**Recap**

**Commands**

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

|  |  |  |
| --- | --- | --- |
| **Statement** | **How to Use It** | **Other Details** |
| SELECT | SELECT **Col1**, **Col2**, ... | Provide the columns you want |
| FROM | FROM **Table** | Provide the table where the columns exist |
| LIMIT | LIMIT **10** | Limits based number of rows returned |
| ORDER BY | ORDER BY **Col** | Orders table based on the column. Used with **DESC**. |
| WHERE | WHERE **Col > 5** | A conditional statement to filter your results |
| LIKE | WHERE **Col LIKE '%me%'** | Only pulls rows where column has 'me' within the text |
| IN | WHERE **Col IN ('Y', 'N')** | A filter for only rows with column of 'Y' or 'N' |
| NOT | WHERE **Col NOT IN ('Y', 'N')** | **NOT** is frequently used with **LIKE** and **IN** |
| AND | WHERE **Col1 > 5 AND Col2 < 3** | Filter rows where two or more conditions must be true |
| OR | WHERE **Col1 > 5 OR Col2 < 3** | Filter rows where at least one condition must be true |
| BETWEEN | WHERE **Col BETWEEN 3 AND 5** | Often easier syntax than using an **AND** |

**Other Tips**

Though SQL is **not case sensitive** (it doesn't care if you write your statements as all uppercase or lowercase), we discussed some best practices. **The order of the key words does matter!** Using what 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** and **WHERE**statements. 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.

Recap

Primary and Foreign Keys

You learned a key element for JOINing tables in a database has to do with primary and foreign keys:

primary keys - are unique for every row in a table. These are generally the first column in our database (like you saw with the id column for every table in the Parch & Posey database).

foreign keys - are the primary key appearing in another table, which allows the rows to be non-unique.

Choosing the set up of data in our database is very important, but not usually the job of a data analyst. This process is known as Database Normalization.

JOINs

In this lesson, you learned how to combine data from multiple tables using JOINs. The three JOIN statements you are most likely to use are:

JOIN - an INNER JOIN that only pulls data that exists in both tables.

LEFT JOIN - pulls all the data that exists in both tables, as well as all of the rows from the table in the FROM even if they do not exist in the JOIN statement.

RIGHT JOIN - pulls all the data that exists in both tables, as well as all of the rows from the table in the JOIN even if they do not exist in the FROM statement.

There are a few more advanced JOINs that we did not cover here, and they are used in very specific use cases. UNION and UNION ALL, CROSS JOIN, and the tricky SELF JOIN. These are more advanced than this course will cover, but it is useful to be aware that they exist, as they are useful in special cases.

Alias

You learned that you can alias tables and columns using AS or not using it. This allows you to be more efficient in the number of characters you need to write, while at the same time you can assure that your column headings are informative of the data in your table.

Looking Ahead

The next lesson is your first SQL lab. You have already learned a ton, and it's time to test your knowledge. Go ahead, submit your work and get your certificate of participation!