

Versão 9 Release 1

*IBM InfoSphere Optim
Usando o Optim Designer*

IBM

Versão 9 Release 1

*IBM InfoSphere Optim
Usando o Optim Designer*

IBM

Nota

Antes de usar estas informações e o produto suportado por elas, leia as informações em “Avisos” na página 129.

Primeira Edição

Esta edição se aplica à versão 2, liberação 2, modificação 3 do Optim Designer e a todas as liberações e modificações subsequentes até que seja indicado de outra forma em novas edições.

© Copyright IBM Corporation 1996, 2012.

Índice

Capítulo 1. Usando o InfoSphere Optim Designer 1

Introdução	1
Perspectiva do Optim	1
Usando o Optim Manager a Partir do Optim Designer.	4
Dados de Amostra	4
Suporte ao Banco de Dados	13
Recursos de Acessibilidade	13

Capítulo 2. Gerenciando as Conexões de Origem de Dados 15

Trabalhando com um Alias de Armazenamento de Dados	15
Definindo um Alias de Armazenamento de Dados	15
Editando um Alias de Armazenamento de Dados	15
Conectando a um Alias de Armazenamento de Dados	16
Trabalhando com o Repositório Optim	16
Conectando-se ao Repositório Optim	16
Editando uma Conexão do Repositório	16
Alterando uma Conexão do Repositório	16
Associando um Diretório Optim ao Repositório	17
Transformando uma Solicitação em um Serviço	17

Capítulo 3. Gerenciando Modelos de Dados 19

Trabalhando com Definições de Acesso	19
Criando uma Definição de Acesso	20
Gerenciando Tabelas em uma Definição de Acesso	20
Gerenciando Critérios de Seleção	21
Gerenciando Relacionamentos em uma Definição de Acesso	23
Gerenciando Variáveis em uma Definição de Acesso	24
Gerenciando as Listas de Apontar e Disparar	25
Trabalhando com Relacionamentos Optim	26
Regras de Compatibilidade para Relacionamentos	28
Criando um Relacionamento Optim	30
Editando um Relacionamento Optim	30
Trabalhando com Chaves Primárias do Optim	31
Criando uma Chave Primária	32
Editando uma Chave Primária	32
Trabalhando com Mapas da Tabela	33
Criando um Mapa da Tabela	34
Editando um Mapa da Tabela	34
Trabalhando com Mapas da Coluna	35
Regras de Compatibilidade para Mapas da Coluna	36
Criando um Mapa da Coluna	37
Editando um Mapa de Coluna	37

Capítulo 4. Projetando Serviços de Gerenciamento de Dados. 41

Trabalhando com Serviços de Extração	41
Criando um Serviço de Extração	41
Editando um Serviço de Extração	42
Trabalhando com Serviços de Conversão	44
Criando um Serviço de Conversão.	44
Editando um Serviço de Conversão	45
Trabalhando com Serviços de Inserção	45
Criando um Serviço de Inserção	46
Editando um Serviço de Inserção	46
Trabalhando com Serviços de Carregamento	47
Criando um Serviço de Carregamento	49
Editando um Serviço de Carregamento	49
Testando um Serviço de Gerenciamento de Dados	51

Capítulo 5. Mascarando os Dados 53

Aplicando uma Função de Mascaramento de Dados	53
Funções de Mascaramento de Dados	53
Funções de Consulta	53
Função Ordenar Aleatoriamente	62
Função TRANS SSN	64
Função TRANS CCN	67
Função TRANS EML	70
Função TRANS COL	75
TRANS NID	77
Função de Subsequência	88
Função Aleatória	89
Função Sequencial	89
Função de Identidade ou Serial.	90
Função de Sequência Oracle.	90
Função Propagar Valor de Chave Estrangeira ou Primária	91
Expressões Concatenadas.	93
Expressões numéricas	93
Funções de Valor e Literal	94
Função Idade.	95
Função de Moeda	97
Nome de Email Gerado Automaticamente.	100
Nome de Email Formatado.	100
Função de Número Aleatório	100
Função de Número Sequencial	100
Usando Rotinas de Saída	100
Gravando Rotinas de Saída.	101
Rotina de Saída Padrão	102
Saída do Formato de Origem	104
Saída do Formato de Destino	106
Gravando Procedimentos de Mapa de Coluna com o Script Lua	109
Funções de Lua para Procedimentos de Mapa de Coluna	109
Limitações de Procedimentos de Mapa de Coluna	111
Exemplo de Procedimento de Mapa de Coluna: Procedimento Genérico	111

Exemplo de Procedimento de Mapa de Coluna:	
Consulta Alternada	113
Exemplo de Procedimento de Mapa de Coluna:	
Máscara de ID Nacional	116

Capítulo 6. Gerenciando Dados. . . . 119

Procurando por Dados	119
Editando Dados	119
Comparando Dados	120
Definindo uma Solicitação de Comparação	120
Editando uma Solicitação de Comparação	121
Executando um Processo de Comparação	121
Criando Tabelas	121

Capítulo 7. Gerenciando Dados com o Optim Designer 123

Extraindo Dados com o Optim Designer	123
Criando uma Pasta do Explorer de Repositório	123
Criando uma Definição de Acesso	123
Definindo Critérios de Seleção.	124
Criando um Serviço de Extração	125
Mascarando Dados com o Optim Designer	125
Criando um Mapa da Tabela	126
Editando um Mapa da Tabela	126
Criando um Mapa da Coluna	126
Aplicando uma Função de Mascaramento de Dados	127
Incluindo um Mapa da Coluna em um Mapa da Tabela	127
Criando um Serviço de Conversão	128

Índice Remissivo. 133

Capítulo 1. Usando o InfoSphere Optim Designer

Use IBM® InfoSphere Optim Designer para definir modelos de dados, políticas de privacidade de dados, serviços de gerenciamento de dados e armazenamento de dados. É possível definir os objetos no Optim Designer e compartilhá-los no repositório Optim. Use o Repository Explorer para gerenciar os objetos no repositório. Use o Directory Explorer para visualizar os objetos e transformar as solicitações em um diretório Optim.

Serviços de gerenciamento de dados

Use um serviço de gerenciamento de dados para extrair, converter, carregar ou inserir os dados. Um serviço usa as definições de acesso para definir os dados a serem extraídos. Um serviço também usa mapas de tabela e mapas de coluna para mapear os dados de origem e de destino. Também é possível transformar as solicitações de processo em um Optim Directory em serviços de gerenciamento de dados. Use Optim Manager no modo integrado para testar os serviços.

Máscara de Dados

É possível mascarar os dados processados por um serviço de gerenciamento de dados. Use a máscara de dados para transformar dados como números de ID nacional, números de cartão de crédito, valores numéricos e informações pessoais. Use um mapa da coluna para inserir uma função de mascaramento de dados ou crie um procedimento de mapa de coluna com um script LUA.

Definições de Acesso

Use uma definição de acesso para definir os modelos de dados e os subconjuntos de dados em esquemas correspondentes ou diferentes. Uma definição de acesso identifica os critérios de seleção e os relacionamentos usados em um serviço de gerenciamento de dados. É possível definir os relacionamentos definidos fora de um DBMS e usar SQL para definir os critérios de seleção.

Optim Manager

É possível abrir o Optim Manager a partir do Optim Designer (modo integrado), permitindo testar os serviços de gerenciamento de dados.

Introdução

Para iniciar, você deve usar o Repository Explorer para se conectar a um repositório Optim.

Perspectiva do Optim

No Optim Designer, a perspectiva Optim fornece as ferramentas necessárias para definir modelos de dados, serviços de gerenciamento de dados e se conectar a um diretório do Optim. Ao abrir pela primeira vez o Optim Designer, a perspectiva Optim é a perspectiva padrão.

A perspectiva do Optim inclui as seguintes visualizações:

Repository Explorer

Use o Repository Explorer para definir os aliases de armazenamento de dados, definições de acesso, serviços de gerenciamento de dados e outros objetos armazenados em um repositório Optim.

Directory Explorer

Use o Directory Explorer para se conectar a um diretório Optim.

Para abrir a perspectiva Optim, clique em **Janela > Abrir Perspectiva > Outro**. Na janela Abrir Perspectiva, selecione **Optim**.

Repository Explorer

Use o Repository Explorer para definir os aliases de armazenamento de dados, definições de acesso, serviços de gerenciamento de dados e outros objetos armazenados em um repositório Optim.

Quando o Optim Designer é conectado a um repositório Optim, o Repository Explorer inclui os objetos a seguir:

Alias de Armazenamento de Dados

Um alias de armazenamento de dados é um objeto definido pelo usuário associado a um banco de dados. Ao definir um alias de armazenamento de dados, forneça os parâmetros que o Optim usa para se comunicar com esse banco de dados. Esses parâmetros incluem o tipo e a versão do sistema de gerenciamento de banco de dados (DBMS) e as propriedades para uma conexão do cliente e uma conexão JDBC.

Um nome de alias de armazenamento de dados serve como um qualificador de alto nível que permite acessar um banco de dados específicos para executar as funções solicitadas. Por exemplo, em uma definição de acesso, você deve qualificar o nome de uma tabela com um nome do alias de armazenamento de dados. O alias de armazenamento de dados referenciado fornece os parâmetros necessários para se conectar ao banco de dados no qual a tabela reside.

Chaves Primárias Optim

Uma chave primária é a coluna ou as colunas que contêm os valores que identificam exclusivamente cada linha em uma tabela. Um banco de dados deve ter uma chave primária para Optim para inserir, atualizar, restaurar ou excluir os dados de uma tabela de banco de dados. O Optim usa as chaves primárias que são definidas para o banco de dados. No entanto, também é possível definir as chaves primárias Optim para complementar as chaves primárias no banco de dados.

Relacionamentos Optim

Optim usa os relacionamentos para determinar os dados a serem recuperados das tabelas relacionadas e depende dos relacionamentos definidos para o banco de dados, quando disponíveis. No entanto, também é possível definir os relacionamentos Optim para complementar os relacionamentos no banco de dados.

Definições de Acesso

Use as definições de acesso para especificar as tabelas, passagem de relacionamento e critérios de seleção para os dados que você deseja processar.

Mapas da Coluna

Um mapa da coluna fornece especificações necessárias para corresponder ou excluir as colunas do processamento em um serviço de gerenciamento de dados. Os serviços de conversão, inserção e carregamento devem referenciar um mapa da tabela, que podem referenciar um ou mais mapas de coluna. É possível usar um mapa da coluna para definir as transformações de dados com funções ou procedimentos de mapa da coluna.

Mapas da Tabela

Use um mapa da tabela para definir as especificações para correlacionar as tabelas de origem e destino de dados compatíveis. É possível mapear as tabelas que possuem diferentes nomes, modificar os nomes de tabela, excluir as tabelas de um processo ou incluir os mapas de coluna para maior controle sobre os dados.

Serviços

Use um serviço de gerenciamento de dados para extrair, converter, carregar ou inserir os dados. Um serviço usa as definições de acesso para definir os dados a serem extraídos. Um serviço também usa mapas de tabela e mapas de coluna para mapear os dados de origem e de destino. É possível mascarar dados aplicando uma política de privacidade de dados a uma entidade processada por um serviço. Também é possível transformar as solicitações de processo em um diretório Optim nos serviços de gerenciamento de dados. Use Optim Manager no modo integrado para testar os serviços.

Pastas

Uma pasta do Explorer de Repositório contém serviços, definições de acesso, mapas de coluna e mapas de tabela. Use a pasta para organizar esses objetos.

Criando uma Pasta do Explorer de Repositório:

Utilize a janela Nova Pasta para criar uma pasta no Explorer de Repositório.

Uma pasta do Explorer de Repositório contém serviços, definições de acesso, mapas de coluna e mapas de tabela. Use a pasta para organizar esses objetos.

Para criar uma pasta do Explorer de Repositório:

1. Dependendo da existência de uma pasta, execute uma das seguintes etapas:
 - a. Para criar uma pasta inicial, clique em **Clique para criar uma nova pasta** no Explorer de Repositório.
 - b. Para criar uma pasta adicional, clique com o botão direito do mouse na visualização Explorer de Repositório e clique em **Novo > Pasta**.

A janela Nova Pasta é aberta.

2. Insira um nome de pasta e clique em **OK**.

A nova pasta é exibida no Explorer de Repositório.

Directory Explorer

Use o Directory Explorer para se conectar a um diretório Optim. Apenas um diretório Optim pode estar associado ao repositório.

Use o Directory Explorer para transformar as solicitações de processo nos serviços de gerenciamento de dados.

As definições a seguir estão disponíveis.

Ícone	Definição
	definição de acesso
	solicitação de archive
	coluna
	mapa de coluna
	solicitação de conversão
	ID criador
	Alias do BD
	excluir solicitação
	solicitação de extração
	solicitação de inserção

Ícone	Definição
	solicitação de carregamento
	chave primária
	relacionamento
	solicitação de restauração
	tabela
	mapa de tabela
	variável

Usando o Optim Manager a Partir do Optim Designer

Use o Optim Manager a partir do Optim Designer para testar os serviços de gerenciamento de dados e os serviços de interoperabilidade Optim. O uso do Optim Manager a partir do Optim Designer também é conhecido como Optim Manager no modo integrado.

O Optim Manager é um aplicativo da web que você pode usar para testar os serviços. O Optim Manager é exibido no navegador interno fornecido pelo Optim Designer ou em um navegador externo. É possível selecionar o navegador que você deseja que o Optim Designer use para o Optim Manager clicando em **Janela > Preferências > Geral > Navegador da Web** no Optim Designer.

Abrindo o Optim Manager a Partir do Optim Designer

É possível abrir o Optim Manager testando um serviço. Também é possível abrir o Optim Manager inserindo a seguinte URL em um navegador da Web: `http://localhost:portnumber/console`, em que *portnumber* é o número da porta designado ao Optim Manager. O número da porta padrão é 60000.

Se o navegador exibir uma mensagem sobre uma página não localizada ao abrir o Optim Manager, poderá existir um conflito de porta e você deverá alterar o número da porta do Optim Manager.

Também é possível abrir o Optim Manager a partir da barra de ferramentas do Optim Designer.

Alterando o Número da Porta do Optim Manager

Para alterar o número da porta, você deve editar a seguinte propriedade no arquivo `eclipse.ini` localizado no diretório de instalação do Optim Designer:

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

em que *portnumber* é o novo número da porta do Optim Manager. Se o Optim Designer estiver aberto, você deverá reiniciar o aplicativo para aplicar o novo número da porta.

Dados de Amostra

O Optim fornece dados de amostra para mascaramento de dados, bem como uma origens de dados de origem e de destino predefinidas.

Tabelas e Estrutura do Banco de Dados de Amostra

Como um grupo, as tabelas de banco de dados incluem informações sobre clientes e pedidos, com instruções de remessa. As tabelas de amostra também incluem informações sobre vendas e inventário.

Pequenas diferenças nos tipos de dados existem, dependendo do DBMS usado para instalar as tabelas de amostra. O diagrama a seguir mostra a estrutura básica do banco de dados de amostra.

Como um grupo, as tabelas de banco de dados incluem informações sobre clientes e pedidos, com instruções de remessa. As tabelas de amostra também incluem informações sobre vendas e inventário. Pequenas diferenças nos tipos de dados existem, dependendo do DBMS usado para instalar as tabelas de amostra.

Um conjunto de tabelas adicional também é instalado com o banco de dados de amostra. As tabelas no conjunto adicionais têm os mesmos nomes que as tabelas no primeiro conjunto, com o sufixo "2" anexado. As quatro tabelas neste conjunto adicional são:

- OPTIM_CUSTOMERS2
- OPTIM_ORDERS2
- OPTIM_DETAILS2
- OPTIM_ITEMS2

As tabelas no conjunto adicional não contêm dados. Eles são utilizados para demonstrar os recursos do Optim.

Tabela OPTIM_SALES:

A tabela OPTIM_SALES identifica cada vendedor por nome, número de ID e gerente.

A tabela OPTIM_SALES possui as seguintes colunas:

SALESMAN_ID

CHAR; até 6 caracteres; não pode conter valor nulo.

FIRST_NAME

VARCHAR; até 15 caracteres; não pode conter valor nulo.

LAST_NAME

VARCHAR; até 15 caracteres; não pode conter valor nulo.

NATIONALITY

VARCHAR; até 30 caracteres

NATIONAL_ID

VARCHAR; até 30 caracteres

PHONE_NUMBER

VARCHAR; até 20 caracteres; não pode conter valor nulo.

EMAIL_ADDRESS

VARCHAR; até 70 caracteres; não pode conter valor nulo.

AGE SMALLINT; não pode conter valor nulo; possui um valor padrão.

SEX CHAR; 1 caractere; não pode conter valor nulo; possui um valor padrão.

TERRITORY

VARCHAR; até 14 caracteres; não pode conter valor nulo.

MANAGER_ID

VARCHAR; até 6 caracteres

Chaves Primárias

A coluna SALESMAN_ID é a coluna de chave primária.

Relacionamentos com Outras Tabelas

A tabela OPTIM_SALES é pai:

- Da tabela OPTIM_CUSTOMERS, por meio de uma chave estrangeira da coluna SALESMAN_ID.
- Da tabela OPTIM_MALE_RATES, por meio de um relacionamento orientado a dados do Optim na coluna AGE com SEX = 'M'.
- Da tabela OPTIM_FEMALE_RATES, por meio de um relacionamento orientado a dados do Optim na coluna AGE com SEX = 'F'.
- A tabela OPTIM_STATE_LOOKUP, por meio de um relacionamento de subsequência do Optim usando SUBSTR(SALESMAN_ID,1,2).

Tabela OPTIM_CUSTOMERS:

A tabela OPTIM_CUSTOMERS contém nomes, números de IDs e endereços de clientes.

A tabela OPTIM_CUSTOMERS possui as seguintes colunas:

CUST_ID

CHAR; até 5 caracteres; não pode conter valor nulo; contém uma restrição de ponto de verificação.

CUSTNAME

CHAR; até 20 caracteres; não pode conter valor nulo.

ADDRESS1

VARCHAR; até 100 caracteres; não pode conter valor nulo.

ADDRESS2

VARCHAR; até 100 caracteres; não pode conter valor nulo.

LOCALITY

VARCHAR; até 56 caracteres

CITY VARCHAR; até 60 caracteres

STATE

VARCHAR; até 30 caracteres

COUNTRY_CODE

CHAR; até 2 caracteres

POSTAL_CODE

VARCHAR; até 15 caracteres

POSTAL_CODE_PLUS4

CHAR; até 4 caracteres; pode conter valor nulo.

EMAIL_ADDRESS

VARCHAR; até 70 caracteres

PHONE_NUMBER

VARCHAR; até 20 caracteres

YTD_SALES

DECIMAL; valor em dinheiro; não pode conter valor nulo; possui um valor padrão.

SALESMAN_ID

CHAR; até 6 caracteres

NATIONALITY

VARCHAR; até 30 caracteres

NATIONAL_ID
VARCHAR; até 30 caracteres

CREDITCARD_NUMBER
VARCHAR; até 19 caracteres

CREDITCARD_TYPE
VARCHAR; até 30 caracteres

CREDITCARD_EXP
CHAR; 4 caracteres

CREDITCARD_CVV
VARCHAR; até 4 caracteres

DRIVER_LICENSE
VARCHAR; até 30 caracteres

CUSTOMER_INFO
XMLTYPE

Chaves Primárias

A coluna CUST_ID é a coluna de chave primária.

Relacionamentos com Outras Tabelas

A tabela OPTIM_CUSTOMERS é pai:

- Da tabela OPTIM_ORDERS, por meio de uma chave estrangeira da coluna CUST_ID.
- Da tabela OPTIM_SHIP_TO, por meio de um relacionamento do Optim na coluna CUST_ID.

A tabela OPTIM_CUSTOMERS é filha:

- Da tabela OPTIM_SALES, por meio de uma chave estrangeira da coluna SALESMAN_ID.

Tabela OPTIM_ORDERS:

A tabela OPTIM_ORDERS contém informações de pedidos, incluindo o número do pedido, ID de cliente e vendedor.

A tabela OPTIM_ORDERS possui as seguintes colunas:

ORDER_ID
DECIMAL; número do ID do pedido; não pode conter valor nulo.

CUST_ID
CHAR; número do ID do cliente; até 5 caracteres, não pode conter valor nulo.

ORDER_DATE
TIMESTAMP; data do pedido; não pode conter valor nulo; possui um valor padrão.

ORDER_TIME
TIMESTAMP; hora do dia; não pode conter valor nulo; possui um valor padrão.

FREIGHT_CHARGES
DECIMAL; quantia em dinheiro

ORDER_SALESMAN
CHAR; até 6 caracteres

ORDER_POSTED_DATE
TIMESTAMP; não pode conter valor nulo; possui um valor padrão.

ORDER_SHIP_DATE

CHAR; data em que o pedido é enviado; até 8 caracteres; não pode conter valor nulo; possui valor padrão.

Chaves Primárias

A coluna ORDER_ID é a coluna de chave primária.

Relacionamentos com Outras Tabelas

A tabela OPTIM_ORDERS é pai da tabela OPTIM_DETAILS, por meio de uma chave estrangeira na coluna ORDER_ID.

A tabela OPTIM_ORDERS é filha da tabela OPTIM_CUSTOMERS, por meio de uma chave estrangeira na coluna CUST_ID.

Tabela OPTIM_DETAILS:

A tabela OPTIM_DETAILS contém informações adicionais de cada pedido na tabela OPTIM_ORDERS.

A tabela OPTIM_DETAILS possui as seguintes colunas:

ORDER_ID

DECIMAL; número do ID do pedido; não pode conter valor nulo.

ITEM_ID

CHAR; até 5 caracteres; número do ID de item; não pode conter valor nulo.

ITEM_QUANTITY

DECIMAL; número de itens; não pode conter valor nulo.

DETAIL_UNIT_PRICE

DECIMAL; preço unitário; quantia em dinheiro; não pode conter valor nulo.

Chaves Primárias

As colunas ORDER_ID e ITEM_ID são a chave primária.

Relacionamentos com Outras Tabelas

A tabela OPTIM_DETAILS é filha:

- Da tabela OPTIM_ORDERS, por meio de uma chave estrangeira da coluna ORDER_ID.
- Da tabela OPTIM_ITEMS, por meio de uma chave estrangeira da coluna ITEM_ID.

Tabela OPTIM_ITEMS:

A tabela OPTIM_ITEMS contém informações sobre cada item de um pedido, incluindo descrição, preço e a quantidade no inventário.

A tabela OPTIM_ITEMS possui as seguintes colunas:

ITEM_ID

CHAR; até 5 caracteres; não pode conter valor nulo.

ITEM_DESCRIPTION

VARCHAR; até 72 caracteres; não pode conter valor nulo.

CATEGORY

VARCHAR; até 14 caracteres; não pode conter valor nulo.

RATING

CHAR; até 4 caracteres; não pode conter valor nulo.

UNIT_PRICE

DECIMAL; quantia em dinheiro; não pode conter valor nulo.

ON_HAND_INVENTORY

INTEGER; não pode conter valor nulo.

Chaves Primárias

A coluna ITEM_ID é a coluna de chave primária.

Relacionamentos com Outras Tabelas

A tabela OPTIM_ITEMS é pai da tabela OPTIM_DETAILS, por meio de uma chave estrangeira na coluna ITEM_ID.

Tabela OPTIM_SHIP_TO:

A tabela OPTIM_SHIP_TO contém informações de remessa de pedido.

A tabela OPTIM_SHIP_TO possui as seguintes colunas:

CUST_ID

CHAR; até 5 caracteres; não pode conter valor nulo.

SHIP_ID

DECIMAL; não pode conter valor nulo.

ADDRESS1

VARCHAR; até 100 caracteres

ADDRESS2

VARCHAR; até 100 caracteres

LOCALITY

VARCHAR; até 56 caracteres

CITY VARCHAR; até 30 caracteres

STATE

VARCHAR; até 30 caracteres

COUNTRY_CODE

CHAR; abreviação de 2 caracteres

POSTAL_CODE

VARCHAR; até 15 caracteres

POSTAL_CODE_PLUS4

CHAR; 4 caracteres

IN_CARE_OF

VARCHAR; até 31 caracteres

SHIPPING_CHANGE_DT

TIMESTAMP; não pode conter valores nulos; possui um valor padrão.

Chaves Primárias

A coluna SHIP_ID é a coluna de chave primária.

Relacionamentos com Outras Tabelas

A tabela OPTIM_SHIP_TO é pai da tabela OPTIM_SHIP_INSTR, por meio de um relacionamento Optim na coluna SHIP_ID.

A tabela OPTIM_SHIP_TO é filha da tabela OPTIM_CUSTOMERS, por meio de um relacionamento Optim na coluna CUST_ID.

Tabela OPTIM_SHIP_INSTR:

A tabela OPTIM_SHIP_INSTR contém informações detalhadas para ordem de remessa.

A tabela OPTIM_SHIP_INSTR possui as seguintes colunas:

SHIP_ID

DECIMAL

SHIP_INSTR_ID

INTEGER

ORDER_SHIP_INSTR

VARCHAR; até 254 caracteres

SHIP_UPDATED

TIMESTAMP; não pode conter valor nulo; possui um valor padrão.

Chaves Primárias

A coluna SHIP_INSTR_ID é a coluna de chave primária.

Relacionamentos com Outras Tabelas

A tabela OPTIM_SHIP_INSTR é filha da tabela OPTIM_SHIP_TO, por meio de um relacionamento Optim na coluna SHIP_ID.

Tabela OPTIM_MALE_RATES:

A tabela OPTIM_MALE_RATES contém taxas de seguros com base na idade.

A tabela OPTIM_MALE_RATES possui as seguintes colunas:

AGE SMALLINT

RATE_PER_1000

DECIMAL; taxa em quantia em dinheiro

Chaves Primárias

A coluna RATE_PER_1000 é a coluna de chave primária.

Relacionamentos com Outras Tabelas

A tabela OPTIM_MALE_RATES é filha da tabela OPTIM_SALES, por meio de um relacionamento orientado a dados do Optim na coluna AGE.

Tabela OPTIM_FEMALE_RATES:

A tabela OPTIM_FEMALE_RATES contém taxas de seguros com base na idade.

A tabela OPTIM_FEMALE_RATES possui as seguintes colunas:

AGE SMALLINT

RATE_PER_1000

DECIMAL; taxa em quantia em dinheiro

Chaves Primárias

A coluna RATE_PER_1000 é a coluna de chave primária.

Relacionamentos com Outras Tabelas

A tabela OPTIM_FEMALE_RATES é filha da tabela OPTIM_SALES, por meio de um relacionamento orientado a dados do Optim na coluna AGE.

Tabela OPTIM_STATE_LOOKUP:

A tabela OPTIM_STATE_LOOKUP contém códigos de estado e abreviações correspondentes.

A tabela OPTIM_STATE_LOOKUP possui as seguintes colunas:

DIST_CODE

CHAR; 3 caracteres; não pode conter valor nulo.

DISTRICT

CHAR; 2 caracteres; não pode conter valor nulo.

Chaves Primárias

A tabela OPTIM_STATE_LOOKUP não possui chave primária:

Relacionamentos com Outras Tabelas

A tabela OPTIM_STATE_LOOKUP é filha da tabela OPTIM_SALES por meio de um relacionamento de subsequência na coluna DISTRICT utilizando SUBSTR(SALESMAN_ID,1,2).

Tabelas de Privacidade de Dados

As tabelas de privacidade de dados estão disponíveis para clientes que possuem uma licença do Optim Data Privacy. Utilize as tabelas para mascarar dados corporativos e pessoais, como nomes de funcionários, nomes de clientes, números de seguridade social, números de cartões de crédito e endereços de email. Utilize as tabelas para gerar os dados transformados que são exclusivos e válidos no contexto do aplicativo.

As tabelas de privacidade de dados podem ser usadas para:

- Evitar violações de privacidade internas ao identificar ou mascarar os dados disponíveis para desenvolvedores, testadores de controle de qualidade, e outra equipe.
- Melhorar as iniciativas de conformidade de privacidade ao substituir os dados do cliente por dados contextualmente precisos, porém fictícios.
- Proteger as informações e dados confidenciais do cliente e dos funcionários em seus ambientes de desenvolvimento e de teste do aplicativo.
- Assegurar resultados de teste válidos ao propagar elementos mascarados entre tabelas relacionadas para garantir a integridade referencial do banco de dados.

Cada categoria de dados pessoais é fornecida em uma tabela separada para os seguintes países (as abreviações estão entre parênteses): Austrália (AU), Canadá (CA), França (FR), Alemanha (DE), Itália (IT),

Japão (JP), Espanha (ES), Reino Unido (UK) e Estados Unidos (US). Cada tabela inclui uma coluna de números sequenciais usada com políticas de consulta que usam valores do hash para selecionar uma linha na tabela de consulta.

Cada nome de tabela é composto de um prefixo de abreviação de país e da categoria (*countryabbreviation_category*). Por exemplo, a tabela de endereços para Canadá é denominada CA_ADDRESSES e a tabela de endereços para Alemanha é denominada DE_ADDRESSES.

O esquema inclui as seguintes categorias:

ADDRESSES

Tabelas que incluem colunas para endereço residencial, cidade, localidade (por exemplo, estado ou província), e código de endereçamento postal.

FIRSTNAME

Tabelas que incluem uma coluna com nomes masculinos e femininos.

FIRSTNAME_F

Tabelas que incluem uma coluna com nomes femininos.

FIRSTNAME_M

Tabelas que incluem uma coluna com nomes masculinos.

LASTNAME

Tabelas que incluem uma coluna com sobrenomes.

PERSON

Tabelas que incluem colunas para data de nascimento, nome, sobrenome, sexo, número de telefone, número do RG nacional, nome da empresa e endereço de e-mail.

CCN Tabelas que incluem números de cartão de crédito para o emissor associado (MasterCard, VISA, etc.).

DOMAIN_NAMES

Tabela que inclui os nomes de domínio para mascarar endereços de email.

Criando Tabelas de Dados de Amostra

Utilize o utilitário de Configuração para criar tabelas de dados de amostra.

Para criar tabelas de dados de amostra:

1. No Optim Designer, clique em **Utilitários > Configurar**. O utilitário Configuração é aberto.
2. Clique em **Tarefas > Carregar/Descartar Dados de Amostra**. O assistente Carregar/Descartar Dados de Amostra é aberto.
3. Conclua as etapas do assistente.
Você deve selecionar um diretório Optim. O nome do diretório padrão do Optim do repositório do Optim é PODREPO.
Quando solicitado a selecionar um alias de BD, selecione um nome de alias de armazenamento de dados.

Criando Tabelas de Privacidade de Dados

Utilize o utilitário de Configuração para criar tabelas de privacidade de dados.

Para criar tabelas de privacidade de dados:

1. No Optim Designer, clique em **Utilitários > Configurar**. O utilitário Configuração é aberto.
2. Clique em **Tarefas > Carregar/Descartar Dados de Privacidade de Dados**. O assistente Carregar/Descartar Dados de Privacidade de Dados é aberto.
3. Conclua as etapas do assistente.

Você deve selecionar um diretório Optim. O nome do diretório padrão do Optim do repositório do Optim é PODREPO.

Quando solicitado a selecionar um alias de BD, selecione um nome de alias de armazenamento de dados.

Suporte ao Banco de Dados

O Optim Designer fornece suporte para diversos sistemas de gerenciamento de banco de dados.

O Optim Designer suporta os seguintes bancos de dados:

- DB2 para z/OS V8.1, V9.1, V10.1
- DB2 para Linux, UNIX e Windows V8.2, V9.1, V9.5, V9.7
- DB2 para i V5.4
- Informix V10
- Microsoft SQL Server 2005, 2008
- Oracle V10.2, V11, V11.2
- Sybase V12.5, V15, V15.5
- Teradata V2.6, V12, V13
- IBM Informix V11.5, V.11.7

Pré-requisitos do DB2

Para permitir que o Optim obtenha metadados JDBC completos de uma instância do DB2 z/OS, o valor DESCSTAT em ZPARMS deve ser configurado como YES. Além disso, você deve executar a tarefa DSNTIJMS de instalar os procedimentos armazenados necessários para JDBC, ligar os pacotes necessários e configurar permissões de segurança. Além disso, definições do WLM (Workload Manager) são necessárias para assegurar que o WLM pode iniciar o espaço de endereço do procedimento armazenado quando lido pelo DB2.

Recursos de Acessibilidade

Os recursos de acessibilidade ajudam as pessoas com alguma deficiência física, como mobilidade restrita ou visão limitada, ou com outras necessidades especiais a utilizarem os produtos de software com êxito.

O Optim Designer utiliza recursos de acessibilidade disponíveis com o ambiente do Eclipse.

Os recursos de acessibilidade ajudam pessoas com deficiências físicas, como mobilidade restrita ou visão limitada, ou com necessidades especiais a usarem os produtos de software com êxito.

Capítulo 2. Gerenciando as Conexões de Origem de Dados

Use o Repository Explorer para criar os aliases de armazenamento de dados. Use o Directory Explorer para se conectar a um diretório Optim.

Trabalhando com um Alias de Armazenamento de Dados

Um alias de armazenamento de dados é um objeto definido pelo usuário associado a um banco de dados. Ao definir um alias de armazenamento de dados, forneça os parâmetros que o Optim usa para se comunicar com esse banco de dados. Esses parâmetros incluem o tipo e a versão do sistema de gerenciamento de banco de dados (DBMS) e as propriedades para uma conexão do cliente e uma conexão JDBC.

Um nome de alias de armazenamento de dados serve como um qualificador de alto nível que permite acessar um banco de dados específicos para executar as funções solicitadas. Por exemplo, em uma definição de acesso, você deve qualificar o nome de uma tabela com um nome do alias de armazenamento de dados. O alias de armazenamento de dados referenciado fornece os parâmetros necessários para se conectar ao banco de dados no qual a tabela reside.

Um alias de armazenamento de dados identifica um banco de dados específico e serve como um prefixo nos nomes completos de chaves primárias, tabelas de banco de dados e relacionamentos. Os aliases de armazenamento de dados são elementos essenciais no gerenciamento dos bancos de dados. As regras a seguir se aplicam aos nomes de alias de armazenamento de dados:

- Cada banco de dados pode ter apenas um alias de armazenamento de dados.
- Cada nome de alias de armazenamento de dados deve ser exclusivo.
- Os objetos no repositório não podem ter o mesmo nome que um alias de armazenamento de dados.

Definindo um Alias de Armazenamento de Dados

Use o assistente Novo Alias do Armazenamento de Dados para definir um novo alias de armazenamento de dados.

Para definir um alias de armazenamento de dados, você precisa tanto a sequência de conexões para uma conexão do cliente e o arquivo .jar para uma conexão JDBC.

Para definir um alias de armazenamento de dados:

1. No Repository Explorer, clique com o botão direito em **Alias de Armazenamento de Dados** e clique em **Novo Alias de Armazenamento de Dados**. O assistente Novo Alias de Armazenamento de Dados se abre.
2. Conclua as etapas do assistente.
Você deve inserir as propriedades para uma conexão do cliente e uma conexão JDBC. Você deve também identificar o conjunto de caracteres usado pelo banco de dados e como gerenciar os pacotes para acessar objetos do Optim.

Editando um Alias de Armazenamento de Dados

Use o assistente Propriedades de Conexão para editar um alias de armazenamento de dados.

Para editar um alias de armazenamento de dados:

1. No Repository Explorer, expanda o nó **Alias de Armazenamento de Dados**, clique com o botão direito no alias de armazenamento de dados e clique em **Abrir**. O assistente Propriedades da Conexão se abre.

2. Conclua as etapas do assistente.

Conectando a um Alias de Armazenamento de Dados

Use o Repository Explorer para se conectar a um alias de armazenamento de dados.

Para conectar-se a um alias de armazenamento de dados:

No Repository Explorer, expanda o nó **Alias de Armazenamento de Dados**, clique com o botão direito no alias de armazenamento de dados e clique em **Conectar**.

Trabalhando com o Repositório Optim

O repositório Optim contém definições, como aliases de armazenamento de dados, serviços e definições de acesso, que são compartilhadas entre os componentes Optim.

Apenas um diretório Optim pode estar associado ao repositório.

Conectando-se ao Repositório Optim

Use a janela Nova Conexão de Repositório para definir uma conexão com o repositório Optim.

O repositório Optim deve estar em execução para criar uma conexão.

Para definir uma conexão para o repositório Optim:

1. No Explorer de Repositório, clique com o botão direito do mouse em **Repositório do Optim** e clique em **Novo > Conexão do Repositório**. A janela Nova Conexão de Repositório é aberta.
2. Insira as informações de conexão e as credenciais de usuário para o repositório.
3. Clique em **Conexão de Teste** para verificar a conexão.
4. Clique em **OK**.

Quando o Optim Designer se conecta a repositório, o Repository Explorer exibe o conteúdo do repositório.

Editando uma Conexão do Repositório

Use a janela Editar Conexão ao Repositório para definir uma conexão ao repositório Optim.

Para definir uma conexão para o repositório Optim:

1. No Repository Explorer, clique com o botão direito em **Repositório Optim** e clique em **Abrir**. A janela Editar Conexão de Repositório se abre.
2. Insira as informações de conexão e as credenciais de usuário para o repositório.
3. Clique em **Conexão de Teste** para verificar a conexão.
4. Clique em **OK**.

Quando o Optim Designer se conecta a repositório, o Repository Explorer exibe o conteúdo do repositório.

Alterando uma Conexão do Repositório

Utilize a janela Alternar Repositório para se conectar a um repositório diferente do Optim.

O repositório Optim deve estar em execução para alterar uma conexão.

Para alterar uma conexão para o repositório Optim:

1. No Explorer de Repositório, clique com o botão direito do mouse em **Repositório Optim** e clique em **Alternar Repositório**. A janela Alternar Repositório é aberta.
2. Selecione uma conexão do repositório.
3. Clique em **OK**.

Quando o Optim Designer se conecta a repositório, o Repository Explorer exibe o conteúdo do repositório.

Associando um Diretório Optim ao Repositório

Use a janela Associar Diretório para associar um diretório Optim ao repositório.

Uma conexão de diretório Optim deve ser definido no registro do computador Optim Designer. Use o programa de Configuração Optim para definir uma conexão no registro.

Apenas um diretório Optim pode estar associado ao repositório.

Para associar um diretório Optim ao repositório:

1. No Directory Explorer, clique em **clique para associar um diretório Optim à conexão do repositório**. A janela Associar Diretório se abre.
2. Selecione um diretório Optim.
3. Clique em **Concluir**.

Quando o Optim Designer se conecta ao diretório Optim, o Directory Explorer exibe o conteúdo do diretório Optim.

Transformando uma Solicitação em um Serviço

Use a Solicitação de Transformação para o assistente de Serviço Optim para transformar uma solicitação em um diretório Optim em um serviço de gerenciamento de dados.

Um diretório Optim deve estar associado ao repositório.

Para transformar uma solicitação em um diretório Optim em um serviço de gerenciamento de dados:

1. Na janela Explorer de Diretório, clique em **Transformar**. A Solicitação de Transformação para o assistente de Serviço Optim se abre.
2. Na página Alias DB e Conexões com o Banco de Dados, revise as informações de conexão. Se um alias de armazenamento de dados não estiver definido para um alias DB associado à solicitação, clique em **Editar Conexão** para definir um alias de armazenamento de dados. Clique em **Avançar**.
3. Se a conversão for bem sucedida, clique em **Concluir**. Se a transformação não for bem-sucedida, clique em **Avançar** para revisar as informações de erro.

Uma pasta com o nome do diretório Optim que continha a solicitação será criada no Repository Explorer. A solicitação transformada será listada como um serviço na pasta.

Capítulo 3. Gerenciando Modelos de Dados

É possível criar os modelos de dados que definem os dados a serem extraídos e mapear os dados de destino para os serviços de gerenciamento de dados. Também é possível criar relacionamentos Optim e chaves primárias para complementar os relacionamentos e as chaves primárias no banco de dados.

Trabalhando com Definições de Acesso

Use as definições de acesso para especificar as tabelas, passagem de relacionamento e critérios de seleção para os dados que você deseja processar.

A definição de acesso inclui os itens a seguir:

Tabelas

Uma definição de acesso deve referenciar pelo menos uma tabela, visualização, alias ou sinônimo. A tabela, visualização, alias ou sinônimo da qual as linhas são selecionadas primeiro é chamada tabela de início. É possível inserir o nome de uma tabela de início e incluir facilmente os nome de todas as tabelas relacionadas à tabela de início (a um máximo de 24.000 tabelas).

Tabela de Início

A tabela de início é a primeira tabela a ser usada ao extrair os dados. É possível especificar qualquer tabela na definição de acesso como a tabela de início, exceto uma tabela de referência. Se você não especificar explicitamente uma tabela de início, a primeira tabela na lista de tabelas será a tabela de início.

Tabela de Referência

Tabelas em que todas as linhas são selecionadas, a menos que sejam especificados critérios de seleção para a tabela de referência. Especifique qualquer tabela como uma tabela de início, exceto a tabela de início.

Relacionamentos

Os relacionamentos determinam o caminho de passagem para selecionar os dados a partir das tabelas. Por padrão, os relacionamentos são atravessados de pai para filho, mas você pode controlar a direção da passagem usando as configurações na definição de acesso. Os relacionamentos entre as tabelas referenciadas pela definição de acesso são listados na guia de relacionamento (para um máximo de 24.000 tabelas). É possível selecionar relacionamentos a serem usados no processamento e a direção na qual são passados.

CrITÉRIOS DE SELEÇÃO

Os critérios de seleção definem um conjunto específico de dados a serem usados a partir das tabelas em uma definição de acesso. É possível especificar os operadores SQL e valores e usar as variáveis de substituição com os valores padrão.

Apontar e Disparar

Use a lista de apontar e disparar para selecionar as linhas específicas a partir de uma tabela de início que será incluída em um serviço.

Variáveis

As variáveis são valores padrão definidos pelo usuário especificados em uma definição de acesso. É possível usar essas variáveis de substituição para especificar os critérios de seleção de coluna ou para criar uma cláusula SQL WHERE.

Parâmetros Adicionais

Use parâmetros adicionais para extrair linhas que correspondem a um valor de coluna específico na tabela de início ou para usar uma taxa de amostra especificada (cada linha nth).

Convenções de Nomenclatura

Um nome de definição de acesso completo está no formato *identifier.name*, em que:

identifier

Qualificador designado à definição de acesso (1 a 8 caracteres).

name Nome base designado à definição de acesso (1 a 12 caracteres).

Um conjunto lógico de convenções de nomenclatura pode identificar o uso para cada definição de acesso e pode ser usado para organizá-las para fácil acesso.

Criando uma Definição de Acesso

Use o assistente de Nova Definição de Acesso para criar uma definição de acesso.

Antes de criar uma definição de acesso, um alias de armazenamento de dados deve existir para o banco de dados que contém a tabela de início.

Para criar uma definição de acesso:

1. Expanda uma pasta no Repository Explorer para conter a definição de acesso, clique com o botão direito em **Definições de Acesso** e clique em **Nova Definição de Acesso**. O assistente Nova Definição de Acesso se abre.
2. Conclua as etapas do assistente.
Você deve selecionar um alias de armazenamento de dados e uma tabela de início a partir do alias. É possível optar por incluir as tabelas relacionadas à tabela de início.

A nova definição de acesso é exibida no Editor de Definição de Acesso.

Gerenciando Tabelas em uma Definição de Acesso

Use a guia **Tabelas** no editor de Definição de Acesso para incluir ou remover as tabelas em uma definição de acesso. Também é possível alterar as tabelas às tabelas de referência ou relacionadas.

Incluindo as Tabelas em uma Definição de Acesso

Use o assistente Incluir Tabelas para incluir as tabelas em um alias de armazenamento de dados para uma definição de acesso.

Para incluir as tabelas em uma definição de acesso:

1. Expanda uma pasta no Repository Explorer que contém a definição de acesso, expanda o nó **Definições de Acesso** e clique duas vezes na definição de acesso. O Editor de Definição de Acesso se abre.
2. Selecione a guia **Tabelas**.
3. Clique em **Incluir Tabelas**. O assistente Incluir Tabelas se abre.
4. Conclua as etapas do assistente.
Você deve selecionar um alias de armazenamento de dados que contém as tabelas a serem incluídas. Você pode incluir as tabelas de referência ou as tabelas relacionadas a uma tabela selecionada.
5. Salve a definição de acesso.

Alterando as Tabelas para Referência ou Relacionado

Use o Editor de Definição de Acesso para alterar a tabela em uma definição de acesso para as tabelas relacionadas ou de Referência.

Para alterar as tabelas para referência ou relacionadas:

1. Expanda a pasta no Repository Explorer que contém a definição de acesso, expanda o nó **Definições de Acesso** e clique duas vezes na definição de acesso. O Editor de Definição de Acesso se abre.

2. Selecione a guia **Tabelas**.
3. Selecione uma tabela para alterar para relacionado ou referência.
4. Clique em **Alterar para Referência** ou **Alterar para Relacionado**. A coluna **Tipo** mostrará o novo tipo de tabela.
5. Salve a definição de acesso.

Removendo as Tabelas de uma Definição de Acesso

Use a Editor de Definição de Acesso para remover as tabelas de uma definição de acesso.

Para remover as tabelas de uma definição de acesso:

1. Expanda a pasta no Repository Explorer que contém a definição de acesso, expanda o nó **Definições de Acesso** e clique duas vezes na definição de acesso. O Editor de Definição de Acesso se abre.
2. Selecione a guia **Tabelas**.
3. Selecione uma tabela a ser removida.
4. Clique em **Remover Tabela**.
5. Salve a definição de acesso.

Visualizando e Editando as Etapas de Passagem

Use a janela Etapas de Passagem para visualizar e editar as etapas de passagem de uma definição de acesso.

É possível avaliar as tabelas, relacionamentos e etapas executadas para selecionar os dados. Esta avaliação pode assegurar que você recupere o conjunto desejado de dados.

As etapas em um processo podem ser repetidas qualquer número de vezes de acordo com os relacionamentos que você seleciona e os critérios que você especifica.

Qualquer tabela pode ser revisitada várias vezes em etapas sucessivas. Os ciclos também podem estar envolvidos. Um ciclo faz com que um conjunto de tabelas seja atravessado repetidamente até que uma passagem completa do ciclo não resulte na seleção de linhas adicionais.

Quando um relacionamento é atravessado para selecionar linhas pais de linhas filhas já selecionadas, qualquer critério de seleção para a tabela pai será ignorada.

Para visualizar e editar as etapas de passagem:

1. Expanda uma pasta no Repository Explorer que contém a definição de acesso, expanda o nó **Definições de Acesso** e clique duas vezes na definição de acesso. O Editor de Definição de Acesso se abre.
2. Selecione a guia **Tabelas**.
3. Clique em **Mostrar Etapas**. A janela Etapas de Passagem se abre.
4. Revise as etapas e edite a ordem na qual as tabelas são selecionadas.
5. Clique em **OK**.
6. Salve a definição de acesso.

Gerenciando Critérios de Seleção

Use os critérios de seleção para focalizar em um conjunto específico de dados relacionados, definindo uma cláusula SQL WHERE e usando as variáveis de substituição com os valores padrão.

Critérios de seleção devem estar em conformidade com a sintaxe SQL e incluir operadores relacionais ou lógicos. Os operadores lógicos e a sintaxe variam entre DBMSs. Consulte a documentação de DBMS apropriada para obter informações.

Para selecionar o conjunto desejado de dados para uma tabela, você pode precisar de uma combinação de operadores lógicos AND e OR.

Se a lista Apontar e Disparar também for usada, ela será incluída logicamente com outros critérios com um operador lógico OR.

Quando um relacionamento for atravessado de filho para pai, qualquer critério de seleção para a tabela pais também será ignorado.

Critérios de Dados

Um operador exclusivo, **BEFORE**, permite selecionar dados na base de valores em uma coluna DATE. A sintaxe para este operador é:

BEFORE (*nD nW nM nY*)

Use os argumentos **D**, **W**, **M** e **Y** em qualquer combinação para indicar o número de dias, semanas, meses ou anos subtraídos da data no tempo de execução. Se nenhum argumento for especificado, a data atual será usada. As linhas com uma data posterior à data calculada são extraídas ou arquivadas. O multiplicador *n* é um número inteiro e pode ser opcionalmente precedido por + ou-.

Definindo Critérios de Seleção

Use a janela Editar Critérios de Seleção para definir os critérios de seleção com uma cláusula SQL WHERE.

Para definir os critérios de seleção:

1. Expanda uma pasta no Repository Explorer que contém a definição de acesso, expanda o nó **Definições de Acesso** e clique duas vezes na definição de acesso. O Editor de Definição de Acesso se abre.
2. Selecione a guia **Tabelas**.
3. Selecione uma tabela na qual incluirá os critérios de seleção.
4. Clique em **Editar Critérios de Seleção**. A janela Editar Critérios de Seleção se abre e exibe uma cláusula WHERE para a tabela.
5. Insira os critérios de seleção.

Para exibir uma lista de colunas e operadores, clique com o botão direito na cláusula WHERE e clique em **Assistente de Conteúdo**.

Para incluir uma variável, selecione um **Delimitador de Variável** e clique em **Inserir Variável**.

Clique em **Verificar Sintaxe** para verificar a sintaxe e identificar os erros.

6. Clique em **OK**.

Se a sintaxe não estiver válida, um prompt será aberto para identificar o erro. Não é possível salvar os critérios, se contiver erros.

7. Salve a definição de acesso.

Especificando Opções de Agrupamento de Dados

Utilize a guia **Propriedades de Grupo de Dados** no Editor de Definição de Acesso para especificar a seleção de grupo, que extrai um número de linhas com base em valores em uma coluna específica na tabela de início. As linhas em cada grupo possuem o mesmo valor na coluna selecionada.

Quando os critérios são utilizados em combinação com os parâmetros de seleção de grupo, os critérios são aplicados primeiro com os parâmetros de seleção de grupo aplicados ao resultado.

Para especificar as opções de agrupamento de dados:

1. Expanda uma pasta no Repository Explorer que contém a definição de acesso, expanda o nó **Definições de Acesso** e clique duas vezes na definição de acesso. O Editor de Definição de Acesso se abre.
2. Selecione a guia **Propriedades de Grupo de Dados**.
3. Selecione uma coluna e especifique as opções de agrupamento.
4. Salve a definição de acesso.

Especificando Opções de Amostragem de Dados e Limite de Linha

Use a guia **Tabelas** no Editor de Definição de Acesso para especificar opções para extrair uma amostragem de linhas ou para limitar o número de linhas a extrair.

Se a lista Apontar e Disparar for usada para selecionar linhas da tabela de início, o serviço de extração ignorará qualquer amostra de dados ou parâmetros de limite de linha para a tabela de início.

Para especificar as opções de amostragem de dados e de limites de linha:

1. Expanda uma pasta no Repository Explorer que contém a definição de acesso, expanda o nó **Definições de Acesso** e clique duas vezes na definição de acesso. O Editor de Definição de Acesso se abre.
2. Selecione a guia **Tabelas**.
3. Insira as seguintes opções:

Cada Enésimo Número de Linhas

Inserir um valor numérico para especificar um fator de amostragem para uma tabela. Por exemplo, se você inserir 5, o processo extrai toda 5ª linha na tabela, iniciando com a 5ª linha. Os valores válidos são de 1 a 9999.

Limite de Linhas

Insira um valor numérico para limitar o número de linhas extraídas de uma tabela. Os valores válidos são de 1 a 999999999999.

4. Salve a definição de acesso.

Gerenciando Relacionamentos em uma Definição de Acesso

Use a guia **Relacionamentos** no Editor de Definição de Acesso para incluir ou remover os relacionamentos em uma definição de acesso. Também é possível especificar a passagem de relacionamento e as opções de acesso da tabela.

Selecionando as Opções de Passagem de Relacionamento

Use o Editor de Definição de Acesso para gerenciar as opções de passagem de relacionamento em uma definição de acesso.

É possível especificar as opções a seguir:

- O número máximo de linhas a partir da tabela-filho a ser selecionada para um relacionamento.
- O número máximo de consultas-chave executadas por vez para uma tabela.
Aumentar o limite de consulta chave pode aumentar significativamente o desempenho. Por exemplo, se você especificar 5 como o limite de consulta chave e a chave tiver uma única coluna, 5 valores chave serão procurados em uma única solicitação para DBMS.
- Os caminhos de passagem:
 - Atravesse o relacionamento de filho para pai, a fim de selecionar uma linha pai para cada linha filho para assegurar a integridade relacional. (Opção 1)
 - Atravesse o relacionamento para selecionar linhas filho adicionais para cada linha pai selecionada como resultado de uma passagem de filho para pai. (Opção 2)

As opções 1 e 2 são relevantes quando a tabela de início for uma tabela-filho ou quando uma tabela tiver mais de uma tabela-pai que seja referenciada pela definição de acesso.

A opção 2 é relevante apenas se você atravessar um relacionamento de filho para pai. Por exemplo, se um processo atravessar de filho para pai (opção 1) e uma linha pai for selecionada, a opção 2 fará com que o processo selecione as filhas filho adicionais para essa linha pai.

Se você selecionar a opção 2 para um relacionamento, consulte um limite filho no número de linhas filhas a serem extraídas.

Para gerenciar as opções de passagem de relacionamento em uma definição de acesso:

1. Expanda a pasta no Repository Explorer que contém a definição de acesso, expanda o nó **Definições de Acesso** e clique duas vezes na definição de acesso. O Editor de Definição de Acesso se abre.
2. Selecione a guia **Relacionamentos** e, em seguida, a guia **Opções de Passagem**.
3. Insira as opções de passagem.
4. Salve a definição de acesso.

Selecionando Opções de Acesso de Tabela

Use o Editor de Definição de Acesso para selecionar o método para acessar a tabela pai ou filho para cada relacionamento.

É possível especificar as opções a seguir:

Padrão

Optim determina o melhor método. Uma consulta chave é usada quando um índice DBMS fica disponível e uma varredura quando um índice não fica disponível. No entanto, se acessar uma parte significativa da tabela, o padrão é varrer, mesmo se existir um índice.

Forçar Varredura

Leia todas as linhas em uma tabela de uma vez.

Forçar Consulta Chave

Localize as linhas usando uma cláusula WHERE para procurar valores de chave primárias ou estrangeiras.

Nota: Substitua o método padrão apenas se as informações estatísticas no relatório do processo indicarem que o método padrão é menos eficiente.

Para gerenciar as opções de acesso de tabela:

1. Expanda a pasta no Repository Explorer que contém a definição de acesso, expanda o nó **Definições de Acesso** e clique duas vezes na definição de acesso. O Editor de Definição de Acesso se abre.
2. Selecione a guia **Relacionamentos** e, em seguida, a guia **Acesso de Tabela**.
3. Insira as opções de acesso de tabela.
4. Salve a definição de acesso.

Gerenciando Variáveis em uma Definição de Acesso

As variáveis são valores padrão definidos pelo usuário especificados em uma definição de acesso. É possível usar essas variáveis de substituição para criar uma Cláusula SQL WHERE.

Ao designar as variáveis, você fornece valores para as variáveis sempre que a definição de acesso é processada. Como opção, é possível fornecer valores padrão para as variáveis de substituição. As variáveis são salvas com a definição de acesso.

Valores Padrão

Ao criar uma variável, é possível inserir um valor padrão opcional a ser usado quando nenhum valor for especificado para a variável no tempo de execução.

Os valores padrão devem ser do tipo de dados apropriado e tamanho para a coluna e deve estar em conformidade com a sintaxe SQL. Por exemplo, imagine que um nome de variável seja **ST** (estado), o delimitador de variável seja dois pontos (:) e a coluna exija os dados de caractere.

- Se usar a variável com aspas simples nos critérios de seleção, você deve especificar o valor sem aspas simples:

Critérios de Seleção	Valor
= 'ST'	CA

- Se usar a variável sem aspas simples nos critérios de seleção, você deve especificar o valor com aspas simples:

Critérios de Seleção	Valor
= :ST	'CA'

Nota: Os valores padrão não são validados até o tempo de execução. Se um valor tiver um tipo ou tamanho de dados incorretos para a coluna ou não estiver em conformidade com a sintaxe SQL, erros de processamento poderão ocorrer.

Sequência de Prompt

Você deve inserir o texto que solicita o valor da variável no tempo de execução. Digite a sequência de prompt exatamente como deseja que apareça no diálogo de solicitação de processo (até 50 caracteres). Este prompt é exibido antes de executar o processo.

Criando uma Variável

Use a janela Incluir uma Variável para criar uma variável em uma definição de acesso.

Para criar uma variável:

1. Expanda a pasta no Repository Explorer que contém a definição de acesso, expanda o nó **Definições de Acesso** e clique duas vezes na definição de acesso. O Editor de Definição de Acesso se abre.
2. Selecione a guia **Variáveis**.
3. Clique em **Incluir**. A janela incluir uma Variável se abre.
4. Insira as informações para a variável. Você deve inserir um nome de variável e texto do prompt. Clique em **OK**.
5. Salve a definição de acesso.

Gerenciando as Listas de Apontar e Disparar

Use a lista de apontar e disparar para selecionar as linhas específicas a partir de uma tabela de início que será incluída em um serviço.

Uma lista de apontar e disparar é incluída em uma definição de acesso. Ao usar apontar e disparar para selecionar as linhas a partir da tabela de início, as chaves primárias para essas linhas serão armazenadas em um arquivo de apontar e disparar. A solicitação de serviço usa as chaves primárias para identificar as linhas para processar primeiro.

Criando um Arquivo Apontar e Disparar Lista

Use a janela do Novo Arquivo de Apontar e Disparar para criar um arquivo de apontar e disparar lista para uma tabela em um alias de armazenamento de dados.

Para criar um arquivo de apontar e disparar lista:

1. No Explorer de Repositório, expanda o nó **Aliases de Armazenamento de Dados** e o alias de armazenamento de dados de destino até que a tabela na qual o arquivo apontar e disparar será inserido seja exibido.
2. Expanda a tabela, clique com o botão direito em **Apontar e Disparar Listas** e clique em **Novo Apontar e Disparar**. A janela Novo Arquivo de Apontar e Disparar se abre.
3. Insira um nome de arquivo e clique em **OK**.

O arquivo de apontar e disparar lista é exibido em **Apontar e Disparar Listas**.

Você deve usar Apontar e Disparar Editor para selecionar as linhas ou apontar e disparar lista.

Selecionando Apontar e Disparar Listar

Use o Editor de Definição de Acesso para selecionar apontar e disparar lista para uma definição de acesso.

Para selecionar apontar e disparar lista, um arquivo de apontar e disparar deve estar disponível para a tabela de início.

Para selecionar apontar e disparar lista:

1. Expanda a pasta no Repository Explorer que contém a definição de acesso, expanda o nó **Definições de Acesso** e clique duas vezes na definição de acesso. O Editor de Definição de Acesso se abre.
2. Selecione a guia **Apontar e Disparar**.
3. Selecione apontar e disparar lista.
4. Salve a definição de acesso.

Trabalhando com Relacionamentos Optim

O Optim usa os relacionamentos para determinar os dados a serem recuperados das tabelas relacionadas e depende dos relacionamentos definidos para o banco de dados, quando disponíveis. No entanto, também é possível definir os relacionamentos Optim para complementar os relacionamentos no banco de dados.

Com relacionamentos Optim, inúmeras restrições de banco de dados são relaxadas. Por exemplo:

- As chaves primárias e as chaves estrangeiras não são necessárias.
- As colunas correspondentes não precisam ser idênticas, mas deve ser compatíveis.
- Pelo menos um par das colunas correspondentes deve ser especificado pelo nome da coluna. No entanto, é possível usar uma expressão para avaliar ou definir o valor na segunda coluna. As expressões podem incluir sequências literais, constantes numéricas, NULL, concatenação e subsequências.

Os relacionamentos Optim mais flexíveis são denominados relacionamentos “estendidos”. Os relacionamentos estendidos podem replicar os relacionamentos gerenciados pelo aplicativo ou implícitos em seu banco de dados, permitindo manipular os conjuntos de dados relacionais da mesma maneira que em um ambiente de produção.

Além disso, um relacionamento Optim pode ser armazenado no Optim Directory como:

- Um relacionamento **explícito**, usado para um único par de tabelas.
- Um relacionamento **genérico**, usado para um ou mais pares de tabelas que possuem o mesmo nome base, nome de colunas e atributos, mas diferentes IDs do Criador.

Os relacionamentos genéricos são úteis quando diversos conjuntos de tabela forem diferentes apenas pelo ID do Criador. (Por exemplo, em um ambiente de teste, cada programador pode usar uma cópia separada das mesmas tabelas de produção. Cada conjunto de tabelas pode ser distinto pelo ID do Criador.) Usando

relacionamentos genéricos, defina um conjunto dos relacionamentos que se aplica a todos os conjuntos de tabelas. Além disso, quando um conjunto dessas tabelas será incluído, os relacionamentos genéricos serão automaticamente aplicados.

Restrições

Embora as regras para criar os relacionamentos Optim sejam mais flexíveis do que as regras para criar os relacionamentos definidos pelo banco de dados, existem algumas restrições:

- Você deve fazer referência a pelo menos uma coluna para cada tabela no relacionamento.
- É possível referenciar um máximo de 64 colunas para qualquer tabela no relacionamento.
- Não é possível corresponder um literal ou uma constante para um literal ou uma constante.
- Não é possível usar a coluna da variante Objeto Grande (LOB) ou SQL.
- O comprimento total de todos os valores especificados na tabela pai ou tabela filha não podem exceder 3584 bytes.
- Não é possível criar um relacionamento usando uma Coluna da Variante SQL.

Em uma definição de Relacionamento para um banco de dados de diversos bytes ou Unicode:

- Não é possível usar a Função de Subsequência.
- Não é possível concatenar os dados de caractere (CHAR ou NCHAR) com binários (RAW).
- Se as semânticas de caractere Oracle forem usadas para alguma coluna CHAR, todas as colunas CHAR no relacionamento devem ter semânticas de caractere ou um tipo de dados NCHAR.

EXEMPLOS:

Pai	Suportado/Não Suportado	Filho	Descrição
CHAR	→	CHAR	Suportado, as semânticas devem ser correspondentes
NCHAR	→	NCHAR	Suportado, semânticas irrelevantes.
CHAR	↘	NCHAR	Não Suportado
CHAR	→	VARCHAR	Suportado, semânticas devem corresponder.
NCHAR	→	NVARCHAR	Suportado, semânticas irrelevantes.
CHAR NCHAR	→	NCHAR CHAR	Suportado, se semânticas de caractere; não suportado, se semânticas de byte
CHAR NCHAR	→	NCHAR NCHAR	Suportado, se semânticas de caractere; não suportado, se semânticas de byte

Regras de Compatibilidade para Relacionamentos

Ao definir um relacionamento Optim, os valores correspondentes devem ser compatíveis.

Tipo de Coluna	É Compatível Com
Coluna de Caracteres	<ul style="list-style-type: none">• Coluna de Caracteres• Coluna Numérica• Sequência Literal• Expressão de Caracteres
Coluna Numérica	<ul style="list-style-type: none">• Coluna Numérica• Constante Numérica• Coluna de Caracteres
Coluna Binária	<ul style="list-style-type: none">• Coluna Binária• Literal Hexadecimal• Expressão Binária
Coluna Booleana	<ul style="list-style-type: none">• Coluna Booleana• Constante Booleano (True ou False)
Coluna de Data Hora	Coluna de Data Hora
Coluna de Data	Coluna de Data
Coluna de Horário	Coluna de Horário
Coluna de Intervalo	Coluna de Intervalo

Nota:

- No processamento, um valor é convertido em tipo de dados necessário para selecionar as linhas relacionadas. Por padrão, o resultado de converter um valor numérico para um tipo de dados de caractere é justificado à direita com zeros à esquerda. As configurações especiais de registro permitem alterar o padrão para a justificação à esquerda com espaços à esquerda ou à direita. Além disso, um caractere para o emparelhamento numérico requer uma escala igual a 0 para a coluna numérica.
- É possível usar NULL para qualquer coluna nula elegível.
- As colunas Unicode ou multi-byte devem ser do mesmo conjunto de caracteres.

Tipos de Dados

As classes de dados a seguir e os tipos de dados associados são suportados. Essas classes de dados são importantes para a compatibilidade de dados quando você usa os valores de coluna nos relacionamentos.

Classe de Caractere

DBMS	Tipos de Dados
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

Nota: As colunas de caractere de byte único não são compatíveis com as colunas de caractere Unicode ou multi-byte.

Classe Numérica

DBMS	Tipos de Dados
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

Classe Binária

DBMS	Tipos de Dados
DB2	CHAR (para Dados de Bit), VARCHAR (para Dados de Bit), BLOB
Oracle	RAW, LONG RAW
Sybase ASE	BINARY, VARBINARY, IMAGE
SQL Server	BINARY, VARBINARY, IMAGE
Informix	BYTE

Classe Booleana

DBMS	Tipos de Dados
Sybase ASE	BOOLEAN (TRUE ou FALSE)

Data/Hora

DBMS	Tipos de Dados
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

Classe de Data

DBMS	Tipos de Dados
DB2	DATE
Oracle	DATE
Informix	DATE

Classe de Tempo

DBMS	Tipos de Dados
DB2	TIME

Classe de Intervalo

DBMS

Oracle

Informix

Tipos de Dados

YEAR/MONTH INTERVAL, DAY/SECOND INTERVAL

YEAR/MONTH INTERVAL, DAY/TIME INTERVAL

Criando um Relacionamento Optim

Use o assistente Novo Relacionamento Optim para criar um novo relacionamento Optim.

Antes de criar um relacionamento Optim, um alias de armazenamento de dados deve existir para as tabelas no relacionamento.

Para criar um relacionamento Optim:

1. No Repository Explorer, clique com o botão direito em **Relacionamentos Optim** e clique em **Novo Relacionamento Optim**. O assistente Novo Relacionamento Optim se abre.
2. Conclua as etapas do assistente.
Você deve selecionar as tabelas pai e filho no relacionamento.

O novo relacionamento Optim é exibido no Editor de Relacionamento.

Você deve usar o Editor de Relacionamento para definir uma expressão de coluna que identifica as colunas pai e filho no relacionamento.

Editando um Relacionamento Optim

Use o Editor de Relacionamento para definir uma expressão de coluna que identifica as colunas pai e filho em um relacionamento Optim. Também é possível usar o editor para criar um relacionamento genérico e editar colunas.

Criando uma Expressão da Coluna

Use a janela Incluir Expressão da Coluna para criar uma expressão da coluna que identifique as colunas pai e filho em um relacionamento Optim.

Para criar uma expressão da coluna em um relacionamento Optim:

1. No Repository Explorer, expanda o nó **Relacionamentos Optim** e clique duas vezes no relacionamento a ser editado. O Editor de Relacionamento se abre.
2. Clique em **Incluir Expressão da Coluna...** A janela Incluir Expressão de Coluna se abre.
3. Clique em **Selecionar Coluna Pai**. A janela Selecionar uma Coluna Pai se abre.
4. Selecione uma coluna pai e clique em **OK** para retornar à janela Incluir Expressão da Coluna.
5. Clique em **Selecionar Coluna Filho**. A janela Selecionar uma Coluna Filha se abre.
6. Selecione uma coluna filha e clique em **OK** para retornar à janela Incluir Expressão da Coluna.
7. Clique em **OK** para retornar ao editor. As colunas pai e filho são listadas no editor.
8. Salve o relacionamento.

Editando as Colunas em um Relacionamento

Use Selecionar uma Coluna Pai e Selecionar uma Coluna Filha para editar as colunas em um relacionamento Optim.

Para editar as colunas em um relacionamento Optim:

1. No Repository Explorer, expanda o nó **Relacionamentos Optim** e clique duas vezes no relacionamento a ser editado. O Editor de Relacionamento se abre.
2. Selecione a linha que contém uma coluna a ser editada. Para editar uma coluna pai:
 - a. Clique em **Selecionar Coluna Pai**. A janela Selecionar uma Coluna Pai se abre.

b. Selecione uma coluna pai e clique **OK** para retornar ao editor.

Para editar uma coluna filha:

a. Clique em **Selecionar Coluna Filho**. A janela Selecionar uma Coluna Filha se abre.

b. Selecione uma coluna filha e clique **OK** para retornar ao editor.

Os itens selecionados são listados no editor.

3. Salve o relacionamento.

Alterando a Ordem da Coluna em um Relacionamento

Use o Editor de Relacionamento para alterar a ordem das colunas em um relacionamento Optim.

Para alterar a ordem das colunas em um relacionamento Optim:

1. No Repository Explorer, expanda o nó **Relacionamentos Optim** e clique duas vezes no relacionamento a ser editado. O Editor de Relacionamento se abre.
2. Selecione a linha que contém as colunas a serem reordenadas.
3. Clique em **Mover Linha para Cima** ou **Mover Linha para Baixo** para alterar a ordem.
4. Salve o relacionamento.

Criando um Relacionamento Genérico

Use o Editor de Relacionamento para criar um relacionamento Optim genérico.

Alguns bancos de dados contêm conjuntos de tabelas que são idênticos, exceto para o ID do criador. Em vez de definir um relacionamento para cada conjunto de tabelas, é possível definir um relacionamento genérico que se aplica a todos os conjuntos de tabelas que tenham o mesmo nome base, independentemente do ID do criador. É possível modificar essas tabelas base em um relacionamento genérico.

Para criar um relacionamento Optim genérico:

1. No Repository Explorer, expanda o nó **Relacionamentos Optim** e clique duas vezes no relacionamento a ser editado. O Editor de Relacionamento se abre.
2. Clique em **Genérico**.
3. Salve o relacionamento.

Trabalhando com Chaves Primárias do Optim

Uma chave primária é a coluna ou as colunas que contêm os valores que identificam exclusivamente cada linha em uma tabela. Um banco de dados deve ter uma chave primária para Optim para inserir, atualizar, restaurar ou excluir os dados de uma tabela de banco de dados. O Optim usa as chaves primárias que são definidas para o banco de dados. No entanto, também é possível definir as chaves primárias Optim para complementar as chaves primárias no banco de dados.

Uma chave primária é necessária:

- Em qualquer tabela que é visitada mais de uma vez em um processo, por exemplo, uma tabela-filho que tenha duas ou mais tabelas pai referenciadas na definição de acesso.
- Para ativar o recurso de apontar e disparar para uma tabela de início.

Nota: Se uma chave primária não for definida e for necessária para executar uma tarefa específica, aparecerá uma mensagem de erro.

Tipos de Chaves Primárias Optim

É possível definir dois tipos de chaves primárias:

- Uma chave primária **explícita** se aplica a uma única tabela.

- Uma chave primária **genérica** se aplica a quaisquer tabelas que tenham o mesmo nome base, nomes de coluna e especificações de atributo, mas diferentes IDs do criador.

Não há diferença na função ou aparência entre as chaves primárias genéricas e explícitas. No entanto, se uma tabela tiver chaves de ambos os tipos, a chave primária explícita será usada.

Convenções de Nomenclatura

O nome completo de uma chave primária é igual ao nome completo da tabela de banco de dados para a qual é definida. Este nome consiste em: *alias.creatorid.tablename*.

alias Alias que identifica o banco de dados no qual a tabela reside (1 a 12 caracteres).

creatorid

ID do Criador designado à tabela (1 a 64 caracteres).

tablename

Nome da tabela base (1 a 64 caracteres).

Nota:

- O comprimento total combinado de colunas para uma chave primária é limitado a 3584 bytes.

Criando uma Chave Primária

Use o assistente Nova Chave Primária para criar uma nova chave primária para uma tabela.

Antes de criar uma chave primária, deve existir um alias de armazenamento de dados para a tabela.

Para criar uma chave primária:

1. No Repository Explorer, clique com o botão direito em **Chave Primária do Optim** e clique em **Nova Chave Primária**. A janela Nova Chave Primária se abre.
2. Conclua as etapas do assistente.
Você deve selecionar uma tabela.

A nova chave primária é exibida no editor Chave Primária.

Você deve usar o Editor de Chave Primária para selecionar as colunas de chave primária para a tabela.

Editando uma Chave Primária

Use o Editor de Chave Primária para editar uma chave primária para uma tabela. É possível selecionar as colunas-chave e definir uma chave primária genérica.

Selecionando Colunas de Chave Primária

Use o Editor de Chave Primária para selecionar as colunas de chave primária para uma tabela. Também é possível configurar a ordem das colunas-chave.

Para selecionar as colunas de chave primária para uma tabela:

1. No Repository Explorer, expanda o nó **Chaves Primárias Optim** e clique duas vezes no relacionamento a ser editado. O Editor de Chave Primária é aberto.
2. Na lista **Colunas Disponíveis**, selecione as colunas-chave.
3. Clique em >> para mover as colunas para a lista **Colunas-chave**.

Para configurar a ordem das colunas-chave:

- a. Na lista **Colunas-chave**, selecione uma coluna a ser movida.
- b. Clique em **Mover para Cima** para mover uma coluna para cima na ordem, ou **Mover para Baixo** para mover uma coluna para baixo na ordem.

4. Salve a chave primária.

Criando uma Chave Primária Genérica

Use o Editor de Chave Primária para criar uma chave primária Optim genérica.

Alguns bancos de dados contêm conjuntos de tabelas que são idênticos, exceto para o ID do criador. Em vez de definir uma chave primária explícita para cada tabela, defina uma chave primária genérica para todas as tabelas que tenha o mesmo nome base, independentemente do ID do criador.

Para criar uma chave primária Optim genérica:

1. No Repository Explorer, expanda o nó **Chaves Primárias Optim** e clique duas vezes no relacionamento a ser editado. O Editor de Chave Primária é aberto.
2. Clique em **Genérico**.
3. Salve a chave primária.

Trabalhando com Mapas da Tabela

Use um mapa da tabela para definir as especificações para correlacionar as tabelas de origem e destino de dados compatíveis. É possível mapear as tabelas que possuem diferentes nomes, modificar os nomes de tabela, excluir as tabelas de um processo ou incluir os mapas de coluna para maior controle sobre os dados.

Use um mapa da tabela para:

- Direcionar o posicionamento de dados em um serviço de conversão, inserção ou carregamento.
- Excluir uma ou mais tabelas de um serviço de conversão, inserção ou carregamento.
- Incluir um mapa da coluna.

Dependendo do processo para o qual você está usando um mapa da tabela, os dois conjuntos de tabelas são referenciados como tabelas de origem e tabelas de destino:

- **Tabelas de Origem** são tabelas extraídas que contêm dados a serem usados em um processo de conversão, inserção ou carregamento.
- **Tabelas de Destino** são as tabelas nas quais os dados são convertidos, inseridos ou carregados.

Nota: As tabelas correspondentes podem ter diferentes IDs do criador ou nomes.

Os mapas da tabela são armazenados no repositório Optim ou integrados em um serviço de gerenciamento de dados.

Como opção, é possível usar um mapa da coluna para qualquer par de tabelas em um mapa da tabela. Os mapas da coluna identificam e correspondem as colunas em um par de tabelas. Um mapa da coluna deve ser usado quando os nomes da coluna ou atributos forem desiguais, ou as transformações de dados forem necessárias.

Convenções de Nomenclatura

O nome completo de uma mapa da tabela está no formato: *identifier.name*.

identifier

O identificador designado ao mapa da tabela (1 a 8 caracteres).

name Nome designado ao mapa da tabela (1 a 12 caracteres).

É útil usar um conjunto lógico de convenções de nomenclatura para identificar o uso para cada um e para organizar as definições para fácil acesso.

Criando um Mapa da Tabela

Use o assistente Novo Mapa da Tabela para criar um novo mapa da tabela.

Antes de criar um mapa da tabela, um alias de armazenamento de dados do arquivo deve existir para a tabela de origem.

Para criar um mapa da tabela:

1. Expanda uma pasta no Repository Explorer para conter o mapa da tabela, clique com o botão direito em **Mapas da Tabela** e clique em **Novo Mapa da Tabela**. O assistente Novo Mapa da Tabela se abre.
2. Conclua as etapas do assistente.
Você deve selecionar um alias de armazenamento de dados do arquivo para a tabela de origem.

O novo mapa da tabela é exibido no Editor de Mapa da Tabela.

Você deve usar o editor para definir os armazenamentos de dados de destino e os esquemas antes de salvar o mapa da tabela.

Editando um Mapa da Tabela

Use o Editor de Mapa da Tabela para editar os dados de destino e incluir um mapa de coluna.

Definindo Dados de Destino Padrão em um Mapa da Tabela

Use o Editor de Mapa da Tabela para definir os armazenamentos de dados de destino padrão e esquemas em um mapa da tabela.

Antes de definir os dados de destino de mapa de tabela padrão, um alias de armazenamento de dados deve existir para os dados de destino.

Para definir os dados de destino padrão em um mapa da tabela:

1. Expanda uma pasta no Repository Explorer que contém o mapa de tabela, expanda o nó **Mapas da Tabela** e clique duas vezes no mapa da tabela. O Editor de Mapa da Tabela se abre.
2. Selecione a guia **Mapa da Tabela**.
3. Na área **Mapa do esquema e alias de armazenamento de dados**, selecione a linha que contém o armazenamento de dados de origem.
4. Para definir um alias de armazenamento de dados de destino, clique na célula **Alias de Armazenamento de Dados de Destino** e selecione o alias de armazenamento de dados de destino a partir da lista.
5. Para definir um esquema de destino, clique na célula **Esquema de Destino** e selecione o esquema de destino a partir da lista.
6. Salve o mapa da tabela.

Editando Dados de Destino em um Mapa da Tabela

Use o Editor de Mapa da Tabela para editar os armazenamentos de destino, esquemas e tabelas para as tabelas de origem em um mapa de tabela.

Antes de editar os dados de destino de mapa de tabela, um alias de armazenamento de dados de destino padrão deve existir para os dados de destino.

Para editar os dados de destino em um mapa da tabela:

1. Expanda uma pasta no Repository Explorer que contém o mapa de tabela, expanda o nó **Mapas da Tabela** e clique duas vezes no mapa da tabela. O Editor de Mapa da Tabela se abre.
2. Selecione a guia **Mapa da Tabela**.
3. Na área **Mapa da Tabela**, selecione a linha que contém a tabela de origem.

4. Para definir um alias de armazenamento de dados de destino, clique na célula **Armazenamento de Dados de Destino** e insira o nome de um alias de armazenamento de dados de destino.
5. Para definir um esquema de destino, clique na célula **Esquema de Destino** e selecione o esquema de destino a partir da lista.
6. Para definir uma tabela de destino, clique na célula **Tabela de Destino** e selecione a tabela de destino a partir da lista.
7. Salve o mapa da tabela.

Incluindo um Mapa da Coluna em um Mapa da Tabela

Use o Editor de Mapa da Tabela para incluir um mapa da coluna em um par de tabelas mapeadas.

Para incluir um mapa da coluna em um mapa da tabela:

1. Expanda uma pasta no Repository Explorer que contém o mapa de tabela, expanda o nó **Mapas da Tabela** e clique duas vezes no mapa da tabela. O Editor de Mapa da Tabela se abre.
2. Selecione a guia **Mapa da Tabela**.
3. Na área **Mapa da Tabela**, selecione a linha que contém as tabelas para o mapa da coluna.
4. Clique em **Incluir Mapa da Coluna**. A janela Novo Mapa da Coluna é aberta com uma lista de mapas da coluna que contém as tabelas selecionadas.
5. Selecione um mapa da coluna ou selecione **Criar um Novo Mapa da Coluna** para criar um mapa da coluna baseado nas tabelas selecionadas. Clique em **OK**.
 - a. Se você optar por criar um novo mapa da coluna, o Editor de Mapa da Coluna será aberto e preenchido com as colunas das tabelas selecionadas.
 - b. Edite e salve o novo mapa da coluna.
6. Salve o mapa da tabela.

Trabalhando com Mapas da Coluna

Um mapa da coluna fornece especificações necessárias para corresponder ou excluir as colunas do processamento em um serviço de gerenciamento de dados. Os serviços de conversão, inserção e carregamento devem referenciar um mapa da tabela, que podem referenciar um ou mais mapas de coluna. É possível usar um mapa da coluna para definir as transformações de dados com funções ou procedimentos de mapa da coluna.

Um mapa da coluna deve ser usado quando os nomes da coluna ou atributos forem desiguais, quando as transformações de dados forem necessárias ou quando excluir uma ou mais colunas do processamento. Um mapa da coluna referenciado em um serviço de conversão, inserção ou carregamento deve modificar os dados, datas da idade ou moeda de conversão. É possível definir as transformações de dados usando as funções ou os procedimentos de mapa da coluna.

Ao usar um novo mapa da coluna, você deve escolher um alias de armazenamento de dados do arquivo como origem para as colunas que deseja mapear. De maneira semelhante, você deve especificar um alias de armazenamento de dados para os dados de destino.

Os mapas da coluna armazenados no repositório Optim estão disponíveis para reutilização ou compartilhamento com outros usuários. Um mapa da coluna local é armazenado como parte de um serviço de gerenciamento de dados e, caso contrário, não fica disponível para outros serviços. Se o mapa da tabela associado for local para um serviço, o mapa da tabela e o mapa da coluna ficarão disponíveis apenas ao serviço específico.

Convenções de Nomenclatura

O nome completo de um mapa da coluna possui duas partes: *identifier.name*.

identifier

O identificador designado ao mapa da coluna (1 a 8 caracteres).

name Nome designado ao mapa da coluna (1 a 12 caracteres).

Ao criar os mapas de colunas, é útil usar um conjunto lógico de convenções de nomenclatura para identificar e organizar as definições para fácil acesso.

Regras de Compatibilidade para Mapas da Coluna

As classes de dados a seguir e os tipos de dados associados são suportados. Essas classes de dados são importantes para compatibilidade de dados quando você especifica os valores de coluna nos relacionamentos e mapas de coluna.

Caractere

DBMS	Tipos de Dados
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

Nota: As colunas de caractere de byte único não são compatíveis com as colunas de caractere Unicode ou multi-byte.

Numérico

DBMS	Tipos de Dados
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

Binário

DBMS	Tipos de Dados
DB2	CHAR (para Dados de Bit), VARCHAR (para Dados de Bit), BLOB
Oracle	RAW, LONG RAW
Sybase ASE	BINARY, VARBINARY, IMAGE
SQL Server	BINARY, VARBINARY, IMAGE
Informix	BYTE

Booleana

DBMS	Tipos de Dados
Sybase ASE	BOOLEAN (TRUE ou FALSE)

Data/Hora

DBMS	Tipos de Dados
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

Data

DBMS	Tipos de Dados
DB2	DATE
Oracle	DATE
Informix	DATE

Hora

DBMS	Tipos de Dados
DB2	TIME

Intervalo

DBMS	Tipos de Dados
Oracle	YEAR/MONTH INTERVAL, DAY/SECOND INTERVAL
Informix	YEAR/MONTH INTERVAL, DAY/TIME INTERVAL

Criando um Mapa da Coluna

Use o assistente Novo Mapa da Coluna para criar um novo mapa da coluna.

Antes de criar um mapa da coluna, um alias de armazenamento de dados do arquivo deve existir para os dados de origem e de destino.

Para criar um mapa da coluna:

1. Expanda uma pasta no Repository Explorer para conter o mapa da coluna, clique com o botão direito em **Mapas da Coluna** e clique em **Novo Mapa da Coluna**. O assistente Novo Mapa da Coluna se abre.
2. Conclua as etapas do assistente.
Você deve selecionar um alias de armazenamento de dados do arquivo e tabela para os dados de origem. Você também deve selecionar um alias de armazenamento de dados e tabela para os dados de destino.

O novo mapa da coluna é exibido no Editor de Mapa da Coluna.

Editando um Mapa de Coluna

Use o Editor do Mapa de Coluna para editar um mapa da coluna e aplicar as funções de mascaramento de dados.

Aplicando uma Função de Mascaramento de Dados

Use o Editor de Mapa da Coluna para aplicar e editar uma função para uma coluna.

Para aplicar uma função de máscara de dados:

1. Expanda uma pasta no Explorer de Repositório que contém o mapa da coluna, expanda o nó **Mapas da Coluna** e clique duas vezes no mapa da coluna. O Editor de Mapa da Coluna se abre.
2. Selecione a coluna para a política.
3. Clique em **Aplicar Função**. A janela Aplicar Função se abre.
4. Selecione a função a ser aplicada. Clique em **OK**. O nome da função aparece na coluna associada e o editor de função é aberto no Editor de Mapa de Coluna.
5. Dependendo da função, execute uma das etapas a seguir:
Selecione a guia de expressão de função e edite a expressão de função.
Selecione uma guia de opção e selecione opções para a função.
6. Salve o mapa da coluna.

Mapeando uma Coluna de Origem

Use o Editor de Mapa da Coluna para mapear uma coluna de origem para um destino. As colunas de origem e de destino com o mesmo nome e tipos de dados compatíveis são mapeadas automaticamente.

Para mapear uma coluna de origem:

1. Expanda uma pasta no Explorer de Repositório que contém o mapa da coluna, expanda o nó **Mapas da Coluna** e clique duas vezes no mapa da coluna. O Editor de Mapa da Coluna se abre.
2. Clique em uma coluna de origem e selecione um nome de coluna a partir da lista.
3. Salve o mapa da coluna.

Trabalhando com Procedimentos de Mapa da Coluna

O procedimento de mapa da coluna é usado para mascarar ou transformar dados em uma coluna quando um serviço é executado. Como o nome indica, você deve incluir os procedimentos de mapa da coluna em um mapa da coluna. É possível gravar procedimentos de mapa da coluna usando a linguagem de script Lua.

Criando um Procedimento de Mapa de Coluna:

Use o editor de script Lua para criar um procedimento de mapa da coluna.

Para criar um procedimento de mapa da coluna:

1. Expanda uma pasta no Explorer de Repositório que contém o mapa da coluna, expanda o nó **Mapas da Coluna** e clique duas vezes no mapa da coluna. O Editor de Mapa da Coluna se abre.
2. Selecione a coluna para procedimento de mapa da coluna.
3. Clique em **Incluir Procedimento**. O editor de script Lua é aberto.
4. Crie e salve o procedimento. A coluna associada ao procedimento exibe **Procedimento**.
5. Salve o mapa da coluna.

Editando um Procedimento de Mapa de Coluna:

Use o editor de script Lua para editar um procedimento de mapa da coluna.

Para editar um procedimento de mapa da coluna:

1. Expanda uma pasta no Explorer de Repositório que contém o mapa da coluna, expanda o nó **Mapas da Coluna** e clique duas vezes no mapa da coluna. O Editor de Mapa da Coluna se abre.
2. Selecione a coluna associada ao procedimento de mapa da coluna.

3. Clique em **Editar Procedimento**. O editor de script Lua é aberto.
4. Edite e salve o procedimento.
5. Salve o mapa da coluna.

Inserindo Valores para Parâmetros em um Procedimento de Mapa de Coluna:

Use o Editor de Mapa de Coluna para inserir valores para parâmetros em um procedimento de mapa de coluna.

Para inserir valores para parâmetros em um procedimento de mapa da coluna:

1. Expanda uma pasta no Explorer de Repositório que contém o mapa da coluna, expanda o nó **Mapas da Coluna** e clique duas vezes no mapa da coluna. O Editor de Mapa da Coluna se abre.
2. Selecione a coluna que contém o procedimento. O editor **Procedimento** é aberto no Editor de Mapa de Coluna.
3. No campo **Parâmetros do Procedimento**, insira valores separados por vírgula para os parâmetros. Por exemplo: abc, def.
4. Salve o mapa da coluna.

Capítulo 4. Projetando Serviços de Gerenciamento de Dados

Use um serviço de gerenciamento de dados para extrair, converter, carregar ou inserir os dados. Um serviço usa as definições de acesso para definir os dados a serem extraídos. Um serviço também usa mapas de tabela e mapas de coluna para mapear os dados de origem e de destino. É possível mascarar dados aplicando uma política de privacidade de dados a uma entidade processada por um serviço. Também é possível transformar as solicitações de processo em um diretório Optim nos serviços de gerenciamento de dados. Use Optim Manager no modo integrado para testar os serviços.

Trabalhando com Serviços de Extração

Use um serviço de extração para copiar um conjunto de linhas relacionadas de uma ou mais tabelas e salvar as linhas para um armazenamento de dados do arquivo.

O serviço de extração especifica o conjunto de parâmetros necessários para extrair os dados e as definições de objeto a partir das tabelas de origem e o armazenamento de dados do arquivo no qual armazenar as informações extraídas.

O serviço de extração sempre extrai as definições para as tabelas e as colunas. Essas definições são usadas para criaras as tabelas de destino, se necessário. Também é possível optar por extrair outras definições de objeto, incluindo as chaves primárias, os relacionamentos e os índices.

Convenções de Nomenclatura

O nome completo de um serviço de extração consiste em: *identifier.name*.

identifier

O identificador que serve como prefixo para o nome do serviço de extração (1 a 8 caracteres).

name Nome designado ao serviço de extração (1 a 12 caracteres).

Ao criar os serviços de extração, é útil usar um conjunto lógico de convenções de nomenclatura para identificar o uso para cada um e para organizar as definições para fácil acesso.

Criando um Serviço de Extração

Use o assistente Novo Serviço para criar um serviço de extração.

Um serviço de extração requer uma definição de acesso. Durante a criação de um serviço, é possível selecionar ou criar uma definição de acesso.

Para criar um serviço de extração:

1. Expanda uma pasta no Repository Explorer para conter o serviço de extração, clique com o botão direito do mouse em **Serviços** e clique em **Novo Serviço**. O assistente de serviço novo se abre.
2. Conclua as etapas do assistente.

Você deve selecionar ou criar uma definição de acesso. Você também deve inserir um nome para o armazenamento de dados do arquivo de destino.

O novo serviço de extração é exibido no Editor de Serviço de Extrato.

Editando um Serviço de Extração

Use o Editor de Serviço de Extração para especificar as opções de processamento de serviço de extração, incluindo os objetos de dados para extração, conversão, compactação de arquivo e substituições de variável.

Especificando Opções de Processamento de Extração

Use a guia **Propriedades de Serviço** no Editor de Serviço de Extração para especificar as opções de processamento de serviço de extração.

É possível especificar as opções de processamento de serviço de extração a seguir:

- Gerenciar o número de conexões com o banco de dados. Aumentar as conexões com o banco de melhora o desempenho ao processar grandes quantidades de dados, permitindo que vários encadeamentos extraiam as linhas simultaneamente.
- Limitar o número de linhas extraídas.
- Permitir anexos de arquivo.
- Incluir informações estatísticas no relatório de serviço de extração.

Para especificar as opções de processamento do serviço de extração:

1. Expanda uma pasta no Repository Explorer que contém o serviço de extração, expanda o nó **Serviços** e clique duas vezes no serviço de extração a ser editado. O Editor de Serviço de Extração se abre.
2. Selecione a guia **Propriedades do Serviço**.
3. Edite as opções de processamento de serviço.
4. Salve o serviço de extração.

Especificando Objetos e Agrupando Opções

Use a guia **Dados e Objetos** no Editor de Serviço de Extração para especificar os objetos de dados para extração.

É possível especificar as opções a seguir:

- Indicar se o serviço deve extrair apenas os dados, apenas os objetos ou ambos.
- Selecionar os objetos a serem extraídos.

Para especificar os objetos de dados para extração:

1. Expanda uma pasta no Repository Explorer que contém o serviço de extração, expanda o nó **Serviços** e clique duas vezes no serviço de extração a ser editado. O Editor de Serviço de Extração se abre.
2. Selecione a guia **Dados e Objetos**.
3. Edite as opções de processamento de serviço.
4. Salve o serviço de extração.

Especificando Opções de Conversão

Use a guia **Conversão** no Editor de Serviço de Extração para especificar as opções de conversão. É possível converter os dados que são extraídos no serviço.

É possível especificar as opções a seguir:

- Indicar se o serviço deve converter os dados extraídos.
- Indicar o número máximo de linhas descartadas, que não podem ser convertidas, a serem permitidas antes de parar o serviço.

Para especificar as opções de conversão:

1. Expanda uma pasta no Repository Explorer que contém o serviço de extração, expanda o nó **Serviços** e clique duas vezes no serviço de extração a ser editado. O Editor de Serviço de Extração se abre.

2. Selecione a guia **Conversão**.
3. Edite as opções de processamento de serviço.
4. Salve o serviço de extração.

Especificando as Opções de Compactação de Arquivo

Use a guia **Opções de Compactação de Arquivo** no Editor de Serviço de Extração para especificar as opções para compactar o armazenamento de dados do arquivo ou as tabelas extraídas.

É possível especificar as opções a seguir:

- Indicar se o arquivo ou as tabelas específicas estão compactadas.
- Para cada tabela, você pode optar por usar um limite de compactação ou compactar o máximo possível. O valor do limite é a quantidade mínima de redução do tamanho que você espera alcançar, compactando a tabela. Insira um valor no intervalo de 1 a 99 para configurar um valor do limite para essa tabela.

Os métodos de compactação a seguir estão disponíveis para um armazenamento de dados do arquivo:

Compactação Sequencial

Os dados são compactados como são extraídos e antes de serem gravados no armazenamento de dados do arquivo. A compactação sequencial possui E/S inferior, quando comparada com a pós-compactação, mas usa os recursos de banco de dados para a duração do serviço de extração.

A Compactação Sequencial requer menos recursos de armazenamento durante o processo de extração quando comparada à pós-compactação.

Pós-compactação

Os dados são extraídos e gravados em um armazenamento de dados de arquivo descompactado. Em uma segunda etapa, Optim lê o armazenamento de dados do arquivo descompactado, e grava uma versão compactada do armazenamento de dados do arquivo. O benefício da pós-compactação é que as conexões com o banco de dados são fechadas anteriormente com a pós-compactação do que com a compactação sequencial. No entanto, com a pós-compactação, o tempo total decorrido será aumentado porque o armazenamento de dados do arquivo descompactado deve ser fechado, lido e, em seguida, uma nova versão compactada será criada.

Os sites interessados na contenção do recurso de banco de dados podem achar a pós-compactação útil porque reduz o tempo em que os recursos do banco de dados são necessários. No entanto, a pós-compactação aumenta o tempo decorrido e os requisitos de armazenamento para processar o serviço de extração. Embora o armazenamento aumentado seja necessário durante a operação de compactação, esse armazenamento temporário necessário será liberado quando a compactação for concluída.

Para especificar as opções de compactação de arquivo:

1. Expanda uma pasta no Repository Explorer que contém o serviço de extração, expanda o nó **Serviços** e clique duas vezes no serviço de extração a ser editado. O Editor de Serviço de Extração se abre.
2. Selecione a guia **Opções de Compactação do Arquivo**.
3. Edite as opções de processamento de serviço.
4. Salve o serviço de extração.

Especificando Opções de Variável

Use a guia **Variáveis** no Editor de Serviço de Extração para substituir os valores padrão da variável.

Para especificar opções variáveis:

1. Expanda uma pasta no Repository Explorer que contém o serviço de extração, expanda o nó **Serviços** e clique duas vezes no serviço de extração a ser editado. O Editor de Serviço de Extração se abre.
2. Selecione a guia **Variáveis**.

3. Insira os valores para substituir os valores padrão.
4. Salve o serviço de extração.

Alterando uma Definição de Acesso para um Serviço de Extração

Use o assistente Alterar Definição de Acesso para alterar uma definição de acesso associada a um serviço de extração.

Para alterar uma definição de acesso associada a um serviço de extração:

1. Expanda uma pasta no Repository Explorer que contém o serviço de extração, expanda o nó **Serviços** e clique duas vezes no serviço de extração a ser editado. O Editor de Serviço de Extração se abre.
2. Clique em **Alterar**. O assistente Alterar Definição de Acesso é aberto.
3. Conclua as etapas do assistente.
Selecione uma definição de acesso ou escolha criar uma definição de acesso local.
4. Salve o serviço de extração.

Trabalhando com Serviços de Conversão

Use um serviço de conversão para transformar os dados em um armazenamento de dados de arquivo. É possível transformar os dados para assegurar a privacidade de dados ou para transformar sistematicamente os dados para atender a seus requisitos de teste de aplicativo.

Especifique um mapa da tabela para corresponder as tabelas no armazenamento de dados do arquivo de origem para as tabelas no armazenamento de dados de arquivo de destino ou para excluir as tabelas do serviço de conversão.

Use os mapas de coluna no mapa da coluna para especificar quais dados converter e como eles devem ser convertidos.

Convenções de Nomenclatura

Nota: O nome completo de um serviço de conversão consiste no seguinte: *identifier.name*.

identifier

O identificador que serve como prefixo para o nome do serviço de conversão (1 a 8 caracteres).

name Nome designado ao serviço de conversão (1 a 12 caracteres).

Ao criar o serviço de conversão, é útil usar um conjunto lógico de convenções de nomenclatura para identificar o uso para cada um e para organizá-los para fácil acesso.

Criando um Serviço de Conversão

Use o assistente Novo Serviço para criar um serviço de conversão.

Um serviço de conversão requer um mapa da tabela. Durante a criação de um serviço, é possível selecionar ou criar um mapa da tabela.

Para criar um serviço de conversão:

1. Expanda uma pasta no Repository Explorer para conter o serviço, clique com o botão direito do mouse em **Serviços** e clique em **Novo Serviço**. O assistente de serviço novo se abre.
2. Conclua as etapas do assistente.
Você deve selecionar ou criar um mapa da tabela. É possível inserir um nome para o armazenamento de dados de arquivo de destino ou optar por sobrescrever o armazenamento de dados do arquivo de origem.

O novo serviço de conversão é exibido no Editor de Serviço de Conversão.

Editando um Serviço de Conversão

Use o Editor de Serviço de Conversão para especificar as opções de processamento de serviço de conversão.

Especificando as Opções de Processamento de Conversão

Use a guia **Opções de Processamento** no Editor de Serviço de Conversão para especificar as opções de processamento de serviço de conversão.

É possível especificar as opções de processamento de serviço de conversão a seguir:

- Configurar um limite para descarte de linha:
- Compactar o arquivo de destino.
- Permitir anexos de arquivo.

Para especificar as opções de processamento de serviço de conversão:

1. Expanda uma pasta em Explorar Repositório que contenha o serviço de conversão, expanda o nó **Serviços** e clique duas vezes no serviço de conversão a ser editado. O Editor de Serviço de Conversão se abrirá.
2. Selecione a guia **Opções de Processamento**.
3. Edite as opções de processamento de serviço.
4. Salve o serviço de conversão.

Alterando um Mapa de Tabela para um Serviço de Conversão

Use o assistente Alterar Mapa da Tabela para alterar um mapa de tabela associado a um serviço de conversão.

Para alterar um mapa da tabela associado a um serviço de conversão:

1. Expanda uma pasta em Explorar Repositório que contenha o serviço de conversão, expanda o nó **Serviços** e clique duas vezes no serviço de conversão a ser editado. O Editor de Serviço de Conversão se abrirá.
2. Clique em **Alterar**. O assistente Alterar Mapa da Tabela é aberto.
3. Conclua as etapas do assistente.
Selecione um mapa da tabela ou escolha criar um mapa da tabela local.
4. Salve o serviço de conversão.

Trabalhando com Serviços de Inserção

Use um serviço de inserção para inserir os dados armazenados em um armazenamento de dados de arquivo em um banco de dados de destino.

Use os mapas de tabela e os mapas de coluna para mapear os dados da origem para o destino. Você deve usar um mapa da tabela para especificar o destino para as tabelas no armazenamento de dados do arquivo. Também é possível usar os mapas da coluna para uma ou mais tabelas de destino. Os mapas da coluna permitem especificar os dados de origem para cada coluna e, opcionalmente, transformar os dados antes que sejam inseridos.

Se as tabelas no armazenamento de dados do arquivo não existirem no destino, use o utilitário Criar para criá-las.

Convenções de Nomenclatura

O nome completo de um serviço de inserção consiste em: *identifier.name*.

identifier

O identificador que serve como prefixo para o nome do serviço de inserção (1 a 8 caracteres).

name Nome designado ao serviço de inserção (1 a 12 caracteres).

Ao criar os serviços de inserção, é útil usar um conjunto lógico de convenções de nomenclatura para identificar o uso para cada um e para organizá-los para fácil acesso.

Criando um Serviço de Inserção

Use o assistente Novo Serviço para criar um serviço de inserção.

Um serviço de inserção requer um mapa da tabela. Durante a criação de um serviço, é possível selecionar ou criar um mapa da tabela.

Para criar um serviço de inserção:

1. Expanda uma pasta no Repository Explorer para conter o serviço, clique com o botão direito do mouse em **Serviços** e clique em **Novo Serviço**. O assistente de serviço novo se abre.
2. Conclua as etapas do assistente.
Você deve selecionar ou criar um mapa da tabela.

O novo serviço de inserção é exibido no Editor de Serviço de Inserção.

Editando um Serviço de Inserção

Use o Editor de Serviço de Inserção para especificar as opções de processamento de serviço de inserção. É possível especificar as opções para inserir e excluir as linhas e também manipular os acionadores e restrições.

Especificando Opções de Processamento de Inserção

Use a guia **Opções de Processamento** no Editor de Serviço de Inserção para especificar as opções de processamento de serviço de inserção.

É possível especificar as opções de processamento a seguir:

- Selecione o tipo de processamento a ser executado. Especifique os parâmetros para bloquear as tabelas, configurar uma frequência de confirmação e configurar um limite de linha de descarte.
- Exclua as linhas para as tabelas todas ou especificadas. Excluir é útil para restaurar dados durante o teste. Se uma linha não puder ser excluída por algum motivo, todas as linhas excluídas até a última confirmação serão restauradas e o processamento de inserção vai parar.
- Desative os acionadores de banco de dados.
- Desative as restrições do banco de dados.

O processamento de linha inclui as opções a seguir:

Inserir

Inserir novas linhas nas tabelas.

- Se a chave primária de uma linha nos dados de origem *não corresponder* à chave primária de uma linha na tabela de destino, a linha será inserida.
- Se a chave primária de uma linha nos dados de origem *corresponder* à chave primária de uma linha na tabela de destino, será efetuado bypass da linha e ela será marcada como descartada.

Combinado

Atualiza, insere ou atualiza/insere de acordo com a sua seleção para cada tabela na janela

Especificar por Tabela. Para usar a janela Especificar por Tabela, clique em **Especificar por Tabela** e selecione uma opção de processo para cada tabela.

- Se você selecionar **Combinado** e não especificar as seleções na janela Especificar por Tabela ou configurar todas as tabelas para a mesma seleção, a opção de processo será alterada para indicar o processo usado para todas as tabelas.

Nota: Não é possível selecionar **Combinado** se **Todas as Linhas Serão Excluídas** estiver selecionado na área **Opções de Exclusão da Linha**.

Atualizar

Atualiza as linhas nas tabelas. As tabelas devem ter uma chave primária.

- Se uma chave primária de uma linha nos dados de origem *corresponder* à chave primária de uma linha na tabela de destino, a linha será atualizada.
- Se a chave primária de uma linha nos dados de origem *não corresponder* à chave primária de uma linha na tabela de destino, a linha será relatada como com falha.

Nota: Não é possível selecionar **Atualizar** se **Todas as Linhas Serão Excluídas** estiver selecionado na área **Opções de Exclusão de Linha**.

Atualizar/Inserir

Atualiza e insere as linhas nas tabelas. As tabelas devem ter uma chave primária.

- Se a chave primária de uma linha nos dados de origem *não corresponder* à chave primária de uma linha na tabela de destino, a linha será inserida.
- Se uma chave primária de uma linha nos dados de origem *corresponder* à chave primária de uma linha na tabela de destino, a linha será atualizada.

Nota: Não é possível selecionar **Atualizar/Inserir** se **Todas as Linhas Serão Excluídas** estiver selecionado na área **Opções de Exclusão da Linha**.

Para especificar as opções do processamento de inserção:

1. Expanda uma pasta no Repository Explorer que contém o serviço de inserção, expanda o nó **Serviços** e clique duas vezes no serviço de inserção a ser editado. O Editor de Serviço de Inserção se abre.
2. Selecione a guia **Opções de Processamento**.
3. Edite as opções de processamento de serviço.
4. Salve o serviço de inserção.

Alterando um Mapa da Tabela para um Serviço de Inserção

Use o assistente Alterar Mapa da Tabela para alterar um mapa da tabela associado a um serviço de inserção.

Para alterar um mapa da tabela associado a um serviço de inserção:

1. Expanda uma pasta no Repository Explorer que contém o serviço de inserção, expanda o nó **Serviços** e clique duas vezes no serviço de inserção a ser editado. O Editor de Serviço de Inserção se abre.
2. Clique em **Alterar**. O assistente Alterar Mapa da Tabela é aberto.
3. Conclua as etapas do assistente.
Selecione um mapa da tabela ou escolha criar um mapa da tabela local.
4. Salve o serviço de inserção.

Trabalhando com Serviços de Carregamento

Use um serviço de carregamento para transformar o conteúdo de um armazenamento de dados do arquivo em um formato que seja apropriado para um determinado carregador DBMS e, em seguida, se especificado, iniciar o utilitário de carregamento de banco de dados correspondente.

O serviço de carregamento gera um arquivo de dados no formato correto para cada tabela no armazenamento de dados do arquivo e um arquivo SQL ou um arquivo BAT (execução em lote), dependendo do DBMS, que contém a sintaxe necessária para iniciar o carregador do banco de dados.

Um serviço de carregamento contém os parâmetros usados para preparar os dados para um carregador DBMS e as instruções necessárias para processar o carregamento. Especifique um mapa da tabela no serviço de carregamento para mapear o destino para o dados a serem carregados. Use os mapas de coluna opcionais no serviço de carregamento para transformar os dados antes de carregar.

Os dados a serem carregados devem estar contidos em um armazenamento de dados do arquivo.

Carregamento versus Inserção

O Optim pode mover os dados para um banco de dados usando o serviço de carregamento ou de inserção. Considere o seguinte ao decidir qual método usar:

- O volume de dados e a velocidade de usar o utilitário de carregamento do banco de dados pode compensar as vantagens do serviço de inserção.
- Os dados podem conter os ciclos de integridade referencial (RI) que excedem a capacidade do serviço de inserção para inserir todos os dados bem-sucedidos.
- O utilitário de carregamento do banco de dados requer o controle exclusivo do banco de dados e evita o acesso do usuário durante o serviço de carregamento. O banco de dados fica disponível a outros usuários enquanto o serviço de inserção é executado.
- O utilitário de carregamento de banco de dados insere novos dados ou substitui os dados existentes. O serviço de inserção permite o processamento de atualização/inserção em uma etapa.

Nomes do Arquivo de Saída

Um serviço de carregamento gera os tipos a seguir de arquivos para suportar o processo de carregamento do utilitário de banco de dados:

Arquivos de Dados

Os arquivos de dados contêm os dados que você deseja carregar, preparados no formato apropriado para o DBMS que você está usando. O Optim gera um arquivo de dados para cada tabela no armazenamento de dados do arquivo. Os arquivos de dados são nomeados da mesma maneira que o armazenamento de dados do arquivo, mas contêm as extensões de nome do arquivo numerado sequencialmente. Por exemplo, um armazenamento de dados do arquivo denominado *demo.xf* que contém três tabelas vai gerar três arquivos denominados: *demo.001*, *demo.002* e *demo.003*.

Arquivos de Mensagens

Os arquivos de mensagens contêm informações que o utilitário de carregamento de banco de dados gera durante o serviço de carregamento. Geralmente, existe um arquivo de mensagens para o serviço de carregamento inteiro. O arquivo de mensagens é denominado da mesma maneira que o armazenamento de dados do arquivo, mas contém a extensão *.msg*. Por exemplo, se o armazenamento de dados do arquivo for denominado *demo.xf*, o arquivo de mensagens será nomeado *demo.msg*.

Arquivo SQL

Para DB2, um arquivo SQL é gerado com uma instrução para cada tabela de destino que contém a sintaxe do carregador para executar manualmente o carregador. O arquivo SQL é denominado da mesma maneira que o armazenamento de dados do arquivo, mas possui a extensão *.sql*.

Arquivo BAT

Para Oracle, Sybase ASE, SQL Server e Informix, é gerado um arquivo BAT que contém a sintaxe para executar manualmente o carregador para cada tabela. Um arquivo BAT é gerado para cada alias de armazenamento de dados especificado no mapa da tabela. Cada arquivo BAT reside no diretório com o armazenamento de dados do arquivo convertido correspondente. Se você optar

por executar manualmente o carregador, o arquivo BAT deve ser editado (no Notepad, por exemplo) para substituir uma sequência de oito pontos de interrogação com informações específicas de senha (exceto Informix).

Além disso, um arquivo de dados é gerado para cada arquivo de dados. Um arquivo de formato possui o mesmo nome que o arquivo de dados correspondentes, exceto se a extensão do nome do arquivo for diferente. Se houver menos de 500 tabelas a serem carregadas, a extensão do nome de arquivo do formato será 500 numericamente maior do que a extensão do nome de arquivo de dados. (Por exemplo, se houver três arquivos de dados denominados *demo.001*, *demo.002* e *demo.003*, os arquivos de formato correspondentes serão denominados *demo.501*, *demo.502* e *demo.503*, respectivamente.) Se houver mais de 500 tabelas, um algoritmo de geração de extensão de arquivo mais complexo será empregado.

Nota: Se o seu servidor de arquivos não permitir longos nomes de arquivo com mais de 8 caracteres e o armazenamento de dados do arquivo tiver um nome longo, o serviço de carregamento falhará. A melhor solução será evitar o uso de nomes de arquivo longos para os armazenamento de dados do arquivo. Se necessário, é possível copiar e renomear um arquivo antes de usá-lo para um serviço de carregamento.

Convenções de Nomenclatura

O nome completo de um serviço de carregamento consiste em: *identifier.name*.

identifier

O identificador que serve como prefixo para o nome do serviço (1 a 8 caracteres).

name Nome designado ao serviço (1 a 12 caracteres).

Ao criar os serviços de carregamento, é útil usar um conjunto lógico de convenções de nomenclatura para identificar o uso para cada um e para organizá-los para fácil acesso.

Criando um Serviço de Carregamento

Use o assistente Novo Serviço para criar um serviço de carregamento.

Um serviço de carregamento requer um mapa da tabela. Durante a criação de um serviço, é possível selecionar ou criar um mapa da tabela.

Para criar um serviço de carregamento:

1. Expanda uma pasta no Repository Explorer para conter o serviço de extração, clique com o botão direito do mouse em **Serviços** e clique em **Novo Serviço**. O assistente de serviço novo se abre.
2. Conclua as etapas do assistente.
Você deve selecionar ou criar um mapa da tabela.

O novo serviço de carregamento é exibido no Editor de Serviço de Carregamento.

Editando um Serviço de Carregamento

Use o Editor de Serviço de Carregamento para especificar as opções de processamento de carregamento. É possível especificar opções para o processo de carregamento e para o carregador DBMS associado a cada armazenamento de dados de destino.

Especificando Opções de Processamento de Carregamento

Use a guia **Opções de Processamento** no Editor de Serviço de Carregamento para especificar as opções de processamento de serviço de carregamento.

É possível especificar as opções a seguir:

- Selecione como executar diversos carregadores DBMS quando mais de um alias de armazenamento de dados for usado: paralelo (para executar diferentes carregadores DBMS ao mesmo tempo) ou em sequência (para executar diferentes carregadores DBMS, um após o outro).
- Pare um carregador DBMS, se ocorrer um erro. Se diversos carregadores de DBMS forem executados em sequência, o processamento parará para o carregador DBMS com um erro e todos os carregadores DBMS subsequentes.
- Desative os acionadores de banco de dados.
- Desative as restrições do banco de dados.

Para especificar as opções de processamento do serviço de carregamento:

1. Expanda uma pasta no Repository Explorer que contém o serviço de carregamento, expanda o nó **Serviços** e clique duas vezes no serviço de carregamento a ser editado. O Editor de Serviço de Carregamento se abrirá.
2. Selecione a guia **Opções de Processamento**.
3. Edite as opções de processamento de serviço.
4. Salve o serviço de carregamento.

Especificando Opções do Carregador DBMS

Use a guia **Opções de Carregamento** no Editor de Serviço de Carregamento para especificar as opções para cada carregador DBMS.

É possível especificar as opções a seguir:

- Um modo de processamento.
- As opções específicas para cada alias de armazenamento de dados de destino.

Dependendo do DBMS do armazenamento de dados do arquivo de destino selecionado, os modos de processamento a seguir estão disponíveis:

Inserir

Inserir as linhas do armazenamento de dados do arquivo de origem nas tabelas de destino vazias. Se as tabelas de destino contiverem dados, o carregador retornará um erro.

Substituir

Limpa e substitui todas as linhas existentes nas tabelas de destino pelas linhas do armazenamento de dados do arquivo de origem. (**Substituir** pode ser significativamente mais intenso em recursos do que **Truncar** porque nenhuma criação de log é executada.)

Anexar

Inserir as linhas do armazenamento de dados do arquivo de origem nas tabelas de destino. Se os valores de chave primários corresponderem, as linhas duplicadas serão descartadas ou inseridas na tabela de exceções (se especificado).

Truncar

Truncar é o mesmo que **Substituir**, mas o banco de dados não registra as linhas que estão sendo excluídas e Truncar requer que as restrições de RI são desativadas.

Para especificar as opções do carregador DBMS:

1. Expanda uma pasta no Repository Explorer que contém o serviço de carregamento, expanda o nó **Serviços** e clique duas vezes no serviço de carregamento a ser editado. O Editor de Serviço de Carregamento se abrirá.
2. Selecione a guia **Opções de Carregamento**.
3. Selecione o alias de armazenamento de dados de destino.
4. Edite as opções de processamento de serviço.
5. Salve o serviço de carregamento.

Alterando um Mapa da Tabela para um Serviço de Carregamento

Use o assistente Alterar Mapa da Tabela para alterar um mapa da tabela associado a um serviço de carregamento.

Para alterar um mapa da tabela associado a um serviço de carregamento:

1. Expanda uma pasta no Repository Explorer que contém o serviço de carregamento, expanda o nó **Serviços** e clique duas vezes no serviço de carregamento a ser editado. O Editor de Serviço de Carregamento se abrirá.
2. Clique em **Alterar**. O assistente Alterar Mapa da Tabela é aberto.
3. Conclua as etapas do assistente.
Selecione um mapa da tabela ou escolha criar um mapa da tabela local.
4. Salve o serviço de carregamento.

Testando um Serviço de Gerenciamento de Dados

Use o Optim Manager no modo integrado para testar um serviço de gerenciamento de dados.

Para testar um serviço de gerenciamento de dados:

1. Expanda uma pasta no Repository Explorer que contém o serviço de extração e expanda o nó **Serviço**.
2. Clique com o botão direito e clique em **Executar Serviço**. O Optim Manager é aberto e a janela Executar Serviço é exibida.
3. Clique em **Executar**. Para monitorar o progresso do serviço, use a guia **Monitoramento de Serviço**.

Capítulo 5. Mascarando os Dados

É possível mascarar dados como números de ID nacional, números de cartão de crédito, datas, valores numéricos e informações pessoais. Use um mapa da coluna para inserir uma função de mascaramento de dados ou crie um procedimento de mapa de coluna com um script LUA. Use um serviço de conversão para transformar os dados.

Aplicando uma Função de Mascaramento de Dados

Use o Editor de Mapa da Coluna para aplicar e editar uma função para uma coluna.

Para aplicar uma função de máscara de dados:

1. Expanda uma pasta no Explorer de Repositório que contém o mapa da coluna, expanda o nó **Mapas da Coluna** e clique duas vezes no mapa da coluna. O Editor de Mapa da Coluna se abre.
2. Selecione a coluna para a política.
3. Clique em **Aplicar Função**. A janela Aplicar Função se abre.
4. Selecione a função a ser aplicada. Clique em **OK**. O nome da função aparece na coluna associada e o editor de função é aberto no Editor de Mapa de Coluna.
5. Dependendo da função, execute uma das etapas a seguir:
Selecione a guia de expressão de função e edite a expressão de função.
Selecione uma guia de opção e selecione opções para a função.
6. Salve o mapa da coluna.

Funções de Mascaramento de Dados

As funções de mascaramento de dados fornecem vários métodos para transformar ou mascarar os dados sensíveis.

Funções de Consulta

Use as funções de consulta para selecionar os valores de uma tabela de consulta que são usadas para preencher a tabela de destino. Use as funções de Consulta e Consulta Hash para selecionar os valores baseados no valor de origem. Como alternativa, use a função de Consulta Aleatória para selecionar os valores de uma tabela de consulta sem considerar o valor de origem.

Função de Consulta

A Função de Consulta obtém o valor para a coluna de destino a partir de uma tabela de consulta, de acordo com o valor em uma coluna de origem. Existem dois formatos de Função de Consulta, coluna única e coluna múltipla.

O formato de coluna única insere um valor em uma coluna de destino única. O formato de múltiplas colunas insere os valores das colunas da tabela de consulta múltipla nas colunas de destino correspondentes.

É possível inserir a Função de Consulta de coluna múltipla para qualquer coluna de origem que será substituída por um valor de tabela de consulta, mas você deve editar o mapa de coluna para remover os nomes das demais colunas de origem que também serão substituídas.

O parâmetro *ignore* permite que você ignore a tabela de consulta e use um valor de origem quando uma linha em uma coluna de origem especificada contiver um valor especificado (NULL, SPACES (para colunas CHAR), ou VARCHAR de comprimento zero).

É possível usar o parâmetro *preserve* para ignorar a tabela de consulta e usar um valor de origem quando uma linha em uma coluna de origem especificada contiver um valor especificado (NULL, SPACES (para colunas CHAR), ou VARCHAR de comprimento zero). *preserve* também pode ser usado para ignorar a tabela de consulta se uma coluna de origem não contiver um valor.

Se uma correspondência não for localizada na tabela de consulta, será relatado um erro de conversão.

A sintaxe é:

```
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

Para a procura de coluna única, o nome da coluna da tabela de origem que contém o valor de procura (opcional). Se não for especificado, será usado o nome da coluna de destino.

SRCSEARCH=

Para procura de coluna múltipla, os nomes das colunas da tabela de origem que contém os valores de procura. Os nomes da coluna devem ser colocados entre parênteses e separados por vírgulas.

dest= Nomes das colunas da tabela de destino em que os valores da tabela de consulta estão inseridos. (Necessário para consulta de coluna múltipla.)

col1,coln

Nomes de coluna da tabela de destino. A ordem dos nomes de coluna devem corresponder às colunas da tabela de consulta no parâmetro *values=*.

lktablename

Nome da tabela de consulta. Você pode especificar o nome da tabela de consulta como *dbalias.creatorid.tablename*, *creatorid.tablename* ou *tablename*. Se você não qualificar totalmente o nome da tabela, os qualificadores para a tabela de destino serão usados.

LookupTableSearchcol

Para a consulta de única coluna, nome da coluna na tabela de consulta que contém um valor a ser correspondido em relação ao valor de procura na coluna de origem.

value Nome da coluna na tabela de consulta que contém o valor de procura convertida para ser inserido no destino. (Obrigatório para consulta de única coluna.)

values=

Nomes das colunas da tabela de consulta que contém os valores a serem inseridos no destino. (Necessário para consulta de coluna múltipla.)

col1,coln

Nomes de coluna da tabela de consulta. A ordem dos nomes de coluna devem corresponder às colunas da tabela de destino no parâmetro *dest=*.

cache | nocache

Especifique *cache* (padrão) para manter uma tabela de valores de consulta localizada na memória ou *nocache* para descartar os valores localizados. Usar o *cache* é mais rápido ao recuperar um valor várias vezes, mas requer memória extra.

ignore=

Lista de colunas de origem com valores que são inseridos no destino em vez do valor de consulta quando a coluna tiver uma linha com um valor indicado (nulo, espaços, zero ou varchar de comprimento zero).

col O nome da coluna de origem.

Para consulta de coluna única, insira apenas um nome da coluna.

Para consulta de coluna múltipla, a ordem dos nomes de coluna deve corresponde às colunas da tabela de destino no parâmetro *dest=*. O coluna de colunas deve ser igual às colunas no parâmetro *dest=* e pelo menos uma coluna deve incluir os valores. Para não especificar os valores para uma coluna, não insira um valor. Por exemplo, *coln()*.

null Ignore a tabela de consulta se a linha da coluna de origem tiver um valor nulo.

SPACES

Ignore a tabela de consulta se a linha da coluna de origem tiver um valor SPACES. Apenas para colunas CHAR.

ZERO_LEN

Ignore a tabela de consulta se a linha da coluna de origem tiver um valor VARCHAR de comprimento zero.

preserve=

Lista de colunas de origem com valores que são inseridos no destino em vez do valor de consulta quando a coluna de origem contiver um valor indicado (NOT_FOUND, nulo, espaço ou varchar de comprimento zero).

NOT_FOUND

Ignore a tabela de consulta se nenhuma correspondência for localizada para a linha da coluna de origem.

Nota:

preserve= e *ignore=* são mutuamente exclusivos. *ignore=* será descontinuado em uma liberação futura.

Os operandos *col*, *null*, *spaces* e *zero_len* possuem o mesmo efeito quando usados com *preserve=* ou *ignore=*.

Exemplo de Coluna Única

Use a Função de Consulta para converter o valor de origem em uma tabela de consulta para um valor correspondente em outra tabela.

Por exemplo, suponha que a coluna de origem, STATE, contenha abreviações de estado (por exemplo, NJ) e a coluna de destino deva conter o nome completo do estado (neste exemplo, **New Jersey**). Uma tabela de consulta denominada STATE_LOOKUP contém uma coluna (CODE) para as abreviações de estado ou códigos e uma coluna (NAME) para os nomes correspondentes.

Para obter o valor para a coluna de destino usando a tabela STATE_LOOKUP, especifique:

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

As Função de Consulta procura um valor na coluna CODE da tabela STATE_LOOKUP que corresponde ao valor (NJ) na coluna STATE da tabela de origem. Quando uma correspondência for localizada, a função vai inserir o valor correspondente da coluna NAME (**New Jersey**) na coluna de destino.

Exemplo de Coluna Múltipla

Use a Função de Consulta para inserir valores de colunas em uma linha da tabela de consulta nas colunas em uma linha da tabela de destino, com base em um valor em uma coluna de origem.

Por exemplo, com base em uma coluna de origem (SOC_SEC) que contém números de seguridade social, é possível substituir os valores nas colunas de destino (FIRST_NAME e LAST_NAME) com nomes e sobrenomes de uma tabela de consulta. Uma tabela denominada NAME_LOOKUP contém uma coluna (SSN) com números de seguridade social a partir da tabela de origem bem como colunas (FIRST_MASK e LAST_MASK) para mascarar os nomes correspondentes no destino.

Para substituir os nomes na tabela de destino com base em um número de seguridade social, especifique:

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

A Função de Consulta procura um valor na coluna SSN da tabela NAME_LOOKUP que corresponde ao valor na coluna SOC_SEC da tabela de origem. Quando uma correspondência é localizada, a função insere os valores correspondentes das colunas FIRST_MASK e LAST_MASK da tabela de consulta nas colunas de destino correspondentes.

Exemplo Ignorar

Use a instrução a seguir para estender o exemplo de coluna única, em que deseja usar os valores *NULL* e *SPACES* de origem em vez dos valores da tabela de consulta:

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

Exemplo de NoCache

Use a instrução a seguir para estender o exemplo de coluna única, em que não deseja manter uma tabela dos valores de consulta localizados na memória:

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

Função de Consulta Hash

A Função de Consulta Hash obtém o valor para a coluna de destino a partir de uma tabela de consulta, de acordo com um valor hash derivado de uma coluna de origem. A Função de Consulta Hash permite mascarar consistentemente os dados quando você usar as mesmas tabelas de origem e de consulta em qualquer ambiente.

A coluna de origem na qual foi executado hash não precisa ser uma coluna que será substituída por valores da tabela de consulta.

A Função de Consulta Hash faz distinção entre maiúsculas e minúsculas. Por exemplo, os valores de origem John e JOHN serão colocados em hash para valores diferentes. É possível usar o parâmetro TRIM para converter o valor de origem para maiúsculas antes que seja colocado em hash.

Existem dois formatos de Função de Consulta Hash, coluna única e coluna múltipla. O formato de coluna única insere um valor em uma coluna de destino única. O formato de múltiplas colunas insere os valores das colunas da tabela de consulta múltipla nas colunas de destino correspondentes, com base em um único valor hash a partir da coluna de origem

É possível inserir a Função de Consulta Hash de coluna múltipla para qualquer coluna de origem que será substituída pelos valores da tabela de consulta, mas você deve editar o mapa de coluna para remover os nomes das demais colunas de origem que também serão substituídas.

A tabela de consulta deve incluir uma coluna-chave que contenha valores de números sequenciais sem quaisquer intervalos e colunas restantes que contenham valores de substituição. A coluna-chave deve ser do tipo de dados numéricos. A tabela de consulta é tipicamente indexada. A função executa o hash em

uma coluna de origem para derivar números sequenciais de 1 até o valor máximo na coluna-chave da tabela de consulta. O valor em que foi executado o hash, a partir da tabela de origem corresponde aos números sequenciais na tabela de consulta e os valores da linha da tabela de consulta correspondente são inseridos no destino.

Se a coluna de origem usada para derivar o valor do hash contém certos valores (NULO, espaços (para colunas CHAR), VARCHAR de comprimento zero), o valor não é hash e os seguintes valores reservados seguintes são usados como chaves para a tabela de consulta:

Valor de Origem	Chave da Tabela de Consulta
NULL	-1
espaços (CHAR ou VARCHAR)	-2
VARCHAR de comprimento zero	-3

A tabela de consulta deve incluir uma linha para cada um destes números, permitindo inserir um valor de consulta para cada um destes valores de origem. Se um destes valores de origem for localizado e um número correspondente não estiver na tabela de consulta, será relatado um erro de conversão.

O parâmetro *ignore* permite que você ignore a tabela de consulta e use um valor de origem quando uma linha em uma coluna de origem especificada contiver um valor especificado (NULL, SPACES (para colunas CHAR), ou VARCHAR de comprimento zero).

É possível usar o parâmetro *preserve* para ignorar a tabela de consulta e usar um valor de origem quando uma linha em uma coluna de origem especificada contiver um valor especificado (NULL, SPACES (para colunas CHAR), ou VARCHAR de comprimento zero). *preserve* também pode ser usado para ignorar a tabela de consulta se uma coluna de origem não contiver um valor.

O parâmetro *trim* permite especificar os caracteres que serão cortados do valor de origem antes que o hash seja executado neles. Por exemplo, se você optar por cortar as vírgulas de um valor de origem, os valores Smith, John e Smith John serão colocados em hash para o mesmo valor. Também é possível usar esse parâmetro para converter o valor de origem para maiúsculas antes que seja colocado em hash.

Se o valor de origem for convertido em maiúsculas, os caracteres de recorte também são convertidos em maiúscula.

É possível usar o parâmetro *seed* para varia o cálculo executado pelo algoritmo hash. O valor do hash da coluna de origem e o valor *seed* são correspondido a um número sequencial da tabela de consulta para obter o valor de substituição para a coluna de destino.

A sintaxe é:

```
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

Nome da coluna da tabela de origem do qual são derivados os valores em hash (opcional). Se não for especificado, será usado o nome da coluna de destino.

trim= Lista de caracteres a serem recortados a partir do valor de origem antes que seja colocado em hash bem como uma opção para converter o valor de origem em maiúscula antes que seja colocado em hash. Se o valor de origem resultante for NULL ou se todos os espaços após os caracteres tiverem sido cortados, o valor de origem não será executado em hash e o valor reservado apropriado será designado (-1 ou -2).

char1char2...

Caracteres que serão cortados do valor de origem antes que o hash seja executado. A lista faz distinção entre maiúsculas e minúsculas. É possível especificar um espaço ou vírgula como um caractere. Após a ocorrência inicial de um caractere, todas as ocorrências adicionais na lista serão ignoradas.

Para especificar uma barra invertida “\” ou um parêntese direito “)”, você deve preceder o caractere com um caractere de escape de barra invertida. Por exemplo, para especificar um parêntese direito, digite: *trim=(\)*.

É possível usar apenas o caractere de escape com uma barra invertida, um parêntese direito ou como parte do indicador de maiúscula.

\u Indica que o valor de origem deve ser convertido em maiúscula antes de ser colocado em hash. Os caracteres a serem cortados também são convertidos em maiúsculos.

dest= Nomes das colunas da tabela de destino em que os valores da tabela de consulta estão inseridos. (Necessário para consulta de coluna múltipla.)

col1,coln

Nomes de coluna da tabela de destino. A ordem dos nomes de coluna devem corresponder às colunas da tabela de consulta no parâmetro *values=*.

lktablename

Nome da tabela de consulta. Você pode especificar o nome da tabela de consulta como **dbalias.creatorid.tablename**, **creatorid.tablename** ou **tablename**. Se você não qualificar totalmente o nome da tabela, os qualificadores para a tabela de destino serão usados.

search Nome da coluna na tabela de consulta que contém valores sequenciais para corresponder aos valores hash da coluna de origem.

value Nome da coluna na tabela de consulta que contém o valor de procura convertida para ser inserido no destino. (Obrigatório para consulta de única coluna.)

values=

Nomes das colunas na tabela de consulta que contém os valores a serem inseridos no destino. (Necessário para consulta de coluna múltipla.)

col1,coln

Nomes de coluna da tabela de consulta. A ordem dos nomes de coluna devem corresponder às colunas da tabela de destino no parâmetro *dest=*.

cache | nocache

Especifique *cache* (padrão) para manter uma tabela de valores de consulta localizada na memória ou *nocache* para descartar os valores localizados. Usar o *cache* é mais rápido ao recuperar um valor várias vezes, mas requer memória extra.

ignore=

Lista de colunas de origem com valores que são inseridos no destino em vez do valor de consulta quando a coluna tiver uma linha com um valor indicado (NULL, SPACES, ZERO ou VARCHAR de comprimento zero).

col O nome da coluna de origem.

Para consulta de coluna única, insira apenas um nome da coluna.

Para consulta de coluna múltipla, a ordem dos nomes de coluna deve corresponde às colunas da tabela de destino no parâmetro *dest=*. O coluna de colunas deve ser igual às colunas no parâmetro *dest=* e pelo menos uma coluna deve incluir os valores. Para não especificar os valores para uma coluna, não insira um valor. Por exemplo, *coln()*.

NULL Ignore a tabela de consulta se a linha da coluna de origem tiver um valor NULL.

SPACES

Ignore a tabela de consulta se a linha da coluna de origem tiver um valor SPACES.
Apenas para colunas CHAR.

ZERO_LEN

Ignore a tabela de consulta se a linha da coluna de origem tiver um valor VARCHAR de comprimento zero.

preserve=

Lista de colunas de origem com valores que são inseridos no destino em vez do valor de consulta quando a coluna contiver um valor indicado (NOT_FOUND, nulo, espaço ou varchar de comprimento zero).

NOT_FOUND

Ignore a tabela de consulta se nenhuma correspondência for localizada para a linha da coluna de origem.

Nota:

preserve= e *ignore=* são mutuamente exclusivos. *ignore=* será descontinuado em uma liberação futura.

Os operandos *col*, *null*, *spaces* e *zero_len* possuem o mesmo efeito quando usados com *preserve=* ou *ignore=*.

seed= Use *seed=* para varia o cálculo do algoritmo hash. Os valores de 1 a 2.000.000.000 podem ser usados. Se você usar um valor 0, o parâmetro *seed=* será ignorado.

Exemplo de Coluna Única

Use a Função de Consulta Hash para inserir valores de uma coluna em uma tabela de consulta em uma coluna da tabela de destino, com base em um valor com hash a partir da coluna de origem.

Por exemplo, suponha que a coluna de origem, FIRST_NAME, contém os primeiros nomes e a coluna de destino incluirá os primeiros nomes de substituição a partir da tabela de consulta. Uma tabela de consulta, NAME_LOOKUP, contém uma coluna (FIRST) com os primeiros nomes e uma coluna (SEQ) que contém valores sequenciais.

Para obter valores para a coluna de destino usando a tabela NAME_LOOKUP, especifique:

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

A Função de Consulta Hash corresponde os valores do hash a partir da coluna de origem com valores na coluna SEQ da tabela NAME_LOOKUP. Quando uma correspondência é localizada, a função insere o valor correspondente a partir da PRIMEIRA coluna na coluna de destino.

Exemplo de Coluna Múltipla

Use a Função de Consulta Hash para inserir valores de colunas em uma linha da tabela de consulta nas colunas em uma linha da tabela de destino, com base no valor hash a partir da coluna de origem.

Por exemplo, com base nos valores hash a partir de uma coluna de origem (FIRST_NAME) que contém o primeiro nome, você pode substituir os valores nas colunas de destino (FIRST e LAST) pelos nomes e sobrenomes de uma tabela de consulta. Uma tabela de consulta denominada NAME_LOOKUP contém uma coluna (SEQ) com valores sequenciais bem como colunas (FIRST_MASK e LAST_MASK) para mascarar os valores no destino.

Para substituir nomes na tabela de destino com base nos valores hash a partir de uma coluna de origem, especifique:

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

A Função de Consulta Hash corresponde os valores do hash a partir da coluna FIRST_NAME de origem com valores na coluna SEQ da tabela NAME_LOOKUP. Quando uma correspondência é localizada, a função insere os valores correspondentes das colunas FIRST_MASK e LAST_MASK da tabela de consulta nas colunas de destino correspondentes.

Exemplo Ignorar

Use a instrução a seguir para estender o exemplo de coluna única, em que deseja usar os valores NULL e SPACES de espaço em vez dos valores de tabela de consulta:

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

Exemplo de NoCache

Use a instrução a seguir para estender o exemplo de coluna única, em que não deseja manter uma tabela dos valores de consulta localizados na memória:

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

Exemplo Trim

Use a instrução a seguir para estender o exemplo de coluna única, em que deseja cortar os espaços e as vírgulas do valor de origem bem como converter o valor de origem em maiúscula antes de colocá-lo em hash:

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

Função de Consulta Aleatória

A Função de Consulta Aleatória seleciona um valor aleatoriamente a partir de uma tabela de consulta especificada para inserir em uma coluna de destino. A função gera um número aleatório entre 1 e o limite ou número de linhas na tabela de consulta para usar como uma inscrição na tabela. O valor da coluna ou os valores da linha que correspondem à inscrição são inseridos na coluna de destino.

Existem dois formatos de Função de Consulta Aleatória, coluna única e coluna múltipla. O formato de coluna única insere um valor em uma coluna de destino única. O formato de múltiplas colunas insere os valores das colunas da tabela de consulta múltipla nas colunas de destino correspondentes.

É possível inserir a Função de Consulta Aleatória de coluna múltipla para qualquer coluna de origem que será substituída por um valor de tabela de consulta, mas você deve editar o mapa de coluna para remover os nomes das colunas de origem restantes que também serão substituídas.

O parâmetro *ignore* permite que você ignore a tabela de consulta e use um valor de origem quando uma linha em uma coluna de origem especificada contiver um valor especificado (NULL, SPACES (para colunas CHAR), ou VARCHAR de comprimento zero).

É possível usar o parâmetro *preserve* para ignorar a tabela de consulta e usar um valor de origem quando uma linha em uma coluna de origem especificada contiver um valor especificado (NULL, SPACES (para colunas CHAR), ou VARCHAR de comprimento zero).

A sintaxe é:

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

lktablename

Nome da tabela de consulta. Você pode especificar o nome da tabela de consulta como

dbalias.creatorid.tablename, *creatorid.tablename* ou *tablename*. Se o nome da tabela de destino não estiver completo, os qualificadores da tabela de destino serão usados.

columnname

O nome da coluna na tabela de consulta que contém os valores a serem selecionados aleatoriamente para inserção no destino. (Obrigatório para consulta de única coluna.)

dest= Nomes das colunas da tabela de destino em que os valores da tabela de consulta estão inseridos. (Necessário para consulta de coluna múltipla.)

col1,coln

Nomes de coluna da tabela de destino. A ordem dos nomes de coluna devem corresponder às colunas da tabela de consulta no parâmetro *values=*.

values=

Nomes das colunas na tabela de consulta que contém os valores a serem inseridos no destino. (Necessário para consulta de coluna múltipla.)

col1,coln

Nomes de coluna da tabela de consulta. A ordem dos nomes de coluna devem corresponder às colunas da tabela de destino no parâmetro *dest=*.

limit O limite opcional sobre o número de linhas na tabela de consulta usada para selecionar os valores da coluna. Especifique um número inteiro, até um valor máximo de 2.000.000.000. Se nenhum limite for especificado, todas as linhas serão usadas.

Nota: Uma tabela dos valores de coluna é gerada na memória. O tamanho desta tabela pode ser limitado pelos recursos do sistema.

ignore=

Lista de colunas de origem com valores que são inseridos no destino em vez do valor de consulta quando a coluna tiver uma linha com um valor indicado (NULL, SPACES ou VARCHAR de comprimento zero).

col O nome da coluna de origem.

Para consulta de coluna única, insira apenas um nome da coluna.

Para consulta de coluna múltipla, a ordem dos nomes de coluna deve corresponde às colunas da tabela de destino no parâmetro *dest=*. O coluna de colunas deve ser igual às colunas no parâmetro *dest=* e pelo menos uma coluna deve incluir os valores. Para não especificar os valores para uma coluna, não insira um valor. Por exemplo, *coln()*.

null Ignore a tabela de consulta se a linha da coluna de origem tiver um valor NULL.

spaces Ignore a tabela de consulta se a linha da coluna de origem tiver um valor SPACES. Apenas para colunas CHAR.

zero_len

Ignore a tabela de consulta se a linha da coluna de origem tiver um valor VARCHAR de comprimento zero.

preserve=

Lista de colunas de origem com valores que são inseridos no destino em vez do valor de consulta quando a coluna contiver um valor indicado (NOT_FOUND, nulo, espaço ou varchar de comprimento zero).

NOT_FOUND

Ignore a tabela de consulta se nenhuma correspondência for localizada para a linha da coluna de origem.

Nota:

preserve= e *ignore=* são mutuamente exclusivos. *ignore=* será descontinuado em uma liberação futura.

Os operandos *col*, *null*, *spaces* e *zero_len* possuem o mesmo efeito quando usados com *preserve=* ou *ignore=*.

Exemplo de Coluna Única

Para selecionar um valor aleatoriamente a partir da coluna STATE nas primeiras 50 linhas de uma tabela denominada STATE_LOOKUP e inseri-la na coluna de destino, especifique:

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

Exemplo de Coluna Múltipla

Para selecionar os valores a partir das colunas CITY, STATE e ZIPCODE em uma linha aleatória de uma tabela denominada STATE_LOOKUP e inseri-las nas colunas de destino correspondentes, especifique:

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

Exemplo Ignorar

Use a instrução a seguir para estender o exemplo de coluna única, em que a coluna de origem é denominada STATES e você deseja usar os valores NULL e SPACES de espaço em vez dos valores de tabela de consulta:

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

Função Ordenar Aleatoriamente

A Função Ordenar Aleatoriamente substitui um valor de origem por outro valor a partir da coluna que é inserida em uma coluna de destino. A linha de origem e a linha que contém o valor de substituição nunca serão as mesmas, mas dependendo dos seus dados, os valores de origem e de substituição podem ser idênticos.

É possível indicar o número de vezes que a função buscará um valor de substituição até que um valor que não corresponda ao valor de origem seja localizado (um “tentar novamente”), ou é possível permitir um valor de substituição para corresponder à origem. Cada Função Ordenar Aleatoriamente opera independentemente de outras Funções Ordenar Aleatoriamente usadas em um mapa de coluna.

Existem dois formatos de Função Ordenar Aleatoriamente, coluna única e coluna múltipla. O formato de coluna única insere um valor de substituição em uma coluna de destino única. O formato de múltiplas colunas insere os valores de substituição a partir de múltiplas colunas em uma linha nas colunas de destino correspondentes. A coluna não pode ser incluída em mais de uma Função Ordenar Aleatoriamente em um mapa da coluna. Se o recurso de nova tentativa for usada com uma coluna múltipla para ordenar aleatoriamente, a função buscará novamente outra linha de substituição se algum valor na coluna da linha de origem corresponder ao valor em uma coluna da linha de substituição correspondente. (O formulário de coluna múltipla não pode ser usado em uma Função de Propagação.)

Para criar uma coluna múltipla Função Ordenar Aleatoriamente, insira a função para uma coluna de origem que será substituída pelos valores ordenados aleatoriamente e edite o mapa da coluna para remover os nomes de qualquer outra coluna de origem com valores que também serão substituídos.

O parâmetro *ignore* evita que a função substitua uma linha de origem ou use uma linha de substituição se contiver um valor especificado (NULL, SPACES (para colunas CHAR) ou VARCHAR de comprimento zero). Se nenhuma nova tentativa for permitida, o parâmetro *ignore* não será aplicado à linha de substituição.

A sintaxe é:

```
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] ) ) ) ]
```

em que:

dest= Os nomes das colunas da tabela de destino em que os valores de substituição estão inseridos. (Necessário para coluna múltipla para ordenar aleatoriamente.)

col1, coln, ...

Nomes de coluna da tabela de destino.

retry Número de vezes para buscar novamente um valor de substituição para localizar um valor que não corresponde à linha de origem. Insira zero para permitir que um valor de substituição corresponda à origem.

Nota: Usar um valor de nova tentativa com colunas que contenham vários valores duplicados aumentará o tempo de processamento. Para essas colunas pode ser melhor usar um valor de nova tentativa de zero.

=number

Insira um valor no intervalo 0-1000. Insira 0 para permitir que um valor de substituição corresponda à origem.

ignore=

Lista de colunas para as quais a função não substituirá um valor de origem ou não usará um valor de substituição se for um valor especificado (NULL, SPACES (para colunas CHAR) ou VARCHAR de comprimento zero). e um valor de substituição for ignorado, a função buscará novamente outro valor de substituição. Se nenhuma nova tentativa for permitida, o parâmetro *ignore=* não se aplicará aos valores de substituição.

col O nome da coluna de origem.

Para ordenação aleatória de coluna única, insira apenas um nome da coluna.

Para ordenação aleatória de coluna múltipla, a ordem dos nomes de coluna deve corresponder às colunas da tabela de destino no parâmetro *dest=*. O nome de colunas deve ser igual às colunas no parâmetro *dest=* e pelo menos uma coluna deve incluir os valores. Para não especificar os valores para uma coluna, não insira um valor. Por exemplo, *coln()*.

null Não substitua o valor de origem ou use um valor de substituição, se for um valor NULL.

spaces Não substitua o valor de origem ou use um valor de substituição, se for um valor SPACES. Apenas para colunas CHAR.

zero_len

Não substitua o valor de origem ou use um valor de substituição, se for um valor VARCHAR de comprimento zero.

Exemplo Padrão de Coluna Única

O exemplo a seguir insere os valores ordenados aleatoriamente em uma única coluna.

```
SHUFFLE
```

Exemplo de Nova Tentativa Única

O exemplo a seguir insere os valores ordenados aleatoriamente em uma única coluna e busca novamente um valor de substituição que não corresponde à origem até 12 vezes.

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

Exemplo de Coluna Múltipla

O exemplo a seguir insere os valores ordenados aleatoriamente nas colunas STATE e ZIP e busca novamente um valor de substituição que não corresponde à origem até 12 vezes.

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

Exemplo Ignorar

O exemplo a seguir insere os valores ordenados aleatoriamente nas colunas STATE e ZIP e busca novamente um valor de substituição que não corresponde à origem até 12 vezes. O exemplo também não substitui um valor de origem ou usa o valor de substituição para a coluna STATE se uma origem ou linha de substituição contiver um espaço NULL ou SPACES, mas não ignorar qualquer origem ou linha de substituição para a coluna ZIP.

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

Função TRANS SSN

Use a função TRANS SSN para gerar um Número de Seguridade Social (SSN) dos EUA válido e exclusivo. Por padrão, o TRANS SSN gera algoritmicamente um SSN de destino consistentemente alterado com base no SSN de origem. O TRANS SSN também pode gerar um SSN aleatório quando os dados de origem não tiverem um valor SSN ou quando não houver necessidade de transformar o SSN de origem de uma maneira consistente.

Um SSN é formado por 3 subcampos. Os 3 primeiros dígitos (área) representam uma área geralmente determinada pelo estado no qual o SSN foi emitido. Os 2 dígitos seguintes (grupo) definem um número de grupo correspondente ao número da área. Os 4 últimos dígitos (serial) são um número de série sequencial. Independentemente do tipo de processamento, padrão ou aleatório, o TRANS SSN vai gerar um SSN com um número de grupo apropriado para o número da área.

O método de processamento padrão gera um SSN que inclui o número da área de origem bem como os números de série e grupo alterado com base no SSN de origem.

O método de processamento aleatório gera um SSN que pode incluir o número da área de origem e usa um número de grupo emitido mais recentemente pela Administração da Seguridade Social para o número da área de destino. Os números de série iniciam com 0001 e são incrementados por 1 para cada SSN adicional gerado para o número da área. Quando o número de série exceder 9999, o número de série será reconfigurado para 0001 e o número do grupo que precede o número emitido mais recentemente para o número da área será usado.

A sintaxe de TRANS SSN é:

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

flags É possível especificar um ou mais sinalizadores de opção de processamento sem distinção entre maiúsculas e minúsculas.

n Gere um SSN aleatório que não seja baseado em um valor de origem.

m Use o grupo máximo de todos os valores da área SSN, incluindo os valores de 773 a 899, e excluindo os números inválidos da área.

r Gere um SSN com um número de área aleatório correspondente ao mesmo estado que o SSN de entrada.

v Valide o número do grupo de origem comparando-o com os números usados pela Administração de Seguridade Social.

- O SSN de destino deve incluir traços que separam os campos (por exemplo, 123-45-6789). Requer uma coluna de destino de tipo de caractere com, pelo menos, 11 caracteres de comprimento.

sourcecol

O nome da coluna de origem. Se um nome de coluna de origem não estiver especificado, o nome da coluna de destino será usado. Se um nome da coluna de origem não estiver especificado e o nome da coluna de destino não corresponder a um nome de coluna na tabela de origem, ocorrerá um erro durante o processamento.

preserve=invalid

Se a coluna de origem contiver um SSN inválido, não o substitua por um valor gerado. O valor da coluna de origem será usado na coluna de destino.

Tipos de Dados Permitidos

Os tipos de dados de origem e de destino a seguir são permitidos:

CHAR

O comprimento de dados na coluna deve ser de 9 a 256 caracteres.

DECIMAL

A precisão da coluna deve ser de 9 a 20 e a escala 0.

INTEGER

Sem restrições.

VARCHAR

O comprimento de dados na coluna deve ser de 9 a 254 caracteres.

Se uma coluna de origem ou de destino não aderir a essas restrições, ocorrerá um erro durante o processamento.

Regras de Processamento de Destino

As regras a seguir se aplicarão ao valor SSN de destino, de acordo com o valor ou tipo de dados de destino:

CHAR

Se o valor de origem for 0, espaços ou VARCHAR de comprimento zero, o valor de destino será configurado para os espaços.

Se um valor de origem for de 11 caracteres ou mais e incluir traços integrados (-), ou se o sinalizador '-' for especificado, o valor de destino incluirá os traços se o comprimento da coluna de destino for de 11 caracteres ou mais.

DECIMAL, INTEGER

Se o valor de origem for 0, espaços ou um VARCHAR de comprimento zero, o valor de destino será 0.

VARCHAR

Se o valor de origem for 0, espaços ou um VARCHAR de comprimento zero, o comprimento de destino será 0.

Se um valor de origem for de 11 caracteres ou mais e incluir traços integrados (-), ou se o sinalizador '-' for especificado, o valor de destino incluirá os traços se o comprimento da coluna de destino for de 11 caracteres ou mais.

NULL Se o valor de origem for NULL, o valor de destino será NULL.

Linhas Ignoradas

As condições a seguir podem fazer com que uma linha de origem seja ignorada e não gravada para o destino:

- O valor de origem é NULL e a coluna de destino não permite um valor NULL.
- A coluna de origem é CHAR ou VARCHAR, e o valor de origem é inferior a 9 caracteres, contém um caractere não numérico (diferente dos traços entre os 3 subcampos) ou é muito grande.
- O número da área de origem não foi usado pela Administração de Seguridade Social.
- O número do grupo de origem não foi usado com o número de área pela Administração de Seguridade Social (apenas se o sinalizador 'v' tiver sido especificado).
- O número de série de origem é 0000, ou o SSN é um valor reservado não emitido pela Administração de Seguridade Social (por exemplo, 078-05-1120).
- O valor de origem não pode ser convertido em um formato que o TRANS SSN suporta.

Mensagens de Erro

As mensagens de erro a seguir podem ser emitidas:

SSN01

Parm on Col cccc ("ppp") é inválido

Explicação

A coluna contém uma função TRANS com um sinalizador de opção de processamento que não é válido.

Ação do Usuário

Assegure que a função TRANS na coluna especificada usa um sinalizador de opção de processamento válido (n, r, v, -).

SSN02

Col cccc não na origem

Explicação

A coluna que foi inserida como um parâmetro *sourcecol* ou nome de coluna de destino (se o parâmetro *sourcecol* tiver sido omitido) não foi localizado na tabela de origem.

Ação do Usuário

Verifique a tabela de origem e resolva qualquer discrepância ou coluna ausente.

SSN03

Source Col cccc-aaa inválida

Explicação

O formato da coluna de origem não é suportado porque o atributo indicado não é válido.

Ação do Usuário

Verifique a coluna de origem e assegure que os valores para tipo, comprimento, precisão e escala sejam apropriados.

SSN04

Dest Col cccc-aaa inválido

Explicação

O formato da coluna de destino não é suportado porque o atributo indicado não é válido.

Ação do Usuário

Verifique a coluna de destino para assegurar que os valores para tipo, comprimento, precisão e escala sejam apropriados.

SSN05

Obter col cccc data-rc=nnn

Explicação

Ocorreu um erro interno inesperado ao obter o valor da coluna de origem.

Ação do Usuário

Verifique os valores das colunas de origem e de destino e assegure que os valores para tipo, comprimento, precisão e escala sejam apropriados. Se o problema persistir, entre em contato com o Suporte ao Software IBM.

SSN08

Colocar col cccc data-rc=nnn

Explicação

Ocorreu um erro interno inesperado ao configurar o valor na coluna de destino.

Ação do Usuário

Verifique os valores das colunas de origem e de destino e assegure que os valores para tipo, comprimento, precisão e escala sejam apropriados. Se o problema persistir, entre em contato com o Suporte ao Software IBM.

Se ocorrer algum outro erro, entre em contato com o Suporte Técnico.

Exemplo 1

O exemplo a seguir usa um nome de coluna de origem que corresponde a coluna de destino e gera um SSN aleatório que não é baseado no valor de origem:

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

Exemplo 2

O exemplo a seguir usa um nome de coluna de origem (NATIONAL_ID) que difere da coluna de destino e gera um SSN usando o método de processamento padrão e incluindo traços:

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

Função TRANS CCN

Use a função TRANS CCN para gerar um número de cartão de crédito (CCN) exclusivo e válido. Por padrão, o TRANS CCN gera algoritmicamente um CCN consideravelmente alterado com base no CCN de origem. O TRANS CCN também pode gerar um valor aleatório quando os dados de origem não tiverem um valor CCN ou quando não houver necessidade de transformar o CCN de origem de uma maneira consistente.

Um CCN, conforme definido por ISO 7812, consiste em um identificador do emissor de 6 dígitos seguido por um número de conta de comprimento variável e em um único dígito de verificação como o número final. O dígito de verificação verifica a exatidão do CCN e é gerado transmitindo o identificador do emissor e números de contas por meio do algoritmo Luhn. O comprimento máximo de um CCN é de 19 dígitos.

O método de processamento padrão gera um CCN incluindo os 4 primeiros dígitos do identificador do emissor a partir do CCN de origem e alterando os 2 dígitos restantes do número do identificador do emissor e do número da conta baseados no CCN de origem. Um dígito de verificação válido também é designado.

O método de processamento aleatório gera um CCN que podem incluir os 4 primeiros dígitos do número do identificador do emissor de origem ou um número de identificador do emissor designado ao American Express, Discover, MasterCard ou VISA. Um dígito de verificação válido também é designado. Se os quatro primeiros dígitos de um número identificador do emissor de origem forem incluídos, o primeiro número de conta baseado nesses dígitos iniciará com 1, e para cada CCN adicional que usa esses dígitos, o número da conta será incrementado por 1.

A sintaxe de TRANS CCN é:

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

flag Especifique um sinalizador de opção para gerar um CCN aleatório.

n Gere um CCN que não seja baseado em um valor de origem e que inclua um número de identificador do emissor designado a American Express, Discover, MasterCard ou VISA.

r Gere um CCN aleatório que inclua os 4 primeiros dígitos do número do identificador do emissor de origem.

sourcecol

O nome da coluna de origem. Se um nome de coluna de origem não estiver especificado, o nome da coluna de destino será usado.

Se um nome da coluna de origem não estiver especificado e o nome da coluna de destino não corresponder a um nome de coluna na tabela de origem, ocorrerá um erro durante o processamento.

preserve=invalid

Se a coluna de origem contiver um CCN inválido, não o substitua por um valor gerado. O valor da coluna de origem será usado na coluna de destino.

Tipos de Dados Permitidos

Os tipos de dados de origem e de destino a seguir são permitidos:

CHAR

O comprimento de coluna deve ser de 13 a 256 caracteres.

VARCHAR

O comprimento de coluna deve ser de 13 a 254 caracteres.

DECIMAL

A precisão da coluna deve ser de 13 a 254 e a escala 0.

Se uma coluna de origem ou de destino não aderir a essas restrições, ocorrerá uma mensagem de erro.

Regras de Processamento de Destino

As regras a seguir se aplicarão ao valor CCN de destino, de acordo com o valor ou tipo de dados de destino:

CHAR

Se o valor de origem for de espaços ou um VARCHAR de comprimento zero, o valor de destino será configurado para os espaços.

VARCHAR

Se o valor de origem for de espaços ou um VARCHAR de comprimento zero, o comprimento de destino será 0.

DECIMAL

Se o valor de origem for 0, o valor de destino será 0.

NULL Se o valor de origem for NULL, o valor de destino será NULL.

Linhas Ignoradas

As condições a seguir podem fazer com que uma linha de origem seja ignorada e não gravada para o destino:

- O valor de origem é NULL e a coluna de destino não permite um valor NULL.
- O valor de origem é inferior a 13 caracteres, contém um caractere não numérico, é muito grande ou possui um dígito de verificação incorreto.
- O comprimento de valor de origem não é válido para o emissor do cartão de crédito.
- O valor de origem não pode ser convertido em um formato que o TRANS CCN suporta.

Mensagens de Erro

As mensagens de erro a seguir podem ser emitidas:

CCN01

Parm on Col cccc ("ppp") é inválido

Explicação

A coluna indicada contém uma função TRANS com um sinalizador de opção de processamento que não é válido.

Ação do Usuário

Assegure que a função TRANS na coluna especificada usa um sinalizador de opção de processamento válido (n, r, 6).

CCN02

Col cccc não na origem

Explicação

A coluna que foi inserida como um parâmetro *sourcecol* ou nome de coluna de destino (se o parâmetro *sourcecol* tiver sido omitido) não foi localizado na tabela de origem.

Ação do Usuário

Verifique a tabela de origem e resolva qualquer discrepância ou coluna ausente.

CCN03

Source Col cccc-aaa inválida

Explicação

O formato da coluna de origem não é suportado porque o atributo indicado não é válido.

Ação do Usuário

Verifique a coluna de origem e assegure que os valores para tipo, comprimento, precisão e escala sejam apropriados.

CCN04

Dest Col cccc-aaa inválido

Explicação

O formato da coluna de destino não é suportado porque o atributo indicado não é válido.

Ação do Usuário

Verifique a coluna de destino para assegurar que os valores para tipo, comprimento, precisão e escala sejam apropriados.

CCN05

Obter col ccccc data-rc=nnn

Explicação

Ocorreu um erro interno inesperado ao obter o valor da coluna de origem.

Ação do Usuário

Verifique os valores das colunas de origem e de destino e assegure que os valores para tipo, comprimento, precisão e escala sejam apropriados. Se o problema persistir, entre em contato com o Suporte ao Software IBM.

CCN08

Colocar col ccccc data-rc=nnn

Explicação

Ocorreu um erro interno inesperado ao configurar o valor na coluna de destino.

Ação do Usuário

Verifique os valores das colunas de origem e de destino e assegure que os valores para tipo, comprimento, precisão e escala sejam apropriados. Se o problema persistir, entre em contato com o Suporte ao Software IBM.

Se ocorrer algum outro erro, entre em contato com o Suporte Técnico.

Exemplo 1

O exemplo a seguir usa um nome de coluna de origem (CREDITCARD) que é diferente da coluna de destino e gera um CCN aleatório não baseado no valor de origem.

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

Exemplo 2

O exemplo a seguir usa um nome de coluna de origem (CREDITCARD) que é diferente da coluna de destino e gera um CCN que usa o método de processamento padrão.

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

Função TRANS EML

Use a função TRANS EML para gerar um endereço de email. Um endereço de email consiste em duas partes, um nome de usuário seguido por um nome de domínio, separado por '@'. Por exemplo, user@domain.com.

O TRANS EML gera um endereço de email com um nome de usuário baseado em dados de destino ou um literal concatenado com um número sequencial. O nome do domínio pode ser baseado em um endereço de email nos dados de origem, um literal, ou selecionados aleatoriamente a partir de uma lista de provedores de serviços de email grandes. O endereço de email também pode ser convertido em maiúsculas ou minúsculas.

O TRANS EML pode gerar um nome de usuário baseado nos valores em uma ou duas colunas de tabela de destino (geralmente contendo o nome de um usuário). As opções de processamento permitem usar

apenas o primeiro caractere do valor na primeira coluna (por exemplo, a letra inicial de um nome) e separar os valores de ambas as colunas usando um ponto e um sublinhado.

Se o nome do usuário for baseado em um único valor de coluna de destino ou um literal, o nome será concatenado com um número sequencial. Se um nome de usuário for baseado nos valores em duas colunas de tabela de destino e um ponto de separação ou sublinhado não for usado, os valores serão concatenados. Se um parâmetro não for fornecido para o nome de usuário, o nome será formado pelo "email" literal concatenado com um número sequencial. Os números sequenciais para os nomes de usuário são sufixos que iniciam com 1 e são incrementados por 1.

A sintaxe de TRANS EML é:

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

flags É possível especificar um ou mais sinalizadores de opção de processamento sem distinção entre maiúsculas e minúsculas.

n Gere um nome de domínio aleatório a partir de uma lista de provedores de serviço de email grande.

. Separe os valores *name1col* e *name2col* por um ponto.

_ Separe os valores *name1col* e *name2col* com um sublinhado.

i Use apenas o primeiro caractere do valor *name1col*.

l Converta o endereço de email em letras maiúsculas e minúsculas.

u Converta o endereço de email em maiúsculas.

sourcecol

O nome da coluna de origem com os endereços de email usados para fornecer o nome de domínio.

Se o sinalizador 'n' e o parâmetro *domain* não estiverem definidos, o nome de domínio na coluna de origem será usado. (Se *sourcecol* não estiver definido, o nome da coluna de origem será baseado no nome da coluna de destino.)

Se um nome da coluna de origem não estiver especificado e o nome da coluna de destino não corresponder a um nome de coluna na tabela de origem, ocorrerá um erro durante o processamento.

domain Um literal, com até 31 caracteres, que forma o nome de domínio.

, Será necessário uma vírgula se o parâmetro *sourcecol* e *domain* não estiverem definidos e você definir um literal ou nome(s) de coluna para o nome de domínio.

name1col

Um nome da coluna da tabela de destino com os valores usados para formar a primeira (ou única) parte do nome do usuário.

name2col

Um nome da coluna da tabela de destino com os valores usados para formar a segunda parte do nome do usuário.

userpfx Um literal, com até 31 caracteres, que é concatenado com um número sequencial para formar o nome de usuário.

preserve=invalid

Se a coluna de origem contiver um endereço de email inválido, não o substitua por um valor gerado. O valor da coluna de origem será usado na coluna de destino.

Tipos de Dados Permitidos

Os tipos de dados de origem e de destino a seguir são permitidos:

CHAR

O comprimento de coluna deve ser de 3 a 4096 caracteres.

VARCHAR

O comprimento de coluna deve ser de 3 a 4094 caracteres.

Se uma coluna de origem ou de destino não aderir a essas restrições, uma mensagem de erro será emitida.

Regras de Processamento de Destino

As regras a seguir se aplicarão ao valor de email de destino, de acordo com o valor ou tipo de dados de destino:

CHAR

Se o valor de origem for de espaços ou um VARCHAR de comprimento zero, o valor de destino será configurado para os espaços.

VARCHAR

Se o valor de origem for de espaços ou um VARCHAR de comprimento zero, o comprimento de destino será 0.

NULL Se o valor de origem for NULL, o valor de destino será NULL.

Linhas Ignoradas

As condições a seguir podem fazer com que uma linha de origem seja ignorada e não gravada para o destino:

- O valor de origem é NULL e a coluna de destino não permite um valor NULL.
- O valor de origem é um VARCHAR inferior a 3 caracteres de comprimento.
- O valor de email de origem não contém um '@'.
- O valor de origem não pode ser convertido em um formato que o TRANS EML suporta.

Mensagens de Erro

As mensagens de erro a seguir podem ser emitidas:

EML01

Parm on Col cccc ("ppp") é inválido

Explicação

A coluna indicada contém uma função TRANS com um sinalizador de opção de processamento que não é válido.

Ação do Usuário

Assegure que a função TRANS na coluna especificada usa um sinalizador de opção de processamento válido (n, ., -, i, l, u).

EML02

Col cccc não na origem

Explicação

A coluna que foi inserida como um parâmetro *sourcecol* ou nome de coluna de destino (se o parâmetro *sourcecol* tiver sido omitido) não foi localizado na tabela de origem.

Ação do Usuário

Verifique a tabela de origem e resolva qualquer discrepância ou coluna ausente.

EML03

Source Col ccccc-aaa inválida

Explicação

O formato da coluna de origem não é suportado porque o atributo indicado não é válido.

Ação do Usuário

Verifique a coluna de origem e assegure que os valores para tipo, comprimento, precisão e escala sejam apropriados.

EML04

Dest Col ccccc-aaa inválido

Explicação

O formato da coluna de destino não é suportado porque o atributo indicado não é válido.

Ação do Usuário

Verifique a coluna de destino para assegurar que os valores para tipo, comprimento, precisão e escala sejam apropriados.

EML05

Obter col ccccc data-rc=nnn

Explicação

Ocorreu um erro interno inesperado ao obter o valor da coluna de origem.

Ação do Usuário

Verifique os valores das colunas de origem e de destino e assegure que os valores para tipo, comprimento, precisão e escala sejam apropriados. Se o problema persistir, entre em contato com o Suporte ao Software IBM.

EML08

Colocar col ccccc data-rc=nnn

Explicação

Ocorreu um erro interno inesperado ao configurar o valor na coluna de destino.

Ação do Usuário

Verifique os valores das colunas de origem e de destino e assegure que os valores para tipo, comprimento, precisão e escala sejam apropriados. Se o problema persistir, entre em contato com o Suporte ao Software IBM.

EML09

Literal de domínio sssss muito longo

Explicação

A sequência especificada como o literal de nome de domínio (domínio) excede o limite máximo de 31 caracteres.

Ação do Usuário

Especifique um nome de domínio que consiste em 31 caracteres ou menos.

EML10

Literal do usuário sssss muito longo

Explicação

A sequência especificada como o literal de nome de usuário (*userpfx*) excede o limite máximo de 31 caracteres.

Ação do Usuário

Especifique um nome de usuário que consiste em 31 caracteres ou menos.

EML11

Name1 Col ccccc não em dest

Explicação

Para executar a função TRANS indicada, uma coluna *name1col* deve ser especificada na tabela de destino.

Ação do Usuário

Verifique se a coluna *name1col* especificada corresponde à coluna *name1col* indicada na função TRANS.

O nome da coluna *name1col* não foi localizado na tabela de destino.

EML12

Name1 Col ccccc-aaa inválido

Explicação

Para executar a função TRANS indicada, a coluna *name1col* especificada deve estar em um formato válido.

Ação do Usuário

Verifique se a disponibilidade da coluna *name1col*, tipo e comprimento estão apropriados.

EML13

Name2 Col ccccc não em dest

Explicação

Para executar a função TRANS indicada, uma coluna *name2col* deve ser especificada na tabela de destino.

Ação do Usuário

Verifique se a coluna *name2col* especificada corresponde à coluna *name2col* indicada na função TRANS.

O nome da coluna *name2col* não foi localizado na tabela de destino.

EML14

Name2 Col ccccc-aaa inválido

Explicação

Para executar a função TRANS indicada, a coluna *name2col* especificada deve estar em um formato válido.

Ação do Usuário

Verifique se a disponibilidade da coluna *name2col*, tipo e comprimento estão apropriados.

Se ocorrer algum outro erro, entre em contato com o Suporte Técnico.

Exemplo 1

O exemplo a seguir usa um literal (optim.com) para formar o nome do domínio e duas colunas da tabela de destino (NAME_FIRST e NAME_LAST) para formar um nome de usuário que inclui um sublinhado:

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

Exemplo 2

O exemplo a seguir usa um nome de domínio a partir da coluna de origem e um literal (OptimUser) para formar um nome de usuário que será colocado no sufixo com um número sequencial:

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

Função TRANS COL

A função TRANS COL pode mascarar os dados que não possuem formato inerente ou um formato que não seja amplamente conhecido. TRANS COL mantém o formato e o tipo de caractere dos dados de origem no destino.

Se os dados de origem estiverem em caractere alfabéticos maiúsculos, o TRANS COL vai gerar caracteres alfabéticos maiúsculos no destino. Esta função mascara os caracteres alfabéticos e numéricos, mas outros caracteres nos dados de origem serão copiados para o destino sem serem alterados. É possível usar TRANS COL para mascarar CHAR, VARCHAR e tipos de dados numéricos não flutuantes. É possível gerar valores exclusivos, um valor diferente para cada ocorrência da mesma origem ou é possível gerar valores com um comprimento diferente da origem.

A sintaxe de TRANS COL é:

```
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 Gere um valor de destino exclusivo. O comprimento do valor de destino será o mesmo que o comprimento do valor de origem.

hash Gere um valor de destino executando hash do valor de origem. Quando o hash é usado, diferentes valores de origem podem produzir os mesmos valores de destino sempre que o processo é executado.

Nota:

Para o mesmo valor de origem, é possível obter o mesmo valor de destino quando o parâmetro exclusivo ou o parâmetro hash for usado. Use o hash com o parâmetro de valor inicial para produzir diferentes valores de destino sempre que o processo é executado.

source=colname

Use este parâmetro para especificar o nome da coluna de origem se a coluna de destino for diferente do nome da coluna de origem. O valor que você especifica será convertido em maiúscula; para evitar que o valor seja convertido para maiúscula, coloque o valor entre aspas duplas.

copy= Um ou mais pares de subsequências a serem copiadas para o destino sem serem mascarados. Se você fornecer uma sequência literal, os caracteres de origem nas posições especificadas serão substituídas. O parâmetro *copy=* será válido apenas para uma coluna de tipo de dados de caractere.

seed= O valor usado para alterar o comportamento dos algoritmos de mascaramento. Especifique uma sequência literal, referência para uma variável de ambiente ou RANDOM.

"lit" Para especificar uma sequência literal, coloque a sequência entre aspas duplas.

var (variable)

Especifique uma variável de ambiente entre parênteses. O nome de variável e seu valor não podem incluir aspas duplas.

RANDOM

Gere um valor inicial aleatório da data e hora do sistema atual.

length={n | max }

Gere um valor de destino com um comprimento diferente do comprimento de valor de origem. Use *length=max* para gerar um valor de destino que preencherá a coluna completamente. Especificar um comprimento menor que o valor de origem fará com que o valor de origem seja

truncado quando gravado para o destino. O valor que você seleciona para *n* não pode exceder o comprimento definido da coluna de destino. O parâmetro *length=* é válido apenas com *hash=*.

preserve=

Liste um ou mais valores de origem que não devam ser substituídos no destino.

null Se a coluna de origem tiver um valor nulo, não substitua o valor no destino.

spaces Se a coluna de origem tiver um valor de espaços que não substitua o valor no destino. Apenas para colunas CHAR.

zero_len

Se a coluna de origem tiver um valor VARCHAR de comprimento zero, não substitua o valor no destino.

TRIM=(char1 [charn,...])

O caractere ou caracteres de coluna de origem especificada não são marcados ou gravados para o destino. Por exemplo, se você especificar *TRIM=(x,y,z,1,2,3)*, se algum dos caracteres x, y, z, 1, 2, 3 aparecer em algum lugar na coluna de origem, eles não serão mascarados ou gravados para o destino.

[\u] Use este parâmetro para converter o(s) caractere(s) em maiúsculas antes do mascaramento. Se um caractere não tiver uma representação em maiúscula, ele permanecerá inalterado. Por exemplo, especificar *TRIM=(x,y \u)* não mascara os caracteres x e y se eles aparecerem em algum lugar na coluna de origem e alterarem algum outro caractere de coluna de origem para maiúscula antes de mascará-los.

[\r] Use este operando para remover os espaços à direita. Por exemplo, *TRIM=(x,y \u \r)* não mascara os caracteres x e y se eles aparecerem em algum lugar na coluna de origem, alterarem algum outro caractere de coluna de origem para maiúscula antes de mascará-los e removerem algum espaço à direita antes do mascaramento.

num Use este parâmetro para fazer com que a transformação de números inteiros em uma coluna de tipo de dados de caractere seja idêntico ao de uma coluna de tipo de dados numéricos. O parâmetro *num* é válido apenas em valores numéricos em uma coluna do tipo de dados de caractere. Quando usado desta maneira, a integridade de chave estrangeira é mantida em diferentes tipos de dados. Se você usar esse parâmetro, não especifique *copy=* ou *length=*.

Exemplos:

- TRANS COL UNIQUE para o valor de origem **CDE-7834**, poderia produzir um valor de destino de: **ZWQ-4598**
- TRANS COL UNIQUE SEED=RANDOM para o valor de origem **CDE-7834-2008** poderia produzir um valor de destino de: **SWX-3162-8451**
- TRANS COL UNIQUE para o valor de origem **Smith, John** poderia produzir um valor de destino de: **Fnxwq, Lrzp**
- TRANS COL UNIQUE para o valor de origem **SMITH JOHN** poderia produzir um valor de destino de: **FNXWQ LRZP**
- TRANS COL UNIQUE (COPY=((1,3)(10,4))) para o valor de origem **CDE-7834-2008** poderia produzir um valor de destino de: **CDE-4032-2008**
- TRANS COL HASH (LENGTH=13) para o valor de origem **CDE-7834** poderia produzir um valor de destino de: **ZWQ-4598RN7A0**
- TRANS COL UNIQUE PRESERVE=(spaces) para o valor de origem **XYZ 477 6835** poderia produzir um valor de destino de: **LMN 623 0972**
- TRANS COL HASH TRIM=(e \u) para o valor de origem **InfoSphere** poderia produzir um valor de destino de: **RBIWACRL**

TRANS NID

Use a função TRANS NID para mascarar os números de ID nacionais.

É possível mascarar os números de ID nacionais com um método repetido que preserve parte do valor de origem ou um método aleatório que não preserve qualquer parte do valor de origem.

É possível especificar o tipo de separadores usados nos valores de saída (traços, pontos, espaços ou nenhum separador).

A função TRANS NID usa a sintaxe a seguir:

```
[TRANS] NID ('SWI=country_code, [FMT=(output_format)], [MTD={mask|random}],  
[SRC=column_name], [VAL={Y|N}], [PRE=INV]')
```

SWITCH ou SWI

Um valor de dois caracteres que indica o tipo de ID nacional a ser mascarado. Este parâmetro é necessário. Apenas um valor do comutador é permitido.

Os valores de dois caracteres a seguir são válidos:

Canadá: Número de Seguro Social (SIN) do Canadá

CA

França: Número do Instituto Nacional de Estudos Estatísticos e Econômicos da França (INSEE)

FR

Itália: Número do Código Fiscal (CF) Italiano

IT

Espanha: Número de Identificação Fiscal da Espanha (NIF)

ES

Reino Unido: Número do Seguro Nacional do Reino Unido (NINO)

UK

Estados Unidos da América: Número de Seguridade Social dos Estados Unidos (SSN)

US

FMT ou FORMAT

Determina o formato de saída e quais partes do valor de origem mascarar. A sintaxe para este parâmetro é determinada pelo ID nacional sendo mascarado:

- Número de Seguro Social (SIN) do Canadá
- Número do Instituto Nacional de Estudos Estatísticos e Econômicos da França (INSEE)
- Número do Código Fiscal (CF) Italiano
- Número de Identificação Fiscal da Espanha (NIF)/Número de Identificação do Estrangeiro (NIE)
- Número de Seguro Nacional do Reino Unido (NINO)
- Número de Seguridade Social dos Estados Unidos (SSN)

Se este parâmetro for omitido e *MTD=mask*, o formato de origem será usado e os campos padrão serão mascarados.

Se este parâmetro for omitido e *MTD=random*, os valores de saída não incluirão os separadores.

Se a coluna de destino não for grande o suficiente para conter um formato de saída que use os separadores, os separadores não serão incluídos.

MTD ou METHOD

Determina qual método de mascaramento usar, repetido ou aleatório. Se este parâmetro for omitido, o mascaramento repetido será executado (*MTD=mask*).

mask Os valores de origem são mascarados de uma maneira repetida. Os valores de saída são baseados nos valores de origem. Padrão.

random

Os valores de saída são gerados por um algoritmo de mascaramento aleatório. Os valores de saída não são baseados nos valores de origem.

Nota:

- *MTD=random* não é compatível com os parâmetros a seguir: *VAL=Y* e *PRE=INV*.
- Se o parâmetro *MTD=random* e *FMT* especificarem que parte do valor de origem deve ser copiado para o valor de saída, o valor de saída não incluirá os valores de origem; no entanto, qualquer separador especificado no parâmetro *FMT* será incluído no valor de saída.

SRC ou SOURCE

O nome da coluna que contém os valores de origem. Use este parâmetro apenas se os nomes das colunas de origem e de destino não forem correspondentes.

VAL ou VALIDATE

Determina se a validação específica do país é executada nos valores de origem. Se este parâmetro for omitido, a validação não será executada (*VAL=N*).

Y Valide os valores de origem.

Nota: A opção não pode ser usada se *MTD=random*.

N Não valide os valores de origem. Padrão.

PRE ou PRESERVE

Determina se os valores de origem inválidos são copiados para a coluna de destino. Se este parâmetro for omitido, os valores de origem inválidos não serão copiados para a coluna de destino e as linhas que contêm esses valores serão ignoradas. Este parâmetro inclui apenas uma opção, *INV* ou *INVALID*.

INV ou INVALID

Copie os valores de origem inválidos para a coluna de destino.

Nota: A opção não pode ser usada se *MTD=random*.

Exemplos

A sintaxe a seguir vai mascarar os Números de Seguridade Social dos Estados Unidos (SSN) com um método repetido. A função copiará os três primeiros dígitos do SSN de origem e incluirá um separador de traço para o valor de saída. A validação será executada no valor de saída.

```
NID('SWITCH=US, FMT=(US=3C-2X-4X), MTD=MASK, VAL=Y')
```

A sintaxe a seguir vai mascarar os números do Instituto Nacional de Estudos Estatísticos e Econômicos da França (INSEE) com um método aleatório. A função usará o formato de saída padrão, que não incluirá os separadores.

```
NID('SWITCH=FR, MTD=RANDOM')
```

A sintaxe a seguir vai mascarar os Números de Identificação Fiscal da Espanha (NIF) com um método repetido. A função vai preservar os valores de origem inválidos e usará o formato de saída padrão.

```
NID('SWITCH=ES, MTD=MASK, PRE=INV')
```

Mascaramento do Número de Seguro Social do Canadá

É possível usar a função TRANS NID para mascarar os Social Insurance Numbers (SIN) do Canadá.

Um SIN é um número de nove dígitos que consiste em um número do código de região de um dígito seguido por um número de série de oito dígitos. Os três primeiros dígitos são chamados de cabeçalho. O último dígito do número de série é um dígito de verificação.

A função TRANS NID gera um SIN mascarado com um dígito de verificação que é calculado com base nos oito dígitos mascarados precedentes do valor de saída.

Formatos de Saída (FMT=)

Os formatos de saída a seguir estão disponíveis para um SIN.

C indica valores que são copiados. X indica valores que são mascarados. Por exemplo, 3C4X indica que os três primeiros caracteres são copiados e os próximos quatro caracteres são mascarados.

Campos a serem mascarados	Formato sem separador	Formato com separador de traço	Formato com separador de espaço	Formato com separador de ponto
Número de série sem dígitos de cabeçalho (padrão <i>MTD=mask</i>)	CA=3C6X	CA=3C-3X-3X	CA=3C 3X 3X	CA=3C.3X.3X
Número de série e dígitos de cabeçalho	CA=9X	CA=3X-3X-3X	CA=3X 3X 3X	CA=3X.3X.3X

Tipos de Dados

Os tipos de dados a seguir são permitidos para as colunas de origem e de destino. As restrições para cada tipo de dados são observadas.

CHAR e NCHAR

O comprimento da coluna deve ter pelo menos 9 caracteres.

DECIMAL

A precisão da coluna deve estar entre 9 e 20 e a escala deve ser 0.

INTEGER

Sem restrições.

VARCHAR e NVARCHAR

O comprimento da coluna deve ter pelo menos 9 caracteres.

Validação (VAL=Y)

Se o parâmetro (VAL=Y) for usado, uma linha de origem será ignorada se algum dos seguintes for aplicado:

- O primeiro dígito é oito.
- Existe três zeros consecutivos nas posições 1-3, 4-6 ou 7-9.
- O dígito de verificação é inválido.

Processamento Especial

As verificações a seguir serão feitas durante o processamento:

- Se o valor de origem for 0 (tipo de dados INTEGER ou DECIMAL), os espaços ou um VARCHAR ou NVARCHAR de comprimento zero e
 - a coluna de destino é um tipo de dados INTEGER ou DECIMAL, o valor de saída será 0.
 - a coluna de destino é um tipo de dados CHAR ou NCHAR, o valor da saída será os espaços.

- a coluna de destino é um tipo de dados VARCHAR ou NVARCHAR, o comprimento do destino será 0.
- Se um comprimento de coluna de destino tiver 11 caracteres ou mais e os separadores forem especificados para o valor de saída, os separadores serão incluídos se a coluna de destino for do tipo de dados CHAR, NCHAR, VARCHAR ou NVARCHAR.
- Se o valor de origem for NULL, o valor de saída será NULL.

Linhas Ignoradas

Uma linha de origem será ignorado e não gravada na tabela de destino, se ocorrer o seguinte:

Valor Nulo

O valor de origem é NULL, mas a coluna de destino não permite um valor NULL.

SIN Inválido

- A coluna de origem é CHAR, NCHAR, VARCHAR ou NVARCHAR, mas o valor de origem é inferior a 9 caracteres (sem incluir os separadores).
- A coluna de origem é CHAR, NCHAR, VARCHAR ou NVARCHAR, mas o valor de origem é superior a 9 caracteres (sem incluir os separadores).
- O valor de origem inclui um caractere não numérico.

Mascaramento de Número do Instituto Nacional de Estudos Estatísticos e Econômicos da França

É possível usar a função TRANS NID para mascarar os números do Instituto Nacional de Estudos Estatísticos e Econômicos da França (INSEE).

Um número INSEE é um número de 15 dígitos com o formato a seguir: *SYMMDDCCCOOKK*.

- S** Informações sobre sexo e cidadania.
- YY** Dois últimos dígitos do ano de nascimento.
- MM** Mês do nascimento.
- DD** Departamento de origem.
- CCC** Comunidade de origem.
- OOO** Número da ordem.
- KK** Chave de controle ou dígitos de verificação.

A função TRANS NID gera um INSEE mascarado para as regras a seguir:

- Se o campo de departamento for mascarado, o campo de comunidade também será mascarado com um valor compatível.
- O campo da origem é sempre mascarada.
- O campo de dígito de verificação é calculado com base nos 13 dígitos mascarados precedentes do valor de saída.

Formatos de Saída (FMT=)

Os formatos a seguir estão disponíveis para um INSEE.

Todos os formatos mascaram os campos de dígito de verificação e de ordem. Se o campo de departamento for mascarado, o campo de comunidade também será mascarado com um valor compatível.

C indica valores que são copiados. X indica valores que são mascarados. Por exemplo, 3C4X indica que os três primeiros caracteres são copiados e os próximos quatro caracteres são mascarados.

Campos a serem mascarados (além da Ordem e Dígito de Verificação)	Formato sem separador	Formato com separador de traço	Formato com separador de espaço
Sexo, Ano, Mês, Comunidade (padrão <i>MTD=mask</i>)	FR=5X2C8X	FR=5X2C6X-2X	FR=5X2C6X 2X
Sexo	FR=1X9C5X	FR=1X9C3X-2X	FR=1X9C3X 2X
Sexo, Ano	FR=3X7C5X	FR=3X7C3X-2X	FR=3X7C3X 2X
Sexo, Mês	FR=1X2C2X5C5X	FR=1X2C2X5C3X-2X	FR=1X2C2X5C3X 2X
Sexo, Comunidade	FR=1X6C8X	FR=1X6C6X-2X	FR=1X6C6X 2X
Sexo, Departamento	FR=1X4C8X	FR=1X4C6X-2X	FR=1X4C6X 2X
Sexo, Ano, Mês	FR=5X5C5X	FR=5X5C3X-2X	FR=5X5C3X 2X
Sexo, Ano, Comunidade	FR=3X4C8X	FR=3X4C6X-2X	FR=3X4C6X 2X
Sexo, Ano, Departamento, Comunidade	FR=3X2C10X	FR=3X2C8X-2X	FR=3X2C8X 2X
Sexo, Mês, Comunidade	FR=1X2C2X2C8X	FR=1X2C2X2C6X-2X	FR=1X2C2X2C6X 2X
Sexo, Mês, Departamento, Comunidade	FR=1X2C12X	FR=1X2C10X-2X	FR=1X2C10X 2X
Sexo, Ano, Mês, Departamento, Comunidade	FR=15X	FR=13X-2X 13X	FR=13X 2X
Ano	FR=1C2X7C5X	FR=1C2X7C3X-2X	FR=1C2X7C3X 2X
Ano, Mês	FR=1C4X5C5X	FR=1C4X5C3X-2X	FR=1C4X5C3X 2X
Ano, Comunidade	FR=1C2X4C8X	FR=1C2X4C6X-2X	FR=1C2X4C6X 2X
Ano, Departamento	FR=1C2X2C10X	FR=1C2X2C8X-2X	FR=1C2X2C8X 2X
Ano, Mês, Comunidade	FR=1C4X2C8X	FR=1C4X2C6X-2X	FR=1C4X2C6X 2X
Ano, Mês, Departamento	FR=1C14X	FR=1C12X-2X	FR=1C12X 2X
Mês	FR=3C2X5C5X	FR=3C2X5C3X-2X	FR=3C2X5C3X 2X
Mês, Comunidade,	FR=3C2X2C8X	FR=3C2X2C6X-2X	FR=3C2X2C6X 2X
Mês, Departamento	FR=3C12X	FR=3C10X-2X	FR=3C10X 2X
Comunidade	FR=7C8X	FR=7C6X-2X	FR=7C6X 2X
Departamento	FR=5C10X	FR=5C8X-2X	FR=5C8X 2X

Tipos de Dados

Os tipos de dados a seguir são permitidos para as colunas de origem e de destino. As restrições para cada tipo de dados são observadas.

CHAR e NCHAR

O comprimento da coluna deve ter pelo menos 15 caracteres.

VARCHAR e NVARCHAR

O comprimento da coluna deve ter pelo menos 15 caracteres.

Validação (VAL=Y)

Se o parâmetro (VAL=Y) for usado, uma linha de origem será ignorada se algum dos seguintes for aplicado:

- O valor do campo da comunidade de origem é inválido.
- O campo do dígito de verificação de origem é inválido.

Processamento Especial

As verificações a seguir serão feitas durante o processamento:

- Se o valor de origem for de espaços ou um VARCHAR ou NVARCHAR de comprimento zero, e
 - a coluna de destino é um tipo de dados CHAR ou NCHAR, o valor da saída será os espaços.
 - a coluna de destino é um tipo de dados VARCHAR ou NVARCHAR, o comprimento do destino será 0.
- Se o valor de origem for NULL, o valor de saída será NULL.
- Se um comprimento da coluna de destino for de 16 caracteres ou mais e um separador foi especificado para o valor de saída, o separador será incluído.

Linhas Ignoradas

Uma linha de origem será ignorada e não gravada na tabela de destino, se o valor de origem incluir os erros a seguir:

Valor Nulo

O valor de origem é NULL, mas a coluna de destino não permite um valor NULL.

INSEE Inválido

- O tamanho do valor INSEE de entrada é maior ou menor do que o válido.
- O valor INSEE de entrada contém separados inválidos ou separadores na posição errada.
- O campo de sexo não é um dos valores a seguir: 1, 2, 7 ou 8.

Mascaramento de Número de Código Fiscal da Itália

É possível usar a função TRANS NID para mascarar os Códigos Fiscais (CF) Italianos.

Um CF é um valor alfanumérico de 16 caracteres com o formato a seguir: *FFF-NNN-YYMDD-RRRRC*.

- FFF** Sobrenome codificado.
- NNN** Nome dado codificado.
- YY** Ano do nascimento.
- M** Mês do nascimento.
- DD** Data do nascimento.
- RRRR** Código de região.
- C** Caractere de controle.

A função TRANS NID gera um CF mascarado para as regras a seguir:

- Qualquer consoante que apareça nos campos de nome dado ou sobrenome é mascarada como uma consoante e qualquer vogal é mascarada como uma vogal. Se um X aparecer após uma vogal, ele deve ser copiado para o valor de saída.
- O campo do caractere de controle é calculado com base nos 15 dígitos mascarados precedentes do valor de saída.

Formatos de Saída (FMT=)

Os formatos de saída a seguir estão disponíveis para um CF.

C indica valores que são copiados. X indica valores que são mascarados. Por exemplo, 3C4X indica que os três primeiros caracteres são copiados e os próximos quatro caracteres são mascarados.

Campos a serem mascarados	Formato sem separador	Formato com separador de traço	Formato com separador de espaço
Data de nascimento, Região (padrão <i>MTD=mask</i>)	IT=6C10X	IT=3C-3C-5X-5X	IT=3C 3C 5X 5X
Sobrenome, Nome Dado, Região	IT=6X5C5X	IT=3X-3X-5C-5X	IT=3X 3X 5C 5X
Sobrenome, Nome Dado, Data de nascimento	IT=11X4C1X	IT=3X-3X-5X-4C1X	IT=3X 3X 5X 4C1X
Sobrenome, Nome dado	IT=6X9C1X	IT=3X-3X-5C-4C1X	IT=3X 3X 5C 4C1X
Data de nascimento	IT=6C5X4C1X	IT=3C-3C-5X-4C1X	IT=3C 3C 5X 4C1X
Região	IT=11C5X	IT=3C-3C-5C-5X	IT=3C 3C 5C 5X
Sobrenome, Nome Dado, Data de nascimento, Região	IT=16X	IT=3X-3X-5X-5X	IT=3X 3X 5X 5X

Tipos de Dados

Os tipos de dados a seguir são permitidos para as colunas de origem e de destino. As restrições para cada tipo de dados são observadas.

CHAR e NCHAR

O comprimento da coluna deve ter pelo menos 16 caracteres.

VARCHAR e NVARCHAR

O comprimento da coluna deve ter pelo menos 16 caracteres.

Validação (*VAL=Y*)

Se o parâmetro *VAL=Y* for usado, uma linha de origem será ignorada se contiver um caractere de controle inválido.

Processamento Especial

As verificações a seguir serão feitas durante o processamento:

- Se o valor de origem for de espaços ou um VARCHAR ou NVARCHAR de comprimento zero, e
 - a coluna de destino é um tipo de dados CHAR ou NCHAR, o valor da saída será os espaços.
 - a coluna de destino é um tipo de dados VARCHAR ou NVARCHAR, o comprimento do destino será 0.
- Se o valor de origem for NULL, o valor de saída será NULL.
- Se um comprimento da coluna de destino for de 19 caracteres ou mais e os separadores foram especificados para um valor de saída, os separadores serão incluídos.

Linhas Ignoradas

Uma linha de origem será ignorado e não gravada na tabela de destino, se ocorrer o seguinte:

Valor Nulo

O valor de origem é NULL, mas a coluna de destino não permite um valor NULL.

CF Inválido

A coluna de origem possui um valor de origem inválido ou o valor de origem é muito grande.

O valor de origem tem menos de 16 caracteres.

Mascaramento do Número de Identificação Fiscal da Espanha e Número de Identificação de Estrangeiro

É possível usar a função TRANS NID para mascarar os Números de Identificação Fiscal Espanhola (NIF) e Números de Identificação de Estrangeiro (NIE).

Um NIF é um valor de oito caracteres no formato a seguir *NNNNNNN-A*, em que os sete primeiros caracteres são um número de série e o caractere final é um sufixo alfabético. O sufixo é um dígito de verificação.

Os estrangeiros de nacionalidade espanhola usam um Número de Identificação de Estrangeiro (NIE), que é um valor de nove caracteres que usa o mesmo formato que um NIF, mas é precedido por um X. Um NIE usa o formato a seguir: *X-NNNNNNN-A*.

A função TRANS NID gera um NIF ou NIE mascarado com um dígito de verificação que é calculado com base nos 7 dígitos mascarados precedentes do valor de saída.

Formatos de Saída (*FMT=*)

Os formatos de saída a seguir estão disponíveis para um NIF e NIE.

Para cada formato, todos os caracteres serão mascarados. Os números NIF e NIE usam as mesmas opções de formato. Um valor de origem NIE sempre incluirá um prefixo X no valor de saída.

Campos a serem mascarados	Formato sem separador	Formato com separador de traço	Formato com separador de espaço
Serial, Sufixo (padrão <i>MTD=mask</i>)	ES=8X	ES=7X-1X	ES=7X 1X

Tipos de Dados

Os tipos de dados a seguir são permitidos para as colunas de origem e de destino. As restrições para cada tipo de dados são observadas.

CHAR e NCHAR

O comprimento da coluna deve ter pelo menos 8 caracteres.

VARCHAR e NVARCHAR

O comprimento da coluna deve ter pelo menos 8 caracteres.

Validação (*VAL=Y*)

Se o parâmetro *VAL=Y* for usado, uma linha de origem será ignorada se contiver um sufixo inválido.

Processamento Especial

As verificações a seguir serão feitas durante o processamento:

- Se o valor de origem for de espaços ou um VARCHAR ou NVARCHAR de comprimento zero, e
 - a coluna de destino é um tipo de dados CHAR ou NCHAR, o valor da saída será os espaços.
 - a coluna de destino é um tipo de dados VARCHAR ou NVARCHAR, o comprimento do destino será 0.
- Se o valor de origem for NULL, o valor de saída será NULL.
- Se um comprimento da coluna de destino for de 11 caracteres ou mais e um separador foi especificado para o valor de saída, o separador será incluído.

Linhas Ignoradas

Uma linha de origem será ignorado e não gravada na tabela de destino, se ocorrer o seguinte:

Comprimento Inválido

O valor de entrada não contém separadores e tem menos de 8 caracteres (NIF) ou tem menos de 9 caracteres (NIE).

O valor de entrada contém separados e tem menos de 9 caracteres (NIF) ou tem menos de 11 caracteres (NIE).

Padrão Inválido

O comprimento do valor de origem é válido, mas o valor não corresponde ao padrão de um NIF ou NIE.

Separadores

Diferentes separadores foram localizados nas posições 2 e 10 em um valor de origem NIE.

O valor de origem contém um separador inválido.

Valor Nulo

O valor de origem é NULL, mas a coluna de destino não permite um valor NULL.

Mascaramento do Número de Seguro Nacional do Reino Unido

É possível usar a função TRANS NID para mascarar os Números de Seguro Nacional (NINO) do Reino Unido.

Um NINO consiste em três partes: duas letras (o prefixo), seis dígitos (o número) e uma letra opcional (o sufixo).

Formatos de Saída (*FMT=*)

Os formatos de saída a seguir estão disponíveis para um NINO.

Um NINO pode ser mascarado em um separador ou com um separador em um formato de três ou cinco partes.

C indica valores que são copiados. X indica valores que são mascarados. Por exemplo, 3C4X indica que os três primeiros caracteres são copiados e os próximos quatro caracteres são mascarados.

Para criar um NINO sem um separador, use os parâmetros a seguir:

Campos a serem mascarados	Formato sem separador
Prefixo, Número	UK=8X1C
Número (padrão <i>MTD=mask</i>)	UK=2C6X1C
Prefixo, Número, Sufixo (apenas <i>MTD=random</i>)	UK=9X

Para criar um NINO com um formato de três ou cinco partes, use os parâmetros a seguir:

Campos a serem mascarados	Formato com separador de traço	Formato com separador de espaço	Formato com separador de ponto
Prefixo, Número (três partes)	UK=2X-6X-1C	UK=2X 6X 1C	UK=2X.6X.1C
Prefixo, Número (cinco partes)	UK=2X-2X-2X-2X-1C	UK=2X 2X 2X 2X 1C	UK=2X.2X.2X.2X.1C
Número (três partes)	UK=2C-6X-1C	UK=2C 6X 1C	UK=2C.6X.1C

Campos a serem mascarados	Formato com separador de traço	Formato com separador de espaço	Formato com separador de ponto
Número (cinco partes)	UK=2C-2X-2X-2X-1C	UK=2C 2X 2X 2X 1C	UK=2C.2X.2X.2X.1C
Prefixo, Número, Sufixo (três partes) (apenas <i>MTD=random</i>)	UK=2X-6X-1X	UK=2X 6X 1X	UK=2X.6X.1X
Prefixo, Número, Sufixo (cinco partes) (apenas <i>MTD=random</i>)	UK=2X-2X-2X-2X-1X	UK=2X 2X 2X 2X 1X	UK=2X.2X.2X.2X.1X

Tipos de Dados

Os tipos de dados a seguir são permitidos para as colunas de origem e de destino. As restrições para cada tipo de dados são observadas.

CHAR e NCHAR

O comprimento da coluna deve ter pelo menos 9 caracteres.

VARCHAR e NVARCHAR

O comprimento da coluna deve ter pelo menos 9 caracteres.

Validação (*VAL=Y*)

O parâmetro *VAL=Y* não é válido para o mascaramento NINO e será ignorado.

Processamento Especial

As verificações a seguir serão feitas durante o processamento:

- Se o valor de origem for de espaços ou um VARCHAR ou NVARCHAR de comprimento zero, e
 - a coluna de destino é um tipo de dados CHAR ou NCHAR, o valor da saída será os espaços.
 - a coluna de destino é um tipo de dados VARCHAR ou NVARCHAR, o comprimento do destino será 0.
- Se o valor de origem for NULL, o valor de saída será NULL.
- Se a coluna de destino não puder acomodar um separador, o valor de saída não incluirá um separador.

Linhas Ignoradas

Uma linha de origem será ignorado e não gravada na tabela de destino, se ocorrer o seguinte:

Valor Nulo

O valor de origem é NULL, mas a coluna de destino não permite um valor NULL.

NINO Inválido

- O tamanho do valor da origem for menor ou maior que o tamanho de um NINO válido.
- O valor de origem inclui separados nas posições erradas.
- O valor de origem inclui um separador inválido.
- O valor de origem inclui um prefixo inválido.
- O valor de origem inclui um sufixo diferente de A, B, C ou D.
- O valor de origem inclui um campo numérico que não está entre 000001 e 999999.

Mascaramento do Número de Seguro Social dos Estados Unidos

É possível usar a função TRANS NID para mascarar os Números de Seguridade Social (SSN) dos Estados Unidos.

Um SSN consiste em 3 subcampos com o formato a seguir: AAAGGSSSS.

AAA Número de área. A área geralmente é determinada pelo estado no qual o SSN é emitido.

GG Número de grupo. Um número de grupo é designado com base no número da área.

SSSS Número de série.

A função TRANS NID gera um SSN mascarado para as regras a seguir:

- Um número de grupo apropriado para o número de área é gerado. O número de grupo será o grupo mais recente usado pela Administração de Seguridade Social para a área.
- Os números de série iniciam com 0001 e são incrementados por 1 para cada SSN adicional gerado para o número da área. Quando o número de série exceder 9999, o número de série será reconfigurado para 0001 e o número do grupo que precede o número emitido mais recentemente para o número da área será usado.
- Quando $MTD=mask$, o valor de saída incluirá um número de área correspondente ao mesmo estado que o número de área de origem.

Formatos de Saída (FMT=)

Os formatos de saída a seguir estão disponíveis para um SSN.

C indica valores que são copiados. X indica valores que são mascarados. Por exemplo, 3C4X indica que os três primeiros caracteres são copiados e os próximos quatro caracteres são mascarados.

Campos a serem mascarados	Formato sem separador	Formato com separador de traço	Formato com separador de espaço	Formato com separador de ponto
Grupo, Número de série (padrão <i>MTD=mask</i>)	US=3C6X	US=3C-2X-4X	US=3C 2X 4X	US=3C.2X.4X
Área, Grupo, Número de Série	US=9X	US=3X-2X-4X	US=3X 2X 4X	US=3X.2X.4X

Tipos de Dados

Os tipos de dados a seguir são permitidos para as colunas de origem e de destino. As restrições para cada tipo de dados são observadas.

CHAR e NCHAR

O comprimento da coluna deve ter pelo menos 9 caracteres.

DECIMAL

A precisão da coluna deve estar entre 9 e 20 e a escala deve ser 0.

INTEGER

Sem restrições.

VARCHAR e NVARCHAR

O comprimento da coluna deve ter pelo menos 9 caracteres.

Validação (VAL=Y)

Se o parâmetro (VAL=Y) for usado, uma linha de origem será ignorada se algum dos seguintes for aplicado:

- O número da área de origem excede o valor máximo.
- O número da área de origem não foi usado pela Administração de Seguridade Social.
- O número do grupo de origem não foi usado para o número da área de origem.

Processamento Especial

As verificações a seguir serão feitas durante o processamento:

- Se o valor de origem for 0 (tipo de dados INTEGER ou DECIMAL), os espaços ou um VARCHAR ou NVARCHAR de comprimento zero e
 - a coluna de destino é um tipo de dados INTEGER ou DECIMAL, o valor de saída será 0.
 - a coluna de destino é um tipo de dados CHAR ou NCHAR, o valor da saída será os espaços.
 - a coluna de destino é um tipo de dados VARCHAR ou NVARCHAR, o comprimento do destino será 0.
- Se um comprimento da coluna de destino for de 11 caracteres ou mais e os separadores foram especificados para um valor de saída, os separadores serão incluídos.
- Se o valor de origem for NULL, o valor de saída será NULL.

Linhas Ignoradas

Uma linha de origem será ignorado e não gravada na tabela de destino, se ocorrer o seguinte:

Valor Nulo

O valor de origem é NULL, mas a coluna de destino não permite um valor NULL.

SSN Inválido

- A coluna de origem é CHAR, NCHAR, VARCHAR ou NVARCHAR, mas o valor de origem tem mais ou menos de 9 caracteres (sem incluir os separadores).
- O valor de origem inclui um caractere não numérico.
- O valor de origem é 0 ou um valor reservado como 078-05-1120 e 457-55-5462.

Função de Subsequência

A Função de Subsequência retorna uma subsequência do conteúdo da coluna nomeada.

`SUBSTR(columnname, start, [length])`

columnname

Nome de um caractere ou coluna binária.

start

A posição do primeiro caractere na sequência.

length

O número de caracteres a serem usados.

- Se o código de idioma usa uma vírgula como separador decimal, você deve deixar um espaço após cada vírgula que separa os parâmetros numéricos (por exemplo, após a vírgula entre *start* e *length*).
- *start* e *length* são números inteiros maiores ou iguais a 1.
- *start* mais *length* não pode exceder o comprimento de dados total mais 1.
- Os valores *column-name* e *start* são necessários. Se você especificar apenas um número inteiro, ele será usado como o valor *start*. A subsequência inicia em *start* e inclui o restante do valor da coluna.

Exemplo

Se a coluna PHONE_NUMBER for definida como CHAR(10), será possível usar a Função de Subsequência para mapear o código da área. Para obter uma subsequência das três primeiras posições do número do telefone (código de área) para a coluna de destino, especifique:

```
SUBSTR(PHONE_NUMBER, 1, 3)
```

Função Aleatória

A Função Aleatória retorna um número selecionado aleatoriamente no intervalo indicado pelos valores alto e baixo.

`RAND(low, high)`

low Valor aleatório mais baixo possível.

high Valor aleatório mais alto possível.

- Use a Função Aleatória com caractere ou dados numéricos.
- Se o código de idioma usa uma vírgula como separador decimal, você deve deixar um espaço após a vírgula.
- *low* e *high* são números inteiros no intervalo **-2,147,483,648** a **2,147,483,647**.
- *low* e *high* são mais limitados pelo tipo de dados e comprimento para a coluna de destino.
- *low* deve ser menor que *high*.
- Ao usar a Função Aleatória em uma expressão concatenada, uma sequência de comprimento variável será retornada.

Exemplo

É possível usar a Função Aleatória para mascarar ou alterar os dados da venda para um banco de dados de teste. Assuma que a coluna `YTD_SALES` é definida como `DECIMAL(7,2)`. O número máximo de dígitos que precedem o decimal é 5; o intervalo possível para esta coluna é -99999 a 99999. Para criar os dados de teste em um intervalo de 1000 (baixo) para 89999 (alto), especifique:

```
RAND(1000, 89999)
```

Neste exemplo, a função retorna os valores de vendas aleatórios no intervalo que você especificou de 1000.00 a 89999.99.

Função Sequencial

A Função Sequencial retorna um número que é incrementado sequencialmente. A sintaxe é:

`SEQ(start, step)`

start Valor inicial.

step Valor incremental.

- Use a Função Sequencial com caractere e dados numéricos.
- Se o código de idioma usa uma vírgula como separador decimal, você deve deixar um espaço após a vírgula.
- *start* e *step* são números inteiros no intervalo de **-2,147,483,648** e **2,147,483,647**.
- *start* e *step* são mais limitados pelo tipo de dados e comprimento da coluna de destino.
- Se o valor calculado exceder o comprimento da coluna de destino, a função reconfigurará automaticamente para o valor *start*.
- Ao usar a Função Sequencial em uma expressão concatenada, uma sequência de comprimento variável será retornada.

Exemplo 1

É possível usar a Função Sequencial para alterar os dados do cliente para um banco de dados de teste. Imagine que a coluna `CUST_ID` seja definida como `CHAR(5)`. Para incrementar por 50, iniciando em 1, especifique:

SEQ(1, 50)

Neste exemplo, a função retorna valores CUST_ID que iniciam em '00001' e incrementa por 50 para gerar '00051', '00101', etc. Quando o resultado exceder '99951', a função será reconfigurada para o valor *start* de 1.

Exemplo 2

É possível usar a Função Sequencial em um mapa de coluna para mascarar os dados de vendas para um banco de dados de teste. Imagine que a coluna YTD_SALES esteja definida como DECIMAL(7,2). Para incrementar por 100 iniciando em 1000, especifique:

SEQ(1000, 100)

Neste exemplo, a função retorna os valores YTD_SALES iniciando em 1000 e incrementa por 100 para gerar 1100, 1200, etc. Quando o resultado excede 99999, a função é reconfigurada para o valor *start* de 1000.

Exemplo 3

Assume que a coluna SALESMAN_ID esteja definida como CHAR(6). Para inserir os valores que iniciam com 'NJ', seguidos por um número que inicia em 50 e incrementado por 10, use a função em uma expressão concatenada.

'NJ' || SEQ(50, 10)

Neste exemplo, a função retorna valores SALESMAN_ID que iniciam em 'NJ50 ' e incrementa por 10 para gerar 'NJ60 ', 'NJ70 ', etc. Quando o resultado excede 'NJ9990', a função é reconfigurada para o valor *start*.

Função de Identidade ou Serial

As Funções de Identidade ou Serial instruem o DBMS a fornecer um valor sequencial (número inteiro) para uma coluna de destino. A sintaxe para essas funções é:

IDENTITY()

SERIAL ()

- Use a Função de Identidade para colunas de Identidade nos bancos de dados DB2, Sybase ASE e SQL Server.
- Use a Função Serial para colunas Seriais em bancos de dados Informix.
- Essas duas funções são válidas para Inserção (atualização/inserção) e Processamento de Carregamento, mas não são válidas para Processamento de Conversão.
- Se as linhas forem atualizadas em um Processo de Inserção (atualização/inserção), a coluna de destino direcionada pela Função de Identidade ou Função Serial reterá o valor original. Além disso, se a coluna de destino fizer parte da chave primária, o valor da coluna permanecerá inalterado quando a linha for atualizada.
- É possível usar a Função de Identidade ou a Função Serial com a Função Propagar para Processamento de Inserção; no entanto, não é possível propagar as colunas Identidade ou Serial em um Processo de Carregamento.

Função de Sequência Oracle

A Função de Sequência Oracle designa um valor para a coluna de destino usando uma Sequência Oracle.

schema.seqname.NEXTVAL [INCL_UPD]

schema Qualificador para o nome da Sequência Oracle.

seqname

Nome da Sequência Oracle que designa os valores sequencial.

NEXTVAL

A palavra-chave que insere o próximo valor Oracle na coluna de destino.

INCL_UPD

A palavra-chave opcional que atualiza um valor de sequência designado a uma coluna quando as linhas forem atualizadas durante um processo de inserção. Se não especificada (padrão), o valor da coluna permanecerá inalterado quando a linha for atualizada.

- É possível usar a Função de Sequência Oracle para designar os valores sequenciais exclusivos para que as linhas sejam inseridas em um banco de dados Oracle.
- A Função de Sequência Oracle é válida quando usada em um mapa da coluna para o processamento de inserção ou carregamento, mas não é válida para o processamento de conversão.
- Se as linhas estiverem atualizadas em um processo de inserção (atualização/inserção) e a coluna de destino fizer parte da chave primária, o valor de coluna permanecerá inalterado quando a linha na tabela de destino for atualizada. Para usar a Sequência Oracle ao executar uma atualização/inserção, inclua *INCL_UPD* com a função.
- Durante um processo de carregamento, o processo usa a Função de Sequência Oracle para designar um novo valor para cada linha de destino. O Carregamento chama o DBMS para obter esses valores. Se você optar por não executar o Carregador Oracle, esses valores de sequência nunca serão usados.

Exemplo 1

Para um valor sequencial para incrementar os números de cliente, em que o nome da Sequência Oracle é *schema.numeven*, especifique:

```
schema.numeven.NEXTVAL
```

Exemplo 2

Para expandir o primeiro exemplo e atualizar um valor de sequência existente, especifique:

```
schema.numeven.NEXTVAL(INCL_UPD)
```

Função Propagar Valor de Chave Estrangeira ou Primária

A Função de Propagação designa um valor para uma chave primária ou coluna de chave estrangeira e propaga esse valor para todas as tabelas relacionadas.

A sintaxe é:

```
PROP( { value [, columnname] ] EXIT exitname |  
PROC { LOCAL | identifier.name } } )
```

value Valor para designar à coluna. Especifique qualquer valor de origem de mapa de coluna válida (por exemplo, um nome de coluna, sequência literal, expressão ou função). O valor deve ser apropriado para a coluna.

columnname

O nome da coluna de origem que contém o valor que é o assunto da função. O valor resultante é inserido na coluna de destino da tabela mapeada e coluna de destino apropriada nas tabelas relacionadas participantes.

O nome da coluna é necessário *apenas* se nenhuma coluna de origem corresponde à coluna de destino no tipo de dados e nome. Se não for especificado, será usado o nome da coluna de destino.

exitname

Um nome de saída do mapa de coluna.

identifier.name

Um nome de procedimento de mapa da coluna.

- Se o código de idioma usa uma vírgula como separador decimal, você deve deixar um espaço após cada vírgula que separa os parâmetros numéricos.
- A Função de Propagação é válida em um mapa de coluna para processamento de inserção (mas não atualização ou atualização/inserção), carregamento ou conversão.
- Ao usar a Função de Propagação, pelo menos uma tabela relacionada deve ser incluída no processo. É possível usar Propagar várias vezes para o mesmo processo.
- Você pode usar a Função de Propagação para uma coluna de chave primária ou sua coluna de chave estrangeira correspondente, mas não ambas.
- Se as diversas colunas definirem uma relacionamento, será possível usar a Função de Propagação para uma ou mais dessas colunas. No entanto, em um relacionamento estendido Optim, você pode especificar a Função de Propagação apenas nas relações coluna a coluna.
- É possível usar a Função de Identidade ou Função Serial na Função de Propagação para inserir o processamento; no entanto, não é possível propagar a Função de Identidade em um processo de carregamento.
- Os parâmetros especificados na Função de Propagação não são validados até o tempo de execução. Se houver conflitos, o processo não será executado.
- A inserção pode ter ciclos de propagação. No entanto, o processamento de carregamento e conversão pode não resultar em ciclos de propagação. Os ciclos são detectados quando o processo é validado no tempo de execução. Se uma solicitação de carregamento ou conversão gerar um ciclo de propagação, o processo não será executado.
- O Optim lembra os valores de origem e os valores designados às colunas de destino correspondentes. Portanto, você pode propagar para as colunas de destino em que a origem é uma expressão. Quando a expressão avaliada corresponde a um valor de origem, Optim designa o valor de destino correspondente. Quando a expressão avaliada não corresponder a algum valor de origem, ocorrerá um erro de conversão.

Antes de executar um processo de inserção, carregamento ou conversão, é possível revisar o mapa de coluna para verificar como a Função de Propagação é usada no processo.

Exemplo 1

É possível gerar um número aleatório, designá-lo à coluna de destino padrão e propagar o número nas colunas de destino das tabelas relacionadas. Para gerar um valor entre 10000 e 99999, insira-o na coluna de destino mapeada e propague-o para as colunas de destino das tabelas relacionadas, especifique:

```
PROP(RAND(10000, 99999))
```

Exemplo 2

É possível executar a mesma função como no Exemplo 1 quando os nomes de coluna de destino e de origem não corresponderem. Para incluir o nome da coluna de origem (CUST_NUMBER) na Função de Propagação, especifique:

```
PROP(RAND(10000, 99999), CUST_NUMBER)
```

Exemplo 3

Você pode usar a Sequência Oracle para gerar o valor da coluna de destino e propagar esse valor nas colunas de destino das tabelas relacionadas. Para propagar a Sequência Oracle nomeada, **schema.numeven**, especifique:

```
PROP(schema.numeven.NEXTVAL)
```

Expressões Concatenadas

A concatenação permite combinar os valores de coluna ou combinar um valor de coluna com outro valor, usando um operador de concatenação (CONCAT, || ou +). Uma expressão concatenada pode incluir valores de caractere ou valores binários, mas não os dois:

Valores de Caractere

Os valores de caractere concatenado podem ser colunas de caracteres, sequências literais, subsequências de valores nas colunas de caractere, função sequencial ou função aleatória.

Valores Binários

Os valores de caractere concatenado podem ser colunas binárias, literais hexadecimais, subsequência de colunas binárias, função sequencial ou função aleatória.

- Uma expressão concatenada não pode incluir uma sequência literal de comprimento zero (' '), um registro especial ou Função Idade.

Exemplo

Suponha que a tabela CUSTOMERS armazene um endereço em duas colunas: ADDRESS1 e ADDRESS2. A tabela SHIP_TO armazena um endereço em uma coluna: ADDRESS. É possível usar uma expressão concatenada para combinar as informações de endereço de duas colunas em uma tabela para uma coluna em outra.

Para combinar o endereço, especifique um dos seguintes exemplos:

Tabela CUSTOMERS

ADDRESS1 || ADDRESS2

ADDRESS1 CONCAT ADDRESS2

ADDRESS1 + ADDRESS2

Tabela SHIP_TO

ADDRESS

ADDRESS

ADDRESS

Expressões numéricas

Use uma expressão numérica para especificar um valor na coluna de origem sempre que os tipos de dados para as colunas correspondentes de origem e de destino forem compatíveis.

Uma expressão numérica consiste na sintaxe a seguir:

```
operand1 operator operand2
```

Cada operando deve ser uma coluna numérica ou uma constante numérica. O operador especifica se adicionar (+), subtrair (-), dividir (/) ou multiplicar (*).

Exemplo 1

Para aumentar o valor em uma coluna UNIT_PRICE definida como DECIMAL(5,2) por 10 por cento, especifique o exemplo a seguir:

```
1.1 * UNIT_PRICE
```

Exemplo 2

Para dividir o valor em uma ON_HAND_INVENTORY do INTEGER pela metade, especifique o exemplo a seguir:

```
ON_HAND_INVENTORY / 2.
```

Funções de Valor e Literal

Use as funções de valor e literal para especificar literais como uma sequência ou hexadecimal e valores como NULL ou um registro especial.

Função	Descrição
Nome da Coluna	Um nome de coluna explícito (nomes de colunas de fazem distinção entre maiúsculas e minúsculas)
NULL	NULL. A coluna de destino deve ser anulável.
Constante Numérica	Uma constante numérica. O valor constante deve caber na coluna de destino conforme definido por seu tipo de dados, precisão e escala.
Constante Booleana	Uma constante booleana (TRUE ou FALSE).
Registro Especial	Um registro especial: <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
Sequência Literal	Uma sequência literal, colocada entre aspas simples. A coluna de destino deve conter os dados de caractere. Exemplo: 'CA' ou '90210'.
Literal Hexadecimal	Um literal hexadecimal. Exemplo: X'1234567890ABCDEF' ou 0X1234567890ABCDEF
Literal Data/Hora	Uma literal data/hora, colocada entre aspas simples. Separe a data e hora com um espaço. Para formatar a data/hora com uma fração decimal, coloque um ponto após o tempo, seguido pela fração. O formato da data é determinado pelas configurações em Opções Regionais no Painel de Controle do seu computador. Nota: Para Registro de Data e Hora Oracle com colunas de Fuso Horário, você deve especificar o sufixo de fuso horário por último.

Função Idade

Use a Função Idade para atribuir idades aos valores em uma coluna de origem. A coluna de origem pode conter dados de caracteres, numéricos, de data ou de registro de data e hora. Uma coluna CHAR ou VARCHAR tem um comprimento máximo de 256 bytes.

A Função Idade é formatada da seguinte forma:

AGE(*parameters*)

- Defina a Função Idade para incluir um ou mais parâmetros de idade.
- Utilize vírgulas ou espaços para separar parâmetros na Função Idade.
- Os parâmetros podem ser especificados em qualquer ordem.

Parâmetro	Formato	Valores Válidos
Nome da Coluna: Especifique o nome da coluna de origem se ele for diferente da coluna de destino.	SC= <i>column-name</i> SRCCOL= <i>column-name</i>	Nome da Coluna
Padrão: As datas de idade com base no valor de ajuste de data especificado em uma solicitação de processo.	DEF	Utiliza o valor de ajuste de data especificado na solicitação de processo.
Nenhum: Nenhum valor de idade.	NONE	O valor não deve expirar, independentemente de especificações no pedido de processo.
Incremental - A Idade Incremental é baseada em uma unidade de tempo conhecida. O Optim suporta unidades únicas (por exemplo, 20 anos) ou diversas unidades (por exemplo, 2 anos, 3 meses e 2 dias).	+ ou -] <i>nY</i> [+ ou -] <i>nM</i> [+ ou -] <i>nW</i> [+ ou -] <i>nD</i> (O sinal de mais [+] é opcional).	<i>nY</i> -2500 a +1581 <i>nM</i> -30000 a +30000 <i>nW</i> -30000 a +30000 <i>nD</i> -99999 a +99999
Ano Específico: Datas de idade com base em um ano específico de quatro dígitos no formato desejado.	<i>nnnnY</i>	1582 - 3999
Múltiplo/Regra: As datas de idade com base no número de vezes que uma regra de negócios é aplicada. Se você definir a Função Idade usando a Múltiplo/Regra, o parâmetro RULE também deverá ser incluído.	<i>nnnnnR</i>	1 - 30000

Expiração da Semântica

A Expiração de Semântica é baseada em um conjunto de regras que são definidas para gerenciar as datas que ocorrem em feriados, fins de semana, e assim por diante. É possível utilizar a Expiração de Semântica para ajustar datas para que elas ocorram nos dias úteis válidos.

Calendário –

Nome do calendário que define as datas especiais às quais as regras se aplicam. Se CALENDAR for usado, você também deverá especificar um RULE.

CA=*calendar-name*

CALENDAR=*calendar-name*

Regra –

Nome da regra que define o ajuste de datas especiais. Se DEF for especificado, a regra padrão especificada na solicitação de processo será utilizada.

RU=*rule-name*

RULE=*rule-name*
RU=DEF
RULE=DEF

Tabela Dinâmica de Século –

Determina os anos do século em dois dígitos. Insira um valor de 00 a 99.

PI=*nn*

PIVOT=*nn*

- Se você especificar **AGE(RU=DEF)**, o RULE especificado em uma solicitação de processo será usado. Você deve especificar valores para quaisquer outros parâmetros de função de idades.
- Se você utilizar RULE e não especificar um CALENDAR, a Função Idade usará o calendário padrão que é especificado em uma solicitação de processo.
- Se você não incluir CALENDAR, RULE e PIVOT onde necessário na Função Idade, serão aplicados os valores padrão especificados em uma solicitação de processo.
- Para especificar o século correto para um ano de dois dígitos, você deve incluir PIVOT na Função Idade.
- Se você especificar um valor PIVOT, todos os anos de dois dígitos iguais ou maiores que o valor PIVOT serão colocados no século 20 (19xx). Todos os anos de dois dígitos menores que o valor PIVOT serão colocados no século 21 (20xx). O PIVOT padrão é **65**.

Formatos de Datas

O formato de data de origem e o formato de data de destino devem conter um formato de data válido único e devem ser menor ou igual a comprimento da coluna de destino. A sequência de formatações deve ser delimitada por aspas simples.

Formato de Data de Origem -

Aplica a sequência de formatações da coluna de origem nas colunas de caractere de idade e numéricas.

SF=*'format-string'*

SRCFMT=*'format-string'*

Se a coluna de origem for caractere ou numérica, você deverá usar SRCFMT ou uma Rotina de Saída de Origem (SRCEXIT) para descrever o conteúdo da coluna. Esses parâmetros são mutuamente exclusivos.

Formato de Data de Destino -

Aplica a sequência de formatações da coluna de destino nas colunas de caractere de idade e numéricas.

DF=*'format-string'*

DSTFMT=*'format-string'*

Se a coluna de destino for caractere ou numérica, será possível especificar DSTFMT ou uma Rotina de Saída de Destino (DSTEXIT). Se você não especificar um formato para o destino, a função de expiração de data usará SRCFMT, por padrão. A coluna de destino para uma função AGE não pode ser binária.

Utilize as sequências de caracteres a seguir para especificar os componentes do formato de data:

Ano	Mês	Dia	Hora	Partes/Segundo
YYYY	MONTH	DDD	HH	FFFFFF
CCYY	MMM	DD	MI	FFFFFF
YY	MM	D	SS	FFFF
	M			FFF
				FF
				F

- Se você especificar um ponto de interrogação (?) em uma sequência de formatações, a Função Idade mapeará o valor do caractere no estado em que se encontra. (Utilize o ponto de interrogação para incluir barras, hífens, pontos e assim por diante, no formato de data).
- Se você especificar um asterisco (*) em uma sequência de formatações, a Função Idade mapeará quaisquer caracteres restantes na coluna de origem para a coluna de destino. (Utilize o asterisco quando o valor da coluna for uma data concatenada com caracteres adicionais).

Nota: É possível utilizar o Utilitário de Calendário para definir um separador padrão e um ano de saída padrão. Esses padrões se aplicam quando os formatos de origem e de destino requererem separadores ou um ano específico.

Exemplo 1

Para expirar uma coluna de data de 2 anos, 6 meses, 40 semanas e 15 dias, aplique uma regra e formate a Função Idade como:

```
AGE(+2Y,+6M,+40W,+15D,RU=NEXTPAYDAY)
```

Exemplo 2

Para expirar apenas a parte do ano em uma coluna de data para o ano 2020 e aplicar uma regra, formate a Função Idade como:

```
AGE(2020Y,RU=NEXTWORKDAY)
```

Exemplo 3

Para expirar uma coluna de data utilizando MULTIPLE/RULE para incrementar por cinco ocorrências de uma regra chamada NEXTSTRQTR, utilizando um calendário chamado PSAPRULE, formate a Função Idade como:

```
AGE(CA=PSAPRULE,RU=NEXTSTRQTR,5R)
```

Exemplo 4

Para expirar dados em uma coluna de caracteres ou numérica pelos seguintes parâmetros:

- Uma coluna de origem nomeada.
- O formato de origem, utilizando os dois primeiros caracteres para os últimos dois dígitos do ano e os 3 dígitos restantes como o dia no calendário Juliano.
- Uma tabela dinâmica de século determina o século correto porque a origem é formatada com um ano de dois dígitos. A tabela dinâmica do século neste exemplo é 42. Todos os anos de dois dígitos maiores ou iguais a 42 são colocados no século 20 (19xx). Todos os anos de dois dígitos inferiores a 42 são colocados no século 21 (20xx).
- Expirar data a cada 5 anos.

Formate a Função Idade como:

```
AGE(5Y,SC=ORDER_DATE,SF='YYDDD',PI=42)
```

Função de Moeda

Use a Função de Moeda para converter um valor monetário em uma coluna de origem de uma moeda para outra. A coluna de origem deve ser definida como numérica, mas não ponto flutuante. Dois métodos de conversão estão disponíveis:

Conversão Direta

Forneça parâmetros de conversão com base nos valores definidos em uma Definição de Moeda. Utilize a Função Moeda para converter um valor monetário em uma coluna (substituindo o valor anterior) ou, ao definir colunas de origem e de destino diferentes, mantenha o valor original e o valor convertido. É possível definir explicitamente os tipos de moeda de origem e de destino ou identificar uma coluna de referência para indicar o tipo de moeda.

A primeira preferência de cálculo é usar a taxa de conversão da moeda de origem na moeda de destino. A segunda preferência de conversão de cálculo é usar a taxa de conversão da moeda de destino na moeda de origem.

Triangulação

Converta o valor em uma coluna da moeda de origem para dólar europeu e, em seguida, converta o dólar europeu na moeda de destino. As duas taxas devem ser fornecidas na tabela de taxas: uma taxa entre o dólar europeu e a moeda de origem e outra taxa entre o dólar europeu e a moeda de destino. A expressão da especificação é TRIANG ou TR.

A Função Moeda é formatada da seguinte forma:

```
CURRENCY( {ST=code | SS=(column-name,Types Table number)}  
{DT=code | DS=(column-name,Types Table number)}  
[SC=column-name] [TR] [CU=Currency Definition name]  
[TD=transaction-date-column-name] [DF='format']  
[NS=scale] )
```

A Função Moeda deve incluir pelo menos uma combinação do tipo de moeda de origem (ST) ou especificação de origem (SS) e o tipo de moeda de destino (DT) ou especificação de destino (DS). Todos os outros parâmetros são opcionais.

Os tipos de moeda de origem e de destino podem ser especificados em uma de duas maneiras:

1. Use as palavras-chave ST/DT para permitir a especificação explícita da moeda utilizando o Código de Moeda ISO 4217 de três caracteres.
2. Use as palavras-chave SS/DS para permitir a especificação indireta do tipo de moeda em que um valor em uma coluna nomeada na linha é usado como uma chave. A chave é correlacionada com um tipo de moeda, conforme definido na Tabela de Tipo de Definição de Moeda especificada.

Se você especificar uma data de transação (TD) e a coluna de data de transação não usar o formato de DATE, você também deverá especificar um formato de data (DF). Se uma data de transação especificada estiver fora dos intervalos de data especificados na Tabela de Taxas de Definição de Moeda, o intervalo de data mais próximo será usado para cálculos de conversão.

Nota: Se quaisquer dados necessários (por exemplo, tipos de moedas ou taxas) estiverem ausentes, ocorrerão erros de conversão no tempo de execução.

A tabela a seguir descreve o formato válido e os valores permitidos para os parâmetros de Função de Moeda. Os parâmetros podem ser especificados em qualquer ordem. Utilize vírgulas ou espaços para separar parâmetros na Função Moeda.

Parâmetro	Formato
Coluna de Origem	SC=column-name SRCCOL=column name
Tipo de Moeda de Origem	ST=code SRCTYP=code em que code = Código da Moeda ISO 4217

Parâmetro	Formato
Tipo de Moeda de Destino	DT= <i>code</i> DSTTYP= <i>code</i> <i>em que code = Código da Moeda ISO 4217</i>
Especificação de Origem	SS= <i>column name, Types Table number</i> SRCSPC= <i>column name, Types Table number</i> <i>em que column name, Types Table number = é a coluna e o número de Tabela de Tipos (especificados na Definição de Moeda) a serem usados para especificar o tipo de moeda de origem.</i>
Especificação de Destino	DS= <i>column name, Types Table number</i> DSTSPC= <i>column name, Types Table number</i> <i>em que column name, Types Table number = é a coluna e o número de Tabela de Tipos (especificados na Definição de Moeda) a serem usados para especificar o tipo de moeda de destino.</i>
triangulação (Força a conversão por meio do dólar europeu)	TR TRIANG
Definição de Moeda	CU= <i>Currency Definition name</i> CURTBL= <i>Currency Definition name</i> <i>em que Currency Definition name = é a Definição de Moeda que contém os parâmetros de conversão apropriados.</i>
Data da Transação	TD= <i>column name</i> TRNDAT= <i>column name</i> <i>em que column name = Transaction Date column é o nome para identificar a data de conversão</i>
Formato de Data	DF= <i>'format'</i> DATFMT= <i>'format'</i> , <i>em que format = é o formato da coluna de data de transação, se não for do tipo Data.</i>
Escala Numérica	NS= <i>scale</i> NUMSCL= <i>scale</i> <i>em que scale = é a escala a ser aplicada nas colunas de destino numéricas do Oracle com uma escala não definida.</i>

Exemplo 1

Para converter de Markkas Finlandesas em Dólares Europeus, formate a Função de Moeda da seguinte forma:

```
CURRENCY(ST=FIM DT=EUR)
```

Se o valor original tiver que ser preservado, utilize a Função de Moeda para fornecer um valor para uma coluna diferente na tabela de destino.

Exemplo 2

Para converter de Markkas Finlandesas em Dólares Europeus, e criar uma nova coluna para reter o valor de origem original (em Markkas Finlandesas) em uma coluna denominada ITEM_COST, formate a função de Moeda como:

```
CURRENCY(ST=FIM DT=EUR SC=ITEM_COST)
```

Nome de Email Gerado Automaticamente

A função de nome de email gerado automaticamente gera um endereço de email com um nome de usuário baseado em um literal concatenado a um número sequencial. Os números sequenciais são sufixos que iniciam com 1 e são incrementados por 1. A função usa o nome de domínio de um endereço de email em uma coluna de origem especificada.

Nome de Email Formatado

A função de nome de email formatado gera um endereço de email com um nome de usuário com base em valores obtidos de um ou dois atributos. A política usa o nome de domínio de um endereço de email em uma coluna de origem especificada.

Função de Número Aleatório

A função de número aleatório gera números selecionados aleatoriamente dentro do intervalo indicado pelos valores baixo e alto.

É possível usar a função de número aleatório para substituir os dados de caractere ou numéricos. Os valores baixo e alto devem ser números inteiros dentro do intervalo -2,147,483,648 até 2,147,483,647. O valor baixo deve ser menor que o valor alto.

Função de Número Sequencial

A função de números sequenciais gera números que são incrementados sequencialmente.

É possível usar a função de números sequenciais para substituir dados de caractere ou numéricos. Você deve inserir um valor inicial e um valor pelo qual os números serão incrementados. Os valores inicial e incremental devem ser números inteiros dentro do intervalo -2,147,483,648 até 2,147,483,647.

O valor gerado é limitado pelo tipo e comprimento dos dados de destino. Se o valor gerado exceder o comprimento da coluna de destino, a função reconfigurará automaticamente para o valor inicial.

Usando Rotinas de Saída

Ao criar um serviço para converter, criar, inserir, carregar ou restaurar dados, é possível especificar um mapa de tabela que inclua um ou mais mapas de coluna para derivar os valores apropriados para colunas de destino.

Há várias maneiras para especificar um valor de coluna em um mapa de colunas. Uma maneira é especificar uma rotina de saída como a origem; a saída configura valores que de outra forma não poderiam ser definidos para as colunas de destino. Outra maneira é usar rotinas de saída para excluir linhas de processamento. É possível usar três tipos de rotinas de saída:

Saída Padrão

A rotina de Saída Padrão é chamada para derivar o valor de uma coluna de destino em um mapa de coluna. Este tipo de rotina de saída é útil quando quiser executar transformações de dados que estão além do escopo de mapas de coluna. Por exemplo, uma saída pode alterar um número de departamento de funcionário para linhas selecionadas de acordo com um algoritmo complexo ou selecionar linhas específicas a serem processadas e descartar todas as outras.

Um de saída padrão pode obter um segmento de subsequência de uma coluna LOB de origem. Para inserir um novo valor de LOB em uma coluna LOB de destino, uma saída pode criar um arquivo e transmitir o nome do arquivo de volta para o processador de mapa de coluna.

Saída do Formato de Origem

A Saída do Formato de Origem é chamada para formatar a coluna de origem em uma Função Idade que, de outra forma, não seria suportado em um mapa da coluna. Essa rotina de saída examina a data de origem em uma coluna de caractere ou de número inteiro e o converte-a em um formato de data utilizado como entrada para a Função Idade.

Saída do Formato de Destino

A Saída do Formato de Destino é chamada para formatar a coluna de destino em uma Função Idade que, de outra forma, não seria suportado em um mapa da coluna. Essa rotina de saída converte uma data em um dos quatro formatos de destino diferentes. O tipo de dados da coluna de destino determina o formato.

Saída em um Mapa da Coluna

Para utilizar uma rotina de saída em um mapa de coluna, você deverá especificar um dos seguintes itens na coluna de origem apropriada:

Saída Padrão

```
EXIT dllname[(parm[,]parm)... ]
```

Saída do Formato de Origem

```
AGE(SRCEXIT=dllname)
```

Saída do Formato de Destino

```
AGE(DSTEXIT=dllname)
```

O processo chama a rotina de saída do mapa de coluna uma vez para cada linha de dados processada e transmite uma chamada de finalização após a última linha ser processada. Os parâmetros opcionais especificados com uma rotina de saída padrão são transmitidos para a saída, devendo ser uma sequência (entre aspas simples) ou literais numéricos (limite de 8).

Gravando Rotinas de Saída

É possível gravar rotinas de saída em qualquer linguagem de programação; no entanto, as chamadas para as sub-rotinas devem estar de acordo com as convenções utilizadas na linguagem de programação C.

Arquivos de Cabeçalho

Para definir os parâmetros, estruturas e códigos de retorno utilizados na rotina de saída, uma rotina de saída deve incluir dois arquivos de cabeçalho do programa C, PSTEXIT.H e, dependendo do formato de caractere dos metadados (nomes de tabelas, nomes de colunas, etc.), PSTCMXIT.H ou PSTCMWXT.H:

PSTEXIT.H

Especifica os tipos de dados, códigos de retorno e estruturas para tipos de dados definidos pelo Optim.

PSTCMXIT.H

Para metadados em formato de byte único (por exemplo, ASCII).

Fornece os protótipos para rotinas de retorno de mapa da coluna e especifica *define* e *definições de estrutura* para os parâmetros de saída do mapa da coluna.

PSTCMWXT.H

Para metadados no formato UTF-16 (WCHAR).

Fornece os protótipos para rotinas de retorno de mapa da coluna e especifica *define* e *definições de estrutura* para os parâmetros de saída do mapa da coluna.

Esses arquivos de cabeçalho estão localizados no mesmo diretório com os arquivos de aplicativo do IBM InfoSphere Optim.

Nota: Se um parâmetro tiver um formulário de byte único e um formulário UTF-16, este capítulo fornecerá o formato UTF-16 entre parênteses.

Utilizando DLLs

Você deve compilar e vincular cada rotina de saída como uma DLL separada. O Optim carrega cada DLL dinamicamente no tempo de execução. Uma DLL pode conter apenas uma ocorrência de um tipo específico de saída do mapa da coluna. No entanto, é possível incluir um de cada tipo de rotina de saída do mapa da coluna na mesma DLL.

Você deve usar o nome da DLL no mapa da coluna. Especifique o nome da DLL, conforme está definido para a plataforma (ou seja, não utilize o nome genérico). Grave o nome de DLL e exporte as funções reais que implementam a rotina de saída:

Saída Padrão

```
PSTColMapExit  
(PSTColMapWExit)
```

Saída do Formato de Origem

```
PSTColMapAgeSrcExit  
(PSTColMapAgeSrcWExit)
```

Saída do Formato de Destino

```
PSTColMapAgeDstExit  
(PSTColMapAgeDstWExit)
```

Requisitos

Cada rotina de saída do mapa da coluna deve atender aos seguintes requisitos:

- A saída deve gerar um valor apropriado para a coluna de destino e não deve alterar nenhuma outra área de armazenamento.
- O Optim utiliza a chave primária para criar áreas de trabalho internas. Se a rotina de saída designar um valor para uma coluna que faz parte da chave primária, e se essa rotina for chamada várias vezes para a mesma linha de origem, certifique-se de que a rotina de saída gere o mesmo valor de saída todas as vezes.
- Como uma saída pode ser chamada com frequência, evite sobrecarga desnecessária. Por exemplo, a saída deve inicializar o processamento na primeira chamada e salvar informações de chamadas subsequentes na área de trabalho.

Rotina de Saída Padrão

Ao gravar uma Saída Padrão, você especifica a função de saída, PSTColMapExit (PSTColMapWExit) e os seguintes parâmetros:

byte único

```
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)
```

Parâmetros

Quando um processo chama uma rotina de saída padrão, o processo transmite os parâmetros a seguir em cada chamada:

pInputParms

Ponteiro para PST_STRUCT_CM_EXIT_PARM (PST_STRUCT_CM_WEXIT_PARM). Essa estrutura contém informações sobre as tabelas de origem e destino, a natureza da chamada atual, o número de parâmetros opcionais especificados com a saída e ponteiros para as funções de retorno de chamada, áreas de trabalho e parâmetros opcionais.

O primeiro campo desta estrutura é o campo **FuncCode** identificado por PST_CM_FUNC_TRANSFORM (PST_CMW_FUNC_TRANSFORM) ou PST_CM_FUNC_TERMINATE (PST_CMW_FUNC_TERMINATE).

O campo *NumParms* contém o número de parâmetros opcionais especificados com a saída do mapa da coluna (0 a 8).

O campo *pParm* contém uma matriz de ponteiros para cada parâmetro opcional especificado com a saída do mapa da coluna.

pScrColList

Ponteiro para PST_STRUCT_CM_EXIT_COL_LIST (PST_STRUCT_CM_WEXIT_COL_LIST). Esta estrutura descreve as colunas de origem.

pDstColList

Ponteiro para PST_STRUCT_CM_EXIT_COL_LIST (PST_STRUCT_CM_WEXIT_COL_LIST). Esta estrutura descreve as colunas de destino.

Funções de Retorno de Chamada

Sempre que um processo do Optim chama uma rotina de saída padrão, o processo transmite os endereços das seguintes funções de retorno de chamada:

pPSTGetColValue()

Recupera dados de todas as colunas de origem e a maioria das colunas de destino na linha de dados atual. Em geral, a rotina de saída chama essa função apenas uma vez para recuperar os dados de uma coluna de origem. No entanto, a rotina de saída pode chamar essa função várias vezes para recuperar os dados de colunas diferentes.

pPSTPutColValue()

Especifica de dados para a coluna de destino na linha de dados atual. A rotina de saída determina o valor para a coluna de destino e retorna o valor para Optim. Esta função deve ser chamada a menos que a linha seja rejeitada ou o processo interrompido.

Processamento

O processamento típico para a rotina de Saída Padrão está resumido nas etapas a seguir:

1. Em cada chamada a partir do Optim, a rotina de saída verifica se há uma chamada de primeira vez. Na primeira chamada, a saída executa todas as tarefas de inicialização e o processamento normal (etapa 2). Em chamadas subsequentes, apenas o processamento normal é executado (etapa 2).
2. O Optim não transmite os dados da coluna para a rotina de saída padrão; entretanto, a rotina de saída pode fazer chamadas para a função de retorno de chamada **pPSTGetColValue()** para obter dados para as colunas de origem. Esses valores são necessários para determinar o valor da coluna de destino.
3. Após a rotina de saída gerar o valor de destino, a saída também chama a função de retorno de chamada **pPSTPutColValue()** para armazenar o valor na coluna de destino ou transmite um código de retorno apropriado instruindo o processo a ignorar a linha de dados ou interromper.

4. Após a última linha de dados ser processada, o Optim passará a chamada de término para a rotina de saída identificada por um valor de PST_CM_FUNC_TERMINATE (PST_CMW_FUNC_TERMINATE) no campo FuncCode. Esta chamada solicita a rotina de saída para liberar qualquer armazenamento alocado dinamicamente. Quando tarefas finais são concluídas, a rotina de saída transmitirá um código de retorno para o Optim.

Códigos de Retorno

Os seguintes códigos de retorno se aplicam às rotinas de saída padrão:

PST_CM_EXIT_SUCCESS
(PST_CMW_EXIT_SUCCESS)

Uma coluna de destino recebe um valor ou é transformada com êxito.

PST_CM_EXIT_REJECT_ROW
(PST_CMW_EXIT_REJECT_ROW)

A coluna de destino não pode receber um valor ou ser transformada. Descarte a linha.

PST_CM_EXIT_ABORT_PROCESS
(PST_CMW_EXIT_ABORT_PROCESS)

Erro fatal. Finalizando processamento. Para retornar uma mensagem de erro, coloque a mensagem na área de trabalho e configure o espaço não utilizado para espaços em branco ou NULL.

Saída do Formato de Origem

Ao gravar uma rotina de Saída do Formato de Origem do Optim, você especifica a função de saída, PSTColMapAgeSrcExit (PSTColMapAgeSrcWExit) e os seguintes parâmetros:

byte único

```
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)
```

Parâmetros

Quando um processo do Optim chama uma rotina de saída do formato de origem, o processo transmitirá os seguintes parâmetros:

pInputParms

Ponteiro para PST_STRUCT_CM_AGE_SRCFMT_PARM (PST_STRUCT_CM_AGE_SRCFMT_WPARAM). Essa estrutura contém informações sobre as tabelas de origem e destino, a natureza da chamada atual e ponteiros para uma função de retorno de chamada e áreas de trabalho.

O primeiro campo desta estrutura é o campo **FuncCode** identificado pelo PST_CM_SRCFMT_TRANSFORM (PST_CMW_SRCFMT_TRANSFORM) ou PST_CM_SRCFMT_TERMINATE (PST_CMW_SRCFMT_TERMINATE).

pSrcColList

Ponteiro para PST_STRUCT_CM_EXIT_COL_LIST (PST_STRUCT_CM_WEXIT_COL_LIST). Esta estrutura descreve as colunas de origem.

pDstColList

Ponteiro para PST_STRUCT_CM_EXIT_COL_LIST (PST_STRUCT_CM_WEXIT_COL_LIST). Esta estrutura descreve as colunas de destino.

Função de Retorno de Chamada

Sempre que um processo do Optim chama uma rotina de saída do formato de origem, o processo transmitirá o endereço da seguinte função de retorno de chamada:

pPSTGetColValue()

Recupera dados de todas as colunas de origem e a maioria das colunas de destino na linha de dados atual. Em geral, a rotina de saída não é necessária para chamar essa função porque os dados de uma coluna de origem são fornecidos no primeiro parâmetro. No entanto, a rotina de saída pode chamar essa função para recuperar os dados de colunas diferentes.

Processamento

O processamento típico para a rotina de Saída do Formato de Origem está resumido nas etapas a seguir:

1. Em cada chamada a partir do Optim, a rotina de saída verifica se há uma chamada de primeira vez. Na primeira chamada, a saída executa todas as tarefas de inicialização e o processamento normal (etapa 2). Em chamadas subsequentes, apenas o processamento normal é executado (etapa 2).
2. A saída recebe o valor da coluna de origem, conforme especificado na Função Idade definida no mapa da coluna. O Optim não aplica a Função Idade antes de chamar a rotina de saída e armazena o valor bruto em um dos campos de união **InputValue**, conforme indicado pelo campo **ValueType** no arquivo de cabeçalho.
3. Se a saída precisar examinar outras colunas para calcular o valor da coluna de destino, a saída deverá chamar a função de retorno de chamada **pPSTGetColValue()** para obter o valor dessas colunas.
4. Após o valor de destino ser gerado, a rotina de saída deverá formatar o valor e colocá-lo no campo **OutputTimeStamp** ou **OutputSybDateTime**. O Optim valida este valor e aplica a Função Idade. A saída deve transmitir um código de retorno apropriado indicando o campo onde os dados são salvos ou instruir o processo a ignorar a linha de dados ou interromper.
5. Após a última linha de dados ser processada, o Optim passará a chamada de término para a rotina de saída identificada por um valor de PST_CM_SRCFMT_TERMINATE (PST_CMW_SRCFMT_TERMINATE) no campo **FuncCode**. Esta chamada solicita a rotina de saída para liberar qualquer armazenamento alocado dinamicamente. Quando tarefas finais são concluídas, a rotina de saída transmitirá um código de retorno para o Optim.

Modos de Interrupção

Existem várias maneiras pelas quais a rotina de Saída do Formato de Origem pode interromper o processamento:

- Processar linhas com datas ignoradas ou datas inválidas. Se você selecionar qualquer uma dessas opções em uma solicitação de processo, e se as colunas de origem e de destino possuírem os mesmos atributos, a coluna de origem será copiada inalterada para a coluna de destino. Se você não selecionar nenhuma das opções, a linha será rejeitada.
- Rejeitar a linha. Rejeite a linha independentemente das opções do processo para datas ignoradas ou inválidas, com base nas especificações na rotina de saída.
- Interromper todo o processo, com base em especificações da rotina de saída.

A rotina de saída transmite um código de retorno indicando qual formato de data ou qual modo de interrupção será usado.

Durante um processo, a rotina de saída pode interrogar todas as colunas na linha de entrada e algumas das colunas na linha de destino. No entanto, a rotina de saída não pode interrogar uma coluna de destino

que incluir uma rotina de saída e é definida no mapa da coluna após a coluna de destino atual. Todas as outras colunas de destino estão disponíveis.

Códigos de Retorno

Os seguintes códigos de retorno se aplicam às saídas do formato de origem:

PST_CM_SRCFMT_USE_TIMESTAMP
(PST_CMW_SRCFMT_USE_TIMESTAMP)

É designado um valor para a coluna de destino no campo **OutputTimeStamp** do primeiro parâmetro transmitido para a saída.

PST_CM_SRCFMT_USE_SYB_DATETIME
(PST_CMW_SRCFMT_USE_SYB_DATETIME)

É designado um valor para a coluna de destino no campo **OutputSybDateTime** do primeiro parâmetro transmitido para a saída.

PST_CM_SRCFMT_SKIP
(PST_CMW_SRCFMT_SKIP)

O fator idade não é aplicado. Se você não selecionar a opção para **Processar linhas com datas ignoradas**, a linha será rejeitada. Caso contrário, os dados serão copiados para a origem, contanto que a origem e o destino sejam compatíveis. Se não forem compatíveis, a linha será rejeitada.

PST_CM_SRCFMT_COL_INVALID
(PST_CMW_SRCFMT_COL_INVALID)

O fator idade não é aplicado. Se você não selecionar a opção para **Processar linhas com datas inválidas**, a linha será rejeitada. Caso contrário, os dados serão copiados para a origem, contanto que a origem e o destino sejam compatíveis. Se não forem compatíveis, a linha será rejeitada.

PST_CM_SRCFMT_REJECT_ROW
(PST_CMW_SRCFMT_REJECT_ROW)

A coluna de origem não pode receber um valor. Rejeite (descarte) a linha.

PST_CM_SRCFMT_ABORT_PROCESS
(PST_CMW_SRCFMT_ABORT_PROCESS)

Erro fatal. Finalizando. Para retornar uma mensagem de erro, coloque a mensagem na área de trabalho e configure o espaço não utilizado para espaços em branco ou NULL.

Saída do Formato de Destino

Ao gravar uma rotina de Saída do Formato de Destino do Optim, você especifica a função de saída, PSTColMapAgeDstExit (PSTColMapAgeDstWExit) e os seguintes parâmetros:

byte único

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)

Parâmetros

Quando um processo do Optim chama uma rotina de saída do formato de destino, o processo transmitirá os seguintes parâmetros:

pInputParms

Ponteiro para PST_STRUCT_CM_AGE_DSTFMT_PARM (PST_STRUCT_CM_AGE_DSTFMT_WPARAM). Essa estrutura contém informações sobre as tabelas de origem e destino, a natureza da chamada atual e ponteiros para uma função de retorno de chamada e áreas de trabalho.

O primeiro campo desta estrutura é o campo **FuncCode**, identificado por:

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), ou
PST_CM_DSTFMT_TERMINATE (PST_CMW_DSTFMT_TERMINATE).

pSrcColList

Ponteiro para um PST_STRUCT_CM_EXIT_COL_LIST (PST_STRUCT_CM_WEXIT_COL_LIST). Esta estrutura descreve as colunas de origem.

pDstColList

Ponteiro para um PST_STRUCT_CM_EXIT_COL_LIST (PST_STRUCT_CM_WEXIT_COL_LIST). Esta estrutura descreve as colunas de destino.

Função de Retorno de Chamada

Sempre que um processo do Optim chama uma rotina de saída do formato de destino, o processo transmite o endereço da seguinte função de retorno de chamada:

pPSTGetColValue()

Recupera dados de todas as colunas de origem e a maioria das colunas de destino na linha de dados atual. Em geral, a rotina de saída não é necessária para chamar essa função porque os dados de uma coluna de origem de idade são fornecidos no primeiro parâmetro. No entanto, a rotina de saída pode chamar essa função para recuperar os dados de colunas diferentes.

Formatos

A data de entrada está nos formatos PST_C_TIMESTAMP e PST_C_SYB_DATETIME. A saída é direcionada para transformar esse data em um dos seguintes formatos, com base no tipo de dados da coluna de destino.

PST_C_CHAR_SZ

Colunas de destino CHAR e VARCHAR.

PST_C_INTEGER_CHAR_SZ

Colunas de destino NUMERIC.

PST_C_TIMESTAMP

Colunas DB2 e Oracle DATE/TIME.

PST_C_SYB_DATETIME

Colunas Sybase ASE DATETIME.

Processamento

O processamento típico para a rotina de Saída do Formato de Destino está resumido nas etapas a seguir:

1. Em cada chamada a partir do Optim, a rotina de saída verifica se há uma chamada de primeira vez. Na primeira chamada, a saída executa todas as tarefas de inicialização e o processamento normal (etapa 2). Em chamadas subsequentes, apenas o processamento normal é executado (etapa 2).
2. A saída recebe o valor da coluna de origem, conforme especificado na Função Idade definida no mapa da coluna. O Optim aplica a Função Idade antes de chamar a saída do mapa da coluna e armazena o valor de idade nos campos **InputTimeStamp** e **InputSybaDateTime** no arquivo de cabeçalho.

3. Se a saída precisa examinar outras colunas para calcular o valor da coluna de destino, a saída deverá chamar a função de retorno de chamada **pPSTGetColValue()** para obter o valor dessas colunas.
4. Após o valor de destino for gerado, a rotina de saída deverá formatar o valor e colocá-lo em um dos campos na união **OutputValue**. O campo **FuncCode** indica o campo na união **OutputValue** no qual o valor deve ser colocado. A saída deve retornar um código apropriado indicando o campo onde os dados são salvos ou instruir o processo a ignorar a linha ou interromper.
5. Após a última linha de dados ser processada, o Optim passará a chamada de término para a rotina de saída identificada por um valor de **PST_CM_DSTFMT_TERMINATE** (**PST_CMW_DSTFMT_TERMINATE**) no campo **FuncCode**. Esta chamada solicita a rotina de saída para liberar qualquer armazenamento alocado dinamicamente. Quando tarefas finais são concluídas, a rotina de saída transmitirá um código de retorno para o Optim.

Modos de Interrupção

Existem várias maneiras pelas quais a rotina de Saída do Formato de Destino pode interromper o processamento:

- Processar linhas com datas ignoradas ou datas inválidas. Se você selecionar qualquer uma dessas opções em uma solicitação de processo, e se as colunas de origem e de destino possuírem os mesmos atributos, a coluna de origem será copiada inalterada para a coluna de destino. Se você não selecionar nenhuma das opções, a linha será rejeitada.
- Rejeitar a linha. Rejeite a linha independentemente das opções do processo para datas ignoradas ou inválidas, com base nas especificações na rotina de saída.
- Interromper todo o processo, com base em especificações da rotina de saída.

A rotina de saída transmite um código de retorno indicando se a conversão foi bem sucedida ou qual modo de interrupção será usado.

Durante um processo, a rotina de saída pode interrogar todas as colunas na linha de entrada e algumas das colunas na linha de destino. No entanto, a rotina de saída não pode interrogar uma coluna de destino que incluir uma rotina de saída e é definida no mapa da coluna após a coluna de destino atual. Todas as outras colunas de destino estão disponíveis.

Códigos de Retorno

Os seguintes códigos de retorno se aplicam às saídas do formato de destino:

PST_CM_DSTFMT_SUCCESS
(**PST_CMW_DSTFMT_SUCCESS**)

É designado um valor para a coluna de destino, conforme especificado no campo **FuncCode** do primeiro parâmetro transmitido para a saída.

PST_CM_DSTFMT_SKIP
(**PST_CMW_DSTFMT_SKIP**)

Se você não selecionar a opção para **Processar linhas com datas ignoradas**, a linha será rejeitada. Caso contrário, os dados de origem serão copiados para o destino, contanto que a origem e o destino sejam compatíveis. Se não forem compatíveis, a linha será rejeitada.

PST_CM_DSTFMT_COL_INVALID
(**PST_CMW_DSTFMT_CÔL_INVALID**)

Se você não selecionar a opção para **Processar linhas com datas inválidas**, a linha será rejeitada. Caso contrário, os dados serão copiados para a origem, contanto que a origem e o destino sejam compatíveis. Se não forem compatíveis, a linha será rejeitada.

PST_CM_DSTFMT_REJECT_ROW
(**PST_CMW_DSTFMT_REJECT_ROW**)

A coluna de destino não pode receber um valor. Rejeite (descarte) a linha.

```
PST_CM_DSTFMT_ABORT_PROCESS  
(PST_CMW_DSTFMT_ABORT_PROCESS)
```

Erro fatal. Finalizando. Para retornar uma mensagem de erro, coloque a mensagem na área de trabalho e configure o espaço não utilizado para espaços em branco ou NULL.

Gravando Procedimentos de Mapa de Coluna com o Script Lua

O procedimento de mapa da coluna é usado para mascarar ou transformar dados em uma coluna quando um serviço é executado. Como o nome indica, você deve incluir os procedimentos de mapa da coluna em um mapa da coluna. É possível gravar procedimentos de mapa da coluna usando a linguagem de script Lua.

Para obter mais informações sobre a linguagem de programação Lua, consulte o website em <http://www.lua.org>

Funções de Lua para Procedimentos de Mapa de Coluna

Os procedimentos de mapa da coluna suportam a maioria das funções de Lua padrão. Os procedimentos de mapa da coluna também suportam funções específicas do Optim.

Nome de Função Padrão para Funções Definidas pelo Usuário

Use os seguintes nomes de função nos procedimentos de mapa da coluna de Lua. Cada uma dessas três funções é chamada automaticamente no ponto indicado. Também é possível criar funções com outros nomes de função e chamá-las a partir das funções padrão.

Nome	Descrição	Exigido
<code>cm_load()</code>	Essa função é chamada antes que quaisquer tabelas sejam processadas.	Não
<code>cm_unload()</code>	Essa função é chamada depois que todas as tabelas são processadas.	Não
<code>cm_starttable()</code>	Essa função é chamada no início do processamento de cada tabela.	Não
<code>cm_endtable()</code>	Essa função é chamada no final do processamento de cada tabela.	Não
<code>cm_transform()</code>	Essa função é chamada para cada linha processada.	Sim

Cada procedimento de mapa da coluna deve conter uma função `cm_transform()`. Se a função `cm_transform()` não existir, o código será avaliado como uma expressão complexa.

Funções Globais

As seguintes funções estão disponíveis em todos os contextos de execução do procedimento de mapa da coluna:

Nome	Descrição
<code>numparms()</code>	Obter o número de parâmetros que são passados ao procedimento de mapa da coluna.
<code>parms.get(<i>n</i>)</code>	Obter o valor do parâmetro no índice <i>n</i> .
<code>print()</code>	Imprimir mensagens para o relatório do processo.
<code>rejectrow()</code>	Ignorar agora e acessar a próxima linha.

Funções de Armazenamento de Dados

Use as funções a seguir para obter informações sobre os armazenamentos de dados de destino e de origem:

Nome	Descrição
<code>source.getdbalias()</code>	Obter o alias do BD do armazenamento de dados de origem.
<code>source.getcreatorid()</code>	Obter o ID do criador do armazenamento de dados de origem.
<code>target.getdbalias()</code>	Obter o alias do BD do armazenamento de dados de destino.
<code>target.getcreatorid()</code>	Obter o ID do criador do armazenamento de dados de destino.

Funções da Tabela

Use as funções a seguir para obter informações sobre as tabelas de origem e destino:

Nome	Descrição
<code>source.gettablename()</code>	Obter o nome da tabela de origem.
<code>target.gettablename()</code>	Obter o nome da tabela de destino.

Funções de Coluna

Use as funções a seguir para obter informações sobre as colunas de origem e destino, transformar dados de coluna, gravar o resultado na coluna de destino e sair:

Nome	Descrição
<code>source.column.getvalue()</code>	Obter um valor não numérico a partir de uma coluna de origem. Um erro de tempo de execução será gerado se a função <code>source.column.getvalue()</code> for usada para obter valores de colunas de tipo de dados <code>PST_SQL_BINARY</code> .
<code>source.column.getasdouble()</code>	Obter um valor numérico a partir de uma coluna de origem (os procedimentos de mapa da coluna processam os dados numéricos no formato de precisão dupla). Use essa função para colunas dos seguintes tipos de dados: <code>PST_SQL_DECIMAL</code> , <code>PST_SQL_DOUBLE</code> , <code>PST_SQL_FLOAT</code> , <code>PST_SQL_INFX_DEC_FLOAT</code> e <code>PST_SQL_ORA_NUMBER</code> .
<code>source.column.getlength()</code>	Obter o comprimento da sequência em uma coluna de origem.
<code>source.column.getname()</code>	Obter o nome da coluna de origem.
<code>source.column.gettype()</code>	Obter o tipo de dados da coluna de origem.
<code>target.column.setvalue()</code>	Configurar o valor de uma coluna de destino. Um erro de tempo de execução será gerado se você usar a função <code>target.column.setvalue()</code> para configurar valores em colunas de tipo de dados <code>PST_SQL_BINARY</code> .

Nome	Descrição
<code>target.column.getlength()</code>	Obter o comprimento da sequência em uma coluna de destino.
<code>target.column.getname()</code>	Obter o nome da coluna de destino.
<code>target.column.gettype()</code>	Obter o tipo de dados da coluna de destino.
<code>optimmask()</code>	Chamar um provedor Optim Data Privacy Providers (ODPP).
<code>userexit()</code>	Chamar uma saída de usuário.

Funções Não Suportadas

As seguintes categorias de funções não são suportadas nos procedimentos de mapa da coluna.

- Recursos de entrada e saída integrados na biblioteca `io` de Lua
- `string.dump()`

Limitações de Procedimentos de Mapa de Coluna

Tenha em mente as seguintes limitações ao gravar um procedimento de mapa da coluna.

Dados Numéricos Processados no Formato de Precisão Dupla

Os procedimentos de mapa da coluna processam dados numéricos no formato de precisão dupla. Portanto, você deverá usar a função `source.column.getasdouble()` para obter os dados numéricos a partir das colunas de origem.

Codificação

Os procedimentos de mapa da coluna usam a codificação UTF-16 em seu processamento interno.

Tipo de Dados `PST_SQL_BINARY` Não Suportado

Um erro de tempo de execução será gerado se a função `source.column.getvalue()` for usada para obter valores de colunas de tipo de dados `PST_SQL_BINARY`. Um erro de tempo de execução também será gerado se você usar a função `target.column.setvalue()` para configurar valores em colunas de tipo de dados `PST_SQL_BINARY`.

Funções Não Suportadas

As seguintes categorias de funções não são suportadas nos procedimentos de mapa da coluna.

- Recursos de entrada e saída integrados na biblioteca `io` de Lua
- `string.dump()`

Exemplo de Procedimento de Mapa de Coluna: Procedimento Genérico

O exemplo de procedimento de mapa da coluna ilustra a estrutura de um procedimento de mapa da coluna com suas funções padrão: `cm_load`, `cm_unload`, `cm_starttable`, `cm_endtable` e `cm_transform`.

```
-----
--
-- Procedimento de mapa de coluna de amostra do IBM Optim
--
-- Nome:          OptimSample
--
-- Revisão:      1.0
--
```

```

-- Descrição:  Demonstra todos os recursos dos procedimentos de mapa da coluna do Optim/Lua
--
-- Entrada:    Zero ou mais parâmetros, que serão simplesmente ecoados de volta para o
--            relatório do processo do Optim
--
-- Saída:     Seção no relatório do processo do Optim mostrando informações deste
--            procedimento de mapa da coluna. A própria coluna é deixada inalterada.
--
--
-----

```

```

-----
-- Função cm_load - Chamada antes que quaisquer tabelas sejam processadas
-----

```

```

function cm_load()

  print(" *** Start of Process ***")

  colinfoshown = false

  -- Exibir parâmetros passados a partir do Mapa de Coluna

  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

```

```

-----
-- Função cm_unload - Chamada após todas as tabelas terem sido processadas
-----

```

```

function cm_unload()

  print(" *** End of Process ***")

end

```

```

-----
-- Função cm_starttable - Chamada no início do processamento para cada tabela
-----

```

```

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

```

```

-----
-- Função cm_endtable - Chamada no final do processamento de cada tabela
-----

```

```

function cm_endtable()

  print(" \nEnd of processing table")

end

```

```

-----
-- Função cm_transform - Chamada para cada linha processada
-----

```

```

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

-- Essa instrução obtém o valor na coluna para a qual cm_transform foi chamada
-- Opcionalmente, o nome de outra coluna pode ser especificado, por exemplo:
-- source.column.getvalue("COL1") retornará o valor na coluna COL1
oldvalue = source.column.getvalue()

-- Esse código configura essa coluna de destino para o mesmo valor que a coluna de origem
-- A lógica para alterar o valor seria colocada aqui.
-- Se você NÃO deseja inserir essa linha na tabela de destino, chame a função rejectrow()
newvalue = oldvalue
target.column.setvalue(newvalue)
end

```

Exemplo de Procedimento de Mapa de Coluna: Consulta Alternada

O exemplo de procedimento de mapa da coluna ilustra como a função `mask_parms` mascara uma coluna usando dados de uma tabela de consulta.

```

-----
--
-- Procedimento de mapa de coluna de amostra do IBM Optim
--
-- Nome:          OptimSwitchedLookup
--
-- Revisão:      1.0
--
-- Descrição:    Mascara uma coluna usando consulta de tabela. A tabela de consulta a
--               ser usada é determinada pelo valor de outra coluna.
--
-- Entrada:      Parâmetro 1 (Obrigatório):
--               Uma sequência que indica o tipo de consulta a ser usada:
--               HASH, RANDOM ou LOOKUP
--
--               Parâmetro 2 .... n-1 (Obrigatório)
--               Uma expressão que indica a tabela de consulta a ser usada. O formato é
--               COND(column-name=value) DATASOURCE(datasource_parameters)
--               Esse parâmetro pode ser repetido diversas vezes. Se uma linha não
--               satisfizer nenhum dos parâmetro COND, ela não será inserida
--               na tabela de destino.
--
--               Parâmetro n (Opcional):
--               Uma sequência contendo parâmetros adicionais a serem
--               copiados na chamada de optimmask.
--               Isso é além do valor datasource_parameters na
--               cláusula COND do Parâmetro 2 e do campo mask_parms_constant
--               que é declarado no início desse script.
--
--
-- Função de saída: - Os dados da coluna mascarada, conforme configurado pela função
target.column.setvalue
--                 - Texto dirigido ao relatório do Optim pela função de impressão
--
-- Códigos de Retorno: 0 - execução bem-sucedida
--                   1 - Rejeitar linha (Use em cm_transform; a linha não será inserida na
--                   tabela de destino)
--                   2 - Interromper processo
--
-- Duas funções auxiliares são especificadas para definir condições diferentes
-- do que o código de retorno 0.

```

```

--          Não há necessidade de codificar uma instrução de retorno ao usar essas
--          funções.
--          error(string) - Essa chamada faz com que o processo do Optim seja
--                          interrompido e sequência seja mostrada no relatório
--                          do Optim como uma mensagem de erro.
--          rejectrow()   - Essa chamada faz com que o Optim rejeite a linha
--                          que está sendo processada atualmente. A linha não
--                          é inserida na tabela de destino.
--
--
-----

```

```
function cm_transform()
```

```

-- Altere esse campo para conter os parâmetros que devem
-- ser colocados em todas as chamadas de optimask
local mask_parms_constant = 'CACHE=Y,WHENNF=PRE'

--
-- Valide o número dos parâmetros:
--
nparm = numparms()
if (nparm < 2) then
    process_error("Call to column map procedure OptimSwitchedLookup must h
ave 2 or more parameters")
end

--
-- Parâmetro de processo 1 (tipo de consulta)
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

--
-- Parâmetros COND/DATASOURCE de processo
--
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

-- Nenhuma COND correspondeu a essa linha, portanto, rejeite a linha
if (datasource_parameters <= "") then
    rejectrow()
end

--

```

```

-- Parâmetro de processo adicional opcional optimmask
--
lastparm = parms.get(nparm-1)
if (string.lower(string.sub(lastparm, 1, 5)) == "cond(") then
    optimmask_additional_parms = ""
else
    optimmask_additional_parms = lastparm
end

--
-- Construa a chamada para optimmask, faça a chamada
-- e coloque o novo valor na coluna de destino
--
mask_parms = "PRO=" .. provalue .. "," .. mask_parms_constant

-- Esse uso das variáveis de ambiente para armazenar ID do usuário e senha
-- para a tabela de sistema é apenas para propósito meramente ilustrativo.
-- Para obter uma segurança maior, armazene essas informações nas variáveis de ambiente
-- em um formato criptografado.
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    -- Aponte para o primeiro caractere após "COND("

    -- Obtenha o nome da coluna
    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)

    -- Obtenha o valor da coluna
    strptr = equalsign + 1
    closeparen = string.find(parm, ")", strptr, true)
    if (not closeparen) then
        process_error("o parêntese de fechamento está ausente da expressão de COND na expressão: " .. parm)
    end
    colvalue = string.sub(parm, strptr, closeparen-1)

    --Debug - Imprimir resultados da varredura:
    --print ("Found colname=" .. colname .. " in parm: " .. parm)
    --print ("Found colvalue=" .. colvalue .. " in parm: " .. parm)

    -- Se a condição COND(column-name=value) não foi correspondida para essa linha,
    -- não execute mais processamentos nesse parâmetro. Retornar "" para indicar nenhuma ocorrência.
    if (source.column.getvalue(colname) ~= colvalue) then
        return ""
    end

    -- Obtida uma correspondência em COND, portanto, obtenha o valor DATASOURCE

```

```

strptr = closeparen + 1
datasourceValuePtr = string.find(parm, "DATASOURCE(", strptr, true)
if (not datasourceValuePtr) then
  process_error("DATASOURCE clause not found in expression: " .. parm)
end
-- (Observe: Essa varredura é muito simples, ela assume que não há
-- texto errôneo entre as cláusulas COND e DATASOURCE)
closeparen = string.find(parm, ")", datasourceValuePtr, true)
if (not closeparen) then
  process_error("o parêntese de fechamento está ausente da expressão de COND na expressão: " .. 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

```

Exemplo de Procedimento de Mapa de Coluna: Máscara de ID Nacional

O exemplo de procedimento de mapa da coluna ilustra como a função `mask_parms` mascara uma coluna que contém um ID nacional.

```

-----
--
-- Procedimento de mapa de coluna de amostra do IBM Optim
--
-- Nome:          OptimNID
--
-- Revisão:      1.0
--
-- Descrição:    Mascara uma coluna de ID nacional com base no valor de outra
--              coluna ou um argumento de script com o código do país.
--
-- Entrada:      Parâmetro 1 (Obrigatório):
--              Um código do país de sequência (por exemplo, US para Estados Unidos)
--              --OR--
--              COL(column-name) em que column-name é o nome de uma coluna
--              que contém o código do país
--
--              Parâmetro 2 (Opcional):
--              Uma sequência contendo parâmetros adicionais a serem copiados para a
--              chamada optimmask
--              Isso é adicional ao campo mask_parms_constant que é
--              declarado no início desse script.
--
--
-- Função de saída: - Os dados da coluna mascarada, conforme configurado pela função
target.column.setvalue
--                  - Texto direcionado ao relatório Optim pela função de exibição
--
-- Códigos de Retorno: 0 - execução bem-sucedida
--                    1 - Rejeitar linha (Use em cm_transform; a linha não será inserida na
--                      tabela de destino)
--                    2 - Interromper processo
--
--              Duas funções auxiliares são especificadas para definir condições diferentes do que o
--              código de retorno 0.
--              Não há necessidade de codificar uma instrução de retorno ao usar essas
--              funções.
--              error(string) - Essa chamada fará com que o processo do Optim seja interrompido
--                                e a sequência seja mostrada no relatório do Optim
--                                como uma mensagem de erro.
--

```

```

--          rejectrow() - Essa chamada fará com que o Optim rejeite a linha
--                        que está sendo processada atualmente. A linha não será
--                        inserida na tabela de destino.
--
--
-----

function cm_transform()

  -- Altere esse campo para conter os parâmetros que devem
  -- ser colocados em todas as chamadas de optimmask
  local mask_parms_constant = 'MTD=REPEATABLE'

  --
  -- Obter parâmetros:
  --
  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

  --
  -- Obter código do país no campo swivalue com base nos parâmetros
  --
  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

  --
  -- Construa a chamada para optimmask, faça a chamada
  -- e coloque o novo valor na coluna de destino
  --
  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

```

Capítulo 6. Gerenciando Dados

Use os utilitários do Optim Designer para editar, procurar, e comparar dados. Também é possível criar tabelas.

Procurando por Dados

Use o utilitário Procurar para rever o conteúdo de um armazenamento de dados do arquivo sem precisar restaurá-lo para o banco de dados.

Utilize o recurso da ajuda no utilitário Procurar para obter mais informações sobre um tópico ou função específica. É possível abrir uma janela de ajuda ao selecionar **Ajuda > Conteúdo**, clicar com o botão direito do mouse em um item e selecionar **O Quê é Isto** ou pressionar F1.

Para procurar por dados:

1. No Optim Designer, clique em **Ferramentas > Procurar**. A janela Procurar é aberta.
2. Clique em **Arquivo > Abrir** para abrir a janela Abrir e selecione um arquivo de dados criado por um serviço. Os arquivos são armazenados no diretório de dados especificado no utilitário de Configuração.
3. Clique em **Abrir**. As tabelas no arquivo de dados são listadas na janela Procurar.
4. Clique em **Ferramentas > Criar Todos os Objetos Selecionados**. A tabela é aberta na janela Procurar Dados.

Editando Dados

Utilize o Editor de Tabelas para procurar por e editar conjuntos de dados relacionais intactos nas tabelas de banco de dados. O editor manipula um modelo de dados arbitrariamente complexo que consiste de qualquer número de tabelas e relacionamentos e assegura um conjunto de dados referencialmente intacto.

Utilize o recurso da ajuda no Editor de Tabela para obter mais informações sobre um tópico ou função específica. É possível abrir uma janela de ajuda ao selecionar **Ajuda > Conteúdo**, clicar com o botão direito do mouse em um item e selecionar **O Quê é Isto** ou pressionar F1.

Para editar dados:

1. No Optim Designer, clique em **Ferramentas > Editar**. A janela Opções de Edição é aberta.
2. Insira um **Nome de Tabela** ou uma **Definição de Acesso Nomeada**, selecione as opções **Exibição Inicial** ou **Modo** e clique em **OK**. A janela Editor de Tabelas é aberta.
3. Edite uma ou mais tabelas ao mesmo tempo.

Confirme as mudanças no banco de dados ao mover o ponteiro para uma linha diferente ou utilizando as opções de menu. Cada instância de uma confirmação é contada como um nível de desfazer. Um nível de desfazer é definido como uma mudança em uma linha que é confirmada no banco de dados.

Se uma condição de erro ocorrer ao tentar confirmar dados no banco de dados, os dados não serão confirmados, mas a tentativa ainda será contada como um nível de desfazer. O editor permite restaurar os dados que forem modificados em um ponto de confirmação específico.

Um conjunto de busca é o conjunto de linhas que o editor lê a partir de uma única tabela no banco de dados.

- Para desfazer as mudanças e restaurar as linhas para as versões originais obtidas na busca atual configurada para todas as tabelas, selecione **Ferramentas > Desfazer**.

- Para desfazer uma mudança em uma linha, clique com o botão direito do mouse na linha e clique em **Desfazer**.
- Para exibir cada versão sucessiva da linha confirmada no conjunto de buscas atual e a versão original da linha no conjunto de buscas, clique com o botão direito do mouse na linha e clique em **Desfazer...**

Comparando Dados

Use o utilitário de Comparação para comparar dados de um conjunto de tabelas de origem com dados de outro conjunto. É possível definir uma solicitação de comparação que está armazenada no repositório do Optim e salvar os resultados de uma comparação em um arquivo de comparação.

Definindo uma Solicitação de Comparação

Utilize o Editor de Solicitação de Comparação para definir origens de dados para comparar e as opções de processamento.

Utilize o recurso da ajuda no utilitário Comparação para obter mais informações sobre um tópico ou função específica. É possível abrir uma janela de ajuda ao selecionar **Ajuda > Conteúdo**, clicar com o botão direito do mouse em um item e selecionar **O Quê é Isto** ou pressionar F1.

Para definir uma solicitação de comparação:

1. No Explorer de Repositório, clique com o botão direito do mouse em **Comparação** e clique em **Nova Comparação**. O Editor de Solicitação de Comparação é aberto.
2. Na guia **Geral**, conclua as seguintes etapas:
 - a. Insira um nome de arquivo de comparação no campo **Comparar Arquivo**.
 - b. Selecione **Procurar Resultados Imediatamente** para visualizar os resultados do processo de comparação depois que o processo for concluído.
 - c. Selecione um modo de **Comparação**.

Tabela Única

Arquivo de Origem - Arquivo de Origem

Compare os dados em uma tabela única em um arquivo de origem com dados em outro arquivo de origem.

Arquivo de Origem - Tabela de Banco de Dados

Compare os dados em uma única tabela em um arquivo de origem com dados em uma única tabela no banco de dados.

Tabela de Banco de Dados - Tabela de Banco de Dados

Compare os dados em uma única tabela em um banco de dados com dados em uma única tabela em outro banco de dados.

Várias Tabelas

Arquivo de Origem - Arquivo de Origem

Compare os dados em tabelas em um arquivo de origem com dados em outro arquivo de origem.

Arquivo de Origem - Definição de Acesso

Compare os dados em tabelas em um arquivo de origem com os dados especificados em uma definição de acesso.

Arquivo de Origem - Todas as Tabelas de Banco de Dados

Compare os dados em tabelas em um arquivo de origem com dados em um banco de dados.

Nota: Esta seleção é extremamente útil ao testar um aplicativo de banco de dados. O arquivo de origem representa a imagem "antes" em comparação com as tabelas de banco de dados, que representam a imagem "depois".

Definição de Acesso - Definição de Acesso

Compare os dados especificados em uma definição de acesso com dados especificados em outra definição de acesso.

Definição de Acesso - Todas as Tabelas de Banco de Dados

Compare os dados especificados em uma definição de acesso com dados em um banco de dados.

3. Na guia **Origem**, selecione as origens de dados para comparação com base no modo de comparação.
4. Clique em **Arquivo > Executar** para iniciar o processo de comparação.
5. Clique em **Arquivo > Salvar** para salvar a solicitação de comparação. Insira um nome de solicitação no formato IDENTIFIER.NAME.

O processo criará um arquivo de comparação no diretório de dados especificado no utilitário de Configuração. Utilize o utilitário Procurar para visualizar os resultados da comparação.

Editando uma Solicitação de Comparação

Use o Editor de Solicitação de Comparação para editar uma solicitação de comparação.

Utilize o recurso da ajuda no utilitário Comparação para obter mais informações sobre um tópico ou função específica. É possível abrir uma janela de ajuda ao selecionar **Ajuda > Conteúdo**, clicar com o botão direito do mouse em um item e selecionar **O Quê é Isto** ou pressionar F1.

Para editar uma solicitação de comparação:

1. No Explorer de Repositório, expanda o nó **Comparação**, clique com o botão direito do mouse na solicitação a ser editada e clique em **Editar**. O Editor de Solicitação de Comparação é aberto.
2. Edite a solicitação.
3. Clique em **Arquivo > Salvar** para salvar a solicitação de comparação.

Executando um Processo de Comparação

É possível executar um processo de comparação para gerar um arquivo de comparação que pode ser visualizado com o utilitário Procurar.

Para executar uma solicitação de comparação:

Execute a solicitação usando um dos seguintes métodos:

Para executar o processo de comparação a partir do Explorer de Repositório, clique com o botão direito do mouse em uma solicitação de comparação e clique em **Executar**.

Para executar o processo de comparação a partir do Editor de Solicitação de Comparação, clique em **Arquivo > Executar**.

O processo criará um arquivo de comparação no diretório de dados especificado no utilitário de Configuração. Utilize o utilitário Procurar para visualizar os resultados da comparação.

Criando Tabelas

Use o utilitário Criar para criar objetos com base nas definições em um arquivo de dados de origem.

Utilize o recurso da ajuda no utilitário Criar para obter mais informações sobre um tópico ou função específica. É possível abrir uma janela de ajuda ao selecionar **Ajuda > Conteúdo**, clicar com o botão direito do mouse em um item e selecionar **O Quê é Isto** ou pressionar F1.

Para criar tabelas:

1. No Optim Designer, clique em **Ferramentas > Criar**. A janela Opções de Criação é aberta.
2. Selecione um arquivo de dados de origem criado por um serviço. Os arquivos são armazenados no diretório de dados especificado no utilitário de Configuração.
3. Selecione uma das **Opções de Mapa de Tabela**.
Se você escolher um mapa de tabela local, o Editor de Mapas de Tabela - Local é aberto. Execute as etapas a seguir:
 - a. No campo **Qualificador**, você deve inserir um alias de armazenamento de dados de destino e o ID do criador no formato `data_store_alias.creator_id`.
 - b. Modifique os nomes da tabela de destino.
 - c. Clique em **Arquivo > Atualizar e Retornar**.Se você selecionar um mapa de tabela nomeado, conclua a seguinte etapa:
 - a. No campo **Nome do Mapa de Tabela**, insira ou selecione um nome de mapa de tabela.
4. Clique em **OK**. A janela Criar é aberta.
5. Editar e selecione os objetos a serem criados.
6. Clique em **Ferramentas > Criar Todos os Objetos Selecionados** para iniciar o processo de criação.

Capítulo 7. Gerenciando Dados com o Optim Designer

O Optim Designer fornece uma única interface de design para soluções de gerenciamento de dados de teste e privacidade de dados do Optim.

Extraindo Dados com o Optim Designer

Este tutorial ensina como usar o Optim Designer para criar um serviço de extração do Optim. Neste tutorial, você especificará uma definição de acesso e um serviço de extração.

Depois de concluir este tutorial, será possível executar um serviço de extração.

Objetivo do aprendizado

Ao concluir os exercícios, você saberá como executar as seguintes tarefas:

- Criar uma pasta do Explorer de Repositório.
- Criar uma definição de acesso para definir os dados a serem extraídos.
- Editar uma definição de acesso para definir critérios de eleição.
- Criar um serviço de extração.
- Testar um serviço de extração.

Tempo necessário

Este módulo leva aproximadamente 60 minutos para ser concluído.

Pré-requisitos

Este tutorial requer um ambiente do Optim Designer que inclui uma conexão com um repositório do Optim e um alias de armazenamento de dados que contém os dados de amostra do Optim.

Este tutorial pode ser concluído no ambiente do Optim Designer.

Criando uma Pasta do Explorer de Repositório

Neste exercício, você criará uma pasta do Explorer de Repositório. Uma pasta do Explorer de Repositório contém serviços, definições de acesso, mapas de coluna e mapas de tabela. Use a pasta para organizar esses objetos.

Para criar um plano de acesso a dados e uma política de seleção:

1. Clique com o botão direito do mouse na visualização Explorer de Repositório e clique em **Novo > Pasta**. A janela Nova Pasta é aberta.
2. Em **Nome**, digite Tutorial e clique em **OK**.

Você criou uma pasta do Explorer do Repositório denominada Tutorial.

Criando uma Definição de Acesso

Neste exercício, você criará uma definição de acesso. Use as definições de acesso para especificar as tabelas, passagem de relacionamento e critérios de seleção para os dados que deseja processar.

Antes de criar uma definição de acesso, um alias de armazenamento de dados deve existir para o banco de dados que contém a tabela de início.

Para criar uma definição de acesso:

1. Expanda a pasta do Tutorial no Explorer de Repositório e clique com o botão direito do mouse em **Definições de Acesso** e clique em **Nova Definição de Acesso**. O assistente Nova Definição de Acesso se abre.
2. Na página Inserir Nome da Definição de Acesso, insira `SAMPLE.AD`. Clique em **Avançar**.
3. Na página Selecionar um Alias de Armazenamento de Dados, selecione o alias de armazenamento de dados que contém os dados de amostra `Optim`. Clique em **Avançar**.
4. Na página Selecionar uma Tabela de Início, execute as seguintes etapas:
 - a. No campo **Tabela de Início**, insira o seguinte padrão de procura: `SCHEMA.OPTIM_CUSTOMERS`, em que `SCHEMA` é o esquema que contém a amostra de dados do `Optim`. Por exemplo:
`SAMPLE.OPTIM_CUSTOMERS`.
 - b. Clique em **Localizar Tabelas**. A lista de tabelas exibe as tabelas que correspondem ao padrão.
 - c. Selecione a tabela `OPTIM_CUSTOMERS`.
 - d. Clique em **Avançar**.
5. Na página Especificar Método de Seleção de Tabela, clique em **Localizar tabelas relacionadas**. Clique em **Avançar**.
6. Na página Inserir Padrão de Tabela e Selecionar Tabelas Relacionadas, execute as seguintes etapas:
 - a. Clique em **Localizar Tabelas**. A lista de tabelas exibe as tabelas relacionadas a `OPTIM_CUSTOMERS`.
 - b. Clique em **Selecionar Todos**.
 - c. Clique em **Concluir**.

A nova definição de acesso é exibida no Editor de Definição de Acesso.

Definindo Critérios de Seleção

Neste exercício, você definirá os critérios de seleção na definição de acesso. Use os critérios de seleção para focalizar em um conjunto específico de dados relacionados, definindo uma cláusula SQL `WHERE` e usando as variáveis de substituição com os valores padrão.

Critérios de seleção devem estar em conformidade com a sintaxe SQL e incluir operadores relacionais ou lógicos. Os operadores lógicos e a sintaxe variam entre DBMSs. Consulte a documentação de DBMS apropriada para obter informações.

Para selecionar o conjunto desejado de dados para uma tabela, você pode precisar de uma combinação de operadores lógicos `AND` e `OR`.

Para definir os critérios de seleção:

1. Expanda uma pasta Tutorial no Explorer de Repositório, expanda o nó **Definições de Acesso** e clique duas vezes na definição de acesso `SAMPLE.AD`. O Editor de Definição de Acesso se abre.
2. Selecione a guia **Tabelas**.
3. Selecione a tabela `OPTIM_CUSTOMERS` na lista de tabelas.
4. Clique em **Editar Critérios de Seleção**. A janela Editar Critérios de Seleção se abre e exibe uma cláusula `WHERE` para a tabela.
5. Insira os seguintes critérios: `COUNTRY_CODE='US'`.
Clique em **Verificar Sintaxe** para verificar a sintaxe e identificar os erros.
6. Clique em **OK** para retornar ao Editor de Definição de Acesso.
Se a sintaxe não estiver válida, um prompt será aberto para identificar o erro. Não é possível salvar os critérios, se contiver erros.
7. No menu principal, clique em **Arquivo > Salvar** para salvar a definição de acesso.

Você definiu critérios de seleção que selecionarão apenas linhas da tabela OPTIM_CUSTOMERS na qual o valor da coluna COUNTRY_CODE é 'US'.

Criando um Serviço de Extração

Neste exercício, você criará um serviço de extração com base nos dados especificados em uma definição de acesso. Use um serviço de extração para copiar um conjunto de linhas relacionadas de uma ou mais tabelas e salvar as linhas para um armazenamento de dados do arquivo.

Para criar um serviço de extração:

1. Expanda a pasta do Tutorial no Explorer de Repositório, clique com o botão direito do mouse em **Serviços** e clique em **Novo Serviço**. O assistente de serviço novo se abre.
2. Na página Inserir um Nome de Serviço e Selecionar um Serviço, execute as seguintes etapas:
 - a. No campo **Nome**, insira SAMPLE.EXTRACT.
 - b. Na lista de tipos de serviço, selecione **Extração**.
 - c. Clique em **Avançar**.
3. Na página Selecionar uma Definição de Acesso, selecione SAMPLE.AD. Clique em **Avançar**.
4. Na página Inserir um Nome de Armazenamento de Dados do Arquivo de Destino, insira SAMPLE_EXT no campo **Nome do armazenamento de dados do arquivo de destino**.
5. Clique em **Concluir**.

O novo serviço de extração exibido no Editor de Serviço de Extração. É possível utilizar este serviço para criar um armazenamento de dados do arquivo para um serviço de conversão, inserção ou de carregamento. Também é possível utilizar o armazenamento de dados do arquivo para criar um mapa da coluna ou mapa da tabela.

Mascarando Dados com o Optim Designer

Este tutorial ensina como usar o Optim Designer para criar um serviço de conversão que transforma dados em um armazenamento de dados do arquivo.

Este tutorial requer o armazenamento de dados do arquivo SAMPLE_EXT criado no tutorial “Extraindo Dados com o Optim Designer” na página 123.

Depois de concluir este tutorial, será possível aplicar uma função de privacidade e executar um serviço de conversão.

Objetivo do aprendizado

Ao concluir os exercícios, você saberá como executar as seguintes tarefas:

- Criar uma pasta do Explorer de Repositório.
- Criar uma definição de acesso para definir os dados a serem extraídos.
- Editar uma definição de acesso para definir critérios de eleição.
- Criar um serviço de extração.
- Testar um serviço de extração.

Tempo necessário

Este módulo leva aproximadamente 60 minutos para ser concluído.

Pré-requisitos

Este tutorial requer um ambiente do Optim Designer que inclui uma conexão com um repositório do Optim e um alias de armazenamento de dados que contém os dados de amostra do Optim.

Este tutorial pode ser concluído no ambiente do Optim Designer.

Criando um Mapa da Tabela

Neste exercício, você criará um mapa da tabela. Use um mapa da tabela para definir as especificações para correlacionar as tabelas de origem e destino de dados compatíveis. É possível mapear as tabelas que possuem diferentes nomes, modificar os nomes de tabela, excluir as tabelas de um processo ou incluir os mapas da coluna para maior controle sobre os dados.

Para criar um mapa da tabela:

1. Expanda a pasta do Tutorial no Explorer de Repositório, clique com o botão direito do mouse em **Mapas da Tabela** e clique em **Novo Mapa da Tabela**. O assistente Novo Mapa da Tabela se abre.
2. Na página Inserir Nome de Mapa da Tabela, insira `SAMPLE.TMAP` no campo **Name**. Clique em **Avançar**.
3. Na página Selecionar um Alias de Armazenamento de Dados, selecione `SAMPLE.EXT`.
4. Clique em **Concluir**.

O novo mapa da tabela é exibido no Editor de Mapa da Tabela.

Você deve usar o editor para definir os armazenamentos de dados de destino e os esquemas antes de salvar o mapa da tabela.

Editando um Mapa da Tabela

Nesse exercício, você usará o Editor de Mapa da Tabela para definir os armazenamentos de dados de destino e esquemas para tabelas de origem em um mapa da tabela.

Para editar os dados de destino em um mapa da tabela:

1. Expanda uma pasta Tutorial no Explorer de Repositório, expanda o nó **Mapas da Tabela** e clique duas vezes no mapa da tabela `SAMPLE.TMAP`. O Editor de Mapa da Tabela se abre.
2. Selecione a guia **Mapa da Tabela**.
3. Na área **Mapa do esquema e alias de armazenamento de dados**, selecione a linha que contém o armazenamento de dados de origem.
4. Clique na célula **Alias de Armazenamento de Dados de Destino** e selecione o alias de armazenamento de dados que contém os dados de amostra do Optim.
5. Clique na célula **Esquema de Destino** e selecione o esquema que contém os dados de amostra do Optim.
6. No menu principal, clique em **Arquivo > Salvar** para salvar o mapa da tabela.

O mapa da tabela usa as mesmas tabelas de origem e destino para mascarar dados de origem e manter o mesmo esquema e alias de armazenamento de dados.

Criando um Mapa da Coluna

Neste exercício, você criará um mapa da coluna. Um mapa da coluna fornece especificações necessárias para corresponder ou excluir as colunas do processamento em um serviço de gerenciamento de dados. Os serviços de conversão, inserção e carregamento devem referenciar um mapa da tabela, que podem referenciar um ou mais mapas de coluna. É possível usar um mapa da coluna para definir as transformações de dados com funções ou procedimentos de mapa da coluna.

Para criar um mapa da coluna:

1. Expanda a pasta do Tutorial no Explorer de Repositório e clique com o botão direito do mouse em **Mapas da Coluna** e clique em **Novo Mapa da Coluna**. O assistente Novo Mapa da Coluna se abre.
2. Na página Inserir Nome de Mapa da Coluna, insira `SAMPLE.CMAP` no campo Nome. Clique em **Avançar**.
3. Na página Selecionar um Alias de Armazenamento de Dados do Arquivo de Origem, selecione `SAMPLE_EXT` e clique em **Avançar**.
4. Na página Selecionar uma Tabela de Origem, execute as seguintes etapas:
 - a. No campo **Padrão de Procura de Tabela**, insira o seguinte padrão de procura: `DATA_STORE_ALIAS.SCHEMA.OPTIM_CUSTOMERS`, em que `DATA_STORE_ALIAS.SCHEMA` são o alia e o esquema de armazenamento de dados que contém a amostra de dados do Optim. Por exemplo: `OPTIM.SAMPLE.OPTIM_CUSTOMERS`.
 - b. Clique em **Localizar Tabelas**. A lista de tabelas exibe as tabelas que correspondem ao padrão.
 - c. Selecione a tabela `OPTIM_CUSTOMERS`.
 - d. Clique em **Avançar**.
5. Na página Selecionar um Alias de Armazenamento de Dados de Destino, selecione o alias de armazenamento de dados que contém os dados de amostra Optim. Clique em **Avançar**.
6. Na página Selecionar uma Tabela de Destino, execute as seguintes etapas:
 - a. No campo **Padrão de Procura de Tabela**, insira o seguinte padrão de procura: `SCHEMA.OPTIM_CUSTOMERS`, em que `SCHEMA` é o esquema que contém a amostra de dados do Optim. Por exemplo: `SAMPLE.OPTIM_CUSTOMERS`.
 - b. Clique em **Localizar Tabelas**. A lista de tabelas exibe as tabelas que correspondem ao padrão.
 - c. Selecione a tabela `OPTIM_CUSTOMERS`.
7. Clique em **Concluir**.

O novo mapa da coluna é exibido no Editor de Mapa da Coluna.

Aplicando uma Função de Mascaramento de Dados

Neste exercício, você aplicará uma função de mascaramento de dados em uma coluna de um mapa da coluna. É possível mascarar dados como números de ID nacional, números de cartão de crédito, datas, valores numéricos e informações pessoais.

Para aplicar uma função de mascaramento de dados.

1. Expanda uma pasta Tutorial no Explorer de Repositório, expanda o nó **Mapas da Coluna** e clique duas vezes no mapa da coluna `SAMPLE.CMAP`. O Editor de Mapa da Coluna se abre.
2. Selecione a coluna `PHONE_NUMBER`.
3. Clique em **Aplicar Função**. A janela Aplicar Função se abre.
4. Selecione a função **Ordenar Aleatoriamente**. Clique em **OK**. O nome da função se abre na coluna **Coluna de Origem** e o editor de função se abre no Editor de Mapa da Coluna.
5. No campo **Expressão do mapa da coluna** do editor de função, insira `SHUFFLE(RETRY=12)`.
6. No menu principal, clique em **Arquivo > Salvar** para salvar o mapa da coluna.

Você aplicou a Função Ordenar Aleatoriamente na coluna `PHONE_NUMBER`. A função irá mascarar dados ao substituir os valores na coluna por outros valores na coluna. A função procurará até doze vezes por um valor de substituição que não corresponder ao valor de origem.

Incluindo um Mapa da Coluna em um Mapa da Tabela

Neste exercício, você incluirá um mapa da coluna em um mapa da tabela. Um mapa da tabela é necessário para um serviço de conversão. Utilize o mapa da coluna associado para ativar o serviço de conversão para desempenhar a função de mascaramento de dados definida no mapa da coluna.

Para incluir um mapa da coluna em um mapa da tabela:

1. Expanda uma pasta Tutorial no Explorer de Repositório, expanda o nó **Mapas da Tabela** e clique duas vezes no mapa da tabela SAMPLE.TMAP. O Editor de Mapa da Tabela se abre.
2. Selecione a guia **Mapa da Tabela**.
3. Na área **Mapa da Tabela**, selecione a tabela OPTIM_CUSTOMERS.
4. Clique em **Incluir Mapa da Coluna**. A janela Novo Mapa da Coluna é aberta com uma lista de mapas da coluna que contém as tabelas selecionadas.
5. Selecione o mapa da coluna SAMPLE.CMAP. Clique em **OK**.
6. No menu principal, clique em **Arquivo > Salvar** para salvar o mapa da tabela.

Você incluiu o mapa da coluna SAMPLE.CMAP (e sua função de mascaramento de dados associada) no mapa da tabela SAMPLE.TMAP, que permitirá que um serviço de conversão transforme dados.

Criando um Serviço de Conversão

Neste exercício, você criará um serviço de conversão para mascarar dados.

Para criar um serviço de conversão:

1. Expanda a pasta do Tutorial no Explorer de Repositório, clique com o botão direito do mouse em **Serviços** e clique em **Novo Serviço**. O assistente de serviço novo se abre.
2. Na página Inserir um Nome de Serviço e Selecionar um Serviço, execute as seguintes etapas:
 - a. No campo **Nome**, insira SAMPLE.CONVERT.
 - b. Na lista de tipos de serviço, selecione **Converter**.
 - c. Clique em **Avançar**.
3. Na página Selecionar um Mapa da Tabela, selecione o mapa da tabela SAMPLE.TMAP. Clique em **Avançar**.
4. Na página Inserir Propriedades de Destino, insira SAMPLE_CONV no campo **Nome do armazenamento de dados do arquivo de destino**.
5. Clique em **Concluir**.

O novo serviço de conversão exibido no Editor de Serviço de Conversão. É possível utilizar este serviço para mascarar a coluna PHONE_NUMBER na tabela OPTIM_CUSTOMERS.

Avisos

Estas informações foram desenvolvidas para produtos e serviços oferecidos nos Estados Unidos.

É possível que a IBM não ofereça os produtos, serviços ou recursos discutidos nesta publicação em outros países. Consulte um representante IBM local para obter informações sobre produtos e serviços disponíveis atualmente em sua área. Qualquer referência a produtos, programas ou serviços IBM não significa que apenas produtos, programas ou serviços IBM possam ser utilizados. Qualquer produto, programa ou serviço funcionalmente equivalente, que não infrinja nenhum direito de propriedade intelectual da IBM poderá ser utilizado em substituição a este produto, programa ou serviço. Entretanto, a avaliação e verificação da operação de qualquer produto, programa ou serviço não IBM são de responsabilidade do Cliente.

A IBM pode ter patentes ou solicitações de patentes pendentes relativas a assuntos tratados nesta publicação. O fornecimento desta publicação não lhe garante direito algum sobre tais patentes. Pedidos de licença devem ser enviados, por escrito, para:

Gerência de Relações Comerciais e Industriais da IBM Brasil
Av. Pasteur, 138-146
Botafogo
Rio de Janeiro, RJ
CEP 22290-240

Para pedidos de licença relacionados a informações de DBCS (Conjunto de Caracteres de Byte Duplo), entre em contato com o Departamento de Propriedade Intelectual da IBM em seu país ou envie pedidos de licença, por escrito, para:

IBM World Trade Asia Corporation
Licensing 2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

O parágrafo a seguir não se aplica a nenhum país em que tais disposições não estejam de acordo com a legislação local: A INTERNATIONAL BUSINESS MACHINES CORPORATION FORNECE ESTA PUBLICAÇÃO “NO ESTADO EM QUE SE ENCONTRA”, SEM GARANTIA DE NENHUM TIPO, SEJA EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS A ELAS NÃO SE LIMITANDO, AS GARANTIAS IMPLÍCITAS DE NÃO INFRAÇÃO, COMERCIALIZAÇÃO OU ADEQUAÇÃO A UM DETERMINADO PROPÓSITO. Alguns países não permitem a exclusão de garantias expressas ou implícitas em certas transações; portanto, esta disposição pode não se aplicar ao Cliente.

Essas informações podem conter imprecisões técnicas ou erros tipográficos. São feitas alterações periódicas nas informações aqui contidas; tais alterações serão incorporadas em futuras edições desta publicação. A IBM pode, a qualquer momento, aperfeiçoar e/ou alterar os produtos e/ou programas descritos nesta publicação a qualquer momento sem aviso prévio.

Referências nestas informações a Web sites que não sejam da IBM são fornecidas apenas por conveniência e não representam de forma alguma um endosso a estes Web sites. Os materiais contidos nesses Web sites não fazem parte dos materiais desse produto IBM e a utilização desses Web sites é de inteira responsabilidade do Cliente.

A IBM pode utilizar ou distribuir as informações fornecidas da forma que julgar apropriada sem incorrer em qualquer obrigação para com o Cliente.

Os licenciados deste programa, que desejam obter informações sobre o mesmo com o objetivo de permitir: (i) a troca de informações entre programas criados independentemente e outros programas (incluindo este) e (ii) a utilização mútua das informações trocadas, devem entrar em contato com:

Gerência de Relações Comerciais e Industriais da IBM Brasil
Av. Pasteur, 138-146
Botafogo
Rio de Janeiro, RJ
CEP 22290-240

Tais informações podem estar disponíveis, sujeitas a termos e condições apropriadas, incluindo em alguns casos, o pagamento de uma taxa.

O programa licenciado descrito nesta publicação e todo o material licenciado disponível são fornecidos pela IBM sob os termos do Contrato com o Cliente IBM, do Contrato Internacional de Licença do Programa IBM ou de qualquer outro contrato equivalente.

Todos os dados de desempenho aqui contidos foram determinados em um ambiente controlado. Portanto, os resultados obtidos em outros ambientes operacionais podem variar significativamente. Algumas medidas podem ter sido tomadas em sistemas em nível de desenvolvimento e não há garantia de que estas medidas serão as mesmas em sistemas disponíveis em geral. Além disso, algumas medidas podem ter sido estimadas por extrapolação. Os resultados reais podem variar. Os usuários deste documento devem verificar os dados aplicáveis para seu ambiente específico.

As informações relativas a produtos não IBM foram obtidas junto aos fornecedores dos respectivos produtos, de seus anúncios publicados ou de outras fontes disponíveis publicamente. A IBM não testou estes produtos e não pode confirmar a precisão de seu desempenho, compatibilidade nem qualquer outra reivindicação relacionada a produtos não IBM. Dúvidas sobre os recursos de produtos não IBM devem ser encaminhadas diretamente a seus fornecedores.

Todas as declarações relacionadas aos objetivos e intenções futuras da IBM estão sujeitas a alterações ou cancelamento sem aviso prévio, e representam apenas metas e objetivos.

Todos os preços IBM mostrados são preços de varejo sugeridos pela IBM, são atuais e estão sujeitos a alteração sem aviso prévio. Os preços do revendedor podem variar.

Estas informações foram projetadas apenas com o propósito de planejamento. As informações aqui contidas estão sujeitas a alterações antes que os produtos descritos estejam disponíveis.

Estas informações contêm exemplos de dados e relatórios utilizados em operações diárias de negócios. Para ilustrá-los da forma mais completa possível, os exemplos podem incluir nomes de indivíduos, empresas, marcas e produtos. Todos estes nomes são fictícios e qualquer semelhança com nomes e endereços utilizados por uma empresa real é mera coincidência.

Se estas informações estiverem sendo exibidas em cópia eletrônica, as fotografias e ilustrações coloridas podem não aparecer.

Marcas Comerciais

Os termos a seguir são marcas registradas da International Business Machines Corporation nos Estados Unidos e/ou em outros países:

IBM
Logotipo IBM
DB2
AIX

Informix
InfoSphere
Optim

Netezza é uma marca registrada da IBM International Group B.V., uma Empresa IBM.

Adobe, Acrobat, PostScript e todas as marcas registradas baseadas em Adobe são marcas registradas ou marcas da Adobe Systems Incorporated nos Estados Unidos e/ou em outros países.

Apache Derby é uma marca registrada da The Apache Software Foundation.

Eclipse é uma marca registrada da Eclipse Foundation, Inc.

Microsoft, Windows, Windows NT e o logotipo Windows são marcas registradas da Microsoft Corporation nos Estados Unidos e/ou em outros países.

Java e todas as marcas registradas baseadas em Java são marcas registradas da Sun Microsystems, Inc. nos Estados Unidos e/ou em outros países.

Linux é uma marca registrada da Linus Torvalds nos Estados Unidos e/ou em outros países.

UNIX é uma marca registrada da The Open Group nos Estados Unidos e em outros países.

Outros nomes de empresas, produtos ou serviços podem ser marcas registradas ou marcas de serviço de terceiros.

Índice Remissivo

A

alias de armazenamento de dados
 assistente Novo Alias do Armazenamento de Dados 15
 assistente Propriedades de Conexão 15
 conectando 16
 definindo 15
 editando 15
 visão geral 15
assistente de Novo Mapa de Tabela 34
assistente Incluir Tabelas 20
assistente Nova Chave Primária 32
assistente Nova Definição de Acesso 20
assistente Novo Alias do Armazenamento de Dados 15
assistente Novo Mapa da Coluna 37
assistente Novo Relacionamento do Optim 30
Assistente Novo serviço 41, 44, 46, 49
assistente Propriedades de Conexão 15
assistente Transformar Solicitação em Serviço do Optim 17

C

chaves primárias Optim
 assistente Nova Chave Primária 32
 como criar 32
 editando 32
 Editor da Chave Primária 32, 33
 explícito 31
 genérico 31, 33
 selecionando colunas-chave 32
 valor de propagação 91
 visão geral 31
codificação
 procedimentos de mapa da coluna 111
comparando dados 120
consulta alternada
 exemplo 113
critérios de seleção
 agrupamento de dados 22
 definindo 22
 gerenciando relacionamentos 23
 janela Editar Critérios de Seleção 22
 lista de apontar e disparar 25, 26
 opções de acesso da tabela 24
 opções de amostragem de dados e limite de linha 23
 passagem de relacionamento 23
 variáveis 24, 25
 visão geral 21

D

dados de amostra
 criando tabelas de amostra do Optim 12

dados de amostra (*continuação*)
 criando tabelas de privacidade de dados 12
 dados de amostra Optim 5
 Tabela OPTIM_CUSTOMERS 6
 Tabela OPTIM_DETAILS 8
 Tabela OPTIM_FEMALE_RATES 11
 Tabela OPTIM_ITEMS 8
 Tabela OPTIM_MALE_RATES 10
 Tabela OPTIM_ORDERS 7
 Tabela OPTIM_SALES 5
 Tabela OPTIM_SHIP_INSTR 10
 Tabela OPTIM_SHIP_TO 9
 Tabela OPTIM_STATE_LOOKUP 11
 tabelas de privacidade de dados 11
 visão geral 4
definições de acesso
 alterando as tabelas para referência ou relacionado 20
 assistente Incluir Tabelas 20
 assistente Nova Definição de Acesso 20
 como criar 20
 critérios de seleção 21
 Editor de Definição de Acesso 20, 21, 22, 23
 etapas de passagem 21
 gerenciando relacionamentos 23
 gerenciando tabelas 20
 incluindo tabelas 20
 janela Editar Critérios de Seleção 22
 janela Etapas de Passagem 21
 lista de apontar e disparar 25, 26
 opções de acesso da tabela 24
 passagem de relacionamento 23
 removendo tabelas 21
 variáveis 24, 25
 visão geral 19
Directory Explorer 3
diretório Optim
 assistente Transformar Solicitação em Serviço do Optim 17
 conectando 17
 janela Associar Diretório 17
 transformando uma solicitação em serviço 17

E

editando dados 119
Editor da Chave Primária 32, 33
Editor de Definição de Acesso 20, 21
 guia Apontar e Disparar 26
 guia Propriedades de Grupo de Dados 22
 guia Relacionamentos 23, 24
 Guia Tabelas 23
Editor de Mapa da Tabela 34
 guia Mapa da Tabela 34
 janela Novo Mapa da Coluna 35
Editor de Mapa de Coluna 38

Editor de Mapa de Coluna (*continuação*)
 editor de script Lua 38
 janela Aplicar Função 38, 53
Editor de Relacionamento 30, 31
 janela Incluir Expressão da Coluna 30
 janela Selecionar uma Coluna Filha 30
 janela Selecionar uma Coluna Pai 30
 editor de script Lua 38
Editor de Serviço de Carregamento 49
 Assistente Alterar Mapa da Tabela 51
 guia Opções de Carregamento 50
 guia Opções de Processamento 49
Editor de Serviço de Conversão 45
 Assistente Alterar Mapa da Tabela 45
 guia Opções de Processamento 45
Editor de Serviço de Extração 42
 Assistente Alterar Definição de Acesso 44
 guia Conversão 42
 guia Dados e Objetos 42
 guia Opções de Compactação de Arquivo 43
 guia Propriedades de Serviço 42
 guia Variáveis 43
Editor de Serviço de Inserção 46
 Assistente Alterar Mapa da Tabela 47
 guia Opções de Processamento 46
Editor de Solicitação de Comparação 120, 121
Editor de Tabelas 119
expressões concatenadas 93
expressões numéricas 93

F

Função Aleatória 89
Função de Consulta 53
Função de Consulta Aleatória 60
Função de Consulta Hash 56
Função de Identidade 90
Função de Moeda 98
 conversão direta 98
 triangulação 98
Função de Nome de Email Formatado 100
Função de Nome de Email Gerado Automaticamente 100
Função de Número Aleatório 100
Função de Número Sequencial 100
Função de Propagação 91
Função de Sequência Oracle 90
Função de Subseqüência 88
Função Idade 95
 expiração de semântica 95
 idade incremental 95
 saída do formato de destino 101
 saída do formato de origem 101
Função Ordenar Aleatoriamente 62, 64
Função Sequencial 89

Função Serial 90
Função TRANS CCN 68
Função TRANS COL 75
Função TRANS EML 70
Função TRANS NID
 CF da Itália 82
 EUA SSN 87
 INSEE da França 80
 NIF da Espanha 84
 SIN do Canadá 79
 U.K. NINO 85
 visão geral 77
Função TRANS SSN 64
Funções de Consulta 53
Funções de Valor e Literal 94

G

gerenciando dados
 comparando dados 120
 criando tabelas 121
 definindo uma solicitação de
 comparação 120
 editando dados 119
 editando uma solicitação de
 comparação 121
 Editor de Solicitação de
 Comparação 120, 121
 Editor de Tabelas 119
 executando uma solicitação de
 comparação 121
 procurando por dados 119
 utilitário Comparar 120
 utilitário Criar 121
 utilitário de Procura 119
 utilitários do Optim Designer 119

J

janela Alternar Repositório 16
janela Aplicar Função 38, 53
janela Associar Diretório 17
janela Editar Conexão de Repositório 16
janela Editar Critérios de Seleção 22
janela Etapas de Passagem 21
janela Executar Serviço 51
janela Incluir uma Variável 25
janela Nova Conexão de Repositório 16
janela Novo Arquivo de Apontar e
 Disparar 25
janela Novo Mapa da Coluna 35

L

lista de apontar e disparar
 como criar 25
 guia Apontar e Disparar 26
 janela Novo Arquivo de Apontar e
 Disparar 25
 selecionando 26
 visão geral 25

M

mapas de coluna
 aplicando a função de mascaramento
 de dados 38, 53
 assistente Novo Mapa da Coluna 37
 como criar 37
 criando um procedimento de mapa da
 coluna 38
 editando 38
 editando o procedimento de mapa da
 coluna 38
 editando parâmetros em um
 procedimento de mapa da
 coluna 39
 Editor de Mapa de Coluna 38
 editor de script Lua 38
 incluindo em mapas da tabela 35
 janela Aplicar Função 38, 53
 mapeando coluna de origem 38
 procedimento de mapa da coluna 38
 regras de compatibilidade de
 dados 36
 rotinas de saída 100
 tipos de dados 36
 visão geral 35
mapas de tabela
 assistente de Novo Mapa de
 Tabela 34
 como criar 34
 dados de destino padrão 34
 editando 34
 editando dados de destino 34
 Editor de Mapa da Tabela 34, 35
 incluindo um mapa da coluna 35
 mapas de coluna 35
 visão geral 33
máscara de ID nacional
 exemplo 116
mascaramento de dados
 aplicando a função de mascaramento
 de dados 38, 53
 Constante Booleana 94
 Constante Numérica 94
 criando um procedimento de mapa da
 coluna 38
 editando o procedimento de mapa da
 coluna 38
 editando parâmetros em um
 procedimento de mapa da
 coluna 39
 expressões concatenadas 93
 expressões numéricas 93
 Função Aleatória 89
 Função de Consulta 53
 Função de Consulta Aleatória 60
 Função de Consulta Hash 56
 Função de Identidade 90
 Função de Moeda 98
 Função de Nome de Email Gerado
 Automaticamente 100
 Função de Número Aleatório 100
 Função de Número Sequencial 100
 Função de Propagação 91
 Função de Sequência Oracle 90
 Função de Subsequência 88
 Função Idade 95
 Função Ordenar Aleatoriamente 62

mascaramento de dados (*continuação*)

 Função Sequencial 89
 Função Serial 90
 Função TRANS CCN 68
 Função TRANS COL 75
 Função TRANS EML 70
 Função TRANS NID 77
 CF da Itália 82
 EUA SSN 87
 INSEE da França 80
 NIF da Espanha 84
 SIN do Canadá 79
 U.K. NINO 85
 Função TRANS SSN 64
 funções 53
 Funções de Consulta 53
 Funções de Valor e Literal 94
 Literal Data/Hora 94
 Literal Hexadecimal 94
 Nome de Email Formatado 100
 NULL 94
 Registro Especial 94
 Sequência Literal 94
 visão geral 53
mascarando dados 53
modelagem de dados
 definições de acesso 19
 visão geral 19
modelo de dados
 definições de acesso 19
 visão geral 19
modo integrado do Optim Manager 4,
 51

O

Optim Designer
 conexões com o banco de dados 15
 Directory Explorer 3
 iniciando 1
 perspectiva do Optim 1
 recursos de acessibilidade 13
 Repository Explorer 2
 suporte ao banco de dados 13
 visão geral 1

P

pasta do Explorer de Repositório 3
perspectiva do Optim
 Directory Explorer 3
 Repository Explorer 2
 visão geral 1
procedimento de mapa da coluna 109
 codificação 111
 como criar 38
 editando 38
 editando parâmetros 39
 exemplo 111, 113, 116
 formato de dados numéricos 111
 funções 109
 limitações 111
 mapas de coluna 38
 nomes reservados 109
 procurando por dados 119
 propagar valor da chave primária 91

R

- relacionamentos Optim
 - assistente Novo Relacionamento do Optim 30
 - como criar 30
 - compatibilidade de dados 28
 - editando 30
 - editando colunas 30
 - Editor de Relacionamento 30
 - estendidos 26
 - explícito 26
 - expressões da coluna 30
 - genérico 26
 - ordem da coluna 31
 - relacionamento genérico 31
 - restrições 27
 - visão geral 26
- repositório Optim
 - alterando a conexão 16
 - conectando 16
 - conexão de diretório Optim 17
 - editando conexão 16
 - janela Alternar Repositório 16
 - janela Editar Conexão de Repositório 16
 - janela Nova Conexão de Repositório 16
 - transformando uma solicitação em serviço 17
 - visão geral 16
- Repository Explorer
 - pasta do Explorer de Repositório 3
 - visão geral 2
- rotinas de saída
 - arquivos de cabeçalho de amostra 101
 - gravando 101
 - requisitos 102
 - saída do formato de destino
 - códigos de retorno 108
 - entrada para Função Idade 101
 - formatos 107
 - função 106
 - função de retorno de chamada 107
 - modos de interrupção 108
 - parâmetros 106
 - processando 107
 - saída do formato de origem
 - códigos de retorno 106
 - entrada para Função Idade 101
 - função 104
 - função de retorno de chamada 105
 - modos de interrupção 105
 - parâmetros 104
 - processando 105
 - saída padrão
 - códigos de retorno 104
 - funções de retorno de chamada 103
 - parâmetros 103
 - processando 103
- Saída Padrão 102
- usando DLLs 102
- visão geral 100

S

- serviço de carregamento
 - alterando mapa da tabela 51
 - arquivos de saída 48
 - Assistente Novo serviço 49
 - como criar 49
 - editando 49
 - Editor de Serviço de Carregamento 49, 50, 51
 - opções do carregador DBMS 50
 - opções do processamento 49
 - visão geral 48
- serviço de conversão
 - alterando mapa da tabela 45
 - Assistente Novo serviço 44
 - como criar 44
 - editando 45
 - Editor de Serviço de Conversão 45
 - opções do processamento 45
 - visão geral 44
- serviço de extração
 - alterando a definição de acesso 44
 - Assistente Novo serviço 41
 - como criar 41
 - editando 42
 - Editor de Serviço de Extração 42, 43, 44
 - opções da variável 43
 - opções de compactação de arquivo 43
 - opções de conversão 42
 - opções de objetos e de agrupamento 42
 - opções do processamento 42
 - visão geral 41
- serviço de inserção
 - alterando mapa da tabela 47
 - Assistente Novo serviço 46
 - como criar 46
 - editando 46
 - Editor de Serviço de Inserção 46, 47
 - opções do processamento 46
 - visão geral 45
- serviços de gerenciamento de dados
 - modo integrado do Optim Manager 51
 - serviço de carregamento 48
 - serviço de conversão 44
 - serviço de extração 41
 - serviço de inserção 45
 - testando 51
 - visão geral 41

T

- testando serviços 51

U

- utilitário Comparar
 - visão geral 120
- utilitário Criar 121
- utilitário de Procura 119
- utilitários do Optim Designer
 - comparando dados 120
 - criando tabelas 121

- utilitários do Optim Designer

- (continuação)*

- definindo uma solicitação de comparação 120
- editando dados 119
- editando uma solicitação de comparação 121
- Editor de Solicitação de Comparação 120, 121
- Editor de Tabelas 119
- executando uma solicitação de comparação 121
- procurando por dados 119
- utilitário Comparar 120
- utilitário Criar 121
- utilitário de Procura 119
- visão geral 119

V

- variáveis

- como criar 25
- janela Incluir uma Variável 25
- visão geral 24



Impresso no Brasil