**Joins (1–15)**

1. Write an SQL query to display customer names with their plan names using JOIN.
2. List all calls with customer name and duration using INNER JOIN.
3. Get customer names who have not made any payments using LEFT JOIN.
4. Show customers who have both call and payment records using INNER JOIN.
5. Find customers and their plans even if the plan details are missing using LEFT JOIN.
6. List each call with its customer name and plan's call rate.
7. Retrieve all customers and their latest payment date using JOIN and GROUP BY.
8. Show calls with charges above ₹20 and their customer names.
9. List all payments along with customer names and plan types.
10. Find total payments made by each customer along with their contact info.
11. Display all customers even if they haven't made any calls or payments.
12. Get customer names and the number of calls they've made.
13. Show each call with the corresponding customer's plan's monthly fee.
14. List customer names and total charges they've incurred.
15. Show all plans and the count of customers subscribed to each.

**UPDATE Queries (16–22)**

1. Increase the monthly fee of all "Gold" plans by ₹100.
2. Reduce the call rate of all "Platinum" plans by 10%.
3. Update a customer's contact number based on customer ID.
4. Change a customer's plan type from "Silver" to "Gold".
5. Set the contact as NULL for customers whose name starts with 'Test'.
6. Update charges for calls longer than 10 minutes to a fixed value of ₹50.
7. Set all call durations to 0 for a given date as part of a cleanup.

**DELETE Queries (23–28)**

1. Delete all calls made before January 2024.
2. Remove payments with an amount of 0.
3. Delete customers with no associated calls or payments (use NOT EXISTS).
4. Delete plans that are not used by any customer.
5. Delete all data from calls table (TRUNCATE).
6. Delete a customer and all their related calls and payments (using transactions).

**Views (29–34)**

1. Create a view to show customer name, plan type, and call count.
2. Create a view to display total payments per customer.
3. Create a view that shows daily call volume (total calls per date).
4. Create a view for all customers with their total charges and total payments.
5. Create a view of customers who have made high payments (₹500+).
6. Drop the view created above.

**📌 Indexes (35–38)**

1. Create an index on customer\_id in the calls table.
2. Create a composite index on plan\_type and monthly\_fee in plans.
3. Drop an index on the payments table.
4. Analyze the query performance difference with and without index on call\_date.

**📥 Import/Export (39–42)**

1. Write the LOAD DATA INFILE statement to import a CSV file into the customers table.
2. Export the calls table data to a CSV file using SELECT ... INTO OUTFILE.
3. Write the command to import SQL data using mysql -u command line.
4. Write the command to export the entire database using mysqldump.

**Transactions (43–46)**

1. Perform a transaction to deduct a payment and insert it into the payments table.
2. Begin a transaction, update customer contact, and rollback.
3. Transfer ₹500 from one customer’s account to another and commit.
4. Show how to use START TRANSACTION, COMMIT, and ROLLBACK.

**Constraints (47–50)**

1. Add a UNIQUE constraint on contact in the customers table.
2. Add a CHECK constraint to ensure charge >= 0 in calls.
3. Create a FOREIGN KEY constraint between calls.customer\_id and customers.customer\_id.
4. Modify the plans table to add a NOT NULL constraint to call\_rate