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

LESSON 28 : BASIC SQL (15-32)

We have already seen the **SELECT** (to choose columns) and **FROM** (to choose tables) statements. The **LIMIT** statement is useful when you want to see just the first few rows of a table. This can be much faster for loading than if we load the entire dataset.

The **LIMIT** command is always the very last part of a query. An example of showing just the first 10 rows of the orders table with all of the columns might look like the following:

```
SELECT *
FROM orders
LIMIT 10;
```

We could also change the number of rows by changing the 10 to any other number of rows.

https://atom.io/ what is the difference between vim and atom? Why atom is called as the hackable text editor ?

https://www.quora.com/How-does-Atom-compare-to-Vim

https://atom.io/packages/github ,

https://stackoverflow.com/questions/3490505/is-there-a-notepad-with-github-plugin

The **ORDER BY** statement allows us to order our table by any row. If you are familiar with Excel, this is similar to the sorting you can do with filters.

The **ORDER BY** statement is always after the **SELECT** and **FROM** statements, but it is before the **LIMIT** statement. As you learn additional commands, the order of these statements will matter more. If we are using the **LIMIT** statement, it will always appear last.

Pro Tip

Remember `DESC` can be added after the column in your **ORDER BY** statement to sort in descending order, as the default is to sort in ascending order.

SELECT * FROM demo ORDER BY occured_at DESC LIMIT 1000;

Write a query to return the 10 earliest orders in the orders table. Include the `id`, `occurred_at`, and `total_amt_usd`?

```
SELECT id, occurred_at, total_amt_usd
FROM orders
ORDER BY occurred_at
LIMIT 10;
```

Write a query to return the top 5 orders in terms of largest `total_amt_usd`. Include the `id`, `account_id`, and `total_amt_usd`?

```
SELECT id, account_id, total_amt_usd
FROM orders
ORDER BY total_amt_usd DESC
LIMIT 5;
```

Write a query to return the bottom 20 orders in terms of least `total`. Include the `id`, `account_id`, and `total`?

```
SELECT id, account_id, total
FROM orders
ORDER BY total
LIMIT 20;
```

`WHERE : where clause allows you to filter a set of results based on specific criteria`

Common symbols used within WHERE statements include:

1. `>` (greater than)
2. `<` (less than)
3. `>=` (greater than or equal to)
4. `<=` (less than or equal to)
5. `=` (equal to)
6. `!=` (not equal to)

Pull the first 5 rows and all columns from the orders table that have a dollar amount of `gloss_amt_usd` greater than or equal to 1000

```
SELECT *
FROM orders
WHERE gloss_amt_usd >= 1000
LIMIT 5;
```

Pull the first 10 rows and all columns from the orders table that have a `total_amt_usd` less than 500.

```
SELECT *
FROM orders
WHERE total_amt_usd < 500
LIMIT 10;
```

OPERATOR :

If u r using an operator with non-numeric , you need to put the quotes in single quotes.

```
SELECT * FROM demo WHERE name = ‘united’
```

Filter the accounts table to include the company `name`, `website`, and the primary point of contact (`primary_poc`) for Exxon Mobil in the accounts table?

```
SELECT name, website, primary_poc
FROM accounts
WHERE name = 'Exxon Mobil';
```

DERIVED COLUMN:A new column that is a manipulation of the existing columns in your DB.This is done by

SELECT glossy,paper,glossy+paper FROM demo ; (+ operator is used to combine two columns)To rename the newly created column the code can be modified as:

```
SELECT glossy,paper,glossy+paper AS Glu FROM demo;
```

Create a column that divides the `standard_amt_usd` by the `standard_qty` to find the unit price for standard paper for each order. Limit the results to the first 10 orders, and include the `id` and `account_id` fields?

```
SELECT id, account_id, standard_amt_usd/standard_qty AS unit_price
FROM orders
LIMIT 10;
```