# Filtering Results with the WHERE Clause



Jon Flanders

@jonflanders www.jonflanders.com

WHERE

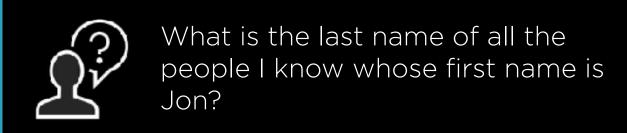
Constrains the result set

Comes after the FROM clause

Contains boolean expressions

Only matching rows are in the result set

SELECT p.last\_name FROM person p; WHERE p.first\_name = 'Jon'



- **◄ SELECT CLAUSE**
- **▼ FROM CLAUSE**
- **◆ WHERE CLAUSE**

### Boolean Operators

#### Operator

=	Equals	True if values on both sides are equal
<b>&lt;&gt;</b>	Not Equal TO	True if value on both sides are not equal
>	Greater Than	True if left side is larger than right side
<	Less Than	True if left side is smaller than right side
>=	Greater or Equal	True if left side is larger or equal to right
<=	Less Than or Equal	True if left side is smaller or equal to right

A Single Expression is Quite Limiting We'd like to ask more complex questions

Additional keywords are needed

These can chain multiple expressions

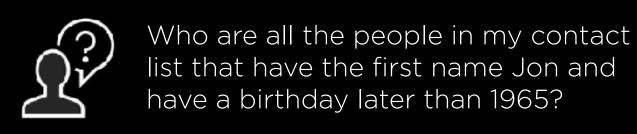
AND

Combines two expressions

If both are TRUE, row is included

If either is FALSE, row is excluded

```
SELECT p.first_name,
p.last_name
FROM person p
WHERE p.first_name = 'Jon'
AND p.birthdate >
'12/31/1965';
```



- **SELECT CLAUSE**
- **▼ FROM CLAUSE**
- **WHERE CLAUSE**
- AND

OR

Also combines two expressions

If either are TRUE, row is included

If both are FALSE, row is excluded

## Who are all the people in my contact list that have the first name Jon or a last name of Flanders?

```
SELECT p.first_name,
p.last_name
FROM person p
```

```
WHERE p.first_name = 'Jon'
OR p.last_name = 'Flanders';
```

- **SELECT CLAUSE**
- **◄ FROM CLAUSE**
- **◆ WHERE CLAUSE**
- ◆ OR

### Other Operators

**BETWEEN** LIKE IN IS NOT IS

BETWEEN

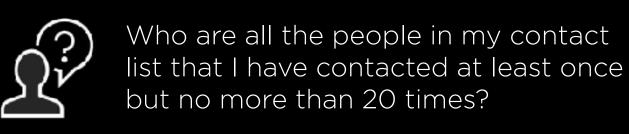
Acts on column and two values

TRUE if column value is between two values

Inclusive - includes two values (like >= & <=)

SELECT p.first\_name,
p.last\_name
FROM person p

WHERE p.contacted BETWEEN 1 AND 20;



- **SELECT CLAUSE**
- **▼ FROM CLAUSE**
- **◆ WHERE CLAUSE**
- **■** BETWEEN

IIKF

A more fuzzy version of =

String with special characters inside

If the match is true, the row is returned

```
SELECT p.first_name,
p.last_name
FROM person p
```

WHERE p.first\_name
LIKE 'J%';



Who are all the people in my contact list that have a first name that begins with the letter J?

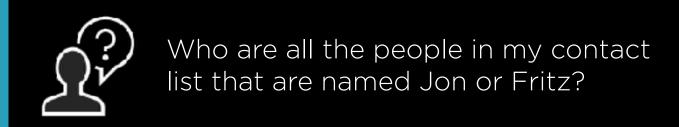
- **SELECT CLAUSE**
- **▼ FROM CLAUSE**
- **WHERE CLAUSE**
- **LIKE**

Like a multi-value = operator

List of potential values

True if any of the values in the list "hit"

```
SELECT p.first_name,
p.last_name
FROM person p
WHERE p.first_name
IN ('Jon', 'Fritz');
```



- **SELECT CLAUSE**
- **▼ FROM CLAUSE**
- **WHERE CLAUSE**
- IN

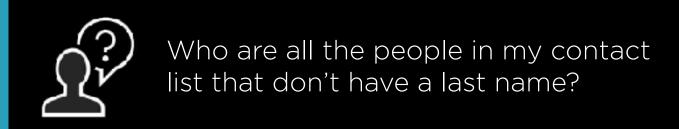
15

**Special operator** 

Like a equals operator

But just for values that might be NULL

```
SELECT p.first_name,
p.last_name
FROM person p
WHERE p.last_name
IS NULL;
```



- **◀ SELECT CLAUSE**
- **▼ FROM CLAUSE**

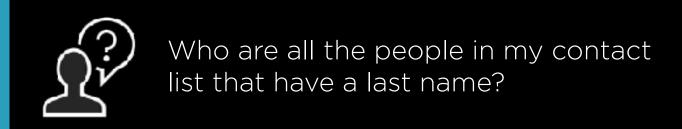
- **WHERE CLAUSE**
- **◆** IS

IS NOT

Also just for NULL

Like a "NOT EQUALS" operator

SELECT p.first\_name,
p.last\_name
FROM person p
WHERE p.last\_name
IS NOT NULL;



- **SELECT CLAUSE**
- **▼ FROM CLAUSE**
- **WHERE CLAUSE**
- **◀ IS NOT**

### Summary

The WHERE clause enables us to restrict the result set of our queries

The more complex the question we want to ask, the more complex the WHERE clause becomes