRESS. Sie können mit einem verketteten Ausdruck Adressinformationen aus zwei Spalten in einer Tabelle in eine Spalte einer anderen kombinieren.

Geben Sie eines der folgenden Beispiel ein, um die Adresse zu kombinieren:

Tabelle CUSTOMERS

```
ADDRESS1 || ADDRESS2
```

```
ADDRESS1 CONCAT ADDRESS2
```

```
ADDRESS1 + ADDRESS2
```

Tabelle SHIP_TO

```
ADDRESS
```

```
ADDRESS
```

```
ADDRESS
```

Numerische Ausdrücke

Mithilfe eines numerischen Ausdrucks können Sie einen Wert in der Quellenspalte angeben, wenn die Datentypen für die entsprechenden Quellenspalten oder Zielspalten kompatibel sind.

Ein numerischer Ausdruck hat folgende Syntax:

Operand1 Operator Operand2

Jeder Operand muss eine numerische Spalte oder eine numerische Konstante sein. Der Operator gibt an, ob addiert (+), subtrahiert (-), dividiert (/) oder multipliziert (*) werden soll.

Beispiel 1

Wenn Sie den Wert in einer Spalte UNIT_PRICE, die als DECIMAL(5,2) definiert ist, um 10 Prozent erhöhen wollen, können Sie Folgendes angeben:

1.1 * UNIT_PRICE

Beispiel 2

Wenn Sie den Wert in einer Spalte ON_HAND_INVENTORY des Typs INTEGER halbieren wollen, können Sie Folgendes angeben:

ON_HAND_INVENTORY / 2.

Literal- und Wertfunktionen

Mit den Literal- und Wertfunktionen können Literale wie Zeichenfolgen oder Hexadezimalwerte und Werte wie NULL oder ein Sonderregister angegeben werden.

Funktion	Beschreibung
Spaltenname	Ein expliziter Spaltenname (bei Spaltennamen wird Groß-/Kleinschreibung nicht berücksichtigt).
NULL	NULL. In der Zielspalte muss die Dateneingabe optional sein.
Numerische Konstante	Eine numerische Konstante. Der Wert der Konstante muss in die Zielspalte passen wie über ihren Datentyp, ihre Genauigkeit und ihre Anzahl von Kommastellen definiert.
Boolesche Konstante	Eine boolesche Konstante (TRUE oder FALSE).

Funktion	Beschreibung
Sonderregister	<p>Ein Sonderregister:</p> <ul style="list-style-type: none"> • CURRENT DATE • CURRENT_DATE • CURRENT TIME • CURRENT_TIME • CURRENT TIMESTAMP • CURRENT_TIMESTAMP • CURRENT SQLID • CURRENT_SQLID • CURDATE() • CURTIME() • GETDATE() • GETTIME() • SYSDATE() • NOW() • WORKSTATION_ID • USER
Zeichenfolge-literal	<p>Ein Zeichenfolgeliteral in einfachen Anführungszeichen. Die Zielspalte muss Zeichendaten enthalten.</p> <p>Beispiel: 'CA' oder '90210'.</p>
Hexadezimaler Literal	<p>Ein hexadezimaler Literal.</p> <p>Beispiel: X'1234567890ABCDEF' oder 0X1234567890ABCDEF</p>
Literal für Datum/Zeit	<p>Ein Datums-/Uhrzeitliteral in einfachen Anführungszeichen. Trennen Sie das Datum und die Uhrzeit mit einem Leerzeichen. Wenn Sie das Datum/die Uhrzeit mit einem Dezimalbruch formatieren wollen, setzen Sie nach der Uhrzeit einen Punkt gefolgt von der Bruchzahl. Das Datumsformat wird durch die Einstellungen der Regionaloptionen in der Systemsteuerung Ihres Computers bestimmt.</p> <p>Anmerkung: Für Oracle-Spalten, die eine Zeitmarke mit Zeitzone umfassen, müssen Sie das Zeitzonensuffix zuletzt angeben.</p>

Alterungsfunktion

Mit der Alterungsfunktion können Sie Werte in einer Quellenspalte nach Verweildauer steuern. Die Quellenspalte kann Zeichendaten, numerische Daten oder Datums- oder Uhrzeitangaben enthalten. Eine CHAR- oder VARCHAR-Spalte hat eine maximale Länge von 256 Byte.

Das Format der Alterungsfunktion ist wie folgt:

AGE(parameters)

- Definieren Sie die Alterungsfunktion so, dass mindestens ein Parameter zur Steuerung nach Verweildauer angegeben wird.
- Verwenden Sie Kommas oder Leerzeichen zum Trennen von Parametern in der Alterungsfunktion.
- Parameter können in beliebiger Reihenfolge angegeben werden.

Parameter	Format	Gültige Werte
Spaltenname: Geben Sie den Namen der Quellenspalte an, wenn er von dem der Zielspalte abweicht.	<p><i>SC=column-name</i></p> <p><i>SRCCOL=column-name</i></p>	Spaltenname

Parameter	Format	Gültige Werte
Standard: Alter von Datumsangaben auf der Basis des Datumsanpassungswerts, der in einer Prozessanforderung angegeben ist, steuern.	DEF	Verwendet den in der Prozessanforderung angegebenen Datenanpassungswert.
Keine Angabe: Wert nicht nach Verweildauer steuern.	NONE	Das Alter des Wert soll nicht gesteuert werden, ungeachtet der Angaben in der Prozessanforderung.
Inkrementell – inkrementelle Alterung basiert auf einer bekannten Zeiteinheit. Optim unterstützt die Datumsalterung in einzelnen Einheiten (z. B. 20 Jahre) oder mehreren Einheiten (z. B. 2 Jahre, 3 Monate, 2 Tage).	+ oder -] nY [+ oder -] nM [+ oder -] nW [+ oder -] nD (Das Pluszeichen [+] ist optional.)	nY -2500 bis +1581 nM -30000 bis +30000 nW -30000 bis +30000 nD -99999 bis +99999
Bestimmtes Jahr: Alter von Datumsangaben auf der Basis eines bestimmten mit vier Ziffern angegebenen Jahrs im gewünschten Format steuern.	nnnnY	1582 - 3999
Mehrfach/Regel: Alter von Datumsangaben auf der Basis der Häufigkeit, mit der eine Geschäftsregel angewendet wird, steuern. Wenn Sie die Alterungsfunktion auf der Basis von "Mehrfach/Regel" definieren, müssen Sie auch den Parameter RULE angeben.	nnnnnR	1 - 30000

Semantische Alterung

Semantische Alterung basiert auf einem Satz von Regeln, die Sie definieren, um Datumsangaben zu verwalten, die auf Feiertage, Wochenenden usw. fallen. Sie können mithilfe von semantischer Alterung Datumsangaben anpassen, sodass sie auf gültige Werkzeuge fallen.

Kalender –

Name des Kalenders, der die besonderen Datumsangaben definiert, auf die die Regeln angewendet werden. Wenn Sie CALENDAR verwenden, müssen Sie auch eine Regel mit RULE angeben.

CA=calendar-name

CALENDAR=calendar-name

Regel –

Der Name der Regel, die die Anpassungen für besondere Datumsangaben definiert. Wenn DEF angegeben ist, wird die in der Prozessanforderung angegebene Standardregel verwendet.

RU=rule-name

RULE=rule-name

RU=DEF

RULE=DEF

Jahrhundert-Pivot-Map –

Gibt das Jahrhundert für zweistellige Jahresangaben an. Geben Sie einen Wert von 00 bis 99 ein.

PI=nn

PIVOT=nn

- Wenn Sie **AGE(RU=DEF)** angeben, wird der in einer Prozessanforderung angegebene Wert für RULE verwendet. Sie müssen Werte für alle anderen Alterungsfunktionsparameter angeben.
- Wenn Sie RULE verwenden und keinen Wert für CALENDAR angeben, verwendet die Alterungsfunktion den Standardkalender, den Sie in einer Prozessanforderung angeben.
- Wenn Sie in der Alterungsfunktion nicht erforderlichenfalls CALENDAR, RULE und PIVOT angeben, gelten die Standardwerte, die Sie in einer Prozessanforderung angegeben haben.
- Wenn Sie das Jahrhundert für ein zweistelliges Jahr angeben wollen, müssen Sie PIVOT in die Alterungsfunktion aufnehmen.
- Wenn Sie einen PIVOT-Wert angeben, werden alle zweistelligen Jahresangaben, die nicht kleiner als der PIVOT-Wert sind, ins 20. Jahrhundert verlegt (19xx). Alle Jahre mit zwei Ziffern, die kleiner als der PIVOT-Wert sind, werden ins 21. Jahrhundert verlegt (20xx). Der PIVOT-Standardwert ist 65.

Datumsformate

Das Quelldatumsformat und das Zieldatumsformat müssen ein einzelnes gültiges Datumsformat enthalten und dürfen nicht größer als die Länge der Zielspalte sein. Die Formatierzeichenfolge muss durch einfache Anführungszeichen begrenzt sein.

Quelldatumsformat –

Wendet die Zeichenfolge für das Quellenspaltenformat auf Zeichen- und numerische Spalten zur Steuerung der Verweildauer an.

SF='format-string'

SRCFMT='format-string'

Wenn die Quellenspalte eine Zeichenspalte oder eine numerische Spalte ist, müssen Sie den Inhalt der Spalte mit SRCFMT oder einer Quellenexitroutine (SRCEXIT) beschreiben. Diese Parameter schließen sich gegenseitig aus.

Zieldatumsformat –

Wendet die Zeichenfolge für das Zielspaltenformat auf Zeichen- und numerische Spalten zur Steuerung der Verweildauer an.

DF='format-string'

DSTFMT='format-string'

Wenn die Zielspalte eine Zeichenspalte oder eine numerische Spalte ist, können Sie DSTFMT oder eine Zielexitroutine (DSTEXIT) angeben. Wenn Sie kein Format für das Ziel angeben, verwendet die Datumsalterungsfunktion standardmäßig SRCFMT. Die Zielspalte für eine Funktion AGE kann nicht binär sein.

Mithilfe der folgenden Zeichenfolgen können Sie Komponenten des Datumsformats angeben:

Jahr	Monat	Tag	Zeit	Teile/Sekunde
JJJJ	MONAT	TTT	HH	FFFFFF
CCJJ	MMM	TT	MI	FFFFFF
JJ	MM	T	SS	FFFF
	M			FFF
				FF
				F

- Wenn Sie ein Fragezeichen (?) in einer Formatierzeichenfolge angeben, ordnet die Alterungsfunktion den Zeichenwert so zu, wie er ist. (Mit dem Fragezeichen können Sie Schrägstriche, Gedankenstriche, Punkte und andere Zeichen in das Datumsformat einfügen.)
- Wenn Sie einen Stern (*) in eine Formatierzeichenfolge einfügen, ordnet die Alterungsfunktion alle verbleibenden Zeichen in der Quellenspalte der Zielspalte zu. (Verwenden Sie den Stern, wenn der Spaltenwert ein Datum ist, das mit zusätzlichen Zeichen verkettet ist.)

Anmerkung: Sie können mit dem Dienstprogramm 'Calendar' ein Standardtrennzeichen und ein Standardausgabejahr definieren. Diese Standardwerte gelten, wenn die Quellen- und Zielformate Trennzeichen oder ein bestimmtes Jahr erfordern.

Beispiel 1

Wenn Sie das Alter einer Datumsspalte um 2 Jahre, 6 Monate, 40 Wochen und 15 Tage erhöhen und dann eine Regel anwenden wollen, geben Sie die Alterungsfunktion mit folgenden Parametern an:

```
AGE(+2Y,+6M,+40W,+15D,RU=NEXTPAYDAY)
```

Beispiel 2

Wenn Sie nur den Jahresteil in einer Datumsspalte auf das Jahr 2020 setzen und eine Regel anwenden wollen, geben Sie die Alterungsfunktion mit folgenden Parametern an:

```
AGE(2020Y,RU=NEXTWORKDAY)
```

Beispiel 3

Wenn Sie das Alter einer Datumsspalte mit "Mehrfach/Regel" so steuern wollen, dass es durch fünfmaliges Anwenden einer Regel mit dem Namen NEXTSTRTQTR unter Verwendung eines Kalenders PSAPRULE erhöht wird, geben Sie die Alterungsfunktion mit folgenden Parametern an:

```
AGE(CA=PSAPRULE,RU=NEXTSTRTQTR,5R)
```

Beispiel 4

Sie wollen das Alter von Daten in einer Zeichenspalte oder numerischen Spalte über die folgenden Parameter steuern:

- Eine benannte Quellenspalte.
- Das Quellenformat, bei dem die ersten zwei Zeichen für die letzten beiden Ziffern des Jahres und die verbleibenden drei Ziffern als Tag im julianischen Kalender verwendet werden.
- Eine Jahrhundert-Pivot-Map zum Festlegen des richtigen Jahrhunderts, weil die Quelle mit einem aus zwei Ziffern bestehenden Jahr formatiert ist. Die Jahrhundert-Pivot-Map ist in diesem Beispiel 42. Alle mit zwei Ziffern dargestellten Jahre, die nicht unter 42 liegen, werden im 20. Jahrhundert (19xx) angesiedelt. Alle mit zwei Ziffern dargestellten Jahre, die unter 42 liegen, werden im 21. Jahrhundert (20xx) angesiedelt.
- Alter des Datums um fünf Jahre erhöhen.

Geben Sie die Alterungsfunktion wie folgt an:

```
AGE(5Y,SC=ORDER_DATE,SF='YYDDD',PI=42)
```

Währungsfunktion

Mit der Währungsfunktion können Sie einen Währungswert in einer Quellenspalte von einer Währung in eine andere konvertieren. Die Quellenspalte muss als numerisch definiert sein, darf aber keine Gleitkommaspalte sein. Es stehen zwei Konvertierungsverfahren zur Verfügung:

Direkte Konvertierung

Stellen Sie Konvertierungsparameter auf der Basis von Werten, die in einer Währungsdefinition festgelegt sind, bereit. Mit der Währungsfunktion können Sie einen Geldwert in einer Spalte konvertieren (wodurch der frühere Wert ersetzt wird) oder, indem unterschiedliche Quellen- und

Zielspalten definiert werden, den ursprünglichen Wert und den konvertierten Wert beibehalten. Sie können explizit die Quellen- und Zielwährungstypen definieren oder eine Referenzspalte zum Anzeigen des Währungstyps angeben.

Die erste Vorgabe für die Berechnung ist die Verwendung der Umrechnungsrate für die Quellenwährung in die Zielwährung. Die zweite Vorgabe für die Berechnung ist die Verwendung der Umrechnungsrate für die Zielwährung in die Quelle.

Triangulation

Konvertieren Sie den Wert in einer Spalte aus dem Quellenwert in den Eurodollar und konvertieren Sie anschließend den Eurodollar in die Zielwährung. Beide Kurse müssen in der Kurstabelle angegeben werden: Einer zwischen dem Eurodollar und der Quellenwährung und einer zwischen dem Eurodollar und der Zielwährung. Der Spezifikationsausdruck ist TRIANG oder TR.

Die Währungsfunktion hat folgende Syntax:

```
CURRENCY( {ST=Code | SS=(Spaltenname, Typentabellennummer)}
{DT=Code | DS=(Spaltenname, Typentabellennummer)}
[SC=Spaltenname] [TR] [CU=Währungsdefinitionsname]
[TD=Name_der_Transaktionsdatumsspalte] [DF='format']
[NS=Anzahl_Kommastellen] )
```

Die Währungsfunktion muss mindestens eine Kombination aus Quellenwährungstyp (ST) oder Quellspezifikation (SS) und Zielwährungstyp (DT) oder Zielspezifikation (DS) enthalten. Alle anderen Parameter sind optional.

Die Quellen- und Zielwährungstypen können auf eine von zwei Arten angegeben werden:

1. Die Schlüsselwörter ST und DT ermöglichen die explizite Angabe der Währung mit dem aus drei Zeichen bestehenden Währungscode nach ISO 4217.
2. Die Schlüsselwörter SS und DS ermöglichen die indirekte Angabe des Währungstyps, wobei ein Wert in einer benannten Spalte in der Zeile als Schlüssel verwendet wird. Der Schlüssel ist mit einem Währungstyp korreliert, wie er in der angegebenen Währungsdefinitionstypentabelle definiert ist.

Wenn Sie ein Transaktionsdatum (TD) angeben und die Transaktionsdatumsspalte nicht das Format DATE verwendet, müssen Sie auch ein Datumsformat (DF) angeben. Wenn ein angegebenes Transaktionsdatum sich außerhalb des Bereichs befindet, der in der Tabelle für Währungsdefinitionskurse angegeben ist, wird der nächstgelegene Datumsbereich für Konvertierungsberechnungen verwendet.

Anmerkung: Wenn erforderliche Daten (z. B. Währungstypen oder Kurse) fehlen, treten zur Ausführungszeit Konvertierungsfehler auf.

In der folgenden Tabelle werden das gültige Format und zulässige Werte für die Währungsfunktionsparameter beschrieben. Parameter können in beliebiger Reihenfolge angegeben werden. Sie können Kommas oder Leerzeichen als Trennzeichen zwischen Parametern in der Währungsfunktion verwenden.

Parameter	Format
Quellenspalte	SC=Spaltenname SRCCOL=Spaltenname
Quellenwährungstyp	ST=Code SRCTYP=Code Dabei ist Code ein Währungscode nach ISO 4217
Zielwährungstyp	DT=Code DSTTYP=Code Dabei ist Code ein Währungscode nach ISO 4217

Parameter	Format
Quellenspezifikation	SS=Spaltenname, Typentabellennummer SRCSPC=Spaltenname, Typentabellennummer Dabei ist Spaltenname, Typentabellennummer die in der Währungsdefinition definierte Typentabellenspalte und -nummer, die zur Angabe des Quellenwährungstyps verwendet wird
Zielspezifikation	DS=Spaltenname, Typentabellennummer DSTSPC=Spaltenname, Typentabellennummer Dabei ist Spaltenname, Typentabellennummer die in der Währungsdefinition definierte Typentabellenspalte und -nummer, die zur Angabe des Zielwährungstyps verwendet wird
Triangulation (erzwingt eine Konvertierung über den Eurodollar)	TR TRIANG
Währungsdefinition	CU=Währungsdefinitionsname CURTBL=Währungsdefinitionsname Dabei ist Währungsdefinitionsname die Währungsdefinition, die die geeigneten Konvertierungsparameter enthält.
Transaktionsdatum	TD=Spaltenname TRNDAT=Spaltenname Dabei ist Spaltenname der Name der Transaktionsdatumsspalte, die das Konvertierungsdatum angibt
Datumsformat	DF='Format' DATFMT='Format'. Dabei ist Format das Format der Transaktionsdatumsspalte, wenn sie nicht den Datentyp DATE hat.
Numerische Skala	NS=Anzahl_Kommastellen NUMSCL=Anzahl_Kommastellen Dabei ist Anzahl_Kommastellen die Anzahl der Kommastellen, die auf numerische Oracle-Zielspalten mit einer nicht definierten Anzahl von Kommastellen angewendet werden soll.

Beispiel 1

Wenn eine Konvertierung von Finnmark in Eurodollar erfolgen soll, verwenden Sie die Währungsfunktion wie folgt:

```
CURRENCY(ST=FIM DT=EUR)
```

Wenn der ursprüngliche Wert beibehalten werden soll, müssen Sie mit der Währungsfunktion einen Wert für eine andere Spalte in der Zieltabelle bereitstellen.

Beispiel 2

Wenn eine Konvertierung von Finnmark in Eurodollar erfolgen soll und eine neue Spalte zum Beibehalten des ursprünglichen Quellenwerts (in Finnmark) in einer Spalte ITEM_COST erstellt werden soll, verwenden Sie die Währungsfunktion wie folgt:

```
CURRENCY(ST=FIM DT=EUR SC=ITEM_COST)
```

Automatisch generierter E-Mail-Name

Mit der Funktion für den automatisch generierten E-Mail-Namen wird eine E-Mail-Adresse mit einem Benutzernamen auf der Basis eines Literals generiert, das mit einer fortlaufenden Zahl verkettet ist.

Die fortlaufenden Zahlen sind Suffixe, die mit 1 beginnen und um 1 erhöht werden. Die Funktion verwendet den Domännennamen aus einer E-Mail-Adresse in einer angegebenen Quellenspalte.

Formatierter E-Mail-Name

Die Funktion für den formatierten E-Mail-Namen generiert eine E-Mail-Adresse mit einem Benutzernamen auf der Basis von Werten, die aus mindestens einem Attribut abgerufen wurden. Die Richtlinie verwendet den Domännennamen aus einer E-Mail-Adresse in einer angegebenen Quellenspalte.

Funktion für Zufallszahlen

Die Funktion für Zufallszahlen generiert Zahlen, die zufällig aus dem Bereich ausgewählt werden, der durch den niedrigsten und den höchsten Wert angegeben wird.

Sie können die Funktion für Zufallszahlen verwenden, um Zeichendaten oder numerische Daten zu ersetzen. Der niedrige und der hohe Wert müssen ganze Zahlen sein, die sich im Bereich von -2.147.483.648 bis 2.147.483.647 befinden. Der niedrige Wert muss kleiner als der hohe Wert sein.

Funktion für fortlaufende Zahlen

Die Funktion für fortlaufende Zahlen generiert Zahlen, die sequenziell erhöht werden.

Sie können die Funktion für fortlaufende Zahlen verwenden, um Zeichendaten oder numerische Daten zu ersetzen. Sie müssen einen Startwert und einen Wert angeben, um den die Zahlen inkrementell erhöht werden. Der Startwert und der Wert für die inkrementelle Erhöhung müssen Ganzzahlen im Bereich von -2.147.483.648 bis 2.147.483.647 sein.

Der generierte Wert wird durch den Datentyp und die Länge der Zielspalte begrenzt. Übersteigt der generierte Wert die Länge der Zielspalte, wird die Funktion automatisch auf den Anfangswert zurückgesetzt.

Exitroutinen verwenden

Wenn Sie einen Service zum Konvertieren, Erstellen, Einfügen, Laden oder Wiederherstellen von Daten erstellen, können Sie eine Tabellenzuordnung angeben, die mindestens eine Spaltenzuordnung enthält, um die geeigneten Werte für die Zielspalten abzuleiten.

Es gibt verschiedene Arten, einen Spaltenwert in einer Spaltenzuordnung anzugeben. Eine Art ist die Angabe einer Exitroutine als Quelle; der Exit legt Werte fest, die andernfalls nicht für Zielspalten definiert werden könnten. Eine andere Methode ist die Verwendung von Exitroutinen zum Ausschließen von Zeilen aus der Verarbeitung. Sie können drei Typen von Exitroutinen verwenden:

Standardexit

Die Standardexitroutine wird aufgerufen, um den Wert für eine Zielspalte in einer Spaltenzuordnung abzuleiten. Dieser Typ von Exitroutine ist nützlich, wenn Sie Datenumsetzungen ausführen wollen, die über den Umfang von Spaltenzuordnungen hinausgehen. Ein Exit kann z. B. die Abteilungsnummer eines Mitarbeiters für ausgewählte Zeilen nach einem komplexen Algorithmus ändern oder bestimmte Zeilen für die Verarbeitung auswählen und alle anderen löschen.

Ein Standardexit kann ein Unterzeichenfolgesegment einer LOB-Quellenspalte abrufen. Ein Exit kann eine Datei erstellen und den Dateinamen an den Spaltenzuordnungsprozessor zurückgeben, um einen neuen LOB-Wert in eine LOB-Zielspalte einzufügen.

Quellenformatexit

Der Quellenformatexit wird zum Formatieren der Quellenspalte in einer Alterungsfunktion aufgerufen, die normalerweise nicht in einer Spaltenzuordnung unterstützt würde. Diese Exitroutine untersucht das Quelldatum in einer Zeichen- oder Ganzzahlspalte und konvertiert es in ein Datumsformat, das als Eingabe in die Alterungsfunktion verwendet werden kann.

Zielformatexit

Der Zielformatexit wird zum Formatieren der Zielspalte in einer Alterungsfunktion aufgerufen, die normalerweise nicht in einer Spaltenzuordnung unterstützt würde. Diese Exitroutine konvertiert ein Datum in eines von vier verschiedenen Zielformaten. Der Datentyp der Zielspalte legt das Format fest.

Exit in einer Spaltenzuordnung

Wenn Sie eine Exitroutine in einer Spaltenzuordnung verwenden wollen, müssen Sie eine der folgenden Angaben in der entsprechenden Quellenspalte machen:

Standardexit

```
EXIT dllname[(parm[,]parm)... ]
```

Quellenformatexit

```
AGE(SRCEXIT=dllname)
```

Zielformatexit

```
AGE(DSTEXIT=dllname)
```

Der Prozess ruft diese Spaltenzuordnungsexitroutine einmal für jede verarbeitete Datenzeile auf und übergibt einen Beendigungsaufwurf, nachdem die letzte Zeile verarbeitet ist. Optionale Parameter, die mit einer Standardexitroutine angegeben werden, werden an den Exit übergeben und müssen Zeichenfolgeli-terale (die in einfache Anführungszeichen eingeschlossen sind) oder numerische Literale (Maximalwert 8) sein.

Exitroutinen schreiben

Sie können Exitroutinen in jeder Programmiersprache schreiben, allerdings müssen die Aufrufe an Subroutinen den in der Programmiersprache C verwendeten Konventionen entsprechen.

Headerdateien

Damit die Parameter, Strukturen und Rückgabecodes definiert werden, die in einer Exitroutine verwendet werden, muss diese zwei C-Programmheaderdateien, PSTEXIT.H und, je nach Zeichenformat der Metadaten (Tabellennamen, Spaltennamen usw.) entweder PSTCMXIT.H oder PSTCMWXT.H beinhalten:

PSTEXIT.H

Gibt die Datentypen, Rückgabecodes und Strukturen für Optim-definierte Datentypen an.

PSTCMXIT.H

Für Metadaten in Einzelbyte-Format (z. B. ASCII).

Stellt Prototypen für Spaltenzuordnungs-Callback-Routinen bereit und gibt die Definitionen (*defines* und *structure definitions*) für die Spaltenzuordnungs-Exitparameter an.

PSTCMWXT.H

Für Metadaten in UTF-16-Format (WCHAR).

Stellt Prototypen für Spaltenzuordnungs-Callback-Routinen bereit und gibt die Definitionen (*defines* und *structure definitions*) für die Spaltenzuordnungs-Exitparameter an.

Diese Dateien befinden sich im selben Verzeichnis wie die IBM InfoSphere Optim-Anwendungsdateien.

Anmerkung: Wenn ein Parameter Einzelbyteformat und UTF-16-Format hat, wird in diesem Kapitel das UTF-16-Format in Klammern angegeben.

DLL-Dateien verwenden

Sie müssen jede Exitroutine als separate DLL-Datei kompilieren und verlinken. Optim lädt jede DLL-Datei dynamisch zur Ausführungszeit. Eine DLL-Datei kann nur ein einziges Vorkommen eines bestimmten Typs von Spaltenzuordnungsexit enthalten. Allerdings können Sie ein Vorkommen jedes Typs von Spaltenzuordnungsexitroutine in eine DLL-Datei aufnehmen.

Sie müssen den Namen der DLL-Datei in der Spaltenzuordnung verwenden. Geben Sie den Namen der DLL-Datei an, mit dem sie auf der Plattform definiert ist (verwenden Sie also nicht den generischen Namen). Schreiben Sie die DLL-Datei, um die wirklichen Funktionen, die die Exitroutine implementieren, zu benennen und zu exportieren:

Standardexit

PSTColMapExit
(PSTColMapWExit)

Quellenformatexit

PSTColMapAgeSrcExit
(PSTColMapAgeSrcWExit)

Zielformatexit

PSTColMapAgeDstExit
(PSTColMapAgeDstWExit)

Voraussetzungen

Jede Spaltenzuordnungsexitroutine muss folgende Anforderungen erfüllen:

- Der Exit muss einen Wert generieren, der für die Zielspalte geeignet ist, und darf keine anderen Speicherbereiche ändern.
- Optim verwendet den Primärschlüssel zum Erzeugen interner Arbeitsbereiche. Wenn die Exitroutine einer Spalte einen Wert zuweist, die Teil des Primärschlüssels ist, und diese Routine mehrmals für dieselbe Quellzeile aufgerufen wird, stellen Sie sicher, dass die Exitroutine jedes Mal denselben Ausgabewert generiert.
- Ein Exit kann häufig aufgerufen werden; vermeiden Sie unnötigen Systemaufwand. Der Exit sollte z. B. die Verarbeitung beim ersten Aufruf initialisieren und Informationen für nachfolgende Aufrufe im Arbeitsbereich speichern.

Standardexitroutine

Wenn Sie einen Standardexit schreiben, geben Sie die Exitfunktion, PSTColMapExit (PSTColMapWExit), und folgende Parameter an:

Einzelbyte

```
PSTColMapExit  
(PST_STRUCT_CM_EXIT_PARM * pInputParms,  
 PST_STRUCT_CM_EXIT_COL_LIST * pSrcColList,  
 PST_STRUCT_CM_EXIT_COL_LIST * pDstColList)
```

UTF-16

```
PSTColMapWExit  
(PST_STRUCT_CM_WEXIT_PARM * pInputParms,  
 PST_STRUCT_CM_WEXIT_COL_LIST * pSrcColList,  
 PST_STRUCT_CM_WEXIT_COL_LIST * pDstColList)
```

Parameter

Wenn ein Prozess eine Standardexitroutine aufruft, übergibt er bei jedem Aufruf folgende Parameter:

pInputParms

Verweis auf PST_STRUCT_CM_EXIT_PARM (PST_STRUCT_CM_WEXIT_PARM). Diese Struktur enthält Informationen zu den Quellen- und Zieltabellen, zur Art des aktuellen Aufrufs und zur Anzahl optionaler Parameter, die mit dem Exit angegeben wurden, sowie Verweise auf die Arbeitsbereiche für Callback-Funktionen und optionale Parameter.

Das erste Feld in dieser Struktur ist das Feld **FuncCode**, das durch PST_CM_FUNC_TRANSFORM (PST_CMW_FUNC_TRANSFORM) oder PST_CM_FUNC_TERMINATE (PST_CMW_FUNC_TERMINATE) angegeben wird.

Das Feld *NumParms* enthält die Anzahl der mit dem Spaltenzuordnungsexit angegebenen optionalen Parameter (0 bis 8).

Das Feld *pParm* enthält ein Array von Verweisen auf jeden mit dem Spaltenzuordnungsexit angegebenen optionalen Parameter.

pScrColList

Verweis auf PST_STRUCT_CM_EXIT_COL_LIST (PST_STRUCT_CM_WEXIT_COL_LIST). Diese Struktur beschreibt die Quellenspalten.

pDstColList

Verweis auf PST_STRUCT_CM_EXIT_COL_LIST (PST_STRUCT_CM_WEXIT_COL_LIST). Diese Struktur beschreibt die Zielspalten.

Callback-Funktionen

Jedes Mal, wenn ein Optim-Prozess eine Standardexitroutine aufruft, übergibt der Prozess die Adressen der folgenden Callback-Funktionen:

pPSTGetColValue()

Ruft Daten für alle Quellenspalten und die meisten Zielspalten in der aktuellen Datenzeile ab. Im Allgemeinen ruft die Exitroutine diese Funktion nur einmal auf, um die Daten für eine Quellenspalte abzurufen. Die Exitroutine kann diese Funktion jedoch mehrmals aufrufen, um Daten für unterschiedliche Spalten abzurufen.

pPSTPutColValue()

Gibt Daten für die Zielspalte in der aktuellen Datenzeile an. Die Exitroutine ermittelt den Wert für die Zielspalte und gibt den Wert an Optim zurück. Diese Funktion muss aufgerufen werden, wenn die Zeile nicht zurückgewiesen wird oder der Prozess nicht abgebrochen wird.

Verarbeitung

In den folgenden Schritten wird eine typische Verarbeitung für die Standardexitroutine zusammengefasst:

1. Bei jedem Aufruf durch Optim prüft die Exitroutine, ob es sich um einen erstmaligen Aufruf handelt. Beim ersten Aufruf führt die Exitroutine alle Initialisierungstasks sowie die normale Verarbeitung (Schritt 2) durch. Bei nachfolgenden Aufrufen wird nur noch die normale Verarbeitung (Schritt 2) durchgeführt.
2. Optim übergibt keine Spaltendaten an die Standardexitroutine; allerdings kann die Exitroutine Aufrufe an die Callback-Funktion **pPSTGetColValue()** ausführen, um Daten für die Quellenspalten abzurufen. Diese Werte werden benötigt, um den Wert der Zielspalte festzulegen.
3. Nachdem die Exitroutine den Zielwert generiert, ruft der Exit die Callback-Funktion zum Speichern des Werts in der Zielspalte auf oder übergibt einen geeigneten Rückgabecode, der den Prozess anweist, die Datenzeile zu überspringen oder den Vorgang abzubrechen.
4. Nachdem die letzte Datenzeile verarbeitet ist, übergibt Optim einen Beendigungsaufruf an die Exitroutine, der durch einen Wert PST_CM_FUNC_TERMINATE (PST_CMW_FUNC_TERMINATE) im

Feld **FuncCode** gekennzeichnet ist. Dieser Aufruf fordert die Exitroutine auf, dynamisch zugeordneten Speicher freizugeben. Wenn die Abschlusstasks beendet sind, übergibt die Exitroutine einen Rückgabecode an Optim.

Rückgabecodes

Die folgenden Rückgabecodes sind für Standardexitroutinen gültig:

```
PST_CM_EXIT_SUCCESS  
(PST_CMW_EXIT_SUCCESS)
```

Der Zielspalte wird ein Wert zugeordnet oder sie wird erfolgreich umgesetzt.

```
PST_CM_EXIT_REJECT_ROW  
(PST_CMW_EXIT_REJECT_ROW)
```

Der Zielspalte kann kein Wert zugeordnet werden oder sie kann nicht umgesetzt werden. Die Zeile wird gelöscht.

```
PST_CM_EXIT_ABORT_PROCESS  
(PST_CMW_EXIT_ABORT_PROCESS)
```

Fataler Fehler. Die Verarbeitung wird beendet. Die Nachricht wird in den Arbeitsbereich gestellt und der nicht verwendete Bereich wird auf Leerzeichen oder NULL gesetzt, damit eine Fehlernachricht zurückgegeben wird.

Quellenformatexit

Wenn Sie eine Quellenformatexitroutine für Optim, schreiben, geben Sie die Exitfunktion, PSTColMapAgeSrcExit (PSTColMapAgeSrcWExit), und folgende Parameter an:

Einzelbyte

```
PSTColMapAgeSrcExit  
(PST_STRUCT_CM_AGE_SRCFMT_PARM * pInputParms,  
 PST_STRUCT_CM_EXIT_COL_LIST * pSrcColList,  
 PST_STRUCT_CM_EXIT_COL_LIST * pDstColList)
```

UTF-16

```
PSTColMapAgeSrcWExit  
(PST_STRUCT_CM_AGE_SRCFMT_WPARAM *  
 pInputParms,  
 PST_STRUCT_CM_WEXIT_COL_LIST * pSrcColList,  
 PST_STRUCT_CM_WEXIT_COL_LIST * pDstColList)
```

Parameter

Wenn ein Optim-Prozess eine Quellenformatexitroutine aufruft, übergibt der Prozess folgende Parameter:

pInputParms

Verweis auf PST_STRUCT_CM_AGE_SRCFMT_PARM (PST_STRUCT_CM_AGE_SRCFMT_WPARAM). Diese Struktur enthält Informationen zu Quellen- und Zieltabellen, zur Art des aktuellen Aufrufs sowie Verweise auf eine Callback-Funktion und Arbeitsbereiche.

Das erste Feld in dieser Struktur ist das Feld **FuncCode**, das durch PST_CM_SRCFMT_TRANSFORM (PST_CMW_SRCFMT_TRANSFORM) oder PST_CM_SRCFMT_TERMINATE (PST_CMW_SRCFMT_TERMINATE) angegeben wird.

pSrcColList

Verweis auf PST_STRUCT_CM_EXIT_COL_LIST (PST_STRUCT_CM_WEXIT_COL_LIST). Diese Struktur beschreibt die Quellenspalten.

pDstColList

Verweis auf PST_STRUCT_CM_EXIT_COL_LIST (PST_STRUCT_CM_WEXIT_COL_LIST). Diese Struktur beschreibt die Zielspalten.

Callback-Funktion

Jedes Mal, wenn ein Optim-Prozess eine Quellenformatexitroutine aufruft, übergibt der Prozess die Adresse der folgenden Callback-Funktion:

pPSTGetColValue()

Ruft Daten für alle Quellenspalten und die meisten Zielspalten in der aktuellen Datenzeile ab. Im Allgemeinen muss die Exitroutine diese Funktion nicht aufrufen, weil die Daten für die Quellenspalte im ersten Parameter bereitgestellt werden. Die Exitroutine kann diese Funktion jedoch zum Abrufen der Daten für unterschiedliche Spalten aufrufen.

Verarbeitung

Eine typische Verarbeitung für die Quellenformatexitroutine wird in den folgenden Schritten zusammengefasst:

1. Bei jedem Aufruf durch Optim prüft die Exitroutine, ob es sich um einen erstmaligen Aufruf handelt. Beim ersten Aufruf führt die Exitroutine alle Initialisierungstasks sowie die normale Verarbeitung (Schritt 2) durch. Bei nachfolgenden Aufrufen wird nur noch die normale Verarbeitung (Schritt 2) durchgeführt.
2. Die Exitroutine empfängt den Wert der Quellenspalte wie in der in der Spaltenzuordnung definierten Alterungsfunktion angegeben. Optim wendet die Alterungsfunktion nicht vor dem Aufrufen der Exitroutine an und speichert den unformatierten Wert in einem der Felder in der Union-Verknüpfung **InputValue**, wie vom Feld **ValueType** in der Headerdatei angegeben.
3. Wenn der Exit andere Spalten untersuchen muss, um den Wert für die Zielspalte zu berechnen, muss er die Callback-Funktion **pPSTGetColValue()** aufrufen, um den Wert für diese Spalten abzurufen.
4. Nachdem der Zielwert generiert ist, muss die Exitroutine ihn formatieren und im Feld **OutputTimeStamp** oder im Feld **OutputSybDateTime** speichern. Optim überprüft diesen Wert und wendet die Alterungsfunktion an. Der Exit muss einen geeigneten Rückgabecode übergeben, der das Feld angibt, in dem die Daten gespeichert sind, oder der den Prozess anweist, die Datenzeile zu überspringen oder den Vorgang abubrechen.
5. Nachdem die letzte Datenzeile verarbeitet ist, übergibt Optim einen Beendigungsaufruf an die Exitroutine, der durch einen Wert **PST_CM_SRCFMT_TERMINATE** (**PST_CMW_SRCFMT_TERMINATE**) im Feld **FuncCode** gekennzeichnet ist. Dieser Anruf fordert die Exitroutine auf, dynamisch zugeordneten Speicher freizugeben. Wenn die Abschlusstasks beendet sind, übergibt die Exitroutine einen Rückgabecode an Optim.

Abbruchsmodi

Es gibt verschiedene Arten, wie die Quellenformatexitroutine die Verarbeitung abbrechen kann:

- Zeilen mit übersprungenen oder ungültigen Datumsangaben verarbeiten. Wenn Sie eine dieser Optionen in einer Prozessanforderung auswählen und Quellen- und Zielspalte dieselben Attribute haben, wird die Quellenspalte unverändert in die Zielspalte kopiert. Wenn Sie keine der Optionen auswählen, wird die Zeile zurückgewiesen.
- Die Zeile zurückweisen. Die Zeile unabhängig von den Verarbeitungsür übersprungene oder ungültige Datumsangaben basierend auf Spezifikationen in der Exitroutine zurückweisen.
- Den gesamten Prozess basierend auf Spezifikationen in der Exitroutine abbrechen.

Die Exitroutine übergibt einen Rückgabecode, der anzeigt, welches Datumsformat oder welcher Abbruchmodus verwendet werden soll.

Während des Prozesses kann die Exitroutine alle Spalten aus der Eingabezeile und einige Spalten aus der Zielzeile abfragen. Allerdings kann die Exitroutine keine Zielspalte abfragen, die eine Exitroutine einschließt und die in der Spaltenzuordnung nach der aktuellen Zielspalte definiert ist. Alle anderen Zielspalten sind verfügbar.

Rückgabecodes

Die folgenden Rückgabecodes sind für Quellenformatexits gültig:

```
PST_CM_SRCFMT_USE_TIMESTAMP  
(PST_CMW_SRCFMT_USE_TIMESTAMP)
```

Der Zielspalte wird ein Wert im Feld **OutputTimeStamp** des ersten an den Exit übergebenen Parameters zugeordnet.

```
PST_CM_SRCFMT_USE_SYB_DATETIME  
(PST_CMW_SRCFMT_USE_SYB_DATETIME)
```

Der Zielspalte wird ein Wert im Feld **OutputSybDateTime** des ersten an den Exit übergebenen Parameters zugeordnet.

```
PST_CM_SRCFMT_SKIP  
(PST_CMW_SRCFMT_SKIP)
```

Alterung wird nicht angewendet. Wenn Sie die Option zum Verarbeiten von Zeilen mit übersprungenen Datumsangaben nicht auswählen, wird die Zeile zurückgewiesen. Andernfalls werden die Daten an die Quelle kopiert, sofern Quelle und Ziel kompatibel sind. Sind diese nicht kompatibel, wird die Zeile zurückgewiesen.

```
PST_CM_SRCFMT_COL_INVALID  
(PST_CMW_SRCFMT_COL_INVALID)
```

Alterung wird nicht angewendet. Wenn Sie die Option zum Verarbeiten von Zeilen mit ungültigen Datumsangaben nicht auswählen, wird die Zeile zurückgewiesen. Andernfalls werden die Daten an die Quelle kopiert, sofern Quelle und Ziel kompatibel sind. Sind diese nicht kompatibel, wird die Zeile zurückgewiesen.

```
PST_CM_SRCFMT_REJECT_ROW  
(PST_CMW_SRCFMT_REJECT_ROW)
```

Der Quellenspalte kann kein Wert zugeordnet werden. Die Zeile wird zurückgewiesen (gelöscht).

```
PST_CM_SRCFMT_ABORT_PROCESS  
(PST_CMW_SRCFMT_ABORT_PROCESS)
```

Fataler Fehler. Beenden. Die Nachricht wird in den Arbeitsbereich gestellt und der nicht verwendete Bereich wird auf Leerzeichen oder NULL gesetzt, damit eine Fehlernachricht zurückgegeben wird.

Zielformatexit

Wenn Sie eine Zielformatexitroutine für Optim schreiben, geben Sie die Exitfunktion, PSTColMapAgeDstExit (PSTColMapAgeDstWExit), und folgende Parameter an:

Einzelbyte

```
PSTColMapAgeDstExit  
(PST_STRUCT_CM_AGE_DSTFMT_PARM * pInputParms,  
 PST_STRUCT_CM_EXIT_COL_LIST * pSrcCollist,  
 PST_STRUCT_CM_EXIT_COL_LIST * pDstCollist)
```

UTF-16

```
PSTColMapAgeDstWExit  
(PST_STRUCT_CM_AGE_DSTFMT_WPARAM * pInputParms,  
 PST_STRUCT_CM_WEXIT_COL_LIST * pSrcCollist,  
 PST_STRUCT_CM_WEXIT_COL_LIST * pDstCollist)
```

Parameter

Wenn ein Optim-Prozess eine Zielformatexitroutine aufruft, übergibt der Prozess folgende Parameter:

pInputParms

Verweis auf PST_STRUCT_CM_AGE_DSTFMT_PARM (PST_STRUCT_CM_AGE_DSTFMT_WPARAM). Diese Struktur enthält Informationen zu Quellen- und Zieltabellen und zur Art des aktuellen Aufrufs sowie Verweise auf eine Callback-Funktion und Arbeitsbereiche.

Das erste Feld in dieser Struktur ist das Feld **FuncCode**, angegeben durch:

PST_CM_DSTFMT_TO_CHAR (PST_CMW_DSTFMT_TO_WCHAR),
PST_CM_DSTFMT_TO_INTEGER (PST_CMW_DSTFMT_TO_INTEGER),
PST_CM_DSTFMT_TO_TIMESTAMP (PST_CMW_DSTFMT_TO_TIMESTAMP),
PST_CM_DSTFMT_TO_SYB_DATETIME (PST_CMW_DSTFMT_TO_SYB_DATETIME) oder PST_CM_DSTFMT_TERMINATE (PST_CMW_DSTFMT_TERMINATE).

pSrcColList

Verweis auf PST_STRUCT_CM_EXIT_COL_LIST (PST_STRUCT_CM_WEXIT_COL_LIST). Diese Struktur beschreibt die Quellenspalten.

pDstColList

Verweis auf PST_STRUCT_CM_EXIT_COL_LIST (PST_STRUCT_CM_WEXIT_COL_LIST). Diese Struktur beschreibt die Zielspalten.

Callback-Funktion

Jedes Mal, wenn ein Optim-Prozess eine Zielformatexitroutine aufruft, übergibt der Prozess die Adresse der folgenden Callback-Funktion:

pPSTGetColValue()

Ruft Daten für alle Quellenspalten und die meisten Zielspalten in der aktuellen Datenzeile ab. Im Allgemeinen muss die Exitroutine diese Funktion nicht aufrufen, weil die Daten für die als veraltet deklarierte Quellenspalte im ersten Parameter bereitgestellt werden. Die Exitroutine kann diese Funktion jedoch zum Abrufen der Daten für unterschiedliche Spalten aufrufen.

Formate

Das Eingabedatum liegt im Format PST_C_TIMESTAMP und im Format PST_C_SYB_DATETIME vor. Der Exit wird angewiesen, dieses Datum auf der Basis des Datentyps der Zielspalte in eines der folgenden Format umzusetzen.

PST_C_CHAR_SZ

CHAR- und VARCHAR-Zielspalten.

PST_C_INTEGER_CHAR_SZ

NUMERIC-Zielspalten.

PST_C_TIMESTAMP

DATE/TIME-Spalten für DB2 und Oracle.

PST_C_SYB_DATETIME

DATETIME-Spalten für Sybase ASE.

Verarbeitung

Eine typische Verarbeitung für die Zielformatexitroutine wird in den folgenden Schritten zusammengefasst:

1. Bei jedem Aufruf durch Optim prüft die Exitroutine, ob es sich um einen erstmaligen Aufruf handelt. Beim ersten Aufruf führt die Exitroutine alle Initialisierungstasks sowie die normale Verarbeitung (Schritt 2) durch. Bei nachfolgenden Aufrufen wird nur noch die normale Verarbeitung (Schritt 2) durchgeführt.
2. Die Exitroutine empfängt den Wert der Quellenspalte wie in der in der Spaltenzuordnung definierten Alterungsfunktion angegeben. Optim wendet die Alterungsfunktion vor dem Aufrufen der Spaltenzu-

ordnungs-Exitroutine an und speichert den nach Verweildauer gesteuerten Wert im Feld **InputTimeStamp** und im Feld **InputSybaDateTime** in der Headerdatei.

3. Wenn der Exit andere Spalten untersuchen muss, um den Wert für die Zielspalte zu berechnen, muss er die Callback-Funktion **pPSTGetColValue()** aufrufen, um den Wert für diese Spalten abzurufen.
4. Nachdem der Zielwert generiert ist, muss die Exitroutine den Wert formatieren und in einem der Felder in der Union-Verknüpfung **OutputValue** speichern. Das Feld **FuncCode** gibt das Feld in der Union-Verknüpfung **OutputValue** an, in dem der Wert gespeichert werden muss. Der Exit muss einen geeigneten Code zurückgeben, der das Feld angibt, in dem die Daten gespeichert sind, oder der den Prozess anweist, die Zeile zu überspringen oder den Vorgang abubrechen.
5. Nachdem die letzte Datenzeile verarbeitet ist, übergibt Optim einen Beendigungsaufruf an die Exitroutine, der durch einen Wert **PST_CM_DSTFMT_TERMINATE** (**PST_CMW_DSTFMT_TERMINATE**) im Feld **FuncCode** gekennzeichnet ist. Dieser Anruf fordert die Exitroutine auf, dynamisch zugeordneten Speicher freizugeben. Wenn die Abschlusstasks beendet sind, übergibt die Exitroutine einen Rückgabecode an Optim.

Abbruchsmodi

Es gibt verschiedene Arten, wie die Zielformatexitroutine die Verarbeitung abbrechen kann:

- Zeilen mit übersprungenen oder ungültigen Datumsangaben verarbeiten. Wenn Sie eine dieser Optionen in einer Prozessanforderung auswählen und Quellen- und Zielspalte dieselben Attribute haben, wird die Quellenspalte unverändert in die Zielspalte kopiert. Wenn Sie keine der Optionen auswählen, wird die Zeile zurückgewiesen.
- Die Zeile zurückweisen. Die Zeile unabhängig von den Verarbeitungsür übersprungene oder ungültige Datumsangaben basierend auf Spezifikationen in der Exitroutine zurückweisen.
- Den gesamten Prozess basierend auf Spezifikationen in der Exitroutine abbrechen.

Die Exitroutine übergibt einen Rückgabecode, der anzeigt, ob die Konvertierung erfolgreich war oder welcher Abbruchmodus verwendet werden soll.

Während des Prozesses kann die Exitroutine alle Spalten aus der Eingabezeile und einige Spalten aus der Zielzeile abfragen. Allerdings kann die Exitroutine keine Zielspalte abfragen, die eine Exitroutine einschließt und die in der Spaltenzuordnung nach der aktuellen Zielspalte definiert ist. Alle anderen Zielspalten sind verfügbar.

Rückgabecodes

Die folgenden Rückgabecodes sind für Zielformatexits gültig:

PST_CM_DSTFMT_SUCCESS
(**PST_CMW_DSTFMT_SUCCESS**)

Der Zielspalte wird ein Wert zugewiesen, wie dies im Feld **FuncCode** des ersten an den Exit übergebenen Parameters angegeben ist.

PST_CM_DSTFMT_SKIP
(**PST_CMW_DSTFMT_SKIP**)

Wenn Sie die Option zum Verarbeiten von Zeilen mit übersprungenen Datumsangaben nicht auswählen, wird die Zeile zurückgewiesen. Andernfalls werden die Quelldaten an das Ziel kopiert, sofern Quelle und Ziel kompatibel sind. Sind diese nicht kompatibel, wird die Zeile zurückgewiesen.

PST_CM_DSTFMT_COL_INVALID
(**PST_CMW_DSTFMT_COL_INVALID**)

Wenn Sie die Option zum Verarbeiten von Zeilen mit ungültigen Datumsangaben nicht auswählen, wird die Zeile zurückgewiesen. Andernfalls werden die Daten an die Quelle kopiert, sofern Quelle und Ziel kompatibel sind. Sind diese nicht kompatibel, wird die Zeile zurückgewiesen.

PST_CM_DSTFMT_REJECT_ROW
(PST_CMW_DSTFMT_REJECT_ROW)

Der Zielspalte kann kein Wert zugeordnet werden. Die Zeile wird zurückgewiesen (gelöscht).

PST_CM_DSTFMT_ABORT_PROCESS
(PST_CMW_DSTFMT_ABORT_PROCESS)

Fataler Fehler. Beenden. Die Nachricht wird in den Arbeitsbereich gestellt und der nicht verwendete Bereich wird auf Leerzeichen oder NULL gesetzt, damit eine Fehlernachricht zurückgegeben wird.

Spaltenzuordnungsprozeduren mit Lua-Scripting schreiben

Eine Spaltenzuordnungsprozedur ist eine Prozedur, mit der Daten in einer Spalte beim Ausführen eines Service maskiert oder umgesetzt werden. Wie der Name sagt, müssen Sie Spaltenzuordnungsprozeduren einer Spaltenzuordnung hinzufügen. Sie können Spaltenzuordnungsprozeduren mithilfe der Lua-Scripting-Sprache schreiben.

Informationen zur Programmiersprache Lua finden Sie auf der Lua-Website unter <http://www.lua.org>

Lua-Funktionen für Spaltenzuordnungsprozeduren

Spaltenzuordnungsprozeduren unterstützen die meisten Standard-Lua-Funktionen. Spaltenzuordnungsprozeduren unterstützen auch Optim-spezifische Funktionen.

Standardfunktionsnamen für benutzerdefinierte Funktionen

Verwenden Sie die folgenden Funktionsnamen in Ihren Lua-Spaltenzuordnungsprozeduren. Jede dieser Funktionen wird automatisch am angegebenen Punkt aufgerufen. Sie können auch Funktionen mit anderen Funktionsnamen erstellen und diese Funktionen in den Standardfunktionen aufrufen.

Name	Beschreibung	Erforderlich
cm_load()	Diese Funktion wird aufgerufen, bevor Tabellen verarbeitet werden.	Nein
cm_unload()	Diese Funktion wird aufgerufen, nachdem alle Tabellen verarbeitet wurden.	Nein
cm_starttable()	Diese Funktion wird zu Beginn der Verarbeitung für jede Tabelle aufgerufen.	Nein
cm_endtable()	Diese Funktion wird am Ende der Verarbeitung für jede Tabelle aufgerufen.	Nein
cm_transform()	Diese Funktion wird für jede verarbeitete Zeile aufgerufen.	Ja

Jede Spaltenzuordnungsprozedur muss eine Funktion `cm_transform()` enthalten. Wenn die Funktion `cm_transform()` nicht vorhanden ist, wird der Code als komplexer Ausdruck ausgewertet.

Globale Funktionen

Die folgenden Funktionen sind in allen Ausführungskontexten für Spaltenzuordnungsprozeduren verfügbar:

Name	Beschreibung
numparms()	Anzahl der Parameter abrufen, die an die Spaltenzuordnungsprozedur übergeben werden.
parms.get(<i>n</i>)	Wert des Parameters bei Index <i>n</i> abrufen.
print()	Nachrichten in den Prozessbericht ausgeben.
rejectrow()	Zeile überspringen und zur nächsten Zeile gehen.

Datenspeicherfunktionen

Mit folgenden Funktionen können Sie Informationen zu den Quellen- und Zieldatenspeichern abrufen:

Name	Beschreibung
source.getdbalias()	Den Datenbankaliasnamen des Quelldatenspeichers abrufen.
source.getcreatorid()	Die Ersteller-ID des Quelldatenspeichers abrufen.
target.getdbalias()	Den Datenbankaliasnamen des Zieldatenspeichers abrufen.
target.getcreatorid()	Die Ersteller-ID des Zieldatenspeichers abrufen.

Tabellenfunktionen

Mit folgenden Funktionen können Sie Informationen zu den Quellen- und Zieltabellen abrufen:

Name	Beschreibung
source.gettablename()	Den Namen der Quellentabelle abrufen.
target.gettablename()	Den Namen der Zieltabelle abrufen.

Spaltenfunktionen

Mit folgenden Funktionen können Sie Informationen zu den Quellen- und Zielspalten abrufen, Spaltendaten umsetzen, das Ergebnis in die Zielspalte schreiben und das Programm beenden:

Name	Beschreibung
source.column.getvalue()	Nicht numerischen Wert aus einer Quellenspalte abrufen. Ein Laufzeitfehler wird generiert, wenn Sie die Funktion source.column.getvalue() zum Abrufen von Werten aus Spalten mit dem Datentyp PST_SQL_BINARY verwenden.
source.column.getasdouble()	Numerischen Wert aus einer Quellenspalte abrufen (Spaltenzuordnungsprozeduren verarbeiten numerische Daten im Format mit doppelter Genauigkeit). Verwenden Sie diese Funktion für Spalten der folgenden Datentypen: PST_SQL_DECIMAL, PST_SQL_DOUBLE, PST_SQL_FLOAT, PST_SQL_INFX_DEC_FLOAT und PST_SQL_ORA_NUMBER.
source.column.getlength()	Länge der Zeichenfolge in einer Quellenspalte abrufen.
source.column.getname()	Name der Quellenspalte abrufen.
source.column.gettype()	Datentyp der Quellenspalte abrufen.

Name	Beschreibung
<code>target.column.setvalue()</code>	Wert einer Zielspalte festlegen. Ein Laufzeitfehler wird generiert, wenn Sie die Funktion <code>target.column.setvalue()</code> zum Festlegen von Werten in Spalten mit dem Datentyp <code>PST_SQL_BINARY</code> verwenden.
<code>target.column.getlength()</code>	Länge der Zeichenfolge in einer Zielspalte abrufen.
<code>target.column.getname()</code>	Name der Zielspalte abrufen.
<code>target.column.gettype()</code>	Datentyp der Zielspalte abrufen.
<code>optimmask()</code>	Einen Optim Data Privacy Providers (ODPP)-Anbieter aufrufen.
<code>userexit()</code>	Einen Benutzerexit aufrufen.

Nicht unterstützte Funktionen

Die folgenden Kategorien von Funktionen werden in Spaltenzuordnungsprozeduren nicht unterstützt.

- Eingabe- und Ausgabefunktionen, die in die Lua-Bibliothek `io` integriert sind
- `string.dump()`

Einschränkungen von Spaltenzuordnungsprozeduren

Beachten Sie beim Schreiben einer Spaltenzuordnungsprozedur folgende Einschränkungen.

Verarbeitung numerischer Daten im Format mit doppelter Genauigkeit

Spaltenzuordnungsprozeduren verarbeiten numerische Daten im Format mit doppelter Genauigkeit. Sie müssen daher numerische Daten mit der Funktion `source.column.getasdouble()` aus Quellenspalten abrufen.

Codierung

Spaltenzuordnungsprozeduren verwenden UTF-16-Codierung in ihrer internen Verarbeitung.

Datentyp `PST_SQL_BINARY` wird nicht unterstützt

Ein Laufzeitfehler wird generiert, wenn Sie die Funktion `source.column.getvalue()` zum Abrufen von Werten aus Spalten mit dem Datentyp `PST_SQL_BINARY` verwenden. Ein Laufzeitfehler wird ebenfalls generiert, wenn Sie die Funktion `target.column.setvalue()` zum Festlegen von Werten in Spalten mit dem Datentyp `PST_SQL_BINARY` verwenden.

Nicht unterstützte Funktionen

Die folgenden Kategorien von Funktionen werden in Spaltenzuordnungsprozeduren nicht unterstützt.

- Eingabe- und Ausgabefunktionen, die in die Lua-Bibliothek `io` integriert sind
- `string.dump()`

Beispiel für Spaltenzuordnungsprozedur: Generische Prozedur

Das Beispiel für eine Spaltenzuordnungsprozedur veranschaulicht die Struktur einer Spaltenzuordnungsprozedur mit ihren Standardfunktionen: `cm_load`, `cm_unload`, `cm_starttable`, `cm_endtable` und `cm_transform`.

```
-----
--
-- Beispiel für IBM Optim-Spaltenzuordnungsprozedur
```

```

--
-- Name:          OptimSample
--
-- Revision:     1.0
--
-- Beschreibung: Demonstriert alle Funktionen von Optim-/Lua-Spaltenzuordnungsprozeduren
--
-- Eingabe:      Null oder mehr Parameter, die einfach an den Optim-Prozessbericht
--              zurückgemeldet werden
--
--
-- Ausgabe:     Abschnitt im Optim-Prozessbericht, der Informationen aus dieser
--              Spaltenzuordnungsprozedur zeigt. Die Spalte selbst bleibt unverändert.
--
-----
-----
-- Funktion cm_load - Aufgerufen, bevor Tabellen verarbeitet werden
-----
function cm_load()

  print(" *** Start of Process ***")

  colinfoshown = false

  -- Aus der Spaltenzuordnung übergebene Parameter anzeigen

  print(" Argument Count: " .. string.format("%d", numparms()))
  for i = 1, numparms(), 1 do
    print("   Argument " .. string.format("%d", i-1) .. " " .. parms.get(i-1))
  end

end

-----
-- Funktion cm_unload - Aufgerufen, nachdem alle Tabellen verarbeitet wurden
-----
function cm_unload()

  print(" *** End of Process ***")

end

-----
-- Funktion cm_starttable - Am Anfang der Verarbeitung für jede Tabelle aufgerufen
-----
function cm_starttable()

  print(" \nStart of processing table")
  fullname = source.getdbalias() .. "." .. source.getcreatorid() .. "." .. source.gettablename()
  print("   Source Table: " .. fullname)
  fullname = target.getdbalias() .. "." .. target.getcreatorid() .. "." .. target.gettablename()
  print("   Target Table: " .. fullname)

end

-----
-- Funktion cm_endtable - Am Ende der Verarbeitung für jede Tabelle aufgerufen
-----
function cm_endtable()

  print(" \nEnd of processing table")

end

```

```

-----
-- Funktion cm_transform - Für jede verarbeitete Zeile aufgerufen
-----
function cm_transform()

  if (not colinfoshown) then
    colinfoshown = true
    print(" Processing column " .. source.column.getname())
    print("      Type: " .. source.column.gettype())
    print("      Length: " .. source.column.getlength())
  end

  -- Diese Anweisung ruft den Wert in der Spalte ab, für die cm_transform aufgerufen wurde
  -- Optional kann der Name einer anderen Spalte angegeben werden, z. B.
  -- gibt source.column.getvalue("COL1") den Wert in Spalte COL1 zurück
  oldvalue = source.column.getvalue()

  -- Dieser Code setzt diese Zielspalte auf denselben Wert wie die Quellenspalte
  -- Der Code zum Ändern des Werts würde hier eingefügt.
  -- Wenn Sie diese Zeile NICHT in die Zieltabelle einfügen wollen, rufen Sie die Funktion rejectrow() auf
  newvalue = oldvalue
  target.column.setvalue(newvalue)
end

```

Beispiel für Spaltenzuordnungsprozedur: Maskierungssuche

Das Beispiel für die Spaltenzuordnungsprozedur veranschaulicht, wie die Funktion mask_parms eine Spalte durch Verwendung von Daten aus einer Suchtabelle maskiert.

```

-----
--
-- Beispiel für IBM Optim-Spaltenzuordnungsprozedur
--
-- Name:          OptimSwitchedLookup
--
-- Revision:      1.0
--
-- Beschreibung:  Maskiert eine Spalte über die Tabellensuche. Die zu verwendende
--                Suchtabelle wird durch den Wert einer anderen Spalte
--                festgelegt.
--
-- Eingabe:       Parameter 1 (erforderlich):
--                Eine Zeichenfolge, die den zu verwendenden Suchtyp angibt:
--                HASH, RANDOM oder LOOKUP
--
--                Parameter 2 .... n-1 (erforderlich)
--                Ein Ausdruck, der die zu zu verwendende Suchtabelle angibt. Das
--                Format ist COND(Spaltenname=Wert) DATASOURCE(Datenquellenparameter).
--                Dieser Parameter kann mehrmals wiederholt werden. Wenn eine Zeile
--                keine der Anforderungen der Parameter COND erfüllt, wird sie nicht
--                in die Zieltabelle eingefügt.
--
--                Parameter n (optional):
--                Eine Zeichenfolge, die zusätzliche Parameter enthält, die
--                in den Aufruf von optimmask kopiert werden sollen.
--                Dies gilt zusätzlich zum Wert 'Datenquellenparameter' in der
--                COND-Klausel von Parameter 2 und dem Feld mask_parms_constant,
--                das am Anfang dieses Scripts deklariert ist.
--
--
-- Ausgabe:       - Maskierte Spaltendaten wie in der Funktion target.column.setvalue
--                angegeben
--                - Text, der von der Funktion 'print' an den Optim-Bericht
--                übertragen wird
--
--

```

```

-- Rückgabecodes: 0 - Erfolgreiche Ausführung
--                1 - Zurückweisung der Zeile (verwendet in cm_transform; Zeile wird
--                  nicht in die Zieltabelle eingefügt)
--                2 - Prozess abbrechen
--
--                Zwei Hilfeprogrammfunktionen sind angegeben, um Bedingungen zu
--                verarbeiten, bei denen der Rückgabecode ungleich 0 ist.
--                Es muss keine Rückkehranweisung codiert werden, wenn diese
--                Funktionen verwendet werden.
--                error(Zeichenfolge) - Dieser Aufruf bewirkt, dass der Optim-Prozess
--                abgebrochen und die Zeichenfolge im
--                Optim-Bericht als Fehlernachricht angezeigt wird.
--                rejectrow()        - Dieser Aufruf bewirkt, dass Optim die aktuell
--                verarbeitete Zeile zurückweist. Die Zeile wird
--                nicht in die Zieltabelle eingefügt.
--
--
--
-----

```

```

function cm_transform()

-- Dieses Feld so ändern, dass es Parameter enthält, die
-- in alle optimmask-Aufrufe aufgenommen werden sollen
local mask_parms_constant = 'CACHE=Y,WHENNF=PRE'

--
-- Anzahl von Parametern überprüfen:
--
nparm = numparms()
if (nparm < 2) then
  process_error("Call to column map procedure OptimSwitchedLookup must have 2 or more parameters")
end

--
-- Prozessparameter 1 (Typ der Suche)
parm = parms.get(0)
if (string.lower(parm) == "hash") then
  provalue = "HASH_LOOKUP"
elseif (string.lower(parm) == "random") then
  provalue = "RANDOM_LOOKUP"
elseif (string.lower(parm) == "lookup") then
  provalue = "LOOKUP"
else
  process_error("Invalid parameter. Expected HASH, RANDOM or LOOKUP. Found " .. parm)
end

--
-- Parameter COND/DATASOURCE
--
gotcond = false
for parmptr = 1, nparm-1 do
  parm = parms.get(parmptr)
  if (string.lower(string.sub(parm, 1, 5)) == "cond(") then
    gotcond = true
    datasource_parameters = process_cond()
    if (datasource_parameters > "") then
      break;
    end
  end
end
if (not gotcond) then
  process_error("No COND parameter found")
end

-- Keine COND-Bedingung traf für diese Zeile zu, daher wird die Zeile zurückgewiesen

```

```

if (datasource_parameters <= "") then
  rejectrow()
end

--
-- Optionale zusätzliche optimmask-Parameter verarbeiten
--
lastparm = parms.get(nparm-1)
if (string.lower(string.sub(lastparm, 1, 5)) == "cond(") then
  optimmask_additional_parms = ""
else
  optimmask_additional_parms = lastparm
end

--
-- Aufruf an optimmask erstellen und ausführen
-- und neue Werte in die Zielspalte schreiben
--
mask_parms = "PRO=" .. provalue .. "," .. mask_parms_constant

-- Diese Verwendung von Umgebungsvariablen zum Speichern von Benutzer-ID und Kennwort
-- für die Systemtabelle dient nur Darstellungszwecken.
-- Speichern Sie diese Informationen zur Sicherheit
-- in einem verschlüsseltes Format in Umgebungsvariablen.
userid = os.getenv("optimmaskuserid")
if (userid) then
  mask_parms = mask_parms .. ",USER=" .. userid
end
password = os.getenv("optimmaskpswd")
if (password) then
  mask_parms = mask_parms .. ",PASS=" .. password
end

mask_parms = mask_parms .. "," .. optimmask_additional_parms

oldvalue = source.column.getvalue()
newvalue = optimmask(oldvalue, mask_parms)
target.column.setvalue(newvalue)
end

function process_cond()

  strptr = 6    -- Auf erstes Zeichen nach "COND(" verweisen

  -- Spaltennamen abrufen
  equalsign = string.find(parm, "=", strptr, true)
  if (not equalsign) then
    process_error("Syntax error around character " .. strptr .. " in expression: " .. parm)
  end
  colname = string.sub(parm, strptr, equalsign-1)

  -- Spaltenwert abrufen
  strptr = equalsign + 1
  closeparen = string.find(parm, ")", strptr, true)
  if (not closeparen) then
    process_error("COND expression is missing closing parenthesis in expression: " .. parm)
  end
  colvalue = string.sub(parm, strptr, closeparen-1)

  --Debugging - Suchergebnisse drucken:
  --print ("Found colname=" .. colname .. " in parm: " .. parm)
  --print ("Found colvalue=" .. colvalue .. " in parm: " .. parm)

  -- Wenn die Bedingung COND(Spaltenname=Wert) nicht für diese Zeile erfüllt ist, keine
  -- weitere Verarbeitung für diesen Parameter. "" zurückgeben (kein Ergebnis).

```

```

if (source.column.getvalue(colname) ~= colvalue) then
    return ""
end

-- Übereinstimmung für COND gefunden, daher den Wert DATASOURCE abrufen
strptr = closeparen + 1
datasourceValuePtr = string.find(parm, "DATASOURCE(", strptr, true)
if (not datasourceValuePtr) then
    process_error("DATASOURCE clause not found in expression: " .. parm)
end
-- (Hinweis: Dieses Durchsuchen ist sehr einfach codiert. Es wird angenommen, dass
-- sich kein fehlgeleiteter Text zwischen den Klauseln COND und DATASOURCE befindet)
closeparen = string.find(parm, ")", datasourceValuePtr, true)
if (not closeparen) then
    process_error("COND expression is missing closing parenthesis in expression: " .. parm)
end
return string.sub(parm, datasourceValuePtr+11, closeparen-1)

end
function process_error(msg)
    errprefix = "Error in column map procedure for column " .. source.column.getname()
    errprefix = errprefix .. " in table " .. source.getdbalias() .. "."
    .. source.getcreatorid() .. "." .. source.gettablename()
    error(errprefix .. ":\n " .. msg)
end

```

Beispiel für Spaltenzuordnungsprozedur: Maskieren länderspezifischer IDs

Das Beispiel für die Spaltenzuordnungsprozedur veranschaulicht, wie die Funktion `mask_parms` eine Spalte maskiert, die eine länderspezifische ID enthält.

```

-----
--
-- Beispiel für IBM Optim-Spaltenzuordnungsprozedur
--
-- Name:          OptimNID
--
-- Revision:      1.0
--
-- Beschreibung:  Maskiert eine Spalte mit einer länderspezifischen ID auf der
--                Basis des Werts einer anderen Spalte oder eines Scriptarguments
--                mit dem Landescode.
--
-- Eingabe:       Parameter 1 (erforderlich):
--                Eine Zeichenfolge als Landescode (z. B. 'US' für 'United States')
--                --ODER--
--                COL(Spaltenname), wobei Spaltenname der Name einer Spalte ist,
--                die den Ländercode enthält
--
--                Parameter 2 (optional):
--                Eine Zeichenfolge, die zusätzliche Parameter enthält, die
--                in den optimmask-Aufruf kopiert werden sollen
--                Dies wird zusätzlich zum Feld mask_parms_constant verwendet, das
--                zu Beginn dieses Scripts deklariert ist.
--
--
-- Ausgabe:       - Maskierte Spaltendaten wie in der Funktion target.column.setvalue
--                angegeben
--                - Text, der von der Anzeigefunktion an den Optim-Bericht
--                übertragen wird
--
-- Rückgabecodes: 0 - Erfolgreiche Ausführung
--                1 - Zurückweisung der Zeile (verwendet in cm_transform; Zeile wird
--                nicht in die Zieltabelle eingefügt)
--                2 - Prozess abbrechen

```

```

--
--      Zwei Hilfeprogrammfunktionen sind angegeben, um Bedingungen zu
--      verarbeiten, bei denen der Rückgabecode ungleich 0 ist.
--      Es muss keine Rückkehranweisung codiert werden, wenn diese
--      Funktionen verwendet werden.
--      error(Zeichenf)- Dieser Aufruf bewirkt, dass der Optim-Prozess
--      abgebrochen wird und dass die Zeichenfolge im
--      Optim-Bericht als Fehlernachricht angezeigt wird.
--      rejectrow()   - Dieser Aufruf bewirkt, dass Optim die aktuell
--      verarbeitete Zeile zurückweist. Die Zeile wird
--      nicht in die Zieltabelle eingeführt.
--
--
-----

```

```

function cm_transform()

  -- Dieses Feld so ändern, dass es Parameter enthält, die
  -- in alle optimmask-Aufrufe aufgenommen werden sollen
  local mask_parms_constant = 'MTD=REPEATABLE'

  --
  -- Parameter abrufen:
  --
  nparm = numparms()
  if nparm ~= 1 and nparm ~= 2 then
    msg = "Call to column map procedure for column "
    msg = msg .. source.column.getname()
    msg = msg .. " must have 1 or 2 parameters"
    error(msg)
  end
  parm1 = parms.get(0)
  if (nparm == 2) then
    parm2 = parms.get(1)
  end

  --
  -- Landescode auf der Basis von Parametern in das Feld swivalue schreiben
  --
  if string.lower(string.sub(parm1, 1, 4)) == 'col(' then
    closeparen = string.find(parm1, ')', 5, true)
    if (not closeparen) then
      error("No closing parenthesis found in expression: " .. parm1)
    end
    colname = string.sub(parm1, 5, closeparen-1)
    swivalue = source.column.getvalue(colname)
  else
    swivalue = parm1
  end

  --
  -- Aufruf an optimmask erstellen und ausführen
  -- und neue Werte in die Zielspalte schreiben
  --
  mask_parms = 'PRO=NID, SWI=' .. swivalue .. ', ' .. mask_parms_constant
  if (parm2) then
    mask_parms = mask_parms .. ', ' .. parm2
  end
  oldvalue = source.column.getvalue()
  newvalue = optimmask(oldvalue, mask_parms)
  target.column.setvalue(newvalue)
end

```

Kapitel 6. Daten verwalten

Mit den Optim Designer-Dienstprogrammen können Sie Daten bearbeiten, anzeigen und vergleichen. Außerdem können Sie Tabellen erstellen.

Daten durchsuchen

Mit dem Dienstprogramm 'Browse' kann der Inhalt eines Dateidatenspeichers überprüft werden, ohne dass er in der Datenbank wiederhergestellt werden muss.

Mit der Hilfefunktion im Dienstprogramm 'Browse' können weitere Informationen zu einem Thema oder einer Funktion abgerufen werden. Sie können ein Hilfefenster durch Auswahl von **Help > Contents**, durch Klicken auf ein Element mit der rechten Maustaste und Auswahl von **What's This** oder durch Drücken von F1 öffnen.

Gehen Sie wie folgt vor, um Daten zu durchsuchen:

1. Klicken Sie in Optim Designer auf **Tools > Browse**. Das Fenster **Browse** wird geöffnet.
2. Klicken Sie auf **File > Open**, um das Fenster **Open** zu öffnen, und wählen Sie eine Datendatei aus, die von einem Service erstellt wurde. Die Dateien werden im Datenverzeichnis gespeichert, das im Dienstprogramm 'Configuration' angegeben wurde.
3. Klicken Sie auf **Open**. Die Tabellen in der Datendatei werden im Fenster **Browse** aufgelistet.
4. Klicken Sie auf **Tools > Create All Selected Objects**. Die Tabelle wird im Fenster **Browse Data** geöffnet.

Daten bearbeiten

Mit dem Tabelleneditor können Sie Gruppen relational intakter Daten in Datenbanktabellen anzeigen und bearbeiten. Der Editor kann ein beliebig komplexes Datenmodell mit einer beliebigen Anzahl von Tabellen und Beziehungen verarbeiten und garantiert eine referenziell intakte Datengruppe.

Mit der Hilfefunktion im Tabelleneditor können weitere Informationen zu einem Thema oder einer Funktion abgerufen werden. Sie können ein Hilfefenster durch Auswahl von **Help > Contents**, durch Klicken auf ein Element mit der rechten Maustaste und Auswahl von **What's This** oder durch Drücken von F1 öffnen.

Gehen Sie wie folgt vor, um Daten zu bearbeiten:

1. Klicken Sie in Optim Designer, auf **Tools > Edit**. Das Fenster **Edit Options** wird geöffnet.
2. Geben Sie einen Tabellennamen in **Table Name** oder eine benannte Zugriffsdefinition in **Named Access Definition** ein, wählen Sie Optionen zur ursprünglichen Anzeige unter **Initial Display** oder zum Modus unter **Mode** aus und klicken Sie auf **OK**. Das Tabelleneditorfenster wird geöffnet.
3. Bearbeiten Sie ein Tabelle oder mehrere Tabellen gleichzeitig.

Schreiben Sie durch Setzen des Zeigers auf eine andere Zeile oder über Menüoptionen Änderungen in der Datenbank fest. Jede Instanz einer Festschreibung zählt als eine Ebene der Rückgängigmachung. Eine Ebene der Rückgängigmachung ist als eine Änderung an einer Zeile definiert, die in der Datenbank festgeschrieben wird.

Wenn eine Fehlerbedingung beim Versuch auftritt, Daten in der Datenbank festzuschreiben, werden die Daten nicht festgeschrieben, aber der Versuch zählt dennoch als eine Ebene der Rückgängigmachung. Der Editor ermöglicht Ihnen das Wiederherstellen der von Ihnen geänderten Daten bis zu einem bestimmten Festschreibungspunkt.

Eine Abrufgruppe ist die Gruppe von Zeilen, die der Editor in einer einzelnen Tabelle in der Datenbank liest.

- Wenn Änderungen rückgängig gemacht und die ursprünglichen Versionen von Zeilen wiederhergestellt werden sollen, die in der aktuellen Abrufgruppe für alle Tabellen abgerufen wurden, wählen Sie **Tools > Undo** aus.
- Wenn Sie eine Änderung für eine Zeile rückgängig machen wollen, klicken Sie mit der rechten Maustaste auf diese und klicken Sie auf **Undo**.
- Wenn Sie jede aufeinanderfolgende Version der Zeile, die in der aktuellen Abrufgruppe festgeschrieben wurde, und die ursprüngliche Version der Zeile in der Abrufgruppe anzeigen wollen, klicken Sie mit der rechten Maustaste auf die Zeile und klicken Sie auf **Undo...**

Daten vergleichen

Mit dem Dienstprogramm 'Compare' können Sie Daten aus einer Gruppe von Quellentabellen mit Daten aus einer anderen vergleichen. Sie können eine Vergleichsanforderung definieren, die im Optim-Repository gespeichert wird, und die Ergebnisse eines Vergleichs in einer Vergleichsdatei speichern.

Vergleichsanforderung definieren

Mit dem Vergleichsanforderungseditor können Sie zu vergleichende Datenquellen und Verarbeitungsoptionen definieren.

Mit der Hilfsfunktion im Dienstprogramm 'Compare' können weitere Informationen zu einem Thema oder einer Funktion abgerufen werden. Sie können ein Hilfefenster durch Auswahl von **Help > Contents**, durch Klicken auf ein Element mit der rechten Maustaste und Auswahl von **What's This** oder durch Drücken von F1 öffnen.

Gehen Sie wie folgt vor, um eine Vergleichsanforderung zu definieren:

1. Klicken Sie im Repository-Explorer mit der rechten Maustaste auf **Compare** und klicken Sie auf **New Compare**. Der Editor für Vergleichsanforderungen wird geöffnet.
2. Führen Sie in der Registerkarte **General** die folgenden Schritte aus:
 - a. Geben den Namen einer Vergleichsdatei im Feld **Compare File** ein.
 - b. Wählen Sie **Browse Results Immediately** aus, um die Ergebnisse des Vergleichsprozesses anzuzeigen, nachdem der Prozess beendet ist.
 - c. Wählen Sie in **Comparison** einen Vergleichsmodus aus.

Einzelne Tabelle

Source File - Source File

Daten in einer einzelnen Tabelle in einer Quellendatei mit Daten in einer anderen Quellendatei vergleichen.

Source File - Database Table

Daten in einer einzelnen Tabelle in einer Quellendatei mit Daten in einer einzelnen Tabelle in der Datenbank vergleichen.

Database Table - Database Table

Daten in einer einzelnen Tabelle in einer Datenbank mit Daten in einer einzelnen Tabelle in einer anderen Datenbank vergleichen.

Mehrere Tabellen

Source File - Source File

Daten in Tabellen in einer Quellendatei mit Daten in einer anderen Quellendatei vergleichen.

Source File - Access Definition

Daten in Tabellen in einer Quellendatei mit in einer Zugriffsdefinition angegebenen Daten vergleichen.

Source File - All Database Tables

Daten in Tabellen in einer Quellendatei mit Daten in einer Datenbank vergleichen.

Anmerkung: Diese Auswahl ist sehr nützlich, wenn eine Datenbankanwendung getestet wird. Die Quellendatei stellt den ursprünglichen Zustand im Vergleich zu den Datenbanktabellen dar, die den späteren Zustand darstellen.

Access Definition - Access Definition

Die in einer Zugriffsdefinition angegebenen Daten mit in einer anderen Zugriffsdefinition angegebenen Daten vergleichen.

Access Definition - All Database Tables

Die in einer Zugriffsdefinition angegebenen Daten mit Daten in einer Datenbank vergleichen.

3. Wählen Sie in der Registerkarte **Source** die Datenquellen aus, die auf der Basis des Vergleichsmodus verglichen werden sollen.
4. Klicken Sie auf **File > Run**, um den Vergleichsprozess zu starten.
5. Klicken Sie auf **File > Save**, um die Vergleichsanforderung zu speichern. Geben Sie einen Anforderungsnamen im Format KENNUNG.NAME an.

Der Prozess erstellt eine Vergleichsdatei im Datenverzeichnis, das im Dienstprogramm 'Configuration' angegeben wurde. Mit dem Dienstprogramm 'Browse' können Sie die Ergebnisse des Vergleichs anzeigen.

Vergleichsanforderung bearbeiten

Mit dem Vergleichsanforderungseditor können Sie eine Vergleichsanforderung bearbeiten.

Mit der Hilfefunktion im Dienstprogramm 'Compare' können weitere Informationen zu einem Thema oder einer Funktion abgerufen werden. Sie können ein Hilfefenster durch Auswahl von **Help > Contents**, durch Klicken auf ein Element mit der rechten Maustaste und Auswahl von **What's This** oder durch Drücken von F1 öffnen.

Gehen Sie wie folgt vor, um eine Vergleichsanforderung zu bearbeiten:

1. Erweitern Sie im Repository-Explorer den Knoten **Compare**. Klicken Sie dann mit der rechten Maustaste auf die Anforderung und klicken Sie auf **Edit**, um sie zu bearbeiten. Der Editor für Vergleichsanforderungen wird geöffnet.
2. Bearbeiten Sie die Anforderung.
3. Klicken Sie auf **File > Save**, um die Vergleichsanforderung zu speichern.

Vergleichsprozess ausführen

Sie können einen Vergleichsprozess zum Generieren einer Vergleichsdatei ausführen, die Sie mit dem Dienstprogramm 'Browse' anzeigen können.

Gehen Sie wie folgt vor, um eine Vergleichsanforderung auszuführen:

Führen Sie die Anforderung mit einer der folgenden Methoden aus:

Wenn Sie den Prozess im Repository-Explorer ausführen wollen, klicken Sie mit der rechten Maustaste auf eine Vergleichsanforderung und klicken Sie auf **Run**.

Wenn Sie den Vergleichsprozess im Vergleichsanforderungseditor ausführen wollen, klicken Sie auf **File > Run**.

Der Prozess erstellt eine Vergleichsdatei im Datenverzeichnis, das im Dienstprogramm 'Configuration' angegeben wurde. Mit dem Dienstprogramm 'Browse' können Sie die Ergebnisse des Vergleichs anzeigen.

Tabellen erstellen

Mit dem Dienstprogramm 'Create' können Sie Objekte auf der Basis von Definitionen in einer Quellendatendatei erstellen.

Mit der Hilfsfunktion im Dienstprogramm 'Create' können weitere Informationen zu einem Thema oder einer Funktion abgerufen werden. Sie können ein Hilfefenster durch Auswahl von **Help > Contents**, durch Klicken auf ein Element mit der rechten Maustaste und Auswahl von **What's This** oder durch Drücken von F1 öffnen.

Gehen Sie wie folgt vor, um Tabellen zu erstellen:

1. Klicken Sie in Optim Designer auf **Tools > Create**. Das Fenster **Create Options** wird geöffnet.
2. Wählen Sie eine Quelldatendatei aus, die von einem Service erstellt wurde. Die Dateien werden im Datenverzeichnis gespeichert, das im Dienstprogramm 'Configuration' angegeben wurde.
3. Wählen Sie eine der Optionen unter **Table Map Options**.

Wenn Sie eine lokale Tabellenzuordnung auswählen, wird das Fenster **Local - Table Map Editor** geöffnet. Führen Sie die folgenden Schritte aus:

- a. Im Feld **Qualifier** müssen Sie einen Aliasnamen für einen Zieldatenspeicher und eine Ersteller-ID im Format `Datenspeicher.aliasname.Ersteller-ID` eingeben.
- b. Modifizieren Sie die Zieltabellennamen.
- c. Klicken Sie auf **File > Update and Return**.

Wenn Sie eine benannte Tabellenzuordnung auswählen, führen Sie den folgenden Schritt aus:

- a. Geben Sie im Feld **Table Map Name** den Namen einer Tabellenzuordnung ein oder wählen Sie einen solchen Namen aus.
4. Klicken Sie auf **OK**. Das Fenster **Create** wird geöffnet.
 5. Bearbeiten Sie die zu erstellenden Objekte und wählen Sie sie aus.
 6. Klicken Sie auf **Tools > Create All Selected Objects**, um den Erstellungsprozess zu starten.

Kapitel 7. Daten mit Optim Designer verwalten

Optim Designer stellt eine einzige Entwicklungsoberfläche für den Optim-Datenschutz und zum Testen von Datenverwaltungslösungen zur Verfügung.

Daten mit Optim Designer extrahieren

In diesem Lernprogramm erfahren Sie, wie Sie über Optim Designer einen Optim-Extraktionsservice erstellen können. In diesem Lernprogramm werden Sie eine Zugriffsdefinition und einen Extraktionsservice definieren.

Wenn Sie das Lernprogramm beendet haben, werden Sie in der Lage sein, einen Extraktionsservice auszuführen.

Lernziele

Nach der Durchführung der Übungen wissen Sie, wie die folgenden Tasks ausgeführt werden:

- Einen Repository-Explorer-Ordner erstellen
- Eine Zugriffsdefinition zum Definieren der zu extrahierenden Daten erstellen
- Eine Zugriffsdefinition zum Definieren von Auswahlkriterien bearbeiten
- Einen Extraktionsservice erstellen
- Einen Extraktionsservice testen

Erforderliche Zeit

Die Ausführung dieses Moduls dauert ungefähr 60 Minuten.

Voraussetzungen

Für dieses Lernprogramm wird eine Optim Designer-Umgebung benötigt mit einer Verbindung zu einem Optim-Repository und einem Datenspeicheraliasnamen, der die Optim-Beispieldaten enthält.

Dieses Lernprogramm kann in der Optim Designer-Umgebung ausgeführt werden.

Repository-Explorer-Ordner erstellen

In dieser Übung werden Sie einen Repository-Explorer-Ordner erstellen. Ein Repository-Explorer-Ordner enthält Services, Zugriffsdefinitionen, Spaltenzuordnungen und Tabellenzuordnungen. Verwenden Sie die Ordner, um diese Objekte zu organisieren.

Gehen Sie wie folgt vor, um einen Datenzugriffsplan und eine Auswahlrichtlinie zu erstellen:

1. Klicken Sie mit der rechten Maustaste auf die Repository-Exploreransicht und klicken Sie auf **Neu > Ordner**. Das Fenster **Neuer Ordner** wird geöffnet.
2. Geben Sie Tutorial in **Name** ein und klicken Sie auf **OK**.

Sie haben einen Repository-Explorer-Ordner mit dem Namen 'Tutorial' erstellt.

Zugriffsdefinition erstellen

In dieser Übung werden Sie eine Zugriffsdefinition erstellen. Mit Zugriffsdefinitionen können Sie die Tabellen, Beziehungstraversierung und Auswahlkriterien für die Daten, die Sie verarbeiten wollen, angeben.

Bevor Sie eine Zugriffsdefinition erstellen können, muss ein Datenspeicheraliasname für die Datenbank vorhanden sein, die die Starttabelle enthält.

Gehen Sie wie folgt vor, um eine Zugriffsdefinition zu erstellen:

1. Erweitern Sie den Lernprogrammordner im Repository-Explorer, klicken Sie mit der rechten Maustaste auf **Zugriffsdefinitionen** und klicken Sie auf **Neue Zugriffsdefinition**. Der Assistent **Neue Zugriffsdefinition** wird geöffnet.
2. Geben Sie **SAMPLE.AD** auf der Seite **Zugriffsdefinitionsnamen eingeben** ein. Klicken Sie auf **Weiter**.
3. Wählen Sie auf der Seite **Datenspeicheraliasnamen auswählen** den Datenspeicheraliasnamen aus, der die Optim-Beispieldaten enthält. Klicken Sie auf **Weiter**.
4. Gehen Sie auf der Seite **Starttabelle auswählen** wie folgt vor:
 - a. Geben Sie im Feld **Starttabelle** folgendes Suchmuster ein: *SCHEMA.OPTIM_CUSTOMERS*. Dabei ist *SCHEMA* das Schema, das die Optim-Beispieldaten enthält. Beispiel: *SAMPLE.OPTIM_CUSTOMERS*.
 - b. Klicken Sie auf **Tabellen suchen**. In der Tabellenliste werden die Tabellen angezeigt, die dem Muster entsprechen.
 - c. Wählen Sie die Tabelle *OPTIM_CUSTOMERS* aus.
 - d. Klicken Sie auf **Weiter**.
5. Klicken Sie auf der Seite **Tabellenauswahlmethode angeben** auf **Zusammengehörige Tabellen suchen**. Klicken Sie auf **Weiter**.
6. Führen Sie auf der Seite **Tabellenmuster eingeben und zusammengehörige Tabellen auswählen** die folgenden Schritte aus:
 - a. Klicken Sie auf **Tabellen suchen**. In der Tabellenliste werden Tabellen angezeigt, die zu *OPTIM_CUSTOMERS* gehören.
 - b. Klicken Sie auf **Alle auswählen** aus.
 - c. Klicken Sie auf **Fertig stellen**.

Die neue Zugriffsdefinition wird im Zugriffsdefinitionseditor angezeigt.

Auswahlkriterien definieren

In dieser Übung definieren Sie Auswahlkriterien in der Zugriffsdefinition. Nutzen Sie Auswahlkriterien, um den Fokus auf eine bestimmte Gruppe zusammengehöriger Daten zu richten, indem Sie eine SQL-Klausel **WHERE** definieren und Substitutionsvariablen mit Standardwerten verwenden.

Auswahlkriterien müssen der SQL-Syntax entsprechen und relationale oder logische Operatoren enthalten. Logische Operatoren und die Syntax variieren je nach Datenbankverwaltungssystem. Nähere Informationen finden Sie in der Dokumentation des entsprechenden Datenbankverwaltungssystems.

Zum Auswählen der gewünschten Datengruppe für eine Tabelle benötigen Sie möglicherweise eine Kombination aus den logischen Operatoren **AND** und **OR**.

Gehen Sie wie folgt vor, um Auswahlkriterien zu definieren:

1. Erweitern Sie den Lernprogrammordner im Repository-Explorer, erweitern Sie den Knoten **Zugriffsdefinitionen** und doppelklicken Sie auf die Zugriffsdefinition *SAMPLE.AD*. Der Zugriffsdefinitionseditor wird geöffnet.
2. Wählen Sie die Registerkarte **Tabellen** aus.
3. Wählen Sie die Tabelle *OPTIM_CUSTOMERS* in der Tabellenliste aus.
4. Klicken Sie auf **Auswahlkriterien bearbeiten**. Das Fenster **Auswahlkriterien bearbeiten** wird geöffnet und zeigt eine Klausel **WHERE** für die Tabelle.
5. Geben Sie folgende Kriterien ein: *COUNTRY_CODE='US'*.
Klicken Sie auf **Syntax prüfen**, um die Syntax zu prüfen und Fehler zu ermitteln.

6. Klicken Sie auf **OK**, um zum Zugriffsdefinitionseditor zurückzukehren.
Wenn die Syntax nicht gültig ist, wird eine entsprechende Nachricht angezeigt. Sie können die Kriterien nicht speichern, wenn sie Fehler enthalten.
7. Klicken Sie im Hauptmenü auf **Datei > Speichern**, um die Zugriffsdefinition zu speichern.

Sie haben Auswahlkriterien definiert, die nur Zeilen in der Tabelle OPTIM_CUSTOMERS auswählen, die in der Spalte COUNTRY_CODE den Wert 'US' aufweisen.

Extraktionsservice erstellen

In dieser Übung werden Sie einen Extraktionsservice auf der Basis der in einer Zugriffsdefinition angegebenen Daten erstellen. Sie können einen Extraktionsservice verwenden, um eine Gruppe zusammengehöriger Zeilen aus mindestens einer Tabelle zu kopieren und die Zeilen in einem Dateidatenspeicher zu speichern.

Gehen Sie wie folgt vor, um einen Extraktionsservice zu erstellen:

1. Erweitern Sie den Lernprogrammordner im Repository-Explorer, klicken Sie mit der rechten Maustaste auf **Services** und klicken Sie auf **Neuer Service**. Der Assistent **Neuer Service** wird geöffnet.
2. Führen Sie auf der Seite **Servicenamen eingeben und Servicetyp auswählen** die folgenden Schritte aus:
 - a. Geben Sie SAMPLE.EXTRACT im Feld **Name** ein.
 - b. Wählen Sie in der Liste von Servicetypen **Extrahieren** aus.
 - c. Klicken Sie auf **Weiter**.
3. Wählen Sie SAMPLE.AD auf der Seite **Zugriffsdefinition auswählen** aus. Klicken Sie auf **Weiter**.
4. Geben Sie SAMPLE_EXT auf der Seite **Namen für Zieldatenspeicher eingeben** in das Feld **Name für Zieldatenspeicher** ein.
5. Klicken Sie auf **Fertig stellen**.

Der neue Extraktionsservice wird im Extraktionsserviceeditor angezeigt. Sie können mit diesem Service einen Dateidatenspeicher für einen Konvertierungs-, Einfüge- oder Ladeservice erstellen. Sie können mit dem Dateidatenspeicher ebenfalls eine Spaltenzuordnung oder eine Tabellenzuordnung erstellen.

Daten mit Optim Designer maskieren

In diesem Lernprogramm erfahren Sie, wie Sie über Optim Designer einen Konvertierungsservice erstellen können, der Daten in einem Dateidatenspeicher umsetzt.

Für dieses Lernprogramm ist der Dateidatenspeicher SAMPLE_EXT erforderlich, der im Lernprogramm „Daten mit Optim Designer extrahieren“ auf Seite 125 erstellt wurde.

Wenn Sie das Lernprogramm beendet haben, werden Sie in der Lage sein, eine Datenschutzfunktion anzuwenden und einen Konvertierungsservice auszuführen.

Lernziele

Nach der Durchführung der Übungen wissen Sie, wie die folgenden Tasks ausgeführt werden:

- Einen Repository-Explorer-Ordner erstellen
- Eine Zugriffsdefinition zum Definieren der zu extrahierenden Daten erstellen
- Eine Zugriffsdefinition zum Definieren von Auswahlkriterien bearbeiten
- Einen Extraktionsservice erstellen
- Einen Extraktionsservice testen

Erforderliche Zeit

Die Ausführung dieses Moduls dauert ungefähr 60 Minuten.

Voraussetzungen

Für dieses Lernprogramm wird eine Optim Designer-Umgebung benötigt mit einer Verbindung zu einem Optim-Repository und einem Datenspeicheraliasnamen, der die Optim-Beispieldaten enthält.

Dieses Lernprogramm kann in der Optim Designer-Umgebung ausgeführt werden.

Tabellenzuordnung erstellen

In dieser Übung werden Sie eine Tabellenzuordnung erstellen. Mit einer Tabellenzuordnung können Sie Spezifikationen für das Korrelieren von Quellen- und Zieltabellen kompatibler Daten definieren. Sie können Tabellen zuordnen, die verschiedene Namen haben, Tabellennamen ändern, Tabellen aus einem Prozess ausschließen oder Spaltenzuordnungen für eine bessere Kontrolle über die Daten einschließen.

Gehen Sie wie folgt vor, um eine Tabellenzuordnung zu erstellen:

1. Erweitern Sie den Lernprogrammordner im Repository-Explorer, klicken Sie mit der rechten Maustaste auf **Tabellenzuordnungen** und klicken Sie auf **Neue Tabellenzuordnung**. Der Assistent **Neue Tabellenzuordnung** wird geöffnet.
2. Geben Sie auf der Seite **Tabellenzuordnungsnamen eingeben** SAMPLE.TMAP in das Feld **Name** ein. Klicken Sie auf **Weiter**.
3. Wählen Sie OPTIMUSER.EXT auf der Seite **Datenspeicheraliasnamen auswählen** aus.
4. Klicken Sie auf **Fertig stellen**.

Die neue Tabellenzuordnung wird im Tabellenzuordnungseditor angezeigt.

Sie müssen den Editor zum Definieren von Zieldatenspeichern und Schemata verwenden, bevor Sie die Tabellenzuordnung speichern.

Tabellenzuordnung bearbeiten

In dieser Übung verwenden Sie den Tabellenzuordnungseditor zum Definieren von Zieldatenspeichern und Schemas für Quellentabellen in einer Tabellenzuordnung.

Gehen Sie wie folgt vor, um Zieldaten in einer Tabellenzuordnung zu bearbeiten:

1. Erweitern Sie den Lernprogrammordner im Repository-Explorer, erweitern Sie den Knoten **Tabellenzuordnungen** und doppelklicken Sie auf die Tabellenzuordnung SAMPLE.TMAP. Der Tabellenzuordnungseditor wird geöffnet.
2. Wählen Sie die Registerkarte **Tabellenzuordnung** aus.
3. Wählen Sie im Bereich **Datenspeicheraliasname und Schemazuordnung** die Zeile aus, die den Quelldatenspeicher enthält.
4. Klicken Sie auf die Zelle **Aliasname für Zieldatenspeicher** und wählen Sie den Datenspeicheraliasnamen aus, der die Optim-Beispieldaten enthält.
5. Klicken Sie auf die Zelle **Zielschema** und wählen Sie das Schema aus, das die Optim-Beispieldaten enthält.
6. Klicken Sie im Hauptmenü auf **Datei > Speichern**, um die Tabellenzuordnung zu speichern.

Die Tabellenzuordnung verwendet dieselben Quellen- und Zieltabellen zum Maskieren von Quelldaten und um dasselbe Schema und den Datenspeicheraliasnamen beizubehalten.

Spaltenzuordnung erstellen

In dieser Übung werden Sie eine Spaltenzuordnung erstellen. Eine Spaltenzuordnung stellt Spezifikationen bereit, die zum Abgleichen von Spalten oder zum Ausschließen von Spalten aus der Verarbeitung in einem Datenverwaltungsservice erforderlich sind. Konvertierungs-, Einfüge- und Ladeservices müssen eine Tabellenspalte referenzieren, die mindestens eine Spaltenzuordnung referenzieren kann. Sie können mit einer Spaltenzuordnung Datenumsetzungen mit Funktionen oder Spaltenzuordnungsprozeduren definieren.

Gehen Sie wie folgt vor, um eine Spaltenzuordnung zu erstellen:

1. Erweitern Sie den Lernprogrammordner im Repository-Explorer, klicken Sie mit der rechten Maustaste auf **Spaltenzuordnungen** und klicken Sie auf **Neue Spaltenzuordnung**. Der Assistent **Neue Spaltenzuordnung** wird geöffnet.
2. Geben Sie auf der Seite **Spaltenzuordnungsnamen eingeben** SAMPLE.CMAP in das Feld **Name** ein. Klicken Sie auf **Weiter**.
3. Wählen Sie SAMPLE_EXT auf der Seite **Aliasnamen für Quellendateidatenspeicher auswählen** aus und klicken Sie auf **Weiter**.
4. Gehen Sie auf der Seite **Quellentabelle auswählen** wie folgt vor:
 - a. Geben Sie im Feld **Tabellensuchmuster** folgendes Suchmuster ein: `DATA_STORE_ALIAS.SCHEMA.OPTIM_CUSTOMERS`. Dabei ist `DATA_STORE_ALIAS.SCHEMA` der Datenspeicheraliasname und das Schema, die die Optim-Beispieldaten enthalten. Beispiel: `OPTIM.SAMPLE.OPTIM_CUSTOMERS`.
 - b. Klicken Sie auf **Tabellen suchen**. In der Tabellenliste werden die Tabellen angezeigt, die dem Muster entsprechen.
 - c. Wählen Sie die Tabelle `OPTIM_CUSTOMERS` aus.
 - d. Klicken Sie auf **Weiter**.
5. Wählen Sie auf der Seite **Aliasnamen für Zieldatenspeicher auswählen** den Datenspeicheraliasnamen aus, der die Optim-Beispieldaten enthält. Klicken Sie auf **Weiter**.
6. Gehen Sie auf der Seite **Zieltabelle auswählen** wie folgt vor:
 - a. Geben Sie im Feld **Tabellensuchmuster** folgendes Suchmuster ein: `SCHEMA.OPTIM_CUSTOMERS`. Dabei ist `SCHEMA` das Schema, das die Optim-Beispieldaten enthält. Beispiel: `SAMPLE.OPTIM_CUSTOMERS`.
 - b. Klicken Sie auf **Tabellen suchen**. In der Tabellenliste werden die Tabellen angezeigt, die dem Muster entsprechen.
 - c. Wählen Sie die Tabelle `OPTIM_CUSTOMERS` aus.
7. Klicken Sie auf **Fertig stellen**.

Die neue Spaltenzuordnung wird im Spaltenzuordnungseditor angezeigt.

Datenmaskierungsfunktion anwenden

In dieser Übung werden Sie eine Datenmaskierungsfunktion auf eine Spalte in einer Spaltenzuordnung anwenden. Sie können Daten wie länderspezifische ID-Nummern, Kreditkartennummern, Datumsangaben, numerische Werte und personenbezogene Daten maskieren.

Gehen Sie wie folgt vor, um eine Datenmaskierungsfunktion anzuwenden:

1. Erweitern Sie den Lernprogrammordner im Repository-Explorer, erweitern Sie den Knoten **Spaltenzuordnungen** und doppelklicken Sie auf die Spaltenzuordnung SAMPLE.CMAP. Der Spaltenzuordnungseditor wird geöffnet.
2. Wählen Sie die Spalte `PHONE_NUMBER` aus.
3. Klicken Sie auf **Funktion anwenden**. Das Fenster **Funktion anwenden** wird geöffnet.
4. Wählen Sie die Funktion **Mischen** aus. Klicken Sie auf **OK**. Der Funktionsname wird in der Spalte **Quellenspalte** angezeigt und der Funktionseditor wird im Spaltenzuordnungseditor geöffnet.
5. Geben Sie im Feld **Spaltenzuordnungsausdruck** des Funktionseditors `SHUFFLE(RETRY=12)` ein.

6. Klicken Sie im Hauptmenü auf **Datei > Speichern**, um die Spaltenzuordnung zu speichern.

Sie haben die Funktion **Mischen** auf die Spalte PHONE_NUMBER angewendet. Die Funktion maskiert Daten über das Ersetzen von Werten in der Spalte durch andere Werte in der Spalte. Die Funktion sucht bis zu 12 Mal nach einem Ersatzwert, der nicht mit dem Quellenwert übereinstimmt.

Spaltenzuordnung einer Tabellenzuordnung hinzufügen

In dieser Übung werden Sie eine Spaltenzuordnung einer Tabellenzuordnung hinzufügen. Eine Tabellenzuordnung ist für einen Konvertierungsservice erforderlich. Mit der zugehörigen Spaltenzuordnung ermöglichen Sie es dem Konvertierungsservice, die in der Spaltenzuordnung definierte Datenmaskierungsfunktion auszuführen.

Gehen Sie wie folgt vor, um eine Spaltenzuordnung einer Tabellenzuordnung hinzuzufügen:

1. Erweitern Sie den Lernprogrammordner im Repository-Explorer, erweitern Sie den Knoten **Tabellenzuordnungen** und doppelklicken Sie auf die Tabellenzuordnung SAMPLE.TMAP. Der Tabellenzuordnungseditor wird geöffnet.
2. Wählen Sie die Registerkarte **Tabellenzuordnung** aus.
3. Wählen Sie im Bereich **Tabellenzuordnung** die Tabelle OPTIM_CUSTOMERS aus.
4. Klicken Sie auf **Spaltenzuordnung hinzufügen**. Das Fenster **Neue Spaltenzuordnung** wird mit einer Liste der Spaltenzuordnungen geöffnet, die die ausgewählten Tabellen enthalten.
5. Wählen Sie die Spaltenzuordnung SAMPLE.CMAP aus. Klicken Sie auf **OK**.
6. Klicken Sie im Hauptmenü auf **Datei > Speichern**, um die Tabellenzuordnung zu speichern.

Sie haben der Tabellenzuordnung SAMPLE.TMAP die Spaltenzuordnung SAMPLE.CMAP (und ihre zugehörige Datenmaskierungsfunktion) hinzugefügt, wodurch ein Konvertierungsservice in die Lage versetzt wird, Daten umzusetzen.

Konvertierungsservice erstellen

In dieser Übung werden Sie einen Konvertierungsservice zum Maskieren von Daten erstellen.

Gehen Sie wie folgt vor, um einen Konvertierungsservice zu erstellen:

1. Erweitern Sie den Lernprogrammordner im Repository-Explorer, klicken Sie mit der rechten Maustaste auf **Services** und klicken Sie auf **Neuer Service**. Der Assistent **Neuer Service** wird geöffnet.
2. Führen Sie auf der Seite **Servicenamen eingeben und Servicetyp auswählen** die folgenden Schritte aus:
 - a. Geben Sie SAMPLE.CONVERT in das Feld **Name** ein.
 - b. Wählen Sie in der Liste von Servicetypen **Konvertieren** aus.
 - c. Klicken Sie auf **Weiter**.
3. Wählen Sie auf der Seite **Tabellenzuordnung auswählen** die Tabellenzuordnung SAMPLE.TMAP aus. Klicken Sie auf **Weiter**.
4. Geben Sie SAMPLE_CONV auf der Seite **Zieleigenschaften eingeben** im Feld **Name für Zieldateidatenspeicher** ein.
5. Klicken Sie auf **Fertig stellen**.

Der neue Konvertierungsservice wird im Editor für den Konvertierungsservice angezeigt. Sie können mit diesem Service die Spalte PHONE_NUMBER in der Tabelle OPTIM_CUSTOMERS maskieren.

Bemerkungen

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden.

Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim zuständigen IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Services von IBM verwendet werden können. An Stelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte der IBM verletzen. Die Verantwortung für den Betrieb von Produkten, Programmen und Services anderer Anbieter liegt beim Kunden.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Director of Licensing
IBM Europe, Middle East & Africa
Tour Descartes
2, avenue Gambetta
92066 Paris La Defense
France

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die hier enthaltenen Informationen werden in regelmäßigen Zeitabständen aktualisiert und als Neuausgabe veröffentlicht. IBM kann ohne weitere Mitteilung jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter werden lediglich als Service für den Kunden bereitgestellt und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Websites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängig voneinander erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Corporation
Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
USA

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des in diesen Informationen beschriebenen Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung bzw. der Allgemeinen Geschäftsbedingungen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer kontrollierten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Gewährleistung, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können davon abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten von IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele von IBM.

Alle von IBM angegebenen Preise sind empfohlene Richtpreise und können jederzeit ohne weitere Mitteilung geändert werden. Händlerpreise können u. U. von den hier genannten Preisen abweichen.

Diese Veröffentlichung dient nur zu Planungszwecken. Die in dieser Veröffentlichung enthaltenen Informationen können geändert werden, bevor die beschriebenen Produkte verfügbar sind.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufs. Sie sollen nur die Funktionen des Lizenzprogramms illustrieren und können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden; Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

Marken

Folgende Namen sind Marken der IBM Corporation in den USA und/oder anderen Ländern:

IBM
IBM Logo
DB2

AIX
Informix
InfoSphere
Optim

Netezza ist eine eingetragene Marke der IBM International Group B.V., einem IBM Unternehmen.

Adobe, Acrobat, PostScript und alle auf Adobe basierenden Marken sind Marken oder eingetragene Marken der Adobe Systems Incorporated in den USA und/oder anderen Ländern.

Apache Derby ist eine Marke von The Apache Software Foundation.

Eclipse ist eine Marke der Eclipse Foundation, Inc.

Microsoft, Windows, Windows NT und das Windows-Logo sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

Java und alle auf Java basierenden Marken und Logos sind Marken von Sun Microsystems, Inc. in den USA und/oder anderen Ländern.

Linux ist eine Marke von Linus Torvalds in den USA und/oder anderen Ländern.

UNIX ist eine eingetragene Marke von The Open Group in den USA und anderen Ländern.

Weitere Unternehmens-, Produkt- oder Servicenamen können Marken anderer Hersteller sein.

Index

A

- Alterungsfunktion 96
 - inkrementelle Alterung 97
 - Quellenformatexit 103
 - semantische Alterung 97
 - Zielformatexit 103
- Assistent 'Neuer Service' 41, 45, 46, 50
- Auswahlkriterien
 - Auswahlkriterien bearbeiten, Fenster 22
 - Beziehungen verwalten 23
 - Beziehungstraversierung 23
 - Datengruppierung 23
 - Definition 22
 - Optionen für Datenstichproben und Zeilenbeschränkungen 23
 - Point-and-shoot-Liste 26
 - Tabellenzugriffsoptionen 24
 - Übersicht 22
 - Variablen 25
- Auswahlkriterien bearbeiten, Fenster 22
- Automatisch generierter E-Mail-Name, Funktion 102

B

- Beispieldaten
 - Datenschutztabellen 11
 - Datenschutztabellen erstellen 12
 - Optim-Beispieldaten 5
 - Optim-Beispieltabellen erstellen 12
 - OPTIM_CUSTOMERS, Tabelle 6
 - OPTIM_DETAILS, Tabelle 8
 - OPTIM_FEMALE_RATES, Tabelle 11
 - OPTIM_ITEMS, Tabelle 8
 - OPTIM_MALE_RATES, Tabelle 10
 - OPTIM_ORDERS, Tabelle 7
 - OPTIM_SALES, Tabelle 5
 - OPTIM_SHIP_INSTR, Tabelle 10
 - OPTIM_SHIP_TO, Tabelle 9
 - OPTIM_STATE_LOOKUP, Tabelle 11
 - Übersicht 4
- Beziehungseditor 30, 31
 - Spaltenausdruck hinzufügen, Fenster 30
 - Übergeordnete Spalte auswählen, Fenster 30, 31
 - Untergeordnete Spalte auswählen, Fenster 30, 31
- Browse, Dienstprogramm 121

C

- Codierung
 - Spaltenzuordnungsprozeduren 113
- Compare, Dienstprogramm
 - Übersicht 122
- Create, Dienstprogramm 124

D

- Daten bearbeiten 121
- Daten durchsuchen 121
- Daten maskieren 53
- Daten vergleichen 122
- Daten verwalten
 - Browse, Dienstprogramm 121
 - Compare, Dienstprogramm 122
 - Create, Dienstprogramm 124
 - Daten bearbeiten 121
 - Daten durchsuchen 121
 - Daten vergleichen 122
 - Optim Designer-Dienstprogramme 121
 - Tabellen erstellen 124
 - Tabelleneditor 121
 - Vergleichsanforderung ausführen 123
 - Vergleichsanforderung bearbeiten 123
 - Vergleichsanforderung definieren 122
 - Vergleichsanforderungseditor 122, 123
- Datenmaskierung
 - Alterungsfunktion 96
 - automatisch generierter E-Mail-Name, Funktion 102
 - boolesche Konstante 95
 - Datenmaskierungsfunktion anwenden 39, 53
 - formatierter E-Mail-Name 102
 - Funktion für fortlaufende Zahlen 102
 - Funktion für Hashsuche 56
 - Funktion für Zufallssuche 60
 - Funktion für Zufallszahlen 102
 - Funktion TRANS CCN 68
 - Funktion TRANS COL 75
 - Funktion TRANS EML 71
 - Funktionen 53
 - hexadezimaler Literal 95
 - Identitätsfunktion 91
 - Literal- und Wertfunktionen 95
 - Literal für Datum/Zeit 95
 - Mischfunktion 62
 - NULL 95
 - numerische Ausdrücke 95
 - numerische Konstante 95
 - Oracle-Sequenzfunktion 92
 - Parameter in Spaltenzuordnungsprozedur bearbeiten 40
 - sequenzielle Funktion 90
 - serielle Funktion 91
 - Sonderregister 95
 - Spaltenzuordnungsprozedur bearbeiten 40
 - Spaltenzuordnungsprozedur erstellen 39
 - Suchfunktion 53
 - Suchfunktionen 53
 - TRANS NID, Funktion 77
 - britische Sozialversicherungsnummern 86

Datenmaskierung (Forts.)

- TRANS NID, Funktion (Forts.)
 - INSEE, Frankreich 81
 - italienische Steuernummern 83
 - kanadische Sozialversicherungsnummern 79
 - spanische Steuernummern 85
 - US-amerikanische Sozialversicherungsnummern 88
 - TRANS SSN, Funktion 64
 - Übersicht 53
 - Unterzeichenfolgefunktion 89
 - verkettete Ausdrücke 94
 - Währungsfunktion 99
 - Weitergabefunktion 93
 - Zeichenfolgeliteral 95
 - Zufallsfunktion 90
- ## Datenmodelle
- Übersicht 19
 - Zugriffsdefinitionen 19
- ## Datenspeicheraliasname
- bearbeiten 15
 - Definition 15
 - Neuer Datenspeicheraliasname, Assistent 15
 - Übersicht 15
 - Verbindung herstellen 16
 - Verbindungseigenschaften, Assistent für 15
- ## Datenverwaltungsservices
- Einfügeservice 46
 - Extraktionsservice 41
 - Konvertierungsservice 44
 - Ladeservice 48
 - Optim Manager (integrierter Modus) 52
 - testen 52
 - Übersicht 41

E

- Einfügeservice
 - Assistent 'Neuer Service' 46
 - bearbeiten 46
 - Einfügeserviceeditor 46, 48
 - erstellen 46
 - Tabellenzuordnung ändern 48
 - Übersicht 46
 - Verarbeitungsoptionen 46
- Einfügeserviceeditor 46
 - Tabellenzuordnung ändern, Assistent 48
- Verarbeitungsoptionen, Registerkarte 46
- Exitroutinen
 - Beispielheaderdateien 103
 - DLL-Dateien verwenden 104
 - Quellenformatexit
 - Abbruchsmodi 107
 - Callback-Funktion 107

Exitroutinen (*Forts.*)

Quellenformatexit (*Forts.*)

Eingabe für die Alterungsfunktion 103

Funktion 106

Parameter 106

Rückgabecodes 108

Verarbeitung 107

schreiben 103

Standardexit 104

Callback-Funktionen 105

Parameter 105

Rückgabecodes 106

Verarbeitung 105

Übersicht 102

Voraussetzungen 104

Zielformatexit

Abbruchsmodi 110

Callback-Funktion 109

Eingabe für die Alterungsfunktion 103

Formate 109

Funktion 108

Parameter 108

Rückgabecodes 110

Verarbeitung 109

Extraktionservice

Assistent 'Neuer Service' 41

bearbeiten 42

Dateikomprimierungsoptionen 43
erstellen 41

Extraktionsserviceeditor 42, 43, 44

Konvertierungsoptionen 42

Optionen für Objekte und Gruppierung 42

Übersicht 41

Variablenoptionen 44

Verarbeitungsoptionen 42

Zugriffsdefinition ändern 44

Extraktionsserviceeditor 42

Dateikomprimierungsoptionen, Registerkarte 43

Daten und Objekte, Registerkarte 42

Konvertierung, Registerkarte 42

Serviceeigenschaften, Registerkarte 42

Variablen, Registerkarte 44

Zugriffsdefinition ändern, Assistent 44

F

Formatierter E-Mail-Name, Funktion 102

Funktion anwenden, Fenster 39, 53

Funktion für fortlaufende Zahlen 102

Funktion für Hashsuche 56

Funktion für Zufallsuche 60

Funktion für Zufallszahlen 102

I

Identitätsfunktion 91

K

Konvertierungsservice

Assistent 'Neuer Service' 45

bearbeiten 45

erstellen 45

Konvertierungsserviceeditor 45

Tabellenzuordnung ändern 45

Übersicht 44

Verarbeitungsoptionen 45

Konvertierungsserviceeditor 45

Tabellenzuordnung ändern, Assistent 45

Verarbeitungsoptionen, Registerkarte 45

L

Ladeservice

Assistent 'Neuer Service' 50

Ausgabedateien 49

bearbeiten 50

Datenbankverwaltungssystem, Optionen für das Ladeprogramm 50
erstellen 50

Ladeserviceeditor 50, 51

Tabellenzuordnung ändern 51

Übersicht 48

Verarbeitungsoptionen 50

Ladeserviceeditor 50

Ladeoptionen, Registerkarte 50

Tabellenzuordnung ändern, Assistent 51

Verarbeitungsoptionen, Registerkarte 50

Länderspezifische ID maskieren

Beispiel 118

Literal- und Wertfunktionen 95

Lua-Scripteditor 39, 40

M

Mischfunktion 62, 64

Modelldaten

Übersicht 19

Zugriffsdefinitionen 19

N

Neue Optim-Beziehung, Assistent 30

Neue Point-and-shoot-Datei, Fenster 26

Neue Repository-Verbindung, Fenster 16

Neue Spaltenzuordnung, Assistent 38

Neue Spaltenzuordnung, Fenster 35

Neue Tabellenzuordnung, Assistent 34

Neue Zugriffsdefinition, Assistent 20

Neuer Datenspeicheraliasname, Assistent 15

Neuer Primärschlüssel, Assistent 32, 33

Numerische Ausdrücke 95

O

Optim-Beziehungen

bearbeiten 30

Beziehungseditor 30

Optim-Beziehungen (*Forts.*)

Datenkompatibilität 28

Einschränkungen 27

erstellen 30

erweiterte 27

explizite 27

generische 27

generische Beziehung 31

Neue Optim-Beziehung, Assistent 30

Spalten bearbeiten 31

Spaltenausdrücke 30

Spaltenreihenfolge 31

Übersicht 26

Optim Designer

Datenbankunterstützung 13

Datenbankverbindungen 15

Erste Schritte 1

Funktionen zur behindertengerechten Bedienung 13

Optim-Perspektive 1

Repository-Explorer 2

Übersicht 1

Verzeichnisexplorer 3

Optim Designer-Dienstprogramme

Browse, Dienstprogramm 121

Compare, Dienstprogramm 122

Create, Dienstprogramm 124

Daten bearbeiten 121

Daten durchsuchen 121

Daten vergleichen 122

Tabellen erstellen 124

Tabelleneditor 121

Übersicht 121

Vergleichsanforderung ausführen 123

Vergleichsanforderung bearbeiten 123

Vergleichsanforderung definieren 122

Vergleichsanforderungseditor 122, 123

Optim Manager (integrierter Modus) 4, 52

Optim-Perspektive

Repository-Explorer 2

Übersicht 1

Verzeichnisexplorer 3

Optim-Primärschlüssel

bearbeiten 33

erstellen 32

explizite 32

generische 32, 33

Neuer Primärschlüssel, Assistent 32, 33

Primärschlüsseleditor 33

Schlüsselspalten auswählen 33

Übersicht 32

Wert weitergeben 93

Optim-Repository

Anforderung in Service umsetzen 17

Neue Repository-Verbindung, Fenster 16

Optim-Verzeichnisverbindung 17

Repository-Verbindung bearbeiten, Fenster 16

Repository wechseln, Fenster 16

Übersicht 16

Verbindung ändern 16

Verbindung bearbeiten 16

Optim-Repository (*Forts.*)
Verbindung herstellen 16
Optim-Verzeichnis
Anforderung in Service umsetzen 17
Umsetzungsanforderung an Optim-Serviceassistent 17
Verbindung herstellen 17
Verzeichnis zuordnen, Fenster 17
Oracle-Sequenzfunktion 92

P

Point-and-shoot-Liste
auswählen 26
erstellen 26
Neue Point-and-shoot-Datei, Fenster 26
Point-and-shoot, Registerkarte 26
Übersicht 26
Primärschlüsseleditor 33
Primärschlüsselwert weitergeben 93

R

Repository-Explorer
Repository-Explorer-Ordner 3
Übersicht 2
Repository-Explorer-Ordner 3
Repository-Verbindung bearbeiten, Fenster 16
Repository wechseln, Fenster 16

S

Sequenzielle Funktion 90
Serielle Funktion 91
Service ausführen, Fenster 52
Services testen 52
Spaltenzuordnungen
bearbeiten 39
Datenkompatibilitätsregeln 37
Datenmaskierungsfunktion anwenden 39, 53
Datentypen 37
erstellen 38
Exitroutinen 102
Funktion anwenden, Fenster 39, 53
Lua-Scripteditor 39, 40
Neue Spaltenzuordnung, Assistent 38
Parameter in Spaltenzuordnungsprozedur bearbeiten 40
Quellenspalte zuordnen 39
Spaltenzuordnungseditor 39
Spaltenzuordnungsprozedur 39
Spaltenzuordnungsprozedur bearbeiten 40
Spaltenzuordnungsprozedur erstellen 39
Übersicht 36
zu Tabellenzuordnungen hinzufügen 35
Spaltenzuordnungseditor 39
Funktion anwenden, Fenster 39, 53
Lua-Scripteditor 39, 40
Spaltenzuordnungsprozedur 111

Spaltenzuordnungsprozedur (*Forts.*)
bearbeiten 40
Beispiel 113, 115, 118
Codierung 113
Einschränkungen 113
erstellen 39
Funktionen 111
numerisches Datenformat 113
Parameter bearbeiten 40
reservierte Namen 111
Spaltenzuordnungen 39
Suchfunktion 53
Suchfunktionen 53
Switchsuche
Beispiel 115

T

Tabellen hinzufügen, Assistent 20
Tabelleneditor 121
Tabellezuordnungen
bearbeiten 34
erstellen 34
Neue Tabellezuordnung, Assistent 34
Spaltenzuordnung hinzufügen 35
Spaltenzuordnungen 35
Standardzieltaten 35
Tabellezuordnungseditor 34, 35
Übersicht 33
Zieltaten bearbeiten 35
Tabellezuordnungseditor 34
Neue Spaltenzuordnung, Fenster 35
Tabellezuordnung, Registerkarte 35
TRANS CCN, Funktion 68
TRANS COL, Funktion 75
TRANS EML, Funktion 71
TRANS NID, Funktion
britische Sozialversicherungsnummern 86
INSEE, Frankreich 81
italienische Steuernummern 83
kanadische Sozialversicherungsnummern 79
spanische Steuernummern für Ausländer 85
Übersicht 77
US-amerikanische Sozialversicherungsnummern 88
TRANS SSN, Funktion 64
Traversierungsschritte, Fenster 21

U

Umsetzungsanforderung an Optim-Serviceassistent 17
Unterzeichenfolgefunktion 89

V

Variable hinzufügen, Fenster 25
Variablen
erstellen 25
Übersicht 25
Variable hinzufügen, Fenster 25

Verbindungseigenschaften, Assistent für 15
Vergleichsanforderungseditor 122, 123
Verkettete Ausdrücke 94
Verzeichnis zuordnen, Fenster 17
Verzeichnisexplorer 3

W

Währungsfunktion 99
direkte Konvertierung 99
Triangulation 100
Weitergabefunktion 93

Z

Zufallsfunktion 90
Zugriffsdefinitionen
Auswahlkriterien 22
Auswahlkriterien bearbeiten, Fenster 22
Beziehungen verwalten 23
Beziehungstraversierung 23
erstellen 20
Neue Zugriffsdefinition, Assistent 20
Point-and-shoot-Liste 26
Tabelle verwalten 20
Tabellen entfernen 21
Tabellen hinzufügen 20
Tabellen hinzufügen, Assistent 20
Tabellen in Referenz- oder zugehörige Tabellen ändern 21
Tabellezugriffsoptionen 24
Traversierungsschritte 21
Traversierungsschritte, Fenster 21
Übersicht 19
Variablen 25
Zugriffsdefinitionseditor 21, 23
Zugriffsdefinitionseditor 21
Beziehungen, Registerkarte 23, 24
Datengruppeneigenschaften, Registerkarte 23
Point-and-shoot, Registerkarte 26
Tabellen, Registerkarte 23

