

Lesson 1:  
Basic SQL

SEARCH

RESOURCES

CONCEPTS

✓ 29. Solutions: WHERE with Non-N...

✓ 30. Video: Arithmetic Operators

✓ 31. Quiz: Arithmetic Operators

✓ 32. Solutions: Arithmetic Operators

✓ 33. Text: Introduction to Logical O...

✓ 34. Video: LIKE

✓ 35. Quiz: LIKE

✓ 36. Solutions: LIKE

✓ 37. Video: IN

✓ 38. Quiz: IN

✓ 39. Solutions: IN

✓ 40. Video: NOT

✓ 41. Quiz: NOT

Text: Recap & Looking Ahead

## Recap

### Commands

You have already learned a lot about writing code in SQL! Let's take a moment to recap the commands we've covered before moving on:

Statement	How to Use It	Other Details
SELECT	SELECT <b>Col1</b> , <b>Col2</b> , ...	Provide the columns you want to retrieve
FROM	FROM <b>Table</b>	Provide the table where the data is stored
LIMIT	LIMIT <b>10</b>	Limits based number of rows returned
ORDER BY	ORDER BY <b>Col</b>	Orders table based on the column specified. Can use <b>ASC</b> or <b>DESC</b> .
WHERE	WHERE <b>Col</b> > <b>5</b>	A conditional statement to filter rows
LIKE	WHERE <b>Col</b> LIKE '%me%'	Only pulls rows where column value matches the text
IN	WHERE <b>Col</b> IN ('Y', 'N')	A filter for only rows with column value in the list
NOT	WHERE <b>Col</b> NOT IN ('Y', 'N')	<b>NOT</b> is frequently used with <b>IN</b>
AND	WHERE <b>Col1</b> > <b>5</b> AND <b>Col2</b> < <b>3</b>	Filter rows where two or more conditions are true
OR	WHERE <b>Col1</b> > <b>5</b> OR <b>Col2</b> < <b>3</b>	Filter rows where at least one condition is true
BETWEEN	WHERE <b>Col</b> BETWEEN <b>3</b> AND <b>5</b>	Often easier syntax than using <b>AND</b>

### Other Tips

Though SQL is **not case sensitive** (it doesn't care if you write your statements in all lowercase), we discussed some best practices. **The order of the key words does matter**. You know so far, you will want to write your statements as:

```
SELECT col1, col2
FROM table1
WHERE col3 > 5 AND col4 LIKE '%os%'
ORDER BY col5
LIMIT 10;
```

Notice, you can retrieve different columns than those being used in the **ORDER BY** statement. Assuming all of these column names existed in this way (`col1`, `col2`, `col3`, `col4`, `col5`) within a table called `table1`, this query would run just fine.

## Looking Ahead

In the next lesson, you will be learning about **JOINS**. This is the real secret (well, not really) behind the success of SQL as a language. **JOINS** allow us to combine multiple tables. The operations we learned here will still be important moving forward, but we will be able to answer more complex questions by combining information from multiple tables! You have a long way to go - potentially writing your first code ever, but it is about to get so much better.