

ソフトウェア工学 データベースについて

2023/1/23

海谷 治彦

目次

- SQLについて
 - 理論的な話はデータベースの授業にて
- mysql の使い方
 - ウェブアプリで扱うデータの初期設定に、直接, mysql へアクセスすることも多い.

実行環境

- ここでのサンプルはWSL2上のUbuntu Linuxで動くことを想定しています.
- 特に, 動かさなくても結構ですが, 気になる方は, WSL2 と Ubuntu をセットアップしてください.
 - 別途, ムービー参照.
- DBのインストールとデモ, PHPからのDB操作については, ムービーを提示します.
 - それぞれ, 16分 3分

WSL2での apache php server起動

- sample09.zip の中に php.sh というファイルを準備しました.
- それを以下のように起動してください.
- WSL2 (Linux) 以外は, それぞれの方法でお願いします.

```
kaiya@KAIYA19LV8:~$ sudo ./php.sh
[sudo] password for kaiya:
+ sudo service apache2 start
* Starting Apache httpd web server apache2
*
+ service mysql start
* Starting MariaDB database server: mariadbd
+ service apache2 status
* apache2 is running
+ service mysql status
* /usr/bin/mysqldadmin Ver 9.1 Distrib 10.3.31-MariaDB, for debian-linux-gnu on x86_64
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Server version      10.3.31-MariaDB-0ubuntu0.20.04.1
Protocol version    10
Connection          Localhost via UNIX socket
UNIX socket         /var/run/mysqld/mysqld.sock
Uptime:             1 sec

Threads: 6  Questions: 4  Slow queries: 0  Opens: 20  Flush tables: 1  Open tables: 11  Queries per second avg: 4.000
kaiya@KAIYA19LV8:~$ ps aux
```

SQL自体について

- データベースの種類は沢山あるが、現在では、**RDB** (Relational Data Base)が主流である.
- **SQL**はRDBを操作するための言語である.
- RDBを管理するためのソフトウェアである **DBMS** (Data Base Management System) も何種類かの実装があるが、本講義では mariadb を使う.
- 講義資料でも若干の混乱があるが,
 - **DB** (Data base) 実際のコンテンツを含むデータ
 - DBMS DBを管理するためのソフトという用語分けをする.
- 通常、一つのDBMSは複数のDBを管理する.
- RDBでは、一つのDBに複数の**表(Table)**が含まれ、この表が、具体的な操作対象となる.

```
mysql> select * from NameScore;
+-----+-----+
| name  | score |
+-----+-----+
| a     | 40    |
| b     | 90    |
| ohnishi | 90    |
+-----+-----+
3 rows in set (0.00 sec)
```

表(Table)
の例

mysqlの利用法とSQL構文

- ウェブアプリ運用の準備のため、DBMSを直接操作する必要がある。例えば、
 - データベースを事前に作っておく。
 - 手動で不要なデータ削除等を行う。
 - バックアップをとる、等
- ここでは、mysqlコマンドを用いて、このような準備を行うための解説を行う。
- 加えて、SQLの典型的な構文についても復習する。

基本的な二つのツール

- 双方, コマンドラインのツールである.
- GUIのツールもあるかもしれないが, 本講義では扱わない.
- 基本, OSの一般ユーザーで実行してよい.
- `mysqladmin`
 - `mysql server` (DBMS)を管理するためのソフト.
 - `mysql`における管理者(`root`)のパスワード設定くらいにしかならないので, 本講義では不用.
- `mysql`
 - `mysql server` を操作するためのツール.
 - `mysql`固有の命令もあるが, 基本, 一般的なSQL命令を受け付ける.

mysql内でのユーザー管理

- mysqlでは、動作するOSのユーザー管理とは関係なく、**独自のユーザー管理**が行われている。
- 通常のUNIX系OS同様、**管理者(root)**で何でもできるが、実運用はrootで行うのは好ましくない。
- 各アプリケーション用のユーザーを作成し、ユーザー毎に、必要最低限の権限を与えるのがよい。
 - ユーザー毎に使えるデータベースを準備するのが、一番、シンプルでしょう。
- mysqlでの管理者 = root (UNIX系OSと同じ名前)
- 一般ユーザー 16文字以下の英数文字

新規ユーザーの登録と確認

- rootのパスワードは rootpasu になっているとする.
- 以下の例ではsaekiさんでパスワードはjindai
- 青字の部分は解説.
- 下記ではsaekiに特段権限が与えられてないので, 新しいDBもtableも作れない.
- ただし, testというDBに対しては操作ができる.

```
root@Linuxから接続する 別に一般@Linuxからでもよい
# sudo mysql -u root -p
rootのパスワードを入れる mysqlのプロンプトになる
mysql> CREATE USER 'saeki'@'localhost'; コレでユーザーsaekiができる
以下でsaekiのパスワードを設定, PASSWORD()は暗号化関数.
mysql> SET PASSWORD FOR 'saeki'@'localhost' = PASSWORD('jindai');
mysql> select user,host from mysql.user; コレで既登録ユーザーを列挙できる
mysql> exit とりあえずmysqlから抜ける
```

rootでデータベースを用意

- 一般ユーザーに全権限を与えるのはやなので, とりあえず, とあるデータベース限定の権限を与える.
- そのため, rootで権限を与えるためのDBを作っておく.
- ついでに, 既存のDBを閲覧.

root@Linuxから接続する 別に一般@Linuxからでもよい

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

rootのパスワードを入れる

```
mysql> show databases;
```

データベースを表示 デフォで information_schema mysql test の3つがある.

```
mysql> show tables from mysql;
```

データベースに含まれるテーブルが列挙される 上記はmysql中のもの

```
mysql> create database test2;
```

とりあえずtest2というDBを作った.

```
mysql> exit; とりあえず抜ける
```

DBアクセス許可を一般ユーザーに

- さきほど作ったDB test2 に対して saeki がアクセスできるように、設定を行います。
- これも mysqlのrootで行う。

```
root@Linuxから接続する 別に一般@Linuxからでもよい
# sudo mysql -u root -p
mysql> grant all privileges on test2.* to 'saeki'@'localhost';
データベース test2に対するフルアクセスをsaekiに許可した.
mysql> exit; とりあえず抜ける
```

```
saekiでmysqlに入り直す
# mysql -usaeki -pjindai
mysql> use test2; 使うDBをtest2に設定
mysql> create table twonum (a integer, b integer);
適当なテーブル twonum を作成できることを確認した.
mysql> exit; とりあえず抜ける
```

これで、
ユーザー saeki が、
データベース test2 を
操作できるように
お膳立てできた。

SQLの主な命令

- CREATE TABLE
- SELECT
- INSERT
- UPDATE
- DELETE
- SHOW DATABASES, SHOW TABLES, DROP
- その他, 色々ありますが, データベースの授業で習ってください.
 - データベースの授業ではシステムの実習はしないようです.

mysqlで使える主なデータ型

- INT, FLOAT, DOUBLE 整数 実数
- CHAR(数値) 固定長文字列, 数値は文字数, 0～255
- VARCHAR(数値) 可変長文字列, 数値は最近は文字数, 2^{16} バイトまで.
- TEXT $2^{16} - 1$ の可変長文字列
- 他はデータベースの授業にて.

CREATE TABLE

- データを保持するためのテーブルを作成.
- 下記の例のように 表中のフィールド名と型を列挙してテーブルを定義.
- 構造体っぽいよね.

saekiでmysqlに入り直す

```
# mysql -usaeki -pjindai
```

```
mysql> use test2; 使うDBをtest2に設定
```

```
mysql> create table NameScore (name text, score int);
```

```
mysql> exit; とりあえず抜ける
```

INSERT

- あるテーブルにデータを追加する操作.
- 下記の例通り.
- コレはおそらくJavaやPHPから操作することのほうが多いので, 直接, mysqlツールで実行することは少ないだろう.

saekiでmysqlに入り直す

```
# mysql -usaeki -pjindai
```

```
mysql> use test2; 使うDBをtest2に設定
```

```
mysql> insert into NameScore values('ohnishi', 90);
```

```
mysql> exit; とりあえず抜ける
```

UPDATE

- あるテーブルにデータを更新する操作.
- 下記の例通り.
- コレもおそらくJavaやPHPから操作することのほうが多いので, 直接, mysqlツールで実行することは少ないだろう.

saekiでmysqlに入り直す

```
# mysql -usaeki -pjindai
```

```
mysql> use test2; 使うDBをtest2に設定
```

下記では, name が kato さんの score を, 75に更新している.

```
mysql> update NameScore set score = 75 where name = 'kato';
```

```
mysql> exit; とりあえず抜ける
```

DELETE

- あるテーブルにデータを削除する操作.
- 下記の例通り.
- コレもおそらくJavaやPHPから操作することのほうが多いので, 直接, mysqlツールで実行することは少ないだろう.

saekiでmysqlに入り直す

```
# mysql -usaeki -pjindai
```

```
mysql> use test2; 使うDBをtest2に設定
```

下記では, name が kato さんのデータを削除している.

```
mysql> delete from NameScore where name = 'kato';
```

```
mysql> exit; とりあえず抜ける
```

SELECT

- あるテーブルにデータを列挙する命令.
- 下記の例通り.
- コレもおそらくJavaから操作することのほうが多いが, mysqlツールで確認する機会も多いかも.

saekiでmysqlに入り直す

```
# mysql -usaeki -pjindai
```

```
mysql> use test2; 使うDBをtest2に設定
```

```
mysql> select * from NameScore;
```

```
mysql> exit; とりあえず抜ける
```

```
mysql> select * from NameScore;
```

name	score
a	40
b	90
ohnishi	90

```
3 rows in set (0.00 sec)
```

SHOW, DROP

- show databases アクセス可能なデータベースを列挙.
- show tables あるDBにあるテーブルを列挙する.
- drop テーブルを削除する.

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| test     |
| test2    |
+-----+
3 rows in set (0.00 sec)

mysql> show tables from test2;
+-----+
| Tables_in_test2 |
+-----+
| NameScore       |
| twonum          |
+-----+
2 rows in set (0.00 sec)
```

```
mysql> use test2;
Database changed
mysql> drop table twonum;
Query OK, 0 rows affected (0.00 sec)

mysql> show tables
-> ;
+-----+
| Tables_in_test2 |
+-----+
| NameScore       |
+-----+
1 row in set (0.00 sec)

mysql> █
```

テーブルの確認 DESC

- 作成したテーブルのデータ構造を確認できる.

```
kaiya@jindai:~  
ファイル(F) 編集(E) 表示(V) 検索 (S) 端末(T) ヘルプ(H)  
mysql> desc nameage;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type   | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| name  | text   | YES  |     | NULL    |       |  
| age   | int(11)| YES  |     | NULL    |       |  
+-----+-----+-----+-----+-----+-----+  
2 rows in set (0.00 sec)  
  
mysql> █
```

以上

合計 20分弱のDBとPHPからの
DB操作のムービーがあるので
見ておいてください.