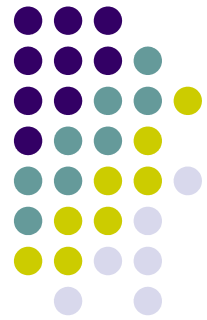


PostgreSQL 入門

3 章 基本的な SQL

SRA OSS, Inc.

<http://www.sraoss.co.jp/>



2. 1. テーブルの作成 (SQL)



● CREATE TABLE 文

● 構文

```
CREATE TABLE テーブル名 (カラム名1 データ型1, カラム名2 データ型2...);
```

- テーブル名、カラム名の大文字／小文字は区別されない
- PostgreSQL の予約語は使えない
- 日本語が使えるが、推奨されない



● テーブルを作成する

```
userDB=> CREATE TABLE fruit
userDB-> (id INTEGER, name TEXT, price INTEGER);
CREATE TABLE
```

- 右上の表のようなテーブルが作られる(中身はまだ無い、0行)

PostgreSQL 入門 3章 基本的な SQL
Copyright © SRA OSS, Inc.

8



CREATE TABLE

テーブルを定義するには CREATE TABLE 文を使います。テーブル名とテーブルを構成する各カラムの名前と型を定義します。

文字としてマルチバイト文字を使うこともできます。大文字小文字は区別されず、小文字としてあつかわれます。ダブルクォート(“)で囲めば大文字小文字の区別がされる任意の文字列を識別子にすることができます。

PostgreSQL の予約語

予約語(SQL 言語で決まった意味を持っている単語、キーワード)になっている語を識別子に使うには ”” で囲う必要があります。その場合、大文字小文字が区別されて扱われます。すべてのリストはマニュアルの「付録 C. SQL キーワード」に掲載されています。

FILLFACTOR

8.2 から、CREATE TABLE に FILLFACTOR 要素が追加されました。

そのテーブルに対しての INSERT 操作は指定した割合(%)までしかテーブルページを使用せず、各ページの残りの部分は、そのページ内の行更新用に予約されます。

これにより、更新時の「更新タプル取得」→「空を確認」→「空いていないので別のブロックを確保」という動作を減らします。たとえば、更新がないテーブルであれば 100、更新が非常に多いテーブルでは小さめの値を設定するのが適切です。

2. 2. データ登録 (SQL)

id	name	price
1	mikan	100
2	melon	2500
3	ringo	150



● INSERT 文

● 構文

```
INSERT INTO テーブル名 (カラム名 1...) VALUES (データ 1...);
```

- 文字列型の場合、' 'で囲み、大文字／小文字が区別される
- 日本語も問題なく使える

● 行を挿入する

```
userDB=> INSERT INTO fruit VALUES (1, 'mikan', 100);  
userDB=> INSERT INTO fruit VALUES (2, 'melon', 2500);  
userDB=> INSERT INTO fruit (name, price, id) VALUES ('ringo', 150, 3);
```

- 右上の表、3行挿入される。
- 順序は不定

カラムの順序は自由



INSERT

行を追加するには INSERT 文を使います。テーブルと対象カラムの並び、データの並びを指定します。

8.2 以降では、SQL 標準である、複数の VALUES 句を記述する挙動が可能になりました。結果、一文の INSERT で複数行の挿入が可能です。

```
userDB=> INSERT INTO fruit  
userDB-> VALUES (4, 'momo', 250), (5, 'suika', 1500);
```

2. 3. データ表示1 (SQL)

id	name	price
1	mikan	100
2	melon	2500
3	ringo	150



● SELECT 文

● 構文

```
SELECT カラム名 FROM テーブル名 [ オプション ] ;
```

- カラム名とテーブル名は複数指定できる

● 検索する

- カラム名に * を指定するとテーブルの全列を表示
- 「カラム名 AS 別名」でカラム名の表示を変更(日本語で表示したいときなど)

```
SELECT id, name AS 品名,  
price AS 値段 FROM fruit;
```

```
userDB=> SELECT name FROM fruit;  
name  
-----  
mikan  
melon  
ringo  
userDB=> SELECT * FROM fruit;  
id | name | price  
----+-----+-----  
1  | mikan | 100  
2  | melon | 2500  
3  | ringo | 150
```

SELECT

テーブルの行の内容を取り出すには SELECT 文を使います。

SELECT 文では対象のテーブル名とカラム名を指定します。カラム名にアスタリスク(*)を指定すると全てのカラム名という意味になります。

SELECT 文で FROM 句に複数のテーブルを指定すると、各テーブルの行の全ての組み合わせ(直積)が対象となります。カラム指定でどのテーブルのカラムなのかを指定するには fruit.name (テーブル fruit のカラム name) という書き方をします。

また、AS を指定して SELECT 文の出力におけるカラム名を変更することができます。

システム列

PostgreSQL のテーブルには明示的に定義したカラム(列)のほかに自動的に定義されるシステム列が含まれます。これらはカラム名として明示的に指定すると表示されず(アスタリスク*指定では表示されません)。

oid、xmin、xmax、cmin、cmax、ctid

CE 対策:

8.0 では default_with_oids というパラメータの on/off によって、行に対して OID を振るかどうかを切り替えられます。なお、7.4 以前では必ず振られていました。

また、8.0 ではデフォルトでは on ですが、8.1 からは off になっています。

データ表示 2 (SQL)

id	name	price
1	mikan	100
2	melon	2500
3	ringo	150



● 検索条件を指定する

● 構文

```
SELECT カラム名 FROM テーブル名 WHERE 条件 ;
```

- 条件には各行に対して真／偽を返す任意の式が指定可能
- 指定しない場合、全ての行が対象となる

● 範囲指定して検索する

```
userDB=> SELECT * FROM fruit WHERE price > 100;
 id | name  | price
----+-----+-----
  2 | melon | 2500
  3 | ringo | 150
```



SELECT 文の WHERE 句

SELECT 文は WHERE 句で条件を指定して出力する行を選択することができます。条件には BOOLEAN 型 (TRUE / FALSE) を返す式を指定します。通常は以下のような等号、不等号など条件式が使われます。不等号は時刻や文字列 (アルファベット順) にも適用できます。複数の条件式を AND や OR で結合することができます。

条件式でよく使われる演算子／キーワード	例
=、!=	WHERE price = 150
<、>、>=、<=	WHERE price >= 100
AND	WHERE price > 100 AND id < 3
OR	WHERE id = 1 OR id = 2