

## 1. contains

คำอธิบาย: ตรวจสอบว่าฟิลด์มีคำหรือสตริงที่ระบุอยู่หรือไม่

ตัวอย่าง:

python

Copy code

```
# SQL: SELECT * FROM employee WHERE name LIKE '%John%'
```

```
Employee.objects.filter(name__contains='John')
```

## 2. icontains

คำอธิบาย: เหมือนกับ contains แต่ไม่สนใจตัวพิมพ์ใหญ่/เล็ก

ตัวอย่าง:

python

Copy code

```
# SQL: SELECT * FROM employee WHERE LOWER(name) LIKE LOWER('%john%')
```

```
Employee.objects.filter(name__icontains='john')
```

## 3. date

คำอธิบาย: กรองข้อมูลตามวันที่ (ใช้กับ DateField)

ตัวอย่าง:

python

Copy code

```
# SQL: SELECT * FROM employee WHERE hire_date = '2024-08-26'
```

```
Employee.objects.filter(hire_date__date='2024-08-26')
```

## 4. day

คำอธิบาย: กรองข้อมูลตามวันในเดือน (1-31)

ตัวอย่าง:

python

Copy code

```
# SQL: SELECT * FROM employee WHERE EXTRACT(DAY FROM hire_date) = 26
```

```
Employee.objects.filter(hire_date__day=26)
```

## 5. endswith

คำอธิบาย: ตรวจสอบว่าฟิลด์จบด้วยสตริงที่ระบุ

ตัวอย่าง:

python

Copy code

```
# SQL: SELECT * FROM employee WHERE name LIKE '%Smith'
```

```
Employee.objects.filter(name__endswith='Smith')
```

## 6. iendswith

คำอธิบาย: เหมือนกับ endswith แต่ไม่สนใจตัวพิมพ์ใหญ่/เล็ก

ตัวอย่าง:

python

Copy code

```
# SQL: SELECT * FROM employee WHERE LOWER(name) LIKE LOWER('%smith')
```

```
Employee.objects.filter(name__iendswith='smith')
```

## 7. exact

คำอธิบาย: ตรวจสอบการจับคู่ที่ตรงตามค่าที่ระบุ

ตัวอย่าง:

python

Copy code

```
# SQL: SELECT * FROM employee WHERE name = 'John Doe'
```

```
Employee.objects.filter(name__exact='John Doe')
```

## 8. iexact

คำอธิบาย: เหมือนกับ exact แต่ไม่สนใจตัวพิมพ์ใหญ่/เล็ก

ตัวอย่าง:

python

Copy code

```
# SQL: SELECT * FROM employee WHERE LOWER(name) = LOWER('john doe')
```

```
Employee.objects.filter(name__iexact='john doe')
```

## 9. in

คำอธิบาย: ตรวจสอบว่าค่าของฟิลด์อยู่ในรายการที่ระบุ

ตัวอย่าง:

python

Copy code

```
# SQL: SELECT * FROM employee WHERE id IN (1, 2, 3)
```

```
Employee.objects.filter(id__in=[1, 2, 3])
```

## 10. isnull

คำอธิบาย: ตรวจสอบค่าที่เป็น NULL

ตัวอย่าง:

python

Copy code

```
# SQL: SELECT * FROM employee WHERE position IS NULL
```

```
Employee.objects.filter(position__isnull=True)
```

## 11. gt

คำอธิบาย: ตรวจสอบค่าที่มากกว่าค่าที่ระบุ

ตัวอย่าง:

python

Copy code

**# SQL: SELECT \* FROM employee WHERE salary > 50000**

**Employee.objects.filter(salary\_\_gt=50000)**

## **12. gte**

**คำอธิบาย: ตรวจสอบค่าที่มากกว่าหรือเท่ากับค่าที่ระบุ**

**ตัวอย่าง:**

**python**

**Copy code**

**# SQL: SELECT \* FROM employee WHERE salary >= 50000**

**Employee.objects.filter(salary\_\_gte=50000)**

## **13. hour**

**คำอธิบาย: กรองข้อมูลตามชั่วโมง (ใช้กับ DateTimeField)**

**ตัวอย่าง:**

**python**

**Copy code**

**# SQL: SELECT \* FROM employee WHERE EXTRACT(HOUR FROM hire\_date) = 9**

**Employee.objects.filter(hire\_date\_\_hour=9)**

## **14. lt**

**คำอธิบาย: ตรวจสอบค่าที่น้อยกว่าค่าที่ระบุ**

**ตัวอย่าง:**

**python**

**Copy code**

**# SQL: SELECT \* FROM employee WHERE salary < 50000**

**Employee.objects.filter(salary\_\_lt=50000)**

## **15. lte**

**คำอธิบาย: ตรวจสอบค่าที่น้อยกว่าหรือเท่ากับค่าที่ระบุ**

ตัวอย่าง:

python

Copy code

```
# SQL: SELECT * FROM employee WHERE salary <= 50000
```

```
Employee.objects.filter(salary__lte=50000)
```

## 16. minute

คำอธิบาย: กรองข้อมูลตามนาทีก (ใช้กับ DateTimeField)

ตัวอย่าง:

python

Copy code

```
# SQL: SELECT * FROM employee WHERE EXTRACT(MINUTE FROM hire_date) = 30
```

```
Employee.objects.filter(hire_date__minute=30)
```

## 17. month

คำอธิบาย: กรองข้อมูลตามเดือน (1-12)

ตัวอย่าง:

python

Copy code

```
# SQL: SELECT * FROM employee WHERE EXTRACT(MONTH FROM hire_date) = 8
```

```
Employee.objects.filter(hire_date__month=8)
```

## 18. quarter

คำอธิบาย: กรองข้อมูลตามไตรมาสของปี (1-4)

ตัวอย่าง:

python

Copy code

```
# SQL: SELECT * FROM employee WHERE EXTRACT(QUARTER FROM hire_date) = 3
```

`Employee.objects.filter(hire_date__quarter=3)`

## 19. range

คำอธิบาย: กรองข้อมูลตามช่วงของค่า

ตัวอย่าง:

python

Copy code

# SQL: SELECT \* FROM employee WHERE salary BETWEEN 40000 AND 60000

`Employee.objects.filter(salary__range=(40000, 60000))`

## 20. regex

คำอธิบาย: ใช้การจับคู่ตาม regular expression

ตัวอย่าง:

python

Copy code

# SQL: SELECT \* FROM employee WHERE name REGEXP 'John|Doe'

`Employee.objects.filter(name__regex=r'John|Doe')`

## 21. iregex

คำอธิบาย: เหมือนกับ regex แต่ไม่สนใจตัวพิมพ์ใหญ่/เล็ก

ตัวอย่าง:

python

Copy code

# SQL: SELECT \* FROM employee WHERE LOWER(name) REGEXP LOWER('john|doe')

`Employee.objects.filter(name__iregex=r'john|doe')`

## 22. second

คำอธิบาย: กรองข้อมูลตามวินาที (ใช้กับ DateTimeField)

ตัวอย่าง:

python

Copy code

```
# SQL: SELECT * FROM employee WHERE EXTRACT(SECOND FROM hire_date) = 45
```

```
Employee.objects.filter(hire_date__second=45)
```

## 23. startswith

คำอธิบาย: ตรวจสอบว่าฟิลด์เริ่มต้นด้วยสตริงที่ระบุ

ตัวอย่าง:

python

Copy code

```
# SQL: SELECT * FROM employee WHERE name LIKE 'John%'
```

```
Employee.objects.filter(name__startswith='John')
```

## 24. istartswith

คำอธิบาย: เหมือนกับ startswith แต่ไม่สนใจตัวพิมพ์ใหญ่/เล็ก

ตัวอย่าง:

python

Copy code

```
# SQL: SELECT * FROM employee WHERE LOWER(name) LIKE LOWER('john%')
```

```
Employee.objects.filter(name__istartswith='john')
```

## 25. time

คำอธิบาย: กรองข้อมูลตามเวลา (ใช้กับ DateTimeField)

ตัวอย่าง:

python

Copy code

```
# SQL: SELECT * FROM employee WHERE EXTRACT(TIME FROM hire_date) = '09:00:00'
```

```
Employee.objects.filter(hire_date__time='09:00:00')
```

## 26. week

คำอธิบาย: กรองข้อมูลตามหมายเลขสัปดาห์ (1-53)

ตัวอย่าง:

python

Copy code

```
# SQL: SELECT * FROM employee WHERE EXTRACT(WEEK FROM hire_date) = 34
```

```
Employee.objects.filter(hire_date__week=34)
```

## 27. week\_day

คำอธิบาย: กรองข้อมูลตามวันในสัปดาห์ (1-7) โดยที่ 1 คือวันอาทิตย์

ตัวอย่าง:

python

Copy code

```
# SQL: SELECT * FROM employee WHERE EXTRACT(DOW FROM hire_date) = 1
```

```
Employee.objects.filter(hire_date__week_day=1)
```

## 28. iso\_week\_day

คำอธิบาย: กรองข้อมูลตามวันในสัปดาห์แบบ ISO 8601 (1-7) โดยที่ 1 คือวันจันทร์

ตัวอย่าง:

python

Copy code

```
# SQL: SELECT * FROM employee WHERE EXTRACT(ISODOW FROM hire_date) = 1
```

```
Employee.objects.filter(hire_date__iso_week_day=1)
```

## 29. year

คำอธิบาย: กรองข้อมูลตามปี

ตัวอย่าง:

python



Copy code

```
# SQL: SELECT * FROM employee WHERE EXTRACT(YEAR FROM hire_date) = 2024
```

```
Employee.objects.filter(hire_date__year=2024)
```

### 30. iso\_year

คำอธิบาย: กรองข้อมูลตามปีแบบ ISO 8601

ตัวอย่าง:

python

Copy code

```
# SQL: SELECT * FROM employee WHERE EXTRACT(ISOYEAR FROM hire_date) = 2024
```

```
Employee.objects.filter(hire_date__iso_year=2024)
```

หวังว่าข้อมูลนี้จะช่วยให้คุณเข้าใจการใช้งาน field lookups ใน Django ORM ได้ดีขึ้นครับ! หากมีคำถามเพิ่มเติมหรือต้องการความช่วยเหลือเพิ่มเติม โปรดแจ้งได้เลยครับ