

Object Relational Mapper (ORM)

Object-Relational Mapper (ORM), which enables application to interact with database such as SQLite, MySQL, PostgreSQL, Oracle.

ORMs automatically create a database schema from defined classes or models. It generate SQL from Python code for a particular database which means developer do not need to write SQL Code.

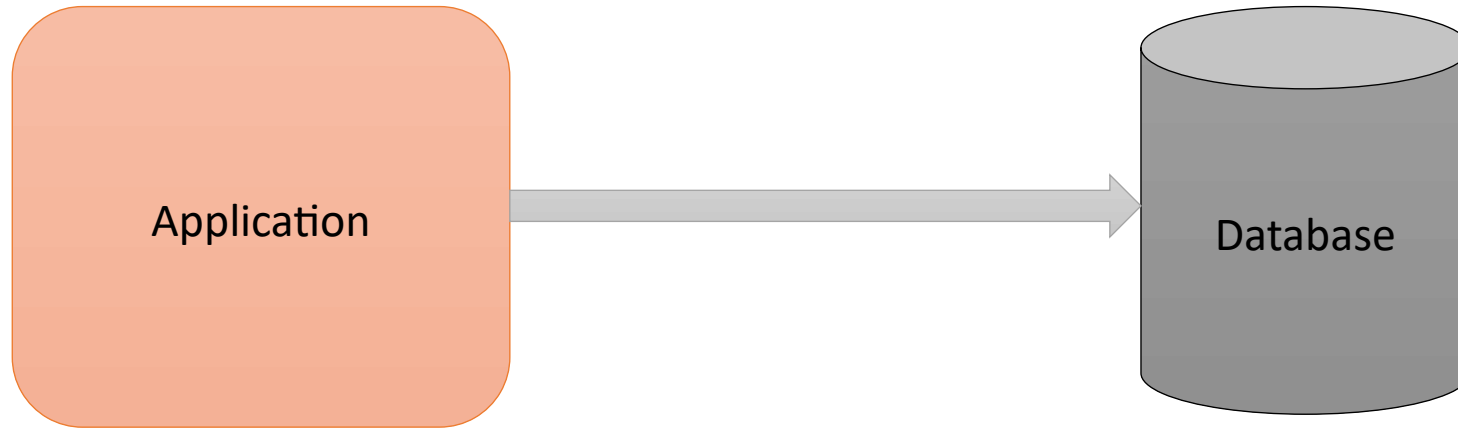
ORM maps objects attributes to respective table fields.

It is easier to change the database if we use ORMs hence project becomes more portable.

Django's ORM is just a way to create SQL to query and manipulate your database and get results in a pythonic fashion.

ORMs use connectors to connect databases with a web application.


Object Relation Mapper (ORM)



Object Relation Mapper (ORM)

```
class Student(models.Model):  
    stuid=models.IntegerField()  
    stuname=models.CharField(max_length=70)  
    stuemail=models.EmailField(max_length=70)  
    stupass=models.CharField(max_length=70)
```

```
CREATE TABLE "enroll_student" (  
    "id" integer NOT NULL PRIMARY KEY  
    AUTOINCREMENT,  
    "stuid" integer NOT NULL,  
    "stuname" varchar(70) NOT NULL,  
    "stuemail" varchar(70) NOT NULL,  
    "stupass" varchar(70) NOT NULL  
);
```



id	stuid	stuname	stuemail	stupass

QuerySet

A QuerySet can be defined as a list containing all those objects we have created using the Django model.

QuerySets allow you to read the data from the database, filter it and order it.