Creating, Inserting and Manupulating Entries in Database

Step 1: python manage.py sqlmigrate office 0001

Note- This will display the SQL code that dango generated using the model we created

```
class details(models.Model):
    first_name=models.CharField(max_length =30)
    last_name=models.CharField(max_length =30)
    age = models.IntegerField()
```

Step 2: python manage.py migrate

Note- This command tells django to look at all the migration that havent't been run yet throughout all the app and run remaining unexecuted migrations

Step 3: python manage.py shell

Note - This opens the shell terminal

Step 4: In [1]: from office.models import details

Note- This imports the our model (details) from the model we created above

Step 5: To create entry

```
In [7]: linc=details(first_name='linc', last_name='pathak', age=22)
```

We can check the about the data you created above befor commiting and saving it to database

```
In [8]: linc.age
Out[8]: 22
In [9]: linc.age>40
```

Out[9]: False

Step 6:

Step 6: To save and insert

In [10]: linc.save()

Step 7: To create and save in one step. It does both step 5 and step 6 in one line

In [11]: details.objects.create(first_name='chari', last_name='pari', age=20)

Out[11]: <details: details object (2)>

Step 8: Bulk entries Create multiple data

- First create a list of data you want to insert into the database

In [12]: my_list=[details(first_name='X', last_name='Y', age=44),
details(first_name='A',last_name='B',age=33)]

- Then create the object instances using that list

In [14]: details.objects.bulk_create(my_list)

Django Model Manager

Each model we creae comes with the Manger which allows us to create QuerySet which can be used to retireve entires from database. These all methods are used inside shell

Model provides bunch of methods:

- .all()
- .get()
- .filter()
- .exclude()

Step: Update model.py

```
from django.db import models

# Create your models here.

class details(models.Model):
    first_name=models.CharField(max_length =30)
    last_name=models.CharField(max_length =30)
    age = models.IntegerField()
    def __str__(self):
        return f"{self.last_name} is {self.age} years old."
```

Note- To get the data in human readable form goto the model.py and add a string function within your model

.all()

Step 1:

In [3]: details.objects.all()

Out[3]: <QuerySet [<details: pathak is 22 years old.>, <details: pari is 20 years old.>, <details: Y is 44 years old.>, <details: B is 33 years old.>]>

Note- This gives us all the entries that we created and saved in database

Step 2:

In [4]: details.objects.all()[1]

Out[4]: <details: pathak is 22 years old.>

Note- The index of entries in database starts from 1 in SQL, This returns the first entry we saved in database

.get()

Note- You can look the information about the model you creaded in 0001_initial.py file on migration folder. Get method throws error if you try to access multiple entries with it

Step 1:

In [5]: details.objects.get(pk=1)

Out[5]: <details: pathak is 22 years old.>

In [8]: details.objects.get(pk=2)

Out[8]: <details: pari is 20 years old.>

Note- We use get() method to get single item only so we use Primary Key (PK). Every time you put in new data entry there will be new id and primary key assined to each entries. So this allows us to use get() method to access unique entires.

Step:2

In [9]: details.objects.filter(last_name='pathak').get()

Out[9]: <details: pathak is 22 years old.>

Note- You can use filter() method with get() method as per requrement

.filter()

Lets first add one more data that matches the last_name ='pari' with another entry to see filter() method

In [10]: details.objects.create(first_name='meri', last_name='pari', age=21)

Out[10]: <details: pari is 21 years old.>

Step 1:

In [13]: details.objects.filter(last_name='pari').all()

Out[13]: <QuerySet [<details: pari is 20 years old.>, <details: pari is 21 years old.>]>

Note- This returns all the entries that have the last_name='pari'

Step 2:

In [14]: details.objects.filter(last_name='pari').filter(age=21).all()

Out[14]: <QuerySet [<details: pari is 21 years old.>]>

Note- Chaining multiple filter method together