**Video file : 68**

**Code file : ch58**

**Working with Database Field Lookups**

At end our own file

## **Working with Database – Field Lookups**

Its equal to where clause that we use in database accourding to youtube instructor

In Django, **Field Lookups** are used inside QuerySet methods (filter(), exclude(), get()) to compare model fields with specific conditions.

They allow you to write queries like **greater than, less than, contains, startswith, exact match, case-insensitive match, etc.**

### ✅ Key Points:

* Syntax:
* Model.objects.filter(fieldname\_\_lookup=value)
  + fieldname → Model field name
  + lookup → Condition type (e.g., exact, gt, lt, icontains)
* Double underscore \_\_ is used before the lookup type.
* Returns a **QuerySet** (unless you use .get()).

### 🔹 Common Field Lookups

from school.models import Student

# Exact match

Student.objects.filter(city\_\_exact="Lahore")

# Case-insensitive match

Student.objects.filter(city\_\_iexact="lahore")

# Contains substring

Student.objects.filter(name\_\_contains="Ali")

# Case-insensitive contains

Student.objects.filter(name\_\_icontains="ali")

# Greater than marks > 60

Student.objects.filter(marks\_\_gt=60)

# Greater than or equal marks >= 60

Student.objects.filter(marks\_\_gte=60)

# Less than marks < 40

Student.objects.filter(marks\_\_lt=40)

# Range query (marks between 50 and 80)

Student.objects.filter(marks\_\_range=(50, 80))

# Starts with / ends with

Student.objects.filter(name\_\_startswith="A")

Student.objects.filter(name\_\_endswith="n")

# Null check

Student.objects.filter(city\_\_isnull=True)

# Date lookups (students who passed in 2023)

Student.objects.filter(pass\_date\_\_year=2023)

👉 **In short:**  
**Field Lookups** let you add conditions to queries (like SQL WHERE clause) using \_\_lookup syntax.  
They make filtering **powerful, readable, and data base-independent**.

Complete views.py:

from django.shortcuts import render

from school.models import Student

from datetime import date, time

# Create your views here.

def home(request):

    # -----------------------------

    # String Lookups

    # -----------------------------

    # Exact match (case-sensitive)

    student\_data = Student.objects.filter(name\_\_exact='sonam')

    # Exact match (case-insensitive)

    student\_data = Student.objects.filter(name\_\_iexact='sonam')

    # Name contains 'r' (case-sensitive)

    student\_data = Student.objects.filter(name\_\_contains='r')

    # Name contains 's' (case-insensitive)

    student\_data = Student.objects.filter(name\_\_icontains='s')

    # -----------------------------

    # Lookup with IDs

    # -----------------------------

    # Get students with IDs in given list

    student\_data = Student.objects.filter(id\_\_in=[1, 3, 7])

    # -----------------------------

    # Numeric Lookups

    # -----------------------------

    # Marks greater than or equal to 200

    student\_data = Student.objects.filter(marks\_\_gte=200)

    # Marks less than 200

    student\_data = Student.objects.filter(marks\_\_lt=200)

    # Marks less than or equal to 200

    student\_data = Student.objects.filter(marks\_\_lte=200)

    # -----------------------------

    # String Lookups with Start/End

    # -----------------------------

    # Name starts with 'S' (case-sensitive)

    student\_data = Student.objects.filter(name\_\_startswith='S')

    # Name starts with 'S' (case-insensitive)

    student\_data = Student.objects.filter(name\_\_istartswith='S')

    # Name ends with 'S' (case-sensitive)

    student\_data = Student.objects.filter(name\_\_endswith='S')

    # Name ends with 't' (case-insensitive)

    student\_data = Student.objects.filter(name\_\_iendswith='t')

    # -----------------------------

    # Date Lookups

    # -----------------------------

    # Students who passed between given dates

    student\_data = Student.objects.filter(pass\_date\_\_range=('2024-1-1', '2025-9-10'))

    # Admission date is exact date

    student\_data = Student.objects.filter(admission\_date\_\_date=date(2025, 9, 2))

    # Admission date greater than a specific date

    student\_data = Student.objects.filter(admission\_date\_\_date\_\_gt=date(2025, 9, 2))

    # Admission date less than a specific date

    student\_data = Student.objects.filter(admission\_date\_\_date\_\_lt=date(2025, 9, 2))

    # -----------------------------

    # Year, Month, Day Lookups

    # -----------------------------

    # Students who passed in year 2024

    student\_data = Student.objects.filter(pass\_date\_\_year=2024)

    # Passed in January (month = 1)

    student\_data = Student.objects.filter(pass\_date\_\_month=1)

    # Passed in month > 5

    student\_data = Student.objects.filter(pass\_date\_\_month\_\_gt=5)

    # Passed in day > 5

    student\_data = Student.objects.filter(pass\_date\_\_day\_\_gt=5)

    # -----------------------------

    # Week & Weekday Lookups

    # -----------------------------

    # Passed in week number 15

    student\_data = Student.objects.filter(pass\_date\_\_week=15)

    # Passed in week number > 15

    student\_data = Student.objects.filter(pass\_date\_\_week\_\_gt=15)

    # Passed in week number < 5

    student\_data = Student.objects.filter(pass\_date\_\_week\_\_lt=5)

    # Passed on weekday = 5 (1=Sunday, 7=Saturday in Django)

    student\_data = Student.objects.filter(pass\_date\_\_week\_day=5)

    # Passed on weekday greater than 5

    student\_data = Student.objects.filter(pass\_date\_\_week\_day\_\_gt=5)

    # -----------------------------

    # Quarter Lookups

    # -----------------------------

    # Passed in quarter 1

    student\_data = Student.objects.filter(pass\_date\_\_quarter=1)

    # Passed in quarter greater than 1

    student\_data = Student.objects.filter(pass\_date\_\_quarter\_\_gt=1)

    # -----------------------------

    # Time Lookups

    # -----------------------------

    # Admission time is exactly 12:46

    student\_data = Student.objects.filter(admission\_date\_\_time=time(12, 46))

    # Admission time greater than 06:25

    student\_data = Student.objects.filter(admission\_date\_\_time\_\_gt=time(6, 25))

    # Admission time is exactly 06:25

    student\_data = Student.objects.filter(admission\_date\_\_time=time(6, 25))

    # Admission hour greater than 5

    student\_data = Student.objects.filter(admission\_date\_\_hour\_\_gt=5)

    # Admission at exact minute = 40

    student\_data = Student.objects.filter(admission\_date\_\_minute=40)

    # Admission at minute > 40

    student\_data = Student.objects.filter(admission\_date\_\_minute\_\_gt=40)

    # Admission at exact second = 40

    student\_data = Student.objects.filter(admission\_date\_\_second=40)

    # Admission at second > 40

    student\_data = Student.objects.filter(admission\_date\_\_second\_\_gt=40)

    # -----------------------------

    # Null Check

    # -----------------------------

    # Students whose roll is NOT null

    student\_data = Student.objects.filter(roll\_\_isnull=False)

    # -----------------------------

    # Print + Render

    # -----------------------------

    print("Students Data:", student\_data)

    return render(request, 'school/home.html', {'students': student\_data})