Todo Application

- Tasks associated with a given user
- Hash of Passwords of the users
- Details associated with the tasks: priority, deadline, description

Why Database?

- As application is used by lot of users, we need some sort of persistent storage
- If we just rely on the memory, we won't be able to store huge amount of data
- If we store it in a code file, only when the code/application gets run, the data will be accessible.

Database

- Retrive data efficiently (SELECT)
- Query data effectively
- Make changes to the data effectively (UPDATE)
- Adding new data (CREATE)

Types

- SQL => Structured Query Language, data is stored in tables
- NoSQL => Not Only SQL, MongoDB, data is like json tables

SQL => uses some of B-Tree DS to store the data

SQLite3 => simpler version of SQL which stores data in a single file

CRUD operations => CREATE, READ, UPDATE, DELETE

Commonly used terms:

1. Schema => Design or the structure of the database, what tables, columns in those tables
2. State / Instance of the DB => rows in the tables, actual data

Tasks table id | title | description | deadline | priority (0/1/2)

1 | task 1| learn django models | 24th Feb 2021 | 0

Run SQLite3 shell

sqlite3 db_file_name

Queries

Create

CREATE TABLE tasks (id INT, title VARCHAR(50), desc VARCHAR(255), deadline DATETIME);

`.schema` => schema of your entire DB

`.schema table_name` => schema of your current table

`.table` => existing tables in your DB

Insert into table

INSERT INTO tasks VALUES (1, "Django", "Learn Django Models", "25-02-2021");

Query / Show data

SELECT * from tasks;

SELECT title from tasks;

Django Models

From next session