

IBM i 2024

IBM i コンテンツ (2024年10月版)

IBM i アプリ開発要員充足の近道!
IBM i SQL入門 (SQL機能は無償でご利用可能)

日本アイ・ビー・エム株式会社
テクノロジー事業本部
IBM Powerテクニカルセールス



IBM i のSQL入門

IBM i は、2種類のデータベースのインターフェースを備えています（オープン系で使用される汎用SQLと、IBM i に特化した DDS）。

昔からのIBM i ユーザーにはDDSによるアクセスの方を好んで使っていましたが、昨今ではIT全般でのSQLの普及もあり、「SQLで実行できることはSQLで」という流れが、IBM i でも定着し始めています。

汎用的なSQLを使ったIBM i 用アプリケーション開発ができれば、**貴社のIBM i 要員の充足の観点でもお役に立つことと存じます**。**ACSのSQLパネルから、SQLのみを使用したアプリ開発もできますし、もしくはRPGIV、FF-RPG、COBOL、Java、Python、PHP、などIT部門の各員がお得意の言語の中でもご利用いただけます**。

従来は有償であった、5250画面での**対話式SQL (Db2 Query Manager & SQL Dev Kit (5770-ST1)) は、2023年10月に無償化**されています（ライセンスキーが不要になるPTFを提供しています）ので、**SWMAもしくはサブスクリプションでIBM iをご使用**されている**すべてのユーザー様はSQLの全機能をご使用いただけます**。今月のコンテンツでは、IBM i ユーザー向けのSQLの入門をご紹介します。次月からも、引き続き、SQL中級編・SQL上級編をお届けする予定です。

目次

1. SQLとは
2. SQLを使うメリット
3. IBM i でのSQL概説
4. IBM i の主要なSQLコマンド
5. SQLを使ってみよう
6. 補足情報

1. SQL (Structured Query Language) とは

構造化照会言語 (SQL) は、リレーショナル・データベース内のデータを定義および操作するための標準化言語です。

リレーショナル・データベースを操作するための言語

- DDL (Data Definition Language): データの構造を定義
データベース (新規テーブル) の新規作成 (CREATE)、変更 (ALTER)、削除 (DROP) 等
- DML (Data Manipulation Language): データの検索と更新
SELECT、INSERT、UPDATE、DELETE等

SQLの実行形式

- 静的SQL: アプリケーション・プログラム内部に組み込まれたSQL
実行されるSQLは常に固定
- 動的SQL: アプリケーション実行時に構成されるSQL
静的SQLよりも柔軟な処理が可能
- 拡張動的SQL: SQLパッケージを使用する動的SQL
IBM i では、IBM I Access ODBC および OLE DB Provider はSQL パッケージをサポート。
- 対話式SQL:
IBM i ではCLのSTRSQL コマンド、またはACSの SQL スクリプト実行が可能

解説：

- DDL (Data Definition Language) はデータ定義言語と呼ばれ、テーブルや索引、シーケンスなどのデータベースオブジェクトを定義する言語です。データベースやテーブルを新規に作成 (CREATE) する場合や、変更 (ALTER)、削除 (DROP) する際に使用します。
- DML (Data Manipulation Language) はデータ操作言語と呼ばれ、データベースを操作し、格納されているデータの検索や削除などを行うための言語です。命令文は、データの検索 (SELECT)、更新 (UPDATE)、挿入 (INSERT)、削除 (DELETE) があります。
- 静的 SQL は、コンパイル時に作業の一部を行います。アクセス・プランがコンパイル時に生成または更新され、プログラム・オブジェクト自体の中に保存されます。コンパイル時の最適化と実行時の効率性により、高速なパフォーマンスを提供します。
- 動的 SQL を使用すると、プログラムの実行時に SQL ステートメントを定義して、実行させることができます。動的 SQL を使用するアプリケーションは、文字ストリングの形で SQL ステートメントを入力として受け取るか、または SQL ステートメントを作成します。
- 拡張動的 SQL は、動的 SQL 機能と同様に、ステートメントを準備し、記述し、実行することができます。SQL パッケージ中に準備された SQL ステートメントは、そのパッケージまたはステートメントが明示的に削除されるまで存続します。

2. SQL を使うメリット

- ✓ SQLを利用するアプリケーションの増加
 - ISVパッケージソフト
 - ODBC接続、JDBC接続、.NET接続・・・

- ✓ 開発生産性の向上
 - コード数の減少
 - テストが容易（データの検索、更新、削除が簡単）
 - 複数のプログラミング言語から利用可能（Java、RPG、PHP、COBOL・・・）
 - パフォーマンス
変化（データ量・システム資源など）に応じて動的にプランを変更可能

- ✓ SQLの特徴
 - 取り出したいデータの要件（欲しい結果）を記述
データ取り出しの手順を記述しない。プログラムは要件を手順にして、その手順を記述
 - データの増減にあわせて手順を自動的に最適化

3. IBM i でのSQL概説

(1) 特長

- ✓ IBM i のSQL は、OSの組み込み機能 (Db2 for IBM i) で、最新のSQL標準に準拠
- ✓ IBM i のSQLツールは、無償で利用可能
(従来有償だったSQL開発キット (5770ST1)は無償化された)
- ✓ 5250からの対話型SQL以外に、ACS (Access Client Solutions)でのデータベース管理機能 (スキーマ操作、SQLスクリプト発行、SQL Performance Center) が使える
- ✓ ホストプログラム (RPG,COBOL…)以外に、ODBC , JDBC , OLE DB, .NET , DRDA などのデータアクセスインターフェイスをサポート
- ✓ 保管・復元は、通常の業務と同じ、SAV,RSTコマンドが使える
- ✓ 従来型のDDSで作成した物理ファイルに対してもSQLが使える

解説：

- IBM iのSQLは最新のSQL標準に準拠している。詳細は下記のURL参照してください。
<https://www.ibm.com/docs/ja/i/7.5?topic=reference-standards-compliance>
- 従来有償であった、5250画面での対話式SQL (Db2 Query Manager & SQL Dev Kit(5770-ST1))は2023年10月に無償化されました (ライセンスキーが不要になるPTFを提供)。
<https://www.ibm.com/support/pages/products-are-no-longer-listed-wrklicinf-show-dsplickey-or-call-qsfwinv>
- Db2 for IBM iのデータベースのファイルの保管/復元は、IBM iの通常の保管(SAVXXX)/復元(RSTXXX)コマンドが使える
- DDSで作成した、物理ファイルに対しても、SQLコマンドが利用できる (論理ファイルにも利用できるが、論理ファイルの設定は適用されない。ビューや、索引の作成が必要)

3. IBM i でのSQL概説

(2) SQL と従来のファイル・アクセス用語の比較

IBM i には、データベース表およびデータを操作するためのアクセス方式として、SQL とシステム・ファイル・アクセス方式の 2 つが用意されている。用語に違いをまとめました。

SQL用語	従来のファイル・アクセス用語	補足
スキーマ	ライブラリー	スキーマ名 は、SQL オブジェクト名 (表、ビュー、索引、トリガーなど) の修飾子として使用します。スキーマは、コレクションまたはライブラリーとも呼ばれる
テーブル	物理ファイル	データが入っているファイル。テーブルは列と行のセット
行	レコード	一連のフィールドを含む表の水平部分
列	フィールド	同じデータ属性を持った表の垂直部分
ビュー	論理ファイル	1つまたは複数の物理ファイルのフィールドまたはレコードのサブセット (データ自体は内部に持たない)
索引	論理ファイル	キー付けされた論理ファイル
パッケージ	SQLパッケージ	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
カタログ	なし	テーブル、パッケージ、ビュー、インデックス、制約の情報を含むテーブルとビュー

解説：

- (IBM iでの) スキーマは、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、およびオプションのデータ・ディクショナリーで構成される関連オブジェクトのグループのこと。スキーマを使用すると、ユーザーが名前オブジェクトを検索できます。スキーマのもう 1 つの名前は「コレクション」です。
- その他のIBM i のSQL利用の考慮点として下記があります。
 - ✓ 従来の物理データベースではジャーナルは必須ではない従来手法でライブラリー、物理ファイルを作成した場合、ジャーナルは自動的に作成されない
 - ✓ 従来の手法ではオブジェクト名の長さに制限がある
SQLで長い名称のオブジェクトを作成した場合は、制限文字数内で自動的に短縮名がアサインされ、実装にはその短縮名が使われる
 - ✓ SQLオブジェクトの名称に漢字（全角文字）は使えない

解説：

- ・対話式 SQL で SQL ステートメントを直接使用することにより、スキーマ、テーブル、およびビューを作成し、操作することができます。
- ・①の5250画面でのSTRSQLコマンドの操作には、無償になったライセンスプログラム (5770-ST1)が導入されている必要があります (GO LICPGMで下記のライセンスが導入されているか確認してください)。

ライセンス- プログラム	導入 状況	記述
5770ST1	*COMPATIBLE	DB2 QUERY MGR AND SQL DEVKIT

- ・②のACSの5250画面の上部にある、「アクション」→「SQLスクリプトの実行」でSQLコマンドを対話式に入力できます (ACSのSQLスクリプトの実行にはIBM i上のライセンスは何も必要ありません)。



4. IBM i の主要なSQLコマンド

(1) スキーマの作成

スキーマ(コレクションとも言う)とは、テーブル、ビュー、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。スキーマを作成するには、CREATE SCHEMA ステートメントを使用。

➤ CREATE SCHEMA <スキーマ名> : 作成するスキーマ名を指定

(例) SAMPLE という名前のスキーマを作成

```
CREATE SCHEMA SAMPLE
```

✓ 作成されるオブジェクト

- ライブラリー

スキーマ名でライブラリーが作成され、関連オブジェクトを論理的にグループ化

- カタログ

そのスキーマの表、ビュー、索引、およびパッケージの記述が入る

カタログは一連のビューで構成

- ジャーナルおよびジャーナル・レシーバー

ジャーナル QSQJRN とジャーナル・レシーバー QSQJRN0001 がスキーマ内に作成される
スキーマ内に作成されるすべての表への変更を記録するのに使用される

解説：

- まずは、表、ビュー、索引、およびパッケージが配置される基本オブジェクトである、「スキーマ」を作成します。スキーマ作成により、同名のライブラリーが作成されます。その中には、下記のような、カタログ、ジャーナル、ジャーナルレシーバーが作成されます。

```

PDM を使用したオブジェクトの処理                                POWERSG
ライブラリー . . . SAMPLE                                位置指定 . . . . .
                                                           タイプの位置指定 . . . . .

オプションを入力して、実行キーを押してください。
2= 変更      3= コピー      4= 削除      5= 表示      7= 名前の変更
8= 記述の表示  9= 保管      10= 復元     11= 移動 . . .

OPT オブジェクト   タイプ   属性   テキスト
---
--- QSQJRN0001   *JRNRCV   コレクション - SQL によって作成
--- QSQJRN       *JRN      コレクション - SQL によって作成
--- SYSCHKCST   *FILE     LF     SQL カタログ・ビュー
--- SYSCOLUMNS *FILE     LF     SQL カタログ・ビュー
--- SYSCST      *FILE     LF     SQL カタログ・ビュー
--- SYSCSTCOL   *FILE     LF     SQL カタログ・ビュー
--- SYSCSTDEP   *FILE     LF     SQL カタログ・ビュー
--- SYSFIELDS   *FILE     LF     SQL カタログ・ビュー

パラメーターまたはコマンド
==>

```

4. IBM i の主要なSQLコマンド

(2) テーブルの作成

テーブルとは、基本的なデータベース・オブジェクトであり、情報を保管するために使用されます。テーブルを作成した後、列の定義、索引の作成、トリガーや制約の追加を行うことができます。CREATE TABLE ステートメントを使用して、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実施します。

- CREATE TABLE <表名> (<列名1> <データ型>, <列名2> <データ型>, . . .)
 - : 作成する表名、列名、列のデータ型を指定
 - NULL値を許さない列には NOT NULL を指定
- PRIMARY KEY <列名>
 - : 主キーとなる列名を指定

(例) INVENTORY という名前で、次の列を持つ表を作成

PARTNO(部品番号): 1 から 9999 の間の整数で、ヌルは許されない

DESCR(記述): 長さ 0 から 24 の文字

QONHAND(在庫数量): 0 から 100000 の間の整数

基本キーは PARTNO

```
CREATE TABLE INVENTORY (PARTNO SMALLINT NOT NULL,  
DESCR VARCHAR(24), QONHAND INT, PRIMARY KEY(PARTNO))
```

解説 1 :

- ・ テーブルの列でサポートされるデータ型について、主要なものを下記に記載します。
詳細はSQL解説書：https://www.ibm.com/docs/ja/ssw_ibm_i_75/pdf/rbafzpdf.pdf
のP.1661からのSQL制約を参照

区分		データ型		説明
符号付数値	浮動小数点	単精度	REAL	およそ1.17549436 X10 ⁻³⁸ から3.40282356 X10 ³⁸ まで
		倍精度	DOUBLE	およそ2.2250738585072014 X10 ⁻³⁰⁸ から1.7976931348623158 X10 ³⁰⁸ まで
	2進整数	16ビット	SMALLINT	-32,768から+32,767まで
		32ビット	INTEGER	-2,147,483,648から +2,147,483,647まで
		64ビット	BIGINT	-9,223,372,036,854,775,808から+9,223,372,036,854,775,807まで
	10進数	ゾーン	NUMERIC	最大63桁、-10 ⁶³ +1から10 ⁶³ -1まで
パック		DECIMAL	最大63桁、-10 ⁶³ +1から10 ⁶³ -1まで	
区分		データ型		説明
ストリング	文字	固定長	CHAR	1から32,766バイトまでの長さ
		可変長	VARCHAR	1から32,740バイトまでの長さ
			CLOB	1から2,147,483,647バイトまでの長さ
	2進	固定長	BINARY	1から32,766バイトまでの長さ
		可変長	VARBINARY	1から32,740バイトまでの長さ
			BLOB	1から2,147,483,647バイトまでの長さ
	グラフィック	固定長	GRAPHIC	1から16,386文字までの長さ
		可変長	VARGRAPHIC	1から16,370文字までの長さ
DBCLOB			1から1,073,741,823文字までの長さ	

解説 2 :

- ・ テーブルの列で、サポートされるデータ型について主要なものを下記に記載します (続き)。

区分		データ型	説明
日付/時刻	日付	DATE	日付のみ、6,8,10バイトのいずれか
	時刻	TIME	時、分、秒、8バイト
	タイムスタンプ	TIMESTAMP	年、月、日、時、分、秒、マイクロ秒、26バイト
行ID		ROWID	40バイト、可変長
外部データ		DATALINK	URL、スキーム、ファイルサーバー名、ファイルパス、アクセス制御トークン、コメント

- ・ EBCDICでの2バイト文字の取り扱い
2バイト文字はシフトアウト (X'0E') とシフトイン (X'0F') で囲まれる
- ・ GRAPHIC、VARGRAPHIC、DBCLOB
2バイト文字専用のカラム (シフトアウト、シフトインで囲まれない、1バイト文字は入らない)
- ・ UNICODEエンコーディングも可能
CHAR, VARCHAR : CCSID(1208)指定でUTF-8エンコーディング
GRAPHIC,VARGRAPHIC : CCSID(13488)指定でUCS-2、CCSID(1200)指定でUTF-16 カラム単位の指定
- ・ 日付/時刻タイプ
文字列表記がINSERT、条件指定で使える
実際の文字列表記はIBM iの設定値で異なる

4. IBM i の主要なSQLコマンド

(3) ビューの作成

データベース内に、必要な情報がすべて含まれている単一のテーブルがない場合があります。また、ユーザーにテーブル内のデータの一部のみへのアクセス権を付与したい場合もあります。ビューを使用すると、テーブルを分割して、必要なデータのみを処理することができます。ビューによって複雑さが軽減されると同時にアクセスを制限できます。ビューは、CREATE VIEW ステートメントを使用して作成できます。

- CREATE VIEW <ビュー名> : 作成するビュー名を指定
AS <SELECT文> : 表から抽出されるデータを指定

(例) すべての管理者の姓と所属部門だけを選択するビューを作成

```
CREATE VIEW EMP_MANAGERS AS SELECT LASTNAME, WORKDEPT  
FROM EMPLOYEE WHERE JOB = 'MANAGER'
```

解説：

- ・ビューを作成するには、ビューの基盤となるテーブルまたは物理ファイルに対する適切な権限が必要です。
- ・ビュー定義で列名を指定しなかった場合、列名は、ビューの基盤となるテーブルの列名と同じになります。
- ・ビューが、1つ以上のテーブルのデータに完全に依存している場合でも、1つのテーブルであるかのように使用することができます。ビューにはデータが含まれていないので、データ用のストレージは必要ありません。

4. IBM i の主要なSQLコマンド

(4) 索引 (インデックス) の作成

索引を使用して、データのソートと選択が可能です。また、索引を作成すると、データをより速く取り出すことができ、照会のパフォーマンスが向上します。

- CREATE INDEX <索引名> : 作成する索引名を指定
ON <表名> (<列名1>, <列名2>, . . .) : 索引を作成する表と列を指定

(例1) EMPLOYEE 表の LASTNAME 列に対する索引 (INX1) を作成
`CREATE INDEX INX1 ON EMPLOYEE (LASTNAME)`

(例2) EMPLOYEE 表に JOB_BY_DPT という名前の索引を作成。
索引項目は、各部門 (WORKDEPT) ごとにジョブ・タイトル (JOB) にしたがって昇順にする
`CREATE INDEX JOB_BY_DPT ON EMPLOYEE (WORKDEPT, JOB)`

解説：

- Db2® for i には、表にアクセスするための 2 つの基本的な方法 (表スキャンと索引ベースの検索) が用意されています。 テーブル行の 20% 未満が選択されるとき、索引による検索は通常、テーブル走査より効果的です。
- 索引を作成するメリット
 - パフォーマンス向上への貢献：テーブルの検索速度の向上
統計情報としての利用 (オプティマイザーの判断材料)
 - システム資源使用率 (CPU) の低減
オプティマイザーがインデックスの作成を必要と判断した場合、一時インデックスが作成され、CPU資源等を消費する
- 索引を作成する際の考慮点
 - メンテナンスコストの発生：データの挿入、削除、更新時にオーバーヘッドが発生
ただしハイスペックの最新マシンでの影響は少ない ⇒ 不要なインデックスは削除すること
- 索引作成の方法とパフォーマンスの最適化については、下記が詳しいです。
<https://www.ibm.com/docs/ja/i/7.5?topic=optimization-creating-index-strategy>

4. IBM i の主要なSQLコマンド

(5) SELECTによるデータ照会 の基本編

SELECT ステートメントは、データ照会機能を提供します。SELECT ステートメントを使用すると、特定の行を取り出したり、特定の方法でデータを取り出したりできます。

SELECT

項目

FROM

ファイル

以下、オプション

WHERE

条件

GROUP BY

グルーピング項目

ORDER BY

読み込み順

項目 … 読みたい項目名
(*指定は全項目)

ファイル … 読みたいファイル名
(ライブラリ名.ファイル名)

条件 … 検索条件

グルーピング項目 … グルーピングし
たい項目名

読み込み順 … 読み込み順の項目名

解説1 :

- SELECT <列名> : 取り出す列名を指定または、 * : 全ての列を指定
FROM <表名> : 取り出し元のテーブル名の指定
- WHERE <条件> : 検索条件を指定
 - 主な検索条件
 - 比較 : =, <>, <, >, <=, >=
 - 範囲 : BETWEEN
 - 列の列挙 : IN
 - 文字列のパターン : LIKE
- GROUP BY <列名> : グループ処理の対象となる列を指定
HAVING <条件> : グループの選択条件を指定
- ORDER BY <列名> ASC/DESC : 並び替える列名を指定 昇順(ASC)、または降順(DESC)の指定
(省略時値はASC)
- SELECT ステートメントを使用したデータの検索の詳細については下記を参照してください。
<https://www.ibm.com/docs/ja/i/7.5?topic=language-retrieving-data-using-select-statement>

解説2：

・SELECT 構文の例

-単純なSELECT

EMPLOYEE表の全ての行の特定の列を選択

```
SELECT EMPNO, LASTNAME, JOB FROM EMPLOYEE
```

-WHERE

部門 (WORKDEPT) が'A00'に等しく、入社年月日が1965-01-01以降の全ての列を選択

```
SELECT * FROM EMPLOYEE  
WHERE WORKDEPT = 'A00' AND HIREDATE >= '1965-01-01'
```

-GROUP BY句によるグループ処理

部門(WORKDEPT)にグループ分けして、部門ごとに部門(WORKDEPT)、行数、給与(SALARY)の合計、平均、最大値、最小値を取り出す

```
SELECT WORKDEPT, COUNT(*), SUM(SALARY), AVG(SALARY), MAX(SALARY), MIN(SALARY) FROM  
EMPLOYEE GROUP BY WORKDEPT
```

-ORDER BY句によるソート

部門(WORKDEPT)の昇順、ラストネーム(LASTNAME)の昇順にソート

```
SELECT WORKDEPT, LASTNAME FROM EMPLOYEE  
ORDER BY WORKDEPT, LASTNAME
```

4. IBM i の主要なSQLコマンド

(5) SELECTによるデータ照会 の応用編 : 結合

結合 (UNION) 演算子による複数テーブルからのデータの取り出し

- ▶ <SELECT文> UNION <SELECT文>
二つ以上のテーブルから値を組み合わせて結果のテーブルを作成する

(例) 「部門 D11 の社員」と「プロジェクト MA2112、MA2113、および AD3111 が割り当てられている社員」の結合したリストを取り出す

```
SELECT EMPNO FROM EMPLOYEE WHERE WORKDEPT = 'D11'  
UNION  
SELECT EMPNO FROM EMPPROJECT WHERE PROJNO = 'MA2112' OR  
PROJNO = 'MA2113' OR PROJNO = 'AD3111'
```

解説：

- UNION キーワードを使用すると、2 つ以上の副選択を結合して全選択にすることができます。
- SQL が UNION キーワードを見つけると、各副選択を処理して中間結果表を作り、次に、各副選択の中間結果表を結合し、重複する行を削除して結合結果表を作成します。 選択文節をコード化するときには、異なる文節および手法を使用できます。
- UNION 演算の結果で重複を残したいときは、UNION ではなく UNION ALL キーワードを指定します。
- UNION ステートメントについては詳細は下記を参照してください。
<https://www.ibm.com/docs/ja/i/7.5?topic=subselects-specifying-union-all-keyword>

4. IBM i の主要なSQLコマンド

(5) SELECTによるデータ照会の応用編 : 副照会

副照会は SELECT ステートメント中の検索条件をさらに絞り込みます。

副照会による条件の絞り込み

- SELECT <列名> FROM <表名> WHERE <条件> (<SELECT文>)
あるSELECT ステートメントを別の SELECT ステートメントの中にネストする

(例1) 会社全体の平均学歴より高い学歴を持つ社員の社員番号、氏名、および給与を取り出す

```
SELECT EMPNO, LASTNAME, SALARY FROM EMPLOYEE  
WHERE EDLEVEL > (SELECT AVG(EDLEVEL) FROM EMPLOYEE)
```

(例2) それぞれの所属部門の平均教育レベルより高い教育レベルを持つすべての社員のリストを取り出す

```
SELECT EMPNO, LASTNAME, WORKDEPT, EDLEVEL FROM EMPLOYEE X  
WHERE EDLEVEL > (SELECT AVG(EDLEVEL) FROM EMPLOYEE  
WHERE WORKDEPT = X.WORKDEPT)
```

解説：

- SQL では、ある SELECT ステートメントを別の SELECT ステートメントの中にネストできます。内部 SELECT ステートメントを副照会と呼びます。副照会を囲んでいる SELECT ステートメントを外部レベル SELECT と呼びます。
- 副照会は、他の副照会の検索条件の中に置くこともできます。このような副照会は、あるネスト・レベルで**ネストされた**副照会と呼びます。例えば、外部レベル SELECT 内の副照会の中の副照会は、ネスト・レベル 2 でネストされていることとなります。SQL では、ネスト・レベル 32 までネストできます。
- (例2)は、副照会から外部の要素を参照しています。このような副照会を相関副照会と呼びます。相関副照会については下記を参照してください。
<https://www.ibm.com/docs/ja/i/7.5?topic=subqueries-correlated>
- 副照会ステートメントについては詳細は下記を参照してください。
<https://www.ibm.com/docs/ja/i/7.5?topic=subqueries-in-select-statements>

4. IBM i の主要なSQLコマンド

(6) INSERT VALUES 文による行の挿入

単一行または複数行を表またはビューに追加するには、INSERT ステートメントの形式を使用します。

- ▶ INSERT INTO <表名> (<列名1>, <列名2>, …) : データを挿入する表名、列名を指定
列名を指定しない場合は全ての列が対象
VALUES(<値1>, <値2>, …) : 挿入する値を指定

(例) DEPARTMENT 表に新規行を挿入します。新規行の列は次のようになります。

- 部門番号 (DEPTNO) は 'E31'。
- 部門名 (DEPTNAME) は 'ARCHITECTURE'。
- 管理者番号 (MGRNO) は '00390'。
- (ADMRDEPT) 部門への報告書は 'E01'。

```
INSERT INTO DEPARTMENT (DEPTNO, DEPTNAME, MGRNO, ADMRDEPT)  
VALUES('E31', 'ARCHITECTURE', '00390', 'E01');
```

解説：

- INSERT ステートメントの列リストに名前を指定した各列について、VALUES 文節で値を指定しなければなりません。表内のすべての列が、VALUES 文節で指定する値を持つ場合には、列名リストを省略することができます。ある列にデフォルト値が入る場合には、VALUES 文節内の値としてキーワード DEFAULT を使用することができます。これにより、その列にデフォルト値が入れられます。
- VALUES 文節を使用して、表に複数行を挿入することもできます。
次の例は PROJECT 表に 2 つの行を挿入します。プロジェクト番号 (PROJNO)、プロジェクト名 (PROJNAME)、部門番号 (DEPTNO)、および担当従業員 (RESPEMP) の値が値リストに与えられます。プロジェクト開始日 (PRSTDATE) の値は現在日付を使用します。列リストにリストされていない表の列の残りには、デフォルト値が割り当てられます。

```
INSERT INTO PROJECT (PROJNO, PROJNAME, DEPTNO, RESPEMP, PRSTDATE)
VALUES('HG0023', 'NEW NETWORK', 'E11', '200280', CURRENT DATE),
      ('HG0024', 'NETWORK PGM', 'E11', '200310', CURRENT DATE);
```

4. IBM i の主要なSQLコマンド

(7) INSERT SELECT 文による行の挿入

INSERT ステートメントの中で選択ステートメントを使用すると、選択ステートメントの結果表から 0 行、1 行、または 2 行以上を表に挿入することができます。

- INSERT INTO <表名> (<列名1>, <列名2>, …) : データを挿入する表名、列名を指定
SELECT文 列名を指定しない場合は全ての列が対象
: 挿入する値を他の表から選択

(例) プロジェクトに対する各社員の従事時間を示す表にデータを挿入

```
INSERT INTO EMPTIME (EMPNUMBER, PROJNUMBER, STARTDATE, ENDDATE)  
SELECT EMPNO, PROJNO, EMSTDATE, EMENDATE FROM EMPPROJECT
```

解説：

- INSERT ステートメントに組み込む選択ステートメントは、データの取り出しに使用する選択ステートメントと変わりありません。SQL は、検索条件を満たすすべての行を指定の表に挿入します。1 つの表から別の表に行を挿入しても、ソース表の既存の行にもターゲット表の既存の行にも影響はありません。
- INSERT ステートメントに暗黙にまたは明示的にリストされた列の数は、選択ステートメントにリストされた列の数と同じでなければなりません。
- 選択ステートメントで指定した INSERT を使用する場合、選択する列のデータは、挿入先の列と互換性がなければなりません。

4. IBM i の主要なSQLコマンド

(8) UPDATE 文による行の更新

表またはビュー内のデータを更新するには、UPDATE ステートメントを使用します。

➤ UPDATE <表名> : 更新する表名を指定

SET <列名1> = <値1>, <列名2> = <値2>, . . . : 更新する列名と値を指定

(例1) 社員番号(EMPNO)が 000270 の行を更新

```
UPDATE EMPLOYEE SET WORKDEPT = 'D11', PHONENO = '7213', JOB = 'DESIGNER'  
WHERE EMPNO = '000270'
```

(例2) 部門が A00 の行に対して給与(SALARY)を1.1倍に更新

```
UPDATE EMPLOYEE SET SALARY = SALARY * 1.1 WHERE EMPNO = 'A00'
```

解説：

- UPDATE ステートメントを使用すると、WHERE 文節の検索条件を満たしている各行の 1 つまたは複数の列の値を変更することができます。UPDATE ステートメントを実行すると、WHERE 文節に指定された検索条件を満たす行の数に応じて、表の 0 個以上の行の 1 つまたは複数の列値が変更されます。
- 更新される行を識別するには、WHERE 文節を使用します。
 - 1 つの行を更新するには、1 つの行だけを選択する WHERE 文節を使用してください。
 - 複数の行を更新するには、更新したい行だけを選択する WHERE 文節を使用してください。
- WHERE 文節は省略することができます。省略すると、表またはビューの各行が、指定した値で更新されます。
- UPDATE ステートメントを使用した表内のデータの変更についての詳細は、下記を参照してください。
<https://www.ibm.com/docs/ja/i/7.5?topic=language-changing-data-in-table-using-update-statement>

4. IBM i の主要なSQLコマンド

(9) DELETE 文による行の削除

表から行を除去するときは、DELETE ステートメントを使用します。

- ▶ DELETE FROM <表名> : 行を削除する表を指定
- WHERE <条件> : 削除する行の条件を指定

(例1) PROJECT 表からPROJNOが HG0023 の行を削除

```
DELETE FROM PROJECT WHERE PROJNO= 'HG0023'
```

(例2) EMPTIME表から全ての行を削除

```
DELETE FROM EMPTIME
```

解説：

- ・ 行の削除では、その行全体が除去されます。DELETE ステートメントは、行から特定の列を取り除くためのものではありません。DELETE ステートメントが実行されると、WHERE 文節で指定された検索条件を満たす行の数に応じて、表の 0 個以上の行が削除されます。DELETE ステートメントで WHERE 文節の指定を省略すると、SQL は表からすべての行を削除します。
- ・ 表の内容とともに表定義を削除するには、DROP ステートメントを実行してください。
- ・ DELETE ステートメントを使用した表内のデータの変更についての詳細は、下記を参照してください。
<https://www.ibm.com/docs/ja/i/7.5?topic=language-removing-rows-from-table-using-delete-statement>

5. SQLを使ってみよう

(1) サンプル表を作ってみよう。

SQLのサンプル表作成のためのストアード・プロシージャが提供されています。

➤ `CALL QSYS.CREATE_SQL_SAMPLE ('<スキーマ名>')` で、作成可能です。

(例) ACSの「SQLスクリプトの実行」で、
`CALL QSYS.CREATE_SQL_SAMPLE ('SQLSAMPLE')`
を実行してみよう。スキーマ名は、既に存在していないものを指定。



```
ファイル(F) 編集(E) 検索(S) 表示(V) 接続(C) 実行(R) Explain(X) モニター(M) Editor ツール(T) ヘルプ(H)
*Untitled 1
1 CALL QSYS.CREATE_SQL_SAMPLE ('SQLSAMPLE')
2
```



```
[2024/10/01 16:56:37] すべて実行...
CALL QSYS.CREATE_SQL_SAMPLE ('SQLSAMPLE')
✓ 戻りコード = 0
✓ ステートメントは正常に実行されました (2.213 ミリ秒 = 2.213 秒)
```

解説1：

- ・ サンプル表は、「SQL プログラミング」、および「SQL 解説書」トピック集で参照および使用されています。これらのリンク先を補足資料を参照。
- ・ サンプル表の詳細は、下記になります。
 - ・ 部門表 (DEPARTMENT) : 社内の各部門が記述されます
 - ・ 社員表 (EMPLOYEE) : 全社員が社員番号で識別され、基本的な個人情報が記述されています
 - ・ 社員の写真表 (EMP_PHOTO) : 社員番号別に保管された、社員の写真が入っています
 - ・ 社員履歴表 (EMP_RESUME) : 社員の経歴が含まれ、社員番号別に保管されています。
 - ・ 社員プロジェクト活動表 (EMPPROJECT) : 各プロジェクト別の各作業を担当する社員が示されます
 - ・ プロジェクト表 (PROJECT) : 社内で現在進行中の各プロジェクトが記述されます
 - ・ プロジェクト活動表 (PROJACT) : 社内で現在進行中の各プロジェクトが記述されます
 - ・ 活動表 (ACT) : 各活動を記述します
 - ・ クラス・スケジュール表 (CL_SCHED) : クラスが記述されます
 - ・ 未処理表 (IN_TRAY) : メッセージ受信バスケットが記述されます
 - ・ 組織表 (ORG) : 企業の組織を記述します
 - ・ スタッフ表 (STAFF) : 従業員の背景情報を記述します
 - ・ 販売表 (SALES) : 各販売員の販売状況に関する情報を記述します

より詳細は下記を参照してください。

<https://www.ibm.com/docs/ja/i/7.5?topic=tables-sample>

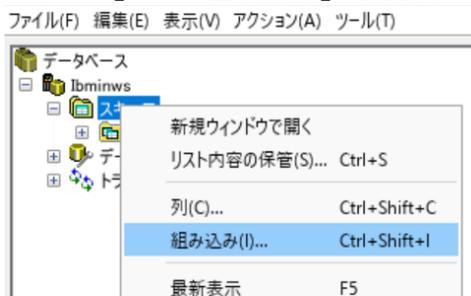
解説2：

- ・ ACSから、SQLSAMPLEというスキーマの作成状況を見てみましょう。

① 「アクション」 → 「スキーマ」 を選択



② 「スキーマ」 → 「組み込み」 を選択



③ スキーマの入力に「SQLSAMPLE」で追加



④ 「SQLSAMPLE」 → 「表」 を選択します。

下記のように、解説1のサンプル表が作成されています。

この「スキーマ」のメニューからは、GUIでデータベースを運用できます。



5. SQLを使ってみよう

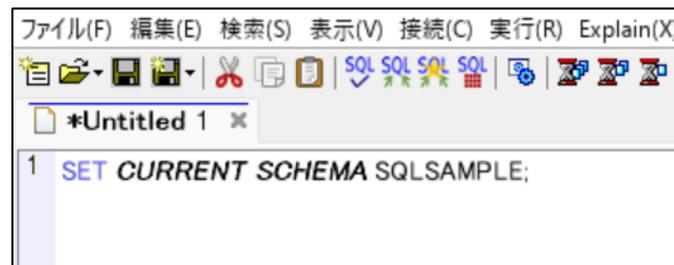
(2) ACSで、サンプル表を使ってみよう

ここまでで紹介した、SQLの操作を「SQLスクリプト実行」で使ってみよう

①ACSで「アクション」→「SQLスクリプト実行」を選択



②現行スキーマを先ほど作成した「SQLSAMPLE」にする



③EMPLOYEE表の全ての行の特定の列を選択

`SELECT EMPNO, LASTNAME, JOB FROM EMPLOYEE`

A screenshot of the SQL editor window showing the command: `1 SELECT EMPNO, LASTNAME, JOB FROM EMPLOYEE;`. Below the editor, a table of results is displayed:

EMPNO	LASTNAME	JOB
000010	HAAS	PRES
000020	THOMPSON	MANAGER
000030	KWAN	MANAGER
000050	GEYER	MANAGER
000060	STERN	MANAGER
000070	PULASKI	MANAGER
000080	HENDERSON	MANAGER
000100	SPENSER	MANAGER
000110	LUCCHESI	SALESREP
000120	O'CONNELL	CLERK
000130	QUINTANA	ANALYST
000140	NICHOLLS	ANALYST
000150	ADAMSON	DESIGNER

④部門(WORKDEPT)の昇順、ラストネーム(LASTNAME)の昇順にソート

`SELECT WORKDEPT, LASTNAME FROM EMPLOYEE ORDER BY WORKDEPT, LASTNAME`

A screenshot of the SQL editor window showing the command: `2 SELECT WORKDEPT, LASTNAME FROM EMPLOYEE ORDER BY WORKDEPT, LASTNAME;`. Below the editor, a table of results is displayed:

WORKDEPT	LASTNAME
A00	HAAS
A00	HEMINGER
A00	LUCCHESI
A00	O'CONNELL
A00	ORLANDO
B01	THOMPSON
C01	KWAN
C01	NATZ
C01	NICHOLLS
C01	QUINTANA
D11	ADAMSON
D11	BROWN
D11	JOHN
D11	JONES
D11	LUTZ
D11	PIANKA

5. SQLを使ってみよう

(3) 5250の対話式SQLで、サンプル表を使ってみよう。

①5250画面から、「STRSQL NAMING(*SQL)」を実行

②下記の画面が表示されます。

```

MAIN                                IBM I メインメニュー
次の 1 つを選択してください。

1. ユーザー・タスク
2. オフィス・タスク
3. 汎用システム・タスク
4. ファイル、ライブラリー、およびフォルダー
5. プログラミング
6. 通信
7. システムの定義または変更
8. 問題処理
9. メニューの表示
10. 情報援助オプション
11. IBM I ACCESS のタスク

90. サインオフ

選択項目またはコマンド
==> STRSQL NAMING(*SQL)
  
```



```

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 IBMINWS である。
==>
_
_
_
_
_
_
_
_
_
_
  
```

④EMPLOYEE表の全ての行の全ての列を選択
SELECT * FROM EMPLOYEE

③現行スキーマを先ほど作成した「SQLSAMPLE」にする

```

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。
> SET CURRENT SCHEMA SQLSAMPLE
SET CURRENT SCHEMA ステートメントが完了しました。
==>
_
_
_
_
_
_
  
```

データの表示

データの幅: 156
行の位置指定: 1 2 3 4 5 6 7 8 9 10 11 12 13

EMPNO	FIRSTNAME	MIDINIT	LASTNAME	WORKDEPT	PHONENO	HIREDATE	JOB	EDLEVEL	SEX	BIRTHDATE	SALARY
000010	CHRISTINE	I	HAAS	AD0	3978	65/01/01	PRES	18	F	++++++	52,750.00
000020	MICHAEL	L	THOMPSON	BO1	3476	73/10/10	MANAGER	18	M	48/02/02	41,250.00
000030	SALLY	A	KWAN	CO1	4738	75/04/05	MANAGER	20	F	41/05/11	38,250.00
000050	JOHN	B	GEYER	EO1	6789	49/08/17	MANAGER	16	M	++++++	40,175.00
000060	IRVING	F	STERN	D11	6423	73/09/14	MANAGER	16	M	45/07/07	32,250.00
000070	EVA	D	PULASKI	D21	7831	80/09/30	MANAGER	16	F	53/05/26	36,170.00
000090	EILEEN	W	HENDERSON	E11	5498	70/08/15	MANAGER	16	F	41/05/15	29,750.00
000100	THEODORE	Q	SPENSER	E21	0972	80/06/19	MANAGER	14	M	56/12/18	26,150.00
000110	VINCENZO	G	LUCCHESSI	AD0	3490	58/05/16	SALESREP	19	M	++++++	46,500.00
000120	SEAN	O	CONNELL	AD0	2167	63/12/05	CLERK	14	M	42/10/18	29,250.00
000130	DELORES	M	QUINTANA	CO1	4578	71/07/28	ANALYST	16	F	++++++	23,800.00
000140	HEATHER	A	NICHOLLS	CO1	1793	76/12/15	ANALYST	18	F	46/01/19	28,420.00
000150	BRUCE	ADARSON	D11	4510	72/02/12	DESIGNER	16	M	47/05/17	25,280.00	
000160	ELIZABETH	R	PLANKA	D11	3782	77/10/11	DESIGNER	17	F	55/04/12	22,250.00

まとめ：SQLを使ってみよう

- ✓ SQLはデータベース言語としてISO（国際標準化機構）で規格化されています。IBM i のSQLは、ISOの標準規格に準拠しており、OS内に組み込まれています。
- ✓ SQLは、複数のテーブルのデータを1つのデータベース内で組み合わせて、複雑なデータ処理を実行することができます（簡単に使えます）。
- ✓ 論理ファイルを大幅に減らせて、シンプルなシステムが実現できます。
- ✓ IBM i には、様々なSQLパフォーマンスのチューニングのツールが備わっています（SQL中上級編で、記述します）。

SQLを、まずは自社のデータ分析から、使ってみましょう。

4. 補足情報

1. Db2 for i SQL 解説書 (IBM i 7.5版)
https://www.ibm.com/docs/ja/ssw_ibm_i_75/pdf/rbafzpdf.pdf
2. SQL プログラミング (IBM i 7.5版)
https://www.ibm.com/docs/ja/ssw_ibm_i_75/pdf/rbafypdf.pdf
3. SQLの誕生と、標準SQL企画について
<https://ja.wikipedia.org/wiki/SQL>

IBM i 関連情報

IBM i ポータル・サイト

<https://ibm.biz/ibmijapan>

i Magazine (IBM i 専門誌。春夏秋冬の年4回発刊)

<https://www.imagazine.co.jp/IBMi/>

IBM i World 2024 オンデマンド・セミナー

<https://video.ibm.com/recorded/133917616>

IBM i World 2023 オンデマンド・セミナー

<https://ibm.biz/ibmiworld2023>

IBM i World 2022 オンデマンド・セミナー

<https://video.ibm.com/recorded/132423205>

月イチIBM Power情報セミナー「IBM Power Salon」

<https://ibm.biz/power-salon>

IBM i 関連セミナー・イベント

<https://ibm.biz/powerevents-j>

IBM i Club (日本のIBM i ユーザー様のコミュニティー)

<https://ibm.biz/ibmiclubjapan>

IBM i 研修サービス (i-ラーニング社提供)

<https://www.i-learning.jp/service/it/iseries.html>

IBM TechXchange Powerユーザーコミュニティー (日本)

<https://ibm.biz/ibm-power-user-community>

IBM i 情報サイト iWorld

<https://ibm.biz/iworldweb>

IBM i サポートロードマップ

<https://public.dhe.ibm.com/systems/support/planning/transfer/IBM.i.Support.Roadmap.pdf>

IBM i 7.5 技術資料

<https://www.ibm.com/docs/ja/i/7.5>

IBM Power ソフトウェアのダウンロードサイト (ESS)

<https://ibm.biz/powerdownload>

Fix Central (HW・SWのFix情報提供)

<https://www.ibm.com/support/fixcentral/>

IBM My Notifications (IBM IDの登録 [無償] が必要)

「IBM i」「9009-41G」などPTF情報の必要な製品を選択して登録できます。

<https://www.ibm.com/support/mynotifications>

IBM i 各バージョンのライフサイクル

<https://www.ibm.com/support/pages/release-life-cycle>

IBM i 以外のSWのライフサイクル (個別検索)

<https://www.ibm.com/support/pages/lifecycle/>

IBM Power Systems Virtual Server 情報

<https://ibm.biz/pvsjapan>

IBM i Advantage 2024 開催決定!

12月3日(火)・4日(水)

IBM虎ノ門イノベーションスタジオ (メイン会場)

IBM大阪中之島フェスティバルタワー・ウエスト (中継)

12月24日(火)

Webセミナー + Q&Aセッション

今からカレンダーブロックをお願いいたします。

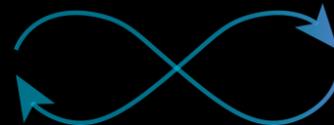
下記サイトに最新情報を掲載します↓

<https://ibm.biz/ibmiadvantage2024>



IBM i

continuous innovation
continuous integration



Advantage 2024

The classic IBM logo, consisting of the letters 'IBM' in a bold, sans-serif font with horizontal stripes.



ワークショップ、セッション、および資料は、IBMによって準備され、IBM独自の見解を反映したものです。それらは情報提供の目的のみで提供されており、いかなる読者に対しても法律的またはその他の指導や助言を意図したのではなく、またそのような結果を生むものでもありません。本資料に含まれている情報については、完全性と正確性を期するよう努力しましたが、「現状のまま」提供され、明示または暗示にかかわらずいかなる保証も伴わないものとします。本資料またはその他の資料の使用によって、あるいはその他の関連によって、いかなる損害が生じた場合も、IBMは責任を負わないものとします。本資料に含まれている内容は、IBMまたはそのサプライヤーやライセンス交付者からいかなる保証または表明を引き出すことを意図したもので、IBMソフトウェアの使用を規定する適用ライセンス契約の条項を変更することを意図したものでなく、またそのような結果を生むものでもありません。

本資料でIBM製品、プログラム、またはサービスに言及していても、IBMが営業活動を行っているすべての国でそれらが使用可能であることを暗示するものではありません。本資料で言及している製品リリース日付や製品機能は、市場機会またはその他の要因に基づいてIBM独自の決定権をもっていつでも変更できるものとし、いかなる方法においても将来の製品または機能が使用可能になると確約することを意図したものではありません。本資料に含まれている内容は、読者が開始する活動によって特定の販売、売上高の向上、またはその他の結果が生じると述べる、または暗示することを意図したもので、またそのような結果を生むものでもありません。パフォーマンスは、管理された環境において標準的なIBMベンチマークを使用した測定と予測に基づいています。ユーザーが経験する実際のスループットやパフォーマンスは、ユーザーのジョブ・ストリームにおけるマルチプログラミングの量、入出力構成、ストレージ構成、および処理されるワークロードなどの考慮事項を含む、数多くの要因に応じて変化します。したがって、個々のユーザーがここで述べられているものと同様の結果を得られると確約するものではありません。

記述されているすべてのお客様事例は、それらのお客様がどのようにIBM製品を使用したか、またそれらのお客様が達成した結果の実例として示されたものです。実際の環境コストおよびパフォーマンス特性は、お客様ごとに異なる場合があります。

IBM、IBM ロゴ、ibm.com、Db2、Rational、Power、POWER8、POWER9、AIXは、世界の多くの国で登録されたInternational Business Machines Corporationの商標です。

他の製品名およびサービス名等は、それぞれIBMまたは各社の商標である場合があります。

現時点での IBM の商標リストについては、www.ibm.com/legal/copytrade.shtml をご覧ください。

インテル、Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Centrino、Intel Centrino ロゴ、Celeron、Xeon、Intel SpeedStep、Itanium、およびPentium は Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

Linuxは、Linus Torvaldsの米国およびその他の国における登録商標です。

Microsoft、Windows、Windows NT および Windows ロゴは Microsoft Corporationの米国およびその他の国における商標です。

ITILはAXELOS Limitedの登録商標です。

UNIXはThe Open Groupの米国およびその他の国における登録商標です。

JavaおよびすべてのJava関連の商標およびロゴは Oracleやその関連会社の米国およびその他の国における商標または登録商標です。