

IBM

@server

iSeries

データベース概説

バージョン 5 リリース 3





@server

iSeries

データベース概説

バージョン 5 リリース 3

ご注意！

本書および本書で紹介する製品をご使用になる前に、43 ページの『特記事項』に記載されている情報をお読みください。

本書は、IBM OS/400 (プロダクト番号 5722-SS1) のバージョン 5、リリース 3、モディフィケーション 0 に適用されます。また、改訂版で断りがない限り、それ以降のすべてのリリースおよびモディフィケーションに適用されます。このバージョンは、すべての RISC モデルで稼動するとは限りません。また CISC モデルでは稼動しません。

本マニュアルに関するご意見やご感想は、次の URL からお送りください。今後の参考にさせていただきます。

<http://www.ibm.com/jp/manuals/main/mail.html>

なお、日本 IBM 発行のマニュアルはインターネット経由でもご購入いただけます。詳しくは

<http://www.ibm.com/jp/manuals/> の「ご注文について」をご覧ください。

(URL は、変更になる場合があります)

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原 典： iSeries
Database overview
Version 5 Release 3

発 行： 日本アイ・ピー・エム株式会社

担 当： ナショナル・ランゲージ・サポート

第1刷 2005.8

この文書では、平成明朝体™W3、平成明朝体™W7、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、および平成角ゴシック体™W7を使用しています。この(書体*)は、(財)日本規格協会と使用契約を締結し使用しているものです。フォントとして無断複製することは禁止されています。

注* 平成明朝体™W3、平成明朝体™W7、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、平成角ゴシック体™W7

© Copyright International Business Machines Corporation 2004,2005. All rights reserved.

© Copyright IBM Japan 2005

目次

データベース概説	1	ジョブ・ログの表示	20
V5R3 の新機能	1	オブジェクトの SQL の生成	20
トピックの印刷	2	SQL アシストによる SQL ステートメントの作成	20
DB2 UDB for iSeries	2	iSeries システム・デバッガの起動	21
iSeries ナビゲーターを初めて使用する場合	3	iSeries ナビゲーターを使用したオブジェクトの作成 および管理	21
iSeries ナビゲーターの開始	3	検査保留制約の管理	22
iSeries ナビゲーターによるスキーマの作成 (SAMPLELIB)	3	SQL を初めて使用する場合	22
iSeries ナビゲーターに表示されるスキーマのリス トの編集	4	スキーマの作成	23
iSeries ナビゲーターによるテーブルの作成および列 の定義	5	テーブルの作成および使用	23
iSeries ナビゲーターによる視点の作成および使用	12	LABEL ON ステートメントの使用	25
iSeries ナビゲーターによるデータベース・オブジ ェクトの削除	16	テーブルへの情報の挿入	27
iSeries ナビゲーター・データベース・タスク	17	1 つのテーブルからの情報の入手	30
「データベース・プログラミング (Database programming)」トピック中のタスク	17	複数のテーブルからの情報の入手	32
「データベース・パフォーマンスおよび Query 最 適化」トピック中のタスク	17	テーブル中の情報の変更	34
データベース・ナビゲーター・マップを使用したデ ータベースのマッピング	18	テーブルからの情報の削除	36
Run SQL Scripts を使用したデータベースの Query	19	視点の作成および使用	37
Run SQL Scripts の停止	19	SQL 対従来のファイル・アクセス用語	39
		コードに関する特記事項	40
		資料に関するご使用条件	40
		付録. 特記事項.	43
		プログラミング・インターフェース情報	44
		商標	44

データベース概説

DB2 UDB™ for iSeries™ は、DB2® の他の多くのインプリメンテーションと特性を共有しています。ただし、iSeries に移行したばかりの場合は、DB2 UDB が他の IBM® プラットフォームとはどのように異なるか疑問に思われるか、IBM の UDB がその他のリレーショナル・データベース・プラットフォームとどのように比較されるか、および iSeries がデータベースの開発にどのような利点をもたらすかを知りたい場合があります。

以下のリンクは、データベース・プラットフォームとしての iSeries のさまざまな利点を理解するのに役立ちます。どのデータ・アクセス方式がお客様の組織に最も適しているかを評価したり、iSeries でのデータベース・インプリメンテーションの開発と維持のためのおおまかなフレームワークを構築する上で役立てることができます。

メイン・ナビゲーション・バーを使用して他のデータベース情報を調べることもできます。

このコードに関する特記事項は、コーディング例に関連しています。



V5R3 の新機能

この項では、「データベース概説」の V5R3 に加えられた変更内容について要約します。

- V5R3 では、Query for iSeries の資料は更新されなかったが、製品には多数の機能上の更新が加えられた。これらの更新については、「管理」トピックの Query および報告書の作成および実行の項に要約されています。
- 検査保留制約機能が追加された。検査保留制約を使用すると、システムによって検査保留状態にされている制約の表示や変更を行えます。詳しくは、22 ページの『検査保留制約の管理』を参照してください。
- SQL および iSeries ナビゲーターを使用して、区分テーブルのサポートを使用できるようになった。詳しくは、「DB2 マルチ・システム」トピックの区分化表 (Partitioned tables) を参照してください。
- SQL および iSeries ナビゲーターを使用して、シーケンスを作成できるようになった。「SQL 解説書」トピックの CREATE SEQUENCE を参照してください。
- SQL および iSeries ナビゲーターを使用して、マテリアライズ照会テーブルを作成できるようになった。「SQL 解説書」トピックの CREATE TABLE を参照してください。
- SQL 診断域には、アプリケーション・プログラム、SQL 関数、SQL プロシージャ、およびトリガーのデバッグに役立つ情報が含まれている。「組み込み SQL プログラミング」トピックの SQL 診断域の使用 (Using the SQL diagnostics area) を参照してください。

新着情報または変更情報の判別方法

以下の情報を利用して、技術上の変更が加えられた箇所を見分けるのに役立ててください。

-  イメージ。これは、新着情報か変更情報の開始位置を示すマークです。
-  イメージ。これは、新着情報か変更情報の終了位置を示すマークです。

新機能または当リリースでの変更に関するその他の情報を見つけるには、プログラム資料説明書を参照してください。

トピックの印刷

これは、この情報の PDF を表示したり印刷したりする場合に使用します。


本書の PDF 版を表示またはダウンロードするには、データベース概説 (約 575 KB) を選択します。

PDF ファイルの保存

表示または印刷のために PDF をワークステーションに保存するには、以下のようになります。

- ブラウザーで PDF を右マウス・ボタン・クリックする (上部のリンクを右マウス・ボタン・クリック)。
- Internet Explorer を使用している場合は、「対象をファイルに保存...」をクリックする。 Netscape Communicator を使用している場合は、「リンクを名前を付けて保存...」をクリックする。
- PDF を保存したいディレクトリーに進む。
- 「保存」をクリックする。

Adobe Acrobat Reader のダウンロード

これらの PDF を表示または印刷するには、Adobe Acrobat Reader が必要です。このアプリケーションは、Adobe Web サイト (www.adobe.com/products/acrobat/readstep.html)  からダウンロードできます。

DB2 UDB for iSeries

このトピックでは、DB2 UDB for iSeries の概要を説明します。

DB2 UDB for iSeries は、ご使用の iSeries で完全に統合されているリレーショナル・データベース・マネージャーです。iSeries で統合されているため、DB2 UDB for iSeries は非常に使いやすだけでなく、管理も容易です。DB2 UDB for iSeries には、トリガー、ストアド・プロシージャ、動的ビットマップ索引付けなどの豊富な機能も用意されており、これらの機能によって広範なアプリケーション・タイプを処理することができます。これらのアプリケーションは、従来のホスト・ベース・アプリケーションからクライアント/サーバー型のソリューションやビジネス・インテリジェンス・アプリケーションにまで及んでいます。

DB2 UDB for iSeries へのインターフェースとして、DB2 Query Manager and SQL Development Kit for iSeries は、対話型 Query および報告書作成インターフェースが追加され、さらに高水準プログラム言語での SQL アプリケーションを作成する際に役立つプリコンパイラーおよびツールが追加されます。業界標準の構造化照会言語 (SQL) に適合させ、OS/400® に SQL をインプリメントすることにより、iSeries データの定義、処理、照会、および iSeries データへのアクセス制御を行うことができます。これは、iSeries のファイルおよび SQL テーブルの両方で等しく機能します。

DB2 UDB for iSeries のトピックには、DB2 UDB for iSeries の利点を利用してアプリケーションまたはユーザー・インターフェースを介した iSeries データへのアクセスおよび管理を行う方法が説明されています。利用に関する情報、基本概念、参照情報、例については、このトピックを検索してください。

iSeries ナビゲーターを初めて使用する場合

このチュートリアルでは、iSeries ナビゲーターを使用してスキーマ、テーブル、および視点の作成と処理を行う方法について説明します。iSeries ナビゲーター・データベースは、多くの共通管理データベース操作の実行に使用できるグラフィカル・インターフェースです。iSeries ナビゲーター操作の多くは構造化照会言語 (SQL) に基づいていますが、これらを実行するために SQL を完全に理解している必要はありません。

このトピックの例では、iSeries ナビゲーターを使用して共通データベース・タスクを実行します。SQL 解説書トピックの対話式 SQL の使用例で作成されるオブジェクトと同じオブジェクトが作成されます。

詳細については、以下のトピックを参照してください。

- iSeries ナビゲーターの開始
- iSeries ナビゲーターによるスキーマの作成 (SAMPLELIB)
- iSeries ナビゲーターに表示されるスキーマのリストの編集
- iSeries ナビゲーターによるテーブルの作成および使用
- iSeries ナビゲーターによるテーブルの列の定義
- iSeries ナビゲーターによる列定義のコピー
- iSeries ナビゲーターによるテーブルへの情報の挿入
- iSeries ナビゲーターによるテーブルの内容の表示
- iSeries ナビゲーターによるテーブル中の情報の変更
- iSeries ナビゲーターによるテーブルからの情報の削除
- iSeries ナビゲーターによるテーブルのコピーおよび移動
- iSeries ナビゲーターによる視点の作成および使用
- WHERE 文節の作成
- iSeries ナビゲーターによるデータベース・オブジェクトの削除

iSeries ナビゲーターのセットアップについて詳しくは、iSeries ナビゲーターの理解を参照してください。

iSeries ナビゲーターを使用して実行できる他のデータベース・タスクに関する情報は、iSeries ナビゲーター・データベース・タスクを参照してください。

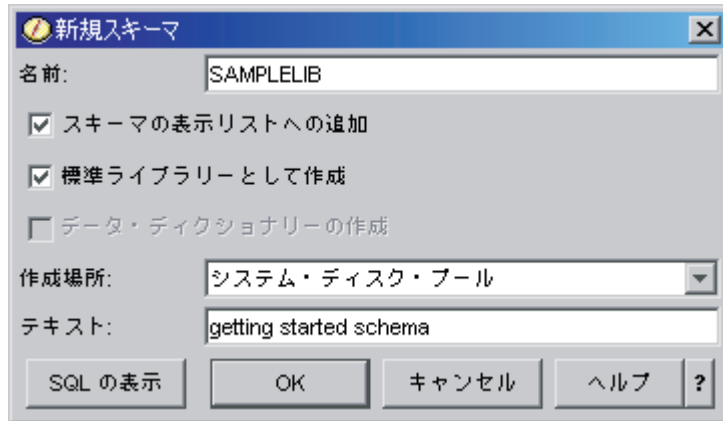
iSeries ナビゲーターの開始

1. 「iSeries ナビゲーター」アイコンをダブルクリックします。
2. 使用したいシステムを展開します。

iSeries ナビゲーターによるスキーマの作成 (SAMPLELIB)

1. 「iSeries ナビゲーター」ウィンドウで、使用したいシステムを展開します。
2. 「データベース」を展開し、処理したいデータベースを展開します。
3. 「スキーマ (Schemas)」を右マウス・ボタン・クリックして、「新規スキーマ」を選択します。
4. 「新規スキーマ」ウィンドウで、名前フィールドに SAMPLELIB と入力します。
5. 表示されるスキーマのリストに追加するには、「スキーマの表示リストへの追加」を選択します。
6. 「標準ライブラリーとして作成」を選択します。

7. スキーマを含めるディスク・プールを指定します。システム・ディスク・プール上にスキーマを作成できるようなディスク・プールを 1 つ選択してください。
8. 説明を入力します (オプション)。
9. 「OK」をクリックします。



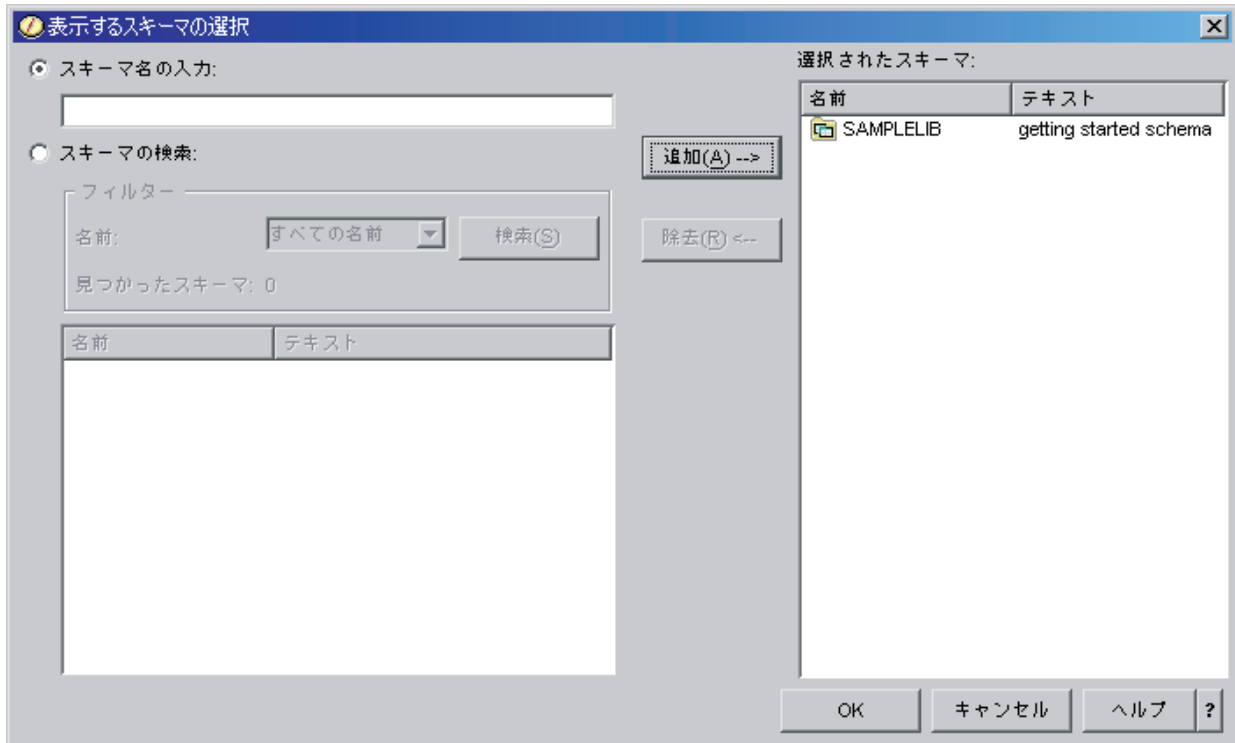
注: ユーザー・ディスク・プール中にスキーマを作成することについての詳細は、複数データベースの処理を参照してください。

iSeries ナビゲーターに表示されるスキーマのリストの編集

スキーマを正常に作成し終わったら、その中にテーブル、視点、索引、ストアド・プロシージャ、ユーザー定義関数、およびユーザー定義タイプを作成できます。

「スキーマ (Schemas)」をクリックすると表示されるスキーマのリストを編集するには、以下のようにします。

1. 「スキーマ (Schemas)」を右マウス・ボタン・クリックして、「表示するスキーマの選択」を選択します。
2. 「表示するスキーマの選択」ウィンドウで、「スキーマ名の入力」を選択してスキーマを指定するか、「スキーマの検索」を選択して検索を実行して、リストを編集できます。表示したいスキーマを選択して、「追加」をクリックします。
3. 表示するスキーマのリストからスキーマを選択し、「除去」をクリックすると、表示するスキーマのリストからそのスキーマを除去できます。



4. ここでは、SAMPLELIB を、表示されるスキーマとして残します。

iSeriesナビゲーターによるテーブルの作成および列の定義

テーブルとは、情報を保管するために使用される基本的なデータベース・オブジェクトです。テーブルを作成し終わったら、「テーブル・プロパティ」ウィンドウを使用して、列を定義したり、索引を作成したり、トリガーや制約を追加したりできます。

テーブルを作成しようとしている場合は、ヌル値とデフォルト値の概念について理解する必要があります。ヌル値とは、ある行の列値が存在しないことを意味します。ヌル値は、ゼロまたはすべてブランクの値とは異なります。「未知」の意味になります。ヌル値は、他のヌル値を含むどの値とも等しくありません。列でヌル値を使用できない場合は、その列に値を割り当てなければなりません。この値は、デフォルト値またはユーザー指定値のいずれかです。

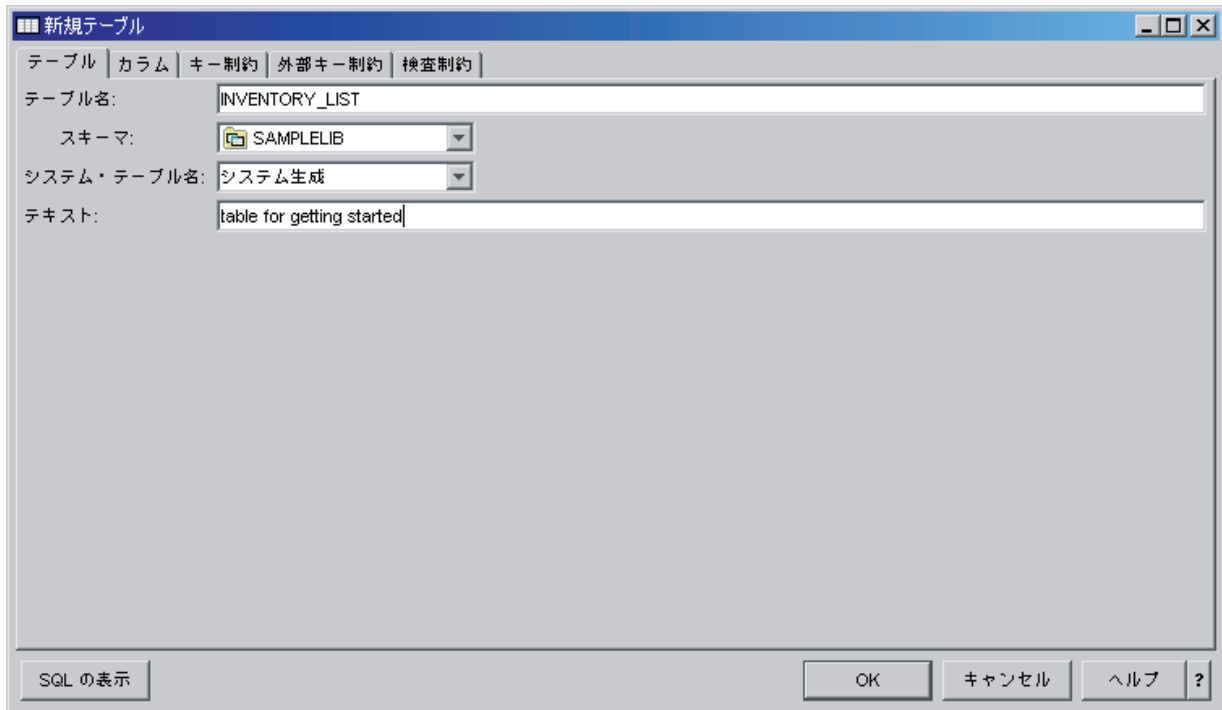
行をテーブルに追加する際に列の値を指定しないと、デフォルト値がその行に割り当てられます。特定のデフォルト値が列に割り当てられていない場合は、その列ではシステム・デフォルト値が使用されます。

この例では、ある業務の現行インベントリーに関する情報を保守するテーブルの作成方法を示します。インベントリー中に保持されている項目、それらの項目のコスト、現在在庫している数量、最新の注文日付、および最新の注文数に関する情報があります。項目番号は必須値です。この値はヌルにできません。項目名、在庫している数量、および注文数量はユーザー指定のデフォルト値になります。最新の注文日付と数量はヌル値にすることができます。

テーブルを作成するには、以下のようにします。

1. 「iSeries ナビゲーター」ウィンドウで、使用したいシステムを展開します。
2. 「データベース」を展開し、処理したいデータベースを展開します。
3. 「スキーマ (Schemas)」を展開します。

4. 「SAMPLELIB」を右マウス・ボタン・クリックして、「新規」を選択します。
5. 「テーブル」 → 「テーブル」を選択します。
6. 「新規テーブル」ウィンドウで、テーブル名として「INVENTORY_LIST」を指定します。
7. 「スキーマ (Schema)」フィールドで、「SAMPLELIB」を選択します。
8. 「システム・テーブル名」フィールドで、「システム生成」を選択します。
9. 「テキスト」フィールドに説明を入力します (オプション)。



10. 次に、新しいテーブルの列を定義します。「カラム」タブをクリックします。
11. 「追加」ボタンをクリックします。
12. 「カラム名」フィールドに、ITEM_NUMBER と入力します。
13. 「ショート・カラム名」フィールドに、短縮名を指定できます。短縮名を指定しないと、システムにより名前が自動的に生成されます。列名が 10 文字以下の場合には、短縮名は列名と同じになります。いずれかの列名を使用して、QUERY を実行できます。ここでは、このスペースをデフォルトの「システム生成」のままにしておきます。
14. データ・タイプとして「CHARACTER」を選択します。
15. この列の長さとして 6 を指定します。サイズが前もって定められているデータ・タイプの場合には、そのサイズが埋められ、その値を変更することはできません。
16. 「エンコード」オプションを、デフォルトの「データ・タイプのデフォルト」のままにしておきます。
17. 「テキスト」フィールドに、列の説明を入力できます。このステップはオプションです。
18. 「見出し」フィールドに、列見出しを入力します。ヘッダーとは、表示中または印刷中に列の先頭に表示されるラベルのことです。60 文字、1 行当たり 20 文字に制限されています。
19. 「ヌル可能」オプションを選択解除します。選択解除すると、行の挿入が正常実行されるには、この列に値が必ず必要になります。
20. 「デフォルト値」フィールドに、0 を入力します。
21. 「OK」をクリックして、テーブルを作成します。

新しいテーブル INVENTORY_LIST が表示されます。

iSeries ナビゲーターによるテーブルの追加列の定義

新規または既存のテーブル上の列を定義できます。今しがた作成したテーブルに列を追加するには、「データベース」→「スキーマ (Schemas)」→「SAMPLELIB」→「テーブル」を展開して、テーブル INVENTORY_LIST にナビゲートします。詳細ペインで、テーブル「INVENTORY_LIST」を右マウス・ボタン・クリックし、「定義」を選択します。

1. 「テーブル定義 (Table Definition)」ウィンドウ上で列を定義するには、「カラム」タブを選択します。
2. 「追加」をクリックします。
3. 以下の列をテーブル INVENTORY_LIST に追加します。

列名	タイプ	長さ	精度	位取り	ヌル可能	デフォルト値
ITEM_NAME	VARCHAR	20			いいえ	UNKNOWN
UNIT_COST	DECIMAL		8	2	いいえ	0
QUANTITY_ON_HAND	SMALLINT				はい	NULL
LAST_ORDER_DATE	DATE				はい	NULL
ORDER_QUANTITY	SMALLINT				はい	20

これらの列を定義し終えたら、「OK」をクリックしてテーブルを作成します。

iSeries ナビゲーターによる供給者テーブル (SUPPLIERS) の作成

この例では、後に 2 つ目のテーブルが必要になります。このテーブルには、インベントリー項目の供給者、およびその供給者の項目のコストに関する情報が含まれます。SAMPLELIB 中に SUPPLIERS というテーブルを作成します。このテーブルには SUPPLIER_NUMBER、ITEM_NUMBER、および SUPPLIER_COST の 3 つの列があります。このテーブルには、テーブル INVENTORY_LIST との共通列

ITEM_NUMBER があることに注意してください。新しく ITEM_NUMBER 列を作成する代わりに、INVENTORY_LIST 中の ITEM_NUMBER に使用されている列定義をコピーできます。

iSeries ナビゲーターによる列定義のコピー

列定義をコピーするには、以下のようにします。

1. 「SUPPLIER テーブルのプロパティ (SUPPLIER Table Properties)」または「新規テーブル」ウィンドウで、「参照」をクリックします。
2. 「テーブルの参照」ウィンドウで、「SAMPLELIB」を展開します。
3. 「INVENTORY_LIST」をクリックします。そのテーブルの列がデータ・タイプ、サイズ、および説明とともにリストされます。
4. 「ITEM_NUMBER」を選択します。
5. 「追加」をクリックして、この列定義をテーブル SUPPLIERS にコピーします。
6. 「テーブルの参照」ウィンドウを閉じます。

以下の値を指定して、テーブル SUPPLIERS の末尾 2 列を追加します。

列名	タイプ	長さ	精度	位取り	ヌル可能	デフォルト値
SUPPLIER_NUMBER	CHAR	4			いいえ	0
SUPPLIER_COST	DECIMAL		8	2	はい	NULL

iSeries ナビゲーターによるテーブルへの情報の挿入

テーブル中にデータを挿入したり、テーブル中のデータの編集や削除を行ったりするには、そのテーブルに対する権限がなければなりません。データをテーブル INVENTORY_LIST に追加するには、以下のようになります。

1. 「iSeries ナビゲーター」ウィンドウで、使用したいシステムを展開します。
2. 「データベース」を展開し、処理したいデータベースを展開します。
3. 「スキーマ (Schemas)」を展開します。
4. 「SAMPLELIB」を選択します。
5. 「テーブル」をダブルクリックします。
6. 「INVENTORY_LIST」を右マウス・ボタン・クリックして、「内容の編集」を選択します。
7. 「行」メニューから、「挿入」を選択します。新しい行が表示されます。

該当する見出しの下に、以下のテーブル中の情報を入力してください。

注: 入力する値は、すべての制約を満たし、個々の列のタイプと一致していなければなりません。テーブルに関する固有の制約または索引がある場合には、入力する値で、固有のキー値が定義されなければなりません。列に値を入力しない場合には、デフォルトの値が入力されます (可能な場合)。この演習の場合、以下のテーブルに示されている値のみ挿入して、デフォルト値が使用されるようにしてください。

ITEM_NUMBER	ITEM_NAME	UNIT_COST	QUANTITY_ON_HAND
153047	赤鉛筆 (Pencils, red)	10.00	25
229740	罫紙 (Lined tablets)	1.50	120
544931		5.00	

ITEM_NUMBER	ITEM_NAME	UNIT_COST	QUANTITY_ON_HAND
303476	書類留めクリップ (Paper clips)	2.00	100
559343	法定封筒 (Envelopes, legal)	3.00	500
291124	標準封筒 (Envelopes, standard)		
775298	秘書用のいす (Chairs, secretary)	225.00	6
073956	黒ペン (Pens, black)	20.00	25

「ファイル」メニューから、「保管」を選択します。

以下の行を SAMPLELIB.SUPPLIERS テーブルに追加します。

ITEM_NUMBER	SUPPLIER_NUMBER	SUPPLIER_COST
153047	1234	10.00
229740	1234	1.00
303476	1234	3.00
153047	9988	8.00
559343	9988	3.00
153047	2424	9.00
303476	2424	2.50
775298	5546	225.00
303476	3366	1.50
073956	3366	17.00

「ファイル」メニューから、「保管」を選択します。この時点で、サンプル・スキーマには 2 つのテーブルが含まれ、個々のテーブルにはデータの行が複数あります。

iSeries ナビゲーターによるテーブルの内容の表示

テーブルや視点の内容を表示できます。内容は表示のみできます。テーブルに変更を加えるには、テーブルを編集しなければなりません。INVENTORY_LIST の内容を表示するには、以下のようになります。

1. 「iSeries ナビゲーター」ウィンドウで、使用したいシステムを展開します。
2. 「データベース」を展開し、処理したいデータベースを展開します。
3. 「スキーマ (Schemas)」を展開します。
4. 「テーブル」を選択します。
5. 「SAMPLELIB」をクリックします。
6. 「INVENTORY_LIST」を右マウス・ボタン・クリックして、「内容の表示」を選択します。

SAMPLELIB.INVENTORY_LIST						
ファイル(F) 編集(E) 表示(V) 行(R) ヘルプ(H)						
	ITEM_NUMBER	ITEM_NAME	UNIT_COST	QUANTITY_ON_HAND	LAST_ORDER_DATE	ORDER_QUANTITY
	153047	Pencils, red	10.00	25		20
	229740	Lined tablets	1.50	120		20
▶	544931	UNKNOWN	5.00			20
	303476	Paper clips	2.00	100		20
	559343	Envelopes, le...	3.00	500		20
	291124	Envelopes, st...	0			20
	775298	Chairs, secret...	225.00	6		20
	073956	Pens, black	20.00	25		20

iSeries ナビゲーターによるテーブル中の情報の変更

iSeries ナビゲーターを使用して、テーブルの列中のデータ値に変更を加えることができます。iSeries ナビゲーターを使用して列を更新し、今日注文を受け取った書類留めクリップの数を増やしたいとします。この列にとって有効な値を入力しなければならないことに注意してください。

1. テーブル INVENTORY_LIST にナビゲートします。このテーブルを右マウス・ボタン・クリックして、「内容の編集」を選択します。
2. 行「書類留めクリップ (Paper clips)」の「LAST_ORDER_DATE」列に現在日付を入力します。ご使用のシステムにとって正しい日付形式を使用していることを確認してください。
3. 「ORDER_QUANTITY」を 50 に変更します。
4. 変更内容を保管してから、「内容の表示」を使用してテーブルの内容を表示します。

加えた変更内容が、書類留めクリップの行に反映されます。

iSeries ナビゲーターによるテーブルからの情報の削除

iSeries ナビゲーターを使用して、テーブルからデータを削除できます。行中の 1 つの列から情報を削除することもできますし、行全体を削除することもできます。列に値が必要な場合は、行全体を削除しないとその列を削除できないことに注意してください。

1. テーブル「INVENTORY_LIST」をダブルクリックして開きます。
2. 「標準封筒 (Envelopes, standard)」行の「ORDER_QUANTITY」の列値を削除します。この列はヌル値を使用できるので、値を削除できます。
3. 「罫紙 (Lined tablets)」行の「UNIT_COST」の列値を削除します。この列はヌル値を使用できないので、削除できません。

列値を個別に除去せずに行を削除する:

すべての列値を一度に 1 つずつ除去せずに、行全体を削除することもできます。

1. テーブル「INVENTORY_LIST」をダブルクリックして開きます。
2. 「UNKNOWN」行の左側のセルをクリックします。クリックすると、行全体が強調表示されます。
3. 「行」メニューから「削除」を選択するか、またはキーボード上の Delete キーを押します。「UNKNOWN」行が削除されます。

4. テーブル「INVENTORY_LIST」から、「QUANTITY_ON_HAND」列に値のないすべての行を削除します。
5. 変更内容を保管し、「内容の表示」を使用して内容を表示します。

テーブルには以下のデータが含まれています。

ITEM_NUMBER	ITEM_NAME	UNIT_COST	QUANTITY_ON_HAND	LAST_ORDER_DATE	ORDER_QUANTITY
153047	赤鉛筆 (Pencils, red)	10.00	25		20
229740	罫紙 (Lined tablets)	1.50	120		20
303476	書類留めクリップ (Paper clips)	2.00	100	2003-09-22	50
559343	法定封筒 (Envelopes, legal)	3.00	500		20
775298	秘書用のいす (Chairs, secretary)	225.00	6		20
073956	黒ペン (Pens, black)	20.00	25		20

iSeries ナビゲーターによるテーブルのコピーおよび移動

iSeries ナビゲーターを使用すると、スキーマやシステム間でテーブルをコピーしたり移動したりできます。テーブルをコピーすると、テーブルのインスタンスが複数作成されます。移動すると、テーブルが新しい場所に転送され、以前の場所からインスタンスが除去されます。

iSeries ナビゲーターによるテーブルのコピー:

LIBRARY1 という新しいスキーマを作成し、表示されるスキーマのリストに追加します。この新しいスキーマを作成し終わったら、INVENTORY_LIST を LIBRARY1 に上書きコピーします。テーブルをコピーするには、以下のようにします。

1. 「iSeries ナビゲーター」ウィンドウで、使用したいシステムを展開します。
2. 「データベース」を展開し、処理したいデータベースを展開します。
3. 「スキーマ (Schemas)」を展開します。
4. 「テーブル」をダブルクリックします。
5. 「SAMPLELIB」をクリックします。
6. 「INVENTORY_LIST」を右マウス・ボタン・クリックして、「コピー」を選択します。
7. 「LIBRARY1」を右マウス・ボタン・クリックして、「貼り付け」を選択します。

iSeries ナビゲーターによるテーブルの移動:

テーブル INVENTORY_LIST を LIBRARY1 にコピーしたので、次にテーブル SUPPLIERS を LIBRARY1 に移動します。テーブルを移動するには、以下のようにします。

1. 「iSeries ナビゲーター」ウィンドウで、使用したいシステムを展開します。
2. 「データベース」を展開し、処理したいデータベースを展開します。
3. 「スキーマ (Schemas)」を展開します。

4. 「**テーブル**」をダブルクリックします。
5. 「**SAMPLELIB**」をクリックします。
6. 「**SUPPLIERS**」を右マウス・ボタン・クリックして、「**切り取り**」を選択します。
7. 「**LIBRARY1**」を右マウス・ボタン・クリックして、「**貼り付け**」を選択します。

注: 新しいスキーマにテーブルをドラッグ・アンド・ドロップして、テーブルを移動できます。テーブルを新しい場所に移動しても、ソース・システムから常に除去されるとは限りません。たとえば、ソースのテーブルからの読み取り権限はあるが、削除権限はない場合、テーブルをターゲット・システムに移動できます。ただし、ソース・システムからテーブルは削除できないので、テーブルの 2 つのインスタンスが存在することになります。

iSeries ナビゲーターによる視点の作成および使用

1 つのテーブルにすべての必要な情報が含まれていないことに気付くことがあります。テーブル中のデータの部分のみに対するユーザー・アクセス権を与えたいこともあります。視点を使用すると、テーブルを分割して、必要なデータのみ扱うことができます。視点を使用すると複雑さを軽減でき、同時にアクセスを制約できます。

視点を作成するには、その視点のベースとなるテーブルや物理ファイルに対する正しい権限がなければなりません。必要な権限のリストについては、「SQL 解説書」トピック中の CREATE VIEW ステートメントを参照してください。

視点定義中で列名を指定しないと、その視点のベースとなるテーブルの列名と同じ列名になります。

視点とテーブルの列や行の数が違う場合でも、視点を利用してテーブルに変更を加えることができます。INSERT の場合、テーブルにあって視点にない列はデフォルト値でなければなりません。

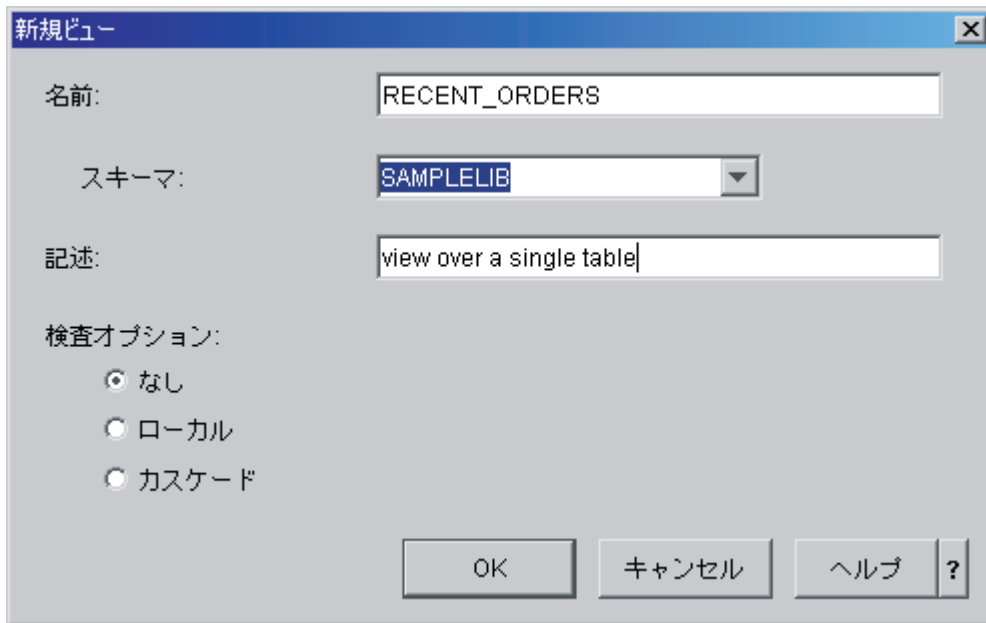
視点のデータが 1 つ以上のテーブルに完全に従属している場合でも、視点をテーブルであるかのように使用できます。視点には独自のデータがないので、データ用のストレージは必要ありません。視点はストレージ中にあるテーブルから派生するので、視点のデータを更新する際には、実際にはテーブル中のデータ更新することになります。したがって、視点が従属するテーブルが更新されるつど、視点は自動的に最新の状態に保たれます。

iSeries ナビゲーターによる 1 つのテーブルに基づく視点の作成

次の例は、1 つのテーブルに基づく視点を作成する方法を示しています。この視点は INVENTORY_LIST テーブルに基づいています。このテーブルには 6 つの列がありますが、視点ではこれらの列のうち ITEM_NUMBER、LAST_ORDER_DATE、および QUANTITY_ON_HAND の 3 つのみ使用します。

- 1 つのテーブルに基づく視点を作成するには、以下のようになります。
1. 「iSeries ナビゲーター」ウィンドウで、使用したいシステムを展開します。
2. 「データベース」を展開し、処理したいデータベースを展開します。
3. 「スキーマ (Schemas)」を展開します。
4. 「SAMPLELIB」を右マウス・ボタン・クリックして、「新規」を選択してから「ビュー」を選択します。
5. 「新規ビュー」ウィンドウで、「名前」フィールドに RECENT_ORDERS と入力します。
6. 「スキーマ」フィールドで、「SAMPLELIB」を指定します。
7. オプションで、説明を入力できます。

8. 検査オプションを選択します。視点の検査オプションは、行に挿入される値または更新される値が視点の条件に適合していなければならないことを指定します。この視点の場合は、「なし」を選択します。
9. 「OK」をクリックします。「新規ビュー」定義ウィンドウが表示されます。



新規ビュー

名前: RECENT_ORDERS

スキーマ: SAMPLELIB

記述: view over a single table

検査オプション:

なし

ローカル

カスケード

OK キャンセル ヘルプ ?

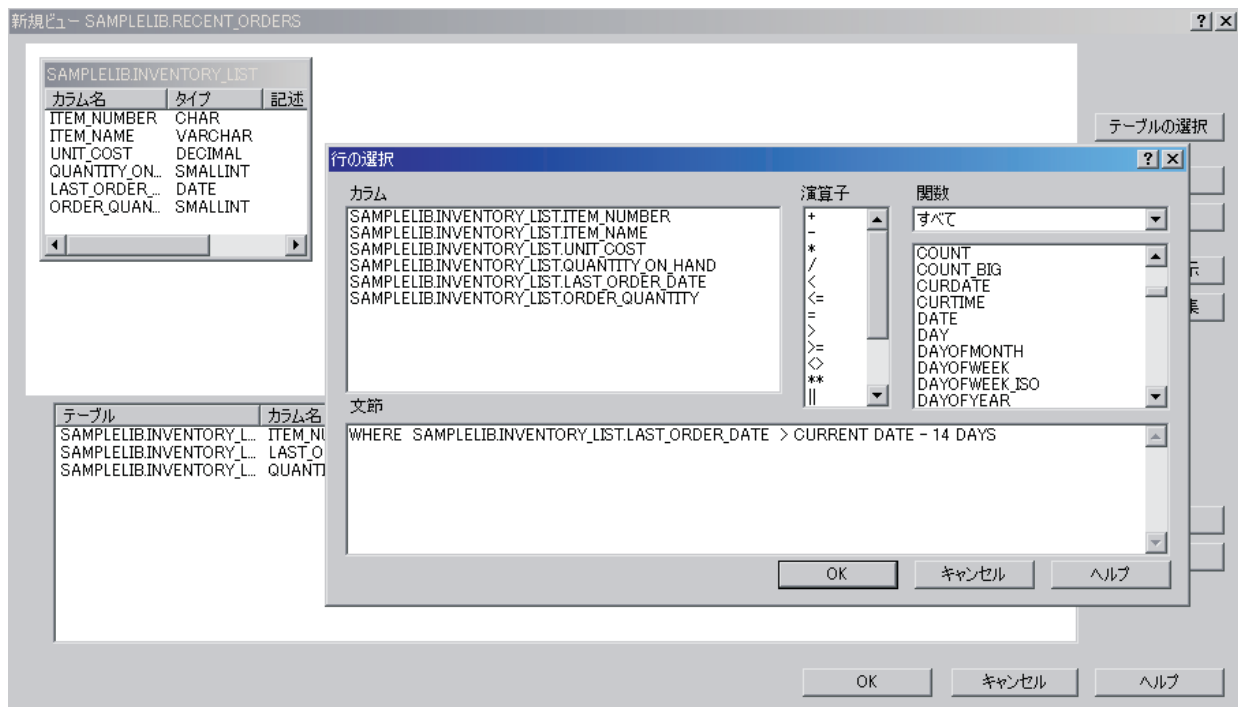
10. 「新規ビュー」ウィンドウで、「テーブルの選択」をクリックします。
11. 「テーブルの参照」ウィンドウで、「SAMPLELIB」を展開してから「INVENTORY_LIST」を選択します。
12. 「追加」をクリックします。
13. 「OK」をクリックします。この時点で、「新規ビュー」ウィンドウの作業域に INVENTORY_LIST があります。
14. 新しい視点に必要な列を選択するには、選択されたテーブルでそれらをクリックし、ウィンドウの下半分の選択グリッドにドラッグ・アンド・ドロップします。
「ITEM_NUMBER」、「LAST_ORDER_DATE」、および「QUANTITY_ON_HAND」を選択します。
15. 列は、選択グリッドに表示されている順序で視点中でも表示されます。順序を変更するには、列を選択して、新しい位置にドラッグします。列を挿入して、ITEM_NUMBER、LAST_ORDER_DATE、QUANTITY_ON_HAND の順序にします。

WHERE 文節の作成

本来この時点で視点は完成しましたが、この例では、最近の 14 日間に注文された項目のみ表示したいと思っています。この情報を指定するには、WHERE 文節を作成する必要があります。

1. 「行の選択」をクリックします。
2. 「行の選択」ウィンドウで、WHERE LAST_ORDER_DATE > CURRENT DATE - 14 DAYS と入力します。表示されたオプションからこの WHERE 文節を構成する要素を選択するか、または「文節」フィールドに入力できます。
3. 「OK」をクリックします。
4. この視点の生成時に使用する SQL を表示するには、「SQL の表示」をクリックします。

- 「OK」をクリックして、視点を作成します。



- RECENT_ORDERS の内容を表示するには、「RECENT_ORDERS」を右マウス・ボタン・クリックして、「内容の表示」を選択します。

以下の情報が表示されます。

ITEM_NUMBER	LAST_ORDER_DATE	QUANTITY_ON_HAND
303476	2003-09-22	100

上記の列では新しい名前を指定しなかったため、視点中の列はテーブル中の列と同じ名前になります。視点が従属しているテーブルのあるスキーマ中に視点も作成する必要はありません。どのスキーマでも使用できます。

iSeries ナビゲーターにより複数のテーブルからデータを結合して視点を作成する

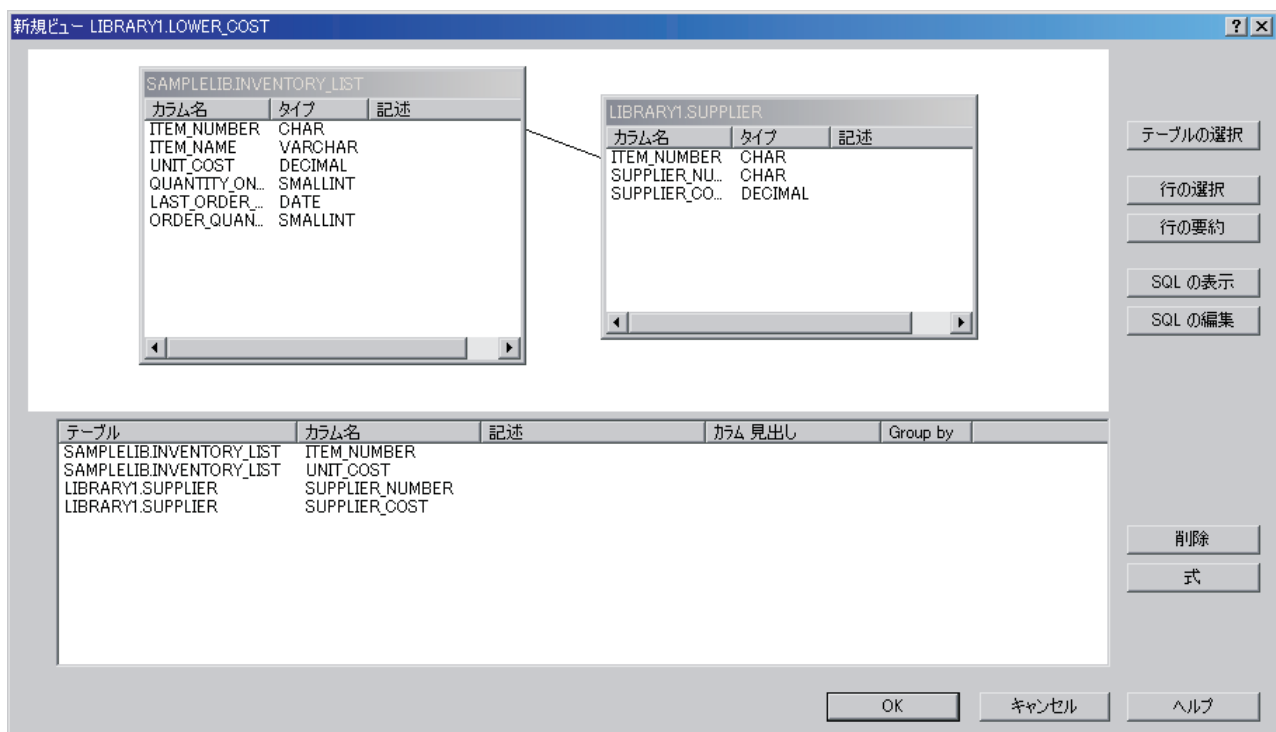
「新規ビュー」ウィンドウの作業域内の複数のテーブルを選択して、複数のテーブルから情報を結合して視点を作成できます。さまざまなテーブルから組み込みたい列を選択して「OK」をクリックし、複数のテーブルから単純な視点を作成できます。しかし、この例では、2種類のテーブルから情報を結合し、WHERE 文節の使用法に類似した方法で、参照したい行のみを戻す視点を作成する方法を示します。

この例では、現行の単位コストより安いコストで項目を供給できる供給者の項目番号のみを含む視点を作成します。この場合、INVENTORY_LIST テーブルから ITEM_NUMBER および UNIT_COST を選択し、SUPPLIERS テーブルの SUPPLIER_NUMBER および SUPPLIER_COST と結合する必要があります。WHERE 文節を使用して、戻される行数を制限します。

LOWER_COST という視点を作成するには、以下のようにします。

- スキーマ LIBRARY1 にナビゲートします。「ビュー」を右マウス・ボタン・クリックして、「新規」を選択します。

- 「SAMPLELIB」から「INVENTORY_LIST」を選択し、「LIBRARY1」から「SUPPLIERS」を選択します。
- 「OK」をクリックします。ウィンドウの作業域に両方のテーブルが表示されます。
- 「INVENTORY_LIST」から「ITEM_NUMBER」と「UNIT_COST」を選択します。
- 「SUPPLIERS」から「SUPPLIER_NUMBER」と「SUPPLIER_COST」を選択します。
- 結合を定義するには、「INVENTORY_LIST」から「ITEM_NUMBER」を選択し、「SUPPLIERS」中の「ITEM_NUMBER」にドラッグします。一方の列から他方の列に線が描画され、「結合 (Join)」ウィンドウが開きます。
- 「結合 (Join)」ウィンドウで、「一致条件を使用して行を戻す (内部結合) (Return rows with a matching condition (Inner Join))」を選択します。
- 「OK」をクリックします。
- 「行の選択」をクリックして、視点の WHERE 文節を作成します。
「LIBRARY1.SUPPLIERS.SUPPLIER_COST」をダブルクリックしてから、「<」演算子をダブルクリックし、最後に「SAMPLELIB.INVENTORY_LIST.UNIT_COST」をダブルクリックします。項目をクリックすると、ウィンドウに表示されます。また、項目を直接入力することもできます。
- 「OK」をクリックして、視点 LOWER_COST を作成します。



注: 「SQL の表示」を選択して、この視点の作成時に使用している SQL を表示できます。「SQL の編集」を選択して、SQL を編集することもできます。「SQL の編集」は Run SQL Scripts を起動し、ここで SQL ステートメントを編集できます。しかし、SQL に変更を加えた場合は、「新規ビュー」ウィンドウに戻るのではなく Run SQL Scripts からステートメントを実行する必要があることに注意してください。「新規ビュー」ウィンドウに戻ると、変更内容は保管されません。

この新しい視点の内容を表示するには、「LOWER_COST」を右マウス・ボタン・クリックして、「内容の表示」を選択します。この視点には、供給者コストが単位コストより小さい行のみ表示されます。

ITEM_NUMBER	UNIT_COST	SUPPLIER_NUMBER	SUPPLIER_COST
153047	10.00	9988	8.00
153047	10.00	2424	9.00
229740	1.50	1234	1.00
303476	2.00	3366	1.50
073956	20.00	3366	17.00

iSeries ナビゲーターによるデータベース・オブジェクトの削除

システム上にこれらのオブジェクトを作成し終えた後に、これらのオブジェクトを除去してシステム資源を節約することもできます。これらの作業を実行するには、削除権限が必要です。

注: これらのテーブル中の情報を保存するには、3 つ目のスキーマを作成し、テーブルとそのテーブルに対する視点をコピーしてください。

1. 最初に INVENTORY_LIST テーブルを LIBRARY1 から除去します。
 - a. 「iSeries ナビゲーター」ウィンドウで、使用したいシステムを展開します。
 - b. 「データベース」を展開し、処理したいデータベースを展開します。
 - c. 「スキーマ (Schemas)」を展開して、「LIBRARY1」を選択します。
 - d. 「テーブル」を選択します。
 - e. 「INVENTORY_LIST」を右マウス・ボタン・クリックして、「削除」を選択するか、Delete キーを押します。
 - f. 「オブジェクトの削除」確認ウィンドウで、「削除」を選択します。INVENTORY_LIST テーブルが除去されます。
2. 次に、SUPPLIERS を LIBRARY1 から削除し、LIBRARY1 を削除します。
 - a. 「SUPPLIERS」を右マウス・ボタン・クリックして、「削除」を選択するか、Delete キーを押します。
 - b. 「オブジェクトの削除」確認ウィンドウで、「はい」を選択します。
 - c. 新しいウィンドウが開き、視点 LOWER_COST が SUPPLIERS に従属していることが示されます。この視点も削除する必要があります。「削除」をクリックします。
 - d. 「SUPPLIERS」と「LOWER_COST」が削除されます。LIBRARY1 が空なので、右マウス・ボタン・クリックし「削除」を選択して削除します。
 - e. 「オブジェクトの削除」確認ウィンドウで、「はい」を選択します。「LIBRARY1」が削除されます。
3. 最後に、SAMPLELIB を削除します。
 - a. 「スキーマ (Schemas)」メニュー内で、SAMPLELIB にナビゲートします。
 - b. 「SAMPLELIB」を右マウス・ボタン・クリックして、「削除」を選択します。
 - c. 「オブジェクトの削除」確認ウィンドウで、「削除」を選択します。
 - d. 新しいウィンドウが開き、テーブル INVENTORY_LIST と視点 RECENT_ORDERS が INVENTORY_LIST に従属していることが示されます。これらも削除する必要があります。「はい」をクリックします。

SAMPLELIB、INVENTORY_LIST、および RECENT_ORDERS が削除されます。

iSeries ナビゲーターの使用法について詳しくは、iSeries Navigator タスク (iSeries Navigator tasks) を参照してください。

iSeries ナビゲーター・データベース・タスク

このトピックでは、iSeries ナビゲーターのインターフェースを使用して実行できるデータベース・タスクについて説明します。

『iSeries ナビゲーターを初めて使用する場合』のトピックで説明されているタスクに加えて、iSeries ナビゲーターを DB2 UDB for iSeries データベースと併用する方法は他にも多数あります。iSeries ナビゲーターをご使用のデータベースと併用する方法に関する情報は、以下のリンクを参照してください。

「データベース・プログラミング (Database programming)」トピック中のタスク

「データベース・プログラミング (Database programming)」トピック中には、以下のような他のタスクが組み込まれています。

- トリガーの追加 (Adding triggers)
- ユーザーの許可 (Authorizing a user or group to files)
- ファイル (テーブル) のコピー (Copying a file (table))
- スキーマの作成 (Creating a schema)
- ファイルの共通権限の定義 (Defining public authority for a file)
- ファイル (テーブル) の属性の表示 (Displaying attributes for a file (table))
- ロックされた行の表示 (Displaying locked rows)
- ファイル (テーブル) の移動 (Moving a file (table))
- ファイル (テーブル) の再編成 (Reorganizing a file (table))
- 新規ファイルのデフォルト共通権限の設定 (Setting a default public authority for new files)
- ジャーナルの処理 (Working with journals):
 - ジャーナルの作成 (Creating a journal)
 - ジャーナル・レシーバーの作成 (Creating a journal receiver)
 - リモート・ジャーナルの追加 (Adding a remote journal)
 - リモート・ジャーナルの除去 (Removing a remote journal)
 - リモート・ジャーナルの活動化 (Activating a remote journal)
 - リモート・ジャーナルの非活動化 (Deactivating a remote journal)
 - ジャーナル情報の表示 (Displaying journal information)
 - ジャーナル・レシーバーのスイッチング (Swapping journal receivers)
 - ジャーナルの開始および停止 (Starting and stopping a journal)

「データベース・パフォーマンスおよび Query 最適化」トピック中のタスク

「データベース・パフォーマンスおよび Query 最適化」トピック中には、以下のタスクが組み込まれています。

- ジョブ・ログの Query 最適化プログラム・デバッグ・メッセージの調査
- PRSQLINF コマンドによる組み込み SQL ステートメントに関する情報収集
- データベース・モニターの開始 (STRDBMON) による QUERY のモニター (Monitoring your queries using Start Database Monitor (STRDBMON))
- メモリー常駐データベース・モニターによるデータベースのモニター (Monitoring your database with the memory-resident database monitor)
- ビジュアル・エクスプレインを使用した照会の効果の表示
- 照会属性の変更 (CHGQRYA) コマンドによる照会の属性の変更 (Change the attributes of your queries with the Change Query Attributes (CHGQRYA) command)
- 統計マネージャーによる照会の分析

データベース・ナビゲーター・マップを使用したデータベースのマッピング

「データベース・ナビゲーター」により、システム上のデータベース・オブジェクトの関係を視覚的に描写することができます。この描写はマップと呼ばれます。本質的には、データベース・ナビゲーター・マップは、データベース、およびマップ内のすべてのオブジェクト間に存在する関係のスナップショットです。

「データベース・ナビゲーター」を使用して、データベース内のテーブル、テーブル間の関係、およびテーブルに付加された索引と制約を示すグラフィカル表示を使用してデータベース・オブジェクトの複雑な関係を探索できます。「データベース・ナビゲーター」の基本ワークスペースは、複数の主な区域に分割されているウィンドウです。マップは右側のペインに表示されます。オブジェクトを右マウス・ボタン・クリックすると、さまざまなタスクを実行できます。ウィンドウの左側に「ロケーター」ペインがあります。このペインを使用して、マップ中に組み込む特定のオブジェクトを見つけたり、マップ中に組み込むオブジェクトのタイプを指定したりできます。

1. システム名、「データベース」、および使用したいデータベースを展開すると、「データベース・ナビゲーター」を使用できます。
2. 右側のペインに既存のマップのリストを表示するには、「データベース・ナビゲーター・マップ (Database Navigator Maps)」をクリックします。
3. 新しいマップを作成するには、「データベース・ナビゲーター・マップ (Database Navigator Maps)」を右マウス・ボタン・クリックして、「新規」→「マップ (Map)」を選択します。

「データベース・ナビゲーター」を使用する場合のヒント:

- ウィンドウの一方の側のサイズを変更するには、左右の側を分離しているバー (スプリッター) をドラッグする。
- ウィンドウの左側と右側の両方でオブジェクトを右マウス・ボタン・クリックしたことを確認してください。右マウス・ボタン・クリックすると表示されるメニューにより、よく使用される機能に即時アクセスできます。
- 即時にスキーマを開いてその中のオブジェクトを表示するには、そのスキーマをダブルクリックする。
- さまざまなデータベース・ナビゲーター・コマンドにアクセスするには、「メニュー・バー」または「ツールバー」を使用する。

オンライン・ヘルプに、「データベース・ナビゲーター」に関する詳細情報があります。

Run SQL Scripts を使用したデータベースの Query

iSeries ナビゲーターの「Run SQL Scripts」ウィンドウを使用すると、SQL ステートメントのスクリプトを作成、編集、実行、および、トラブルシューティングすることができます。スクリプトの処理が終了した時には、スクリプトをユーザーの PC に保管することができます。システム名および「データベース」を展開し、接続したいデータベースを右マウス・ボタン・クリックすると、Run SQL Scripts を起動できます。

「例」のリストを使用して、スクリプトを作成したり、ステートメントを手動で作成したり、SQL 生成関数を使用して既存のオブジェクトの SQL を検索したり、SQL アシストを使用してスクリプトを作成したりできます。詳細については、20 ページの『SQL アシストによる SQL ステートメントの作成』を参照してください。

また、「**構文検査**」をクリックして、SQL の構文を検査することもできます。プログラムやスクリプトをデバッグする追加の方法には、ジョブ・ログ中のメッセージのデバッグと、iSeries システム・デバッガーの起動が含まれます。詳細については、20 ページの『ジョブ・ログの表示』および 21 ページの『iSeries システム・デバッガーの起動』を参照してください。構文検査が完了したら、「**ファイル**」メニューから「**保管**」を選択して、スクリプトを保管できます。

SQL スクリプトを実行するには、「**実行**」メニューから以下のいずれかのオプションを選択してください。

- 「**すべて**」 - SQL スクリプトを最初から最後まで実行します。エラーが起こって、「**エラー時に停止**」オプションがオンになっている場合には、プログラムは停止し、エラーの起こったステートメントは選択されたままの状態に残ります。
- 「**選択元**」 - 選択された最初のステートメントから SQL スクリプトを開始し、スクリプトの終わりまで続行します。
- 「**選択**」 - 選択されたステートメントを実行します。

結果は「**メッセージ**」タブの最後に追加されます。「**オプション**」メニューの「**スマート・ステートメント選択**」オプションにチェックがない場合には、選択されたテキストは単一の SQL ステートメントとして実行されます。

詳細については、以下のトピックを参照してください。

- 『Run SQL Scripts の停止』
- 20 ページの『ジョブ・ログの表示』
- 20 ページの『オブジェクトの SQL の生成』
- 20 ページの『SQL アシストによる SQL ステートメントの作成』
- 21 ページの『iSeries システム・デバッガーの起動』

Run SQL Scripts の停止

SQL スクリプトの実行を停止したり取り消したりするには、「**実行**」メニューから以下のいずれかのオプションを選択してください。

- 「**現行処理後に停止 (Stop After Current)**」 - 現在実行されているステートメントが終了した後で SQL スクリプトの実行を停止します。
- 「**要求のキャンセル**」 - 現行 SQL ステートメントのシステム取り消しを要求します。しかし、取り消せない SQL ステートメントがある場合は、このオプションの使用後もその SQL ステートメントは引き続き最後まで実行されます。「**要求のキャンセル**」を押す前にホスト処理が完了している SQL ステートメント

ントも、引き続き最後まで実行されます。例えば、すでに QUERY 処理を完了しているがまだクライアントに結果を戻していない SELECT ステートメントは、一般には取り消せません。

オンライン・ヘルプに、Run SQL Scripts に関する詳細情報があります。

ジョブ・ログの表示

「ジョブ・ログ」には、ユーザーのジョブに関連したメッセージが表示されます。

- 照会最適化プログラムおよびその他のデータベース・デバッグ・メッセージを表示するには、「オプション」メニューから「ジョブ・ログへのデバッグ・メッセージの組み込み」を選択し、ステートメントを再び実行する。この作業時に「ジョブ・ログ」ダイアログが開いている場合には、表示を最新表示して、新しいメッセージを表示してください。
- ジョブ・ログを表示するには、「表示」メニューから「ジョブ・ログ」を選択する。

「実行履歴の消去」の使用時には「ジョブ・ログ」は消去されないで、「ジョブ・ログ」を使用することにより、「出力」ペインに残っていないメッセージを表示することができます。

オンライン・ヘルプに、ジョブ・ログに関する詳細情報があります。

オブジェクトの SQL の生成

SQL の生成によって、既存のデータベース・オブジェクトの作成に使用された SQL を再構成できます。このプロセスは、「リバース・エンジニアリング」といわれています。ほとんどのデータベース・オブジェクトの SQL を生成できます。さらに、制約またはトリガーが関連付けられているテーブルの SQL を生成する場合には、これらの SQL も生成されることになります。一度に 1 つまたは多数のオブジェクトの SQL を生成できます。また、生成された SQL を Run SQL Scripts ウィンドウに送信して実行や編集を行うオプションもありますし、生成された SQL をデータベースか PC ファイルに直接書き込むこともできます。

- オブジェクトの SQL を生成するには、そのオブジェクトを右マウス・ボタン・クリックして「SQL の生成」を選択する。
- 「編集」メニューから「生成された SQL の挿入 (Insert Generated SQL)」を選択して、Run SQL Scripts から「SQL の生成」を起動することもできる。

オンライン・ヘルプに、「SQL の生成」に関する詳細情報があります。

SQL アシストによる SQL ステートメントの作成

SQL アシストを使用して、SQL ステートメントを対話式に作成できます。SQL アシストは、選択、挿入、更新、および削除の各ステートメントを作成するのに役立ちます。

- SQL アシストを起動するには、Run SQL Scripts の「編集」メニューから「SQL アシスト」を選択する。SQL アシストのインターフェースから、処理対象のテーブルを選択したり、選択基準を作成したりできます。このインターフェースの下部でステートメントを作成します。
- 作成したステートメントを Run SQL Scripts に戻すには、「OK」をクリックする。
- ステートメントの編集、実行、および保管を行えます。

オンライン・ヘルプに、SQL アシストに関する詳細情報があります。

iSeries システム・デバッガーの起動

iSeries システム・デバッガーは、iSeries サーバー上に新しいグラフィカル・ユーザー・デバッグ環境を備えます。iSeries システム・デバッガーを使用して、iSeries サーバー上で実行するプログラム (OS/400 PASE 環境で実行するプログラムを含む) のデバッグやテストを行えます。

Run SQL Scripts からシステム・デバッガーを起動するには、「実行」メニューから「デバッガー (Debugger)」を選択してください。

iSeries システム・デバッガーに関する詳細は、iSeries システム・デバッガー (iSeries System Debugger) トピックまたはオンライン・ヘルプを参照してください。

iSeries ナビゲーターを使用したオブジェクトの作成および管理

iSeries ナビゲーターで多数のオブジェクトを作成し、管理することができます。

以下のオブジェクトの作成と管理を行えます。

- スキーマ
- テーブル - マテリアライズ照会テーブルおよび区分テーブルを含む
- 別名
- シーケンス
- SQL パッケージ
- ユーザー定義関数 (UDF)
- ユーザー定義の特殊タイプ (UDT)
- プロシージャ
- 索引
- トリガー
- 制約
- 視点
- ジャーナル
- ジャーナル・レシーバー

ほとんどのオブジェクトは、スキーマ・コンテナー・オブジェクトから作成されます。スキーマ・コンテナーにナビゲートするには、以下のようにします。

- システム名、「データベース」、および使用したいデータベースを展開します。「スキーマ (Schemas)」を展開し、処理したいスキーマを右マウス・ボタン・クリックして、「新規」を選択します。
- 作成したいオブジェクトのタイプを選択します。
- 代わりに、処理したいスキーマを展開し、作成したいコンテナー・タイプを右マウス・ボタン・クリックすることもできます。「スキーマ (Schemas)」コンテナーからスキーマが作成されます。SQL パッケージはシステム・レベルで作成されます。システム名を右マウス・ボタン・クリックし、「新規」 → 「SQL パッケージ」を選択します。

オンライン・ヘルプに、オブジェクトの作成に関する詳細情報があります。

検査保留制約の管理

システムによって検査保留状態にされている制約の表示や変更を行えます。検査保留は、参照制約の場合はいずれかの親キーと外部キーの間に不一致がある状態を参照し、チェック制約の場合は列値とチェック制約定義の間に不一致がある状態を参照します。

1. 検査保留状態にされている制約を表示するには、システム名および「データベース」を展開します。使用したいデータベースを右マウス・ボタン・クリックし、「**検査保留制約の管理 (Manage check pending constraints)**」を選択します。
2. このインターフェースから、制約の定義や、制約規則に違反している行を表示できます。処理したい制約を選択してから、「ファイル」メニューから「**検査保留制約の編集 (Edit Check Pending Constraint)**」を選択します。
3. 違反状態の行に変更を加えたり削除したりできます。

検査保留制約の詳細については、「データベース・プログラミング (Database programming)」トピック中の参照制約における検査保留状況 (Check pending status in referential constraints) およびオンライン・ヘルプを参照してください。

SQL を初めて使用する場合

この項では、対話式 SQL で SQL ステートメントを使用してスキーマ、テーブル、および視点の作成と処理を行う方法について説明します。

SQL 解説書トピックでは、この章で使用されている個々の SQL ステートメントの構文について詳述され、さらに複雑な状態で SQL ステートメントと文節を使用する方法について説明されています。

この項の例では、対話式 SQL インターフェースを使用して、SQL ステートメントの使用法を示します。個々の SQL インターフェースにより、SQL ステートメントを使用してテーブルや視点などのオブジェクトを定義したり、オブジェクトを更新したり、オブジェクトからデータを読み取ったりできるようになります。

詳細については、以下のトピックを参照してください。

- スキーマの作成
- テーブルの作成および使用
- LABEL ON ステートメントの使用
- テーブルへの情報の挿入
- 1 つのテーブルからの情報の入手
- 複数のテーブルからの情報の入手
- テーブル中の情報の変更
- テーブルからの情報の削除
- 視点の作成および使用

まず、対話式 SQL を開始します。

1. STRSQL NAMING(*SQL) と入力します。
2. **Enter** キーを押します。

「SQL ステートメントの入力」画面が表示されたら、SQL ステートメントの入力を始められます。対話式 SQL および STRSQL コマンドに関する詳細については、SQL プログラミング (SQL Programming) を参照してください。

既存の対話式 SQL セッションを再利用する場合は、命名モードを **SQL 命名** に設定していることを確認してください。F13 (サービス) パネルのオプション 1 (セッション属性の変更) で指定できます。

スキーマの作成

スキーマとは、テーブル、視点、索引、およびパッケージが組み込まれる基本オブジェクトのことです。スキーマの作成について詳しくは、「SQL 解説書」トピックの SQL CREATE SCHEMA ステートメントを参照してください。

注: スキーマの同義語としてコレクション という用語が使われる場合があります。

SAMPLECOLL という名前のサンプル・スキーマを作成するには、以下のようになります。

SQL ステートメントの入力

SQL ステートメントを入力して、実行キーを押してください。
現在の接続相手はリレーショナル・データベース SYSTEM1 である。

```
====> CREATE SCHEMA SAMPLECOLL
```


_____ 終わり

F3= 終了 F4=プロンプト F6= 行の挿入 F9=コマンドの複写 F10= 行のコピー
F12= 取り消し F13= サービス F24= キーの続き

1. 「SQL ステートメントの入力」画面で、以下の SQL ステートメントを入力します。
2. **Enter** キーを押します。

注: このステートメントを実行すると複数のオブジェクトが作成され、この実行には数秒かかります。

スキーマを正常に作成し終わったら、その中にテーブル、視点、および索引を作成できます。スキーマではなくライブラリー中にテーブル、視点、および索引を作成することもできます。

テーブルの作成および使用

SQL CREATE TABLE ステートメントを使用して、テーブルを作成できます。CREATE TABLE ステートメントを使用すると、テーブルの作成、テーブル中の列の物理属性の定義、およびテーブルで使用できる値を制約する制約の定義を行えます。

テーブルを作成する場合は、ヌル値とデフォルト値の概念について理解する必要があります。ヌル値とは、ある行の列値が存在しないことを意味します。ヌル値は、ゼロまたはすべてブランクの値とは異なります。「未知」の意味になります。ヌル値は、他のヌル値を含むどの値とも等しくありません。列でヌル値を使用できない場合は、デフォルト値かユーザー指定値をその列に割り当てなければなりません。

行をテーブルに追加する際に、その列の値を指定しないと、デフォルト値がその列に割り当てられます。列の特定のデフォルト値を定義しないと、システム・デフォルト値が使用されます。INSERT で使用されるデフォルト値について詳しくは、SQL 解説書を参照してください。

ここでは、ある業務の現行インベントリーに関する情報を保守するテーブルを作成します。インベントリー中に保持されている項目、それらの項目のコスト、現在在庫している数量、最新の注文日付、および最新の

注文数に関する情報があります。項目番号は必須値です。この値はヌルにできません。項目名、在庫している数量、および注文数量はユーザー指定のデフォルト値になります。最新の注文日付と注文数量はヌル値にすることができます。

2 つ目のテーブルを作成することもできます。このテーブルには、インベントリー項目の供給者、およびその供給者の項目のコストに関する情報が含まれます。

1. 最初に、以下のステップに従って、 INVENTORY_LIST という名前の 1 つ目のテーブルを作成します。

a. 「SQL ステートメントの入力」画面で、 CREATE TABLE と入力して、F4 (プロンプト) キーを押します。以下の画面が表示されます (入力域はまだ入力されていません)。

CREATE TABLE ステートメントの指定

情報を入力して、実行キーを押してください。

テーブル INVENTORY_LIST _____ 名前
 コレクション SAMPLECOLL _____ 名前, リスト F4 キー

ヌル値 : 1=NULL, 2=NOT NULL, 3=NOT NULL WITH DEFAULT

列	FOR 列	タイプ	長さ	スケール	NULL
ITEM_NUMBER _____	_____	CHAR _____	6 _____	_____	2 _____
ITEM_NAME _____	_____	VARCHAR _____	20 _____	_____	3 _____
UNIT_COST _____	_____	DECIMAL _____	8 _____	2 _____	3 _____
QUANTITY_ON_HAND _____	_____	SMALLINT _____	_____	_____	1 _____
LAST_ORDER_DATE _____	_____	DATE _____	_____	_____	1 _____
ORDER_QUANTITY _____	_____	SMALLINT _____	_____	_____	1 _____
_____	_____	_____	_____	_____	3 _____
					3 終わり

テーブルの制約事項 N Y=Yes, N=No
 分散テーブル N Y=Yes, N=No

F3= 終了 F4= プロンプト F5= 最新表示 F6= 行挿入 F10= 行のコピー
 F11= 属性の続きの表示 F12= 取り消し F14= 行削除 F24= キーの続き

b. 上記のように、テーブル名 INVENTORY_LIST とスキーマ名 SAMPLECOLL を「テーブル」および「コレクション」プロンプトに入力します。

c. テーブルの定義したい個々の列は、画面の下部にあるリスト内の項目で表されます。列ごとに、列の名前、列のデータ・タイプ、列の長さや位取り、およびヌル属性を入力します。

d. 指定できる列属性をさらに表示するには、F11 キーを押します。ここにデフォルト値を指定することもできます。

CREATE TABLE ステートメントの指定

情報を入力して、実行キーを押してください。

テーブル INVENTORY_LIST _____ 名前
 コレクション SAMPLECOLL_ _____ 名前, リストは F4 キー

データ: 1=BIT, 2=SBCS, 3=MIXED, 4=CCSID

列	データ	割振り	CCSID	制約	省略時の値
ITEM_NUMBER _____	-	_____	_____	N	_____
ITEM_NAME _____	-	_____	_____	N	'***UNKNOWN***' _____
UNIT_COST _____	-	_____	_____	N	_____
QUANTITY_ON_HAND _____	-	_____	_____	N	NULL _____
LAST_ORDER_DATE _____	-	_____	_____	N	_____
ORDER_QUANTITY _____	-	_____	_____	N	20 _____
_____	-	_____	_____	-	_____
終わり					
テーブルの制約事項				N	Y=Yes, N=No
分散テーブル				N	Y=Yes, N=No

F3= 終了 F4= プロンプト F5= 最新表示 F6= 行挿入 F10= 行のコピー
 F11= 属性の続きの表示 F12= 取り消し F14= 行削除 F24= キーの続き

注: 列定義を入力する別の方法として、リスト内の列項目の 1 つにカーソルを置いて F4 (プロンプト) キーを押すという方法もあります。1 つの列を定義する属性をすべて表示した画面が表示されます。

- e. すべての値を入力し終わったら、Enter キーを押してテーブルを作成します。「SQL ステートメントの入力」画面が再表示され、テーブルが作成されたことを示すメッセージが表示されます。

注: 以下のように、この CREATE TABLE ステートメントを「SQL ステートメントの入力」画面に入力できます。

```
CREATE TABLE SAMPLECOLL.INVENTORY_LIST
(ITEM_NUMBER CHAR(6) NOT NULL,
 ITEM_NAME VARCHAR(20) NOT NULL WITH DEFAULT '***UNKNOWN***',
 UNIT_COST DECIMAL(8,2) NOT NULL WITH DEFAULT,
 QUANTITY_ON_HAND SMALLINT DEFAULT NULL,
 LAST_ORDER_DATE DATE,
 ORDER_QUANTITY SMALLINT DEFAULT 20)
```

- 2. 次に、SUPPLIERS という名前の 2 つ目のテーブルを作成します。以下の 2 つの方式を使用できます。

- a. 「SQL ステートメントの入力」画面で、以下のコマンドを直接入力します。
- b. F4 (プロンプト) キーを押し、対話式 SQL の画面を使用して定義を作成します。

```
CREATE TABLE SAMPLECOLL.SUPPLIERS
(SUPPLIER_NUMBER CHAR(4) NOT NULL,
 ITEM_NUMBER CHAR(6) NOT NULL,
 SUPPLIER_COST DECIMAL(8,2))
```

このコードに関する特記事項は、コーディング例に関連しています。

LABEL ON ステートメントの使用

通常、対話式 SQL 中で SELECT ステートメントの出力が表示される際には、列見出しとして列名が使用されます。LABEL ON ステートメントを使用すると、より内容の分かりやすい列名のラベルを作成することもできます。対話式 SQL でこの例を実行しようとしているので、LABEL ON ステートメントを

使用して列見出しを変更することになります。列名が分かりやすい場合でも、列見出し中に名前の個々の部分が 1 行で示されている方が読みやすくなります。1 つの画面にデータの列をさらに表示することもできます。

例の列のラベルを変更するには、以下のようになります。

1. 「SQL ステートメントの入力」画面で、LABEL ON COLUMN と入力します。
2. F4 (プロンプト) キーを押します。以下の画面が表示されます。

LABEL ON ステートメントの指定

選択項目を入力して、実行キーを押してください。

ラベル・オン	2		
		1= テーブルまたはビュー	
		2= 列	
		3= パッケージ	
		4= 別名	
テーブルまたはビュー	INVENTORY_LIST _____	名前, リストは F4 キー	
コレクション . . .	SAMPLECOLL _____	名前, リストは F4 キー	
オプション	1	1= 列見出し	
		2= テキスト	

F3= 終了 F4= プロンプト F5= 最新表示 F12= 取り消し
F20= 名前全体の表示 F21=ステートメントの表示

3. ラベルを追加したい列を含むテーブルとスキーマの名前を入力します。
4. Enter キーを押します。以下の画面が表示され、テーブル中の列ごとにプロンプトが出されます。

LABEL ON ステートメントの指定

情報を入力して、実行キーを押してください。

	列見出し			
列1.....2.....3.....4.....5.....			
ITEM_NUMBER	'ITEM	NUMBER'	_____	
ITEM_NAME	'ITEM	NAME'	_____	
UNIT_COST	'UNIT	COST'	_____	
QUANTITY_ON_HAND	'QUANTITY	ON	HAND'	_____
LAST_ORDER_DATE	'LAST	ORDER	DATE'	_____
ORDER_QUANTITY	'NUMBER	ORDERED'	_____	

終わり

F3= 終了 F5= 最新表示 F6= 行挿入 F10= 行のコピー F12= 取り消し
F14= 行削除 F19= システム列名の表示 F20= 名前全体の表示 F24= キーの続き

5. 列ごとに列見出しを入力します。列見出しは 20 文字のセクションで定義します。SELECT ステートメントの出力が表示される際に、個々のセクションは別々の行で表示されます。列見出しの入力域の上部に引かれているけい線を利用して、見出しの間隔を正しくあけることができます。
6. Enter キーを押します。

以下のメッセージは、LABEL ON ステートメントが正常に実行されたことを示します。

LABEL ON for INVEN00001 in SAMPLECOLL completed. (SAMPLECOLL 中の INVEN00001 の LABEL ON が完了しました。)

メッセージ中のテーブル名は、実際にステートメントで指定した名前ではなく、このテーブルのシステム・テーブル名になります。DB2 UDB for iSeries には、10 文字より長いテーブルの名前として 2 つの名前が保守されています。システム・テーブル名について詳しくは、SQL 解説書トピックの CREATE TABLE ステートメントを参照してください。

注: 以下のように、LABEL ON ステートメントを「SQL ステートメントの入力」画面に直接キー入力することもできます。

```
LABEL ON SAMPLECOLL.INVENTORY_LIST
(ITEM_NUMBER      IS 'ITEM      NUMBER ',
ITEM_NAME         IS 'ITEM      NAME ',
UNIT_COST        IS 'UNIT      COST ',
QUANTITY_ON_HAND IS 'QUANTITY ON      HAND ',
LAST_ORDER_DATE  IS 'LAST      ORDER DATE ',
ORDER_QUANTITY   IS 'NUMBER    ORDERED ')
```

テーブルへの情報の挿入

テーブルの作成後に、SQL INSERT ステートメントを使用して、そのテーブルに情報 (データ) の挿入したり追加したりできます。

1. 「SQL ステートメントの入力」画面で、INSERT と入力して、F4 (プロンプト) キーを押します。「INSERT ステートメントの指定 (Specify INSERT Statement)」画面が表示されます。

INSERT ステートメントの指定

選択項目を入力して、実行キーを押してください。

INTO テーブル	INVENTORY_LIST _____	名前, リストは F4 キー
コレクション	SAMPLECOLL_	名前, リストは F4 キー
INTO の挿入列の選択	Y	Y=YES, N=NO
挿入方法	1	1=VALUES の入力 2= 部分選択

選択項目を入力して、実行キーを押してください。

WITH 分離レベル	1	1= 現行レベル, 2=NC (NONE) 3=UR (CHG), 4=CS, 5=RS (ALL) 6=RR
----------------------	---	---

F3= 終了 F4= プロンプト F5= 最新表示 F12= 取り消し
F20= 名前全体の表示 F21=ステートメントの表示

2. 示されているように、入力フィールドにテーブル名とスキーマ名を入力します。
3. 「挿入先の列の選択 (Select columns to insert INTO)」プロンプトを「はい」に変更します。
4. Enter キーを押して、値の挿入先にしたい列を選択できる画面を表示します。

INSERT ステートメントの指定

選択するためには順序番号 (1-999) を入力して、実行キーを押してください。

SEQ	列	タイプ	長さ	スケール
1	ITEM_NUMBER	CHARACTER	6	
2	ITEM_NAME	VARCHAR	20	
3	UNIT_COST	DECIMAL	8	2
4	QUANTITY_ON_HAND	SMALLINT	4	
	LAST_ORDER_DATE	DATE		
	ORDER_QUANTITY	SMALLINT	4	

終わり

F3= 終了 F5= 最新表示 F12= 取り消し F19= システム列名の表示
F20= 名前全体の表示 F21=ステートメントの表示

この例では、4 つの列に挿入します。その他の列にはデフォルト値が挿入されるようにします。この画面上のシーケンス番号は、INSERT ステートメント中で列と値がリストされる順序を示します。

5. Enter キーを押して、選択した列の値を入力できる画面を表示します。

INSERT ステートメントの指定

挿入する値を入力して、実行キーを押してください。

列	値
ITEM_NUMBER	'153047'
ITEM_NAME	'Pencils, red'
UNIT_COST	10.00
QUANTITY_ON_HAND	25

終わり

F3= 終了 F5= 最新表示 F6= 行挿入 F10= 行のコピー F11= タイプの表示
F12= 取り消し F14= 行削除 F15= 行分割 F24= キーの続き

注: 挿入リスト中の列ごとにデータ・タイプと長さを表示するには、F11 (タイプの表示 (Display type)) キーを押してください。このキーを押すと、別の値挿入画面が表示され、列定義に関する情報が表示されます。

6. すべての列の挿入値を入力して、Enter キーを押します。これらの値を含む行がテーブルに追加されます。列の値を指定しないと、その列にはデフォルト値が挿入されます。LAST_ORDER_DATE の場合、デフォルトが指定されず、この列はヌル値を使用できるので、ヌル値になります。ORDER_QUANTITY の場合、CREATE TABLE ステートメントでデフォルト値として指定されている値の 20 になります。
7. 以下のように、INSERT ステートメントを「SQL ステートメントの入力」画面に入力できます。

```
INSERT INTO SAMPLECOLL.INVENTORY_LIST
  (ITEM_NUMBER,
   ITEM_NAME,
   UNIT_COST,
   QUANTITY_ON_HAND)
```

```
VALUES ('153047 ',
       'Pencils,red ',
       10.00,
       25)
```

このコードに関する特記事項は、コーディング例に関連しています。

8. 次の行をテーブルに追加するには、「SQL ステートメントの入力」画面で F9 (検索) キーを押します。このキーを押すと、前述の INSERT ステートメントが入力域にコピーされます。前述の INSERT ステートメントの値を上書きするか、F4 (プロンプト) キーを押して対話式 SQL の画面を使ってデータを入力できます。
9. 引き続き INSERT ステートメントを使用して、以下の行をテーブルに追加します。

以下の図表に示されていない値は挿入せずに、デフォルトを使用できるようにしてください。INSERT ステートメント列のリストに、値を挿入したい列名のみ指定してください。例えば、3 つ目の行を挿入するには、列名として ITEM_NUMBER および UNIT_COST のみ指定し、VALUES リスト中にこれらの列の 2 つの値のみ指定してください。

ITEM_NUMBER	ITEM_NAME	UNIT_COST	QUANTITY_ON_HAND
153047	赤鉛筆 (Pencils, red)	10.00	25
229740	罫紙 (Lined tablets)	1.50	120
544931		5.00	
303476	書類留めクリップ (Paper clips)	2.00	100
559343	法定封筒 (Envelopes, legal)	3.00	500
291124	標準封筒 (Envelopes, standard)		
775298	秘書用のいす (Chairs, secretary)	225.00	6
073956	黒ペン (Pens, black)	20.00	25

以下の行を SAMPLECOLL.SUPPLIERS テーブルに追加します。

SUPPLIER_NUMBER	ITEM_NUMBER	SUPPLIER_COST
1234	153047	10.00
1234	229740	1.00
1234	303476	3.00
9988	153047	8.00
9988	559343	3.00
2424	153047	9.00
2424	303476	2.50
5546	775298	225.00
3366	303476	1.50
3366	073956	17.00

この時点で、サンプル・スキーマには 2 つのテーブルが含まれ、個々のテーブルにはデータの行が複数あります。

1 つのテーブルからの情報の入手

例のテーブルにすべての情報を挿入し終えたので、再表示しなければなりません。SQL では、SELECT ステートメントを使用してこの作業を行います。SELECT ステートメントは、すべての SQL ステートメントの中で最も複雑なステートメントです。このステートメントは、3 つの主要な文節で構成されます。

1. SELECT 文節。データを含む列を指定します。
2. FROM 文節。データのある列を含む 1 つまたは複数のテーブルを指定します。
3. WHERE 文節。どのデータ行を検索するか判別する条件を指定します。

3 つの主要な文節に加えて、戻されるデータの最終的な形式に影響する文節が他にも複数あります。これらの文節については、SQL プログラミング (SQL Programming) トピックと、「SQL 解説書」トピックで説明されています。

1. INVENTORY_LIST テーブルに挿入した値を表示するには、SELECT と入力して、F4 (プロンプト) を押します。以下の画面が表示されます。

SELECT ステートメントの指定

SELECT ステートメント情報を入力してください。リストの表示は、F4 キーを押してください。

FROM テーブル SAMPLECOLL.INVENTORY_LIST

SELECT 列 * _____

WHERE 条件 _____

GROUP BY 列 _____

HAVING 条件 _____

ORDER BY 列 _____

FOR UPDATE OF 列 _____

終わり

選択項目を入力して、実行キーを押してください。

結果テーブル中の DISTINCT 行 N Y=Yes, N=No

別の SELECT との UNION N Y=Yes, N=No

追加オプションの指定 N Y=Yes, N=No

F3= 終了 F4= プロンプト F5= 最新表示 F6= 行挿入 F9=SUBQUERY の指定
 F10= 行のコピー F12= 取り消し F14= 行削除 F15= 行分割 F24= キーの続き

2. 画面上の「**選択元のテーブル (FROM tables)**」フィールドにテーブル名を入力します。このテーブル中の列をすべて選択するには、画面上の「**選択列 (SELECT columns)**」フィールドに * を入力します。

データの表示

データの幅 : 71

行の位置指定 桁移動

.....1.....2.....3.....4.....5.....6.....7.

ITEM NUMBER	ITEM NAME	UNIT COST	QUANTITY ON HAND	LAST ORDER DATE	NUMBER ORDERED
153047	Pencils, red	10.00	25	-	20
229740	Lined tablets	1.50	120	-	20
544931	***UNKNOWN***	5.00	-	-	20
303476	Paper clips	2.00	100	-	20
559343	Envelopes, legal	3.00	500	-	20
291124	Envelopes, standard	.00	-	-	20
775298	Chairs, secretary	225.00	6	-	20
073956	Pens, black	20.00	25	-	20

***** データの終わり *****

終わり

F3= 終了 F12= 取り消し F19= 左 F20= 右 F21= 分割

- Enter キーを押すと、ステートメントが実行され、テーブル中のすべての列のすべてのデータが選択されます。以下の出力が表示されます。

LABEL ON ステートメントを使用して定義した列見出しが表示されます。3 つ目の項目の ITEM_NAME には、CREATE TABLE ステートメントで指定したデフォルト値が入っています。QUANTITY_ON_HAND 列の、値が挿入されていない行には、ヌル値が入っています。LAST_ORDER_DATE 列にはすべてヌル値が含まれています。なぜなら、この列はどの INSERT ステートメント中にもなく、またデフォルト値を含めるよう定義されていないからです。同様に、ORDER_QUANTITY 列のすべての行にはデフォルト値が含まれています。

以下のように、このステートメントを「SQL ステートメントの入力」画面に入力できます。

```
SELECT *
FROM SAMPLECOLL.INVENTORY_LIST
```

- SELECT ステートメントによって戻される列の数を制限するには、表示したい列を指定しなければなりません。戻される出力行の数を制約するには、WHERE 文節を使用します。列 ITEM_NUMBER、UNIT_COST、および ITEM_NAME の値のみ戻し、かつコストが 10 ドルより高い項目のみ表示するには、SELECT と入力して F4 (プロンプト) キーを押します。「SELECT ステートメントの指定 (Specify SELECT Statement)」画面が表示されます。

SELECT ステートメントの指定

SELECT ステートメント情報を入力してください。リストの表示は、F4 キーを押してください。

FROM テーブル	SAMPLECOLL.INVENTORY_LIST	
SELECT 列	ITEM_NUMBER, UNIT_COST, ITEM_NAME	
WHERE 条件	UNIT_COST > 10.00	
GROUP BY 列		
HAVING 条件		
ORDER BY 列		
FOR UPDATE OF 列		

終わり

選択項目を入力して、実行キーを押してください。

結果テーブル中の DISTINCT 行	N	Y=Yes, N=No
別の SELECT との UNION	N	Y=Yes, N=No
追加オプションの指定	N	Y=Yes, N=No

F3= 終了 F4= プロンプト F5= 最新表示 F6= 行挿入 F9=SUBQUERY の指定
F10= 行のコピー F12= 取り消し F14= 行削除 F15= 行分割 F24= キーの続き

最初は「SELECT ステートメントの指定 (Specify SELECT Statement)」画面上のプロンプトごとに 1 行のみ表示されますが、F6 (行の挿入) キーを使用して、画面の上部の入力域にさらに行を追加できます。SELECT 列リストにさらに列を入力しようとしている場合や、もっと長くて複雑な WHERE 条件が必要な場合に、このキーを使用できます。

- 上記のように画面に入力します。
- Enter キーを押して、SELECT ステートメントを実行します。以下の出力が表示されます。

```

                                データの表示
                                データの幅 . . . . . : 41
行の位置指定 . . . . .      桁移動 . . . . .
.....1.....2.....3.....4.
ITEM          UNIT  ITEM
NUMBER       COST  NAME
775298      225.00 Chairs, secretary
073956      20.00  Pens, black
***** データの終わり *****
                                終わり
F3= 終了   F12= 取り消し   F19= 左   F20= 右   F21= 分割

```

WHERE 文節で指定した条件とデータ値が比較された行のみ戻されます。さらに、SELECT 文節で明示的に指定した列のデータ値のみ戻されます。明示的に識別されていない列のデータ値は戻されません。

以下のように、このステートメントを「SQL ステートメントの入力」画面に入力できます。

```

SELECT ITEM_NUMBER,UNIT_COST,ITEM_NAME
FROM SAMPLECOLL.INVENTORY_LIST
WHERE UNIT_COST > 10.00

```

このコードに関する特記事項は、コーディング例に関連しています。

複数のテーブルからの情報の入手

SQL を使用して、複数のテーブルに含まれる列から情報を入手できます。この操作のことを、結合操作といいます。SQL で結合操作を指定するには、SELECT ステートメントの同じ FROM 文節に結合したいテーブルの名前を含めます。

すべての供給者と、供給される項目の項目番号および項目名のリストを表示したいとします。項目名は SUPPLIERS テーブル中にありません。項目名は INVENTORY_LIST テーブルにあります。共通列 ITEM_NUMBER を使用して、3 つの列すべてがあたかも 1 つのテーブルの列にあるかのように表示できます。

結合される複数のテーブルに同じ列名がある場合には、必ず列名はテーブル名によって修飾され、どの列が実際に参照されているか指定します。この SELECT ステートメント中に、両方のテーブル中の列名 ITEM_NUMBER を定義して、列名をテーブル名で修飾する必要があることを指定します。列の名前が違う場合は、混同しないので、修飾する必要はありません。

1. この結合を実行するには、以下の SELECT ステートメントを使用できます。「SQL ステートメントの入力」画面に直接入力するか、またはプロンプトを使用します。プロンプトを使用する場合は、両方のテーブル名とも FROM テーブルの入力行に入力する必要があります。

```

SELECT SUPPLIER_NUMBER, SAMPLECOLL.INVENTORY_LIST.ITEM_NUMBER, ITEM_NAME
FROM SAMPLECOLL.SUPPLIERS, SAMPLECOLL.INVENTORY_LIST
WHERE SAMPLECOLL.SUPPLIERS.ITEM_NUMBER
      = SAMPLECOLL.INVENTORY_LIST.ITEM_NUMBER

```

2. この同じステートメントを入力する別の方法として、関連名を使用する方法があります。関連名を使用すると、1 つのステートメント中でテーブル名の別名として使用できます。テーブル名が同じ場合は、関連名を使用しなければなりません。関連名は、FROM リスト中の個々のテーブル名の後に指定できます。前述のステートメントを以下のように書き直すことができます。

```

SELECT SUPPLIER_NUMBER, Y.ITEM_NUMBER, ITEM_NAME
FROM SAMPLECOLL.SUPPLIERS X, SAMPLECOLL.INVENTORY_LIST Y
WHERE X.ITEM_NUMBER = Y.ITEM_NUMBER

```

このコードに関する特記事項は、コーディング例に関連しています。

この例では、SAMPLECOLL.SUPPLIERS には相関名 X が指定され、SAMPLECOLL.INVENTORY_LIST には相関名 Y が指定されます。次にこの名前 X および Y を使用して、ITEM_NUMBER 列名を修飾します。

列および相関名について詳しくは、「SQL 解説書」トピックの『相関名』を参照してください。

この例を実行すると、以下の出力が戻されます。

```
          データの表示
          データの幅 . . . . . : 45
行の位置指定 . . . . .      桁移動 . . . . .
.....1.....2.....3.....4...+
SUPPLIER_NUMBER  ITEM  ITEM
                  NUMBER NAME
1234             153047 Pencils, red
1234             229740 Lined tablets
1234             303476 Paper clips
9988             153047 Pencils, red
9988             559343 Envelopes, legal
2424             153047 Pencils, red
2424             303476 Paper clips
5546             775298 Chairs, secretary
3366             303476 Paper clips
3366             073956 Pens, black
***** データの終わり *****
                                                                終わり
F3= 終了   F12= 取り消し   F19= 左   F20= 右   F21= 分割
```

注: QUERY に関する ORDER BY 文節が指定されていないので、QUERY によって戻される行の順序は違う可能性があります。

結果テーブルのデータ値は、2つのテーブル INVENTORY_LIST および SUPPLIERS に含まれているデータ値を複合したものを表します。この結果テーブルには、SUPPLIER テーブルの供給者番号と、INVENTORY_LIST テーブルの項目名が含まれます。SUPPLIER テーブルにない項目番号は、この結果テーブルにもありません。SELECT ステートメントの ORDER BY 文節を指定しないと、結果の順序は保証されません。SUPPLIER テーブルの列見出しに変更を加えていないので、SUPPLIER_NUMBER の列名が列見出しとして使用されます。

ORDER BY を使用して行の順序を保証する例を以下に示します。このステートメントは、まず SUPPLIER_NUMBER 列別に結果テーブルを配列します。SUPPLIER_NUMBER の同じ値の行は、ITEM_NUMBER 別に配列されます。

```
SELECT SUPPLIER_NUMBER,Y.ITEM_NUMBER,ITEM_NAME
FROM SAMPLECOLL.SUPPLIERS X,SAMPLECOLL.INVENTORY_LIST Y
WHERE X.ITEM_NUMBER = Y.ITEM_NUMBER
ORDER BY SUPPLIER_NUMBER,Y.ITEM_NUMBER
```

このコードに関する特記事項は、コーディング例に関連しています。

前述のステートメントを実行すると、以下の出力が作成されます。


```

          データの表示
データの幅 . . . . . : 45
桁移動 . . . . .
行の位置指定 . . . . .
.....1.....2.....3.....4...+
SUPPLIER_NUMBER  ITEM  ITEM
                   NUMBER NAME
1234             153047 Pencils, red
1234             229740 Lined tablets
1234             303476 Paper clips
2424             153047 Pencils, red
2424             303476 Paper clips
3366             073956 Pens, black
3366             303476 Paper clips
5546             775298 Chairs, secretary
9988             153047 Pencils, red
9988             559343 Envelopes, legal
***** データの終わり *****

                               終わり
F3= 終了   F12= 取り消し   F19= 左   F20= 右   F21= 分割

```

テーブル中の情報の変更

SQL UPDATE ステートメントを使用して、テーブルの一部またはすべての列のデータ値に変更を加えることができます。

1 回のステートメントの実行時に変更が加えられる行数を制限したい場合は、UPDATE ステートメントと WHERE 文節を使用します。WHERE 文節を指定しないと、指定したテーブルのすべての行に変更が加えられます。しかし、WHERE 文節を使用すると、指定した条件を満たす行のみシステムによって変更が加えられます。詳細については、SQL プログラミング (SQL Programming) トピックを参照してください。

対話式 SQL を使用して、今日の書類留めクリップの発注を増やしたいとします。

1. 項目番号 303476 の LAST_ORDER_DATE と ORDER_QUANTITY を更新するには、UPDATE と入力して F4 (プロンプト) を押します。「UPDATE ステートメントの指定 (Specify UPDATE Statement)」画面が表示されます。

```

          UPDATE ステートメントの指定
選択項目を入力して、実行キーを押してください。
テーブル . . . . . INVENTORY_LIST _____ 名前, リストは F4 キー
コレクション . . . . . SAMPLECOLL_ _____ 名前, リストは F4 キー
相関 . . . . . _____ 名前

F3= 終了   F4= プロンプト   F5= 最新表示   F12= 取り消し
F20= 名前全体の表示   F21=ステートメントの表示

```

2. 示されているように、テーブル名とスキーマ名を入力します。
3. Enter キーを押します。画面が再表示され、テーブル中の列のリストが表示されます。

UPDATE ステートメントの指定

選択項目を入力して、実行キーを押してください。

テーブル INVENTORY_LIST _____ 名前, リストは F4 キー
 コレクション SAMPLECOLL_ _____ 名前, リストは F4 キー
 相関 _____ 名前

情報を入力して、実行キーを押してください。

列	値
ITEM_NUMBER	_____
ITEM_NAME	_____
UNIT_COST	_____
QUANTITY_ON_HAND	_____
LAST_ORDER_DATE	CURRENT DATE _____
ORDER_QUANTITY	50 _____

終わり

F3= 終了 F4= プロンプト F5= 最新表示 F6= 行挿入 F10= 行のコピー
 F11= タイプの表示 F12= 取り消し F14= 行削除 F24= キーの続き

- 「LAST_ORDER_DATE」フィールドに CURRENT DATE を指定して、値を今日の日付に変更します。
- 示されているように、更新値を入力します。
- Enter キーを押して、WHERE 条件を指定できる画面を表示します。WHERE 条件を指定しないと、前述の画面の値を使用してテーブル中のすべての行が更新されます。

UPDATE ステートメントの指定

WHERE 条件を入力して、実行キーを押してください。リストの表示は、F4 キーを押してください。

ITEM_NUMBER = '303476' _____

終わり

選択項目を入力して、実行キーを押してください。

WITH 分離レベル 1
 1= 現行レベル, 2=NC (NONE)
 3=UR (CHG), 4=CS, 5=RS (ALL)
 6=RR

F3= 終了 F4= プロンプト F5= 最新表示 F6= 行挿入 F9=SUBQUERY の指定
 F10= 行のコピー F12= 取り消し F14= 行削除 F15= 行分割 F24= キーの続き

- WHERE 条件フィールドに ITEM_NUMBER = '303476' と入力します。
- Enter キーを押して、テーブルに対する更新を実行します。機能が完了したことを示すメッセージが表示されます。

SELECT ステートメントを実行してテーブルからすべての行を入手する (SELECT * FROM SAMPLECOLL.INVENTORY_LIST) と、以下の結果が戻されます。

データの表示

データの幅 : 71

桁移動

.....1.....2.....3.....4.....5.....6.....7.

ITEM NUMBER	ITEM NAME	UNIT COST	QUANTITY ON HAND	LAST ORDER DATE	NUMBER ORDERED
153047	Pencils, red	10.00	25	-	20
229740	Lined tablets	1.50	120	-	20
544931	***UNKNOWN***	5.00	-	-	20
303476	Paper clips	2.00	100	04/05/30	50
559343	Envelopes, legal	3.00	500	-	20
291124	Envelopes, standard	.00	-	-	20
775298	Chairs, secretary	225.00	6	-	20
073956	Pens, black	20.00	25	-	20
***** データの終わり *****					

終わり

F3= 終了 F12= 取り消し F19= 左 F20= 右 F21= 分割

「書類留めクリップ (Paper clips)」の項目のみ変更が加えられています。「LAST_ORDER_DATE」は現在日付に変更されています。常にこの日付は、更新が実行された日付になります。「NUMBER_ORDERED」には更新値が示されています。

以下のように、このステートメントを「SQL ステートメントの入力」画面に入力できます。

```
UPDATE SAMPLECOLL.INVENTORY_LIST
SET LAST_ORDER_DATE = CURRENT DATE,
ORDER_QUANTITY = 50
WHERE ITEM_NUMBER = '303476'
```

このコードに関する特記事項は、コーディング例に関連しています。

テーブルからの情報の削除

SQL DELETE ステートメントを使用して、テーブルからデータを削除できます。テーブルの行全体に必要な情報が含まれなくなった場合には行全体を削除できますし、DELETE ステートメントと WHERE 文節を使用して 1 回のステートメントの実行時に削除する行を識別することもできます。詳しくは、「SQL 解説書」トピックの DELETE を参照してください。

テーブル中の QUANTITY_ON_HAND 列がヌル値の行をすべて除去したいとします。

- 「SQL ステートメントの入力」画面で、以下のステートメントを入力します。

```
DELETE
FROM SAMPLECOLL.INVENTORY_LIST
WHERE QUANTITY_ON_HAND IS NULL
```

ヌル値の列を検査するには、IS NULL 比較を使用します。

このコードに関する特記事項は、コーディング例に関連しています。

- 削除の完了後に、別の SELECT ステートメントを実行します。この結果、以下のテーブルになります。

データの表示						
データの幅						71
桁移動						
1	2	3	4	5	6	7
ITEM NUMBER	ITEM NAME	UNIT COST	QUANTITY ON HAND	LAST ORDER DATE	NUMBER ORDERED	
153047	Pencils, red	10.00	25	-	20	
229740	Lined tablets	1.50	120	-	20	
303476	Paper clips	2.00	100	04/05/30	50	
559343	Envelopes, legal	3.00	500	-	20	
775298	Chairs, secretary	225.00	6	-	20	
073956	Pens, black	20.00	25	-	20	
***** データの終わり *****						
						終わり
F3= 終了 F12= 取り消し F19= 左 F20= 右 F21= 分割						

QUANTITY_ON_HAND がヌル値の行が削除されています。

視点の作成および使用

1 つのテーブルにすべての必要な情報が含まれていないことに気付くことがあります。 テーブル中のデータの部分のみに対するユーザー・アクセス権を与えたいこともあります。 視点を使用すると、テーブルをサブセット化して、必要なデータのみ扱うことができます。 視点を使用すると複雑さを軽減でき、同時にアクセスを制約できます。

SQL CREATE VIEW ステートメントを使用して視点を作成できます。 CREATE VIEW ステートメントを使用して、テーブルに対する視点を定義することは、必要な列と行のみを含む新しいテーブルを作成することに似ています。アプリケーションで視点を使用すると、視点に組み込まれていないテーブルの行や列にはアクセスできません。しかし、SQL WITH CHECK OPTION を使用していない場合は、引き続き視点を使用して選択基準と一致しない行を挿入できます。 WITH CHECK OPTION の使用法についての詳細は、「SQL プログラミング (SQL Programming)」トピックの視点に対する WITH CHECK OPTION (WITH CHECK OPTION on a View) を参照してください。

視点を作成するには、その視点のベースとなるテーブルや物理ファイルに対する適切な権限がなければなりません。必要な権限のリストについては、「SQL 解説書」トピック中の CREATE VIEW ステートメントを参照してください。

視点定義中で列名を指定しないと、その視点のベースとなるテーブルの列名と同じ列名になります。

視点とテーブルの列や行の数が違う場合でも、視点を利用してテーブルに変更を加えることができます。INSERT の場合、テーブルにあって視点にない列はデフォルト値でなければなりません。

視点のデータが 1 つ以上のテーブルに完全に従属している場合でも、視点をテーブルであるかのように使用できます。視点には独自のデータがないので、データ用のストレージは必要ありません。視点はストレージ中にあるテーブルから派生するので、視点のデータを更新する際には、実際にはテーブル中のデータ更新することになります。したがって、視点が従属するテーブルが更新されるつど、視点は自動的に最新の状態に保たれます。

1 つのテーブルに対する視点を作成する

次の例は、1 つのテーブルに基づく視点を作成する方法を示しています。この視点は INVENTORY_LIST テーブルに基づいています。このテーブルには 6 つの列がありますが、視点ではこれらの列のうち ITEM_NUMBER、LAST_ORDER_DATE、および QUANTITY_ON_HAND の 3 つのみ使用します。列

は、SELECT 文節中の順序で視点中でも表示されます。視点には、最近 2 週間に注文された項目の行のみ含まれます。CREATE VIEW ステートメントは、以下のようになります。

1. 以下のコマンドを使用して、視点を作成します。

```
CREATE VIEW SAMPLECOLL.RECENT_ORDERS AS
SELECT ITEM_NUMBER, LAST_ORDER_DATE, QUANTITY_ON_HAND
FROM SAMPLECOLL.INVENTORY_LIST
WHERE LAST_ORDER_DATE > CURRENT DATE - 14 DAYS
```

上記の列では、視点名の後に列のリストがないので、視点中の列はテーブル中の列と同じ名前になります。視点が従属しているテーブルのあるスキーマ中に視点も作成する必要はありません。どのスキーマまたはライブラリーでも使用できます。

このコードに関する特記事項は、コーディング例に関連しています。

2. このステートメントを実行します。

```
SELECT *FROM SAMPLECOLL.RECENT_ORDERS
```

結果は次のようになります。

データの表示

データの幅 :	26
桁移動 :	
行の位置指定	
....+.1....+.2....+	
ITEM LAST QUANTITY	
NUMBER ORDER ON	
DATE HAND	
303476 04/01/26 100	
***** データの終わり *****	終わり

F3= 終了 F12= 取り消し F19= 左 F20= 右 F21= 分割

現在日付に更新した行のみ視点によって選択されています。テーブル中の他のすべての日付は依然としてヌル値なので、戻されません。

複数のテーブルからデータを結合して視点を作成する

FROM 文節で複数のテーブルを指名して、複数のテーブルのデータを結合して視点を作成できます。以下の例で INVENTORY_LIST テーブルには、ITEM_NUMBER という項目番号の列と、UNIT_COST という項目のコストに関する列が含まれています。これらの列は、SUPPLIERS テーブルの ITEM_NUMBER 列および SUPPLIER_COST 列と結合されています。WHERE 文節を使用して、戻される行数を制限します。視点には、現行の単位コストより安いコストで項目を供給できる供給者の項目番号のみ含まれます。

1. 以下のコマンドを使用して、視点を作成します。

```
CREATE VIEW SAMPLECOLL.LOWER_COST AS
SELECT SUPPLIER_NUMBER, A.ITEM_NUMBER,UNIT_COST, SUPPLIER_COST
FROM SAMPLECOLL.INVENTORY_LIST A, SAMPLECOLL.SUPPLIERS B
WHERE A.ITEM_NUMBER = B.ITEM_NUMBER
AND UNIT_COST > SUPPLIER_COST
```

このコードに関する特記事項は、コーディング例に関連しています。

2. このステートメントを実行します。

```
SELECT *FROM SAMPLECOLL.LOWER_COST
```

結果は次のようになります。

```

          データの表示
          データの幅 . . . . . : 51
  行の位置指定 . . . . .      桁移動 . . . . .
  ....+....1....+....2....+....3....+....4....+....5.
  SUPPLIER_NUMBER  ITEM          UNIT  SUPPLIER_COST
                   NUMBER       COST
    1234           229740        1.50      1.00
    9988           153047        10.00     8.00
    2424           153047        10.00     9.00
    3366           303476        2.00      1.50
    3366           073956        20.00     17.00
  ***** データの終わり *****
                                           終わり
  F3= 終了   F12= 取り消し   F19= 左   F20= 右   F21= 分割

```

注: QUERY に関する ORDER BY 文節が指定されていないので、QUERY によって戻される行の順序は違う可能性があります。

この視点には、供給者コストが単位コストより小さい行のみ表示されます。

対話式 SQL の使用法について詳しくは、「SQL プログラミング (SQL Programming)」トピックの対話式 SQL の使用 (Using Interactive SQL) を参照してください。

SQL 対従来のファイル・アクセス用語

DB2 UDB for iSeries は、複数のアクセス方法論をサポートしています。それらの相違点、およびどのような状況で 1 つの方法が組織または特定のプロジェクトで利点を提供するかを理解する必要があります。

DB2 Universal Database for iSeries データベース・テーブルおよびデータを処理するために次の 2 つのアクセス方式を提供します。

- 構造化照会言語 (SQL) は、データベースの開発とアクセスについての iSeries の戦略的な方向を表す。
- システム・ファイル・アクセス方式 (しばしば、「システム」または従来のファイル・アクセス方式と呼ばれ、ときによっては既存のファイル・アクセス方式と呼ばれる) は、V3R7 より前のカスタマーにはおなじみのものであり、実際、多くのカスタマーのデータベースは、システム・レベルのファイル・アクセス方式におけるかなりの開発投資を反映しています。

DB2 UDB for iSeries へのインターフェースとして、DB2 Query Manager and SQL Development Kit for iSeries は、対話型 Query および報告書作成インターフェースが追加され、さらに高水準プログラム言語での SQL アプリケーションを作成する際に役立つプリコンパイラーおよびツールが追加されます。業界標準の構造化照会言語 (SQL) に適合させ、OS/400 に SQL をインプリメントすることにより、iSeries データの定義、処理、照会、および iSeries データへのアクセス制御を行うことができます。これは、OS/400 のファイルおよび SQL テーブルの両方で等しく機能します。

SQL 対従来のファイル・アクセス用語

SQL 用語	従来のファイル・アクセス用語
コレクション。ライブラリー、ジャーナル、ジャーナル・レシーバー、SQL カタログ、およびオプション・データ・ディクショナリーから構成されます。コレクションは関連オブジェクトをグループ化し、ユーザーが名前でもオブジェクトを検索できるようにします。	ライブラリー。関連のあるオブジェクトをグループ化し、ユーザーがオブジェクトを名前で見つけることができますようにします。
テーブル。列と行のセット。	物理ファイル。レコードのセット。

SQL 用語	従来のファイル・アクセス用語
行。 一連の列を含むテーブルの水平部分。	レコード。 フィールドのセット。
列。 データ・タイプのテーブルの垂直部分。	フィールド。 1 データ・タイプの 1 バイトまたは複数バイトの関連情報。
視点。 1 つまたは複数のテーブルの列と行のサブセット。	論理ファイル。 32 以内の物理ファイルのフィールドかレコード (またはその両方) のサブセット。
索引。 論理的に昇順または降順に配置されたテーブルの列内にあるデータのコレクション。	論理ファイルの 1 タイプ。
パッケージ。 アプリケーション・サーバーに使用する SQL ステートメントの制御構造が入っているオブジェクト。	SQL パッケージ。 SQL という用語と同じ意味。
カタログ。 テーブル、パッケージ、視点、索引、および制約に関する情報が入っているテーブルと視点のセット。	類似オブジェクトなし。 ただし、ファイル記述表示 (DSPFD) およびファイル・フィールド記述表示 (DSPFFD) コマンドは、SQL カタログの照会が提供するのと同じ情報の一部を提供します。

コードに関する特記事項

本書には、プログラミングの例が含まれています。

IBM は、お客様に、すべてのプログラム・コードのサンプルを使用することができる非独占的な著作使用権を許諾します。お客様は、このサンプル・コードから、お客様独自の特別のニーズに合わせた類似のプログラムを作成することができます。

すべてのサンプル・コードは、例として示す目的でのみ、IBM により提供されます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。

ここに含まれるすべてのプログラムは、現存するままの状態を提供され、いかなる保証も適用されません。商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任の保証の適用も一切ありません。

資料に関するご使用条件

お客様がダウンロードされる資料につきましては、以下の条件にお客様が同意されることを条件にその使用が認められます。

個人使用: これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、非商業的な個人による使用目的に限り複製することができます。ただし、IBM の明示的な承諾をえずに、これらの資料またはその一部について、二次的著作物を作成したり、配布 (頒布、送信を含む) または表示 (上映を含む) することはできません。

商業的使用: これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、お客様の企業内に限り、複製、配布、および表示することができます。ただし、IBM の明示的な承諾をえずにこれらの資料の二次的著作物を作成したり、お客様の企業外で資料またはその一部を複製、配布、または表示することはできません。

ここで明示的に許可されているもの以外に、資料や資料内に含まれる情報、データ、ソフトウェア、またはその他の知的所有権に対するいかなる許可、ライセンス、または権利を明示的にも黙示的にも付与するものではありません。

資料の使用が IBM の利益を損なうと判断された場合や、上記の条件が適切に守られていないと判断された場合、IBM はいつでも自らの判断により、ここで与えた許可を撤回できるものとさせていただきます。

お客様がこの情報をダウンロード、輸出、または再輸出する際には、米国のすべての輸出入関連法規を含む、すべての関連法規を遵守するものとします。IBM は、これらの資料の内容についていかなる保証もしません。これらの資料は、特定物として現存するままの状態を提供され、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任なしで提供されます。

これらの資料の著作権はすべて、IBM Corporation に帰属しています。

お客様が、このサイトから資料をダウンロードまたは印刷することにより、これらの条件に同意されたものとさせていただきます。

付録. 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

- | 〒106-0032
- | 東京都港区六本木 3-2-31
- | IBM World Trade Asia Corporation
- | Licensing

以下の保証は、国または地域の法律に沿わない場合は、適用されません。 IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとしします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

- | IBM は、お客様が提供するいかなる情報も、お客様に対してなら義務も負うことのない、自ら適切と信
- | ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

- | IBM Corporation
- | Software Interoperability Coordinator, Department 49XA
- | 3605 Highway 52 N
- | Rochester, MN 55901
- | U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。お客様は、IBM のアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (西暦年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。 © Copyright IBM Corp. _年を入れる_. All rights reserved.

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

プログラミング・インターフェース情報

- 1 本書には、プログラムを作成するユーザーが DB2 UDB for iSeries のサービスを使用するためのプログラミング・インターフェースが記述されています。

商標

以下は、IBM Corporation の商標です。

DB2
DB2 Universal Database
IBM
iSeries
OS/400

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。



Printed in Japan