https://classroom.udacity.com/courses/ud002-bert/lessons/614cf95a-13bf-406c-b092-e757178e633b/concepts/9f672e93-fdcd-4332-a6ef-41edcf8416f1

https://discussions.udacity.com/t/guide-how-to-install-postgresql-and-launch-parch-posey-database/746233/6

LESSON 28 : BASIC SQL (33-49)

Introduction to Logical Operators

In the next concepts, you will be learning about Logical Operators. Logical Operators include:

1. **LIKE: This allows you to perform operations similar to using WHERE and `=`, but for cases when you might not know exactly what you are looking for.**
2. **IN :This allows you to perform operations similar to using WHERE and `=`, but for more than one condition.**
3. **NOT : This is used with IN and LIKE to select all of the rows NOT LIKE or NOT IN a certain condition.**
4. **AND & BETWEEN: These allow you to combine operations where all combined conditions must be true.**

PRO-TIP: Write a filter that will capture all web traffic using LIKE operator.

SELECT * FROM demo WHERE url LIKE '%google%'

LIKE function requires the use of wildcard characters like %

The LIKE operator is extremely useful for working with text. You will use LIKE within a WHERE clause. The LIKE operator is frequently used with `%`. The `%` tells us that we might want any number of characters leading up to a particular set of characters or following a certain set of characters, as we saw with the google

syntax above. Remember you will need to use single quotes for the text you pass to the LIKE operator, because of this lower and uppercase letters are not the same within the string. Searching for 'T' is not the same as searching for 't'. In other SQL environments (outside the classroom), you can use either single or double quotes.

IN=Allows you to filter data based on several possible values

The IN operator is useful for working with both numeric and text columns. This operator allows you to use an `=`, but for more than one item of that particular column. We can check one, two or many column values for which we want to pull data, but all within the same query. In the upcoming concepts, you will see the OR operator that would also allow us to perform these tasks, but the IN operator is a cleaner way to write these queries.

Expert Tip

In most SQL environments, you can use single or double quotation marks - and you may NEED to use double quotation marks if you have an apostrophe within the text you are attempting to pull.

In the work spaces in the classroom, note you can include an apostrophe by putting two single quotes together. Example Macy's in our work space would be 'Macy''s'.

Use the accounts table to find the account `name`, `primary_poc`, and `sales_rep_id` for Walmart, Target, and Nordstrom?

```
SELECT name, primary_poc, sales_rep_id
FROM accounts
WHERE name IN ('Walmart', 'Target', 'Nordstrom');
```

NOT : Provides the inverse results for IN,LIKE and SIMILAR operators

SELECT sales FROM demo WHERE sales NOT IN (10,20) ORDER BY sales

Use the accounts table to find the account name, primary poc, and sales rep id for all stores except Walmart, Target, and Nordstrom?

```
SELECT name, primary_poc, sales_rep_id
FROM accounts
WHERE name NOT IN ('Walmart', 'Target', 'Nordstrom');
```

Use the accounts table to find:All the companies whose names do not start with 'C'.

```
SELECT name
FROM accounts
WHERE name NOT LIKE 'C%';
```

AND : The AND operator is used within a WHERE statement to consider more than one logical clause at a time. Each time you link a new statement with an AND, you will need to specify the column you are interested in looking at. You may link as many statements as you would like to consider at the same time. This operator works with all of the operations we have seen so far including arithmetic operators (+, *, -, /). LIKE, IN, and NOT logic can also be linked together using the AND operator.

BETWEEN Operator

Sometimes we can make a cleaner statement using BETWEEN than we can using AND. Particularly this is true when we are using the same column for different parts of our AND statement. In the previous video, we probably should have used BETWEEN.

Instead of writing :

```
WHERE column >= 6 AND column <= 10
```

we can instead write, equivalently:

```
WHERE column BETWEEN 6 AND 10
```

Use the web_events table to find all information regarding individuals who were contacted via `organic` or `adwords` and started their account at any point in 2016 sorted from newest to oldest?

Solution : You will notice that using BETWEEN is tricky for dates! While BETWEEN is generally inclusive of endpoints, it assumes the time is at 00:00:00 (i.e. midnight) for dates. This is the reason why we set the right-side endpoint of the period at '2017-01-01'.

```
SELECT *
FROM web_events
WHERE channel IN ('organic', 'adwords') AND occurred_at BETWEEN '2016-01-01'
AND '2017-01-01'
ORDER BY occurred_at DESC;
```

`OR:` Similar to the AND operator, the OR operator can combine multiple statements. Each time you link a new statement with an OR, you will need to specify the column you are interested in looking at. You may link as many statements as you would like to consider at the same time. This operator works with all of the operations we have seen so far including arithmetic operators (`+`, `*`, `-`, `/`), LIKE, IN, NOT, AND, and BETWEEN logic can all be linked together using the OR operator.

When combining multiple of these operations, we frequently might need to use parentheses to assure that logic we want to perform is being executed correctly.

Find list of orders ids where either `gloss_qty` or `poster_qty` is greater than 4000. Only include the `id` field in the resulting table?

```
SELECT id
FROM orders
WHERE gloss_qty > 4000 OR poster_qty > 4000;
```

Write a query that returns a list of orders where the `standard_qty` is zero and either the `gloss_qty`or `poster_qty` is over 1000?

```
SELECT *
FROM orders
WHERE standard_qty = 0 AND (gloss_qty > 1000 OR poster_qty > 1000);
```

Find all the company names that start with a 'C' or 'W', and the primary contact contains 'ana' or 'Ana', but it doesn't contain 'eana'?

```
SELECT *
FROM accounts
WHERE (name LIKE 'C%' OR name LIKE 'W%')
           AND ((primary_poc LIKE '%ana%' OR primary_poc LIKE '%Ana%')
           AND primary_poc NOT LIKE '%eana%');
```

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.