1. In [1]: from veges.models import \*
2. In [2]: vege = Receipe.objects.all()
3. In [3]: vege
   1. Out[3]: <QuerySet [<Receipe: Receipe object (24)>, <Receipe: Receipe object (26)>, <Receipe: Receipe object (29)>, <Receipe: Receipe object (30)>, <Receipe: Receipe object (32)>, <Receipe: Receipe object (33)>]>
4. In [4]: import random
5. In [5]: for i in vege:
6. ...: i.receipe\_view\_count = random.randint(10,
7. ...: 1000)
8. ...: i.save()
9. In [6]: vege[0].receipe\_view\_count
   1. Out[6]: 62
10. In [7]: vege[2].receipe\_view\_count
    1. Out[7]: 55
11. In [8]: for i in vege:
12. ...: print(i.receipe\_view\_count)
    1. 62
    2. 577
    3. 55
    4. 633
    5. 999
    6. 391
13. In [9]: vege = Receipe.objects.all().order\_by("receipe\_view\_count)
14. vege = Receipe.objects.all().order\_by("receipe\_view\_count)
15. In [10]: vege = Receipe.objects.all().order\_by("receipe\_view\_count")
16. In [11]: vege
    1. Out[11]: <QuerySet [<Receipe: Receipe object (29)>, <Receipe: Receipe object (24)>, <Receipe: Receipe object (33)>, <Receipe: Receipe object (26)>, <Receipe: Receipe object (30)>, <Receipe: Receipe object (32)>]>
17. In [12]: for i in vege:
18. ...: print(i.receipe\_view\_count)
    1. 55
    2. 62
    3. 391
    4. 577
    5. 633
    6. 999

Prints receipe view count in ascending order

1. In [13]: vege = Receipe.objects.all().order\_by("-receipe\_view\_count")
2. ...:
3. In [14]: for i in vege:
4. ...: print(i.receipe\_view\_count)
   1. 999
   2. 633
   3. 577
   4. 391
   5. 62
   6. 55

## Prints receipe view count in descending order

1. In [15]:
2. In [15]: vege = Receipe.objects.all().order\_by("-rece
3. ...: ipe\_view\_count")[0:2]
4. In [16]: for i in vege:
5. ...: print(i.receipe\_view\_count)
   1. 999
   2. 633
   3. Here we get only 2 records
6. : Receipe.objects.filter(receipe\_view\_count=55 )
   1. Out[17]: <QuerySet [<Receipe: Receipe object (29)>]>

Gives receipes with view count == 55

1. Receipe.objects.filter(receipe\_view\_count\_\_gte=55)
   1. Out[20]: <QuerySet [<Receipe: Receipe object (24)>, <Receipe: Receipe object (26)>, <Receipe: Receipe object (29)>, <Receipe: Receipe object (30)>, <Receipe: Receipe object (32)>, <Receipe: Receipe object (33)>]>

Here we can see receipes with view count greater than 55

1. In [21]: Receipe.objects.filter(receipe\_view\_count\_\_lte=55)
   1. Out[21]: <QuerySet [<Receipe: Receipe object (29)>]>

\_\_lte implies lessthan

1. In [6]: student = Student.objects.filter(student\_name="abhi")
2. In [7]: student
   1. Out[7]: <QuerySet []>

Filters students with name abhi

1. In [13]: student = Student.objects.filter(student\_name\_\_startswith="ab")
2. In [14]: student
   1. Out[14]: <QuerySet []>

Filters students names starts with “ab”

1. In [11]: student = Student.objects.filter(student\_name\_\_startswith="a")
2. In [12]: student
   1. Out[12]: <QuerySet [<Student: Alison Ingram>, <Student: Amber Patton>, <Student: Amber Vargas>, <Student: Amy Barnes>, <Student: Andrea Rogers>, <Student: Andrew Malone>, <Student: Angel Washington>, <Student: Anita Harper>, <Student: Autumn Wallace>]>

Filters students names starts with “a”

1. In [15]: student = Student.objects.filter(student\_name\_\_endswith="th")
2. In [16]: student
   1. Out[16]: <QuerySet [<Student: Matthew Smith>, <Student: Wesley Smith>]>

Gives students names wnding with th

1. In [25]: student = Student.objects.filter(student\_name\_\_icontains="an")
2. In [27]: student
   1. Out[27]: <QuerySet [<Student: Andrea Rogers>, <Student: Andrew Malone>, <Student: Angel Washington>, <Student: Anita Harper>, <Student: Brandi Tapia>, <Student: Brenda Sandoval>, <Student: Brian Green>, <Student: Brittany Franklin>, <Student: Christopher Landry>, <Student: Christopher Randolph>, <Student: Corey Christian>, <Student: Daniel Davis>, <Student: David Coleman>, <Student: Derek Duran>, <Student: Francisco Stevens>, <Student: Janet Anderson>, <Student: Jeffery Bean>, <Student: Jessica Santana>, <Student: Joseph Sanchez>, <Student: Juan Lynch>, '...(remaining elements truncated)...']>
3. In [28]: student[0]
   1. Out[28]: <Student: Andrea Rogers>
4. In [29]: student[0].department
   1. Out[29]: <Department: Computers>
5. S
6. In [30]: student = Student.objects.filter(department\_\_department="Computers")
7. In [31]: student
   1. Out[31]: <QuerySet [<Student: Alison Ingram>, <Student: Amber Patton>, <Student: Andrea Rogers>, <Student: Anita Harper>, <Student: Beth Hines>, <Student: Christopher Landry>, <Student: Daisy Taylor>, <Student: Jeffery Bean>, <Student: Karen Rush>, <Student: Kenneth Lindsey>, <Student: Leslie Wilson>, <Student: Marissa Ryan>, <Student: Matthew Harris>, <Student: Monique Fitzgerald>, <Student: Robert Wheeler>, <Student: Tamara Harper>, <Student: Teresa Banks>]>

Here department\_\_ says that it is foreign key of student and second deparment says that it is column of department table. If there are other rows example date of establishment to access them we have to write it as department\_\_dateofestablishment

And now if department is connected with other table with a foreign key lets say head, then to access the first\_name of head we have to write department\_\_head\_\_date\_ob\_first\_name.

1. D = [‘Computers’, ‘Mining’]
2. In [35]: student = Student.objects.filter(department\_\_department\_\_in=d)
3. In [36]: student
   1. Out[36]: <QuerySet [<Student: Alison Ingram>, <Student: Amber Patton>, <Student: Andrea Rogers>, <Student: Anita Harper>, <Student: Beth Hines>, <Student: Bobby Gonzales>, <Student: Brenda Sandoval>, <Student: Brittany Franklin>, <Student: Christopher Landry>, <Student: Daisy Taylor>, <Student: Dale Carr>, <Student: David Cole>, <Student: Derek Duran>, <Student: Elizabeth Novak>, <Student: James Bullock>, <Student: Jeffery Bean>, <Student: Jennifer Lamb>, <Student: Julie Hart>, <Student: Karen Rush>, <Student: Kathryn Estrada>, '...(remaining elements truncated)...']>

Gives the list of students who are in computers department or mining department

1. In [39]: student = Student.objects.exclude(department\_\_department="Computers ")
2. In [40]: student
   1. Out[40]: <QuerySet [<Student: Amber Vargas>, <Student: Amy Barnes>, <Student: Andrew Malone>, <Student: Angel Washington>, <Student: Autumn Wallace>, <Student: Barbara Russo>, <Student: Becky Cruz>, <Student: Benjamin Long>, <Student: Bobby Gonzales>, <Student: Brandi Tapia>, <Student: Brenda Miller>, <Student: Brenda Sandoval>, <Student: Brett Hubbard>, <Student: Brett Waters>, <Student: Brian Green>, <Student: Brittany Franklin>, <Student: Casey Allen>, <Student: Christine Johnson>, <Student: Christopher Randolph>, <Student: Christopher Reynolds>, '...(remaining elements truncated)...']>
3. In [4]: Student.objects.aggregate(Avg("student\_age"))
   1. Out[4]: {'student\_age\_\_avg': 22.34090909090909}
4. In [5]: Student.objects.aggregate(Max("student\_age"))
   1. Out[5]: {'student\_age\_\_max': 30}
5. In [6]: Student.objects.aggregate(Min("student\_age"))
   1. Out[6]: {'student\_age\_\_min': 15}
6. In [7]: Student.objects.aggregate(Sum("student\_age"))
   1. Out[7]: {'student\_age\_\_sum': 2949}
7. In [10]: student = Student.objects.values("student\_age").annotate(Count("student\_age"))
8. In [11]: student
   1. Out[11]: <QuerySet [{'student\_age': 15, 'student\_age\_\_count': 7}, {'student\_age': 16, 'student\_age\_\_count': 11}, {'student\_age': 17, 'student\_age\_\_count': 12}, {'student\_age': 18, 'student\_age\_\_count': 4}, {'student\_age': 19, 'student\_age\_\_count': 3}, {'student\_age': 20, 'student\_age\_\_count': 14}, {'student\_age': 21, 'student\_age\_\_count': 7}, {'student\_age': 22, 'student\_age\_\_count': 6}, {'student\_age': 23, 'student\_age\_\_count': 7}, {'student\_age': 24, 'student\_age\_\_count': 17}, {'student\_age': 25, 'student\_age\_\_count': 10}, {'student\_age': 26, 'student\_age\_\_count': 9}, {'student\_age': 27, 'student\_age\_\_count': 3}, {'student\_age': 28, 'student\_age\_\_count': 6}, {'student\_age': 29, 'student\_age\_\_count': 6}, {'student\_age': 30, 'student\_age\_\_count': 10}]>
9. In [12]: student = Student.objects.values("student\_age").annotate(Count("student\_name"))
10. In [14]: student
    1. Out[14]: <QuerySet [{'student\_age': 15, 'student\_name\_\_count': 7}, {'student\_age': 16, 'student\_name\_\_count': 11}, {'student\_age': 17, 'student\_name\_\_count': 12}, {'student\_age': 18, 'student\_name\_\_count': 4}, {'student\_age': 19, 'student\_name\_\_count': 3}, {'student\_age': 20, 'student\_name\_\_count': 14}, {'student\_age': 21, 'student\_name\_\_count': 7}, {'student\_age': 22, 'student\_name\_\_count': 6}, {'student\_age': 23, 'student\_name\_\_count': 7}, {'student\_age': 24, 'student\_name\_\_count': 17}, {'student\_age': 25, 'student\_name\_\_count': 10}, {'student\_age': 26, 'student\_name\_\_count': 9}, {'student\_age': 27, 'student\_name\_\_count': 3}, {'student\_age': 28, 'student\_name\_\_count': 6}, {'student\_age': 29, 'student\_name\_\_count': 6}, {'student\_age': 30, 'student\_name\_\_count': 10}]>
11. In [15]: student = Student.objects.values("student\_name").annotate(Count("student\_name"))
12. In [16]: student
    1. Out[16]: <QuerySet [{'student\_name': 'Alison Ingram', 'student\_name\_\_count': 1}, {'student\_name': 'Amber Patton', 'student\_name\_\_count': 1}, {'student\_name': 'Amber Vargas', 'student\_name\_\_count': 1}, {'student\_name': 'Amy Barnes', 'student\_name\_\_count': 1}, {'student\_name': 'Andrea Rogers', 'student\_name\_\_count': 1}, {'student\_name': 'Andrew Malone', 'student\_name\_\_count': 1}, {'student\_name': 'Angel Washington', 'student\_name\_\_count': 1}, {'student\_name': 'Anita Harper', 'student\_name\_\_count': 1}, {'student\_name': 'Autumn Wallace', 'student\_name\_\_count': 1}, {'student\_name': 'Barbara Russo', 'student\_name\_\_count': 1}, {'student\_name': 'Becky Cruz', 'student\_name\_\_count': 1}, {'student\_name': 'Benjamin Long', 'student\_name\_\_count': 1}, {'student\_name': 'Beth Hines', 'student\_name\_\_count': 1}, {'student\_name': 'Bobby Gonzales', 'student\_name\_\_count': 1}, {'student\_name': 'Brandi Tapia', 'student\_name\_\_count': 1}, {'student\_name': 'Brenda Miller', 'student\_name\_\_count': 1}, {'student\_name': 'Brenda Sandoval', 'student\_name\_\_count': 1}, {'student\_name': 'Brett Hubbard', 'student\_name\_\_count': 1}, {'student\_name': 'Brett Waters', 'student\_name\_\_count': 1}, {'student\_name': 'Brian Green', 'student\_name\_\_count': 1}, '...(remaining elements truncated)...']>

Gives the count of each value mentioned in first bracket, naming it as value mentioned in second bracet.