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

Post :
https://stackoverflow.com/questions/51490923/postgresql-installation-in-windows-10-64-bit-system

LESSON 28 : BASIC SQL (1-14)

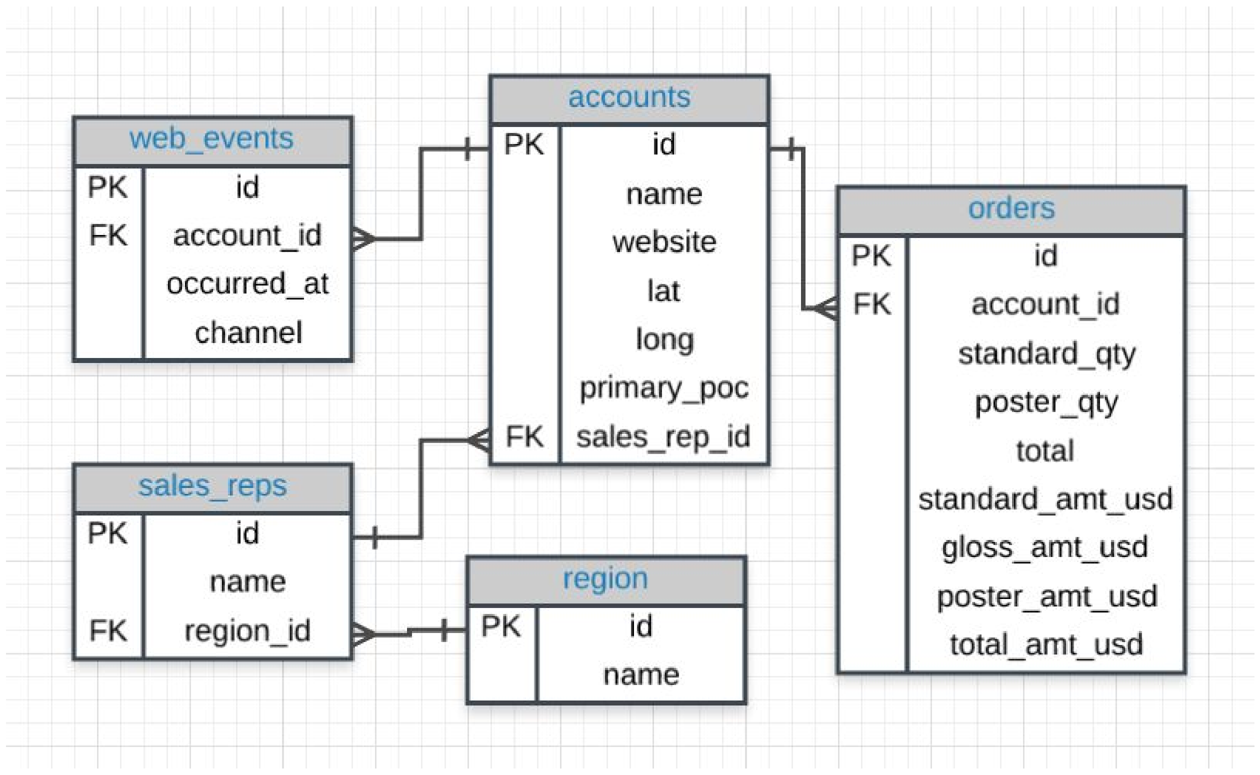
ERD:

An **entity relationship diagram** (ERD) is a common way to view data in a database.A diagram that shows how data is structured in a database.

Below is the ERD for the database we will use from Parch & Posey. These diagrams help you visualize the data you are analyzing including:

1. The names of the tables.
2. The columns in each table.
3. The way the tables work together.

You can think of each of the boxes below as a spreadsheet :



Those three arrow in the picture represents the crow foot notation
Primary_poc=column name
Web_events, sales_reps=table name

Job website : https://www.zippia.com/advice/what-jobs-use-sql/

There are some major advantages to using **traditional relational databases,** which we interact with using SQL. The five most apparent are:

- SQL is easy to understand.
- Traditional databases allow us to access data directly.
- Traditional databases allow us to audit and replicate our data.
- SQL is a great tool for analyzing multiple tables at once.
- SQL allows you to analyze more complex questions than dashboard tools like Google Analytics.

SQL vs. NoSQL

You may have heard of NoSQL, which stands for not only SQL. Databases using NoSQL allow for you to write code that interacts with the data a bit differently than what we will do in this course. These NoSQL environments tend to be particularly popular for web based data, but less popular for data that lives in spreadsheets the way we have been analyzing data up to this point. One of the most popular NoSQL languages is called MongoDB.(https://www.mongodb.com/)

Udacity's free course on MongoDB :
https://in.udacity.com/course/data-wrangling-with-mongodb--ud032

WHY DO BUSINESS CHOOSE SQL :

Business choose sql bcoz of Data integrity

Data integrity is the maintenance of, and the assurance of the accuracy and **consistency** of, **data** over its entire life-cycle, and is a critical aspect to the design, implementation and usage of any system which stores, processes, or retrieves **data**.

Why Businesses Like Databases

1. **Data integrity is ensured -** only the data you want entered is entered, and only certain users are able to enter data into the database.
2. **Data can be accessed quickly -** SQL allows you to obtain results very quickly from the data stored in a database. Code can be optimized to quickly pull results.

3. **Data is easily shared** - multiple individuals can access data stored in a database, and the data is the same for all users allowing for consistent results for anyone with access to your database.

How Databases store data : In DB column all the data in the column must be of the same type. This makes performing analysis on the DB pretty simple.

Doubt : What is meant by a programming framework ?
https://www.quora.com/What-is-the-difference-between-a-programming-language-and-a-framework , https://www.quora.com/What-is-a-framework-in-programming
Postgresql link : https://www.postgresql.org/download/windows/ Some of the most popular databases include: 1.MySQL , 2.Access , 3.Oracle , 4.Microsoft SQL Server,5.Postgres You can also write SQL within other programming frameworks like Python, Scala, and Hadoop .

Small Differences

Each of these SQL databases may have subtle differences in syntax and available functions -- for example, MySQL doesn't have some of the functions for modifying dates as Postgres. **Most** of what you see with Postgres will be directly applicable to using SQL in other frameworks and database environments. For the differences that do exist, you should check the documentation. Most SQL environments have great documentation online that you can easily access with a quick Google search.

The article here compares three of the most common types of SQL: SQLite, PostgreSQL, and MySQL. Though you will use PostgreSQL in the classroom, you will utilize SQLite for the project.

Statements : Tell the DB what u would like to do with the Data

The key to SQL is understanding statements. A few statements include:

1. **CREATE TABLE i**s a statement that creates a new table in a database.
2. **DROP TABLE** is a statement that removes a table in a database.
3. **SELECT** allows you to read data and display it. This is called a query.

https://stackoverflow.com/questions/4735856/difference-between-a-statement-and-a-query-in-sql

A *statement* is the general term for a piece of complete, correct SQL that you can send to a DBMS. A *query* is a statement that will return data, thus a query is a special kind of statement.

A `SELECT ...` would be a query, a `DELETE...` just a statement.

SELECT= filling out a form to get a set of results. When writing out the SELECT statement each of these questions is represented by a single word like SELECT or FROM. These words are called clauses. The FROM clause tells the query what data to use. The SELECT clause tells the query which columns to read from that table.Column names are separated by commas.You can also select all the column names by using *

```
SELECT * FROM demo
```

Here you were introduced to two statements that will be used in every query you write

1. **SELECT** is where you tell the query what columns you want back.
2. **FROM** is where you tell the query what table you are querying from. Notice the columns need to exist in this table.

Formatting Your Queries

Capitalization

You may have noticed that we have been capitalizing SELECT and FROM, while we leave table and column names lowercase. This is a common formatting convention. It is common practice to capitalize commands (SELECT, FROM), and keep everything else in your query lowercase. This makes queries easier to read, which will matter more as you write more complex queries. For now, it is just a good habit to start getting into.

Avoid Spaces in Table and Variable Names

It is common to use underscores and avoid spaces in column names. It is a bit annoying to work with spaces in SQL. In Postgres if you have spaces in column or table names, you need to refer to these columns/tables with double quotes around them (Ex: FROM "Table Name" as opposed to FROM table_name). In other environments, you might see this as square brackets instead (Ex: FROM [Table Name]).

Use White Space in Queries

SQL queries ignore spaces, so you can add as many spaces and blank lines between code as you want, and the queries are the same. This query

```
SELECT account_id FROM orders
```

is equivalent to this query:

```
SELECT account_id
FROM orders
```

and this query (but please don't ever write queries like this):

```
SELECT               account_id

FROM                 orders
```

SQL isn't Case Sensitive

If you have programmed in other languages, you might be familiar with programming languages that get very upset if you do not type the correct characters in terms of lower and uppercase. SQL is not case sensitive. The following query:

```
SELECT account_id
FROM orders
```

is the same as:

```
select account_id
from orders
```

which is also the same as:

```
SeLeCt AcCoUnt_id
FrOm oRdErS
```

However, I would again urge you to follow the conventions outlined earlier in terms of fully capitalizing the commands, while leaving other pieces of your code in lowercase.

Semicolons

Depending on your SQL environment, your query may need a semicolon at the end to execute. Other environments are more flexible in terms of this being a "requirement." It is considered best practices to put a semicolon at the end of each statement, which also allows you to run multiple commands at once if your environment is able to show multiple results at once.

Best practice:

```
SELECT account_id
FROM orders;
```

Since, our environment here doesn't require it, you will see solutions written without the semicolon:

```
SELECT account_id
FROM orders
```