ORM module 5 part2

Reference a field using F object:

* Sometimes I want to compare two fields in the same table or in a different table.
* And to do it I add field\_name = other\_field\_name, this will compare values for each field
* The f object compares these two for me by passing the column name using F(‘field\_name’)
* How to reference a field?
  + from Django.db.models import F
* wrap my field name with F object
  + F(‘field\_name’)
* This will give me a reference to that object.

Getting values instead of full object.

* What if I am interested in one or two values but not all columns’ values inside object?
* In this case I need to use values () method: Product.objects.values(‘value1’, ‘value2’)
  + It will give me result dictionary.
  + If I want result to be tuple instead of dictionary I use values\_list() method: Product.objects.values\_list(‘value1’, ‘value2’)
* To get a value in related tables I use double underscore then value: this will retrieve values in another field (make joins)
  + exampl: Product.objects.values\_list( ‘ value1‘, ‘value2’, ‘related\_table\_\_value’ )
* to get values of product with id in specific list of ids I use: id\_\_in
  + I can use it when I have relation between two tables, I make a filter based on given list of the shared column (foreign key)

Defer(‘value’): select object and defer specified values for later.

* I have to be careful with using this method.
* I might end up querying the database n times normal time, e.g., query the entire rows inside table.
* Let’s say I have queried all objects and render them on template (1000 record, I will get value related to that object, but each time Django will make a query and it will consume a lot of time)

only(‘value’)

joins and select\_related vs prefetrch\_related(‘value’)

if I get object and use its foreign key to get the values of that object Django will run a query to get that object and I might end up running millions of queries if it’s in for loop.

* To solve this, I use select\_related or prefetch related.

select\_related(‘object\_field’):

* I use this method to preload all the data related to that field (inner join based on common column, usually a fk).
* One field I mean is a relation like one to one or one to many.
* Django will preload all the fields related to that foreign key.

Prefetch\_related(‘objects’)

* I use it to preload all data related to list of values inside one field.
* That field is many to many relations with other table.
* It will preload all data related to values inside list.
* Both select and prefetch if table objects, they have relation with other tables objects I can query them:
  + Select\_related(order\_\_customer)

Aggregation in Django.

* Sometimes I want to do operations on objects, like counting, finding min value, max value, average and sum.
* It will return for me a dictionary
* To do that I need to import first the aggregate classes from aggregation parent
  + From Django.db.model.aggregate import Count, Sum, Avg, Max, Min
* Then I call. aggregate () on a query set.
* Then I pass the class child.
  + Aggregate (count= Count(‘id’))
    - It will count for me total number of objects and return single value and it will get me all objects because it’s the id,
    - If it happened to call different field, it will reduce them based on field.
  + Aggregate (Sum (‘unit\_price’)). Or Aggregate (Sum (‘quantity))
    - It will count for me all values and return a single value
  + Aggregate (Avg (‘unit\_price’))
    - It will count for me all data average
  + Aggregate (min (‘unit\_price’))
    - Will give me minimum value in unit\_ price
  + Aggregate (max (‘unit\_price’))
    - It will give me maximum value in unit price.
* Since I can run it on a query set, I can add it to filtered data.
* I can change the value of the name in result by adding preferred name before calling method.