ORMS

* ORM stands for object relational mapper.
* It maps my model objects into records in the database.
* It creates all the necessary queries for us so we don’t have to write them from scratch.
* It does for us the creation, manipulations and adds relations between objects.
* The advantage is that it reduces the amount of code writing.
* It reduces complexity.
* Save time.

DJANGO MANAGER AND QUERY SET

DJANGO MANAGER

* Every Django model have a manage.
* To access that manager type: ModelName.objects
* The objects attribute is a model or let’s say it’s an interface that help me interact with database
* It gives me access to bunch of methods:
  + Objects.all() return the list of records for that model
  + Objects.get() return a single recod
  + Objects.filter() to filter the result.

QUERY SET

* Most of the manager methods return a query\_set.
* The query\_set does not return the records list, but it only encapsulates the set.
* The Django will execute the query inside the query\_set when I evaluate it.
  + These scenarios are:
    - Convert it to list:
      * List(query\_set)
    - Looping over the query\_set.
    - Access specific record in list or slice some of it:
      * query\_set[0]
      * query[0:15]
* The reason Django does not return for me the records and only return an encapsulated query\_set because I might want to run complex query.
  + For example I want to query\_set.filter().order\_by().sort()
  + Because I might have millions of results, and I want to filter it.
  + I don’t have to run a query on millions of records, I want it filterd before that.

Retrieving objects:

* To get one item from database I use method .get()
  + Objects.get(id=1).
  + Instead of id I cak use pk=1, the benefit of that is I don’t have to remember what is the primary key name.
  + This method return exception if item with id does not eixt. ObjectDoesNotExit.
  + The next method solves it.
* Objects.filter(pk=1): it get me a set based on provided value to filter, then I add method .first() to get first item.
* .first() gives me None of item with provided filter does not exit and it will not throw exception.
* If I want to check if it exist I can use .exists() and it return a Boolean.