データベース構築論1

10章「PostgreSQL」

(発展)

中島康彦

§10. 1 再度，データベースへ接続

1. % /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.

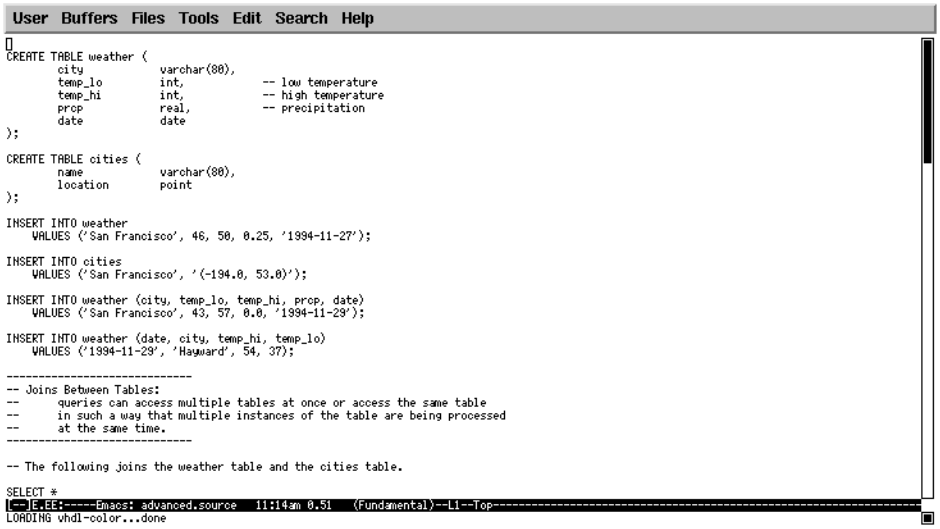
§10. 1 再度，データベースへ接続(つづき)

- サンプルファイル

  /usr/local/pgsql/tutorial/advanced.source

  --　　… 行の最後までをコメントとみなす.

  /* */　… コメント



§10. 2 サンプルファイルを使った演習)

-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 advanced.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
);
```

§10. 3 新しい行の登録(前回と同じ)

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);
```

§10. 4 VIEWの作成

テーブル間の結合(ここまでは以前と同じ)

```
9. 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)
```

VIEWを使う(いちいちQUERYを作らずに済む)

```
10. CREATE VIEW myview AS
    SELECT city, temp_lo, temp_hi, prcp, date, location
    FROM weather, cities WHERE city = name;

11. SELECT * FROM myview;
    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)
```

§10. 5 外部キー

参照一貫性の確保

```
12. CREATE TABLE cities (
        city     varchar(80) primary key,
        location point);
13. CREATE TABLE weather (
        city      varchar(80) references cities(city),
        temp_lo   int,
        temp_hi   int,
        prcp      real,
        date      date);
14. INSERT INTO weather VALUES
     ('Berkeley', 45, 53, 0.0, '1994-11-28');
    psql:advanced.source:66: ERROR:  insert or update
         on table "weather" violates foreign key
         constraint "weather_city_fkey"
    DETAIL:  Key (city)=(Berkeley) is not present
         in table "cities".
```

§10. 6 トランザクション

預金データベースの作成

```
15. CREATE TABLE accounts (
        name            varchar(80),
        branch_name     varchar(80),
        balance         int);
16. CREATE TABLE branches (
        name            varchar(80),
        balance         int);
17. INSERT INTO accounts VALUES ('Alice','Kyoto', 900);
18. INSERT INTO branches VALUES (       'Kyoto', 400);
19. INSERT INTO accounts VALUES ('Bob',  'Osaka', 900);
20. INSERT INTO branches VALUES (       'Osaka', 400);
21. SELECT * FROM accounts;
    SELECT * FROM branches;
     name  | branch_name | balance
    -------+-------------+---------
     Alice | Kyoto       |     900
     Bob   | Osaka       |     900
    (2 rows)

     name  | balance
    -------+---------
     Kyoto |     400
     Osaka |     400
    (2 rows)
```

§10. 6 トランザクション(つづき)

AliceからBobに，100だけ移動したい．以下は正しいか?

```
22. UPDATE accounts SET balance = balance - 100.00 WHERE name = 'Alice';
    UPDATE branches SET balance = balance - 100.00
       WHERE name = (SELECT branch_name FROM accounts WHERE name = 'Alice');
    UPDATE accounts SET balance = balance + 100.00 WHERE name = 'Bob';
    UPDATE branches SET balance = balance + 100.00
       WHERE name = (SELECT branch_name FROM accounts WHERE name = 'Bob');
```

途中でエラーが発生した場合は全てがなかったことにする

```
23. BEGIN;
     :
    COMMIT;
     name   | branch_name | balance
    -------+-------------+---------
     Alice | Kyoto       |     800
     Bob   | Osaka       |    1000
    (2 rows)

     name   | balance
    -------+---------
     Kyoto |     300
     Osaka |     500
    (2 rows)
```

§10. 6 トランザクション(つづき)

部分的に，なかったことにする．(ROLLBACK)
AliceからBobに100だけ移動するのを途中で止めて，
Wallyに100だけ移動する．

```
24. BEGIN;
    UPDATE accounts SET balance = balance - 100.00 WHERE name = 'Alice';
    SAVEPOINT my_savepoint;

    UPDATE accounts SET balance = balance + 100.00 WHERE name = 'Bob';
    -- oops ... forget that and use Wally's account

    ROLLBACK TO my_savepoint;
    UPDATE accounts SET balance = balance + 100.00 WHERE name = 'Wally';
    COMMIT;
     name   | branch_name | balance
    -------+-------------+---------
     Bob   | Osaka       |    1000
     Alice | Kyoto       |     700
     Wally | Tokyo       |     100
    (3 rows)
```

§10. 7 継承

非州都と州都をデータベース化するには以下でもできるが…

```
25. CREATE TABLE capitals (
      name        text,
      altitude    int,     -- (in ft)
      state       char(2));

    CREATE TABLE non_capitals (
      name        text,
      altitude    int      -- (in ft));

    CREATE VIEW cities AS
      SELECT name, population, altitude FROM capitals
      UNION
      SELECT name, population, altitude FROM non_capitals;
```

§10. 7 継承(つづき)

INHERITANCEを使う方がエレガント

```
26. CREATE TABLE cities (
      name        text,
      altitude    int      -- (in ft));
    CREATE TABLE capitals (
      state       char(2)) INHERITS (cities);
27. INSERT INTO cities   VALUES ('Las Vegas', 2174);
    INSERT INTO cities   VALUES ('Mariposa', 1953);
    INSERT INTO capitals VALUES ('Madison', 845, 'WI');
28. SELECT * FROM cities;
    SELECT * FROM capitals;
       name     | altitude
    ------------+----------
     Las Vegas  |     2174
     Mariposa   |     1953
     Madison    |      845
    (3 rows)

      name    | altitude | state
    ----------+----------+-------
     Madison  |      845 | WI
    (1 row)
```

§10. 7 継承(つづき)

検索

```
29. SELECT name, altitude FROM cities WHERE altitude > 500;
      name     | altitude
    -----------+----------
     Las Vegas |     2174
     Mariposa  |     1953
     Madison   |      845
    (3 rows)

30. SELECT name, altitude FROM ONLY cities WHERE altitude > 500;
      name     | altitude
    -----------+----------
     Las Vegas |     2174
     Mariposa  |     1953
    (2 rows)
```

§10. 8 レポート課題

適当な実世界を仮定し，各スキーマを記述せよ.

- 概念スキーマ
  対応する実世界を明記すること.
- 内部スキーマ
  PostgreSQLを用いること.
- 外部スキーマ
  PostgreSQLのVIEWを用いること.
  2つ以上.
  各々の利用者を明記すること.

実際にテーブルを作成し，検索結果まで添付すること.

〆切　7／X(X) 講義終了時

今日はここまで