



gdi

INTRO TO SQL SESSION 1

APRIL 18, 2023 – MAY 4TH 2023

WEEK 1: APRIL 28 & APRIL 24TH

INSTRUCTOR: SYLVIA VARGAS SQLSYLVIA@GMAIL.COM

LINKEDIN: [HTTPS://WWW.LINKEDIN.COM/IN/SYLVIAVARGAS/](https://www.linkedin.com/in/sylviavargas/)

BLOG: [HTTPS://SYLVIAVARGAS.COM/](https://sylviavargas.com/)

BLOG (IN PROGRESS): [HTTP://SHESATECHIE.ORG/](http://shesatechie.org/)

ALL SLIDES ON GITHUB: [HTTPS://GITHUB.COM/SQLSYLVIA/GDI-SQL](https://github.com/SQLSYLVIA/GDI-SQL)

GDI'S INTRO TO SQL

- This class will focus on understanding how to use SQL and its basic syntax.
- Topics in this course will include: querying and aggregating data in individual tables, querying multiple related tables (JOINS), and writing subqueries.
- In this course, online SQL database tools will be used to teach students how to write queries. No installation is required.

Welcome!

Girl Develop It is here to provide affordable and accessible programs to learn software through mentorship and hands-on instruction.

Our code of conduct

Some rules

- We are here for you!
- Every question is important
- Help each other
- Have fun

Tell Us About Yourself

- Who are you?
- What's your experience level with SQL?
- What do you hope to get out of the class?
- What is your favorite movie, and how many times have you watched it?

Please fill out the following form so we can collect this information so we can use it to create class database for use in week 3 of our class. [GDI Intro SQL class- Tell us about yourself](#)

About Me

- Retired Sr Data scientist with Microsoft 12/28/2022!
- 40+ years of experience with SQL.
- Worked in tech for over 41 years as a developer, manager and data scientist in a number of companies.

Check out my LinkedIn profile [Sylvia Vargas](#)

- Favorite movie is a three way tie between "West Side Story", "Hamilton" and "Encanto"

- This is a compliment to the database design class
- We'll be using the same terminology and building on those concepts
- But we'll be hands-on



By the end of the class, you will be able to query and modify a

Plan for the week 1: CRUD

- What's a database? What's SQL?
- How to query a database:
 - Select statement (from statement)
 - Select, Distinct
 - Count
 - Group By, Order By
 - Clauses
 - Where
 - Like
 - Joins(Inner/Outer Left/Right)
 - Practice

Plan for the week 2: CRUD

- Super-awesome bonus round:
 - Subqueries
- Practice
- How to modify a database
 - Creating a table
 - Updating a table with rows
 - Deleting Rows VS Deleting Tables
- Optimizing queries
- Resources and links
- Practice

Plan for the week 3: Projects

- Special Projects
- GDI Survey

What's a relational database?

A virtual organized collection of structured information stored electronically that makes data available to people and computers across the organization.



The language of relational databases (1/2)

- Database: A group of data organized in a logical way
(Similar to an Excel workbook, but more structured)
 - Schema: Layout of the database, sometimes interchangeable with database (Excel workbook + name of workbook)
 - Table: A subset of a database (Excel worksheet)
 - Rows (records) and columns (fields): Parts of a table
- CRUD (Create, Read, Update, Delete): How you change the databases and tables in the database
- Relational Data Model : Data related to each other are stored a certain way and "point" to each other (Like Excel cell references)

The language of relational databases (2/2)

- Database keys (primary and foreign): The pointers in each table that can link to other tables
- Indexes: Make databases easier to search
- RDBMS systems: Types of relational databases that have all the properties of relational data models: MySQL, Oracle, Access, Postgres
- SQL is also used in Big Data systems like AWS/Redshift, Azure SQL with Synapse, PySpark/SparkSQL.
- Queries: How to ask questions of the tables and databases

Databases, schemas, and tables

DB Schema → Table → Row → Value

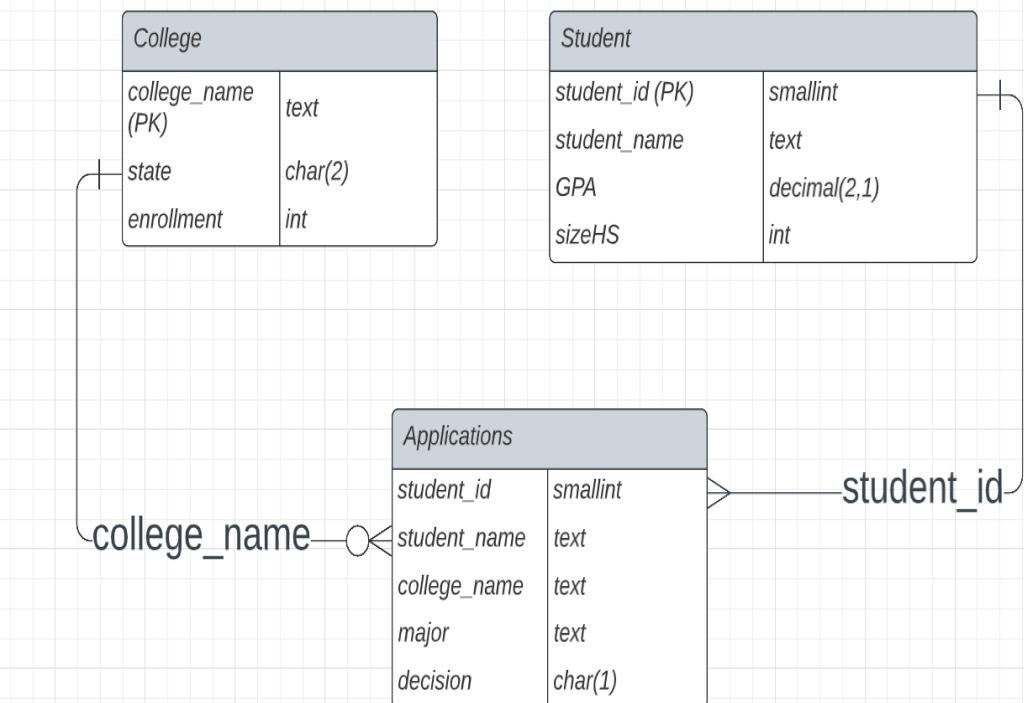


Excel File → Workbook → Row → Cell

WHAT'S A RELATIONAL DATABASE

- *Excel spreadsheets you can join without Vlookups? NO!*
- A relational database is a type of database that stores and provides access to data points that are related to one another.
- Relational databases are based on the relational model, an intuitive, straightforward way of representing data in tables. In a relational database, each row in the table is a record with a unique ID called the key.

College Student Applications Database



DIFFERENT FLAVORS OF SQL

How do you say that you want a can of Coca-Cola?

Coke, soda or pop?

How do you write SQL?

- Every SQL query starts with a SELECT statement.
- But every SQL database vendor has a different "enhancements".

All different flavors of the same language



WHAT'S SQL

- It's the language that the database speaks to bring you back data
- Has a lot of words similar to English, but they have their own meaning as keywords.

```
SELECT user_name,  
SUM(total_purchases)  
FROM billing_database.user_table  
WHERE user_name LIKE 'Henderson%'  
AND billing_month = 'January'  
GROUP BY user_name;
```

Databases, schemas, and tables



The goal of this class is to make you
comfortable with

Databases and CRUD

In the 1980s, as relational databases were being created, the acronym CRUD was developed to define the 4 basic operations of storing data: Create, Read, Update and Delete

- Create : INSERT, CREATE
 - Create Table and Adding(INSERT) rows to your table
- Read: SELECT
 - Picking specific information from your table
- Update: UPDATE, ALTER
 - Changing specifc information in your table
- Delete: DELETE, DROP, TRUNCATE

What can you do with SQL?

The power is in the relationship between data sets. The business value is knowing SQL is that you can change the DATA into INFORMATION for you and businesses to make informed decisions.

- How many customers do we have in Los Angeles?
- What's that person's Facebook and Twitter handle?
- Did profits go up or down this quarter?
- How many users are clicking on our new site?

Body of a SQL statement: SQL clauses and required order

1 SELECT selects variables

2 FROM opens datasets/table

3 WHERE restricts observations

4 GROUP BY groups observations

5 HAVING restricts groups

6 ORDER BY sorts results

Let's develop it!

Let's get started reading tables!

First go to

<https://github.com/sqlsylvia/GDI-SQL> for the online databases we will be using.

Practicing

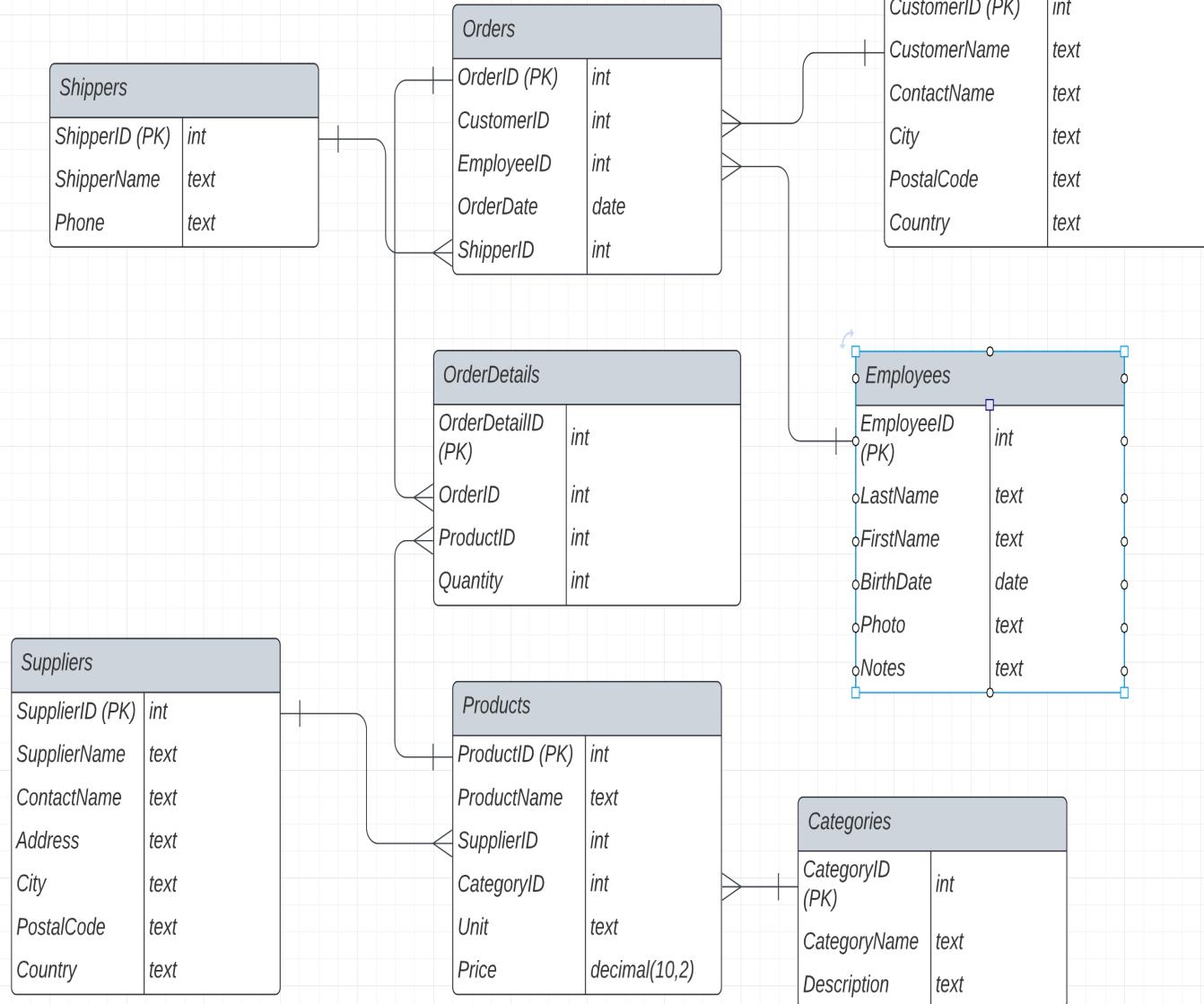
ONLINE SQL TOOLS

- W3Schools has a SQL database to query -
https://www.w3schools.com/sql/trysql.asp?filename=trysql_editor
- DB Fiddle is a site used by companies to assess SQL skills, but it is also an easy way to learning to using SQL as well as create tables.
<https://www.db-fiddle.com/f/6eXpPSRFQgzdKUCjSsbSF4/9>
- REPLIT.com is another way to learn SQL using SQLLite.
I have created some examples for you to learn with.
<https://replit.com/@sqlSylvia/GDI-SQLIntro>

W3 SCHOOLS – DATA BASE

- In your browser open the following url to work with a SQL database in W3Schools.
- https://www.w3schools.com/sql/trysql.asp?filename=trysql_editor

W3Schools - Customer Orders



SELECT * FROM <TABLE>

- **SELECT statement is used to select data from a database.**
- **The FROM command is used to specify which table to select data from.**
- * is used to return all columns of data in the table.
- The absence of a **WHERE** command means return all rows.

Your Database:

Tablename	Records
Customers	91
Categories	8
Employees	10
OrderDetails	518
Orders	196
Products	77
Shippers	3
Suppliers	29

SQL Statement: [Get your own SQL server](#)

```
SELECT * FROM Customers;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

[Run SQL »](#)

Result:

Number of Records: 91

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería del Maestro	Antonio Moreno	El Mburucuyá 7	Panamá	90110	Panama
4	Centro comercial Invertia	Francisco Perez	Avda. 5 de Mayo 2222	Montevideo	11000	Uruguay
5	Ernesto Cane Carnes y Frutas	Ernesto Cane	Avda. Constitución 2222	Montevideo	11000	Uruguay
6	Isabella Giga Gigantour	Isabella Giga	Avda. 5 de Mayo 2222	Montevideo	11000	Uruguay
7	Jalbert Comidas rápidas	Jalbert	Avda. 5 de Mayo 2222	Montevideo	11000	Uruguay
8	La Grande Epicerie de Paris	La Grande Epicerie de Paris	1 rue de la Paix	Paris	75002	France
9	Magazzini Alimentari Del Frantoio	Francesco Sarti	Via Monte Carlo 7	Napoli	80137	Italy
10	Monica	Monica	23 Via Monte Carlo	Napoli	80137	Italy
11	Pastrana Gourmet Etcetera	Pastrana	Avda. 5 de Mayo 2222	Montevideo	11000	Uruguay
12	Reyes	Reyes	Avda. 5 de Mayo 2222	Montevideo	11000	Uruguay
13	Siempre Pescanova	Pere García	Avda. 5 de Mayo 2222	Montevideo	11000	Uruguay
14	Tiedemann Käse und Wurst Fabrik	Kurt Tiedemann	Obere Str. 57	Berlin	12209	Germany
15	Trädgårdsföreningen Blomsterbutiken Hilma	Hilma Karlsson	Årstaviken 2222	Stockholm	11000	Sweden
16	Uncle Bob's Bakery	Bob Belcher	121 8th Ave N.W. Seattle	Seattle	98101	USA
17	Uva	Uva	Avda. 5 de Mayo 2222	Montevideo	11000	Uruguay
18	Vista Alegre Frutos Secos	Manoel Pinto	Avda. 5 de Mayo 2222	Montevideo	11000	Uruguay
19	Wistow Furniture Supply	Wistow Furniture Supply	121 8th Ave N.W. Seattle	Seattle	98101	USA
20	Zen Garden Café	Yukio Yokoyama	121 8th Ave N.W. Seattle	Seattle	98101	USA

SELECT DISTINCT AND ORDER BY

- The DISTINCT clause returns only UNIQUE records (removes duplicate records).
- ORDER BY sorts the results in alphabetical or numeric order depending on table fields/columns in the SELECT statement
- Try the following
 - SELECT DISTINCT City, Country
FROM [Customers]
ORDER BY COUNTRY

SQL Statement:

```
SELECT DISTINCT City, Country FROM [Customers] ORDER BY Country
```

Edit the SQL Statement, and click "Run SQL" to see the result.

[Run SQL »](#)

Result:

Number of Records: 69

City	Country
Buenos Aires	Argentina
Graz	Austria
Salzburg	Austria
Bruxelles	Belgium
Charleroi	Belgium
São Paulo	Brazil
Campinas	Brazil
Rio de Janeiro	Brazil
Resende	Brazil
Tsawassen	Canada

SQL AGGREGATE FUNCTIONS

- TRY the following in Customer Orders Database
- `SELECT MAX(OrderDate) , COUNT(*) FROM [Orders]`
- `SELECT MIN(Price), MAX(Price) , AVG(PRICE) FROM [Products]`
- `SELECT DISTINCT Country FROM [Customers]`

SQL function	returns
AVG()	the mean average of the elements in the column
COUNT()	the total number of elements in the column
DISTINCT()	the number of distinct values across the column
MAX()	the largest-value element in the column
MIN()	the smallest-value element in the column
SUM()	the arithmetic total of all values in the column

USING THE WHERE CLAUSE

The WHERE clause is used to returns the rows meeting the criteria

Try the following and let's discuss your results

- SELECT * from Suppliers WHERE Country = "USA"
- SELECT * FROM [Customers] where City = "London"
- SELECT * FROM [Products] Where Price >= 18
- SELECT * FROM [OrderDetails] where Quantity < 5



SQL OPERATORS: AND, OR, NOT

SQL Operators are used within the WHERE Statement and CASE Statement.

Let's focus on WHERE Statement

```
SELECT * FROM Customers  
WHERE Country = "Argentina" OR  
Country = "Venezuela"
```

```
SELECT * FROM Customers  
WHERE Country = "Venezuela" AND  
PostalCode = 5022
```

LIKE CLAUSE

- The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.
- There are two wildcards often used in conjunction with the LIKE operator:
 - The percent sign (%) represents zero, one, or multiple characters
 - The underscore sign (_) represents one, single character

JOIN – JOINING TABLES

- A JOIN combines columns from one or more tables to a new table.
 - INNER JOIN – returns each row in the two joined tables that have matching column values
 - LEFT OUTER JOIN - The result of a **left outer join** (or simply **left join**) for tables A and B always contains all rows of the "left" table (A), even if the join-condition does not find any matching row in the "right" table (B).
 - RIGHT OUTER JOIN - A **right outer join** (or **right join**) closely resembles a left outer join, except with the treatment of the tables reversed.
 - FULL OUTER JOIN - combines the effect of applying both left and right outer joins. Where rows in the FULL OUTER JOINed tables do not match, the result set will have NULL values for every column of the table that lacks a matching row. For those rows that do match, a single row will be produced in the result set (containing columns populated from both tables).
 - CROSS JOIN – returns the Cartesian product of rows from tables in the join.
In other words, it will produce rows that combine each row from the first table with each row from the second table.

INNER JOIN EXAMPLE

- ```
SELECT ORDERID ,
 SUM(OrderDetails.Quantity *
 Products.Price)

 FROM [OrderDetails]
 JOIN [Products]
 ON OrderDetails.ProductID =
 Products.ProductID
 GROUP BY ORDERID
```

## SQL Statement:

```
SELECT ORDERID, SUM(OrderDetails.Quantity * Products.Price)
FROM [OrderDetails]
JOIN [Products]
 ON OrderDetails.ProductID = Products.ProductID
GROUP BY ORDERID
```

Edit the SQL Statement, and click "Run SQL" to see the result.

[Run SQL »](#)

## Result:

Number of Records: 196

| OrderID | SUM(OrderDetails.Quantity * Products.Price) |
|---------|---------------------------------------------|
| 10248   | 566                                         |
| 10249   | 2329.25                                     |
| 10250   | 2267.25                                     |
| 10251   | 839.5                                       |
| 10252   | 4662.5                                      |
| 10253   | 1806                                        |
| 10254   | 781.5                                       |
| 10255   | 3115.75                                     |
| 10256   | 648                                         |
| 10257   | 1400.5                                      |

# JOINS: OUTER LEFT/RIGHT

```
SELECT Customers.CustomerName, count(Orders.OrderID)
FROM [Customers]
LEFT JOIN [Orders] ON Orders.CustomerID = Customers.CustomerID
GROUP BY Customers.CustomerName
```

ACTION: Re-write this statement using RIGHT JOIN

# WEEK 1: PRACTICE ASSIGNMENT

- Using the W3Schools Database answer the following questions:

1. Which customer has the most number of Orders?
2. Which customer has spent the most money in Orders?
3. Are there any employees that went to Boston College? Who?
4. How many products are there by category? List out CategoryName, and number of products?
5. How many orders used “Speedy Express” Shipper?

What other insights can you provide?

# LEARNING AND PRACTICING: SQL RESOURCES

- Online Databases for learning
  - [https://www.w3schools.com/sql/trysql.asp?filename=trysql\\_editor](https://www.w3schools.com/sql/trysql.asp?filename=trysql_editor)
  - The following 2 databases are the same schema, but in.
    - <https://www.db-fiddle.com/f/6eXpPSRFQgzdKUCjSsbSF4/9> – No login required but you need to save your work elsewhere for backup.
    - <https://replit.com/@sqlSylvia/GDI-SQLIntro>. - replit you need an account to save your work
- SQL Reference information
  - <https://www.w3schools.com/sql/default.asp>
  - <https://www.tutorialspoint.com/sql/index.htm>