



Trainer & Career Counsellor

Saiful Hoda, Exeter MBA, B. Tech. in Computer Science

Ex. Citibank, Refinitiv, Virtu Financials and Nomura Holding (Hong Kong)

Contents of the Program

1. Introduction to Databases & MySQL	4
1.1. What is a Database?	4
1.2. Relational Database Concepts	4
1.3. Introduction to SQL & MySQL	4
1.4. Setting up MySQL Environment.....	4
2. Basic Data Retrieval	5
2.1. SELECT Statement	5
2.2. FROM Clause.....	5
2.3. WHERE Clause.....	5
2.4. ORDER BY Clause (Sorting Data)	5
2.5. LIMIT Clause (Limiting Results).....	5
3. Basic Data Manipulation (DML) & Table Creation (DDL).....	6
3.1. CREATE DATABASE & USE	6
3.2. CREATE TABLE	6
3.3. INSERT INTO	6
3.4. UPDATE	6
3.5. DELETE FROM	6
3.6. DROP TABLE / DATABASE.....	6
4. Data Types, Indexes and Relational Design.....	6
4.1. MySQL Data Types in Detail	6
4.2. Database Normalization (Conceptual)	7
4.3. Indexing	7
5. Joining Multiple Tables	7
5.1. Understanding Joins	7
5.2. INNER JOIN	7
5.3. LEFT (OUTER) JOIN	7
5.4. RIGHT (OUTER) JOIN	7
5.5. FULL (OUTER) JOIN (Conceptual for MySQL)	7
5.6. Self-Join	8
5.7. UNION & UNION ALL	8

6. Querying & Aggregation for Analysis	8
6.1. Aggregate Functions	8
6.2. GROUP BY Clause	8
6.3. HAVING Clause (Filtering Groups)	8
7. Subqueries, CTEs and Views	8
7.1. Subqueries (Nested Queries)	9
7.2. Common Table Expressions (CTEs)	9
7.3. Views	9
8. Advanced analytical functions	9
8.1. Window Functions	9
8.2. Date and Time Functions	9
8.3. String Functions	10
8.4. CASE Statements	10
8.5. Query Optimization Techniques	10
9. Stored Procedures, Functions & Triggers	10
9.1. Stored Procedures	10
9.2. User-Defined Functions (UDFs)	11
9.3. Triggers (Brief Introduction)	11
10. Connecting MySQL with Python	11
10.1. Establish a connection from Python to MySQL	11
11. Practical Application & Projects	11
11.1. Data Modelling for Analytics	11
11.2. Common Data Science Use Cases in SQL	11
12. Capstone Projects (Hands-on)	12
12.1. E-commerce Sales Analysis	12
12.2. User Engagement & Retention:	12
12.3. Build a Simple Dashboard Data Source	12
12.4. Data Cleaning & Transformation Pipeline	12

1. Introduction to Databases & MySQL

○ What is a Database?

- Conceptual understanding
- Data
- Information
- Database Management System (DBMS)
- Relational Database (RDB)
- Tables
- Rows
- Columns.

1.1. Relational Database Concepts

Core principles of relational databases.

- Schema
- Primary Key (PK)
- Foreign Key (FK)
- Relationships (One-to-One, One-to-Many, Many-to-Many).

1.2. Introduction to SQL & MySQL

- DDL (Data Definition Language - `CREATE`, `ALTER`, `DROP`)
- DML (Data Manipulation Language - `INSERT`, `UPDATE`, `DELETE`)
- DQL (Data Query Language - `SELECT`).

1.3. Setting up MySQL Environment

Guide on installing MySQL Server and a client tool.

- MySQL Server
- MySQL Workbench (GUI tool)
- Command Line Client.

2. Basic Data Retrieval

2.1. SELECT Statement

Overview: The most fundamental command for retrieving data.

- SELECT *
- SELECT column1, column2` (specific columns)
- Aliasing columns (`AS`).

2.2. FROM Clause

Specify the table(s) from which to retrieve data.

2.3. WHERE Clause

- Filter rows based on specified conditions.
- Comparison operators (`=`, `!=`, `>`, `<`, `>=`, `<=`)
- Logical operators (`AND`, `OR`, `NOT`)
- LIKE` (pattern matching with `%`, `_`)
- IN (list of values)
- BETWEEN (range)
- IS NULL/IS NOT NULL

2.4. ORDER BY Clause (Sorting Data)

Sorts the result set in ascending or descending order.

- ASC
- DESC
- sorting by multiple columns
- Default behaviour

2.5. LIMIT Clause (Limiting Results)

Restricts the number of rows returned by the query.

3. Basic Data Manipulation (DML) & Table Creation (DDL)

3.1. CREATE DATABASE & USE

Create a new database and select it for use.

3.2. CREATE TABLE

- Defining the structure of a new table.
- Column names
- Data Types (e.g., `INT`, `VARCHAR`, `DATE`, `DECIMAL`)
- Constraints (`PRIMARY KEY`, `NOT NULL`, `UNIQUE`, `DEFAULT`).

3.3. INSERT INTO

- Adding new rows of data into a table.

3.4. UPDATE

- Importance of `WHERE` clause to avoid updating all rows.

3.5. DELETE FROM

- Importance of `WHERE` clause.

3.6. DROP TABLE / DATABASE

- Overview: Deleting entire tables or databases.

4. Data Types, Indexes and Relational Design

4.1. MySQL Data Types in Detail

- A deeper look into choosing the correct data types for various data scenarios

4.2. Database Normalization (Conceptual)

The process of organizing the columns and tables of a relational database to minimize data redundancy and improve data integrity.

- 1NF
- 2NF
- 3NF (brief explanation of goals, not strict rules)
- Denormalization for reporting/performance.

4.3. Indexing

Overview: Creating special lookup tables to speed up data retrieval operations.

5. Joining Multiple Tables

5.1. Understanding Joins

- Conceptual understanding and Best Practices.

5.2. INNER JOIN

- Conceptual understanding, how and when to use.

5.3. LEFT (OUTER) JOIN

- Conceptual understanding, how and when to use.

5.4. RIGHT (OUTER) JOIN

- Conceptual understanding, how and when to use.

5.5. FULL (OUTER) JOIN (Conceptual for MySQL)

- Conceptual understanding, how and when to use.

5.6. Self-Join

Joining a table to itself to compare rows within the same table.

5.7. UNION & UNION ALL

Combining the result sets of two or more `SELECT` statements. Conceptual understanding with examples on the usage.

6. Querying & Aggregation for Analysis

This section builds on the basics, introducing more powerful querying techniques essential for data analysis.

6.1. Aggregate Functions

Functions that perform calculations on a set of rows and return a single value

- COUNT()
- SUM()
- AVG()
- MIN()
- MAX()
- DISTINCT with aggregates (COUNT(DISTINCT column))

6.2. GROUP BY Clause

- Conceptual examples and usage.

6.3. HAVING Clause (Filtering Groups)

- Conceptual examples: HAVING aggregate condition, Difference from `WHERE`.

7. Subqueries, CTEs and Views

7.1. Subqueries (Nested Queries)

- A query nested inside another SQL query. Conceptual examples and when to use.

7.2. Common Table Expressions (CTEs)

- Named temporary result sets.

7.3. Views

- Conceptual Overview
- Points worth noting
- Practical Examples

8. Advanced analytical functions

8.1. Window Functions

Perform calculations across a set of rows related to the current row, without collapsing rows (unlike aggregate functions).

- OVER() clause,
- PARTITION BY
- ORDER BY` within OVER()
- Ranking functions ROW_NUMBER(), RANK(), DENSE_RANK()
- Aggregate window functions SUM(), OVER(), AVG(), OVER()
- Lag/Lead functions.

8.2. Date and Time Functions

Conceptual Overview; when to use and how to use NOW()

- CURDATE()
- YEAR()
- MONTH()
- DAY()

- WEEK()
- DATEDIFF()
- DATE_ADD()
- DATE_SUB()
- STR_TO_DATE()
- DATE_FORMAT()

8.3. String Functions

Conceptual Overview; when to use and how to use

- CONCAT()
- LENGTH()
- SUBSTRING()
- TRIM()
- REPLACE()
- UPPER()
- LOWER()

8.4. CASE Statements

- Conditional logic within SQL queries, when to use.

8.5. Query Optimization Techniques

- Strategies to write efficient queries that run faster on large datasets.

9. Stored Procedures, Functions & Triggers

9.1. Stored Procedures

- Key Concepts: DELIMITER, CREATE PROCEDURE, CALL.
- Control flow statements ('IF', 'CASE', 'LOOP').

9.2. User-Defined Functions (UDFs)

- Create custom functions and learn their usage

9.3. Triggers (Brief Introduction)

- Understand data integrity mechanisms that might be in place in production databases.

10. Connecting MySQL with Python

10.1. Establish a connection from Python to MySQL

Download essential Python libraries and set up a connection to perform SQL operation on a database from Python application

11. Practical Application & Projects

11.1. Data Modelling for Analytics

Overview: Discussing how data is structured for analytical purposes, often differing from transactional (OLTP) systems (e.g., star schema, snowflake schema - briefly).

- Fact tables
- Dimension tables.

11.2. Common Data Science Use Cases in SQL

Overview: Demonstrating how to tackle typical data science problems using SQL.

- Customer Segmentation: Using `GROUP BY`, `COUNT`, `AVG` to identify customer groups.
- Tracking user progression through steps using joins and conditional aggregates.
- Analysing user behaviour over time using date functions and joins.
- Creating new features directly in SQL using `CASE`, string functions, date functions, or window functions.
- Directly applying SQL skills to solve real business problems.

12. Capstone Projects (Hands-on)

12.1. E-commerce Sales Analysis

Analyse customer behaviour, product performance, and sales trends using a simulated e-commerce dataset (e.g., customers, orders, products, order items tables).

12.2. User Engagement & Retention:

Analyze user activity logs to understand engagement, identify active users, and potentially perform cohort analysis.

12.3. Build a Simple Dashboard Data Source

Create a series of views or stored procedures that generate summarized data ready for a dashboard or reporting tool.

12.4. Data Cleaning & Transformation Pipeline

Simulate a common ETL (Extract, Transform, Load) scenario where raw data needs to be cleaned and transformed within MySQL before being loaded into a "clean" table.