Django Models and Django Admin

sqlite3

C: Create R: Read U: Update D: Delete

.schema => tables and their columns, data is separate

.tables

Trello App

1. List
2. Tasks

- Requirements
    - User can add lists => scaler, college (user => next sessions)
    - A list can have multiple tasks
    - One task belongs to just one list
    - Each task can have => name, description, due_date
    - Each list can have => name, created_at
- Database schema
    - nouns: tables
    - detect attributes
    - relationships
- Tables
    - Task: id, name, description, due_date, created_at, list_id
    - TaskList: id, name, created_at
- Relationships
    - one to one: one task belongs to one list
    - many to one: many tasks belongs to one list
    - one to many: one list has many tasks

  Above can be done using foreign key
    - many to many

  Extra table to required to store this.
- Primary Key In a class, student id, roll no unique identifier
- Foreign Key Relating to primary key of some other table

Changes to models

1. Create a new DB, move your previous data to new DB by manual entry => hefty
2. Prev DB schema is known, current models, compare them to write a migration script manually => everytime you will make changes, you need to create a new script
3. general script that identifies the difference in old DB schema and the current models, generate a migration script => Django provides us out of the box

Migration

- python manage.py makemigrations => generates migration script
- python manage.py migrate => run script

Django Admin

1. Create super user python3 manage.py createsuperuser
2. Register our models to django admin