

データベース入門 資料 8

DBMS と SQL

2011 年 6 月 13 日

目 次

1	SQL とは	1
1.1	データベース	1
1.2	ファイルとデータベースからの検索の実際	1
1.2.1	UNIX コマンドを使ったファイル内検索	1
1.2.2	データベースシステムを使った検索例	2
1.3	関係データベースと SQL	2
1.3.1	関係データベース	2
1.3.2	SQL	2
1.4	SQL 文の記述に関する注意	3
2	psql コマンドの使い方	3
2.1	起動方法	3
2.2	起動後の操作法	4
3	練習	4

1 SQL とは

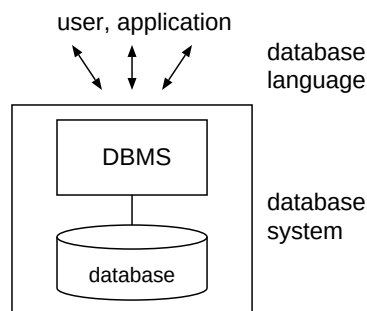
1.1 データベース

データベース 複数のユーザーやプログラムで共有されることを意図したデータの集まりをデータベース (database) といいます。

データベース管理システム (DBMS) データベースとその利用者 (ユーザーやアプリケーション) の仲介をするのがデータベース管理システム (database management system; DBMS) です。データベース管理システムはデータベースを一括して管理し、利用者からのデータの蓄積や検索、更新の要求に応え、その処理を効率良く確実にを行います。

データベースシステム データベースとデータベース管理システムを合わせてデータベースシステム (database system) といいます。データベースという用語はデータベースシステムを指して使われることもあります。

データベース言語 利用者はデータベース言語 (database language) を利用して、データベース管理システムを通じてデータベースを操作します。データベース言語は、データベースの統一的な利用法を利用者に提供するものです。代表的なデータベース言語として SQL があります。



1.2 ファイルとデータベースからの検索の実際

1.2.1 UNIX コマンドを使ったファイル内検索

1. less コマンドを使って大きなテキストファイル (全国郵便番号データ; 約 12 万行) を表示して検索する

```
less /pub/db/postcode/ken_all
```

- (a) less の / コマンドで 01202 を検索する
- (b) / で 47201 を検索 (少し時間がかかります)

2. grep コマンドで同じファイルを検索

```
grep 47201 /pub/db/postcode/ken_all |less
```

1.2.2 データベースシステムを使った検索例

ファイル `ken_all` と同等のデータを含むデータベースを用意しています。データベース管理システムを通じて、そこから検索を行ってみます。

1. `psql` コマンドを実行してデータベースに接続する

```
psql gamera (データベース gamera に接続)
```

2. 検索

```
select * from postcode where pub_auth_code = '47201';
```

もし、検索結果が一画面に収まらないために、`more` や `less` コマンドのようなページャを通じて表示されたなら、`q` を押して検索結果の表示を終了する。

3. `psql` を終了してデータベースシステムとの接続を絶ち、UNIX のシェルに戻る

```
\q (psql の終了)
```

1.3 関係データベースと SQL

1.3.1 関係データベース

データベースシステムには幾つかの種類がありますが、この授業では、現在最も標準的に用いられている**関係データベース** (*relational database*) システムを扱います。

関係データベースシステムにおけるデータベースは**テーブル** (表; table) の集まりです。テーブルを関係 (relation) と呼ぶことがあります¹。

テーブルは**行** (*row*) と**列** (*column*) で構成され、列には必ず名前があります。行をタプル (tuple), 列を属性 (attribute), 列の名前を属性名と呼ぶことがあります²。

1.3.2 SQL

SQL (structured query language) は関係データベースを扱うためのデータベース言語です。SQL では、C などの手続き型言語とは違い、原則として目的を達成するための手順 (how) ではなく、何をしたいか (what) を記述します。

この授業で主に扱う SQL は、ANSI (米国規格協会) と ISO (国際標準化機構) が 1992 年に制定した SQL2 規格 (通称 SQL-92) です。

SQL2 は日本語の利用が可能になるよう、1995 年に JIS (日本工業規格) 規格化もされています。SQL 規格の改訂はその後も続けられています。

¹集合論での「関係」に由来します

²`awk` での用語に対応させれば、行はレコードであり、列はフィールドです。

1.4 SQL 文の記述に関する注意

SQL 文

```
select * from postcode where new_post_code = '0400044';
```

は、テーブル `postcode` から、列 (属性) `new_post_code` の値が `0400044` である行を表示するものです。この文における `select` や `from`, `where` は、SQL において特別の意味を持つ語であり、これらを SQL の**キーワード**と呼びます。

SQL 文の記述では、キーワードやテーブル名および属性名に大文字／小文字の区別はありませんので、どちらで書いても構いません。ただし、書籍等では文を見やすくするために、キーワードを大文字で表記し、テーブル名や属性名を小文字にするのが一般的です。この資料でも、これ以降は原則として、キーワードを大文字で記しますが、`psql` で実行する際には小文字でタイプして構いません。

一方、`0400044` はデータベースからの検索の際に、検索条件として使う値 (文字) です。値の大文字/小文字は区別されます。また、これから当面は、値を SQL 文に記述する場合には、それを、(単一引用符; シングルクォート) で囲みます。ただし、値を単一引用符で囲むか否かは列の「データ型」に依存します。データ型については後の回で説明します。なお、値として、自身を表現したければ、`''` とします。

語と語との区切りには連続する空白文字や改行を使うことができます。したがって、上記の SQL 文は

```
SELECT *  
FROM postcode  
WHERE new_post_code = '0400044';
```

と書くこともできますし、逆に複数行の SQL 文を一行で書いても構いません。

2 psql コマンドの使い方

この授業では、データベースシステムとして PostgreSQL というソフトウェアを利用します。`psql` は PostgreSQL に付属する、データベース操作のための対話型コマンドです³。この授業では `psql` コマンドを使って SQL 文を実行します。

2.1 起動方法

`psql` の基本的な起動の仕方は

```
psql データベース名
```

です。

`psql` コマンドの引数には、オプションや接続するデータベース名を指定しますが、実際に何を指定するかについては、授業での指示に従ってください。なお、`psql` コマンドの一般的な実行の書式は `psql --help` でわかります。

³`psql` は標準的な UNIX コマンドではないので、PostgreSQL がインストールされていないシステムでは使えません。

2.2 起動後の操作法

1. psql を起動すると、次のような表示が表れます。

```
Welcome to psql 8.1.11, the PostgreSQL interactive terminal.
```

```
Type: \copyright for distribution terms
      \h for help with SQL commands
      \? for help with psql commands
      \g or terminate with semicolon to execute query
      \q to quit
```

```
gamera=>
```

最後の `gamera=>` は psql におけるプロンプトです。これに続き SQL 文などをタイプし、`<Enter>` キーを押して実行します。プロンプトにおける `gamera` は接続中のデータベース名を表しています。

2. psql の終了は `\q` です。
3. SQL 文の書式を確認するには `\h` でヘルプを表示します。
4. SQL 以外の psql の独自コマンド (`\` で始まるコマンド) のヘルプは `\?` です。
5. SQL の命令は文を単位として行います。psql を利用している場合、SQL 文 (`\` 以外で始まる命令) はセミコロン (`;`) で終わります。

セミコロンを打たずに `<Enter>` を押すと、SQL は実行されずに `gamera->` のようなプロンプトが現れます (`=>` ではなく `->` である点に注意)。SQL 文では、改行は無視されますので、続いてセミコロンを打ってから `<Enter>` を押せば、元の SQL を実行できます。

SQL を実行せずに、元のプロンプト (`=>`) に戻すには、`CTRL-c` を打ちます。

6. tcsh のコマンドラインとほぼ同様の履歴 (ヒストリ) 機能や補完機能が使えます。

3 練習

1. 第 1.2 節の操作を試してください。
2. 再度 psql `gamera` を実行してから、次の SQL 文による検索 (問い合わせ) を行ってください。すべて小文字でタイプして構いません。

```
SELECT * FROM postcode WHERE pub_auth_code = '01202';
```

3. 次の問い合わせを実行してください。ただし、psql の履歴 (ヒストリ) 機能を活用してください。

```
SELECT * FROM postcode WHERE pub_auth_code = '13101';
```

4. 再度、同じ問い合わせを実行します。ただし、今回は 3 行に分けてタイプしてください。

```
select * from postcode  
where pub_auth_code = '13101'  
;
```

5. 次の SQL 文による問い合わせを実行して、郵便番号から住所を検索してください。

```
SELECT * FROM postcode WHERE new_post_code = '0400044';
```

6. 次の SQL 文も試してください。

```
SELECT * FROM postcode WHERE new_post_code = '0400000';
```

7. 自分の家の郵便番号などを使って検索してみてください。