**SQL Masterclass: End-to-End Database Project**

**Project Overview**

* Create a **Customer Order Management System** (or another relevant real-world database).
* Cover **database design, querying, data analysis, and optimization**.
* Use **MySQL or SQLite** to implement it.
* Include a final **portfolio project** where students apply their skills.

**Chapter 1: Introduction to Databases & SQL**

1. What is a **database** and how is data stored?
2. What is **SQL**, and why is it important?
3. Installing **MySQL/SQLite** and setting up a database.
4. Introduction to **Relational Database Concepts** (Tables, Rows, Columns).
5. **Hands-on:** Creating a simple database.

**Chapter 2: SQL Stack & Environment Setup**

1. Installing and configuring MySQL or SQLite.
2. Using the **SQL Query Editor** (query pane, result pane, message pane).
3. Basic SQL **commands and syntax**.

**Hands-on Exercise:** Set up a database for an online store.

**Chapter 3: Writing Queries to Extract Data**

1. **SELECT statements** – Retrieving data.
2. **Filtering with WHERE** – Getting specific data.
3. **Sorting & Limiting Results** – ORDER BY & LIMIT.
4. **Aliases** – Creating readable column names.
5. **Best practices** for writing queries.

**Code Challenge:** Retrieve all orders above $100 sorted by date.

**Chapter 4: Data Filtering & Analysis**

1. Using **comparison operators** (=, >, <, >=, <=, !=).
2. **BETWEEN & IN** – Filtering ranges of values.
3. **LIKE** – Searching for patterns in text.
4. **Date filtering** – Filtering data by date ranges.
5. **Logical operators (AND, OR, NOT)** – Combining multiple conditions.
6. **CASE statements** – IF-THEN logic.

**Hands-on Exercise:** Find all orders placed in the last 30 days.

**Chapter 5: Accessing Data from Multiple Tables (Joins & Relationships)**

1. Understanding **table relationships** & entity diagrams.
2. **INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN**.
3. **Joining multiple tables**.
4. Using **self-joins & cross joins**.

**Project Task:** Retrieve customer orders with customer details.

**Chapter 6: SQL Functions & Aggregations**

1. **Mathematical functions** (SUM, AVG, MIN, MAX, COUNT).
2. **String functions** (UPPER, LOWER, CONCAT, SUBSTRING).
3. **Date functions** (NOW, DATE\_FORMAT, DATEDIFF).
4. Nesting functions for advanced querying.

**Challenge:** Find the top 5 best-selling products with total sales.

**Chapter 7: Grouping & Aggregating Data**

1. **GROUP BY** – Summarizing data.
2. **HAVING vs WHERE** – Filtering aggregated data.
3. Grouping by **multiple fields**.

**Task:** Find the total revenue per month for the last 6 months.

**Chapter 8: Subqueries & Nested Queries**

1. Using **subqueries** inside SELECT, WHERE, and FROM clauses.
2. **Aggregated subqueries** – Complex filtering.
3. **IN & EXISTS subqueries** – Checking conditions.

**Exercise:** Retrieve customers who have placed more than 3 orders.

**Chapter 9: Stored Procedures & Views**

1. What are **Views**?
2. Creating, updating, and deleting **views**.
3. Understanding **Stored Procedures**.
4. **Materialized Views** and their advantages.

**Task:** Create a view that displays customer order history.

**Chapter 10: Managing & Modifying Data**

1. **INSERT INTO** – Adding new records.
2. **UPDATE** – Modifying existing data.
3. **DELETE** – Removing records.

**Final Challenge:** Implement a feature to update order status.

**Final Portfolio Project: Build a Full Database System**

Students apply everything they’ve learned by building a **real-world SQL database**:

**Example Project:** *Customer Order Management System*

* Create **tables** for Customers, Orders, Products, Payments.
* Write queries to **retrieve, filter, and analyze data**.
* Use **joins, subqueries, functions, and stored procedures**.
* Optimize queries for **performance**.
* Generate **SQL reports & insights**.