INTERNSHIP PROJECT PHASE (2 WEEKS)

PROJECT GUIDELINES

- **Choose any 1 or 2 projects from the given list.**
- **You are free to improvise — take the given project as a base and modify it as you like.**
- **You can use any tools, technologies, or steps you're comfortable with — there are no restrictions.**
- **Focus and work sincerely so that you have complete clarity and can explain the project confidently in interviews.**
- **Go through the Top 50 Interview Questions for your domain (attached at the end).**
- **Update your project status regularly when the Google Form is shared in group.**
- **while working on the project YOU CAN CHOOSE ANY DATASET RELAVENT TO THE PROJECT.**

.

After project completion, prepare a 1–2 page report in PDF format, containing:

- Introduction
- Abstract
- Tools Used
- Steps Involved in Building the Project
- Conclusion

◆ Note: Report must not exceed 2 pages.

DEAR INTERNS,

YOU HAVE TO UPDATE STATUS OF YOUR PROJECT EVERY 3 OR 4 DAYS ONCE WHEN THE UPDATION LINK IS SHARED IN THE GROUP.

Final submission links will be shared later.

❗ READ ALL THE GUIDELINES CAREFULLY ❗

LIST OF PROJECTS

1. Online Retail Sales Database Design

Objective: Design a normalized SQL schema for an e-commerce platform.

Tools: MySQL / PostgreSQL, dbdiagram.io

Mini Guide:

1. Identify key entities: Products, Customers, Orders, Payments.
2. Design ER diagram using dbdiagram.io.
3. Normalize schema to 3NF.
4. Write DDL scripts to create tables with constraints.
5. Populate tables with sample data.
6. Write JOIN queries and views for sales reports.

Deliverables: ER diagram, SQL schema, sample data, query report.

2. Employee Management and Attendance Tracker

Objective: Build a database to manage employee data and attendance.

Tools: PostgreSQL, pgAdmin

Mini Guide:

1. Create tables for Employees, Departments, Roles, Attendance.
2. Insert dummy data (200+ records).
3. Write queries for monthly attendance and late arrivals.
4. Add triggers for timestamp and status.
5. Create functions to calculate total work hours.
6. Generate reports using GROUP BY and HAVING.

Deliverables:

SQL scripts, functions, triggers, attendance reports.

3. COVID-19 Data Analytics

Objective: Analyze COVID-19 statistics using SQL queries.

Tools: SQLite + DB Browser, Kaggle dataset

Mini Guide:

1. Import dataset into SQLite.
2. Clean and transform data (formatting, nulls).
3. Create necessary tables and constraints.
4. Write analytical queries (top countries, daily trends).
5. Use GROUP BY and window functions for trends.
6. Export result views and summary report.

Deliverables:

SQLite DB, SQL queries, reports, cleaned dataset.

4. Inventory and Warehouse Management System

Objective: Design a SQL backend for warehouse inventory tracking.
Tools: MySQL / DBeaver
Mini Guide:

1. Model schema for Products, Warehouses, Suppliers, Stock.
2. Insert sample inventory records.
3. Create queries to check stock levels and reorder alerts.
4. Write triggers for low-stock notification.
5. Create stored procedure to transfer stock.
6. Document schema and queries.

Deliverables:
SQL schema, triggers, procedures, inventory queries.

5. Library Management System

Objective: Build a digital library database with loan tracking.
Tools: PostgreSQL + pgAdmin
Mini Guide:

1. Define schema: Books, Authors, Members, Loans.
2. Insert test data for each entity.
3. Handle many-to-many relationships with bridge tables.
4. Create views for borrowed and overdue books.
5. Write triggers for due-date notifications.
6. Write reports using aggregation and JOINs.

Deliverables:
ERD, SQL dump, overdue report queries, documentation.

6. Airline Reservation System

Objective: Create a SQL system to manage flights and bookings.
Tools: MySQL Workbench
Mini Guide:

1. Design schema: Flights, Customers, Bookings, Seats.
2. Normalize schema and define constraints.
3. Insert sample flight and booking records.
4. Write queries for available seats, flight search.
5. Add triggers for booking updates and cancellations.
6. Generate booking summary report.

Deliverables:
SQL schema, triggers, queries, flight availability views.

7. Real Estate Listings and Analytics

Objective: Track property listings and generate price insights.

Tools: PostgreSQL, DBeaver

Mini Guide:

1. Model schema: Properties, Agents, Buyers, Transactions.
2. Insert mock property data with prices and dates.
3. Write queries to analyze average prices by region.
4. Create views for high-demand areas.
5. Generate price trend reports using window functions.
6. Export result tables to CSV.

Deliverables:

Database schema, queries, trend reports.

8. SQL ETL Pipeline Simulation

Objective: Simulate ETL (Extract, Transform, Load) using SQL.

Tools: SQLite + DB Browser

Mini Guide:

1. Import raw data (CSV) into staging tables.
2. Clean data (remove duplicates, nulls).
3. Transform and move to production tables.
4. Create audit table for tracking insert logs.
5. Automate cleanup using triggers.
6. Export final tables and documentation.

Deliverables:

SQL scripts, ETL logs, cleaned production tables.

9. Personal Finance Tracker

Objective: Build a budget tracking SQL database.

Tools: MySQL / SQLite

Mini Guide:

1. Create schema: Users, Income, Expenses, Categories.
2. Insert dummy transactions and user data.
3. Write queries to summarize expenses monthly.
4. Use GROUP BY for category-wise spending.
5. Create views for balance tracking.
6. Export monthly reports.

Deliverables:

SQL schema, reports, budget queries.

10. Social Media Analytics Backend

Objective: Create a SQL system to track post engagement.
Tools: PostgreSQL
Mini Guide:

1. Design schema: Users, Posts, Likes, Comments.
2. Populate data with realistic user activities.
3. Create views for top posts, engagement scores.
4. Use window functions for rankings.
5. Write triggers to update like counts.
6. Export engagement reports.

Deliverables:
DB design, views, ranking queries, data reports.

11. Hospital Management Database

Objective: Manage patient records, appointments, and billing.
Tools: MySQL + DBeaver
Mini Guide:

1. Model schema: Patients, Doctors, Visits, Bills.
2. Insert sample records.
3. Write queries for appointments and payments.
4. Create stored procedures for billing calculations.
5. Add triggers for discharge and status updates.
6. Generate visit reports.

Deliverables:
SQL schema, functions, visit and billing reports.

12. Sports Tournament Tracker

Objective: Manage match results and player statistics.
Tools: MySQL
Mini Guide:

1. Create schema: Teams, Players, Matches, Stats.
2. Insert sample tournament data.
3. Write queries for match results, player scores.
4. Create views for leaderboards and points tables.
5. Use CTE for average player performance.
6. Export team performance reports.

Deliverables:
Schema, queries, leaderboards, summary views.

13. Movie Review and Recommendation Engine

Objective: Analyze movie ratings and user preferences.

Tools: PostgreSQL

Mini Guide:

1. Define schema: Movies, Ratings, Users, Reviews.
2. Insert sample IMDb-style data.
3. Write average rating and ranking queries.
4. Create views for recommended movies.
5. Use window functions to track top-rated content.
6. Export movie recommendation results.

Deliverables:

SQL script, rating views, recommendation report.

14. Crime Record & Investigation Database

Objective: Store and query investigation and criminal records.

Tools: PostgreSQL

Mini Guide:

1. Define schema: Cases, Suspects, Officers, Evidence.
2. Add indexing on case IDs and suspect names.
3. Write queries for solved/unsolved case analysis.
4. Create views for officer workloads.
5. Add trigger for evidence chain updates.
6. Export investigation summaries.

Deliverables:

SQL schema, queries, officer & case reports.

15. Student Result Processing System

Objective: Build SQL system to manage grades and CGPA.

Tools: MySQL

Mini Guide:

1. Design schema: Students, Courses, Grades, Semesters.
2. Insert student and exam data.
3. Write queries for GPA and pass/fail statistics.
4. Create rank lists using window functions.
5. Add triggers for GPA calculation.
6. Export semester-wise result summary.

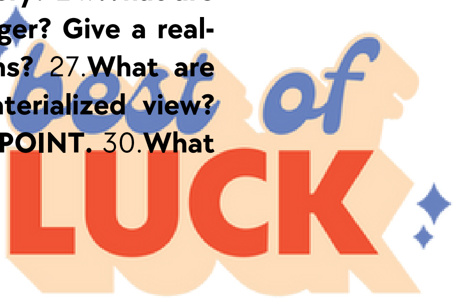
Deliverables:

SQL schema, GPA logic, grade reports, rank list queries.

TOP 50 INTERVIEW QUESTIONS FOR SQL

1. **What is SQL? Howis it different from MySQL or PostgreSQL?**
2. **What are the different types of SQL statements?**
3. **Explain the difference between WHERE and HAVING.**
4. **What are PRIMARY KEY, FOREIGN KEY, UNIQUE, and CHECK constraints?**
5. **What is the difference between DELETE, TRUNCATE, and DROP?**
6. **What is normalization? Explain different normal forms.**
7. **What is denormalization and when is it useful?**
8. **Explain the difference between CHAR and VARCHAR.**
9. **What are ACID properties in databases?**
10. **What is the difference between INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL JOIN?**
11. **Write a query to find the second highest salary from an Employee table.**
12. **Write a query to get the department-wise average salary.**
13. **How would you retrieve duplicate records from a table?**
14. **How do you update a column with a calculation (e.g., 10% tax added)?**
15. **How would you delete only duplicate rows from a table?**
16. **Write a query to list customers who have placed more than 5 orders.**
17. **Write a query to join three or more tables.**
18. **What is a subquery? How is it different from a JOIN?**
19. **What is a correlated subquery? Give an example.**
20. **How do you filter data based on a date range?**
21. **What are WINDOW FUNCTIONS? Name a few.**

22.**What is the use of RANK(), DENSE_RANK(), and ROW_NUMBER()?** 23.**What is a Common Table Expression (CTE)? How is it different from a subquery?** 24.**What are stored procedures? When should they be used?** 25.**What is a trigger? Give a real-world example.** 26.**What is a VIEW? What are its pros and cons?** 27.**What are indexes? How do they improve performance?** 28.**What is a materialized view?** 29.**What are transactions? Explain COMMIT, ROLLBACK, and SAVEPOINT.** 30.**What are aggregate functions? List a few with examples.**



31. **How can you optimize a slow-running SQL query?**
32. **What is the EXPLAIN or EXPLAIN PLAN statement used for?**
33. **How does indexing affect INSERT, UPDATE, and DELETE performance?**
34. **What is a composite index and when should it be used?**
35. **What is normalization overhead and how do you deal with it?**
36. **How do you avoid Cartesian products in JOINs?**
37. **What is partitioning in SQL?**
38. **What causes a deadlock in SQL, and how can you prevent it?**
39. **What is the difference between clustered and non-clustered indexes?**
40. **What tools do you use to monitor SQL query performance?**
41. **You are asked to design a student-course grading system. What tables and relationships would you create?**
42. **How would you store and retrieve attendance for employees in a scalable way?**
43. **In a library system, how would you track overdue books and fines using SQL?**
44. **What would you do if a production database is missing some records due to a failed update?**
45. **How would you implement role-based access to sensitive information in SQL?**
46. **You are given a raw CSV with dirty data. How would you load and clean it using SQL?**
47. **How would you calculate monthly retention from a user login dataset?**
48. **What measures would you take to secure a database with sensitive data?**
49. **How do you create daily backup and restore plans for a SQL database?**