mySQL

・概要

　データベース

・公式サイト

　dev.mysql.com

・必要となる知識

　Linux環境

　iMac OS X環境（コマンドライン）

・インストール方法

　mySQL for iMac OS X:

<http://dev.classmethod.jp/etc/mac_mysql_install/>

(**Homebrewのインストールが必要)**

・役に立つサイト：

<http://dotinstall.com/lessons/basic_mysql_v2>

**（一部のレッスンはプレミアム会員が要り）**

データベースの構成：

データベース（スプレッドシート）

　ーテーブル（その中のシート）

　　ーフィールド（列）

　　ーレコード（行）

mySQLのコード(データベース編)：

mySQLサーバの起動：

$ mysql.server start

mySQLサーバの停止：

$ mysql.server stop

mySQLデータベースへのログイン：

$ mysql -u root

mySQLのパスワードを設定する方法：

mysql> set password for root@localhost=password(‘データベースのパスワード’);

mySQLを終了させる方法：

mysql> exit;

mySQLのパスワード付きログイン：

$ mysql -u root -p

(パスワードが聞かれます)

新しいデータベースの作成：

mysql> create database データベース名;

データベース一覧を見る：

mysql> show databases;

データベースの削除：

mysql> drop database データベース名;

使用するデータベースの切り替え：

mysql> use データベース名;

作業ユーザーを設定する：

例：

mysql> create database test;

mysql> grant all on test.\* to dbuser@localhost identified by ‘データベースのパスワード’;

という方法で一般ユーザとしてデータベースにアクセスすることができます。

mySQLのコード(フィールド編)：

新しいテーブルの作成：

mysql> create table テーブル名 (

id int,

neme varchar(255),

email varchar(255),

password char(32)

);

テーブル一覧を見る：

mysql> show tables;

テーブルの削除：

mysql> drop table テーブル名;

テーブルの構造を見る：

mysql> desc テーブル名;

例：

create table user (

id int,

neme varchar(255),

email varchar(255),

password char(32),

score double,

sex enum(‘male’, ‘female’),

memo text,

created date time

);

扱えるデータ型について：

・数値

-- int

-- double

・文字列

-- char

-- varchar

-- text

・日付

-- date

-- datetime

・それ以外

-- enum

フィールドの高度な設定：

例：

create table users (

id int not null auto\_increment primary key,

name varchar(255),

email varchar(255) unique,

password char(32),

score double,

sex enum('male', 'female') default 'male',

memo text,

created datetime,

key score (score)

);

・入力を必須

not null

・デフォルト値

default ‘male’

・自動連番

auto\_increment

・索引（インデックス）

ー主キー primary key

ーキー key 新しいキーの名前 (キーの名前)

ーユニークキー unique

mySQLのコード(レコード編)：

例：

create table users (

id int not null primary key auto\_increment,

name varchar(255),

email varchar(255),

team enum('blue', 'red', 'yellow'),

score double,

created datetime

);

insert into users (name, email, team, score, created) values

('taguchi', 'taguchi@dotinstall.com', 'blue', 5.5, '2012-05-11 11:00:00'),

('fkoji', 'fokoji@dotinstall.com', 'yellow', 8.2, '2012-06-21 12:00:00');

データの中身を入れる：

例：

insert into users (name, email, password, score, memo, created) values ('taguchi', ['taguchi@gmail.com](mailto:%27taguchi@gmail.com)', 'xxx', 5.5, 'memoeomeoe', '2012-06-12 11:00');

書式：

mysql> insert into テーブル名 (name, email, password, score, memo, created) values ('taguchi', ['taguchi@gmail.com](mailto:%27taguchi@gmail.com)', 'xxx', 5.5, 'memoeomeoe', '2012-06-12 11:00');

レコードを見る：

mysql> select \* from テーブル名;

（select文はフィールドを抽出することができます、’＊’ではなく、具体的にフィールド名を指定することができます）

例：

①　mysql> select \* from users;  
 ②　mysql> select name, email from users;

フィールドを多すぎる時に、使えるコマンド：

\G //最後のセミコロンは要らない

例：

mysql> select \* from users \G

条件付きで抽出する：

レコードの条件付き：

例：

①　mysql> select \* from users where score >= 5.0;

②　mysql> select \* from users where score = 5.5;

③　mysql> select \* from users where ccore != 5.5;

④　mysql> select \* from users where score <> 5.5;  
 ⑤　mysql> select \* from users where created > '2012-06-01 11:00:00';

⑥　mysql> select \* from users where email like ['%@dotinstall.com](mailto:%27%@dotinstall.com)';

（％は正規表現の１文字以上があること）

⑦　mysql> select \* from users where email like '%@dotinstall.\_\_';

（＿は正規表現の１文字だけがあること）

⑧　mysql> select \* from users where score between 5.0 and 8.0;

（between命令は指定の範囲内のレコードを抽出することができます）

⑨　mysql> select \* from users where team in ('red', 'yellow');

（in命令は項目に当てはまる項目のレコードを抽出する）

⑩　mysql> select \* from users where score >= 4.0 and team = 'blue';

（and命令）

　　mysql> select \* from users where score >= 4.0 or team = 'blue';

（or命令）

(WHERE命令を使って、条件を指定することができます)

レコードの並び替えと件数の制限：

昇順に並び替え：

mysql> select \* from users order by score;

（文字列も並び替えできます）

降順に並び替え：

mysql> select \* from users order by score desc;

（文字列も並び替えできます）

上から３位までを抽出する：

mysql> select \* from users limit 3;

開始位置を指定してN位を抽出する：

mysql> select \* from users limit 2, 2;

（開始位置は０から数えます）

並び替えと件数制限を組み合わせ：

mysql> select \* from users order by score desc limit 3;

データの集計：

レコードの件数を集計する：

mysql> select count(\*) from users;

ユニックの値だけを引っ張ってくる：

mysql> select distinct team from users;

最大値を抽出する：

mysql> select max(score) from users;

平均値を抽出する：

mysql> select avg(score) from users;

合計値を抽出する：

mysql> select sum(score) from users;

グループ集計して、平均値を抽出する：

mysql> select team,avg(score) from users group by team;

ランダムの値を出す：

select rand();

ランダム１つレコードを抽出する：

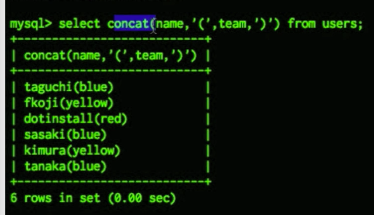
select \* from users order by rand() limit 1;

文字列、日付関数：

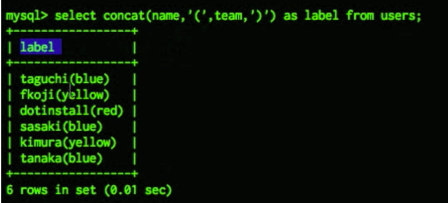
文字列の桁数を調べる：

mysql> select email,length(email) frim users;

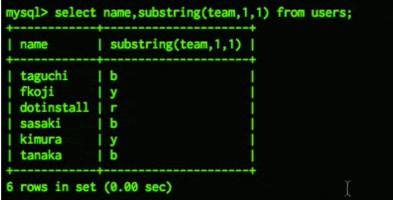
文字列を連結する：

mesql> select concat(name,'(', team, ')') from users;

文字列を連結して、新しい名前を作る：

mysql> select concat(name,'(', team, ')') as label from users;

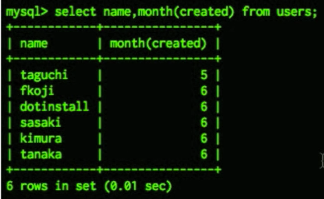
文字列の部分を表示する：

mysql> select name, substring(team, 1, 1) from users;

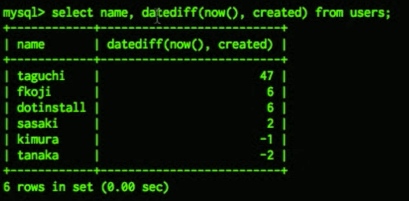
日付時刻を表示する：

mysql> select now();

月だけ知りたい：

mysql> select name, month(created) from users;

現在時刻と比べて、何日の差があるかを調べる：

select name, datediff(now(), created) from users;

レコードの更新、削除する：

レコードの更新：

mysql> update users set email = ['kimura@dotinstall.jp](mailto:%27kimura@dotinstall.jp)' where id = 5;

（whereでレコード指定するのは一般的です）



レコードの削除する：

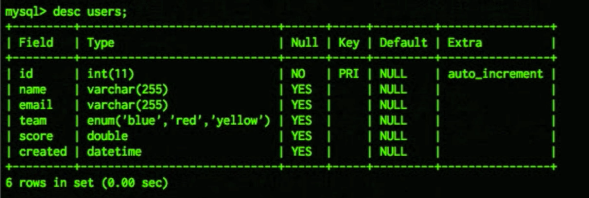
mysql> delete from where score <= 3.0;

（whereでレコード指定するのは一般的です）



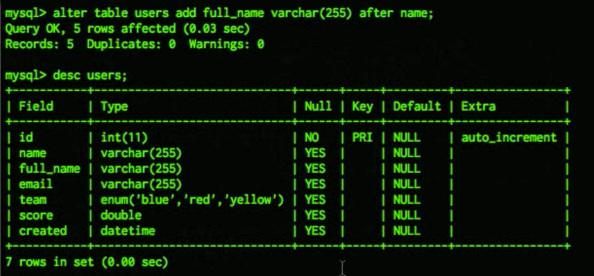
デーブルの構造を変更する：

フィールドの型を一覧で見る：

mysql> desc users;

フィールドの追加：

mysql> alter table users add full\_name varchar(255) after name;



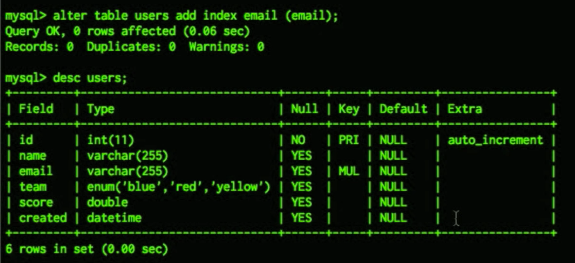
フィールドの変更：

mysql> alter table users change full\_name full\_name varchar(100);

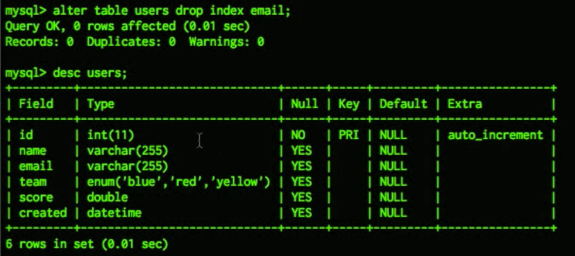
フィールドの削除：

mysql> alter table users drop full\_name;

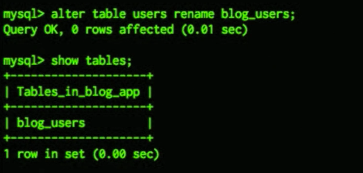
フィールドにインデックスを付ける：

mysql> alter table users add index email (email);

付けたインデックスを削除する：

mysql> alter table users drop index email;

テーブルの名前を変える：

mysql> alter table users rename blog\_users;

複数のテーブルを扱う：

使うデータ：

create table users (

id int not null primary key auto\_increment,

name varchar(255),

email varchar(255),

team enum('blue','red','yellow'),

score double,

created datetime

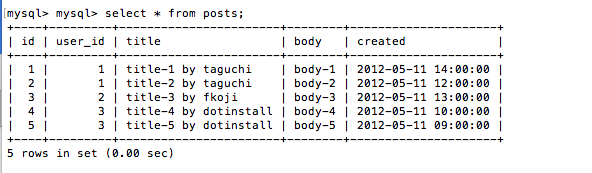
);

insert into users (name,email,team,score,created) values

('taguchi', 'taguchi@dotinstall.com', 'blue', 5.5, '2012-05-11 11:00:00'),

('fkoji', 'fkoji@dotinstall.com', 'yellow', 8.2, '2012-06-21 12:00:00'),

('dotinstall', 'dotinstall@dotinstall.com', 'red', 2.3, '2012-06-21 13:00:00');



create table posts (

id int not null primary key auto\_increment,

user\_id int not null,

title varchar(255),

body text,

created datetime

);

insert into posts (user\_id,title,body,created) values

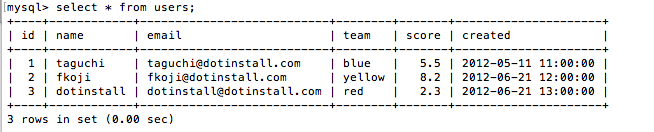
(1, 'title-1 by taguchi', 'body-1', '2012-05-11 14:00:00'),

(1, 'title-2 by taguchi', 'body-2', '2012-05-11 12:00:00'),

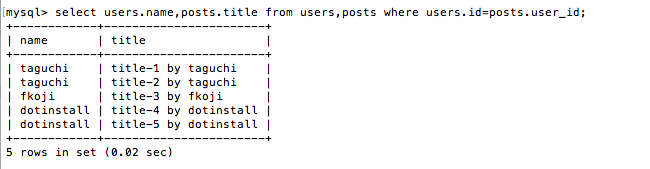
(2, 'title-3 by fkoji', 'body-3', '2012-05-11 13:00:00'),

(3, 'title-4 by dotinstall', 'body-4', '2012-05-11 10:00:00'),

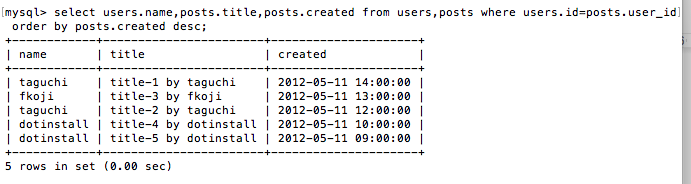
(3, 'title-5 by dotinstall', 'body-5', '2012-05-11 09:00:00');



複数のテーブルから抽出する：

mysql> select users.name,posts.title from users,posts where users.id=posts.user\_id;

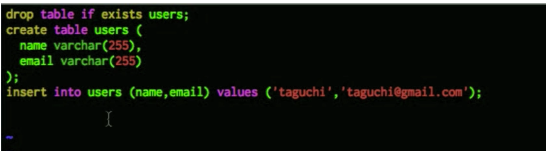
order byと組み合わせ：

mysql> select users.name,posts.title,posts.created from users,posts where users.id=posts.user\_id order by posts.created desc;

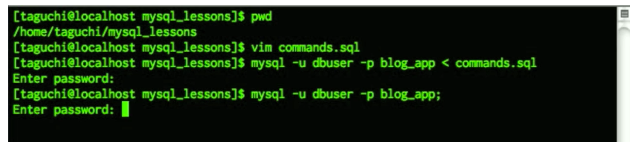
外部ファイルのコマンドを実行する：

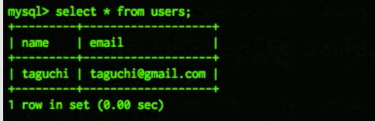
外部ファイルを作成：

$ vim commands.sql



（:wqで保存する）

$ mysql -u dbuser -p blog\_app < commands.sql  
 



バックアップと復元：

$ mysqldump -u dbuser -p blog\_app > blog\_app.dump.sql

（これでバックアップファイルが作成しています）

$ mysql -u dbuser -p blog\_app < blog\_app.dump.sql

（これでデータを復元することができます）