>mysql -u root -p

>show databases;

>create database(table) database(table)\_name;

>use database\_name;

>create table table\_name(id int primary key auto\_increment, name varchar(10) not null unipue)

>drop table(database) table(database)\_name;

>alter table table(column)\_name rename to new\_table(column)\_name;

>alter table table\_name add column\_name varchar(64)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| usersテーブル | | |  | companiesテーブル | |
| id | name | company-id |  | id | name |
| 1 | 津田 | 1 |  | 1 | パナソニック |
| 2 | 原田 | 2 |  | 2 | ソニー |
| 3 | 中西 | 4 |  | 3 | 日立 |

(内部結合)>SELECT \* FROM users INNER JOIN companies ON users.company-id = companies-id

※両方にあって合体できるデータのみを取り出す

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | name | company-id | id | Name |
| 1 | 津田 | 1 | 1 | パナソニック |
| 2 | 原田 | 2 | 2 | ソニー |

(外部結合)＞SELECT \* FROM users LEFT OUTER JOIN companies ON users.company-id = companies-id

※どちらかのテーブルにデータがあれば取り出す

※ハッシュ化、SALT

Python-import : <https://genchan.net/it/programming/python/4949/>

INSERT INTO table\_name (id, name, pass) VALUES (0, ‘name’,’pass’)

リポジトリ作成> git init

インデックスに登録> git add ファイル名

ローカルリポジトリにコミット＞ git commit -m ‘add new file’

リモートリポジトリの状態確認＞ git state

リポートリポジトリの情報を追加＞ git remote add origin リモートリポジトリのURL

リモートリポジトリにプッシュ＞　git push origine master

WSGI

サーバー上ではwebサーバーとwebアプリケーションとの間で通信が発生しており、webアプリケーション毎にその通信内容が異なる(Django,flask,bottle)など

Webサーバーとwebアプリケーションとの間に汎用WSGIを設置することで、その違いを許す。