

第2章 MySQLの練習

MySQLはリレーショナルデータベース(RDB)を生成し、管理するためのソフトウェアである。以降の章では全てPerlやPHP等のスクリプト(=プログラム)からSQL文を呼び出してデータベースを操作することになるが、本章ではコマンドラインからSQL文を直接発行できるMySQL clientを使用し、以降で使用するSQL文の意味を理解するための訓練を行う。

2.1 MySQLのrootユーザ設定

どのOSにおいても、まずMySQLをインストールしたら、rootユーザと一般ユーザをきちんと登録しておこう。以下その手順を示す。Windows環境の場合は、Administrator権限で実行すること。

1. MySQL-clientを

```
# mysql -u root
```

とし、MySQLのrootユーザ(パスワードなし)で起動できることを確認する。

2. MySQLプロンプト"mysql>"が出ている状態になっていることを確認の上、SQL文が実行できることを確認する。とりあえず、MySQL(テーブル名: "mysql.user")に登録されているユーザ、ホスト、パスワードを表示してみよう。

```
mysql> select user,host,password from mysql.user;
```

下記のように、rootユーザのパスワードが空になっていることを確認する。

```
+-----+-----+-----+
| user | host      | password |
+-----+-----+-----+
| root | localhost |          |
```

```

| root | centos5 | |
|      | localhost | |
|      | centos5 | |
+-----+-----+-----+
4 rows in set (0.00 sec)

```

```
mysql>
```

3. このままではセキュリティ的にまずいので、localhostにおけるrootユーザのパスワードを設定する。もし"sist"という文字列をパスワードとしたければ

```
mysql> set password for root@localhost=password('sist');
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> flush privileges;
Query OK, 0 rows affected (0.01 sec)
```

とする。これから設定するMySQLのrootユーザのパスワードをメモしたら、それを上記コマンドで設定すること。設定が終わったら、quitコマンドでMySQLクライアントを一端終了する。

```
mysql> quit;
```

4. もう一度、rootユーザでログインし直してみる。今度はパスワード付きなので

```
# mysql -u root -p
```

を添付すること。パスワードを聞いてきたら、先ほど自分が設定した文字列を入力してログインできることを確認せよ。もし出来なければMySQLのインストールからやり直しとなる。確認できたら、MySQLクライアントは終了しておこう。

```
Enter password:パスワードを入力(画面には出ない)
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3 to server version: 4.0.25-standard
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql> select user,host,password from mysql.user;
```

```
+-----+-----+-----+
| user | host      | password      |
+-----+-----+-----+
| root | localhost | 6b5540c16b1e3d82 |
| root | centos5   |                  |
|      | localhost |                  |
|      | centos5   |                  |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

```
mysql> quit;
```

```
Bye
```

```
#
```

5. 一般ユーザアカウントでログインしなおし，同様にしてMySQL client をMySQL の root ユーザで起動でき，SQL 文が実行できることを確認する。

```
$ mysql -u root -p
```

```
Enter password:パスワードを入力(画面には出ない)
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
```

```
Your MySQL connection id is 3 to server version: 4.0.25-standard
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql> select user,host,password from mysql.user;
```

```
+-----+-----+-----+
| user | host      | password      |
+-----+-----+-----+
| root | localhost | 6b5540c16b1e3d82 |
| root | centos5   |                  |
|      | localhost |                  |
|      | centos5   |                  |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

```
mysql> quit;  
Bye
```

2.2 データベースを作成・削除する

以上のようにして MySQL client を使用する。大まか流れをまとめておくと、以下のようなになる。

1. コマンドライン (CentOS では端末, Windows ではコマンドプロンプト) から, ユーザ名とパスワードを入力して MySQL にログインする。

```
$ mysql -u ユーザ名 -p  
Enter password:パスワードを入力(画面には出ない)  
(略)  
mysql> <----- プロンプトが出ることを確認!
```

2. SQL 文を打ち込んで実行。

```
mysql> SQL 文; <----- 終端には必ずセミコロンを付ける
```

3. MySQL からログアウトする。

```
mysql> quit;
```

RDB は次の 3 要素 (DB, テーブル, フィールド) から成り立っている。

DB(データベース) テーブルを格納しておくための入れ物

テーブル フィールドで指定されたデータ型のデータを入れておくための入れ物

フィールド 指定されたデータ型のデータを入れておくためのラベル

この 3 者には全て名前 (DB 名, テーブル名, フィールド名) があり, 操作する際にはこれを指定する。DB やテーブルは複数作成することもできる。この 3 者の関係を図にしたのが図 2.1 である。

フィールドに指定できるデータ型の一覧は表 1, 表 2, 表 3 にある通りである。
以下, jikken という DB を作り, その中に

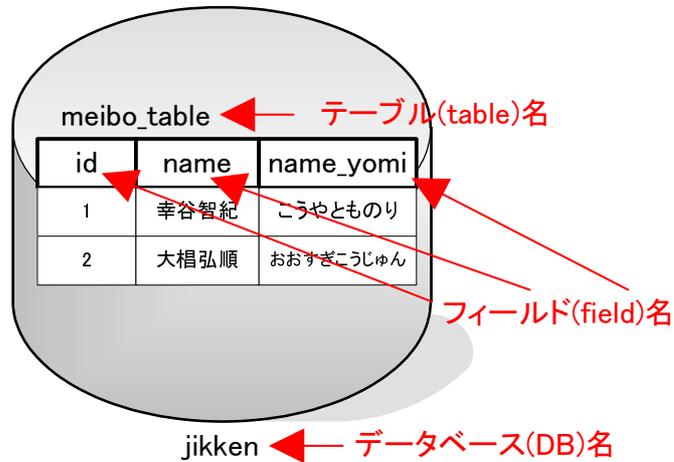


図 2.1: データベース・テーブル・フィールドの関係

フィールド名	データ型	役割
id	int & auto.increament	氏名の ID(自動更新)
name	varchar(20)	氏名(漢字)
yomi	varchar(40)	氏名(よみがな)

という3つのフィールドを持つ meibo_table テーブルを作成してみよう。

1. 実験用 DB(jikken) を作成する。

```
mysql> create database jikken;
Query OK, 1 row affected (0.00 sec)
```

2. 一度 MySQL を抜け、今度は使用するデータベース名も指定してログインする。

```
$ mysql -u root -p jikken
```

エラーが出てログインできなかったら、間違ったデータベース名を指定している可能性がある。その際には、次の項を参照してもう一度データベース作成からやり直すこと。なお、データベース名を指定せずに起動したときには、MySQL Client のプロンプトから

```
mysql> use データベース名;
```

として指定されたデータベースを使用することができる。

3. jikken データベースが存在することを"show databases"コマンドで確認。

```
mysql> show databases;
+-----+
| Database |
+-----+
| jikken   |
| mysql    |
| test     |
+-----+
3 rows in set (0.00 sec)
```

もしデータベース名が間違っていたら，一度MySQLを抜け，データベース名を指定せずにログインし，次の"show databases"コマンドで不要なデータベース名を確認して

```
mysql> drop database データベース名;
```

として削除する。

4. jikken データベースの中に名簿テーブル(meibo_table)を作成し，正常に作成されていることを確認する。

```
mysql> create table meibo_table (
-> id int not null auto_increment,
-> name varchar(20) not null,
-> name_yomi varchar(40) not null,
-> primary key(id)
-> );
Query OK, 0 rows affected (0.02 sec)
```

途中打ち間違えたら，セミコロン";"を打って一度抜け，再度上記のCREATE文を打ち直すこと。

5. 以下，"SHOW TABLES"でテーブル名(赤字部分)の確認を，"SHOW FIELDS"コマンドでフィールド名とデータ型の確認をする。

```
mysql> show tables;   テーブル名の確認
```

```
+-----+
| Tables_in_jikken |
+-----+
| meibo_table      |
+-----+
1 row in set (0.00 sec)
```

```
mysql> show fields from meibo_table;   フィールド名と型の確認
```

```
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)       |      | PRI | NULL     | auto_increment |
| name       | varchar(20)   |      |     |          |                |
| name_yomi  | varchar(40)   |      |     |          |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

すべて正常であることを確認したら mysql を終了する。もしテーブル作成に失敗していたら、

```
mysql> drop table テーブル名;
```

で削除し、もう一度 "create table" からやり直すこと。

なお、データベース内のテーブル名を、他のデータベースから指定する際には

データベース名. テーブル名

として指定することもできる。例えば、先の root ユーザのパスワード変更時に使用した mysql.user は、mysql データベース内の user テーブルを使った、ということになる。

2.3 データの追加・削除・検索

以上で作成した DB にデータを追加、更新、削除したり、検索するためには次の SQL 文を使用する。

追加 (**insert**) … insert into テーブル名 (フィールド名1, フィールド名2, …)
values(データ1, データ2, …)

例) 「吉田茂 (よしだしげる)」を追加

```
mysql> insert into meibo_table(name, name_yomi) values('吉田茂',  
'よしだしげる');
```

更新 (**update**) … update テーブル名 set フィールド名1='データ1', フィー
ルド名2='データ2', … where フィールド名='データ'

例) id=1 のデータを更新

```
mysql> update meibo_table set name='鳩山一郎', name_yomi='はとやま  
いちろう' where id=1;
```

削除 (**delete**) … delete from テーブル名 where フィールド名='データ'

例) id=1 のデータを削除

```
mysql> delete from meibo_table where id=1;
```

検索 (**select**) … select フィールド名1, … from テーブル名 (検索指定)

例) id=1 のデータを検索して表示

```
mysql> select name, name_yomi from meibo_table where id=1;
```

strut

例) “福田”を含む文字列のデータを検索して表示

```
mysql> select name, name_yomi from meibo_table like %福田%;
```

2.4 MySQL に一般ユーザを追加する

以上のSQL文を使って、一般ユーザを追加してみよう。MySQLのユーザは“mysql”データベース内のuserテーブルにあるので、ここに一般ユーザを追加し(insert文)、その後パスワードを指定(update文)すればよいことになる。

例えば、一般ユーザ名として‘user01’をlocalhostに追加するには

```
mysql> insert into mysql.user(user,host) values('user01','localhost');
```

と指定し、パスワードは

```
mysql> update mysql.user set password=password('sist') where user='user01';  
mysql> flush privileges;
```

と指定する。ここでflush文はprivileges キャッシュの内容をデータベースファイルに保存し、反映させるために行っている。

これでuser01というユーザがログインできることになる。但し、このままではこのユーザには何も権限が与えられていないので、データベースを作ったりすることもできない。

```
$ mysql -u user01 -p
```

```
Enter password:
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
```

```
Your MySQL connection id is 9 to server version: 5.0.22
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql> create database newdb;
```

```
ERROR 1044 (42000): Access denied for user 'user01'@'localhost'  
to database 'newdb'
```

もしこのユーザにすべての権限を与えるのであれば、rootユーザでログインし、grant文を使って例えば

```
mysql> grant all on *.* to 'user01'@'localhost';
```

```
mysql> flush privileges;
```

と指定する。

以降はこうして作成した一般ユーザを使って作業を行うようにすること。

問題 2.4.1

1. 次の10人分のデータを meibo_table に登録せよ。

id	name	yomi
1	吉田茂	よしだしげる
2	鳩山一郎	はとやまいちろう
3	石橋湛山	いしばしたんざん
4	岸信介	きしのぶすけ
5	池田勇人	いけだはやと
6	佐藤栄作	さとうえいさく
7	田中角栄	たなかかくえい
8	大平正芳	おおひらまさよし
9	鈴木善幸	すずきぜんこう
10	海部俊樹	かいふとしき

2. 「福田康夫 (ふくだやすお)」を追加せよ。
3. id が 8, 9 のデータを次のように更新せよ。

id	name	yomi
8	三木武夫	みきたけお
9	福田赳夫	ふくだたけお