

## ORM + Queryset methods

In this section we are having the first touch with querysets, we will learn more about them in the future videos. At the beginning of the next section we will learn how to generate dummy data so that we can practice writing querysets more in the later parts of the course. For now I'm leaving you with a short read on Django ORM and queryset methods:

ORM is a tool that allows developers to interact with a database using an object-oriented programming language (like Python in Django) instead of writing SQL queries directly. The ORM maps database tables to classes and database rows to objects, making it easier for developers to create, read, update, and delete data without having to worry about the underlying SQL. This allows developers to focus on the application logic rather than the database details, making development faster.

QuerySets are a way to retrieve and manipulate data from a database with Django's ORM to avoid writing SQL queries. Most popular queryset methods are:

- **filter()**: Returns a QuerySet containing objects that match the specific lookup parameters you provide.
- **exclude()**: Returns a QuerySet containing objects that don't match the specific lookup parameters you provide.
- **get()**: Returns a single object that matches the specific lookup parameters you provide.
- **first()**: Returns the first object in the QuerySet that you are currently working with, or None if the QuerySet is empty.
- **last()**: Returns the last object in the QuerySet that you are currently working with, or None if the QuerySet is empty.

- **order\_by()**: Returns a QuerySet ordered by the specific field(s) that you specify.
- **values()**: Returns a QuerySet containing dictionaries for each object, with keys corresponding to the specific field(s) that you specify.
- **annotate()**: Returns a QuerySet with additional calculated fields, such as counts or averages, based on your specific requirements.
- **count()**: Returns the number of objects in the QuerySet that you are currently working with.
- **distinct()**: Returns a QuerySet with duplicate objects removed, based on your specific requirements.
- **exists()**: Returns True if the QuerySet contains any objects, or False otherwise, based on your specific requirements.
- **update()**: Updates one or more fields on all objects in the QuerySet, based on your specific requirements.
- **delete()**: Deletes all objects in the QuerySet that you are currently working with.
- **aggregate()**: Returns a dictionary of calculated values for the QuerySet, such as sums or averages, based on your specific requirements.
- **all()**: Returns a QuerySet of all model objects that you have access to.