</talentlabs>

CHAPTER 5

SELECT Queries with Conditions

</talentlabs>

AGENDA

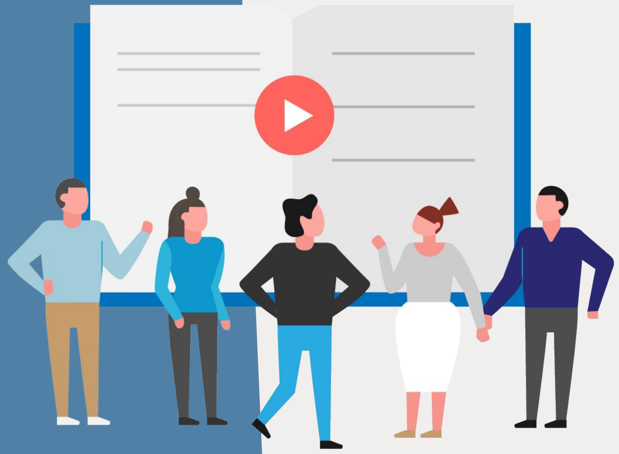
- WHERE clause
- Compound Conditions with AND/OR
- BETWEEN and IN
- LIKE

Where Clause

WHERE Clause

- WHERE clause is used to set a series of logic to filter out unwanted data
- We use different logic operators and SQL keywords to build conditions.

```
SELECT columns    FROM table    WHERE condition(s)
GROUP BY column(s)    ORDER BY column(s)    LIMIT n
```



</talentlabs>

WHERE Clause

- Using SELECT FROM, we query the whole movies table

```
SELECT * FROM movies
```

- If we want to specify to query movies of which year(s), we can add a comparison logic after a WHERE keyword

- Example: select all the movies with year equals 1990

```
SELECT * FROM movies WHERE year=1990
```

WHERE Clause

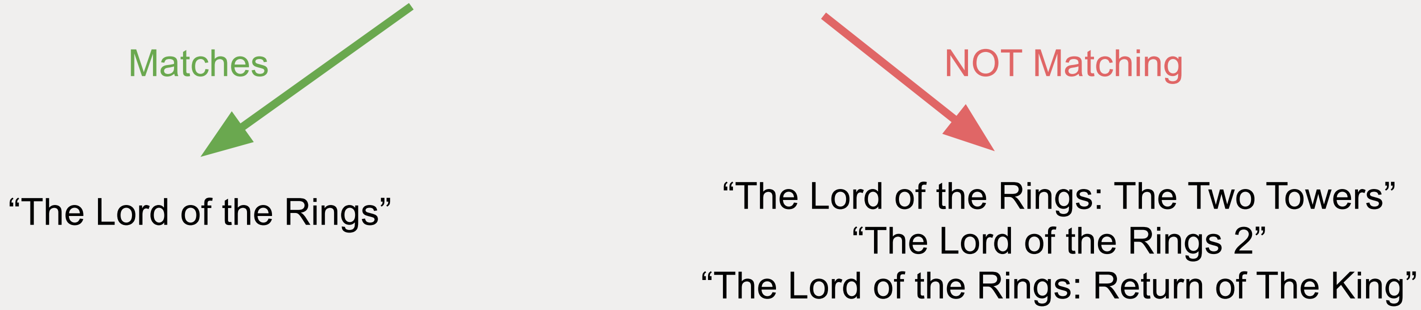
- WHERE clause supports common comparison operators

```
=   Equal
>   Bigger Than
<   Smaller Than
>=  Bigger Than or Equal
<=  Smaller Than or Equal
<>  Not equal
```

WHERE Clause for Strings

- Equal sign can also searches for ***exact match*** of the condition input to the column values
- To specify a text, use double quotes - ""

```
SELECT * FROM movies WHERE title = "The Lord of the Rings"
```



Matches

"The Lord of the Rings"

NOT Matching

"The Lord of the Rings: The Two Towers"
"The Lord of the Rings 2"
"The Lord of the Rings: Return of The King"

Compound Conditions

Compound Conditions - AND

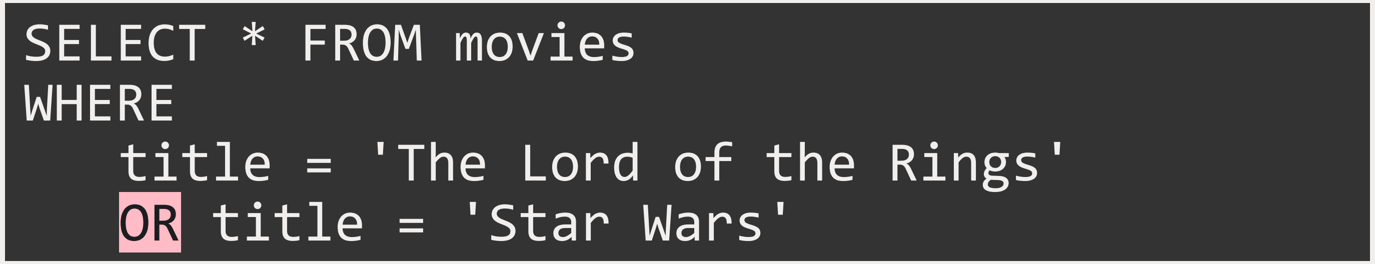
Example 1: Get movies which release year is later than 1990 and earlier than 2000

```
SELECT * FROM movies
WHERE year > 1990 AND year < 2000
```

</talentlabs>

Compound Conditions - OR

Example 2: Get movies which title is The Lord of the Rings or Star Wars



```
SELECT * FROM movies
WHERE
   title = 'The Lord of the Rings'
   OR title = 'Star Wars'
```

Compound Conditions - NOT

- You can also use NOT keyword to represent “exception”

Example - movies that are not released after 2000

```
SELECT * FROM movies
WHERE
   NOT year > 2000
```

Compound Conditions - Parentheses

- As we are chaining more and more conditions and we want to make sure our logic is correct and clear, we can use Parentheses - ()

Example - movies that are not released between 1990 and 2000

```
Correct
SELECT * FROM movies
WHERE
   NOT (year > 1990 and year <2000)
```

Compound Conditions - Parentheses

- As we are chaining more and more conditions and we want to make sure our logic is correct and clear, we can use Parentheses - ()

Example - movies that are not released between 1990 and 2000

```
Correct
SELECT * FROM movies
WHERE
   NOT (year > 1990 AND year < 2000)
```

- movies that are before 1990
- moves that are after 2000

```
Wrong
SELECT * FROM movies
WHERE
   NOT year > 1990 AND year < 2000
```

ONLY movies that are before 1990

Simplifying Queries with BETWEEN and IN

BETWEEN

- We can use BETWEEN keyword to specific a range of value
- Consider the below SQL query

```
SELECT * FROM movies WHERE year >= 1995 AND year <=2010
```

- We can simplify the query like below

```
SELECT * FROM movies WHERE year BETWEEN 1995 AND 2010
```

- Note that 1995 and 2010 are included in the condition

IN

- We can use IN keyword to simplify a series of OR condition to a single field
- Consider the below SQL query

```
SELECT * FROM movies
WHERE
    title='The Lord of the Ring' OR title='Star Wars'
```

- We can simplify the query like below

```
SELECT * FROM movies
WHERE
    title IN ('The Lord of the Ring', 'Star Wars')
```

In-Class Exercise

- Rewrite this query to simplify it using IN and BETWEEN

```
SELECT * FROM movies
WHERE
    year=1995 OR year=1996 OR year=1997
    OR title='The Lord of the Ring' OR title='Star Wars'
```

In-Class Exercise

- Rewrite this query to simplify it using IN and BETWEEN

```
SELECT * FROM movies
WHERE
    year=1995 OR year=1996 OR year=1997
    OR title='The Lord of the Ring' OR title='Star Wars'
```

```
SELECT * FROM movies
WHERE
    year BETWEEN 1995 AND 1997
    OR title IN ('The Lord of the Ring', 'Star Wars')
```

LIKE

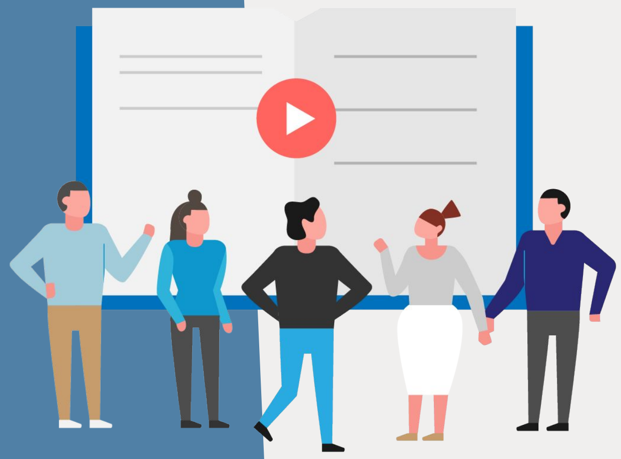
</talentlabs>

LIKE

You can specify a text pattern to the condition using the LIKE keyword

Example 1: we want to select all movies which title is start with “Star Wars” we can query as below

```
SELECT * FROM movies
WHERE title LIKE 'Star Wars%'
```



SQL LIKE with _

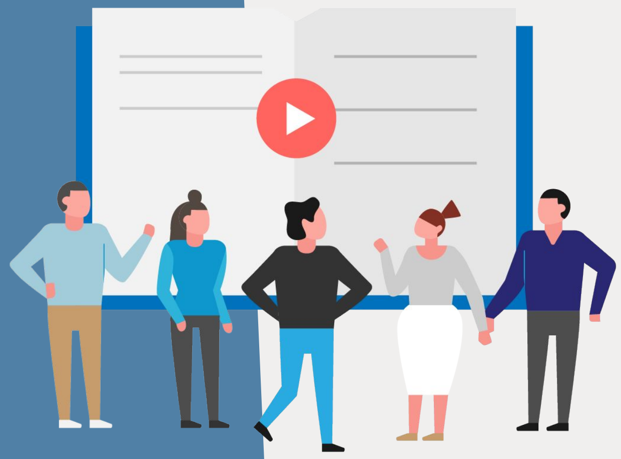
Table: party_name_list

first_name	last_name
Tom	Smith
Jerry	Jones
Lisa	Miller
Tom	Davis
Jerry	Johnson

```
SELECT *
FROM party_name_list
WHERE
    last_name LIKE 'J____'
```

first_name	last_name
Jerry	Jones

- one low dash represents any single character. Here we have 4 i.e. 4 any characters.
- The word “Johnson” has more than 4 characters after “J”. So that row isn't returned



SQL LIKE with %

Table: party_name_list

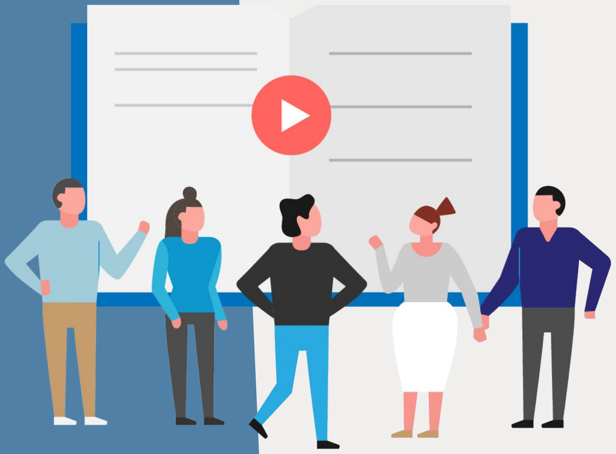
first_name	last_name
Tom	Smith
Jerry	Jones
Lisa	Miller
Tom	Davis
Jerry	Johnson

```
SELECT *
FROM party_name_list
WHERE
    last_name LIKE 'J%'
```

→

first_name	last_name
Jerry	Jones
Jerry	Johnson

- % represents any characters in any length (from 0 to any)



</talentlabs>

SQL LIKE with %

Table: party_name_list

first_name	last_name
Tom	Smith
Jerry	Jones
Lisa	Miller
Tom	Davis
Jerry	Johnson

```
SELECT *
FROM party_name_list
WHERE
    last_name LIKE '%i%'
```

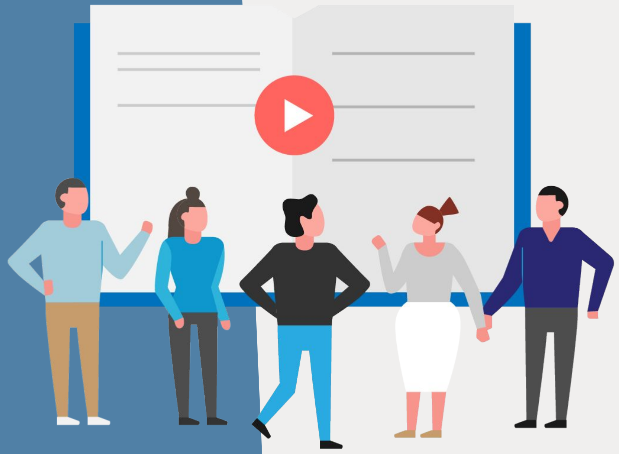
- add % to both side of the keyword for search the keyword appearance in any position

first_name	last_name
Tom	Smith
Lisa	MIller
Tom	Davis

Summary

- We've learnt a few key SQL keywords for filtering data in the database

WHERE	BETWEEN	AND
IN	LIKE	OR



</talentlabs>