Dropping Tables: This section contains the **DROP TABLE** statements to delete any existing tables in the database related to customers, staff, and reservations.

Dropping Sequences: This section contains the **DROP SEQUENCE** statements to delete any existing sequences in the database related to customers and staff.

Creating Sequences: This section contains the **CREATE SEQUENCE** statements to create new sequences for customer and staff tables.

Creating Tables: This section contains the **CREATE TABLE** statements to create new tables in the database, such as customer, staff, waiter, kitchen worker, receptionist, request, reservation, personal table, and VIP room.

Inserting Data: This section contains the **INSERT** statements to insert sample data into the tables. This includes customer information, staff information, waiter details, kitchen worker details, receptionist details, request details, reservation details, personal table details, and VIP room details.

Select Queries: This section contains a series of **SELECT** statements designed to retrieve and display various pieces of information from the database. These queries include:

Displaying all customers with their reservation details.

Displaying all staff members who are waiters and their tip details.

Displaying all customers with their request details and the staff members responsible for handling their requests.

Calculating the total prices of requests based on their status (e.g., pending or happened).

Calculating the average tip received by waiters.

Displaying all customers who have booked a VIP room.

Displaying all reservations made by customers who have a pending request.

Displaying all staff members who are kitchen workers and their experience levels.

Dropped two triggers (trg\_request\_price\_increase and trg\_viproom\_check\_price) if they exist.

Dropped two stored procedures (list\_pending\_requests and update\_customer\_type).

Created two triggers: a. trg\_request\_price\_increase: Increases the request price by 10% if the request is assigned to a VIP customer. b. trg\_viproom\_check\_price: Checks if the VIP room price is above the minimum allowed price.

Created two stored procedures: a. list\_pending\_requests: Lists all pending requests with customer and staff details. b. update\_customer\_type: Updates customer type based on the total amount spent on requests.

Demonstrated the usage of the stored procedures.

Created an index on the customer\_type column of the customer table to speed up the check for VIP customers in the trg\_request\_price\_increase trigger.

Demonstrated the usage of EXPLAIN PLAN to understand and optimize a query by creating an index.

Granted access rights to another team member (XMASHL00) for various tables.

Dropped a materialized view (vip\_customers) if it exists, and created a materialized view to store the number of reservations for each VIP customer.

Refreshed the materialized view.

Written a complex SELECT query using the WITH clause and the CASE operator to get the total amount spent by each customer, and their calculated and actual customer types.