

# “Hello, SQL”



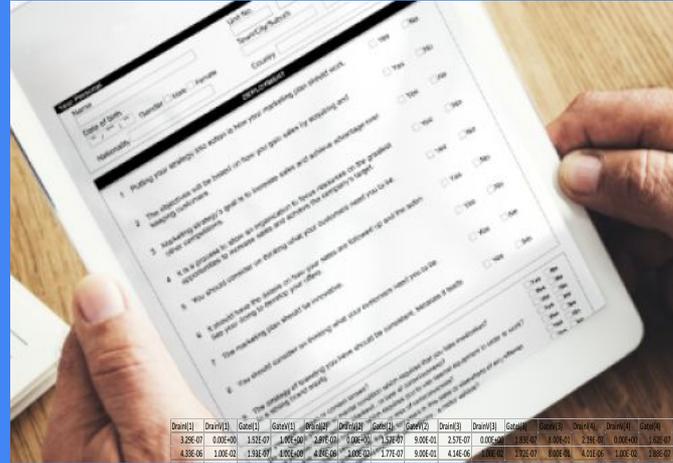
## An Introduction

By Olivia Schulist  
Biostatistics Computing Club Spring 2024  
Columbia Mailman School of Public Health

# Case Study

# Premise

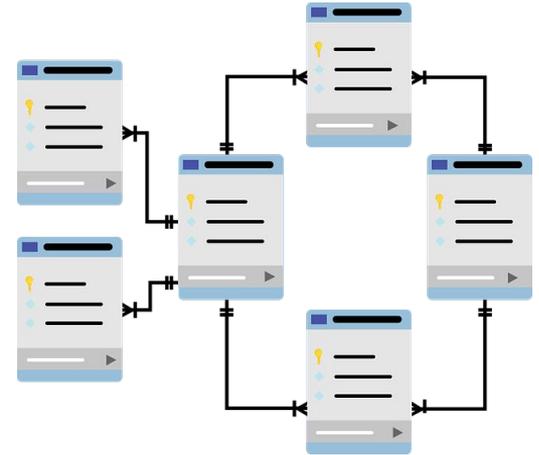
Columbia's MNE research group administers surveys to migrant populations in active transit. The team wishes to create a relational database to store and expand upon their data efficiently, and prepare queries to generate datasets for preliminary data analysis using STATA.



Drawn(1)	Drawn(2)	Case(1)	Case(2)	Drawn(3)	Drawn(4)	Case(3)	Case(4)	Drawn(5)	Drawn(6)	Case(5)	Case(6)	Drawn(7)	Drawn(8)	Case(7)	Case(8)	Drawn(9)	Drawn(10)	Case(9)	Case(10)
3.92E-07	3.00E-06	1.53E-07	1.85E-06	2.87E-07	1.19E-07	5.00E-01	1.57E-07	1.02E-06	1.83E-06	2.79E-07	1.11E-07	7.09E-07	1.14E-07	1.11E-07	1.11E-07	1.11E-07	1.11E-07	1.11E-07	1.11E-07
4.52E-04	1.02E-01	1.34E-07	4.95E-07	4.84E-06	1.07E-07	5.00E-01	1.16E-05	4.69E-07	3.76E-07	8.00E-01	7.12E-06	2.82E-07	7.82E-01	1.83E-04	1.00E-01	1.52E-01	6.00E-01	1.00E-01	1.00E-01
1.27E-05	2.08E-02	1.43E-07	1.94E-04	1.23E-06	2.01E-04	1.22E-07	9.00E-01	1.17E-05	2.07E-02	1.52E-07	8.07E-01	1.11E-05	2.05E-02	1.83E-07	7.89E-01	1.45E-05	2.00E-02	2.18E-07	6.00E-01
1.95E-05	3.00E-02	1.42E-07	1.90E-04	1.88E-05	3.00E-02	1.62E-07	9.00E-01	1.79E-05	3.00E-02	1.62E-07	8.07E-01	1.66E-05	3.00E-02	1.62E-07	7.89E-01	1.54E-05	3.00E-02	1.62E-07	6.00E-01
1.25E-05	4.00E-02	1.32E-07	1.90E-04	2.44E-05	4.00E-02	8.11E-08	9.00E-01	2.33E-05	4.00E-02	1.38E-07	8.07E-01	2.21E-05	4.00E-02	1.62E-07	7.89E-01	2.08E-05	4.00E-02	1.27E-07	6.00E-01
3.18E-05	5.00E-02	1.01E-07	1.90E-04	3.05E-05	5.00E-02	1.27E-07	9.00E-01	2.91E-05	5.00E-02	1.52E-07	8.07E-01	2.78E-05	5.00E-02	1.22E-07	7.89E-01	2.84E-05	5.00E-02	1.98E-07	6.00E-01
1.80E-05	6.00E-02	1.01E-07	1.90E-04	3.88E-05	6.00E-02	1.17E-07	9.00E-01	3.49E-05	6.00E-02	1.62E-07	8.07E-01	3.31E-05	6.00E-02	1.52E-07	7.89E-01	3.12E-05	6.00E-02	1.52E-07	6.00E-01
1.44E-05	7.00E-02	6.98E-08	1.90E-04	4.86E-05	7.00E-02	1.17E-07	9.00E-01	4.07E-05	7.00E-02	1.62E-07	8.07E-01	3.84E-05	7.00E-02	1.56E-07	7.89E-01	3.81E-05	7.00E-02	1.17E-07	6.00E-01
1.07E-05	8.00E-02	1.55E-07	1.90E-04	6.00E-05	8.00E-02	1.17E-07	9.00E-01	4.64E-05	8.00E-02	1.52E-07	8.07E-01	4.36E-05	8.00E-02	1.38E-07	7.89E-01	4.32E-05	8.00E-02	1.37E-07	6.00E-01
1.57E-05	9.00E-02	1.42E-07	1.90E-04	5.47E-05	9.00E-02	1.22E-07	9.00E-01	5.21E-05	9.00E-02	1.62E-07	8.07E-01	4.94E-05	9.00E-02	1.38E-07	7.89E-01	4.83E-05	9.00E-02	1.67E-07	6.00E-01
6.34E-05	1.00E-01	7.61E-08	1.90E-04	6.97E-05	1.00E-01	1.79E-07	9.00E-01	5.79E-05	1.00E-01	1.93E-07	8.07E-01	5.47E-05	1.00E-01	2.03E-07	7.89E-01	5.34E-05	1.00E-01	1.93E-07	6.00E-01
6.97E-05	1.10E-01	1.57E-07	1.90E-04	6.68E-05	1.10E-01	1.47E-07	9.00E-01	6.36E-05	1.10E-01	1.42E-07	8.07E-01	6.01E-05	1.10E-01	2.43E-07	7.89E-01	5.63E-05	1.10E-01	1.27E-07	6.00E-01
7.60E-05	1.20E-01	1.27E-07	1.90E-04	7.28E-05	1.20E-01	1.47E-07	9.00E-01	6.98E-05	1.20E-01	1.37E-07	8.07E-01	6.54E-05	1.20E-01	1.77E-07	7.89E-01	6.16E-05	1.20E-01	1.32E-07	6.00E-01
8.24E-05	1.30E-01	1.06E-07	1.90E-04	7.88E-05	1.30E-01	1.31E-07	9.00E-01	7.51E-05	1.30E-01	1.47E-07	8.07E-01	7.09E-05	1.30E-01	1.83E-07	7.89E-01	6.64E-05	1.30E-01	1.57E-07	6.00E-01
8.86E-05	1.40E-01	1.17E-07	1.90E-04	8.48E-05	1.40E-01	1.42E-07	9.00E-01	8.07E-05	1.40E-01	1.42E-07	8.07E-01	7.61E-05	1.40E-01	1.67E-07	7.89E-01	7.16E-05	1.40E-01	1.12E-07	6.00E-01
9.50E-05	1.50E-01	1.04E-07	1.90E-04	9.09E-05	1.50E-01	1.43E-07	9.00E-01	8.64E-05	1.50E-01	1.62E-07	8.07E-01	8.16E-05	1.50E-01	1.28E-07	7.89E-01	7.64E-05	1.50E-01	1.47E-07	6.00E-01
1.01E-04	1.60E-01	1.01E-07	1.90E-04	9.68E-05	1.60E-01	1.59E-07	9.00E-01	9.21E-05	1.60E-01	1.18E-07	8.07E-01	8.68E-05	1.60E-01	1.09E-07	7.89E-01	8.13E-05	1.60E-01	1.32E-07	6.00E-01
1.08E-04	1.70E-01	8.61E-08	1.90E-04	1.03E-04	1.70E-01	1.32E-07	9.00E-01	9.78E-05	1.70E-01	1.32E-07	8.07E-01	9.22E-05	1.70E-01	1.67E-07	7.89E-01	8.64E-05	1.70E-01	1.88E-07	6.00E-01
1.14E-04	1.80E-01	6.04E-08	1.90E-04	1.09E-04	1.80E-01	1.42E-07	9.00E-01	1.03E-04	1.80E-01	1.72E-07	8.07E-01	9.74E-05	1.80E-01	1.83E-07	7.89E-01	9.13E-05	1.80E-01	1.93E-07	6.00E-01
1.20E-04	1.90E-01	1.22E-07	1.90E-04	1.15E-04	1.90E-01	1.22E-07	9.00E-01	1.09E-04	1.90E-01	1.57E-07	8.07E-01	1.01E-04	1.90E-01	2.33E-07	7.89E-01	9.61E-05	1.90E-01	1.57E-07	6.00E-01
1.27E-04	2.00E-01	1.32E-07	1.90E-04	1.21E-04	2.00E-01	1.31E-07	9.00E-01	1.15E-04	2.00E-01	1.12E-07	8.07E-01	1.08E-04	2.00E-01	2.43E-07	7.89E-01	1.04E-04	2.00E-01	1.47E-07	6.00E-01
1.32E-04	2.10E-01	1.27E-07	1.90E-04	1.27E-04	2.10E-01	1.31E-07	9.00E-01	1.20E-04	2.10E-01	1.32E-07	8.07E-01	1.13E-04	2.10E-01	1.55E-07	7.89E-01	1.09E-04	2.10E-01	1.22E-07	6.00E-01
1.39E-04	2.20E-01	1.42E-07	1.90E-04	1.33E-04	2.20E-01	1.37E-07	9.00E-01	1.26E-04	2.20E-01	1.18E-07	8.07E-01	1.18E-04	2.20E-01	1.65E-07	7.89E-01	1.14E-04	2.20E-01	1.32E-07	6.00E-01
1.45E-04	2.30E-01	1.31E-07	1.90E-04	1.39E-04	2.30E-01	1.48E-07	9.00E-01	1.32E-04	2.30E-01	1.62E-07	8.07E-01	1.24E-04	2.30E-01	1.52E-07	7.89E-01	1.19E-04	2.30E-01	1.57E-07	6.00E-01
1.52E-04	2.40E-01	1.22E-07	1.90E-04	1.45E-04	2.40E-01	1.48E-07	9.00E-01	1.37E-04	2.40E-01	1.77E-07	8.07E-01	1.29E-04	2.40E-01	1.32E-07	7.89E-01	1.24E-04	2.40E-01	1.42E-07	6.00E-01
1.58E-04	2.50E-01	8.61E-08	1.90E-04	1.51E-04	2.50E-01	1.03E-07	9.00E-01	1.43E-04	2.50E-01	1.47E-07	8.07E-01	1.34E-04	2.50E-01	1.62E-07	7.89E-01	1.29E-04	2.50E-01	1.12E-07	6.00E-01
1.64E-04	2.60E-01	1.12E-07	1.90E-04	1.57E-04	2.60E-01	1.77E-07	9.00E-01	1.49E-04	2.60E-01	1.47E-07	8.07E-01	1.39E-04	2.60E-01	1.72E-07	7.89E-01	1.34E-04	2.60E-01	1.12E-07	6.00E-01
1.71E-04	2.70E-01	9.11E-08	1.90E-04	1.63E-04	2.70E-01	1.51E-07	9.00E-01	1.54E-04	2.70E-01	1.57E-07	8.07E-01	1.45E-04	2.70E-01	1.12E-07	7.89E-01	1.39E-04	2.70E-01	1.07E-07	6.00E-01
1.77E-04	2.80E-01	1.01E-07	1.90E-04	1.69E-04	2.80E-01	1.42E-07	9.00E-01	1.60E-04	2.80E-01	1.52E-07	8.07E-01	1.50E-04	2.80E-01	1.02E-07	7.89E-01	1.39E-04	2.80E-01	1.22E-07	6.00E-01
1.83E-04	2.90E-01	1.17E-07	1.90E-04	1.75E-04	2.90E-01	1.12E-07	9.00E-01	1.65E-04	2.90E-01	1.37E-07	8.07E-01	1.55E-04	2.90E-01	9.62E-08	7.89E-01	1.44E-04	2.90E-01	1.17E-07	6.00E-01
1.90E-04	3.00E-01	6.93E-08	1.90E-04	1.81E-04	3.00E-01	1.57E-07	9.00E-01	1.71E-04	3.00E-01	1.72E-07	8.07E-01	1.60E-04	3.00E-01	1.17E-07	7.89E-01	1.49E-04	3.00E-01	1.57E-07	6.00E-01
1.96E-04	3.10E-01	6.94E-08	1.90E-04	1.87E-04	3.10E-01	1.31E-07	9.00E-01	1.76E-04	3.10E-01	1.27E-07	8.07E-01	1.65E-04	3.10E-01	1.09E-07	7.89E-01	1.54E-04	3.10E-01	1.27E-07	6.00E-01
2.02E-04	3.20E-01	1.01E-07	1.90E-04	1.93E-04	3.20E-01	1.12E-07	9.00E-01	1.82E-04	3.20E-01	1.06E-07	8.07E-01	1.70E-04	3.20E-01	8.11E-08	7.89E-01	1.58E-04	3.20E-01	1.37E-07	6.00E-01
2.08E-04	3.30E-01	1.01E-07	1.90E-04	1.99E-04	3.30E-01	1.22E-07	9.00E-01	1.87E-04	3.30E-01	1.27E-07	8.07E-01	1.75E-04	3.30E-01	1.12E-07	7.89E-01	1.61E-04	3.30E-01	1.47E-07	6.00E-01
2.15E-04	3.40E-01	1.21E-07	1.90E-04	2.04E-04	3.40E-01	2.18E-07	9.00E-01	1.93E-04	3.40E-01	1.63E-08	8.07E-01	1.81E-04	3.40E-01	1.52E-07	7.89E-01	1.61E-04	3.40E-01	1.67E-07	6.00E-01
2.22E-04	3.50E-01	9.61E-08	1.90E-04	2.10E-04	3.50E-01	1.71E-07	9.00E-01	1.98E-04	3.50E-01	1.32E-07	8.07E-01	1.86E-04	3.50E-01	1.22E-07	7.89E-01	1.67E-04	3.50E-01	1.77E-07	6.00E-01
2.27E-04	3.60E-01	1.17E-07	1.90E-04	2.16E-04	3.60E-01	1.47E-07	9.00E-01	2.04E-04	3.60E-01	1.17E-07	8.07E-01	1.91E-04	3.60E-01	1.07E-07	7.89E-01	1.71E-04	3.60E-01	1.52E-07	6.00E-01
2.34E-04	3.70E-01	1.04E-07	1.90E-04	2.22E-04	3.70E-01	2.48E-07	9.00E-01	2.09E-04	3.70E-01	1.88E-07	8.07E-01	1.96E-04	3.70E-01	1.42E-07	7.89E-01	1.81E-04	3.70E-01	2.08E-07	6.00E-01
2.40E-04	3.80E-01	9.61E-08	1.90E-04	2.28E-04	3.80E-01	1.31E-07	9.00E-01	2.15E-04	3.80E-01	1.22E-07	8.07E-01	2.01E-04	3.80E-01	9.62E-08	7.89E-01	1.84E-04	3.80E-01	1.77E-07	6.00E-01

# What is a relational database?

- Stores data in tables and relations
- Each row contains unique information
- Minimize redundancies
- Efficient data retrieval
- Queries merge tables based on shared attributes
- Handle big data
- **Attributes** (columns) and **records**/instances (rows)



## Relational Database Management System (RDBMS)

- Types: MySQL (Oracle), SQL Server (Microsoft), Oracle, MS Access



# What is SQL?

Structured Query Language

Standard programming language for  
maintaining a database

Dr. E.F. Codd, IBM, 1970

Query a database to retrieve, add,  
alter, update, delete

```
SELECT  
FROM  
WHERE  
GROUP BY  
HAVING  
ORDER BY
```

# Why SQL?

- **Standard** for **big data** storage and retrieval
- Protections for data **integrity**
- Powers YouTube, Twitter, Facebook
- Integrate with R Shiny app, Python, HTML, R and SAS procedures
- Common in job descriptions
- Similar to the English language
- “Easy to learn”, **intuitive**
- Documentation and resources



# MySQL

- One of the earliest open source relational database management systems (RDBMS) to be developed and launched (1995)
- Supports SQL

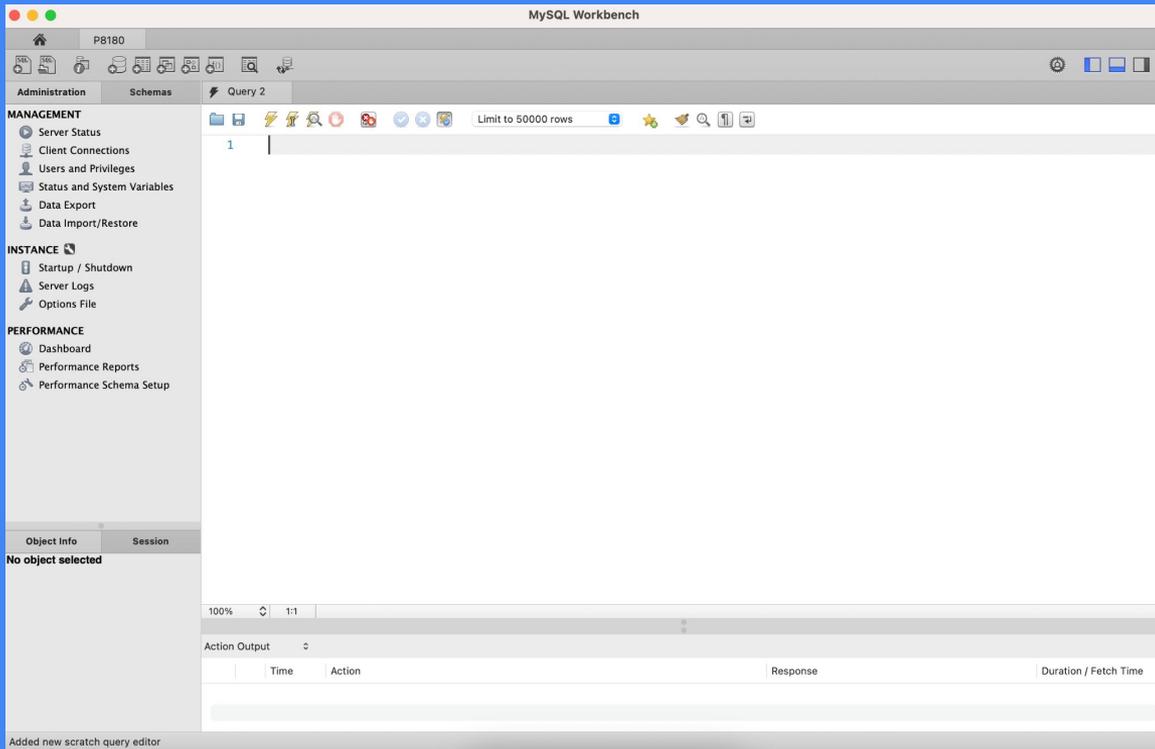


# MySQL Workbench

- Graphical user interface



MySQLWorkbench  
Application - 242.4 MB



# MySQL Workbench



MySQL Workbench interface showing the Administration tab and the Query Editor.

**Administration** | Schemas | Query 2

**MANAGEMENT**

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

**INSTANCE**

- Startup / Shutdown
- Server Logs
- Options File

**PERFORMANCE**

- Dashboard
- Performance Reports
- Performance Schema Setup

Limit to 50000 rows

1

Object Info | Session

No object selected

100% | 1:1

Action Output

Time	Action	Response	Duration / Fetch Time
------	--------	----------	-----------------------

P8180



Administration

Schemas

Query 2

## MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

## INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

## PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Object Info

Session

No object selected



1

100% 1:1

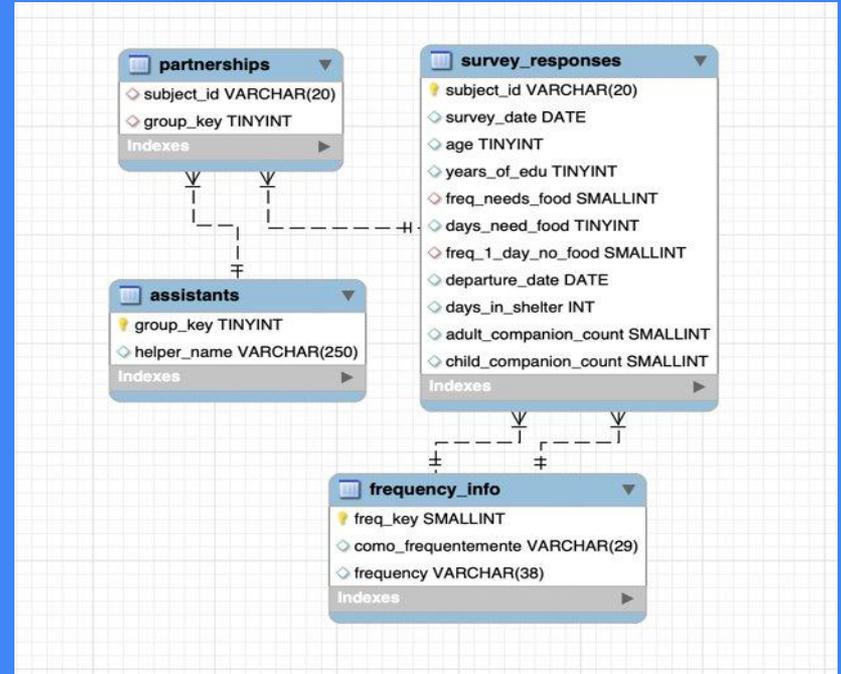
Action Output

Time	Action	Response	Duration / Fetch Time
------	--------	----------	-----------------------

# Relational Database Schema

EER Diagram

Enhanced-Entity Relationship



# Conceptual Framework

## Questionnaire -> Data Entry ->

### Dataset:

author_id	author_name_first	author_name_last	author_email	author_gender	author_nationality	author_website	book_isbn	book_title	author_id	book_copies	book_genre	page_length	publication_year
1000000	Mahmoud	Darwish	<a href="mailto:mdarwish@gmail.com">mdarwish@gmail.com</a>	Male	Palestine	<a href="https://www.poetryfoundation.org/poets/mahmoud-darwish">https://www.poetryfoundation.org/poets/mahmoud-darwish</a>	1-7432-4626-4	In the Presence of Absence	1000000	5	Poetry	200	2006
1000000	Mahmoud	Darwish	<a href="mailto:mdarwish@gmail.com">mdarwish@gmail.com</a>	Male	Palestine	<a href="https://www.poetryfoundation.org/poets/mahmoud-darwish">https://www.poetryfoundation.org/poets/mahmoud-darwish</a>	1-7432-4627-4	The Butterfly's Burden	1000000	1	Poetry	327	2007
1000000	Mahmoud	Darwish	<a href="mailto:mdarwish@gmail.com">mdarwish@gmail.com</a>	Male	Palestine	<a href="https://www.poetryfoundation.org/poets/mahmoud-darwish">https://www.poetryfoundation.org/poets/mahmoud-darwish</a>	1-7432-4628-4	A River Dies of Thirst	1000000	1	Poetry	153	2009
1000000	Mahmoud	Darwish	<a href="mailto:mdarwish@gmail.com">mdarwish@gmail.com</a>	Male	Palestine	<a href="https://www.poetryfoundation.org/poets/mahmoud-darwish">https://www.poetryfoundation.org/poets/mahmoud-darwish</a>	1-7432-4629-4	The Adam of Two Edens	1000000	1	Poetry	206	2000
1000001	Piketty	Thomas	<a href="mailto:tpiketty@gmail.com">tpiketty@gmail.com</a>	Male	France	<a href="http://piketty.pse.ens.fr/en/">http://piketty.pse.ens.fr/en/</a>	1-9352-9204-4	Capital	1000001	1	Economics	696	2013
1000002	Wilde	Oscar	<a href="mailto:owilde@gmail.com">owilde@gmail.com</a>	Male	Ireland	<a href="https://www.poetryfoundation.org/poets/oscar-wilde">https://www.poetryfoundation.org/poets/oscar-wilde</a>	1-4919-2943-4	Only Dull People Are Brilliant at Breakfast	1000002	1	Humor	64	1946
1000003	Kelton	Stephanie	<a href="mailto:skelton@gmail.com">skelton@gmail.com</a>	Female	United States	<a href="https://stephaniekelton.com/">https://stephaniekelton.com/</a>	1-8253-4326-4	The Deficit Myth	1000003	1	Economics	336	2020
1000004	Kahneman	Daniel	<a href="mailto:dkaneman@gmail.com">dkaneman@gmail.com</a>	Male	Israeli-American	<a href="https://kahneman.scholar.princeton.edu/">https://kahneman.scholar.princeton.edu/</a>	1-2075-9388-4	Noise	1000004	1	Nonfiction	464	2021
1000004	Kahneman	Daniel	<a href="mailto:dkaneman@gmail.com">dkaneman@gmail.com</a>	Male	Israeli-American	<a href="https://kahneman.scholar.princeton.edu/">https://kahneman.scholar.princeton.edu/</a>	1-2075-9389-4	Thinking, Fast and Slow	1000004	1	Nonfiction	499	2011
1000004	Kahneman	Daniel	<a href="mailto:dkaneman@gmail.com">dkaneman@gmail.com</a>	Male	Israeli-American	<a href="https://kahneman.scholar.princeton.edu/">https://kahneman.scholar.princeton.edu/</a>	1-2075-9390-4	Choices, Values, and Frames	1000004	1	Nonfiction	860	1984

# Issues?

## Questionnaire -> Data Entry ->

### Dataset:

author_id	author_name_first	author_name_last	author_email	author_gender	author_nationality	author_website	book_isbn	book_title	author_id	book_copies	book_genre	page_length	publication_year
1000000	Mahmoud	Darwish	<a href="mailto:mdarwish@gmail.com">mdarwish@gmail.com</a>	Male	Palestine	<a href="https://www.poetryfoundation.org/poets/mahmoud-darwish">https://www.poetryfoundation.org/poets/mahmoud-darwish</a>	1-7432-4626-4	In the Presence of Absence	1000000	5	Poetry	200	2006
1000000	Mahmoud	Darwish	<a href="mailto:mdarwish@gmail.com">mdarwish@gmail.com</a>	Male	Palestine	<a href="https://www.poetryfoundation.org/poets/mahmoud-darwish">https://www.poetryfoundation.org/poets/mahmoud-darwish</a>	1-7432-4627-4	The Butterfly's Burden	1000000	1	Poetry	327	2007
1000000	Mahmoud	Darwish	<a href="mailto:mdarwish@gmail.com">mdarwish@gmail.com</a>	Male	Palestine	<a href="https://www.poetryfoundation.org/poets/mahmoud-darwish">https://www.poetryfoundation.org/poets/mahmoud-darwish</a>	1-7432-4628-4	A River Dies of Thirst	1000000	1	Poetry	153	2009
1000000	Mahmoud	Darwish	<a href="mailto:mdarwish@gmail.com">mdarwish@gmail.com</a>	Male	Palestine	<a href="https://www.poetryfoundation.org/poets/mahmoud-darwish">https://www.poetryfoundation.org/poets/mahmoud-darwish</a>	1-7432-4629-4	The Adam of Two Edens	1000000	1	Poetry	206	2000
1000001	Piketty	Thomas	<a href="mailto:tpiketty@gmail.com">tpiketty@gmail.com</a>	Male	France	<a href="http://piketty.pse.ens.fr/en/">http://piketty.pse.ens.fr/en/</a>	1-9352-9204-4	Capital	1000001	1	Economics	696	2013
1000002	Wilde	Oscar	<a href="mailto:owilde@gmail.com">owilde@gmail.com</a>	Male	Ireland	<a href="https://www.poetryfoundation.org/poets/oscar-wilde">https://www.poetryfoundation.org/poets/oscar-wilde</a>	1-4919-2943-4	Only Dull People Are Brilliant at Breakfast	1000002	1	Humor	64	1946
1000003	Kelton	Stephanie	<a href="mailto:skelton@gmail.com">skelton@gmail.com</a>	Female	United States	<a href="https://stephaniekelton.com/">https://stephaniekelton.com/</a>	1-8253-4326-4	The Deficit Myth	1000003	1	Economics	336	2020
1000004	Kahneman	Daniel	<a href="mailto:dkaneman@gmail.com">dkaneman@gmail.com</a>	Male	Israeli-American	<a href="https://kahneman.scholar.princeton.edu/">https://kahneman.scholar.princeton.edu/</a>	1-2075-9388-4	Noise	1000004	1	Nonfiction	464	2021
1000004	Kahneman	Daniel	<a href="mailto:dkaneman@gmail.com">dkaneman@gmail.com</a>	Male	Israeli-American	<a href="https://kahneman.scholar.princeton.edu/">https://kahneman.scholar.princeton.edu/</a>	1-2075-9389-4	Thinking, Fast and Slow	1000004	1	Nonfiction	499	2011
1000004	Kahneman	Daniel	<a href="mailto:dkaneman@gmail.com">dkaneman@gmail.com</a>	Male	Israeli-American	<a href="https://kahneman.scholar.princeton.edu/">https://kahneman.scholar.princeton.edu/</a>	1-2075-9390-4	Choices, Values, and Frames	1000004	1	Nonfiction	860	1984

## Issues?

Redundancies

Memory-intensive variable lengths

Number of fields per table can slow query performance  
when queries do not require all fields

# Normalization



*More  
detail  
later!*

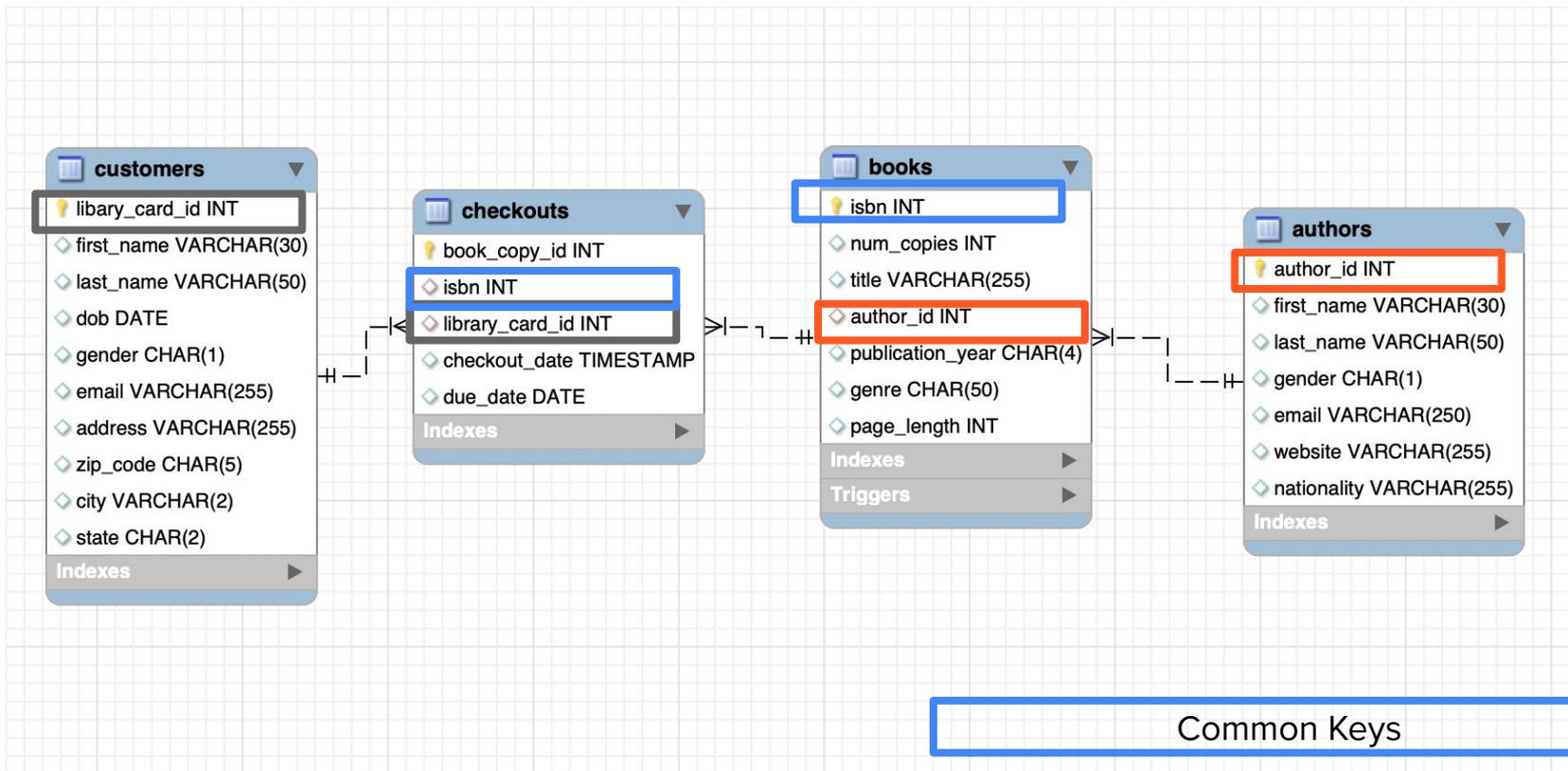
## Basic Principles:

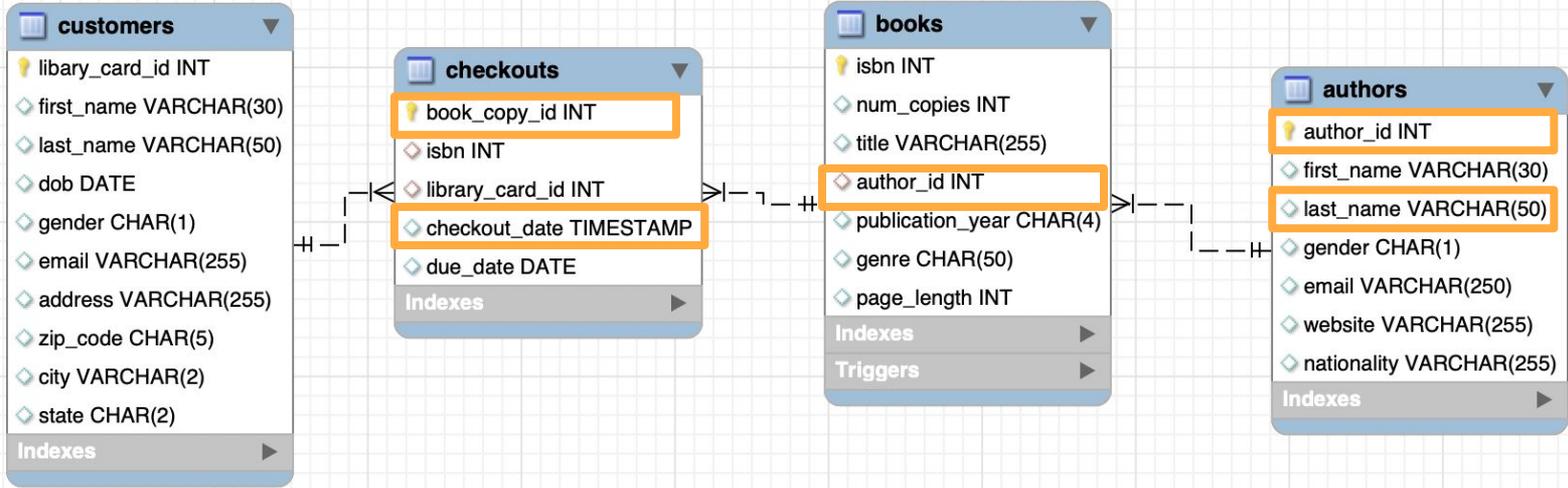
- Separate tables per topic
- Separation of multi-component variables into new variables of smallest component parts
- Character-saving keys for table merges



Can merge  
to find just  
what you  
need later!

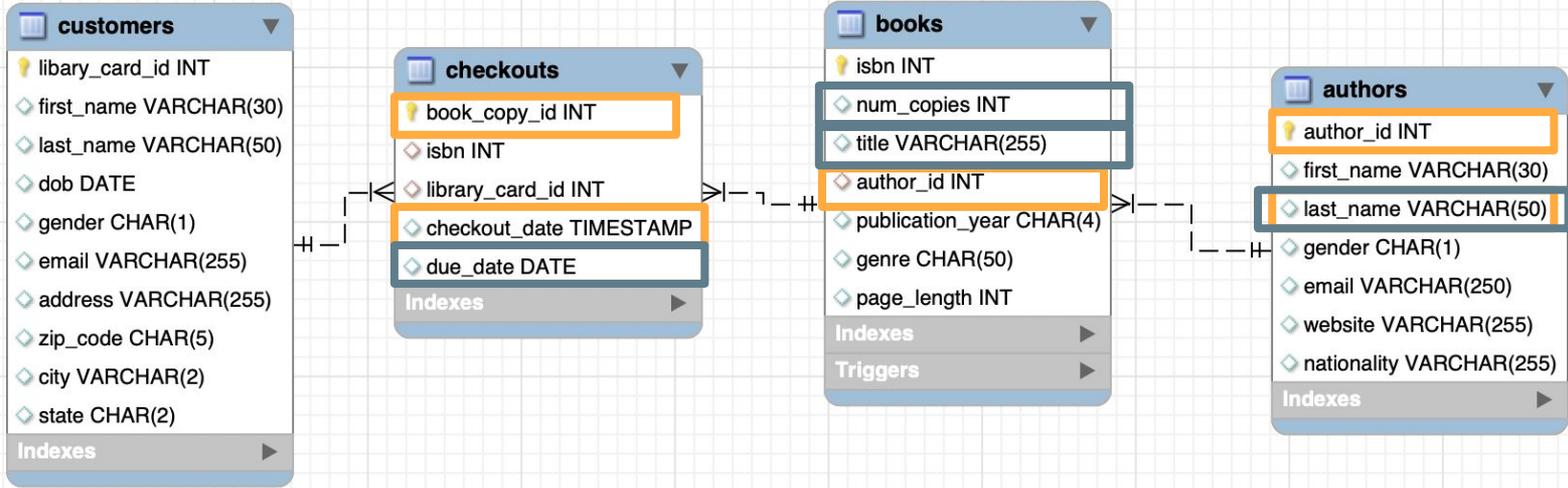
# Library Schema





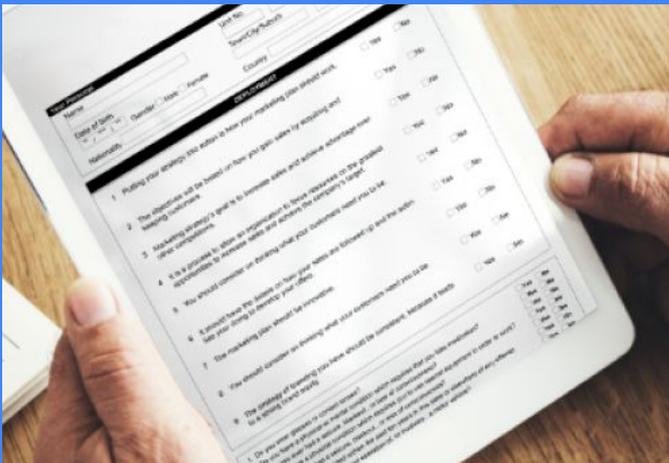
How many books by Dickens were checked out in February?

Can merge to find just what you need later!



Can merge to find just what you need later!

**How many books by Dickens were checked out in February?**  
**Which Dickens titles are completely missing from the shelves and when should each missing title return to the shelves?**



Drain(1)	DrainV(1)	Gate(1)	GateV(1)	Drain(2)	DrainV(2)	Gate(2)	GateV(2)	Drain(3)	DrainV(3)	Gate(3)	GateV(3)	Drain(4)	DrainV(4)	Gate(4)	GateV(4)	Drain(5)	DrainV(5)	Gate(5)	GateV(5)
3.29E-07	0.00E+00	1.52E-07	1.00E+00	2.97E-07	0.00E+00	1.57E-07	9.00E-01	2.57E-07	0.00E+00	1.83E-07	8.00E-01	2.19E-07	0.00E+00	1.62E-07	7.00E-01	1.83E-07	0.00E+00	1.42E-07	6.00E-01
4.33E-06	1.00E-02	1.93E-07	1.00E+00	1.93E-07	1.00E-02	1.77E-07	9.00E-01	4.24E-06	1.00E-02	1.72E-07	8.00E-01	4.01E-06	1.00E-02	1.88E-07	7.00E-01	3.87E-06	1.00E-02	1.52E-07	6.00E-01
1.27E-05	2.00E-02	1.47E-07	1.00E+00	1.22E-05	2.00E-02	1.22E-07	9.00E-01	1.17E-05	2.00E-02	1.52E-07	8.00E-01	1.11E-05	2.00E-02	1.88E-07	7.00E-01	1.05E-05	2.00E-02	2.18E-07	6.00E-01
1.91E-05	3.00E-02	1.42E-07	1.00E+00	1.83E-05	3.00E-02	1.62E-07	9.00E-01	1.75E-05	3.00E-02	1.62E-07	8.00E-01	1.66E-05	3.00E-02	1.47E-07	7.00E-01	1.56E-05	3.00E-02	1.62E-07	6.00E-01
2.55E-05	4.00E-02	1.32E-07	1.00E+00	2.44E-05	4.00E-02	8.11E-08	9.00E-01	2.33E-05	4.00E-02	2.38E-07	8.00E-01	2.21E-05	4.00E-02	1.62E-07	7.00E-01	2.08E-05	4.00E-02	1.27E-07	6.00E-01
3.18E-05	5.00E-02	1.01E-07	1.00E+00	3.05E-05	5.00E-02	1.27E-07	9.00E-01	2.91E-05	5.00E-02	1.52E-07	8.00E-01	2.76E-05	5.00E-02	1.22E-07	7.00E-01	2.60E-05	5.00E-02	1.98E-07	6.00E-01
3.81E-05	6.00E-02	1.01E-07	1.00E+00	3.66E-05	6.00E-02	1.17E-07	9.00E-01	3.49E-05	6.00E-02	1.62E-07	8.00E-01	3.31E-05	6.00E-02	1.52E-07	7.00E-01	3.11E-05	6.00E-02	1.52E-07	6.00E-01
4.45E-05	7.00E-02	6.08E-08	1.00E+00	4.26E-05	7.00E-02	1.72E-07	9.00E-01	4.07E-05	7.00E-02	1.62E-07	8.00E-01	3.85E-05	7.00E-02	1.06E-07	7.00E-01	3.62E-05	7.00E-02	1.17E-07	6.00E-01
5.07E-05	8.00E-02	1.52E-07	1.00E+00	4.86E-05	8.00E-02	1.17E-07	9.00E-01	4.64E-05	8.00E-02	1.52E-07	8.00E-01	4.39E-05	8.00E-02	1.98E-07	7.00E-01	4.13E-05	8.00E-02	1.52E-07	6.00E-01
5.71E-05	9.00E-02	1.62E-07	1.00E+00	5.47E-05	9.00E-02	1.22E-07	9.00E-01	5.21E-05	9.00E-02	1.62E-07	8.00E-01	4.94E-05	9.00E-02	1.88E-07	7.00E-01	4.63E-05	9.00E-02	1.67E-07	6.00E-01
6.34E-05	1.00E-01	7.61E-08	1.00E+00	6.07E-05	1.00E-01	1.72E-07	9.00E-01	5.79E-05	1.00E-01	1.93E-07	8.00E-01	5.47E-05	1.00E-01	2.03E-07	7.00E-01	5.14E-05	1.00E-01	1.62E-07	6.00E-01
6.97E-05	1.10E-01	1.57E-07	1.00E+00	6.68E-05	1.10E-01	1.47E-07	9.00E-01	6.36E-05	1.10E-01	1.42E-07	8.00E-01	6.01E-05	1.10E-01	2.43E-07	7.00E-01	5.65E-05	1.10E-01	1.27E-07	6.00E-01
7.60E-05	1.20E-01	1.27E-07	1.00E+00	7.29E-05	1.20E-01	1.47E-07	9.00E-01	6.93E-05	1.20E-01	1.37E-07	8.00E-01	6.56E-05	1.20E-01	1.77E-07	7.00E-01	6.15E-05	1.20E-01	1.32E-07	6.00E-01
8.24E-05	1.30E-01	1.06E-07	1.00E+00	7.88E-05	1.30E-01	1.32E-07	9.00E-01	7.51E-05	1.30E-01	1.47E-07	8.00E-01	7.09E-05	1.30E-01	1.83E-07	7.00E-01	6.64E-05	1.30E-01	1.57E-07	6.00E-01
8.88E-05	1.40E-01	1.12E-07	1.00E+00	8.48E-05	1.40E-01	1.42E-07	9.00E-01	8.07E-05	1.40E-01	1.42E-07	8.00E-01	7.62E-05	1.40E-01	1.67E-07	7.00E-01	7.15E-05	1.40E-01	1.17E-07	6.00E-01
9.50E-05	1.50E-01	1.06E-07	1.00E+00	9.09E-05	1.50E-01	1.83E-07	9.00E-01	8.64E-05	1.50E-01	1.62E-07	8.00E-01	8.16E-05	1.50E-01	2.38E-07	7.00E-01	7.64E-05	1.50E-01	1.42E-07	6.00E-01
1.01E-04	1.60E-01	1.01E-07	1.00E+00	9.69E-05	1.60E-01	1.93E-07	9.00E-01	9.21E-05	1.60E-01	2.18E-07	8.00E-01	8.69E-05	1.60E-01	1.06E-07	7.00E-01	8.13E-05	1.60E-01	1.32E-07	6.00E-01
1.08E-04	1.70E-01	8.62E-08	1.00E+00	1.03E-04	1.70E-01	1.32E-07	9.00E-01	9.78E-05	1.70E-01	1.32E-07	8.00E-01	9.22E-05	1.70E-01	1.67E-07	7.00E-01	8.64E-05	1.70E-01	1.88E-07	6.00E-01
1.14E-04	1.80E-01	6.08E-08	1.00E+00	1.09E-04	1.80E-01	1.42E-07	9.00E-01	1.03E-04	1.80E-01	1.72E-07	8.00E-01	9.76E-05	1.80E-01	1.83E-07	7.00E-01	9.13E-05	1.80E-01	1.93E-07	6.00E-01
1.20E-04	1.90E-01	1.22E-07	1.00E+00	1.15E-04	1.90E-01	1.22E-07	9.00E-01	1.09E-04	1.90E-01	1.57E-07	8.00E-01	1.03E-04	1.90E-01	2.33E-07	7.00E-01	9.61E-05	1.90E-01	1.57E-07	6.00E-01
1.27E-04	2.00E-01	1.32E-07	1.00E+00	1.21E-04	2.00E-01	1.32E-07	9.00E-01	1.15E-04	2.00E-01	1.12E-07	8.00E-01	1.08E-04	2.00E-01	2.43E-07	7.00E-01	1.01E-04	2.00E-01	1.47E-07	6.00E-01
1.33E-04	2.10E-01	1.77E-07	1.00E+00	1.27E-04	2.10E-01	1.32E-07	9.00E-01	1.20E-04	2.10E-01	1.32E-07	8.00E-01	1.13E-04	2.10E-01	1.57E-07	7.00E-01	1.06E-04	2.10E-01	1.83E-07	6.00E-01
1.39E-04	2.20E-01	1.42E-07	1.00E+00	1.33E-04	2.20E-01	1.37E-07	9.00E-01	1.26E-04	2.20E-01	1.88E-07	8.00E-01	1.19E-04	2.20E-01	1.88E-07	7.00E-01	1.11E-04	2.20E-01	1.52E-07	6.00E-01
1.45E-04	2.30E-01	1.32E-07	1.00E+00	1.39E-04	2.30E-01	1.88E-07	9.00E-01	1.32E-04	2.30E-01	1.62E-07	8.00E-01	1.24E-04	2.30E-01	1.52E-07	7.00E-01	1.15E-04	2.30E-01	1.57E-07	6.00E-01
1.52E-04	2.40E-01	1.22E-07	1.00E+00	1.45E-04	2.40E-01	1.88E-07	9.00E-01	1.37E-04	2.40E-01	1.77E-07	8.00E-01	1.29E-04	2.40E-01	1.32E-07	7.00E-01	1.20E-04	2.40E-01	1.42E-07	6.00E-01
1.58E-04	2.50E-01	8.62E-08	1.00E+00	1.51E-04	2.50E-01	2.03E-07	9.00E-01	1.43E-04	2.50E-01	1.47E-07	8.00E-01	1.34E-04	2.50E-01	1.62E-07	7.00E-01	1.25E-04	2.50E-01	1.27E-07	6.00E-01
1.64E-04	2.60E-01	1.12E-07	1.00E+00	1.57E-04	2.60E-01	1.77E-07	9.00E-01	1.49E-04	2.60E-01	1.47E-07	8.00E-01	1.39E-04	2.60E-01	1.72E-07	7.00E-01	1.30E-04	2.60E-01	1.12E-07	6.00E-01
1.71E-04	2.70E-01	9.13E-08	1.00E+00	1.63E-04	2.70E-01	1.52E-07	9.00E-01	1.54E-04	2.70E-01	1.57E-07	8.00E-01	1.45E-04	2.70E-01	1.12E-07	7.00E-01	1.35E-04	2.70E-01	1.67E-07	6.00E-01
1.77E-04	2.80E-01	1.01E-07	1.00E+00	1.69E-04	2.80E-01	1.42E-07	9.00E-01	1.60E-04	2.80E-01	1.52E-07	8.00E-01	1.50E-04	2.80E-01	1.06E-07	7.00E-01	1.39E-04	2.80E-01	1.52E-07	6.00E-01
1.83E-04	2.90E-01	1.17E-07	1.00E+00	1.75E-04	2.90E-01	1.12E-07	9.00E-01	1.65E-04	2.90E-01	1.37E-07	8.00E-01	1.55E-04	2.90E-01	9.63E-08	7.00E-01	1.44E-04	2.90E-01	1.77E-07	6.00E-01
1.90E-04	3.00E-01	6.99E-08	1.00E+00	1.81E-04	3.00E-01	1.93E-07	9.00E-01	1.71E-04	3.00E-01	1.72E-07	8.00E-01	1.60E-04	3.00E-01	1.17E-07	7.00E-01	1.49E-04	3.00E-01	1.57E-07	6.00E-01
1.96E-04	3.10E-01	6.08E-08	1.00E+00	1.87E-04	3.10E-01	1.32E-07	9.00E-01	1.76E-04	3.10E-01	1.27E-07	8.00E-01	1.65E-04	3.10E-01	1.06E-07	7.00E-01	1.54E-04	3.10E-01	1.22E-07	6.00E-01
2.02E-04	3.20E-01	1.01E-07	1.00E+00	1.92E-04	3.20E-01	1.12E-07	9.00E-01	1.82E-04	3.20E-01	1.06E-07	8.00E-01	1.70E-04	3.20E-01	8.11E-08	7.00E-01	1.58E-04	3.20E-01	1.37E-07	6.00E-01
2.09E-04	3.30E-01	1.01E-07	1.00E+00	1.99E-04	3.30E-01	1.22E-07	9.00E-01	1.87E-04	3.30E-01	1.27E-07	8.00E-01	1.75E-04	3.30E-01	1.12E-07	7.00E-01	1.63E-04	3.30E-01	1.47E-07	6.00E-01
2.15E-04	3.40E-01	1.22E-07	1.00E+00	2.04E-04	3.40E-01	2.18E-07	9.00E-01	1.93E-04	3.40E-01	9.63E-08	8.00E-01	1.81E-04	3.40E-01	1.52E-07	7.00E-01	1.67E-04	3.40E-01	1.67E-07	6.00E-01
2.21E-04	3.50E-01	9.63E-08	1.00E+00	2.10E-04	3.50E-01	1.77E-07	9.00E-01	1.99E-04	3.50E-01	1.32E-07	8.00E-01	1.85E-04	3.50E-01	1.27E-07	7.00E-01	1.72E-04	3.50E-01	1.77E-07	6.00E-01
2.27E-04	3.60E-01	1.17E-07	1.00E+00	2.16E-04	3.60E-01	1.47E-07	9.00E-01	2.04E-04	3.60E-01	1.77E-07	8.00E-01	1.90E-04	3.60E-01	1.57E-07	7.00E-01	1.77E-04	3.60E-01	1.52E-07	6.00E-01
2.34E-04	3.70E-01	1.05E-07	1.00E+00	2.22E-04	3.70E-01	2.48E-07	9.00E-01	2.09E-04	3.70E-01	1.88E-07	8.00E-01	1.96E-04	3.70E-01	1.42E-07	7.00E-01	1.86E-04	3.70E-01	2.08E-07	6.00E-01
2.40E-04	3.80E-01	9.63E-08	1.00E+00	2.28E-04	3.80E-01	1.32E-07	9.00E-01	2.15E-04	3.80E-01	1.22E-07	8.00E-01	2.01E-04	3.80E-01	9.63E-08	7.00E-01	1.86E-04	3.80E-01	1.77E-07	6.00E-01

# MINE Example

# Process

- 1. Identify and name variables**
- 2. Create topical tables**
- 3. Establish table relationships**
- 4. Create indices**
- 5. Prepare queries for use cases**

## Select Variables

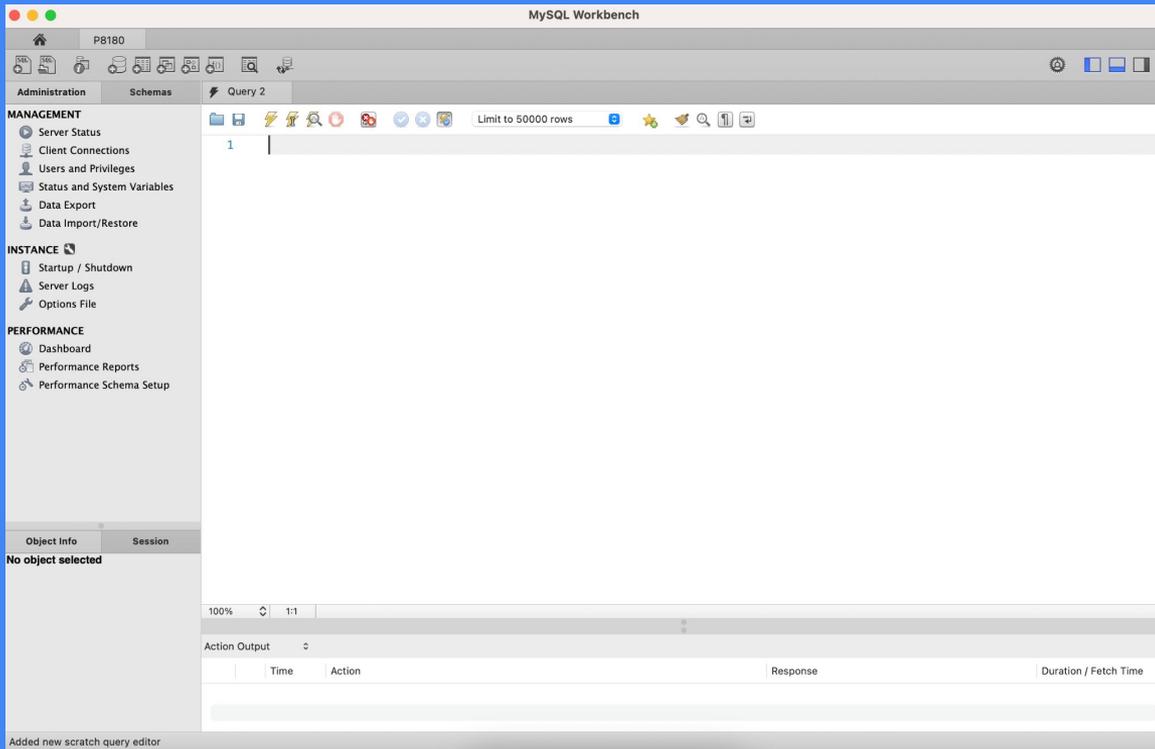
- Elements that vary by subject
- Keys for repetitive strings

## Name Variables

- Letters and numbers, start with a **letter**
- **Underscores**, no periods
- **Singular** descriptors (ex: store\_name)
- Lowercase **snake\_case** over camelCase or PascalCase

## Cluster Variables

- Plural descriptors for tables (ex: stores)

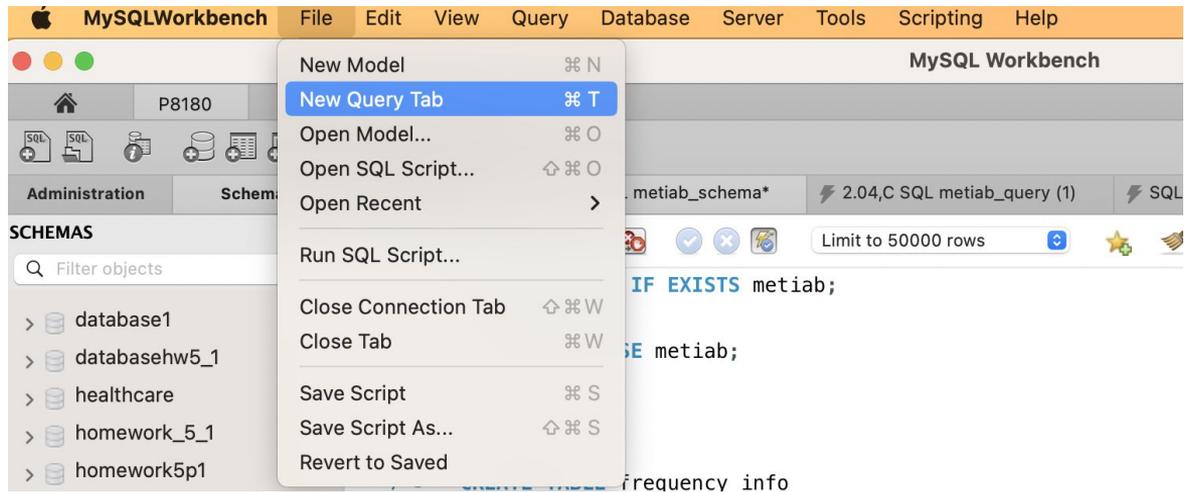


# MySQL Workbench



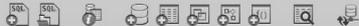


MySQLWorkbench  
Application - 242.4 MB





P8180



Administration Schemas Query 2 2.04,C SQL metiab\_schema\* 2.04,C SQL metiab\_query (1) SQL File 5

**SCHEMAS**

Filter objects

- > database1
- > databasehw5\_1
- > healthcare
- > homework\_5\_1
- > homework5p1
- > iOS
- > library
- > **metiab**
  - > Tables
  - > Views
  - > Stored Procedures
  - > Functions
- > netflix\_data
- > retail
- > rOS
- > sakila
- > sys



Limit to 50000 rows

1

Object Info Session

Schema: metiab

100%



1:1

Action Output



Time	Action	Response	Duration / Fetch Time

```
CREATE DATABASE metiab;
```

```
USE metiab;
```

Query 2    2.04,C SQL metiab\_schema\*    2.04,C

          Limit to 50

1

Execute the selected portion of the script or everything, if there is no selection

Administration    Schemas

**SCHEMAS** 

Q Filter objects

- > database1
- > databasehw5\_1
- > healthcare
- > homework\_5\_1
- > homework5p1
- > iOS
- > library
- ▼ **metiab**
  - > Tables
  - > Views
  - > Stored Procedures
  - > Functions
- > netflix\_data
- > retail
- > rOS
- > sakila
- > sys

# Table Creation

Simple Example:



```
CREATE TABLE assistants  
(  
    group_key TINYINT PRIMARY KEY,  
    helper_name VARCHAR(250)  
);
```



# Variable Definition

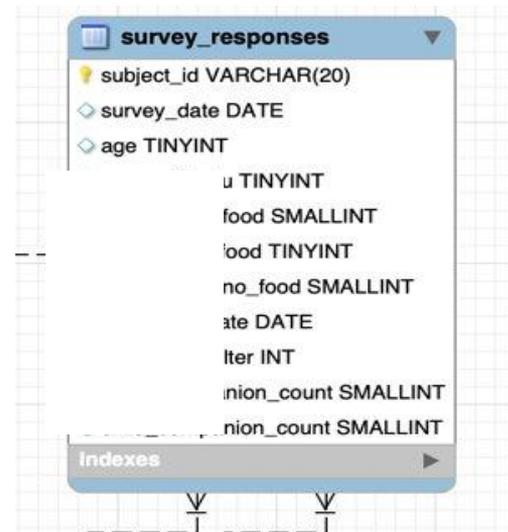
## Types:

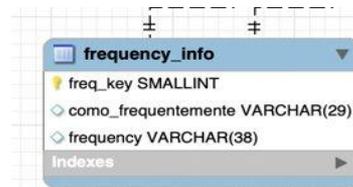
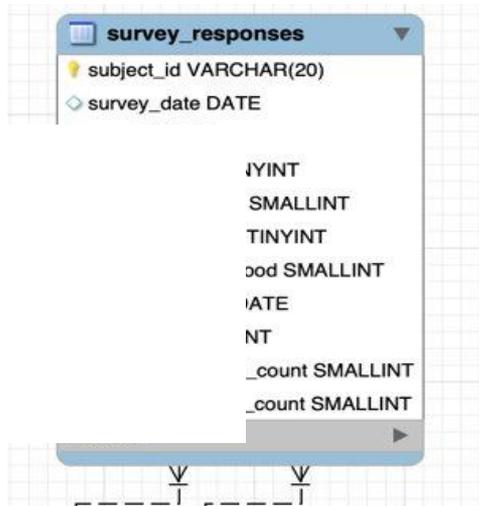
- **CHAR, VARCHAR** (variable length)
- **INT, TINYINT, SMALLINT, BIGINT**
- **DECIMAL, FLOAT** (approximate)
- **DATE, DATETIME, TIMESTAMP**

## Properties:

- **UNSIGNED** to reduce storage load
- **UNIQUE** constraint
- **DEFAULT**
- **FOREIGN KEY, PRIMARY KEY:**

```
subject_id VARCHAR(20) PRIMARY KEY,  
survey_date DATE,  
age TINYINT UNSIGNED,  
    u TINYINT UNSIGNED,  
    food SMALLINT UNSIGNED,  
    food TINYINT UNSIGNED,  
no_food SMALLINT UNSIGNED,  
date DATE,  
iter INT,  
union_count SMALLINT UNSIGNED,  
union_count SMALLINT UNSIGNED,
```





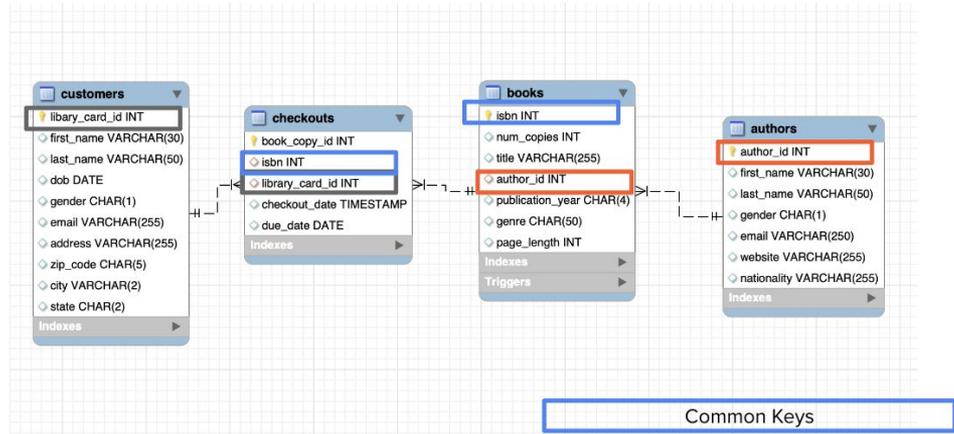
# Table Relationships

## Primary Key

- Column(s) whose values act as unique row identifiers
- “No duplicates, ever” after normalization
- **Composite primary key:** multiple variables that in concert uniquely identify rows
- **Autonumber primary key:** gives each row a unique identifier number

## Foreign Key

- Links to another table

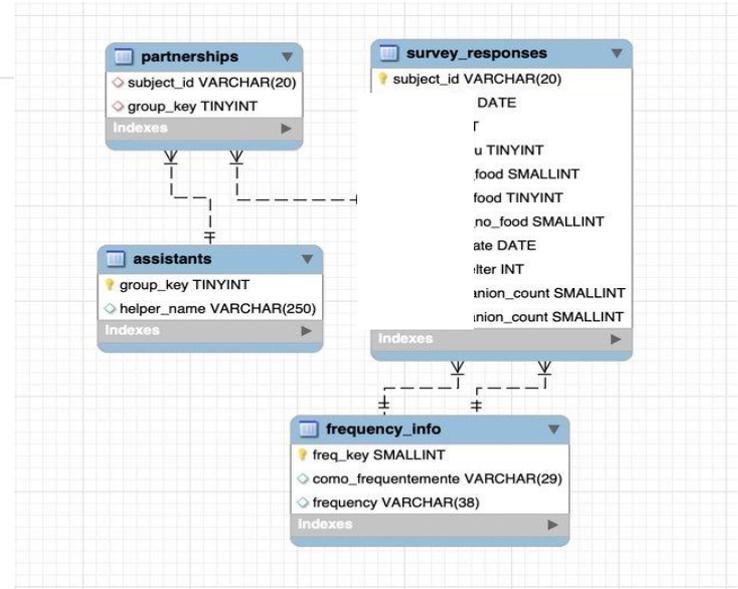


- **CREATE TABLE** survey\_responses

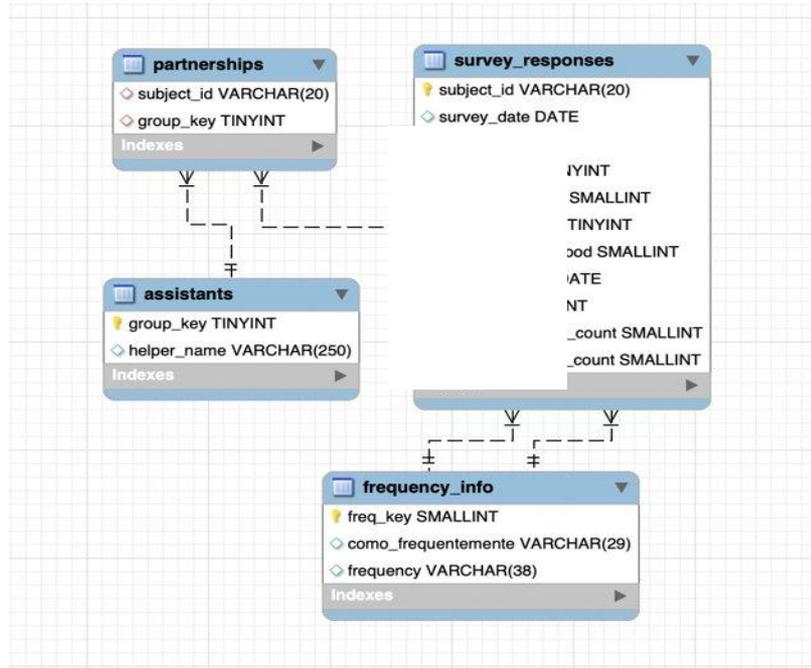
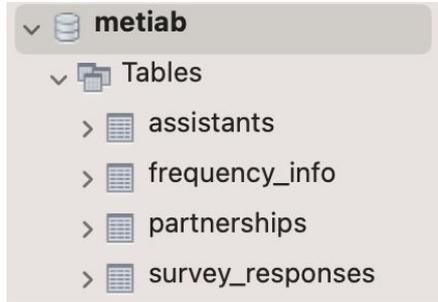
(

```
subject_id VARCHAR(20) PRIMARY KEY,  
survey_date DATE,  
u TINYINT UNSIGNED,  
food SMALLINT UNSIGNED,  
no_food TINYINT UNSIGNED,  
date DATE,  
alter INT,  
union_count SMALLINT UNSIGNED,  
_ union_count SMALLINT UNSIGNED,  
FOREIGN KEY (freq_needs_food) REFERENCES frequency_info(freq_key),  
FOREIGN KEY (freq_1_day_no_food) REFERENCES frequency_info(freq_key)
```

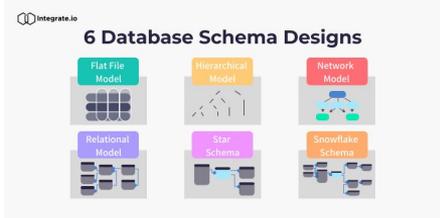
);



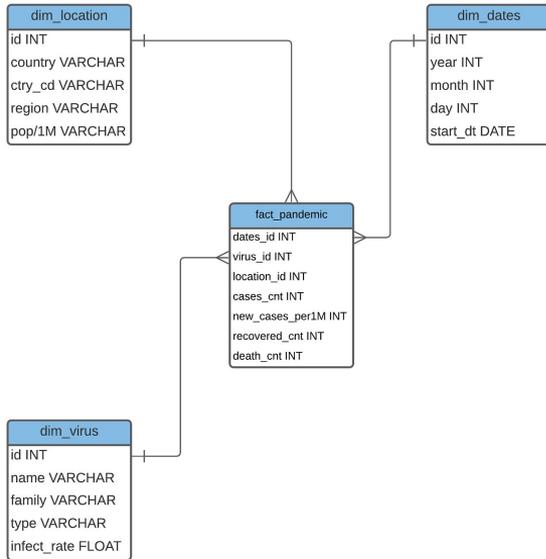
# Migration Example:



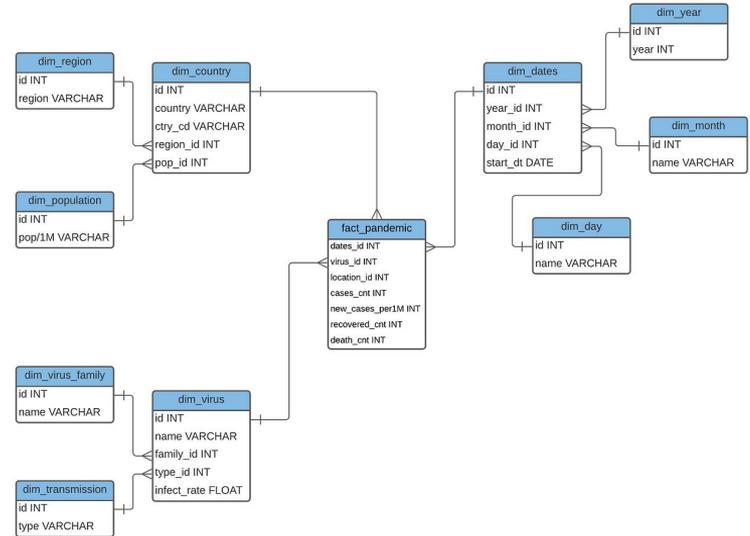
# Schema Examples



## Star



## Snowflake



# DQL Syntax

**“Hello, SQL”**

# Terminology

**DDL** - Data Definition Language - CREATE database, ALTER table, DROP variable, TRUNCATE, RENAME

**DQL** - Data Query Language - SELECT, WHERE, JOIN, GROUP BY, ORDER BY

**DML** - Data Manipulation Language - SELECT, INSERT, UPDATE, DELETE

**DCL** - Data Control Language - GRANT, REVOKE

**TCL** - Transaction Control Language - COMMIT, ROLLBACK, SAVEPOINT, SET TRANSACTION

## Inserting Values

```
INSERT INTO table_name  
    (column_1, column_2)  
VALUES (value_1, value_2);
```

```
INSERT INTO staff (staff_id,  
    last_name, first_name, age)  
VALUES (1, 'Smith', 'Bob', 32),  
(2, 'Jacobs', 'Elizabeth', 21);
```

## Editing Tables

```
UPDATE staff  
SET first_name = Robert  
WHERE staff_id = 1;
```

```
ALTER TABLE table_name  
ADD [COLUMN] column_name  
    data_type [FIRST | AFTER  
Column_name];
```

# Basics

- Command words
- Semicolons, parentheses, commas, operators
- Conventional to capitalize main functions
- Clauses
- Ending with semicolon
- Standard to use aliases in queries

# Queries

**SELECT**  
**FROM**  
**WHERE**  
**GROUP BY**  
**HAVING**  
**ORDER BY**

**SELECT**  
**FROM**  
**WHERE**  
**GROUP BY**  
**HAVING**  
**ORDER BY**

**Order of processing:**

**FROM**  
**WHERE**  
**GROUP BY**  
**HAVING**  
**ORDER BY**

1. FROM

Chooses table to get the base data records.



2. JOIN

Obtains matching data records from other tables.



3. WHERE

Filters the base data



4. GROUP BY

Aggregates the base data.



5. HAVING

Filters the aggregated data.



6. SELECT

Returns the final data.



7. ORDER BY

Sorts the final data.



## Order of processing:

**FROM:** **table** from which to extract the base data records

**WHERE:** **filters** base data

**GROUP BY:** **aggregates** base data

**HAVING:** **filters** aggregated base data

**SELECT:** **returns** final data

**ORDER BY:** **sorts** final data

**SELECT** variable\_or\_column  
**FROM** table\_name  
**WHERE** filter  
**GROUP BY** aggregate\_by  
**HAVING** filter\_on\_aggregated\_vlaue  
**ORDER BY** arrange\_by

**WITH cte\_name AS (SELECT...)**

**SELECT**

**FROM (SELECT...)**

**WHERE**

**GROUP BY**

**HAVING**

**ORDER BY**

**JOIN/UNION**

**SELECT...**

```
SELECT column1,  
       AGG(column2) AS alias_name, column1  
FROM table_name1 AS t1  
WHERE WHERE_condition  
GROUP BY column1  
HAVING HAVING_condition  
ORDER BY AGG(column2) DESC
```

```
JOIN table_name2 AS t2  
ON t1.column2 = t2.column3;
```

# Tools

Asterisks (\*) for ALL

**AS**

DISTINCT (SELECT DISTINCT)

UNIQUE (primary keys)

RANK

LEAD

LAG

NOT LIKE “”

RANK()

DENSE\_RANK()

ROW\_NUMBER()

OVER()

PARTITION\_BY

TOP

LIMIT

AUTONUMBER

AUTO\_INCREMENT

**Examples**

**SELECT** patient, weight, height

**FROM** hospital;

patient	weight	height
Patient 1	100	62
Patient 2	120	70
Patient 3	90	51
Patient 4	80	59
Patient 5	200	61
Patient 6	170	73

**SELECT** patient, weight, height

**FROM** hospital

**WHERE** weight > 100;

patient	weight	height
Patient 2	120	70
Patient 5	200	61
Patient 6	170	73

```
SELECT facility, provider, AVG(weight) AS avg_weight  
FROM hospital  
GROUP BY facility, provider;
```



<b>facility</b>	<b>provider</b>	<b>weight</b>
Facility A	Provider X	150
Facility B	Provider Y	120
Facility C	Provider X	90
Facility B	Provider Z	125

```
SELECT patient, weight, height  
FROM hospital  
ORDER BY weight  
LIMIT 3;
```

```
SELECT COUNT(facility) AS facility_count  
FROM hospital;
```

```
SELECT COUNT(*) AS doctor_count  
FROM doctors  
WHERE doc_last_name = 'Miller';
```

```
SELECT COUNT(facility) AS facility_count  
FROM hospital;
```



facility_count
6

```
SELECT provider, COUNT(patient) AS patient_count
FROM hospital
GROUP BY provider
HAVING patient_count >= 2;
```

```
SELECT customer_name
FROM customers
WHERE customer_name LIKE "%Ltd";
```

```
SELECT doctor_id, COUNT(*) AS record_count
FROM medical_records
GROUP BY doctor_id
ORDER BY record_count DESC;
```

# Joins

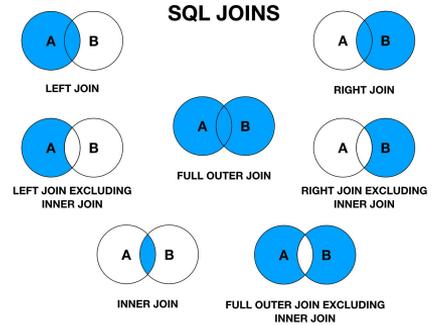
SELECT \*

FROM

LEFT(/RIGHT/CENTER/INNER/OUTER/FULL OUTER) JOIN

```
SELECT a.*, b.drug_name
FROM table1 AS a
LEFT JOIN table2 AS b
ON a.drug_key = b.drug_key
AND a.dtype_key = b.dtype_key;
```

```
SELECT a.name, c.drug_type, b.drug_name
FROM table1 AS a
LEFT JOIN table2 AS b
ON a.drug_key = b.drug_key
LEFT JOIN table3 AS c
ON b.dtype_key = c.dtype_key;
```



EmpDetails		
ID	Name	Salary
1	John	40000
2	Alex	25000
3	Simon	43000



MaritalStatus		
ID	Name	Status
1	John	Married
3	Simon	Married
4	Stella	Unmarried

ID	Name	Salary	Status
1	John	40000	Married
2	Alex	25000	NULL
3	Simon	43000	Married

Left Join

ID	Name	Salary	Status
1	John	40000	Married
2	Alex	25000	NULL
3	Simon	43000	Married
NULL	NULL	NULL	Unmarried

Full Join

ID	Name	Salary	Status
1	John	40000	Married
3	Simon	43000	Married
NULL	NULL	NULL	Unmarried

Right Join

# Conditional Logic

```
SELECT patient, weight  
FROM hospital  
WHERE provider = 'Provider X' OR provider = 'Provider Y';
```

```
SELECT patient, weight  
FROM hospital  
WHERE provider = 'Provider X' AND weight > 90;
```

```
SELECT *  
FROM doctors  
WHERE doctor_id IN (1, 2, 3);
```

**CASE** statements ->

# CASE (new variable)

```
CASE
  WHEN condition1 THEN result1
  WHEN condition2 THEN result2
  WHEN conditionN THEN resultN
  ELSE result
END AS variable_name
```

```
SELECT patient, weight,
CASE
  WHEN height > 70 THEN "Tall"
  WHEN height > 65 THEN "Average"
  WHEN height <= 65 THEN "Short"
  ELSE "N/A"
END AS height_cat
FROM hospital;
```

student_id	category	points
A	quizzes	96
A	participation	100
A	assignments	92
A	final_exam	96
B	quizzes	87
B	participation	98
B	assignments	89
B	final_exam	93
C	quizzes	91
C	participation	96
C	assignments	99
C	final_exam	90

- Using SUM, a CASE statement, and GROUP BY we can create a new variable called quizzes which contains the point value for each student

```
SELECT student_id,  
SUM(CASE  
    WHEN category = "quizzes" THEN points  
    ELSE 0  
    END  
    ) AS quizzes  
FROM class  
GROUP BY student_id;
```

student_id	quizzes
A	96
B	87
C	91

# UNION (Append)

- UNION ALL allows duplicates



table1

a	b	c
1	0	1
2	0	2
1	1	4



a	b	c
1	0	1
2	0	2
1	1	4
2	1	5

SELECT \*  
FROM table1

**UNION**

SELECT \*  
FROM table2;

```
SELECT student_id, 'quizzes' AS category, quizzes AS points
FROM class_wide
```

C	91	96	99	90
---	----	----	----	----

UNION ALL

```
SELECT student_id, 'participation' AS category, participation AS points
FROM class_wide
```

UNION ALL

```
SELECT student_id, 'assignments' AS category, assignments AS points
FROM class_wide
```

UNION ALL

```
SELECT student_id, 'final_exam' AS category, final_exam AS points
FROM class_wide;
```

student_id	category	points
A	quizzes	96
B	quizzes	87
C	quizzes	91
A	participation	100
B	participation	98
C	participation	96
A	assignments	92
B	assignments	89
C	assignments	99
A	final_exam	96
B	final_exam	93
C	final_exam	90

# **Aggregate Functions**

## **Window Functions**

# Window Functions

- Calculation across a set of rows
- Repeat values

**Important:** We are NOT grouping by any column in this example



```
SELECT facility, patient, weight, AVG(weight) OVER() AS avg_weight  
FROM Hospital;
```

facility	patient	weight	avg_weight
Facility A	Patient 1	100	126.7
Facility B	Patient 2	120	126.7
Facility C	Patient 3	90	126.7
Facility B	Patient 4	80	126.7
Facility A	Patient 5	200	126.7
Facility B	Patient 6	170	126.7

# Aggregate Function

- Groups outputs
- Reduces redundancies

```
SELECT column_names,  
       Aggregate_Function  
       OVER(PARTITION BY column_names) AS alias  
FROM table_name;
```

**Important:** We ARE grouping by a column in this example

```
SELECT facility, patient, weight,  
       AVG(weight) OVER(PARTITION BY facility) AS avg_fac_weight  
FROM Hospital;
```



facility	patient	weight	avg_fac_weight
Facility A	Patient 1	100	150
Facility B	Patient 2	120	123.3
Facility C	Patient 3	90	90
Facility B	Patient 4	80	123.3
Facility A	Patient 5	200	150
Facility B	Patient 6	170	123.3

**Temporary Tables**

**Derived Tables\*\***

**Subqueries**

**CTEs**

**Views**

# CTE

```
WITH cte_name AS  
    (SELECT column_names  
     FROM table_name  
     GROUP BY column_names)
```

```
SELECT column_names  
FROM cte_name;
```

# Temporary Table

```
CREATE TEMPORARY TABLE avg_wt AS  
  (SELECT facility, AVG(weight) AS avg_weight  
   FROM Hospital  
   GROUP BY facility);
```

```
SELECT h.facility, h.patient, h.weight, a.avg_weight  
FROM Hospital AS h  
LEFT JOIN Avg_wt AS a  
  ON h.facility = a.facility;
```

## Derived Table

```
SELECT column_names
```

```
FROM (SELECT column_names
```

```
FROM table_name
```

```
GROUP BY column_names) AS derived
```

```
WHERE condition;
```

# Subquery

```
SELECT patient, weight  
FROM hospital  
WHERE weight > (SELECT weight  
                FROM hospital  
                WHERE patient = 'patient 2');
```

```
SELECT patient, weight  
FROM hospital  
WHERE weight IN (SELECT MIN(weight)  
                FROM hospital  
                GROUP BY facility);
```

# View

```
CREATE VIEW avg_wt AS  
  (SELECT facility, AVG(weight) AS avg_weight  
   FROM Hospital  
   GROUP BY facility);
```

```
SELECT h.facility, h.patient, h.weight, a.avg_weight  
FROM Hospital AS h  
LEFT JOIN avg_wt AS a  
  ON h.facility = a.facility;
```

**More Advanced  
& Msc.**

# Database Normalization

## Goals:

- Efficiency
- Scalability
- Minimize runtime
- Maximize disk space
- Maintain referential integrity

*“Most databases are normalized to the third form” - Ifra Fayyaz*

## Third normal form:

- Columns cannot be computed based on other columns
  - **Second normal form:**
    - Primary key dependency
  - **First normal form:**
    - No repeating rows
    - No redundant attributes
    - No comma-separated lists
-

# Pivoting Data

## Wide to long:

```
SELECT student_id, 'quizzes' AS category,  
quizzes AS points  
FROM class_wide;
```

### **UNION ALL**

```
SELECT student_id, 'participation' AS  
category, participation AS points  
FROM class_wide,
```

## Long to wide:

```
SELECT student_id, category, points  
FROM class  
WHERE category = "quizzes";
```

```
SELECT student_id,  
SUM(CASE  
    WHEN category = "quizzes" THEN points  
    ELSE 0  
    END  
) AS quizzes  
FROM class  
GROUP BY student_id;
```

Or JOIN

# Referential Integrity

**Orphaned record:** lost row in parent table

Declarative referential integrity (DRI):

UPDATE CASCADE: Updates primary table automatically

DELETE CASCADE

SET NULL

SET DEFAULT

NO ACTION

# Variable Constraints

```
CREATE TABLE invoices
(
    invoice_id INT NOT NULL UNIQUE,
    vendor_id INT NOT NULL,
    invoice_number VARCHAR(50) NOT NULL,
    invoice_date DATE,
    invoice_total DECIMAL(9,2) NOT NULL,
    payment_total DECIMAL(9,2) DEFAULT 0
);
```

```
CREATE TABLE table_name
(
    column_name_1 data_type [column_attributes],
    column_name_2 data_type [column_attributes],
    ....
    CONSTRAINT name_constraint UNIQUE (column_name_1, column_name_2 ..)
);
```

# Triggers

```
DELIMITER //
CREATE TRIGGER trigger_name
    [BEFORE | AFTER]
    [INSERT | UPDATE | DELETE]
    ON table_name
    FOR EACH ROW
BEGIN
    trigger body
END; //
DELIMITER ;
```

```
DELIMITER//
CREATE TRIGGER payment_check
    BEFORE INSERT ON payment
    FOR EACH ROW
BEGIN
    IF NEW.amount < 0 THEN
        SET NEW.amount = 0;
    ELSEIF NEW.amount > 11.99 THEN
        SIGNALS SQL STATE 'HY000'
        SET MESSAGE_TEXT =
        'Invalid entry: amount must
        not exceed $11.99';
    END IF;
END; //
DELIMITER;
```

**Msc.**

# Indexing

- Automatic for primary keys, foreign keys, and UNIQUE constraints
- For variables that do not change often, to quicken runtime
- Caution: over-indexing will actually slow runtime

```
CREATE [UNIQUE] INDEX index_name  
ON table_name(column_name);
```

# Images

Relatively large data type

“Might not want to keep in tables that will be referenced often”

	Studentname	Subject	Marks	Rank
1	Isabella	Maths	70	1
2	Isabella	Science	70	1
3	Isabella	english	90	2
4	Lily	Maths	65	1
5	Lily	english	70	2
6	Lily	Science	80	3
7	Olivia	Maths	55	1
8	Olivia	Science	60	2
9	Olivia	english	89	3

RANK()

DENSE\_RANK()

	Studentname	Subject	Marks	Rank
1	Isabella	Maths	70	1
2	Isabella	Science	70	1
3	Isabella	english	90	3
4	Lily	Maths	65	1
5	Lily	english	70	2
6	Lily	Science	80	3
7	Olivia	Maths	55	1
8	Olivia	Science	60	2
9	Olivia	english	89	3

# Lead/Lag

Table Name: visits

patient	visit_no	weight
Patient 1	1	100
Patient 1	2	120
Patient 1	3	90
Patient 2	3	170
Patient 2	1	180
Patient 2	2	200



```
SELECT *,  
       LAG(weight)  
       OVER(PARTITION BY patient  
            ORDER BY visit_no) AS wt_before  
FROM visits;
```

patient	visit_no	weight	wt_before
Patient 1	1	100	<i>NULL</i>
Patient 1	2	120	100
Patient 1	3	90	120
Patient 2	1	180	<i>NULL</i>
Patient 2	2	200	180
Patient 2	3	170	200

Table Name: Visits

patient	visit_no	weight
Patient 1	1	100
Patient 1	2	120
Patient 1	3	90
Patient 2	3	170
Patient 2	1	180
Patient 2	2	200



```
SELECT *,  
       LEAD(weight, 2)  
       OVER(PARTITION BY patient  
            ORDER BY visit_no) AS wt_2_after  
FROM visits;
```

patient	visit_no	weight	wt_2_after
Patient 1	1	100	90
Patient 1	2	120	NULL
Patient 1	3	90	NULL
Patient 2	1	180	170
Patient 2	2	200	NULL
Patient 2	3	170	NULL

**Conclusion**

# Altering Table

```
ALTER TABLE vendors  
MODIFY COLUMN vendor_name VARCHAR(50) NOT NULL;
```

```
ALTER TABLE vendors  
ALTER vendor_name SET DEFAULT 'Starbucks';
```

```
ALTER TABLE vendors  
ALTER vendor_name DROP DEFAULT;
```

```
ALTER TABLE vendors  
AUTO_INCREMENT = 100;
```

**EXPLAIN ANALYZE** for monitoring performance (runtimes)

**SHOW WARNINGS**

**ALTER TABLE ADD CONSTRAINT**

**DROP TABLE** temp\_table;

**DROP VIEW** view\_name;

**DROP DATABASE [IF EXISTS]** db\_name;

# Resources

Library resources:

<https://web.p.ebscohost.com/ehost/ebookviewer/ebook/ZTAyNXhuYV9fMzI2Nzc5MI9fQU41?sid=9b015d9f-cbeb-459c-b33c-c1c1cdb6a229@redis&vid=0&format=EB&rid=1>

MySQL Shorts:

<https://www.youtube.com/playlist?list=PLWx5a9Tn2EvG4C90YFJ9eU61lpALeE0SN>

LinkedIn Learning

ChatGTP is an okay editor

Email: [ofs2111@cumc.columbia.edu](mailto:ofs2111@cumc.columbia.edu)

**Questions?**

# Other Examples

facility	patient	weight	avg_fac_weight
Facility A	Patient 1	100	150
Facility B	Patient 2	120	123.3
Facility C	Patient 3	90	90
Facility B	Patient 4	80	123.3
Facility A	Patient 5	200	150
Facility B	Patient 6	170	123.3



```
WITH avg_wt AS  
  (SELECT facility,  
           AVG(weight) AS avg_fac_weight  
  FROM Hospital  
  GROUP BY facility)
```

```
SELECT h.facility, h.patient, h.weight, a.avg_fac_weight  
FROM Hospital AS h  
LEFT JOIN avg_wt AS a  
  ON h.facility = a.facility;
```

**Table Name: Hospital**

Facility	Provider	Patient	Weight	Height	row_num
Facility A	Provider X	Patient 1	100	61	1
Facility B	Provider Y	Patient 2	120	70	2
Facility C	Provider X	Patient 3	90	51	3
Facility B	Provider Z	Patient 4	80	59	4
Facility A	Provider X	Patient 5	200	61	5
Facility B	Provider Z	Patient 6	170	73	6

```
SELECT *,  
       ROW_NUMBER()  
       OVER() AS row_num  
FROM Hospital;
```

# Other Terminology

**One-to-one relationship:** one record links to many rele

**One-to-many relationship:** one record in

**Parent table:** referenced table

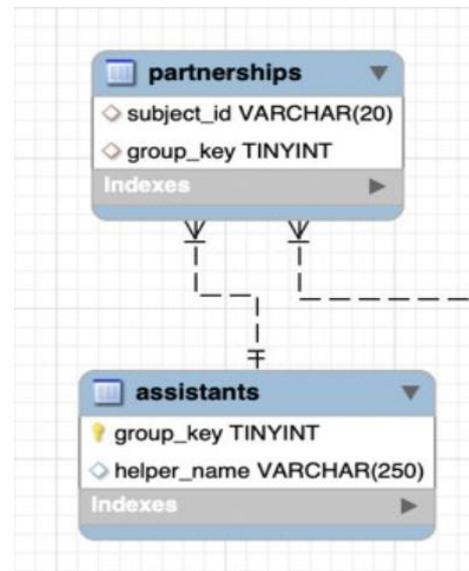
**Child table:** contains the foreign key that links to the child table

**Lookup table:** look up values by key

**Fact table:** stores mainly numeric values, highly optimized

**Dimension table:** details dimension table with longer labels

Linking/connecting/associate/intermediary table: temporary data to be referenced



**One** group via group key is linked to **many** partnerships

