1. Overview

2. DB Drivers

2.1 psycopg2 - used to connect to the DB

```
pip install psycopg2
```

3. Django ORM(Object Relational Mapping)

```
class test_table(models.Model:
      test = models.CharField()
      count = models.IntegerField()

                  --->

test_table
id          int
test  varchar
cout  int

class Task(models.Model):
      name = models.CharField(max_length = 100)
      project = models.CharField( max_length = 50)

from tasks import models
models.Task
models.Task.objects.create(name='Test Project', project='First Project')

pip install Django
django-admin startproject my_first_project
pip freeze > requirements.txt
python manage.py --help
python manage.py runserver
python manage.py makemigrations
python manage.py migrate
python manage.py createsuperuser
python manage.py startapp tasks
python manage.py showmigrations - lists both applied and unapplied migrations
```

3.1. Popular ORM Tools for Python:

- Django
- web2py - an open source full-stack Python framework
- SQLObject - an ORM that provides an object interface to your DB
- SQLAlchemy - provides persistant patterns designed for efficient and high-performing DB access
- dbshell - generate SQL

3.2. Migrations - used to apply changes to our tables in the DB

4. Django dbshell

```
python manage.py dbshell
\dt - shows all tables in the current DB
\d <table_name> - shows a specific table
SELECT * FROM <table_name>
```

################

ADDITIONAL INFO:

refactoring.guru/design-patterns