

```
models.py
```

```
from django.db import models  
  
class Employee(models.Model):  
    eno = models.IntegerField()  
    ename = models.CharField(max_length = 65)  
    esal = models.IntegerField()  
    eaddr = models.CharField(max_length=70)
```

Use:

```
python manage.py shell:
```

Q1. How to check Database is perfectly connected to database or not
on shelle:

```
python manage.py shell  
>>> from django.db import connection  
>>> c = connection.cursor()  
if we are not getting any error that means database connected perfectly.
```

Q2: List out All data Present inside Employee model?

Ans: Employee.objects.all()

Q3: find Total No of Data present in Employee models?

Ans: Employee.objects.all().count()

Q4: List out first Data present inside Employee models?

Ans: Employee.objects.all()[0]

Q5: List out all Data from Employee models where ename is contains s and write for S and write for Both?

Ans:

1. Employee.objects.filter(ename__contains='s')
2. Employee.objects.filter(ename__contains='S')
3. Employee.objects.filter(ename__contains= 's')

Q6: List out all Data from Employee where ename startswith sam use case sensetive or esensetive?

Ans:

1. Employee.objects.filter(ename__startswith='sam')
2. Employee.objects.filter(ename__startswith='Sam')
3. Employee.objects.filter(ename__istartswith='sam')

Q7: List out all Documents from Employee model where id = 5?

Ans: Employee.objects.get(id=5)

Note: Write All Possible Script:

=====

Q8. How to Get sql Quires from Orm Query, give a example?

Ans:

```
emp_data = Employee.objects.all()
print(str(emp_data.query))
```

Q9. Find all data from Employee model where esal is greater than 15000?

Ans:

```
emp_data = Employee.objects.filter(esal__gt=15000)
emp_data = Employee.objects.exclude(esal__lte=15000)
emp_data = Employee.objects.filter(~Q(esal__lte=15000))
```

Q10. Find all data from Employee model where esal is greater than or equal 15000?

Ans:

```
emp_data = Employee.objects.filter(esal__gte=15000)
emp_data = Employee.objects.exclude(esal__lt=15000)
emp_data = Employee.objects.filter(~Q(esal__lt=15000))
```

Q11. Find all data from Employee model where esal is less than 20000?

Ans:

```
emp_data = Employee.objects.filter(esal__lt=20000)
emp_data = Employee.objects.filter(~Q(esal__gte=20000))
emp_data = Employee.objects.exclude(esal__gte=20000)
```

Q12. Find all data from Employee model where esal is less than or equal 20000?

Ans:

```
emp_data = Employee.objects.filter(esal__lte=20000)
emp_data = Employee.objects.filter(~Q(esal__gt=20000))
emp_data = Employee.objects.exclude(esal__gt=20000)
```

Q13. Find all data from Employee where ename contains 's', write case sensitive and esensitive and both?

Ans:

```
---
```

```
emp_data = Employee.objects.filter(ename__contains='s')
emp_data = Employee.objects.filter(ename__contains='S')
emp_data = Employee.objects.filter(ename__icontains='s')
```

Q14. Find all data from Employee where ename startswith s, write case sensitive and esensitive and both?

Ans:

```
emp_data = Employee.objects.filter(ename__startswith='s')
emp_data = Employee.objects.filter(ename__startswith='S')
emp_data = Employee.objects.filter(ename__istartswith='s')
```

Q15. Find all data from Employee where ename endswith s, write case sensitive and esensitive and both?

Ans:

```
emp_data = Employee.objects.filter(ename__endswith='s')
emp_data = Employee.objects.filter(ename__endswith='S')
emp_data = Employee.objects.filter(ename__iendswith='s')
```

Q16. Find all data from Employee where ename exact 'sunny', write case sensetive and esensetive and both?

Ans:

emp_data = Employee.objects.filter(ename__exact='sunny')
emp_data = Employee.objects.filter(ename__iexact='sunny')

=====

Q17. find all data form employee model where id in list 1,3,4?

Ans:

emp_data = Employee.objects.filter(id__in=[1,3,4])

=====

Q18. find all data form employee where esal in the range of 12000 to 16000?

Ans:

emp_data = Employee.objects.filter(esal__range(12000,16000))

it will filter data from Employee model where esal is greater and equal to 12000 and less or equal to 16000

Note: __range function include both values start and end values.

This Command is exactly like:

emp_data = Employee.objects.filter(esal__gte=12000) &
Employee.objects.filter(esal__lte=16000)

emp_data = Employee.objects.filter(Q(esal__gte=12000) &
Q(esal__lte=16000))

emp_data = Employee.objects.filter(esal__gte=12000, esal__lte=16000)

=====

Q19. To get all employee whose name startswith 'A' OR esal is less than 15000?

Ans:

emp_data = Employee.objects.filter(ename__startswith='A') |
Employee.objects.filter(esal__lt=15000)

emp_data = Employee.objects.filter(Q(ename__startswith='A')|
Q(esal__lt=15000))

Q20. To get all employee whose name startswith 'j' And esal is less than 15000?

Ans:

emp_data = Employee.objects.filter(ename__startswith='j') &
Employee.objects.filter(esal__lt=15000)

emp_data = Employee.objects.filter(Q(ename__startswith='j') &
Q(esal__lt=15000))

emp_data = Employee.objects.filter(ename__startswith='j',
esal__lt=15000)

Q21. To Select all employees whose name not startswith 'j'?

Ans:

emp_data1 = Employee.objects.exclude(ename__startswith='j')
emp_data2 = Employee.objects.filter(~Q(ename__startwith='j'))

Q22. To select only ename and esal from employee ?

Ans:

```
----  
emp_data1 = Employee.objects.all().values('ename','esal')  
emp_data2 = Employee.objects.all().values_list('ename','esal')  
emp_data3 = Employee.objects.all().only('ename','esal')  
=====
```

Q23. Write all aggregate function for Employee?

Ans:

```
----  
from django.db.models import Min, Max, Avg, Count, Sum  
min_sal = Employee.objects.all().aggregate(Min('esal'))  
print(min_sal['esal__min'])  
-----
```

```
max_sal = Employee.objects.all().aggregate(Max('esal'))  
print(max_sal['esal__max'])  
-----
```

```
avg_sal = Employee.objects.all().aggregate(Avg('esal'))  
avg_sal['esal__avg']  
-----
```

```
count = Employee.objects.all().aggregate(Count('esal'))  
print(count['esal__count'])  
-----
```

```
sum_sal = Employee.objects.all().aggregate(Sum('esal'))  
print(sum_sal['esal__sum'])  
=====
```

Q24. Create a new record using shell?

Data: eno = 1, ename ='Sunny', esal= 10000, eaddr = 'Mumbai'

Ans:

```
emp = Employee(eno = 1, ename ='Sunny', esal= 10000, eaddr =  
'Mumbai')
```

```
emp.save()
```

Q25. Create a bulk recor using shell?

Data:

```
eno = 2, ename ='Bunny', esal= 20000, eaddr = 'Hyderabad'
```

```
eno = 3, ename ='Dunny', esal= 30000, eaddr = 'Delhi'
```

```
eno = 4, ename ='Chinny', esal= 40000, eaddr = 'Noida'
```

```
eno = 5, ename ='Pinny', esal= 50000, eaddr = 'Banglore'
```

Ans:

```
emp_data = Employee.objects.bulk_create([Employee(eno = 2, ename  
='Bunny', esal= 20000, eaddr = 'Hyderabad'),Employee(eno = 3, ename  
='Dunny', esal= 30000, eaddr = 'Delhi'), Employee(eno = 4, ename  
='Chinny', esal= 40000, eaddr = 'Noida'), Employee(eno = 5, ename  
='Pinny', esal= 50000, eaddr = 'Banglore')])
```

Q26. Update ename 'Sunny Leone' where eno = 1?

Ans:

```
>>>emp_data = Employee.objects.get(eno=1)
```

```
>>>emp_data.ename = 'Sunny Leone'
```

```
>>>emp_data.save()
```

Q27. Delete Employee record where eno = 1?

Ans:

```
----  
>>> emp = Employee.objects.get(eno=1)  
>>> emp.delete()
```

Q28. Delete Multiple record where esal is greater than 15000?

```
>>> emp = Employee.objects.filter(esal__gt = 15000)  
>>> emp.delete()
```

Q29. Delete all employee records?

```
>>> emp_data = Employee.objects.all()  
>>> emp_data.delete()
```

Q30. Sort Employee all record in ascending and descending based on esal field?

```
>>> emp_data = Employee.objects.all().order_by('esal') #Ascending order  
>>> emp_data = Employee.objects.all().order_by('-esal') #Descending order
```

Q31. find first Highest salary employee record?

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

```
>>> emp_data.eaddr
```

Q32. find second Highest salary employee record?

```
>>> emp_data = Employee.objects.all().order_by('-esal')[1]
```

```
>>> emp_data.id
```

```
>>> emp_data.eno
```

```
>>> emp_data.ename
```

```
>>> emp_data.esal
```

```
>>> emp_data.eaddr
```

Q33. find Third Highest salary employee record?

```
>>> emp_data = Employee.objects.all().order_by('-esal')[2]
```

```
>>> emp_data.id
```

```
>>> emp_data.eno
```

```
>>> emp_data.ename
```

```
>>> emp_data.esal
```

```
>>> emp_data.eaddr
```

Q34. find first, second and third highest salary in single entity?

```
>>> emp_data = Employee.objects.all().order_by('-esal')[0:3]
```

Q35. find alternative highest salary records thats means skip one record?

```
>>> emp_data = Employee.objects.all().order_by('-esal')[:2]
```