May 03

How to create, update and delete records?

1st way: py manage.py shell

```
D:\Django_20MAR_7PM\ormproject1>py manage.py shell
>>> from testapp.models import Employee
>>> e = Employee(eno=1234,ename='Mahesh',esal=1234.0,eaddr='Vja')
>>> e.save() #This employee will be inserted into database
```

2nd way:

```
>>>
Employee.objects.create(eno=2345,ename='Kareena',esal=123.0,eaddr='Chen
nai')
```

How to add multiple records at a time:

By using method bulk_create

```
Employee.objects.bulk_create(
[Employee(eno=3333,ename='Sachin',esal=33333.0,eaddr='Mumbai'),
Employee(eno=6666,ename='Kohli',esal=66666.0,eaddr='Delhi'),
)
```

How to delete single record

```
>>> e = Employee.objects.get(eno=8888)
>>> e.delete()
(1, {'testapp.Employee': 1})
```

How to delete multiple records:

```
>>> qs = Employee.objects.filter(esal__gte=15000)
>>> qs.count() #26
>>> qs.delete()
(26, {'testapp.Employee': 26})
>>> qs.count() #0
```

How to delete all records:

```
>>> qs = Employee.objects.all()
>>> qs.delete()
          or
>>> Employee.objects.all().delete()
```

<u>How to update record:</u>

```
>>> e = Employee.objects.get(eno=6775)
>>> e.ename
>>> e.esal
>>> e.esal=23000
>>> e.save()
>>> e.ename='sunny'
>>> e.save()
```

<u>How to order queries in sorting order</u>

```
emp_list = Employee.objects.all()
```

1).To display all employees according to ascending order eno.

```
emp_list = Employee.objects.all().order_by('eno')
```

2).To sort all employees according to descending order eno.

```
emp_list = Employee.objects.all().order_by('-eno')
```

3).How to get highest salaried employee object?

Arrange all employees in descending order and select first employee.

```
>>> e = Employee.objects.all().order_by('-esal')[0]
>>> e.ename
>>> e.esal
```

4).To get all employees based on alphabatical order of names.

```
emp_list = Employee.objects.all().order_by('ename')
```

5).To ignore case?

```
from django.db.models.functions import Lower
emp_list = Employee.objects.all().order_by(Lower('ename'))
```

<u>How to perform union operations for query set:</u>

By using union operation, we can combine results of 2 or more queries from same model or from different models.

```
q1 = Employee.objects.filter(esal__lte=12000)
q2 = Employee.objects.filter(ename__startswith='S')
q3 = q1.union(q2)
emp_list = q3
```

CHAPTER-13

Working with Django Middleware

-->At pre processing of request or at post processing of request, if we want to perform any activity automatically then we should go for middleware.

http://127.0.0.1:8000
http://127.0.0.1:8000/

http://127.0.0.1:8000/agg
http://127.0.0.1:8000/agg/

submit the form--->csrf verification
AuthenticationMiddleware

http====>https===>SecurityMiddleware

-->Middleware is applicable for every incoming request and outgoing response.

Middleware Structure:

Based on our requirement, we can configure our own middleware also.
Every customized middleware is a python class and it is the child class of object.
class A(object):
class A:

This python class should contains 2-mandatory methods.
1).def __init__(self,get_response):
-->get_response is a function which can be used to send request to the next level and to get required response.
-->This method will be executed only once at the time of creating middleware class object, which is mostly happended at the time of server starting.

```
2).def __call__(self,request):
        This method will be executed for every request separately
        #code for preprocessing of request
        response = self.get_response(request) #Trigger request to the next
level
        #code for post processing of request.
        return response
```

Middleware classes we have to define middleware.py file(inside testapp)