Object Relational Model(ORM)

* What is ORM?
* An Object relational model is a combination of a Object oriented database model and a Relational database model. So, it supports objects, classes, inheritance etc. just like Object Oriented models and has support for data types, tabular structures etc. like Relational data model.

**Flask**

SQLAlchemy:

* In default Flask use SQLAlchemy to create object relational model
* How SQLAlchemy works?
* SQLAlchemy is a library that facilitates the communication between Python programs and databases. Most of the times, this library is used as an Object Relational Mapper (ORM) tool that **translates Python classes to tables on relational databases and automatically converts function calls to SQL statements**.

For more details visit this site - <https://auth0.com/blog/sqlalchemy-orm-tutorial-for-python-developers/>

**Django**

In default Django has its own ORM

* How Django ORM Works?
* The Django web framework includes a default **object-relational mapping layer** (ORM) that can be used to interact with application data from various relational databases such as SQLite, PostgreSQL and MySQL. The Django ORM is an implementation of the object-relational mapping (ORM) concept.

For More details to see about Django ORM visit here –

<https://www.fullstackpython.com/django-orm.html#:~:text=The%20Django%20web%20framework%20includes,relational%20mapping%20(ORM)%20concept>.

Djongo

Djongo provides a unified approach to database interfacing. It is an extension to the traditional [Django ORM](https://www.djangoproject.com/) framework. It maps python objects to MongoDB documents, a technique popularly referred to as Object Document Mapping or ODM.

Comparing to PyMongo, Django is much easier to use queries, and more reliable