**Session 7 SQL – Tasks**

* **Task 1 : List other data types**

So the data types in MySQL divided into 3 categories: Strings, Numeric and Data and time data types.

I will start with String data types but let me tell you something some data types I won’t mention them because they are the same data types I will mention with the difference that they are have a bigger size in byte or bit depending on they are binary data type or not so put this in your head and let’s start:

String:

CHAR(), VARCHAR(), BINARY(), VARBINARY(), TEXT(), BLOB(), ENUM(), SET()

Every type have a TINY and MEDIUM and LONG with fixed num of byte or bits or characters however ENUM and SET don’t have.

Numeric:

BIT(), INT(), INTGER(), BOOL, BOOLEAN, FLOAT(), DOUBLE(), DECIMAL(), DOUBLE PRECISION()

DATE and TIME:

DATE, TIMESTAMP(), DATETIME(), TIME(), YEAR

* **Task 2 : List other constraints**

CHECK and CREATE INDEX

* **Task 3 : List other operators**

MySQL Arithmetic Operators: like +,-,\*,/ and %

MySQL Bitwise Operators like: &,| and ^

MySQL Logical Operators like: AND, OR, ALL, ANY, NOT.

* **Task 4 : What is AS and example**

The AS command is used to rename a column or table with an alias.

Example:

SELECT CustomerID AS ID, CustomerName AS Customer

FROM Customers;

An alias only exists for the duration of the query.

* **Task 5 : What is the wildcard in SQL and example**

A wildcard character is used to substitute one or more characters in a string.

Wildcard characters are used with the LIKE operator. The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.

Example:  
WHERE CustomerName LIKE 'a%'; will result in Finds any values that starts with "a".

* **Task 6 : What is a nested query**

A nested query, also known as a subquery, in SQL is essentially a query within another query. It allows you to use the result of an inner SELECT statement (the subquery) in the outer SELECT statement.

Pretty much SELECT inside another SELECT.

* **Task 7 : What are the relationships in a database**

Relationships in a database are the crucial links between different tables that organize and connect data for efficient retrieval and manipulation. These connections act like bridges, allowing you to navigate across various pieces of information within your database.

There are three main types of relationships in a database:

1. One-to-One (1:1) Relationship:

2. One-to-Many (1:M) Relationship:

3. Many-to-Many (M:M) Relationship:

This relationship allows for multiple records in both tables to be associated with each other. Consider a database for movies and actors. Each movie record can have multiple actors (the "many" to "many" connection), and each actor can be part of multiple movies.

To properly establish relationships, databases use keys:

Primary Key: A unique identifier within a table. In the book loan example, the loan ID would be the primary key in the loan table.

Foreign Key: A column in one table that references the primary key of another table, establishing the linkage. The book ID in the loan table would be a foreign key referencing the primary key (book ID) in the book table.

* **Task 8 : Create a school database and run some queries**

Done in GitHub repo files.

* **Task 9 : What is INTERSECT in SQL and example**

The INTERSECT operator function job is to identifying and returning only the rows that are present in both of two specified result sets. Here's how it works:

You have 2 tables it will try and find any thing that is common in the 2 tables and return the result but the result will be just the query you used in the code not all row.

* **Task 10 : The idea of triggers in SQL**

In SQL, triggers are special database objects that automatically execute a set of SQL statements when a specific event occurs on a table. They act as "event-driven" procedures, ensuring data integrity, automating actions, and enforcing business rules within the database.

* **Task 11 : Search More About Database Normalization**

Database normalization is a fundamental process in database design that aims to organize data in a structured and consistent manner. It involves a series of rules, called normal forms, that guide the structuring of tables and relationships to reduce redundancy and improve data integrity.

Here's a breakdown of the key concepts and benefits of normalization:

1. Normal Forms:

First Normal Form (1NF): Eliminates repeating groups within a table by ensuring each row has a unique primary key and each column holds atomic values (single, indivisible values).

Second Normal Form (2NF): Ensures that all non-key columns are fully dependent on the primary key, eliminating partial dependencies.

Third Normal Form (3NF): Removes transitive dependencies, where a non-key column depends on another non-key column, making all columns directly dependent on the primary key.

There are more forms however they are very specific for specific usage.

2. Benefits of Normalization:

Reduced Data Redundancy, Improved Data Integrity, Easier Data Updates, Enhanced Data Retrieval and Simplified Data Maintenance.