

# 9章「PostgreSQL」

## (基本)

中島康彦

### §9.1 Postgres概要

当初UCBが開発したSQLデータベースサーバー

- ▶ `postmaster` -> `postgres`

クライアントインタフェース

- ▶ Windows用ODBCドライバ
- ▶ JAVA用JDBCドライバ
- ▶ Perlモジュール
- ▶ C言語ライブラリ
- ▶ TCL/TKライブラリ
- ▶ CGI(シェルスクリプト+psqlコマンドなど)

ドキュメント

- ▶ `/usr/local/pgsql/doc/html/index.html`
  - ▶ `/usr/local/pgsql/doc/jhtml/index.html`
-

## §9. 2 Postgresのインストール

---

postgresql-8.0.1.tar.gzを取得

インストール(管理者が1度だけ)

```
zcat postgresql-8.0.1.tar.gz | tar xpf -
cd postgresql-8.0.1
./configure
gmake
su
gmake install
```

pgsql管理用アカウント/データ置場作成(管理者が1度だけ)

```
adduser pgsql
mkdir /usr/local/pgsql/data
chown pgsql /usr/local/pgsql/data
```

---

## §9. 3 Postgresのインストール(つづき)

---

pgsqlとしてinitdbを実行(管理者が1度だけ)

```
su - pgsql
/usr/local/pgsql/bin/initdb -D
/usr/local/pgsql/data
```

```
The files belonging to this database system will be owned by user "pgsql".
This user must also own the server process.
The database cluster will be initialized with locale C.
fixing permissions on existing directory /usr/local/pgsql/data ... ok
creating directory /usr/local/pgsql/data/global ... ok
:
selecting default max_connections ... 40
selecting default shared_buffers ... 1000
creating configuration files ... ok
creating template1 database in /usr/local/pgsql/data/base/1 ... ok
initializing pg_shadow ... ok
enabling unlimited row size for system tables ... ok
initializing pg_depend ... ok
creating system views ... ok
loading pg_description ... ok
creating conversions ... ok
setting privileges on built-in objects ... ok
creating information schema ... ok
vacuuming database template1 ... ok
copying template1 to template0 ... ok
WARNING: enabling "trust" authentication for local connections
You can change this by editing pg_hba.conf or using the -A option the
next time you run initdb.
Success. You can now start the database server using:
```

---

## §9. 4 Postgresの起動

---

pgsqlとしてpostmasterサーバを実行(OS起動時に1度)

(/etc/rc.localなどを書いておけば良い)

```
/usr/local/pgsql/bin/postmaster -D  
/usr/local/pgsql/data >logfile 2>&1 &
```

```
LOG: database system was shut down at 2005-02-21 22:13:34 JST  
LOG: checkpoint record is at 0/A2C840  
LOG: redo record is at 0/A2C840; undo record is at 0/0; shutdown TRUE  
LOG: next transaction ID: 544; next OID: 17230  
LOG: database system is ready
```

以上で、最低限必要な環境は整った。

---

## §9. 5 ユーザの登録/削除

---

必要に応じて、DBを操作できる一般ユーザを登録する

- ▶ DBを作る権限をpgsqlに限るか、一般ユーザにも与えるかは、サーバの管理ポリシー次第。
- ▶ 同様に、別の一般ユーザを登録する権限をpgsqlに限るか、一般ユーザにも与えるかも、サーバの管理ポリシー次第。

```
su - postgres  
/usr/local/pgsql/bin/createuser  
Enter name of user to add: nakashim  
Shall the new user be allowed to create databases? (y/n) y  
Shall the new user be allowed to create more new users? (y/n) y  
CREATE USER
```

一般ユーザの削除

```
/usr/local/pgsql/bin/dropuser
```

---

## §9.6 データベースの作成/削除

権限のあるユーザはDBを作成/削除できる。ここから演習

- ▶ データベース数の制限はない。
- ▶ 作成したユーザがそのデータベースの管理者となる。
- ▶ データベース名は一意。アルファベットに始まる32文字以内。

1. % **/usr/local/pgsql/bin/createdb xxx**

データベースの作成権限がない場合,

```
NOTICE:user "your username" is not allowed to create/destroy databases
createdb: database creation failed on xxx.
```

2. % **/usr/local/pgsql/bin/createdb xxx**

```
CREATE DATABASE
```

3. % **/usr/local/pgsql/bin/dropdb xxx**

データベースの管理者のみが削除できる。

一度削除したら元に戻せない。

```
DROP DATABASE
```

---

## §9.7 データベースへの接続

SQL操作(クライアントプログラムから実行)

- ▶ テーブル (関係表) の作成/消去, 情報の入出力
- ▶ WEBサーバやパソコンソフトなどから行う
- ▶ ただしPostgreSQLに対する問い合わせ ≠ SQL標準

4. % **/usr/local/pgsql/bin/psql xxx**

```
Welcome to psql 8.0.1, 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
```

```
xxx=#
```

SQL入力待ち状態.

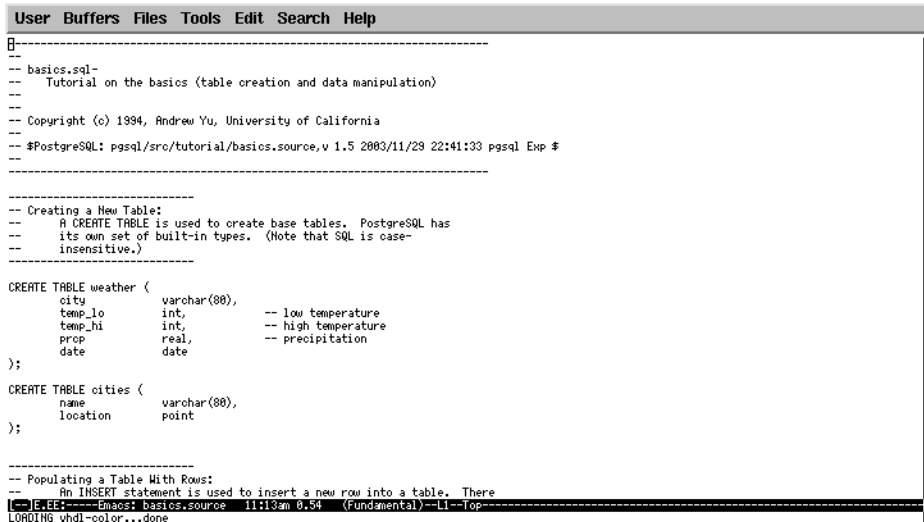
---

## §9.7 データベースへの接続(つづき)

キーボードから入力できるが、編集済のファイルを使う方が確実

- ▶ `\i fileName ...` SQLをファイルから読み込む
- ▶ サンプルファイル

```
/usr/local/pgsql/tutorial/basics.source
--          ... 行の最後までをコメントとみなす.
/* */      ... コメント
```



```
User Buffers Files Tools Edit Search Help
-----
-- basics.sql-
-- Tutorial on the basics (table creation and data manipulation)
--
-- Copyright (c) 1994, Andrew Yu, University of California
--
-- $PostgreSQL: pgsql/src/tutorial/basics.source,v 1.5 2003/11/29 22:41:33 pgsql Exp $
-----
-- Creating a New Table:
-- A CREATE TABLE is used to create base tables. PostgreSQL has
-- its own set of built-in types. (Note that SQL is case-
-- insensitive.)
-----
CREATE TABLE weather (
  city          varchar(80),
  temp_lo      int,          -- low temperature
  temp_hi      int,          -- high temperature
  prcp         real,         -- precipitation
  date         date
);

CREATE TABLE cities (
  name          varchar(80),
  location      point
);

-----
-- Populating a Table With Rows:
-- An INSERT statement is used to insert a new row into a table. There
-- is one row in the weather table.
--
-- [E]EE:--Emaas: basics.source 11:13am 0.54 (Fundamental)--L1--top
LOADING vhd1-color...done
```

## §9.8 サンプルファイルを使った演習(新しいテーブルの作成)

-sを付けてシングルステップモードにて実行

1. `% cd /usr/local/pgsql/tutorial`
2. `% /usr/local/pgsql/bin/psql -s xxx`
3. `xxx=# \i basics.source`

```
CREATE TABLE weather (
  city          varchar(80) ,
  temp_lo      int,          -- 低温
  temp_hi      int,          -- 高温
  prcp         real,         -- 降水量
  date         date
);
```
4. **\*\* Enter \*\***

```
CREATE TABLE cities (
  name          varchar(80) ,
  location      point
);
```

## §9. 8 新しい行の登録

---

以後, 実際には, 表示内容を入力していく.

5. `INSERT INTO weather  
VALUES ('San Francisco',  
46, 50, 0.25, '1994-11-27');`
6. `INSERT INTO cities  
VALUES ('San Francisco', '(-194.0, 53.0)');`
7. `INSERT INTO weather (city, temp_lo,  
temp_hi, prcp, date)  
VALUES ('San Francisco',  
43, 57, 0.0, '1994-11-29');`
8. `INSERT INTO weather (date, city,  
temp_hi, temp_lo)  
VALUES ('1994-11-29', 'Hayward', 54, 37);`

---

## §9. 9 検索

---

9. `SELECT * FROM weather;`

city	temp_lo	temp_hi	prcp	date
San Francisco	46	50	0.25	1994-11-27
San Francisco	43	57	0	1994-11-29
Hayward	37	54		1994-11-29

(3 rows)

10. `SELECT city, (temp_hi+temp_lo)/2  
AS temp_avg, date FROM weather;`

city	temp_avg	date
San Francisco	48	1994-11-27
San Francisco	50	1994-11-29
Hayward	45	1994-11-29

(3 rows)

---

## §9. 9 検索

---

演算子 AND, OR, NOT

```
11. SELECT * FROM weather WHERE
    city = 'San Francisco' and prcp > 0.0;
```

city	temp_lo	temp_hi	prcp	date
San Francisco	46	50	0.25	1994-11-27

(1 row)

並べ替え, 重複行の削除

```
12. SELECT DISTINCT city FROM weather
    ORDER BY city;
```

city
Hayward
San Francisco

(2 rows)

---

## §9. 9 Join

---

テーブル間の結合

```
13. SELECT *
    FROM weather, cities
    WHERE city = name;
```

city	temp_lo	temp_hi	prcp	date	name	location
San Francisco	46	50	0.25	1994-11-27	San Francisco	(-194,53)
San Francisco	43	57	0	1994-11-29	San Francisco	(-194,53)

(2 rows)

```
14. SELECT city, temp_lo, temp_hi, prcp, date, location
    FROM weather, cities
    WHERE city = name;
```

city	temp_lo	temp_hi	prcp	date	location
San Francisco	46	50	0.25	1994-11-27	(-194,53)
San Francisco	43	57	0	1994-11-29	(-194,53)

(2 rows)

---

## §9. 9 Join(つづき)

---

15. `SELECT weather.city, weather.temp_lo,  
weather.temp_hi, weather.prcp, weather.date,  
cities.location  
FROM weather, cities  
WHERE cities.name = weather.city;`

city	temp_lo	temp_hi	prcp	date	location
San Francisco	46	50	0.25	1994-11-27	(-194,53)
San Francisco	43	57	0	1994-11-29	(-194,53)

(2 rows)

16. `SELECT *  
FROM weather JOIN cities ON  
(weather.city = cities.name);`

city	temp_lo	temp_hi	prcp	date	name	location
San Francisco	46	50	0.25	1994-11-27	San Francisco	(-194,53)
San Francisco	43	57	0	1994-11-29	San Francisco	(-194,53)

(2 rows)

---

## §9. 9 Outer join

---

17. `SELECT *  
FROM weather LEFT OUTER JOIN cities ON  
(weather.city = cities.name);`

city	temp_lo	temp_hi	prcp	date	name	location
San Francisco	46	50	0.25	1994-11-27	San Francisco	(-194,53)
San Francisco	43	57	0	1994-11-29	San Francisco	(-194,53)
Hayward	37	54		1994-11-29		

(3 rows)

18. `SELECT W1.city, W1.temp_lo, W1.temp_hi,  
W2.city, W2.temp_lo, W2.temp_hi  
FROM weather W1, weather W2  
WHERE W1.temp_lo < W2.temp_lo  
and W1.temp_hi > W2.temp_hi;`

city	temp_lo	temp_hi	city	temp_lo	temp_hi
San Francisco	43	57	San Francisco	46	50
Hayward	37	54	San Francisco	46	50

(2 rows)

---



## §9. 10 統計

---

```
19. SELECT max(temp_lo)
    FROM weather;
```

```
max
----
 46
(1 row)
```

```
20. SELECT city FROM weather
    WHERE temp_lo = (SELECT max(temp_lo) FROM weather);
```

```
city
-----
San Francisco
(1 row)
```

```
21. SELECT city, max(temp_lo)
    FROM weather
    GROUP BY city;
```

```
city | max
-----+-----
Hayward | 37
San Francisco | 46
(2 rows)
```

---

## §9. 10 統計

---

```
22. SELECT city, max(temp_lo)
    FROM weather
    GROUP BY city
    HAVING max(temp_lo) < 40;
```

```
city | max
-----+-----
Hayward | 37
(1 row)
```

---

## §9. 11 更新

---

```
23. UPDATE weather
    SET temp_hi = temp_hi - 2, temp_lo = temp_lo - 2
    WHERE date > '1994-11-28';
```

```
24. SELECT * FROM weather;
```

city	temp_lo	temp_hi	prcp	date
San Francisco	46	50	0.25	1994-11-27
San Francisco	41	55	0	1994-11-29
Hayward	35	52		1994-11-29

(3 rows)

---

## §9. 12 削除

---

```
25. DELETE FROM weather WHERE city = 'Hayward';
```

```
26. SELECT * FROM weather;
```

city	temp_lo	temp_hi	prcp	date
San Francisco	46	50	0.25	1994-11-27
San Francisco	41	55	0	1994-11-29

(2 rows)

```
27. DELETE FROM weather;
```

```
28. SELECT * FROM weather;
```

city	temp_lo	temp_hi	prcp	date
------	---------	---------	------	------

(0 rows)

```
29. DROP TABLE weather, cities;
```

```
DROP TABLE
```

---

今日はここまで