Power Learn Project

Essential Data Retrieval & Filtering with SQL

Power Learn Project

Database Design

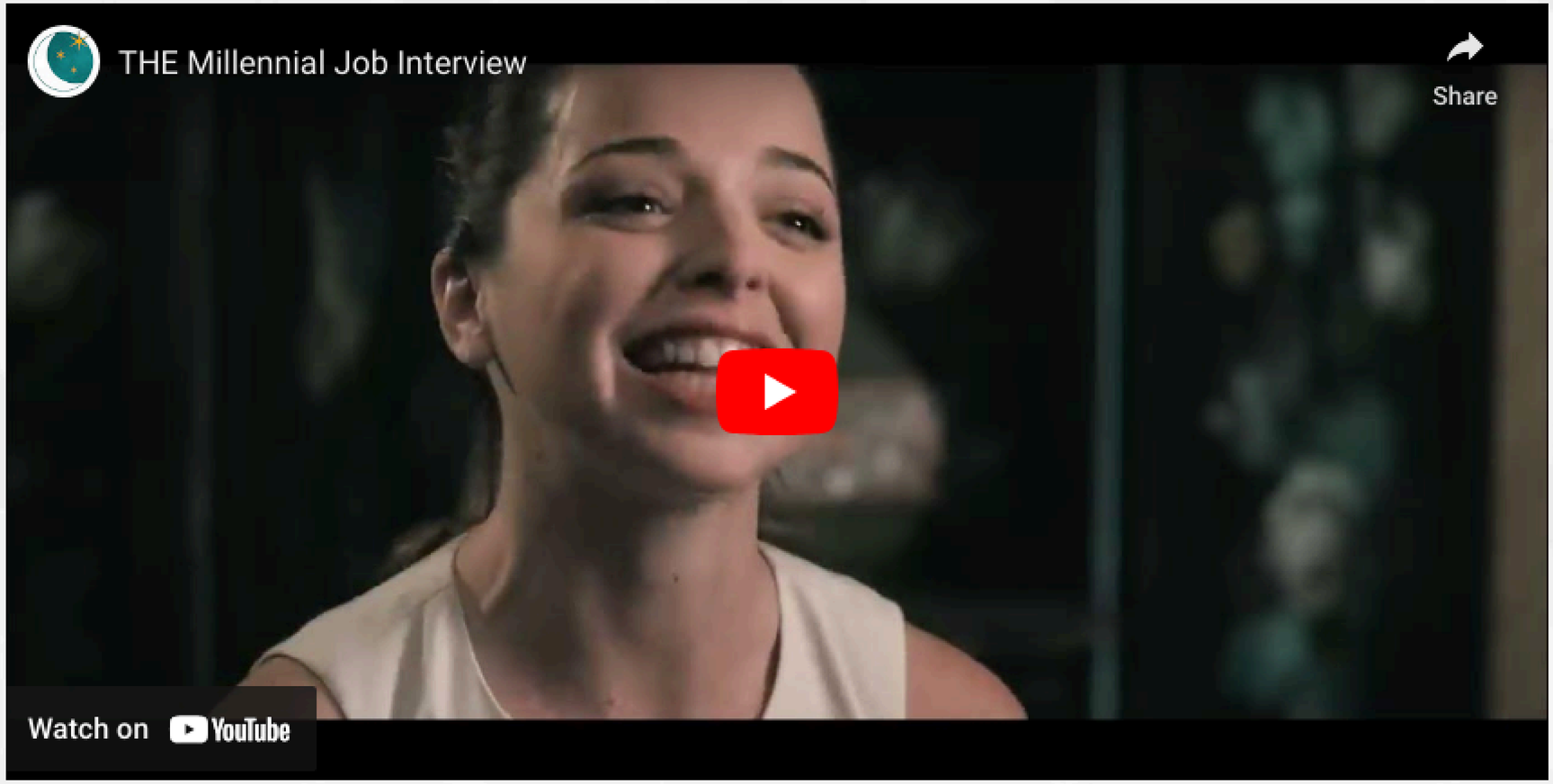
You are live with Gerald Macherechedze

Week 2

Learning Objectives

- **Exploring the SELECT Statement:** Learn to retrieve specific data from database tables.
- **Mastering Filtering Techniques:** Use wildcards, comparison operators, and the WHERE clause to pinpoint data.
- **Sorting Data:** Organize and analyze data using the ORDER BY clause.

Ice-Breaker



Database Design

You are live with Gerald Macherechedze

To Play

Crow's Foot Notation

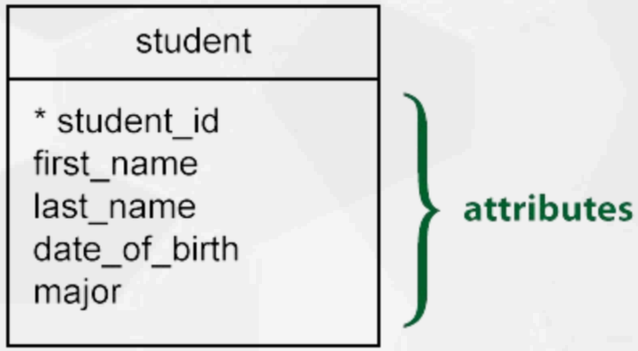
Power Learn Project

Mapping Database Relationships:
Crow's Foot notation is a visual language used to depict relationships between tables in relational databases.

Understanding Connections:
It helps visualize how data flows and interacts across different tables, making it easier to understand the overall database structure.

An entity is represented by a rectangle, with its name on the top. The name is singular (entity) rather than plural (entities).

The attribute(s) that uniquely distinguishes an instance of the entity is the identifier. Usually, this type of attribute is marked with an asterisk.



Tables and Relationships

Power Learn Project

In relational databases, tables can be linked together based on shared columns, allowing you to combine data from different sources for a more comprehensive view.

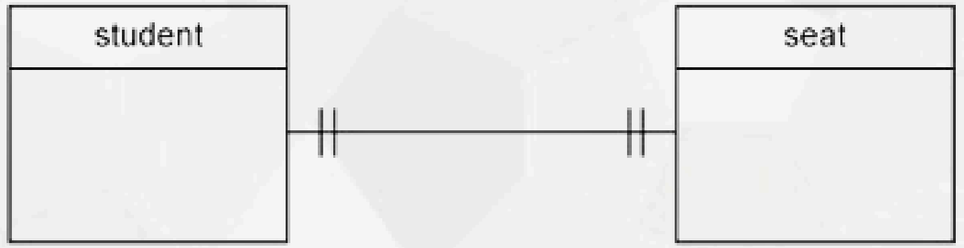
Types of Relationships:

- One-to-One (1:1): A single record in one table links to exactly one record in another (uncommon).
- One-to-Many (1:M): The most common type. One record in a table (the "one") links to many records in another table (the "many"). (e.g., Customers & Orders)
- Many-to-Many (M:M): Many records in one table connect to many records in another, often using a junction table. (e.g., Students & Courses with Enrollments table)

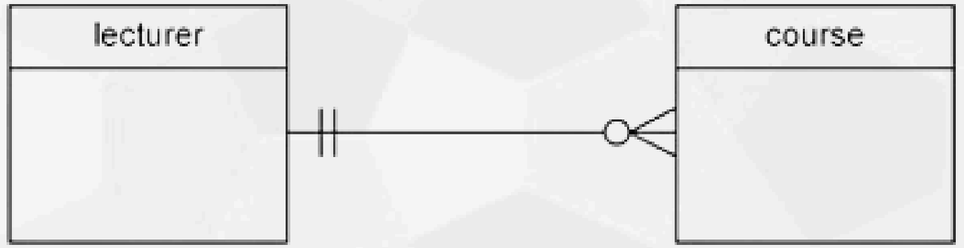
Mandatory vs. Optional: Solid lines represent mandatory relationships, while dashed lines indicate optional relationships.

Crow's Foot Notation

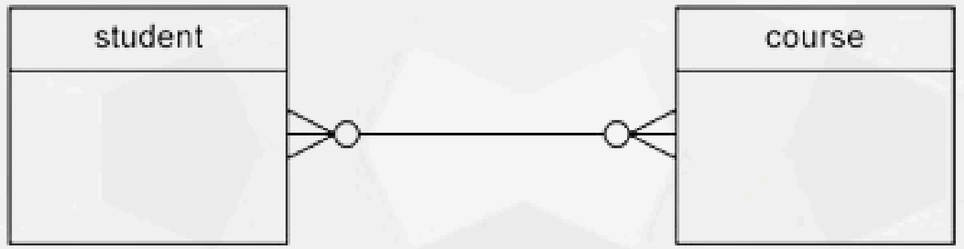
- One-to-one



- One-to-many



- Many-to-many



Enforcing Data Integrity

Power Learn Project

Garbage In

Garbage Out

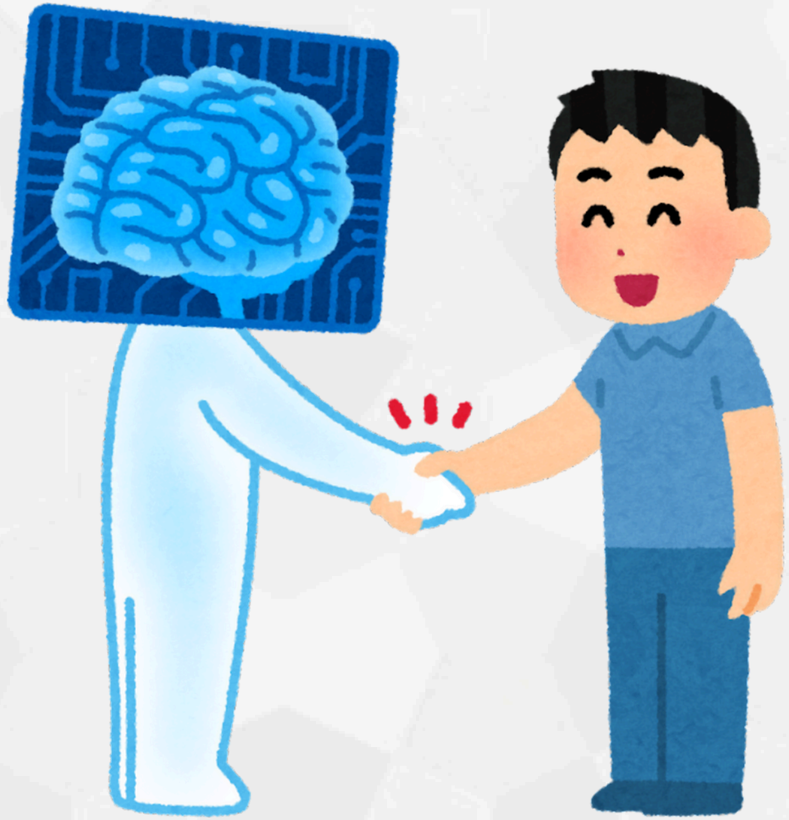
Enforcing Data Integrity with Constraints

Power Learn Project

Data integrity ensures the accuracy and consistency of your data in a database. Constraints are like rules that enforce these standards.

Main Constraint Types

- Primary Key (PK)
- Foreign Key (FK)
- Unique
- NOT NULL
- AUTO_INCREMENT



Enforcing Data Integrity with Constraints

Power Learn Project

Primary Key (PK):

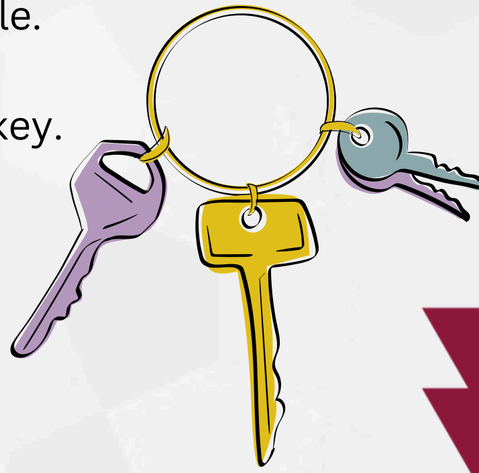
- Uniquely identifies each row in a table (like a social security number).
- Only one record can have the same primary key value.
- Usually enforced on a single column, but can involve multiple columns for composite primary keys.

Foreign Key (FK):

- Establishes a link between two tables based on a shared column.
- The foreign key value must exist in the referenced table's primary key or a unique key.
- Ensures data consistency by preventing "orphaned" records (referencing non-existent data).

Unique:

- Guarantees that specific values within a column (or set of columns) are unique across the entire table.
- Can be applied to multiple columns for a more comprehensive check.
- Differs from primary key in that a table can have multiple unique constraints, but only one primary key.



Enforcing Data Integrity with Constraints

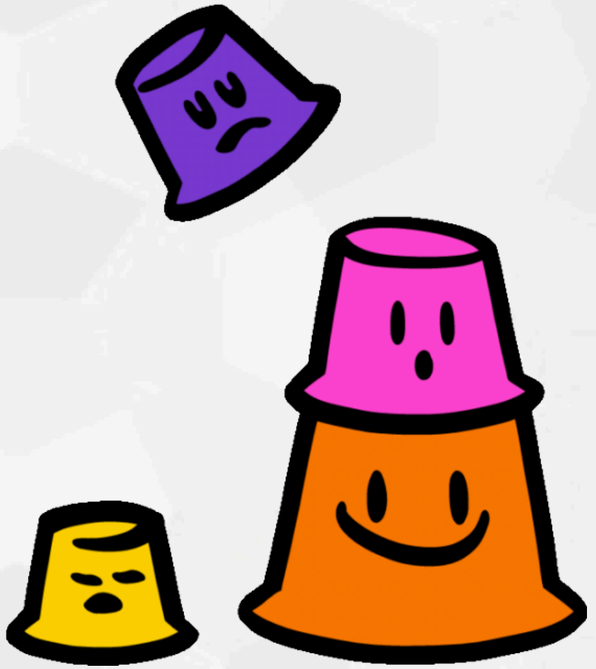
Power Learn Project

NOT NULL:

- Prevents null values (empty or missing data) from being inserted into a column.
- Ensures data completeness for critical columns.

AUTO_INCREMENT:

- Often used with primary keys that are numeric values.
- Automatically generates a unique, sequential number for each new row inserted.
- Saves you from manually assigning IDs and simplifies record identification.



Code Formatting Best Practices

Writing Clean & Readable Queries Includes:

- **Consistent Indentation:** Indentation helps visually organize your code, making it easier to understand complex queries.

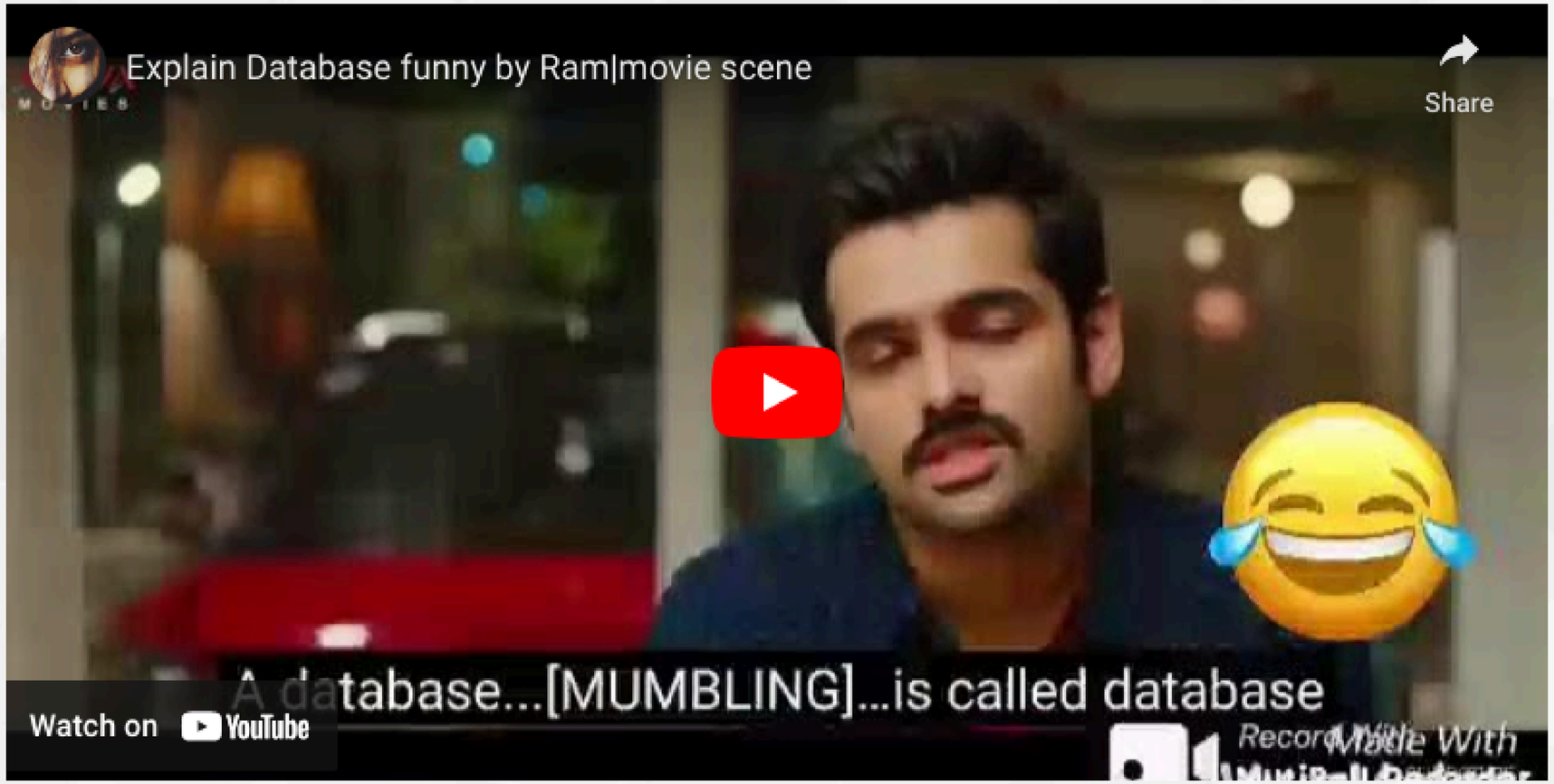
```
SELECT  -- Capitalized keyword
    Employee_Name,  -- Column names
    Department
FROM Employees;  -- Capitalized keyword

```

- **Capitalization:** Capitalize all SQL keywords (SELECT, FROM, WHERE, etc.) for better readability and consistency.

- **Inline Comments:** Use comments (--) to explain complex queries or specific parts of your code. This improves understanding for yourself and others.

Recap...



To Play

Dropping Tables and Databases

This slide covers two methods for removing database objects:

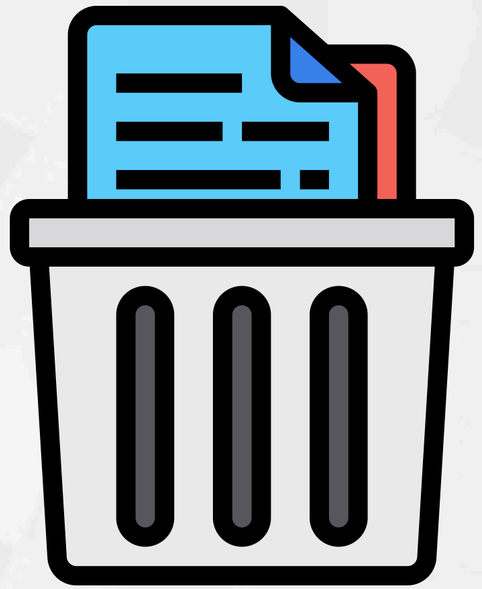
- DROP TABLE: Allows you to remove a table from the database.
- DROP DATABASE: Allows you to remove an entire database from the system.

Use with Caution! Dropping a table or database permanently deletes all the data stored within it.

Example:

```
DROP TABLE appointments;  -- Drops the 'appointments' table (if it exists)

DROP DATABASE telemedicine;  -- Drops the 'telemedicine' database (if it exists)
```



Unveiling the SELECT Statement

Power Learn Project

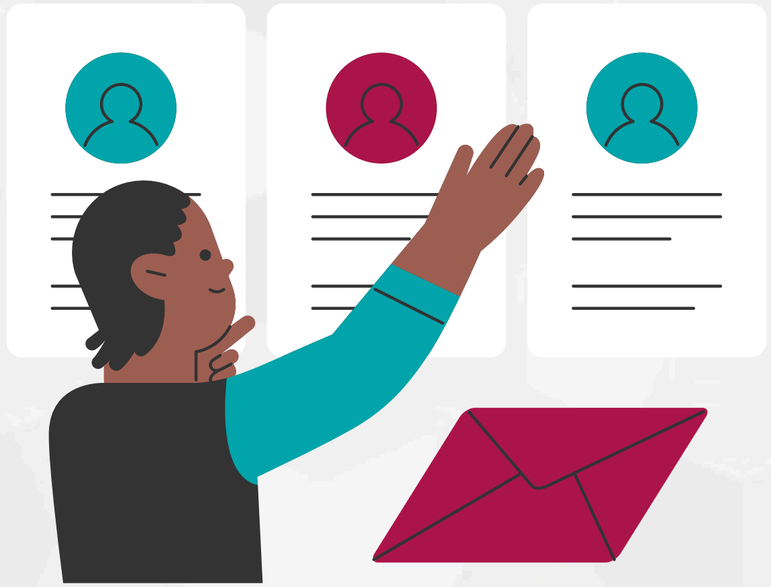
The SELECT statement is our key tool for retrieving data from database tables.

Specifying What to Retrieve:

We use SELECT to specify which columns (or all columns with *) we want to retrieve from the table.

From Where?
The FROM clause specifies the table name where the data resides.

```
SELECT first_name, last_name
FROM patients;  -- Retrieve all patient names
```



Power
Learn
Project

Hands On

Time to get busy!!!

Power
Learn
Project

Database Design

You are live with Gerald Macherechedze

Mastering Data Filtering with WHERE

Power Learn Project

The WHERE clause allows us to filter retrieved data based on specific conditions.

Comparison Operators:
We use comparison operators (e.g., =, >, <, LIKE) to define the filtering criteria.

Wildcards for Flexible Matching:
Wildcards are special characters that substitute for one or more characters in the search criteria, allowing for broader matches in your filtering.

```
-- A. Retrieve all patients from 1970
SELECT *
FROM patients
WHERE date_of_birth  > ‘1970-01-01’;

-- B. Filter by gender
SELECT *
FROM patients
WHERE gender = 'female';
```

```
-- C. Filter using wildcard characters (%) for partial matches
SELECT *
FROM patients
WHERE first_name LIKE 'Ab%'; -- Matches names starting with Ab

-- D. Wildcard at the beginning (%)
SELECT *
FROM patients
WHERE first_name LIKE '%se'; -- Matches names ending with se
```

Power Learn Project

Database Design

You are live with Gerald Macherechedze

Sorting Retrieved Data with ORDER BY

The ORDER BY clause allows you to sort retrieved data in a specific order, making it easier to analyze.

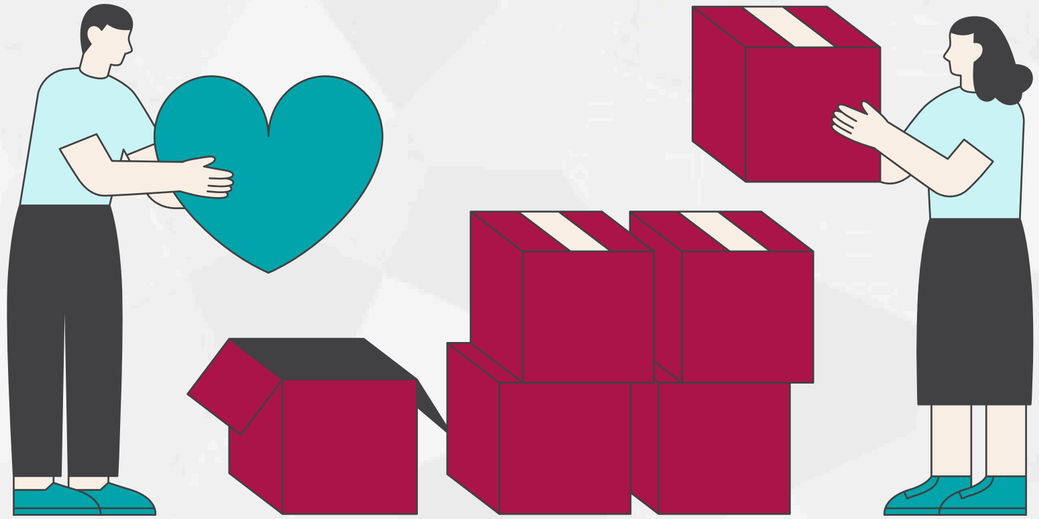
Ascending or Descending:

You can sort data in ascending (A-Z, lowest to highest numbers) or descending (Z-A, highest to lowest numbers) order.

```
-- Sort all providers by firstname (ascending)
SELECT *
FROM providers
ORDER BY first_name;

-- Sort all providers by firstname (descending)
SELECT *
FROM providers
ORDER BY first_name DESC;

-- Sort by multiple columns (primary vs secondary sorting)
SELECT *
FROM providers
ORDER BY provider_specialty,  first_name DESC;  -- Sort by specialty first (ascending), then by first_name (descending)
```



Example Scenario

Let's imagine you want to analyze :

```
-- Retrieve all visits for February 2019
SELECT *
FROM visits
WHERE date >= '2019-02-01' AND Date <= '2019-02-29';  -- Adjust date range as needed

-- Retrieves all the patients treated by the doctor with id 1 and blood pressure is more than 150 and orders them by date
SELECT patient_id, blood_pressure_systolic
FROM visits
WHERE provider_id = 1
AND blood_pressure_systolic > 150
ORDER BY date_of_visit ASC;
```

Summary

Power Learn Project

Recap: SQL Retrieval & Wrangling

CREATE Databases and Tables:
Formulate a database from scratch.

SELECT the Right Data:
Mastered retrieving data using SELECT statements.

Filter Like a Pro:
Learned to filter data with WHERE and wildcards for pinpoint accuracy.

Sort It Out:
Used ORDER BY to organize results for easy analysis.

Power Learn Project

Challenge

Time to get busy!!!

Power Learn Project

Database Design

You are live with Allan Njoroge