**DJANGO ORM**

**Command line (Shell)**

PS C:\Users\kamal\django\_projects\passengerProject> py -m manage shell

Python 3.12.3 (tags/v3.12.3:f6650f9, Apr 9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license" for more information.

(InteractiveConsole)

>>> from passengerApp.models import Passengers

>>> qs = Passengers.object.all()

Traceback (most recent call last):

File "<console>", line 1, in <module>

AttributeError: type object 'Passengers' has no attribute 'object'. Did you mean: 'objects'?

>>> qs = Passengers.objects.all()

>>> print(type(qs))

<class 'django.db.models.query.QuerySet'>

>>> pas = Passengers.objects.get(id=1)

>>> print(type(pas))

<class 'passengerApp.models.Passengers'>

>>> print(pas)

Passengers object (1)

>>> print(pas.firstName)

Kamal

>>> print(qs.query)

SELECT `passengerApp\_passengers`.`id`, `passengerApp\_passengers`.`firstName`, `passengerApp\_passengers`.`lastName`, `passengerApp\_passengers`.`age`, `passengerApp\_passengers`.`gender`, `passengerApp\_passengers`.`travelFrom`, `passengerApp\_passengers`.`travelTo`, `passengerApp\_passengers`.`travelDate`, `passengerApp\_passengers`.`travelMode` FROM `passengerApp\_passengers`

>>>

>>> pas = Passengers.objects.filter(age\_\_gt=18)

>>> print(pas)

<QuerySet [<Passengers: Passengers object (1)>, <Passengers: Passengers object (2)>]>

>>> pas = Passengers.objects.filter(age\_\_gte=18)

>>> print(pas)

<QuerySet [<Passengers: Passengers object (1)>, <Passengers: Passengers object (2)>]>

>>> pas = Passengers.objects.filter(age\_\_gt=17)

>>> print(pas)

<QuerySet [<Passengers: Passengers object (1)>, <Passengers: Passengers object (2)>]>

>>> pas = Passengers.objects.filter(age\_\_gte=17)

>>> print(pas)

<QuerySet [<Passengers: Passengers object (1)>, <Passengers: Passengers object (2)>, <Passengers: Passengers object (4)>]>

>>> pas = Passengers.objects.filter(age\_\_lt=17)

>>> print(pas)

<QuerySet [<Passengers: Passengers object (3)>]>

>>> pas = Passengers.objects.filter(age\_\_lte=17)

>>> print(pas)

<QuerySet [<Passengers: Passengers object (3)>, <Passengers: Passengers object (4)>]>

>>> pas = Passengers.objects.get(firstName\_\_contains='Aug')

>>> print(pas)

Passengers object (2)

>>> pas = Passengers.objects.get(firstName\_\_contains='jer')

Traceback (most recent call last):

File "<console>", line 1, in <module>

File "C:\Users\kamal\AppData\Local\Programs\Python\Python312\Lib\site-packages\django\db\models\manager.py", line 87, in manager\_method

return getattr(self.get\_queryset(), name)(\*args, \*\*kwargs)

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

File "C:\Users\kamal\AppData\Local\Programs\Python\Python312\Lib\site-packages\django\db\models\query.py", line 649, in get

raise self.model.DoesNotExist(

passengerApp.models.Passengers.DoesNotExist: Passengers matching query does not exist.

>>> pas = Passengers.objects.get(firstName\_\_icontains='jer')

>>> print(pas)

Passengers object (3)

>>> pas = Passengers.objects.get(firstName\_\_startswith='J')

Traceback (most recent call last):

File "<console>", line 1, in <module>

File "C:\Users\kamal\AppData\Local\Programs\Python\Python312\Lib\site-packages\django\db\models\manager.py", line 87, in manager\_method

return getattr(self.get\_queryset(), name)(\*args, \*\*kwargs)

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

File "C:\Users\kamal\AppData\Local\Programs\Python\Python312\Lib\site-packages\django\db\models\query.py", line 652, in get

raise self.model.MultipleObjectsReturned(

passengerApp.models.Passengers.MultipleObjectsReturned: get() returned more than one Passengers -- it returned 2!

>>> pas = Passengers.objects.filter(firstName\_\_startswith='J')

>>> print(pas)

<QuerySet [<Passengers: Passengers object (3)>, <Passengers: Passengers object (4)>]>

>>> pas = Passengers.objects.get(firstName\_\_endswith='l')

>>> print(pas)

Passengers object (1)

>>> pas = Passengers.objects.filter(firstName\_\_startswith='Au') | Passengers.objects.filter(lastName\_\_endswith='ju')

>>> print(pas)

<QuerySet [<Passengers: Passengers object (2)>]>

>>> from django.db.models import Q

>>> pas = Passengers.objects.filter(Q(firstName\_\_startswith='Au') | Q(lastName\_\_endswith='ju'))

>>> print(pas)

<QuerySet [<Passengers: Passengers object (2)>]>

>>> pas = Passengers.objects.filter(Q(firstName\_\_startswith='Au') & Q(lastName\_\_endswith='xu'))

>>> print(pas)

<QuerySet []>

>>> pas = Passengers.objects.filter(firstName\_\_startswith='Ja',lastName\_\_endswith='ar')

>>> print(pas)

<QuerySet [<Passengers: Passengers object (4)>]>

<QuerySet [<Passengers: Passengers object (1)>, <Passengers: Passengers object (2)>, <Passengers: Passengers object (3)>, <Passengers: Passengers object (4)>]>

>>> pas = Passengers.objects.all().values\_list('firstName','lastName','travelDate')

>>> print(pas)

<QuerySet [('Kamal', 'Ranjan', datetime.date(2024, 5, 12)), ('Augusta', 'Raju', datetime.date(2024, 5, 12)), ('Jerrick', 'Kumar', datetime.date(2024, 5, 12)), ('Jazlyn', 'Kumar', datetime.date(2024, 5, 12))]>

>>> pas = Passengers.objects.all().values('firstName','lastName','travelDate')

>>> print(pas)

<QuerySet [{'firstName': 'Kamal', 'lastName': 'Ranjan', 'travelDate': datetime.date(2024, 5, 12)}, {'firstName': 'Augusta', 'lastName': 'Raju', 'travelDate': datetime.date(2024, 5, 12)}, {'firstName': 'Jerrick', 'lastName': 'Kumar', 'travelDate': datetime.date(2024, 5, 12)}, {'firstName': 'Jazlyn', 'lastName': 'Kumar', 'travelDate': datetime.date(2024, 5, 12)}]>

>>> pas = Passengers.objects.all().only('firstName','lastName','travelDate')

>>> print(pas)

<QuerySet [<Passengers: Passengers object (1)>, <Passengers: Passengers object (2)>, <Passengers: Passengers object (3)>, <Passengers: Passengers object (4)>]>

>>>

>>> from django.db.models import Avg,Sum,Max,Min,Count

>>> avg = Passengers.objects.all().aggregate(Avg('age'))

>>> print(avg)

{'age\_\_avg': 30.75}

>>> sum = Passengers.objects.all().aggregate(Sum('age'))

>>> print(sum)

{'age\_\_sum': 123}

>>> min = Passengers.objects.all().aggregate(Min('age'))

>>> print(min)

{'age\_\_min': 10}

>>> max = Passengers.objects.all().aggregate(Max('age'))

>>> print(max)

{'age\_\_max': 48}

>>> cnt = Passengers.objects.all().aggregate(Count('age'))

>>> print(cnt)

{'age\_\_count': 4}

>>>

**ADD NEW RECORD IN DATABASE USING SHELL ORM**

>>> from passengerApp.models import Passengers

>>> ps = Passengers(firstName='Test',lastName='Test',age=24,gender='test',travelFrom='test',travelTo='test',travelDate='2024-05-06',travelMode='test'

... )

>>> ps.save()

>>> Passengers.objects.all().count()

5

>>>

A screenshot of a computer screen

Description automatically generated

>>> Passengers.objects.create(firstName='Test1',lastName='Test1',age=24,gender='test1',travelFrom='test1',travelTo='test1',travelDate='2024-05-06',travelMode='test1')

<Passengers: Passengers object (6)>

>>> Passengers.objects.all().count()

6

>>>

A screenshot of a computer

Description automatically generated

**ADD BULK RECORDS IN DATABASE USING SHELL ORM**

>>> from passengerApp.models import Passengers

>>>Passengers.objects.bulk\_create([Passengers(firstName='Bulk1',lastName='Bulk1',age=28,gender='Male',travelFrom='Dublin',travelTo='Cork',travelDate='2024-10-06',travelMode='Train'),Passengers(firstName='Bulk2',lastName='Bulk2',age=29,gender='Female',travelFrom='Dublin',travelTo='Cork',travelDate='2024-05-06',travelMode='Train'),Passengers(firstName='Bulk3',lastName='Bulk3',age=35,gender='Male',travelFrom='Dublin',travelTo='Cork',travelDate='2024-05-06',travelMode='Train'),Passengers(firstName='Bulk4',lastName='Bulk4',age=42,gender='Female',travelFrom='Dublin',travelTo='Cork',travelDate='2024-05-06',travelMode='Train'),Passengers(firstName='Bulk5',lastName='Bulk5',age=48,gender='Male',travelFrom='Dublin',travelTo='Cork',travelDate='2024-05-06',travelMode='Train'),Passengers(firstName='Bulk6',lastName='Bulk6',age=58,gender='Female',travelFrom='Dublin',travelTo='Cork',travelDate='2024-05-06',travelMode='Train')])

[<Passengers: Passengers object (None)>, <Passengers: Passengers object (None)>, <Passengers: Passengers object (None)>, <Passengers: Passengers object (None)>, <Passengers: Passengers object (None)>, <Passengers: Passengers object (None)>]

>>> Passengers.objects.all().count()

12

>>>

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**DELETE RECORDS IN DATABASE USING SHELL ORM**

>>> from passengerApp.models import Passengers

>>> ps = Passengers.objects.get(id=5)

>>> ps.delete()

(1, {'passengerApp.Passengers': 1})

>>>

A screenshot of a computer screen

Description automatically generated

**DELETE BULK RECORDS IN DATABASE USING SHELL ORM**

>>> ps = Passengers.objects.filter(firstName\_\_contains='Test')

>>> ps.delete()

(3, {'passengerApp.Passengers': 3})

>>>

A screenshot of a computer

Description automatically generated

>>> ps = Passengers.objects.all().delete() 🡺 Delete all the records of the table.

**UPDATE RECORDS IN DATABASE USING SHELL ORM**

Added 1 record based on the value with Test1.

A screenshot of a computer

Description automatically generated

<Passengers: Passengers object (15)>

>>> ps = Passengers.objects.get(id=15)

>>> ps.firstName

'Test1'

>>> ps.lastName

'Test1'

>>> ps.firstName='updatedName'

>>> ps.save()

>>> ps.lastName='updatedName'

>>> ps.save()

>>> ps.firstName

'updatedName'

>>> ps.lastName

'updatedName'

>>>

A screenshot of a computer

Description automatically generated

**SORT RECORDS IN DATABASE USING SHELL ORM**

>>> pas=Passengers.objects.all()

>>> pas

<QuerySet [<Passengers: Passengers object (1)>, <Passengers: Passengers object (2)>, <Passengers: Passengers object (3)>, <Passengers: Passengers object (4)>, <Passengers: Passengers object (7)>, <Passengers: Passengers object (8)>, <Passengers: Passengers object (9)>, <Passengers: Passengers object (10)>, <Passengers: Passengers object (11)>, <Passengers: Passengers object (12)>, <Passengers: Passengers object (15)>]>

**Ascending**

>>> pas=Passengers.objects.all().order\_by('age')

>>> pas

<QuerySet [<Passengers: Passengers object (3)>, <Passengers: Passengers object (4)>, <Passengers: Passengers object (15)>, <Passengers: Passengers object (7)>, <Passengers: Passengers object (8)>, <Passengers: Passengers object (9)>, <Passengers: Passengers object (10)>, <Passengers: Passengers object (1)>, <Passengers: Passengers object (2)>, <Passengers: Passengers object (11)>, <Passengers: Passengers object (12)>]>

**Descending**

>>> pas=Passengers.objects.all().order\_by('-age')

>>> pas

<QuerySet [<Passengers: Passengers object (12)>, <Passengers: Passengers object (1)>, <Passengers: Passengers object (2)>, <Passengers: Passengers object (11)>, <Passengers: Passengers object (10)>, <Passengers: Passengers object (9)>, <Passengers: Passengers object (8)>, <Passengers: Passengers object (7)>, <Passengers: Passengers object (15)>, <Passengers: Passengers object (4)>, <Passengers: Passengers object (3)>]>

**Index of a particular record after sort**

>>> pas=Passengers.objects.all().order\_by('-age')[5]

>>> pas

<Passengers: Passengers object (9)>

>>>

**Index of records after sort using Range (‘-‘ Symbol)**

>>> pas=Passengers.objects.all().order\_by('-age')[3:7]

>>> pas

<QuerySet [<Passengers: Passengers object (11)>, <Passengers: Passengers object (10)>, <Passengers: Passengers object (9)>, <Passengers: Passengers object (8)>]>

>>>

**Ignore case sensitivity and sort the records**

>>> from django.db.models.functions import Lower

>>> pas=Passengers.objects.all().order\_by(Lower('firstName'))

>>> pas

<QuerySet [<Passengers: Passengers object (2)>, <Passengers: Passengers object (7)>, <Passengers: Passengers object (8)>, <Passengers: Passengers object (9)>, <Passengers: Passengers object (10)>, <Passengers: Passengers object (11)>, <Passengers: Passengers object (12)>, <Passengers: Passengers object (4)>, <Passengers: Passengers object (3)>, <Passengers: Passengers object (1)>, <Passengers: Passengers object (15)>]>

>>>

**QUIZ**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated