Hotel Management System

PROJECT REPORT

Group Members

MAHNDSE23.2F-007 J.G. Sachini Gayanika

MADHNSE23.2F-008 W.G. Ashen Wishwa Geeth Jayarathna

ADVANCED DATABASE MANAGEMENT SYSTEM

Assessment NO:02

Higher Diploma in Software Engineering

National Institute of Business Management

Under supervision

Ms. Kaushalya Dissanayake

Consultant/Lecture

DECLARATION

"I certify that this project does not incorporate without acknowledgement, any material previously submitted for a Diploma in any institution and to the best of my knowledge and belief ,it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my project report, if accepted, to be made available for photocopying and for interlibrary loans, and for the title and summary to be made available to outside organizations"

Index no	**Name**	**Signature**
MAHNDSE23.2F-007	J.G. Sachini Gayanika	……..
MAHNDSE23.2F-008	W.G. Ashen Wishwa Geeth Jayarathna	……..

Acknowledgement

We would like to express our sincere appreciation to everyone who contributed to the successful completion of this assignment. First and foremost, we extend our gratitude to our lecturer, MS Kaushalya Dissanayake for providing us with the opportunity to explore and deepen our knowledge of Oracle database management and PL/SQL programming. We would also like to thank our fellow group members for their collaboration and shared commitment to the project. Their insights and hard work greatly enriched the final deliveries. Additionally, we are thankful for the support and resources provided by our institution, which made this assignment possible. Last but not least, we appreciate the patience and understanding of our friends and family during the intense work on this project. Their encouragement and support were invaluable throughout this journey.

TABLE OF CONTENT

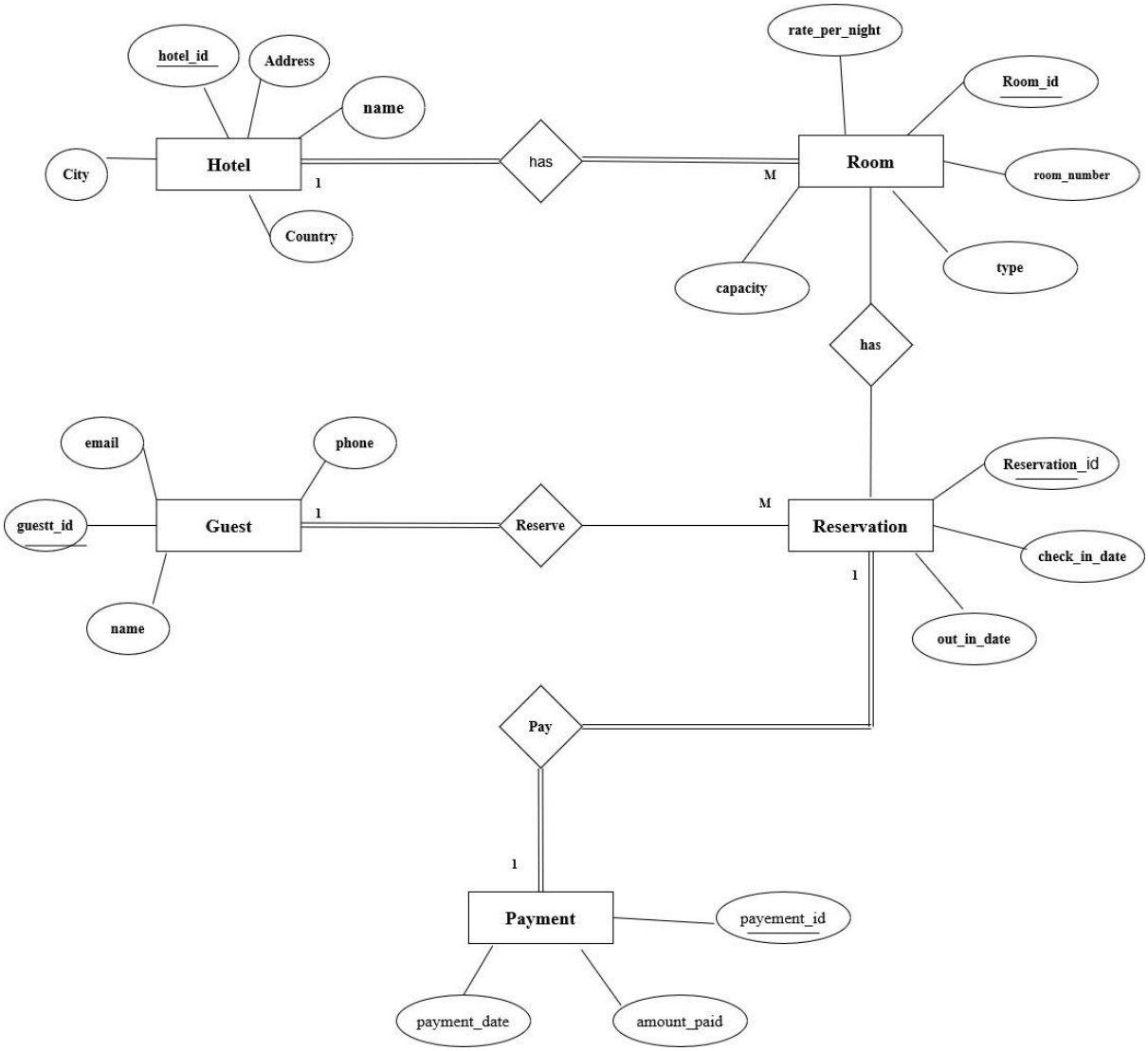
Contents

LIST OF TABLES

LIST OF FIGURES

ER DIAGRAM

ER diagram for hotel management system



Chapter 1
Introduction

In the dynamic hospitality industry l, efficient management of hotel operations is paramount for providing, exceptional guest experience and maximizing profitability.

A Hotel management system facilitates the automation of numerous operational tasks. reducing manual errors and improving are all efficiency. This system handing guest reservations and check-in to manage room in inventory and billing processes.

The primary goal is a enhance efficiency reduce manual workload and improve organizational effectiveness.

1.1 Background

The hotel management system project aims to provide an efficient and organized solution for managers a hotel resources among employees and guests. In this comprehensive database design, we have created tables for hotels, rooms, guests, reservations, and payment, each meticulously structured with primary keys, foreign keys, check constraints, and other integrity features.

1.2 Problem Statement

The objective of this project is to design and develop a comprehensive Hotel Management System that streamlines and automates various management processes within a hotel. The system aims to improve overall efficiency, communication, and data management while providing a user-friendly system for all employees and guests Control waste time unnecessarily.

- Unnecessary waste of money.
- Prevention of Data Loosing.
- Weaknesses in file usage

Aim and Objectives

This project's major goal is to provide a comprehensive and user-friendly system that makes it easier for guests and employees to manage and communicate effectively. The system tries to address issues with resource allocation, data management, and communication breakdowns the hotels.

- **In short,**
- Efficiency
- Guest Satisfaction
- Revenue Optimization
- Data-Driven Decisions
- Communication
- Security and Compliance
- Adaptability

CHAPTER 2

Design and Result

Hotel table

Table 1:Hotel Table

NAME	DATA TYPE	CONSTRAINT
Hotel_id	INT	PRIMARY KEY
name	VARCHAR	NOT NULL
address	VARCHAR	
city	VARCHAR	
country	VARCHAR	

Room table

Table 2:Room Table

NAME	DATA TYPE	CONSTRAINT
Room_id	INT	PRIMARY KEY
Hotel_id	INT	Foreign key
Room_number	INT	
type	VARCHAR	
Rate_per_night	NUMBER	
capacity	INT	

Guest table

Table 3:Guest Table

NAME	DATA TYPE	CONSTRAINT
Guest_id	INT	PRIMARY KEY
name	VARCHAR	NOT_NULL
email	VARCHAR	
phone	VARCHAR	

Reservation table

Table 4:Reservation Table

NAME	DATA TYPE	CONSTRAINT
Reservation_id	INT	PRIMARY KEY
Guest_id	INT	Foreign key
Room_id	INT	Foreign key
Check_in_date	DATE	
Chek_out_date	DATE	

Payment table

Table 5:Payment Table

NAME	DATA TYPE	CONSTRAINT
Payment_id	INT	PRIMARY KEY
Reservation_id	INT	Foreign key
Amount_paid	NUMBER	
Payment_date	DATE	

1) Create a new user with assigning necessary privileges [Should provide explanation of user].

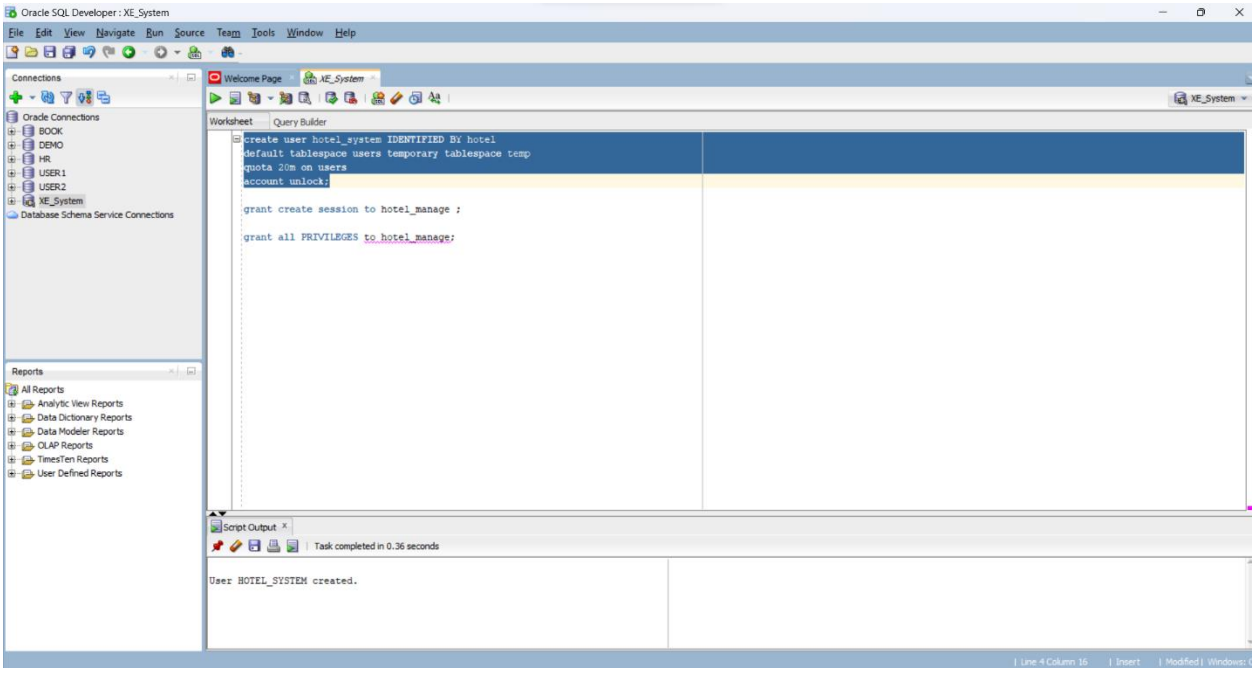


Figure 1 Create User

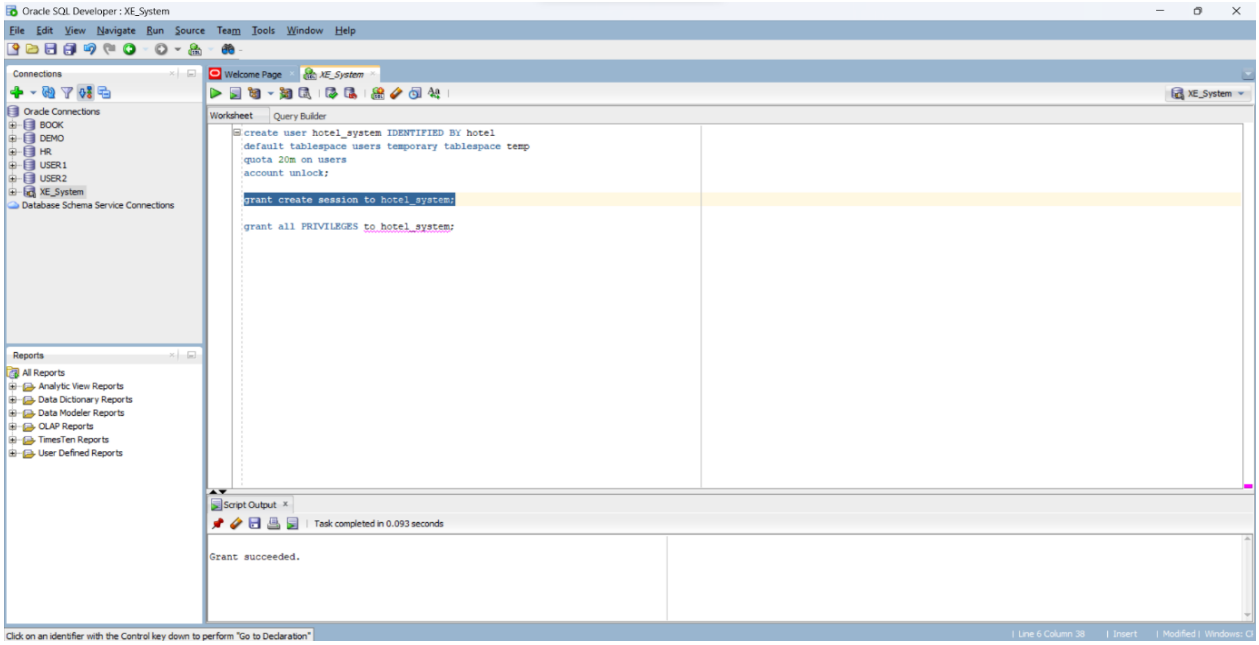


Figure 2Grant

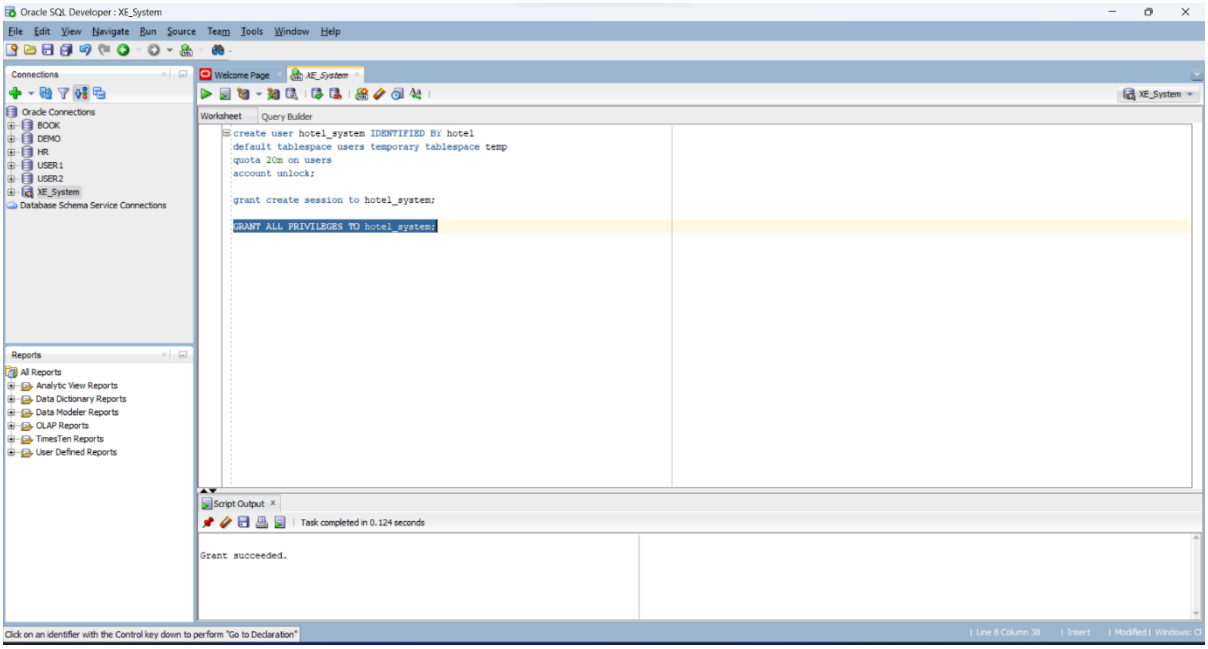


Figure 3Grant All

2) Create a new database connection for above created user.

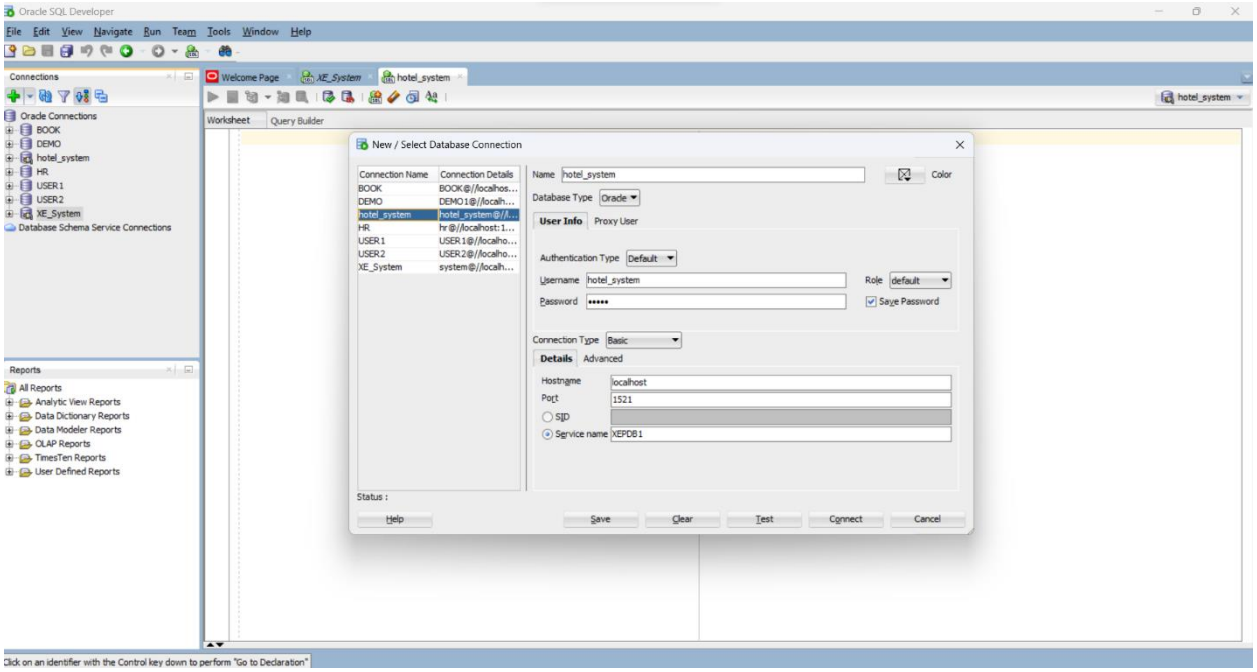


Figure 4:Create Connection

3) The tables should create with considering following features.

- Primary key and foreign key constraints
- Check Constraint • Not Null Constraint

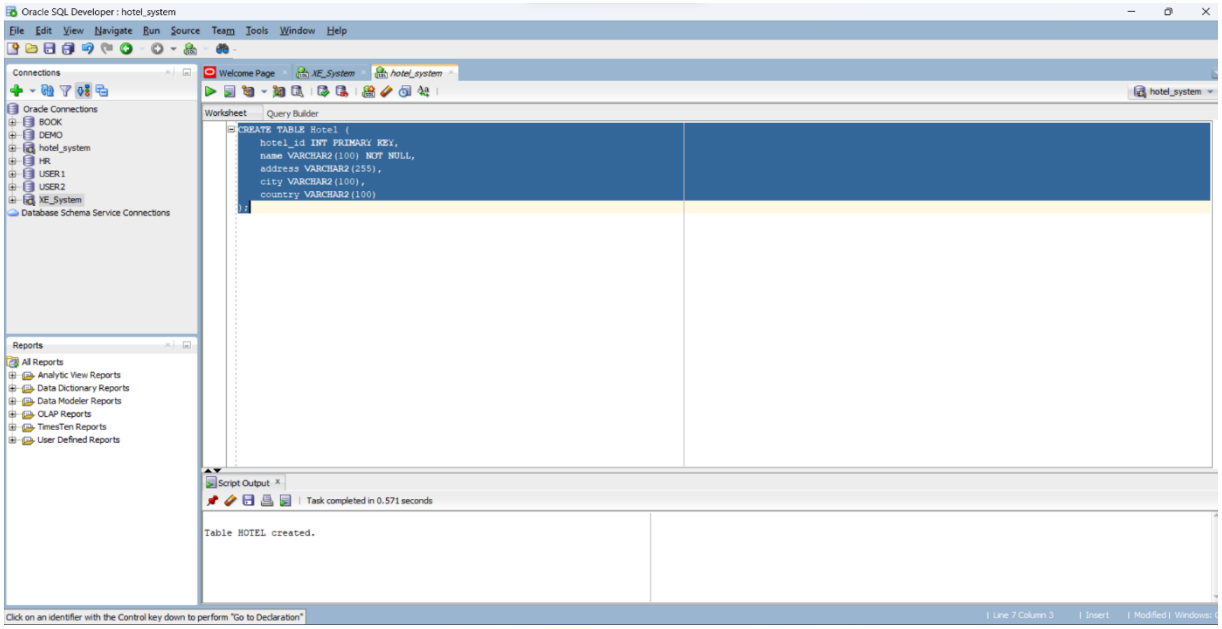


Figure 5:Create Hotel Table

To identify the identity of the hotel, details such as hotel_id, name, address, city and country are in this hotel table. To identify each hotel, identity is identified by hotel_id as its primary key

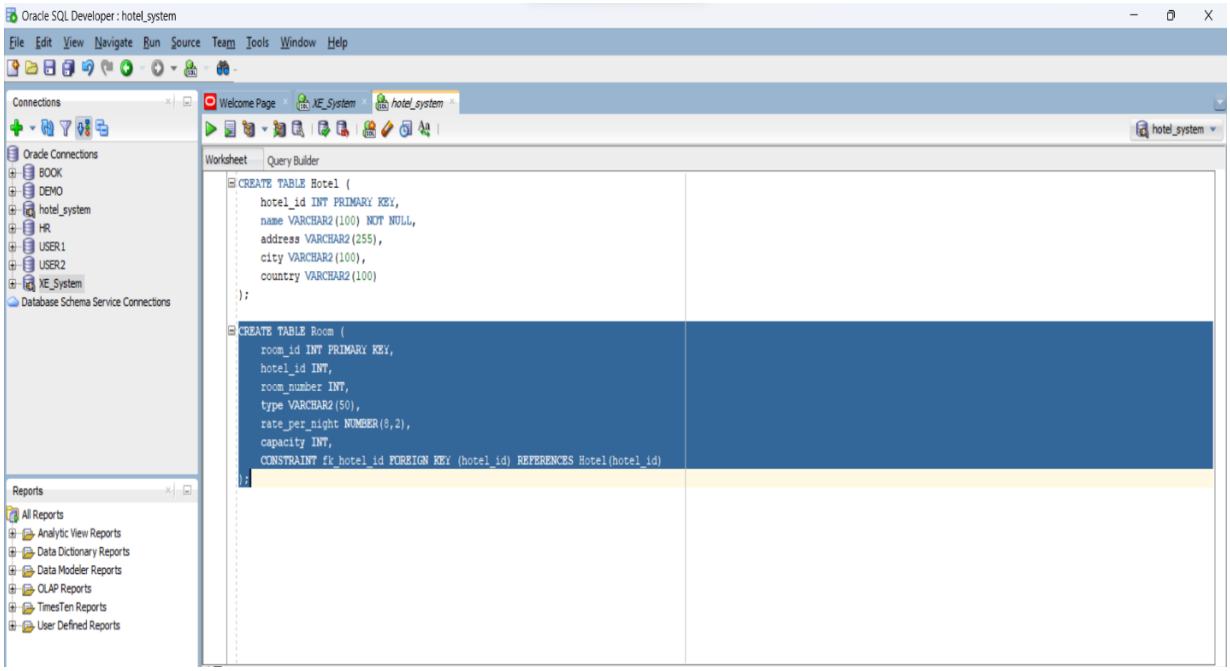


Figure 6:Create Room Table

Here each room acts as a single room and includes attributes such as room_id, hotel_id, room number, room type (single, double), rate_per_night and capacity. This facilitates room reservation and price management.

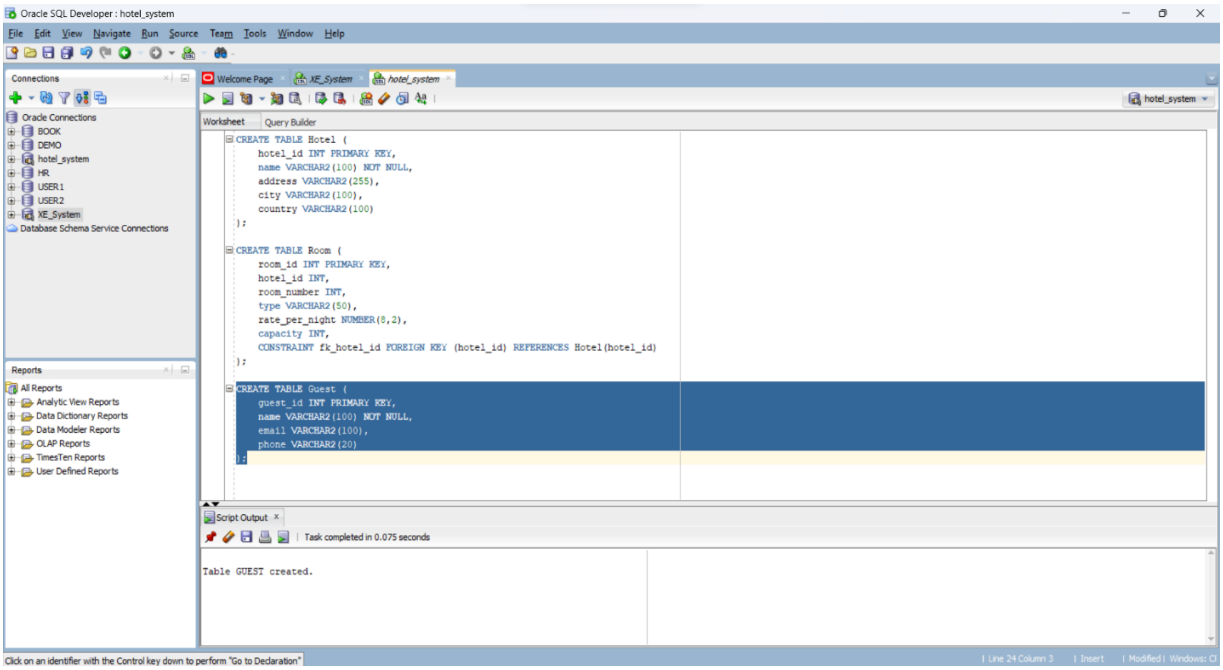


Figure 7:Create Guest Table

It provides information about the guests staying at the hotel. it include guest id,name,email,address and phone number.guest id is the primary key is the table.its used identified the guest separately.this table can be taken record of customers can be taken from this table.

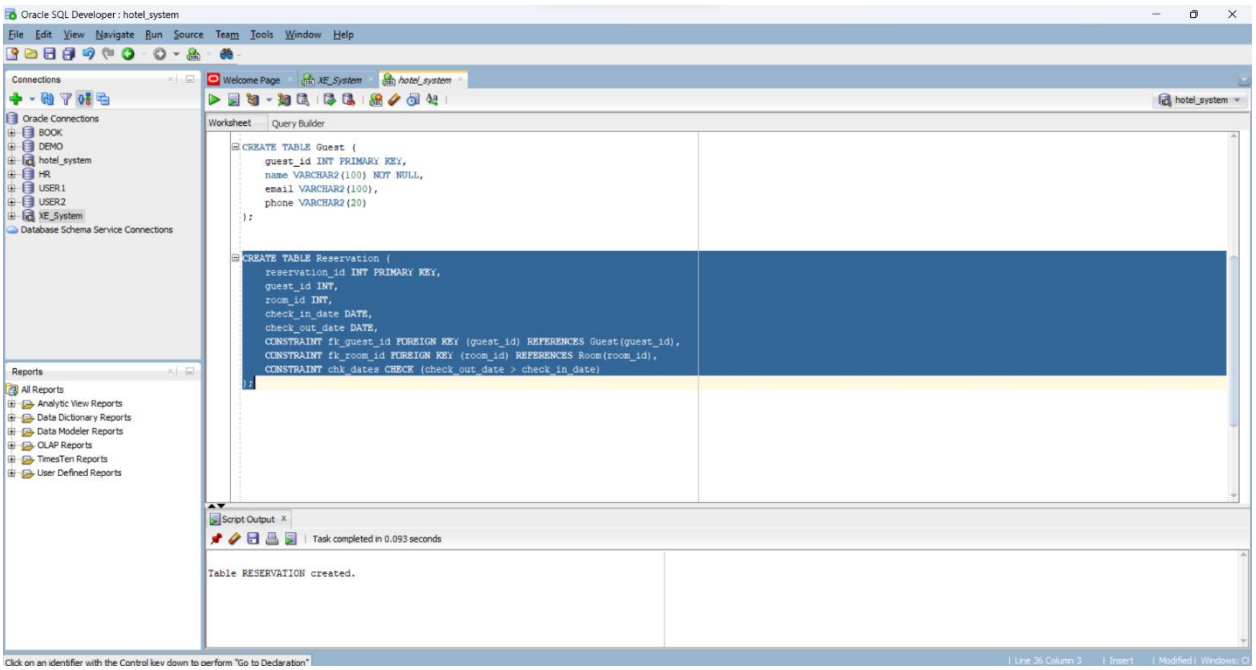


Figure 8:Create Reservation Table

This table shows us deta details of reservations, including the reservation ID, guest ID, guest name, guest email, room ID, check-in date, and check-out date.

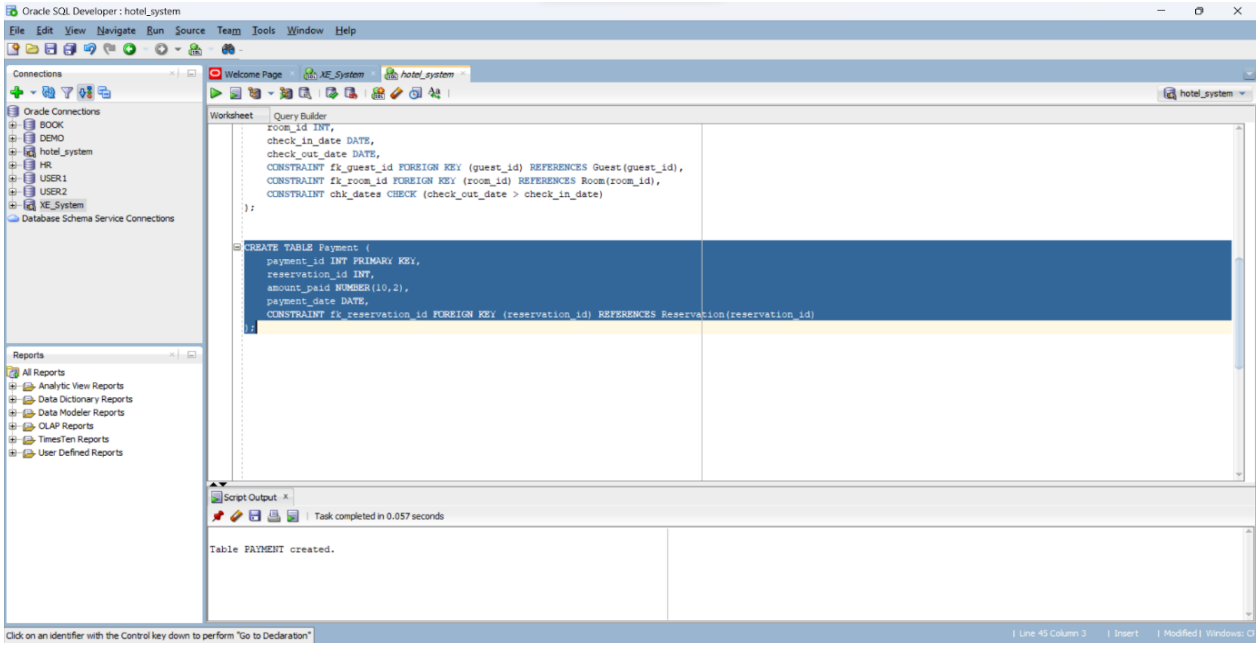


Figure 9:Create Payment Table

Payment

The payment table is managed on the financial side.it include unique identifier payment_id, reservation_id,amount_paid and payment_date. reservation_id is link to the reservation table. This table makes it easy to determine the payment related to guest bookings

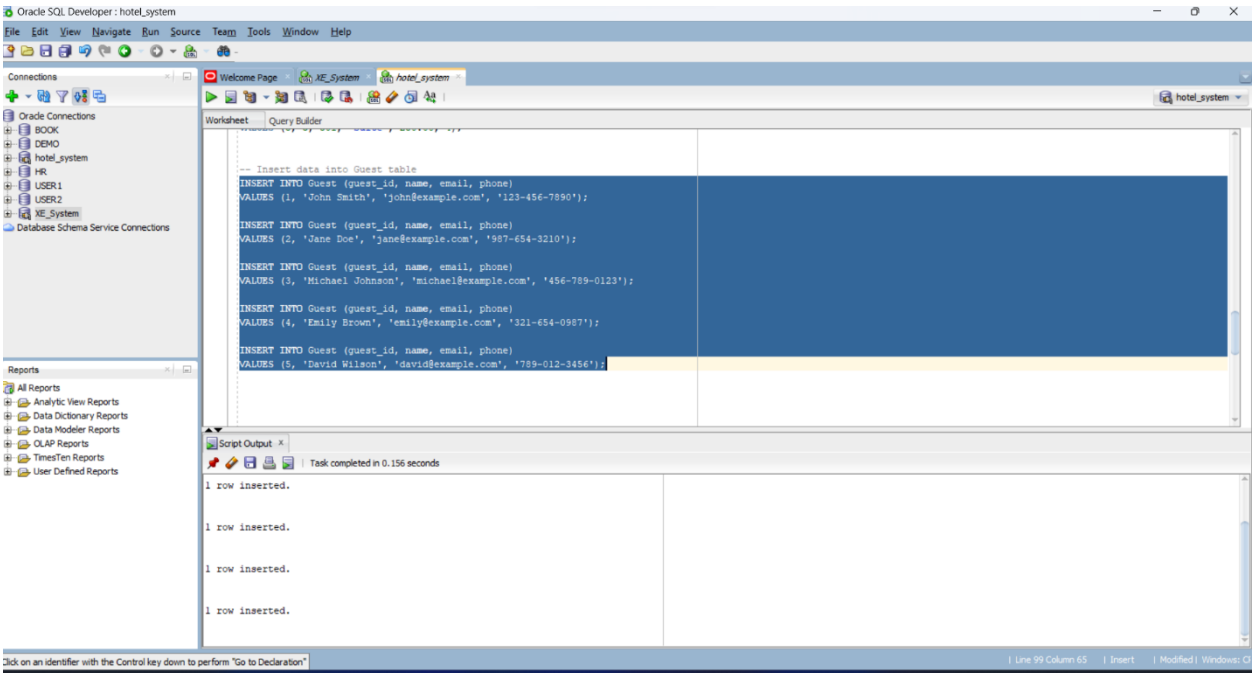


Figure 10:Insert data to Guest table

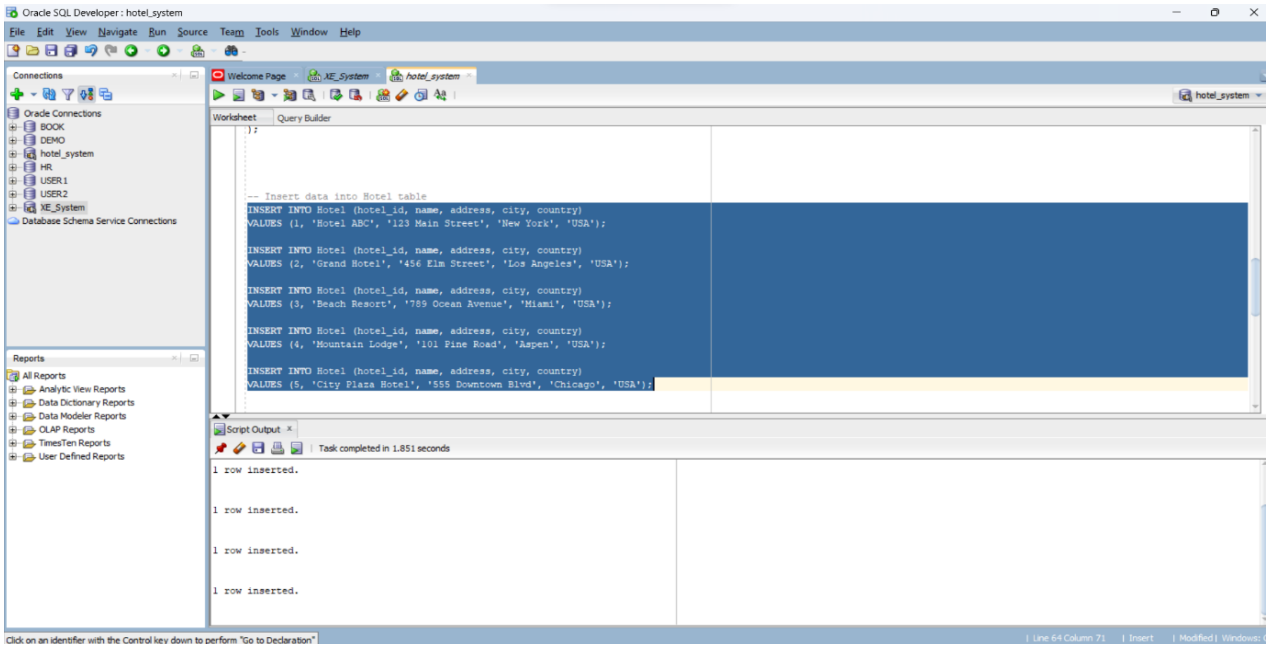


Figure 11:Insert Data to Hotel Table

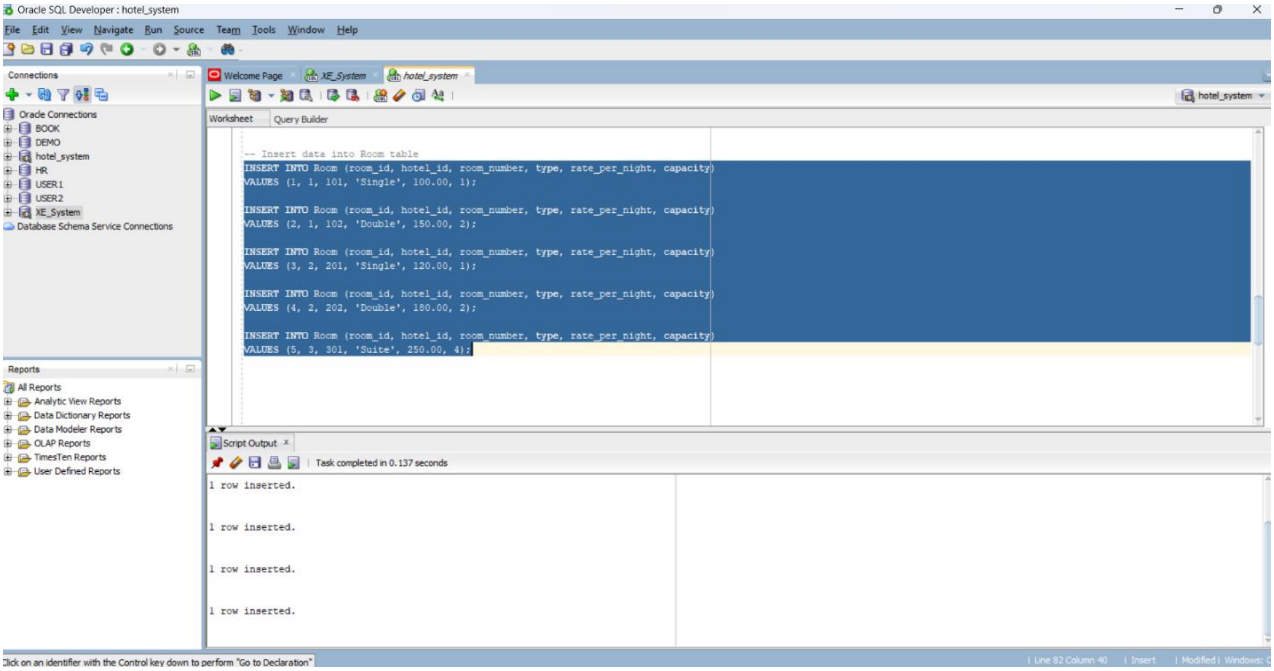


Figure 12:Inset Data to Room Table

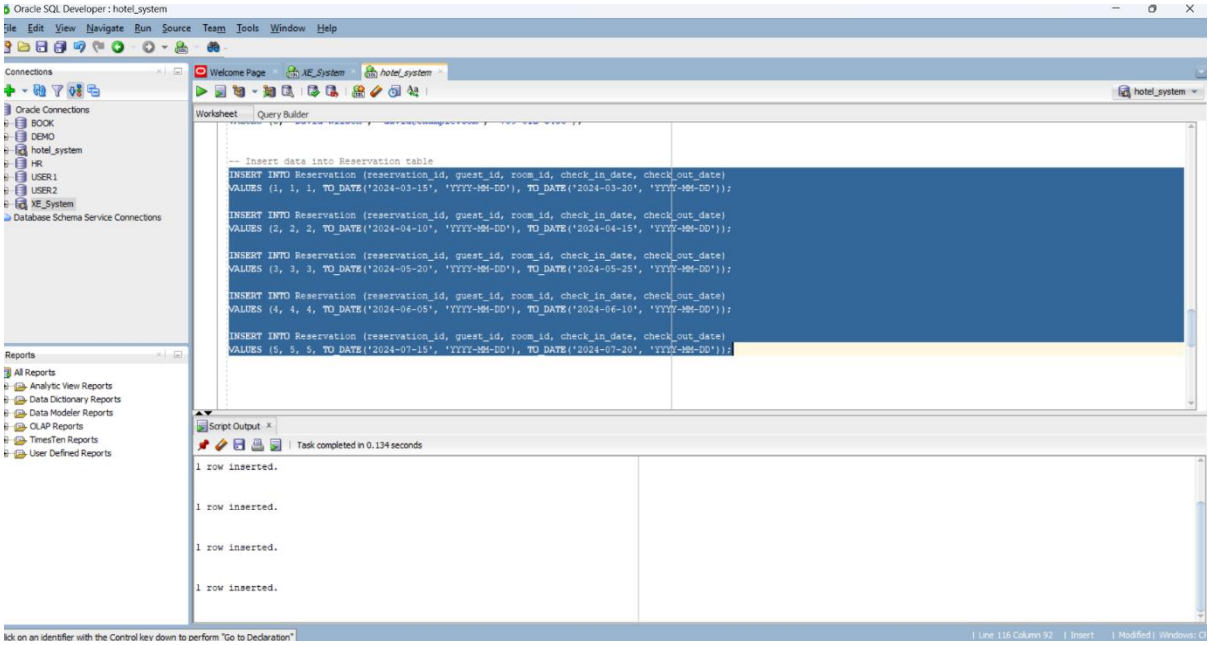


Figure 13:Insert Data to Reservation Table

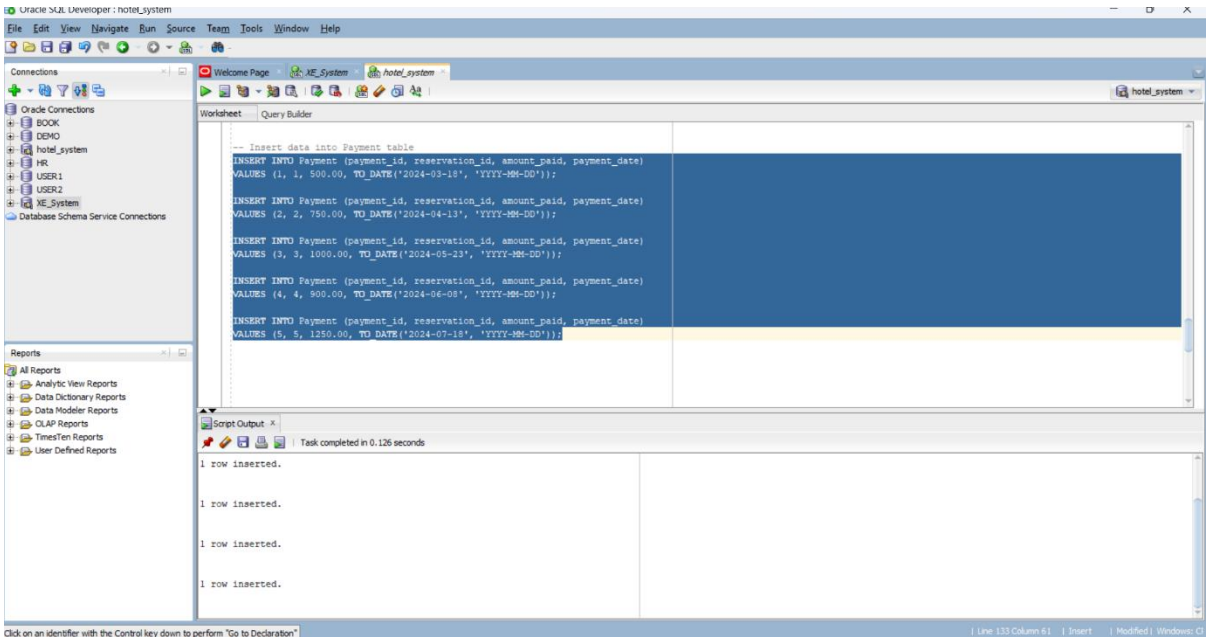


Figure 14:Insert Data to Payment Table

4) Write 20 DQL [Data Query Language] Command with covering following criteria.
   • Where • Group by • Having • Order by • Like • Count() • Length() • initcap() • Max() • Min()
   • AND • OR • Join

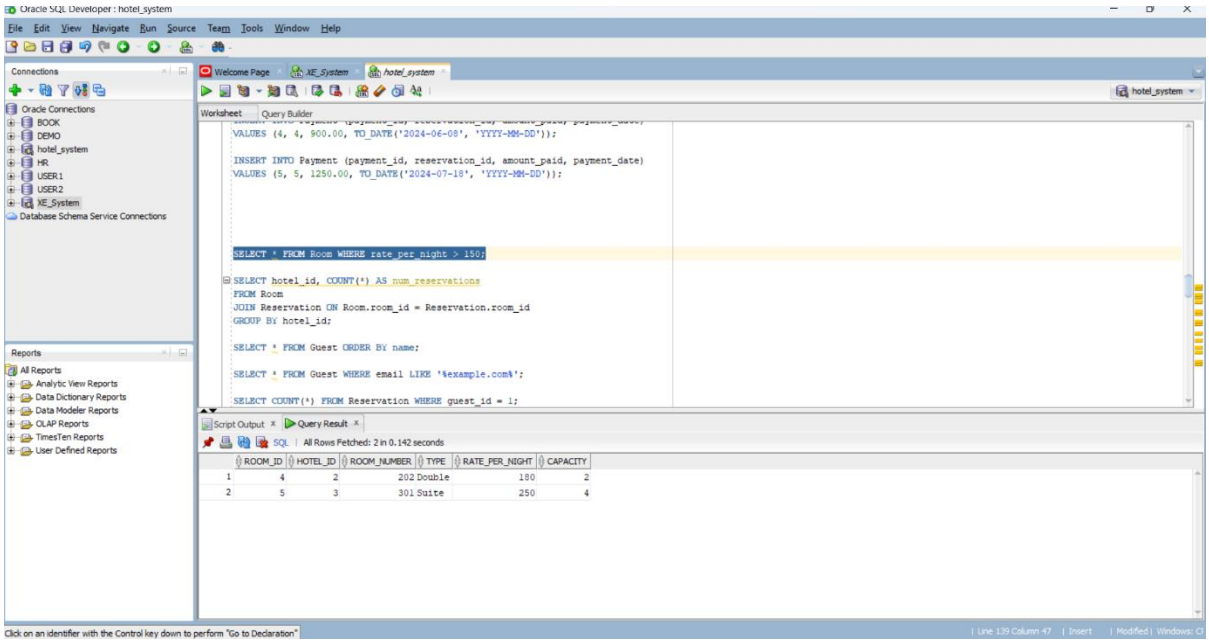


Figure 15:DQL Query

Retrieves all rooms where the rate per night is greater than $150.

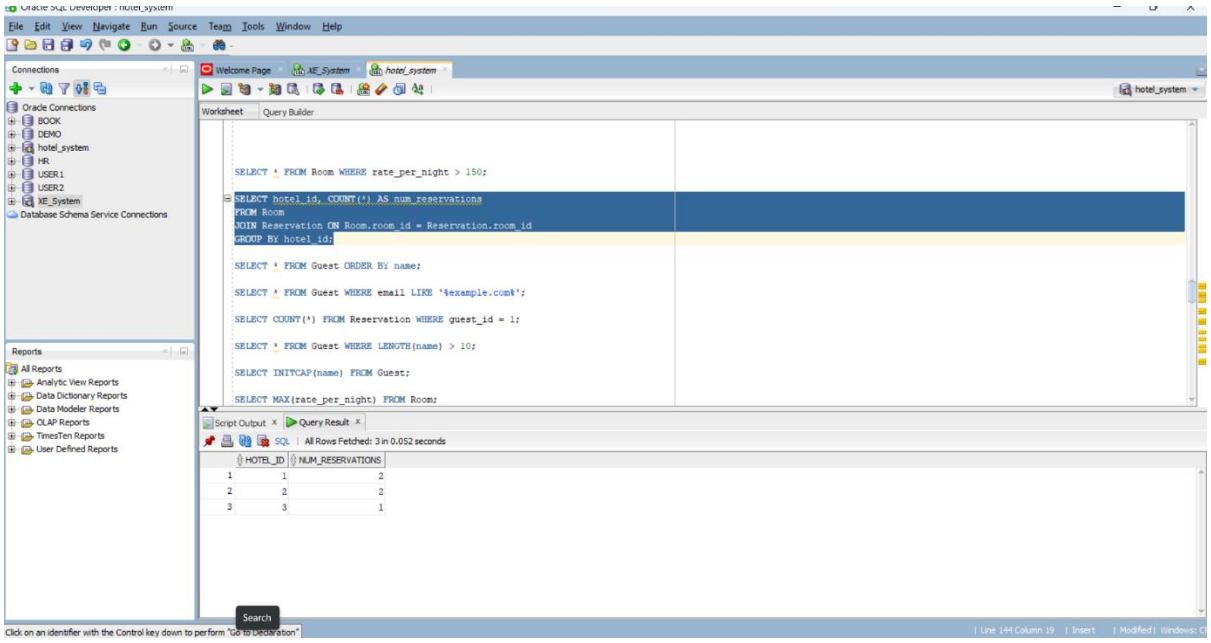


Figure 16:DQL Query

Counts the number of reservations for each hotel.

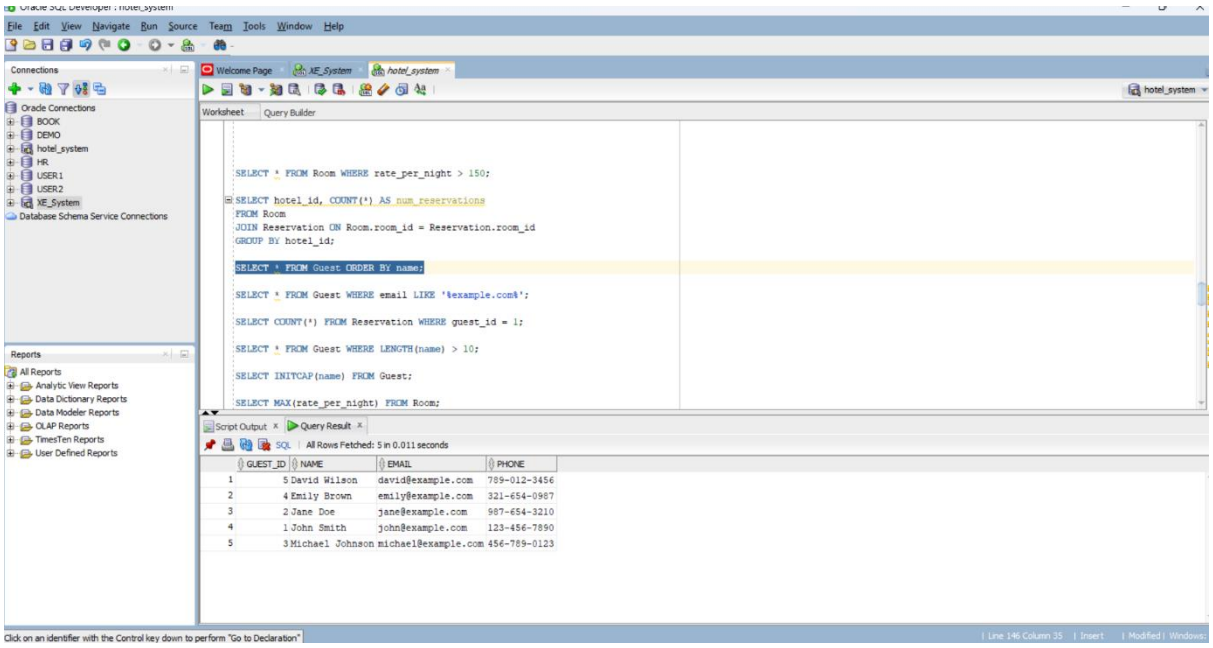


Figure 17:DQL Query

Retrieves all guests and sorts them alphabetically by name.

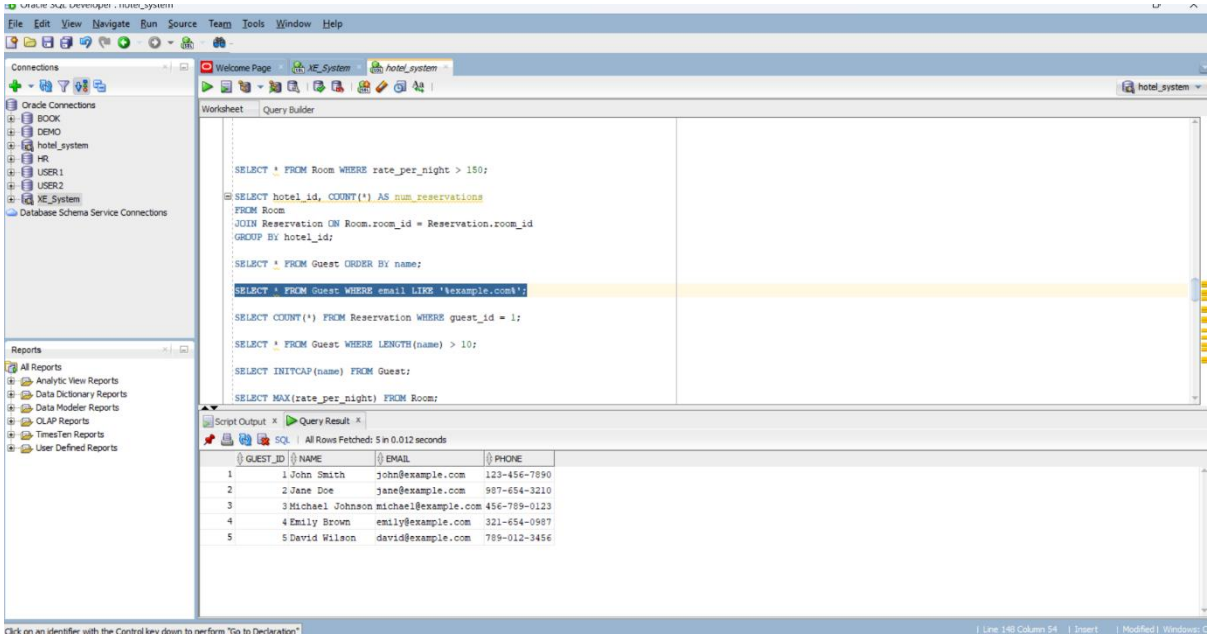


Figure 18:DQL Query

Retrieves guests whose email addresses contain the domain 'example.com'.

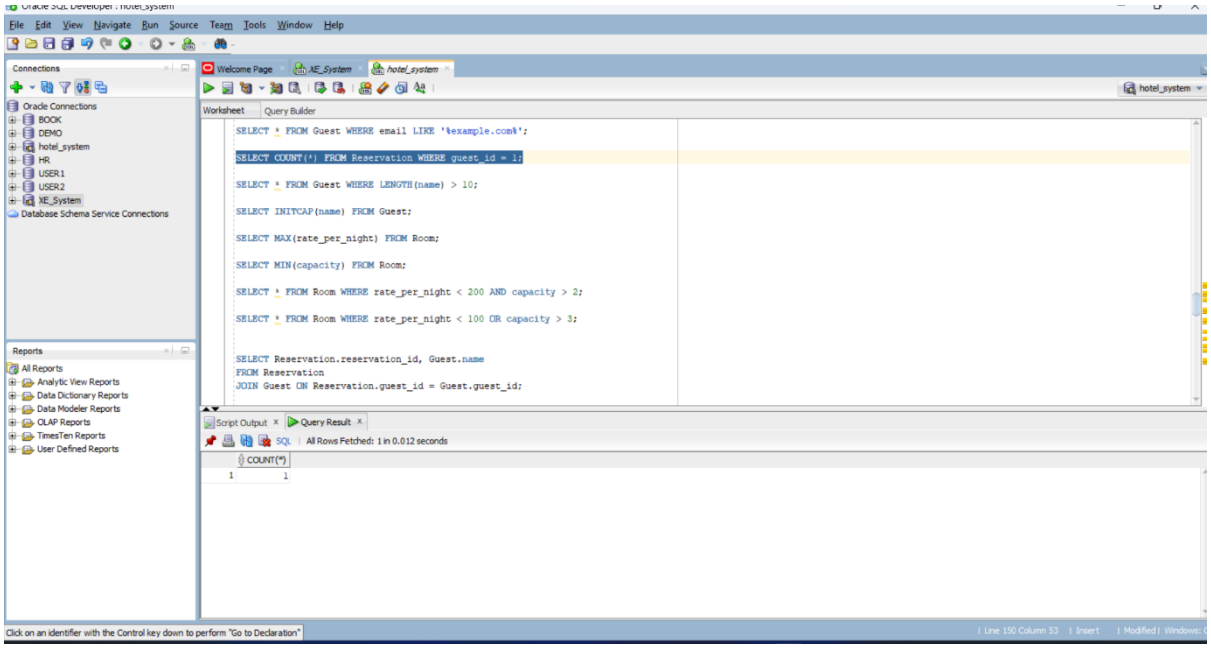


Figure 19:DQL Query

Counts the number of reservations made by a specific guest (guest_id = 1).

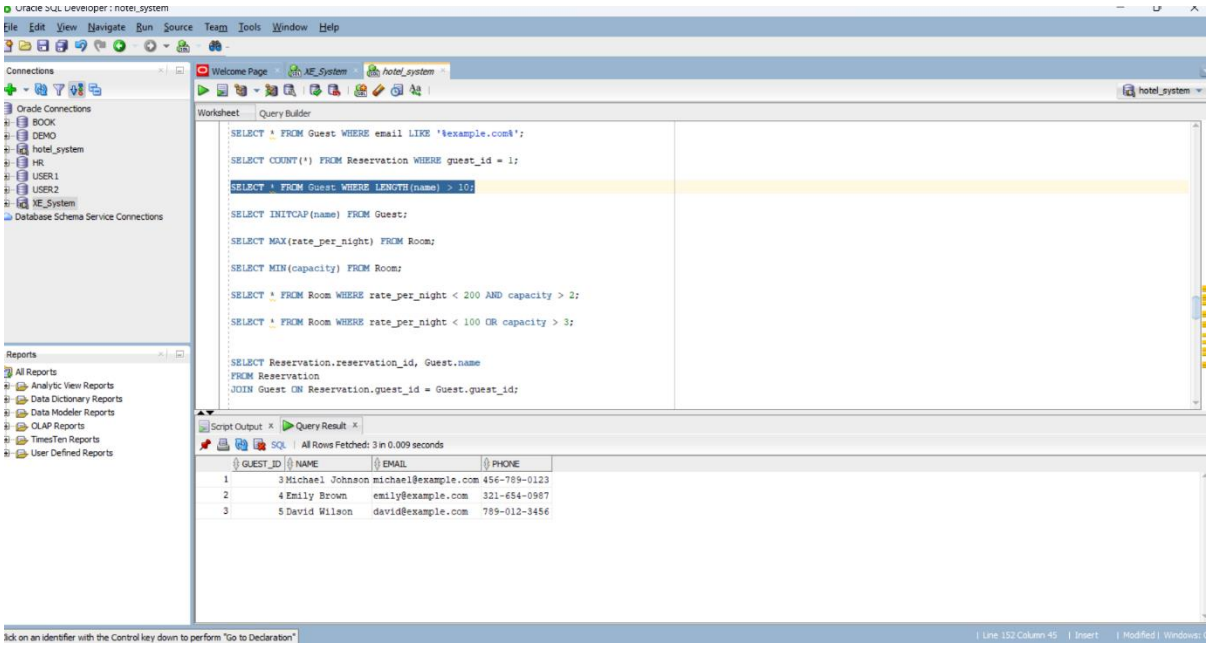


Figure 20:DQL Query

Retrieves guests with names longer than 10 characters.

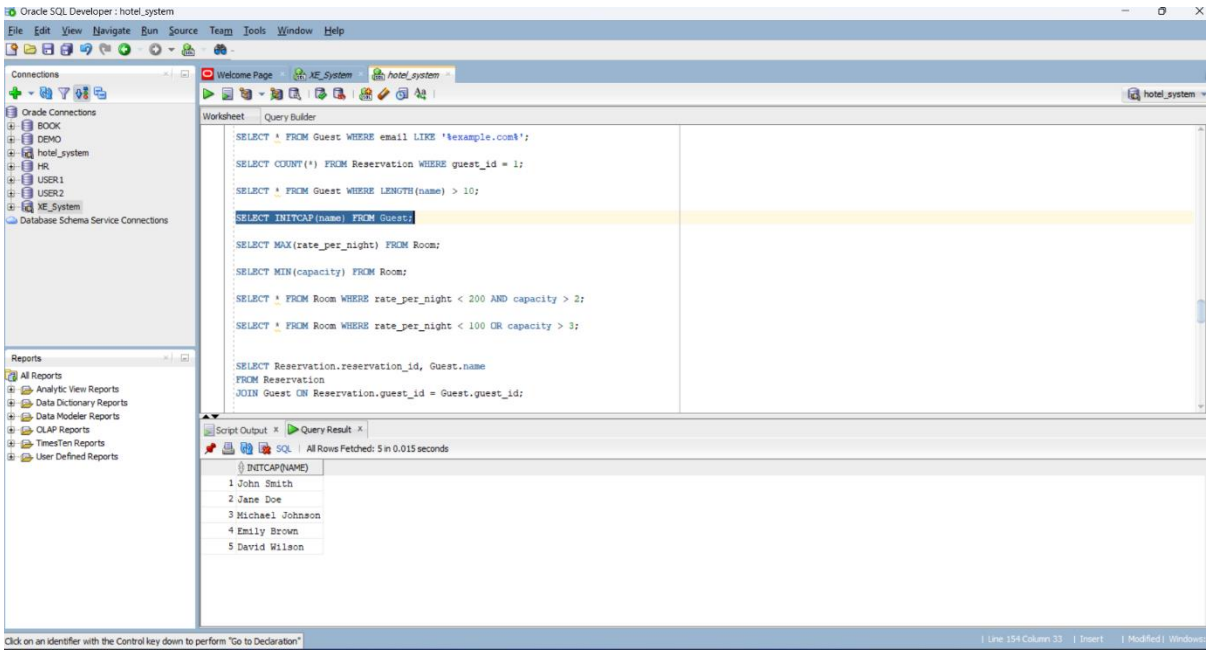


Figure 21:DQL Query

Retrieves guest names with the first letter of each word capitalized.

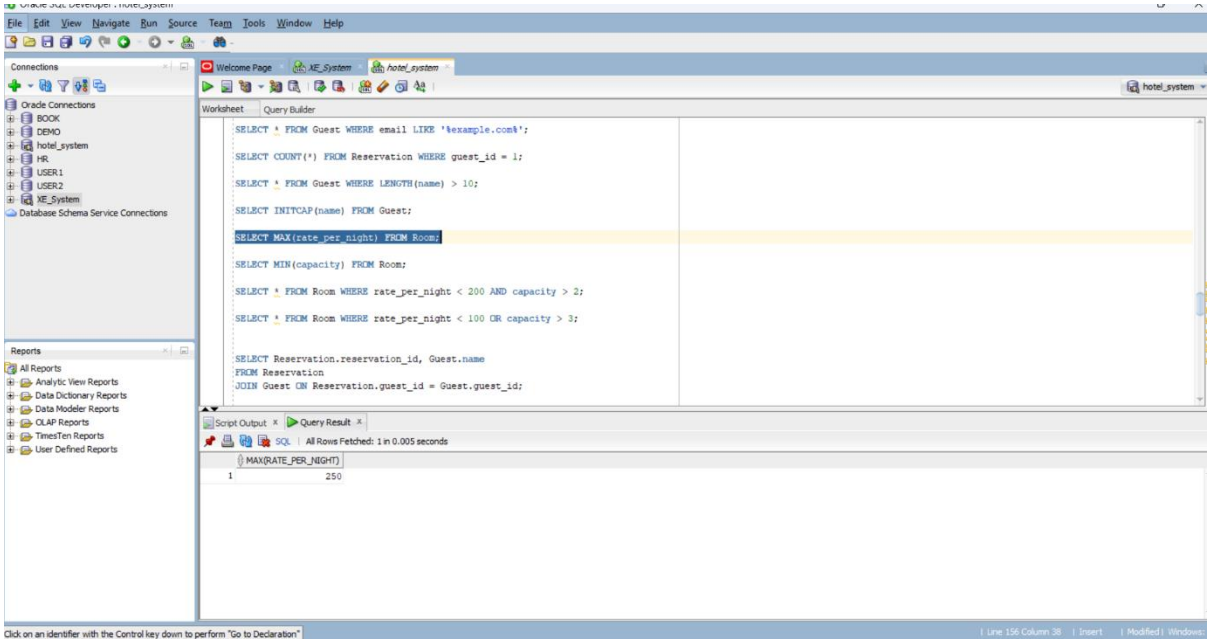


Figure 22:DQL Query

Retrieves the maximum rate per night across all rooms

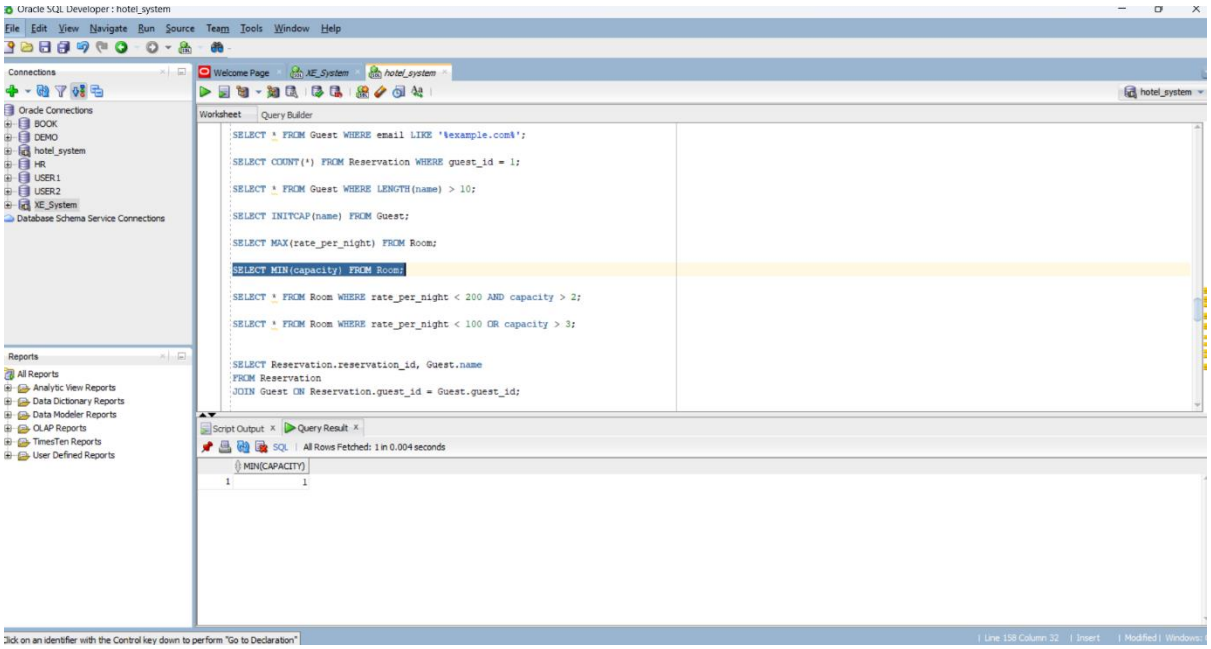


Figure 23:DQL Query

Retrieves the minimum capacity across all rooms.

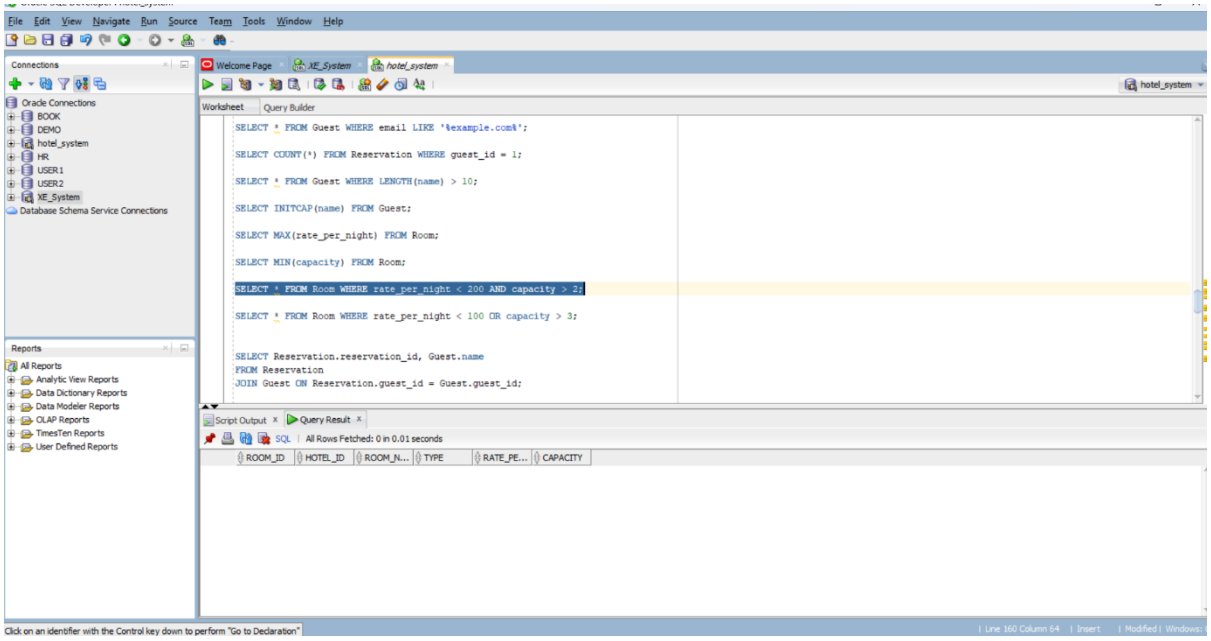


Figure 24:DQL Query

Retrieves rooms with a rate per night less than $200 and a capacity greater than 2.

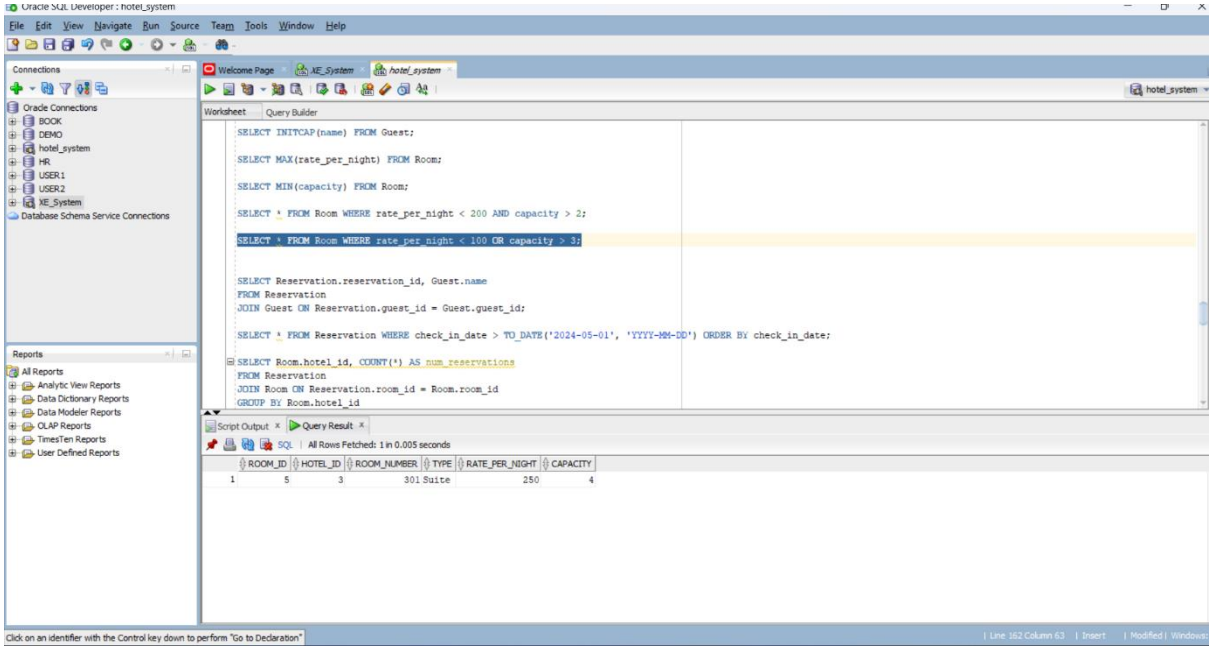


Figure 25:DQL Query

Retrieves rooms with a rate per night less than $100 or a capacity greater than 3.

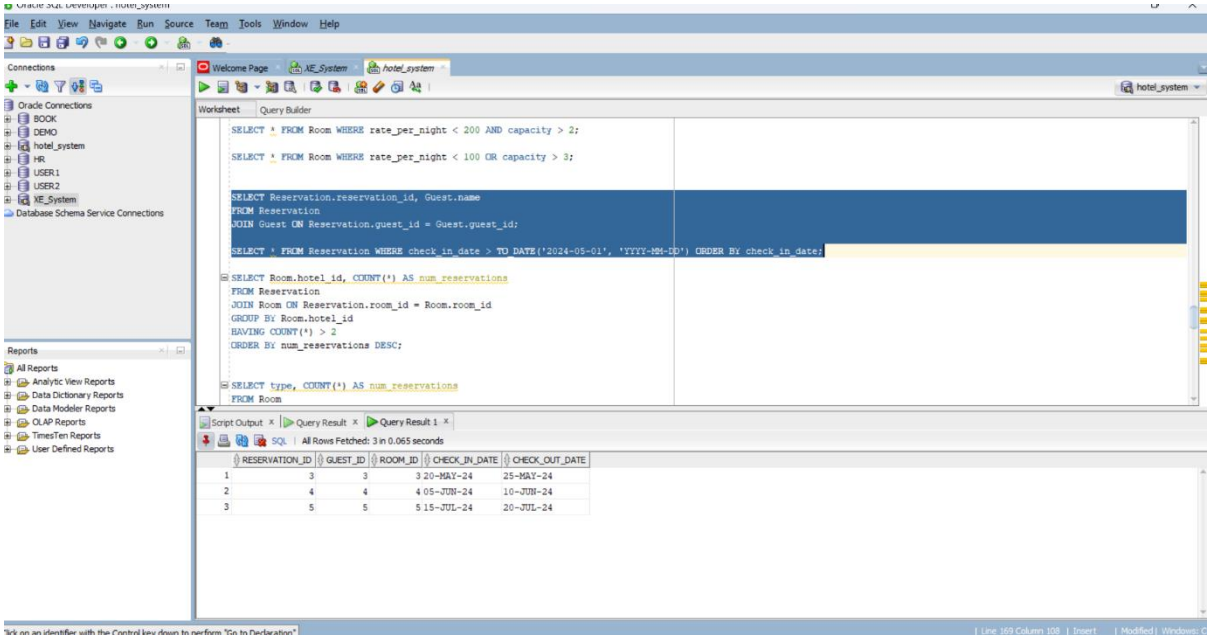


Figure 26:DQL Query

Retrieves reservation IDs along with the names of guests who made those reservations.

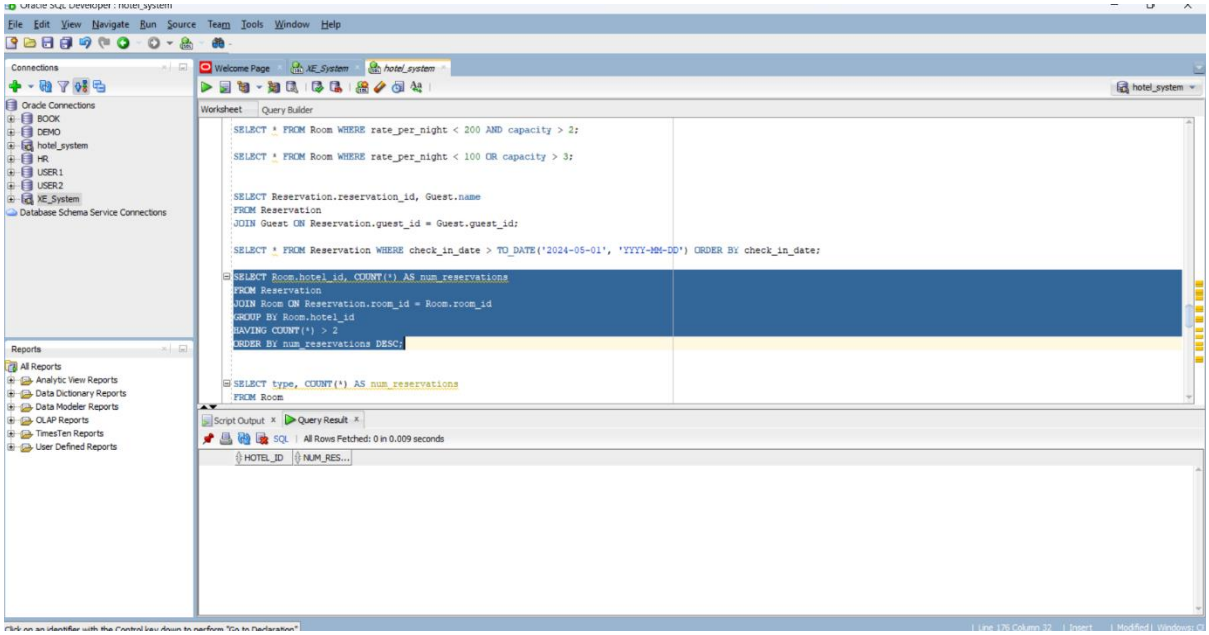


Figure 27:DQL Query

Counts the number of reservations per hotel and filters for hotels with more than 2 reservations, ordering them by the number of reservations.

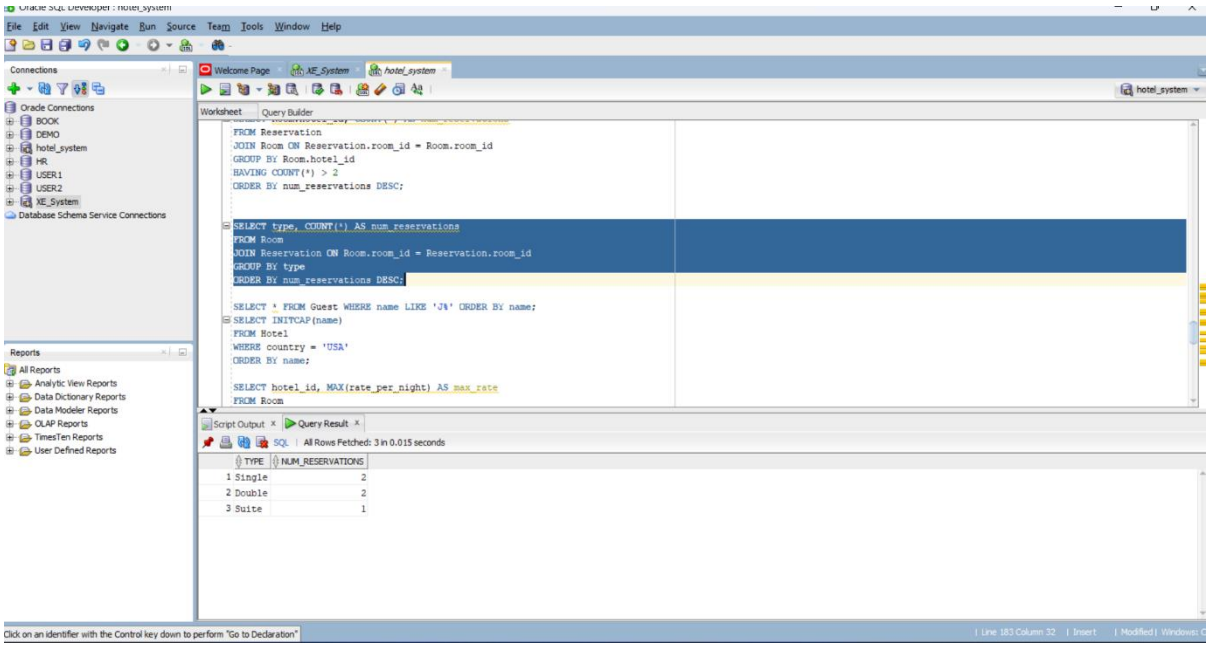


Figure 28:DQL Query

Counts the number of reservations for each room type, ordering them by the number of reservations.

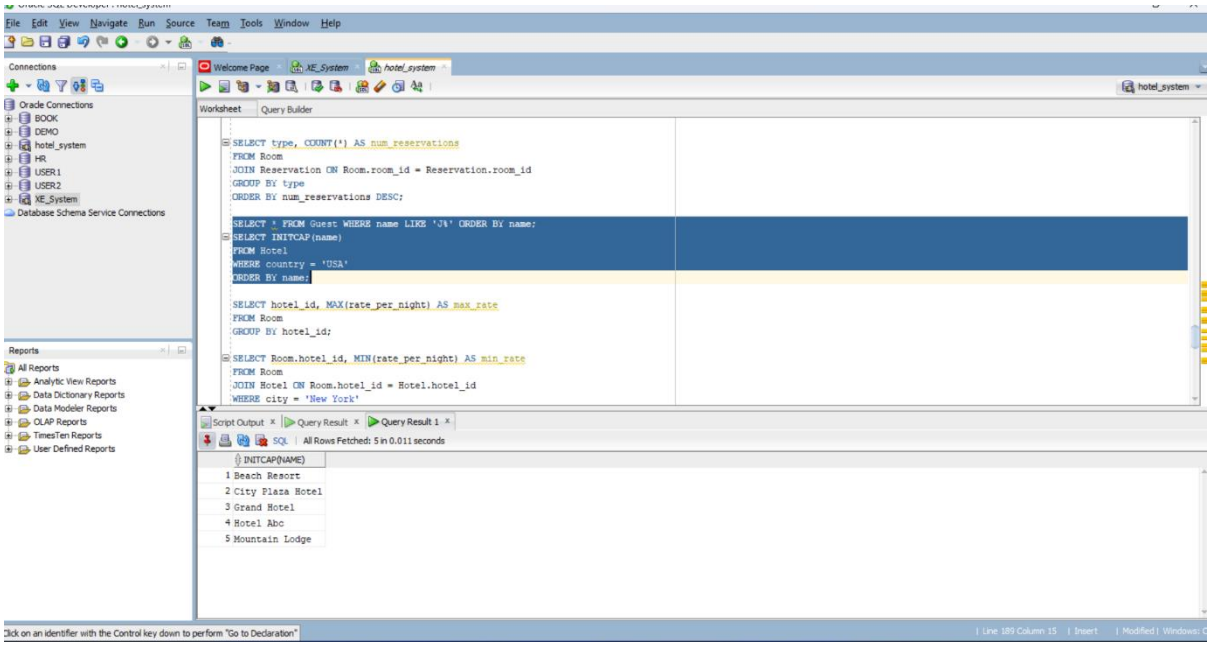


Figure 29:DQL Query

Retrieves hotel names in the USA with the first letter of each word capitalized, ordered by name.

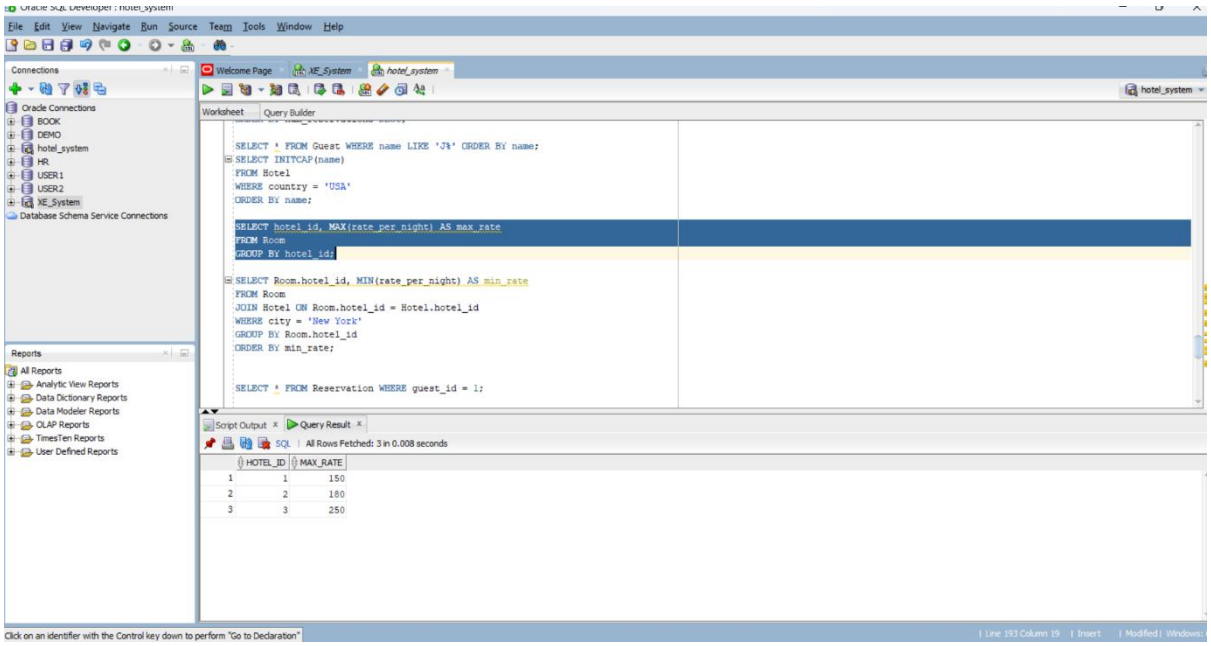


Figure 30:DQL Query

Retrieves the maximum rate per night for each hotel.

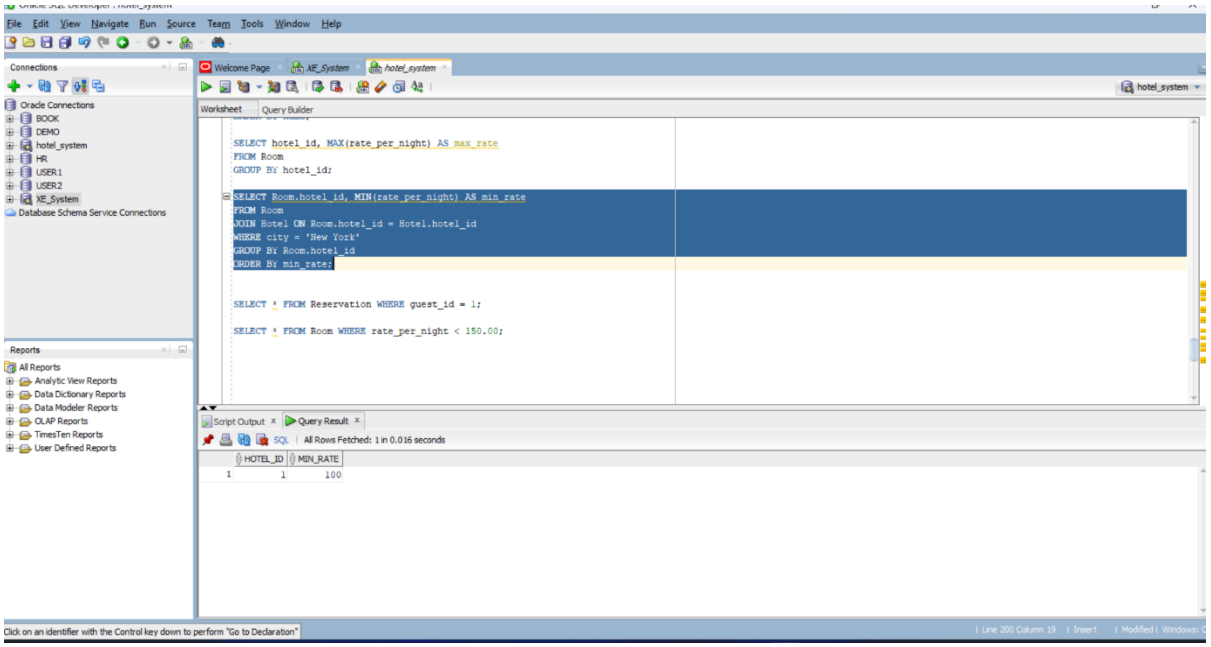


Figure 31:DQL Query

Retrieves the minimum rate per night for hotels in New York, ordered by the minimum rate.

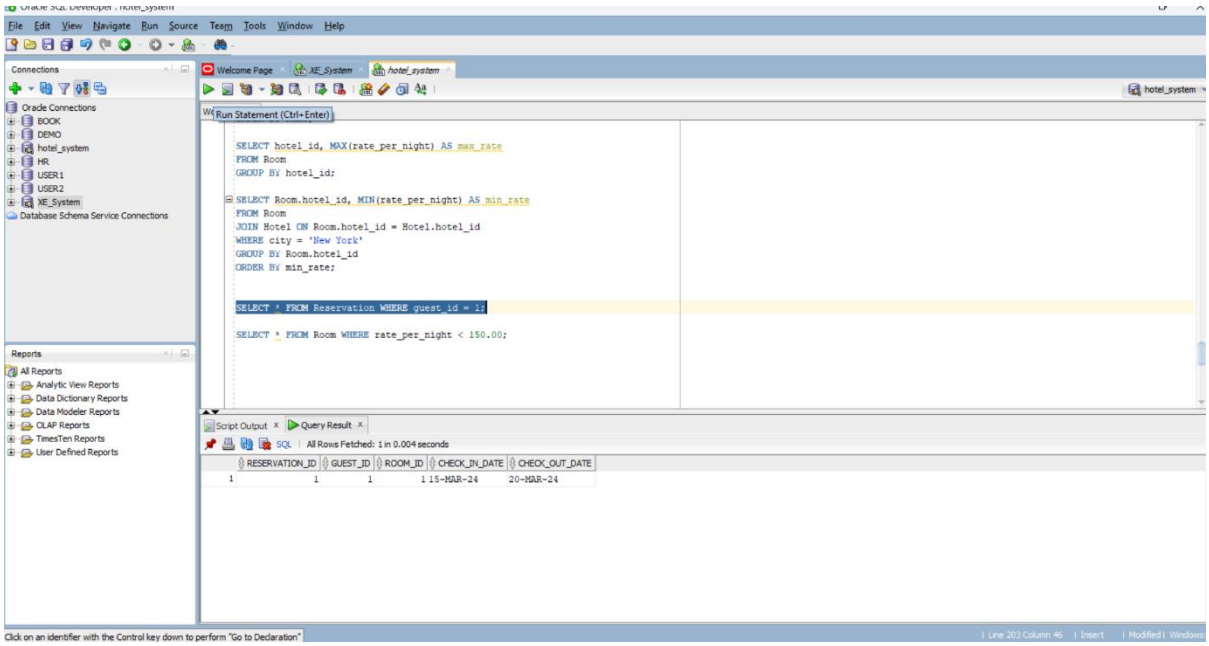


Figure 32:DQL Query

Retrieves reservations made by a specific guest (guest_id = 1).

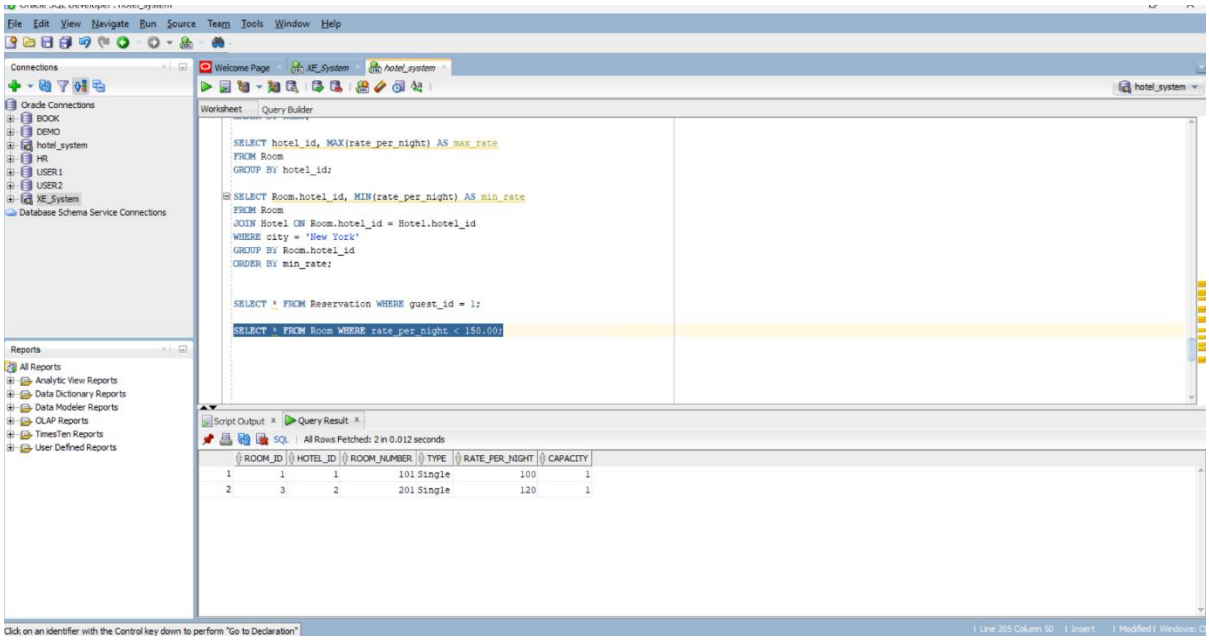


Figure 33:DQL Query

Retrieves rooms with a rate per night less than $150.00.

5) Write five PL/SQL programs to retrieve data based on the user given inputs.

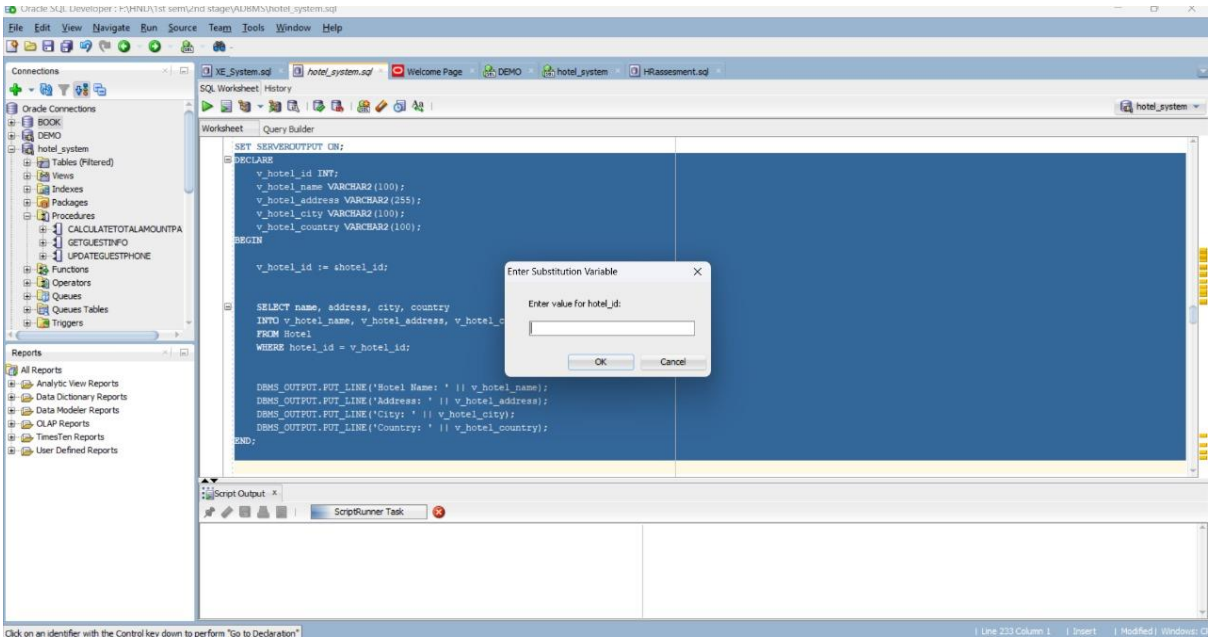


Figure 34:User Given Inputs

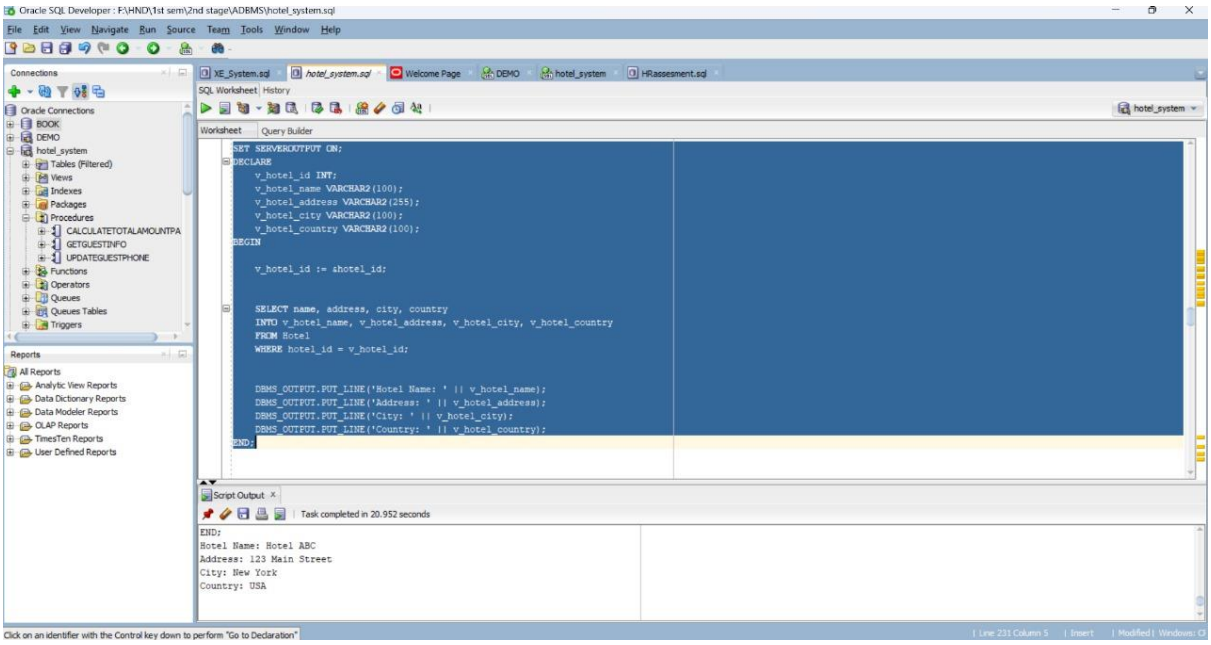


Figure 35:User Given Inputs

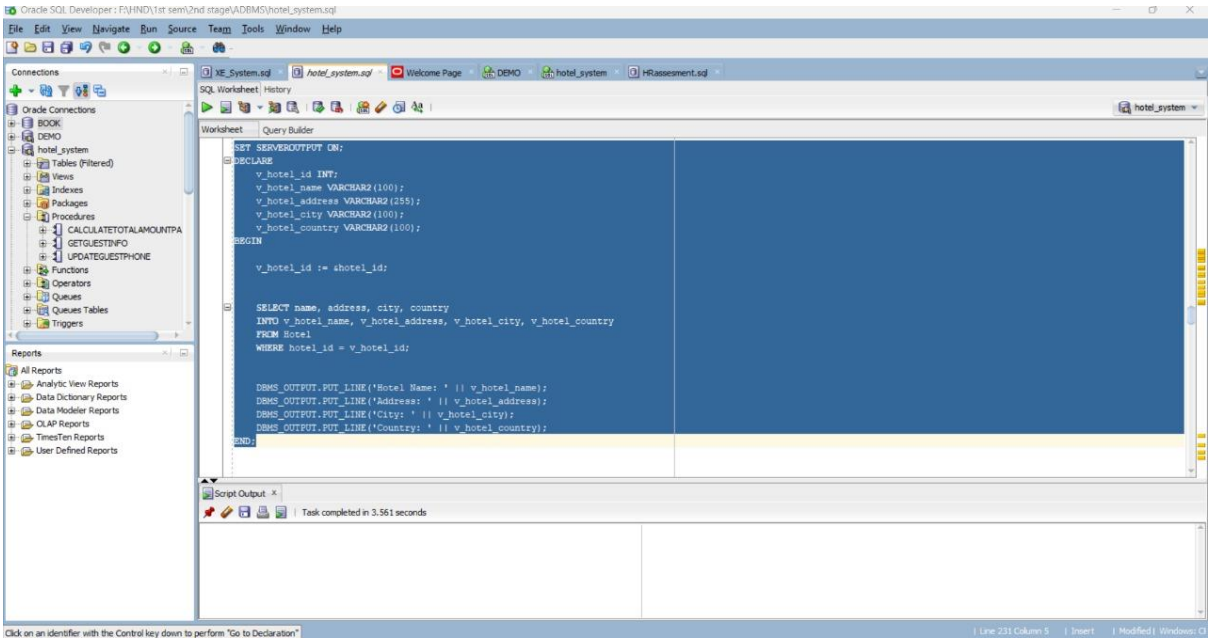


Figure 36:User Given Inputs

This will get the hotel id and the corresponding hotel name and address from the city and country database

……

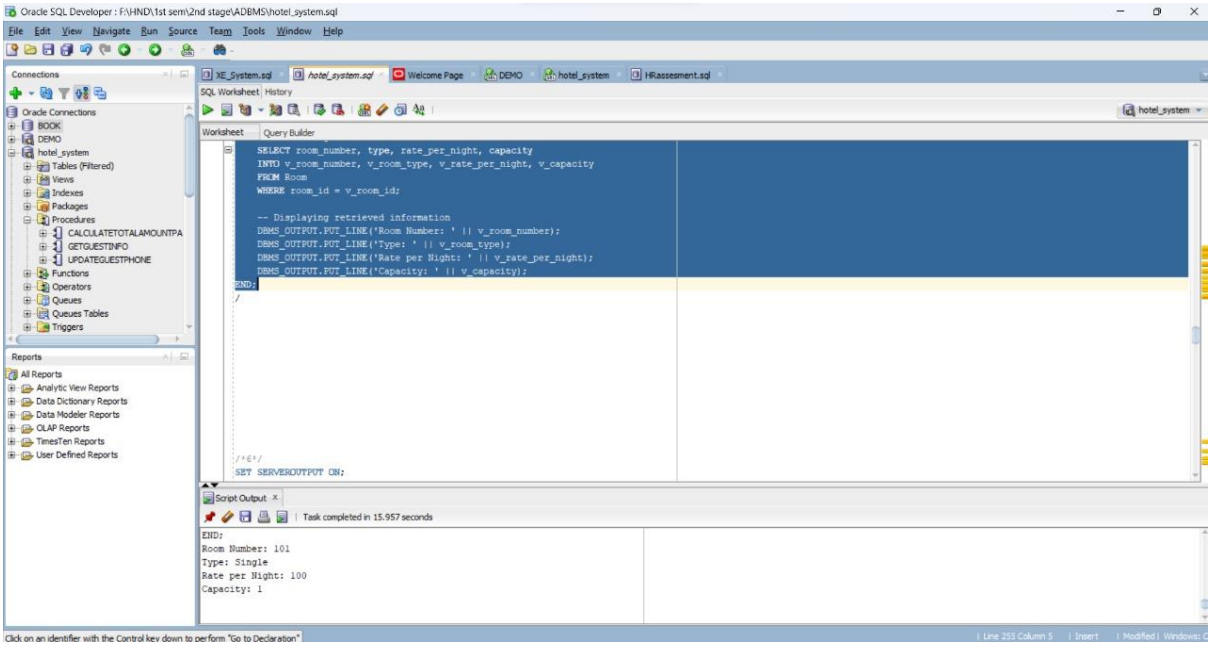


Figure 37:User Given Inputs

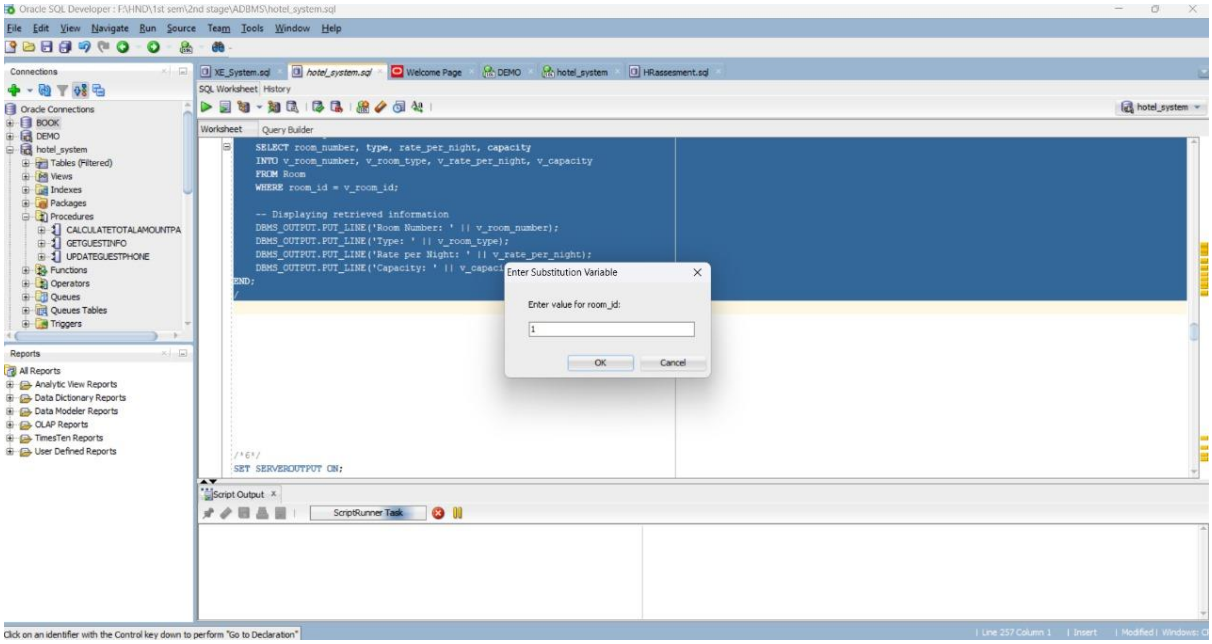


Figure 38:User Given Inputs

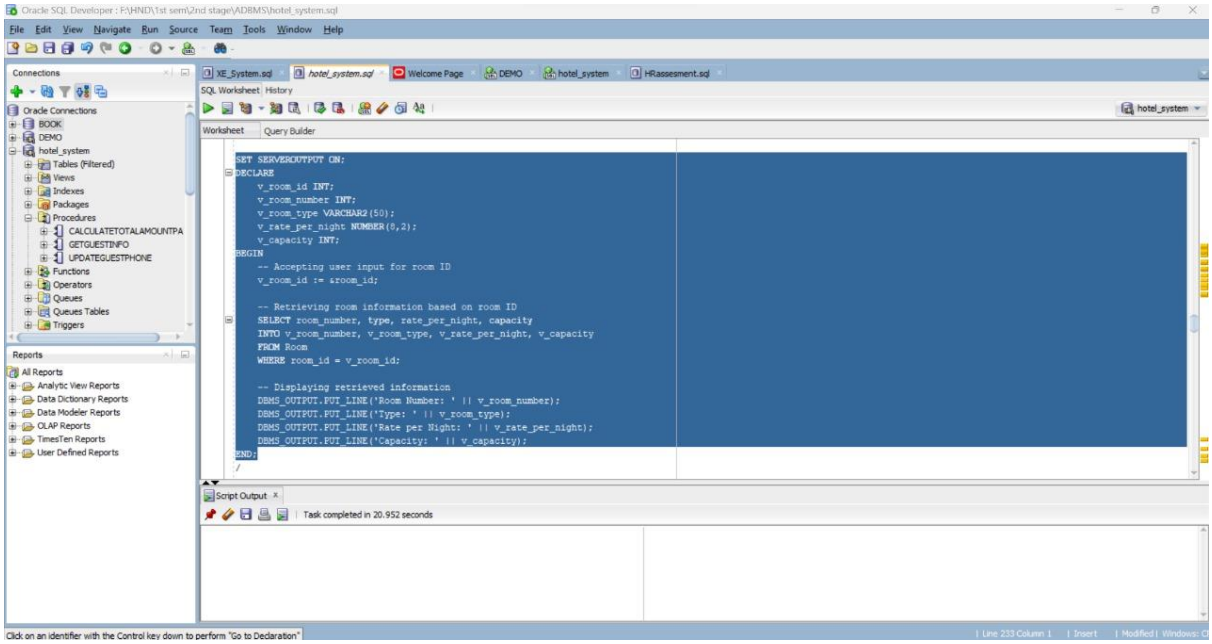


Figure 39:User Given Inputs

Here the guest input the room number and gets the details of the room related to that room number. The database displays the room number type rate per night and capacity.

……..

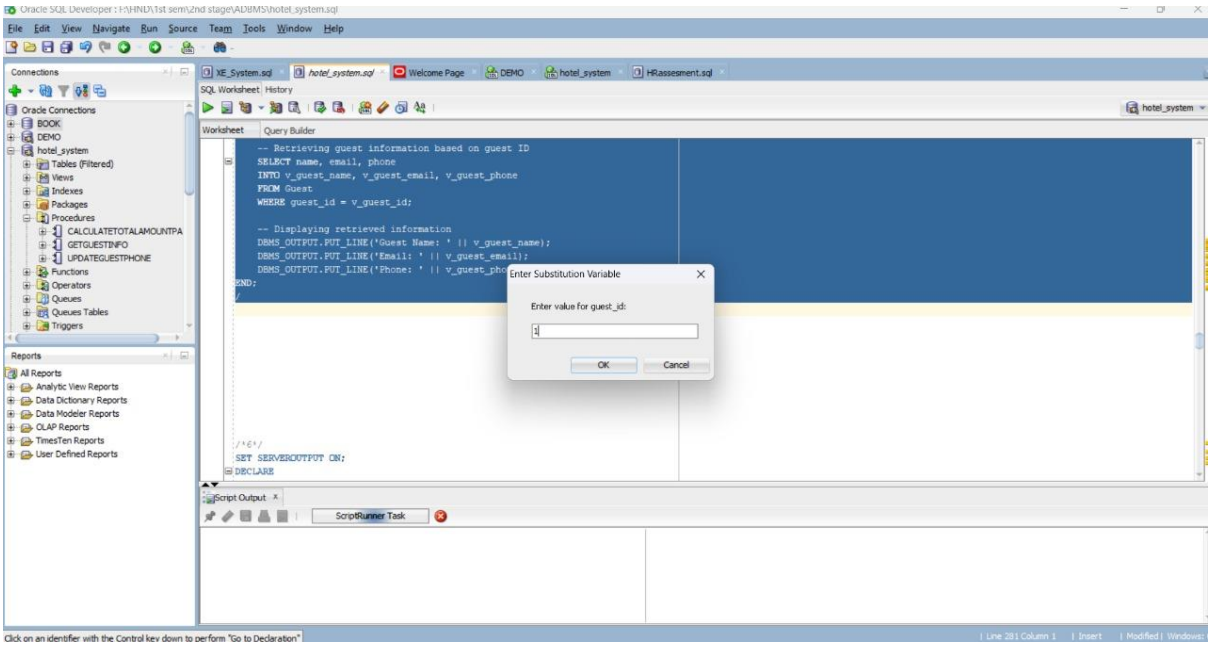


Figure 40:User Given Inputs

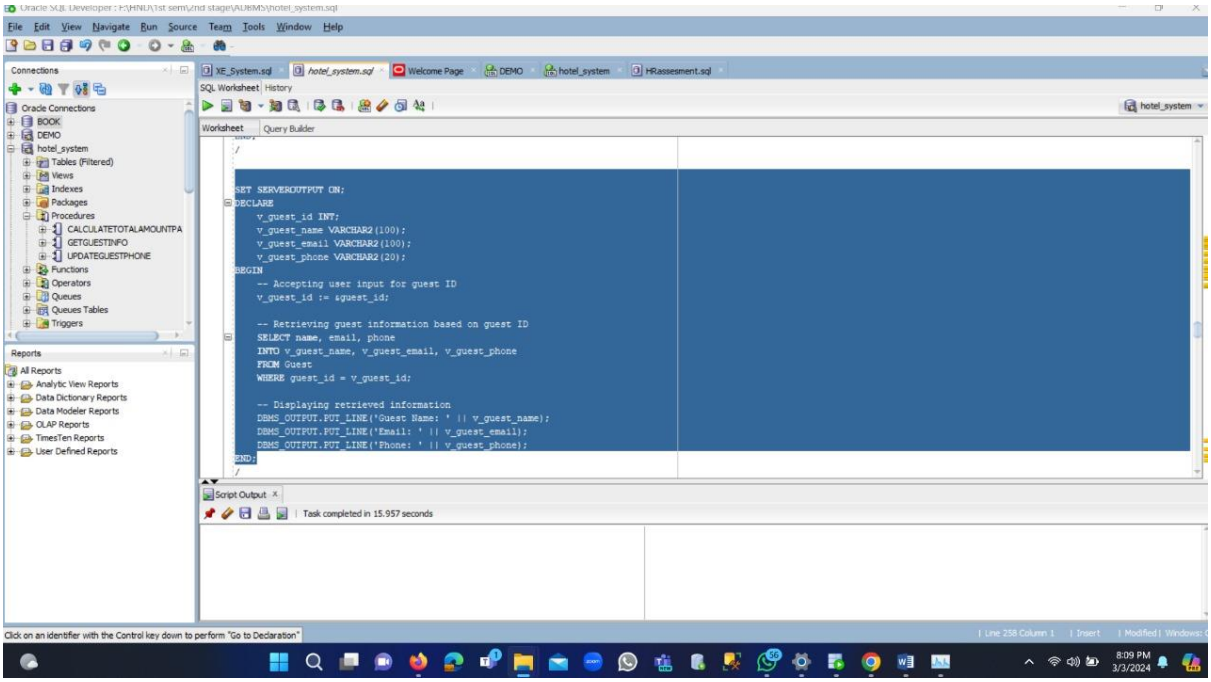


Figure 41:User Given Inputs

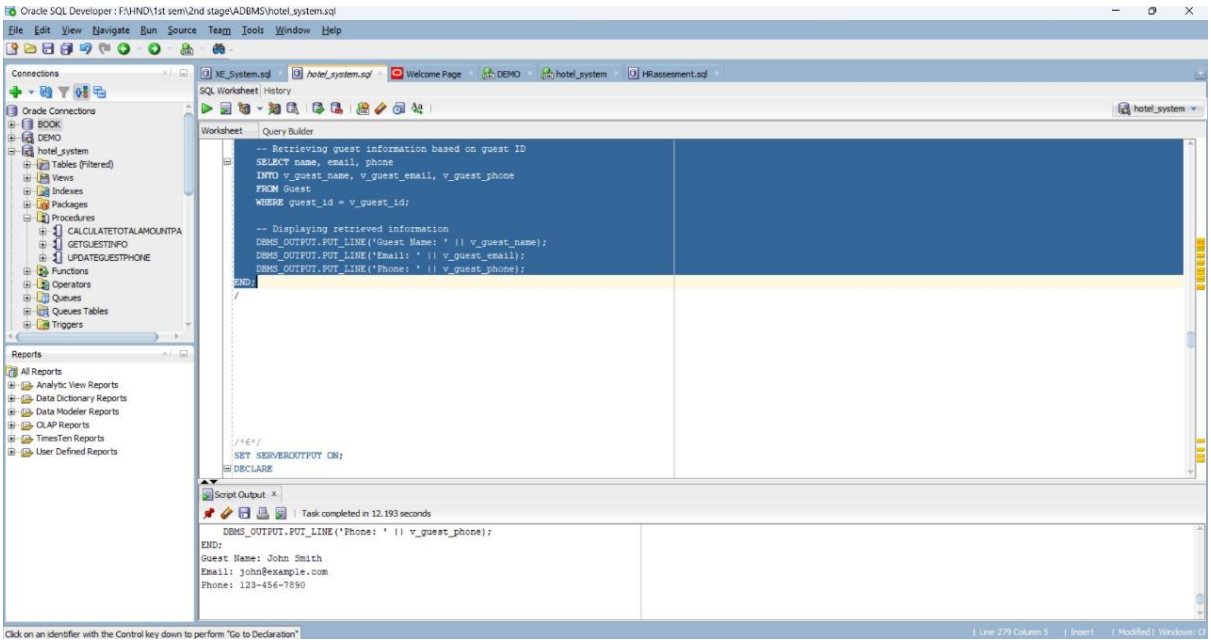


Figure 42:User Given Inputs

This one first input the guest id and information of the guest related to that id is given.it includes guest name,email and phone number.

……

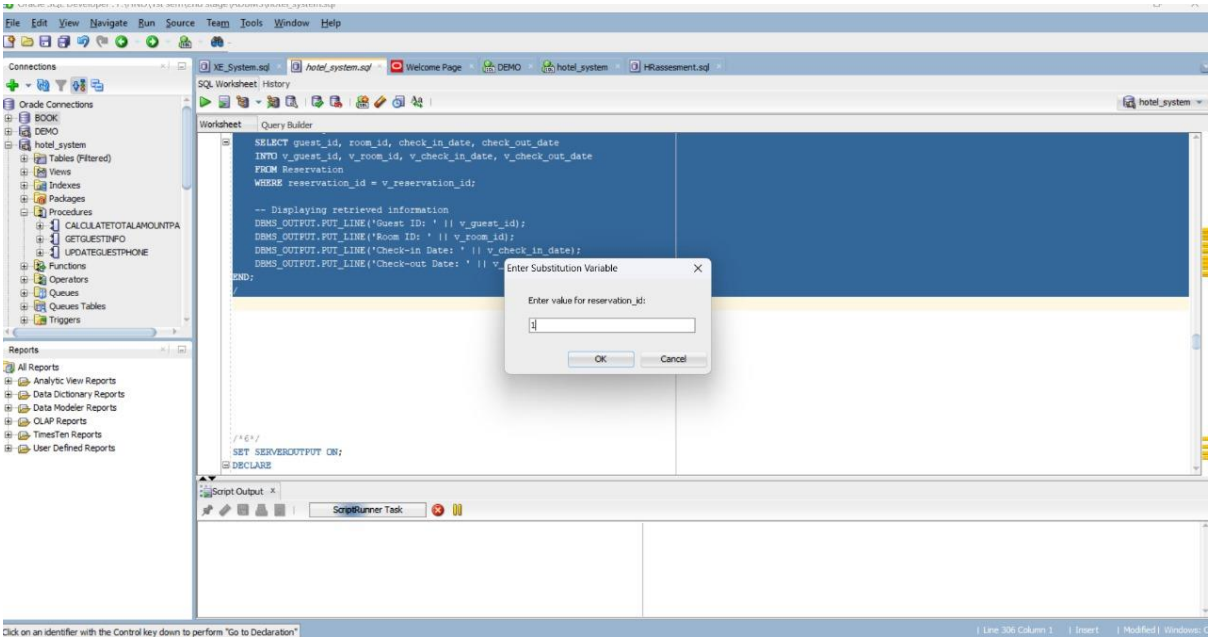


Figure 43:User Given Inputs

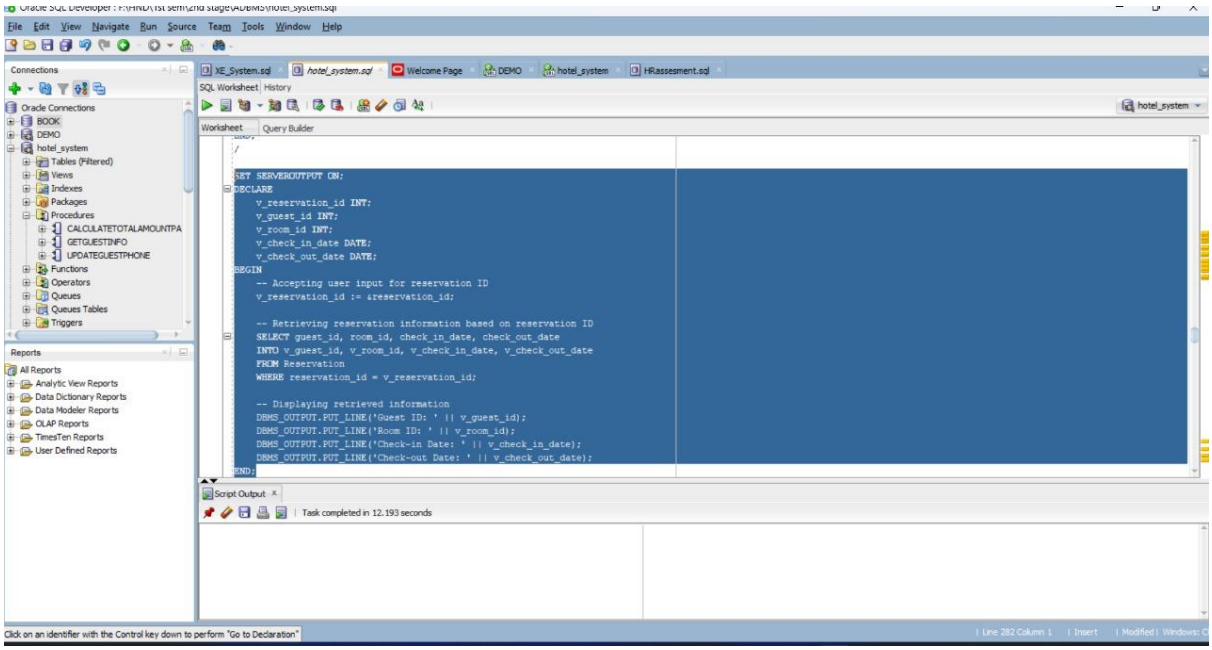


Figure 44:User Given Inputs

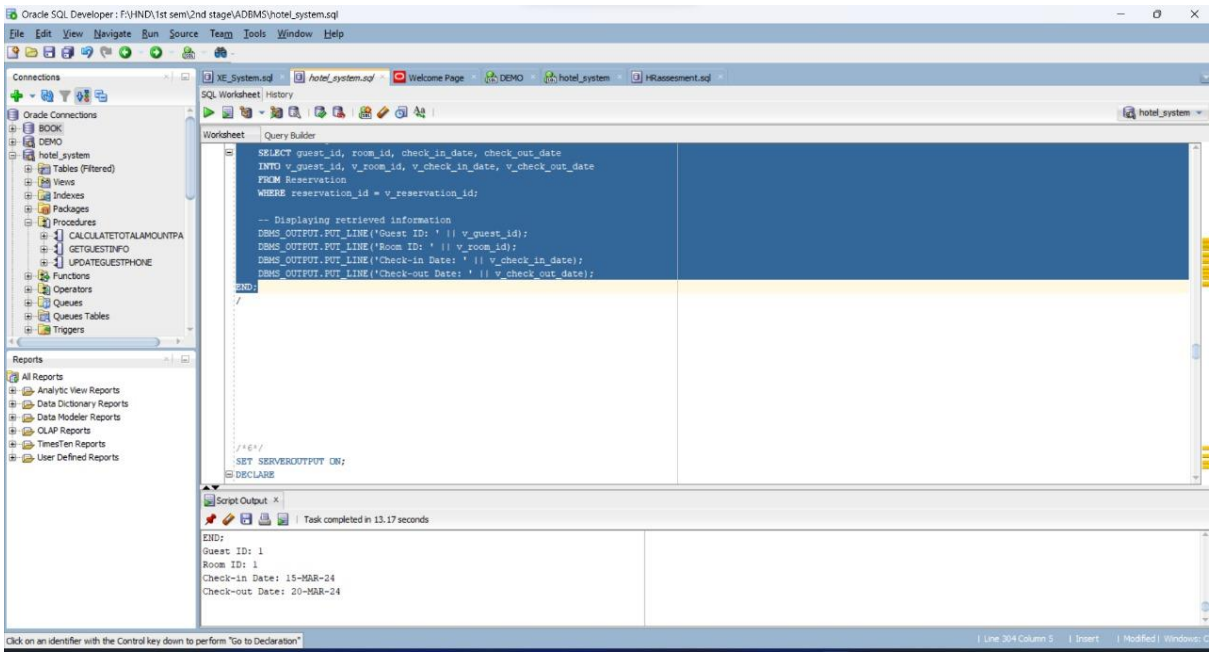


Figure 45:User Given Inputs

Here the booking is done using the relevant reservation id and the details related to the reservation are obtained. From the database, guest id, room id, check in date, check out date are obtained and displayed

……

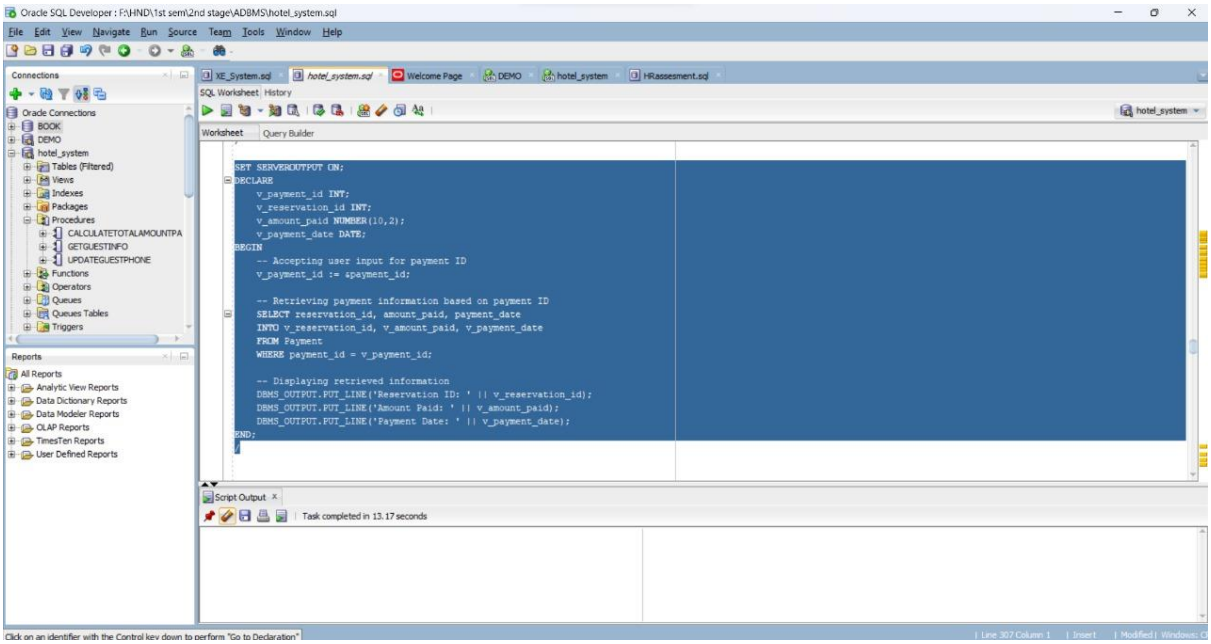


Figure 46:User Given Inputs

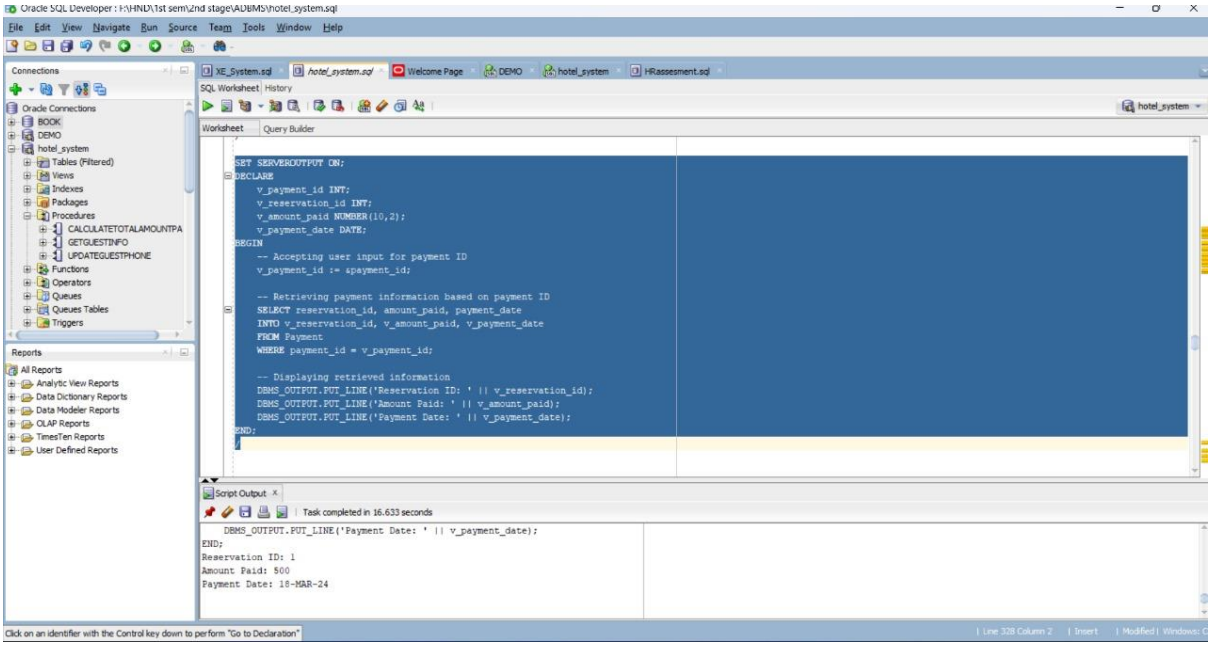


Figure 47:User Given Inputs

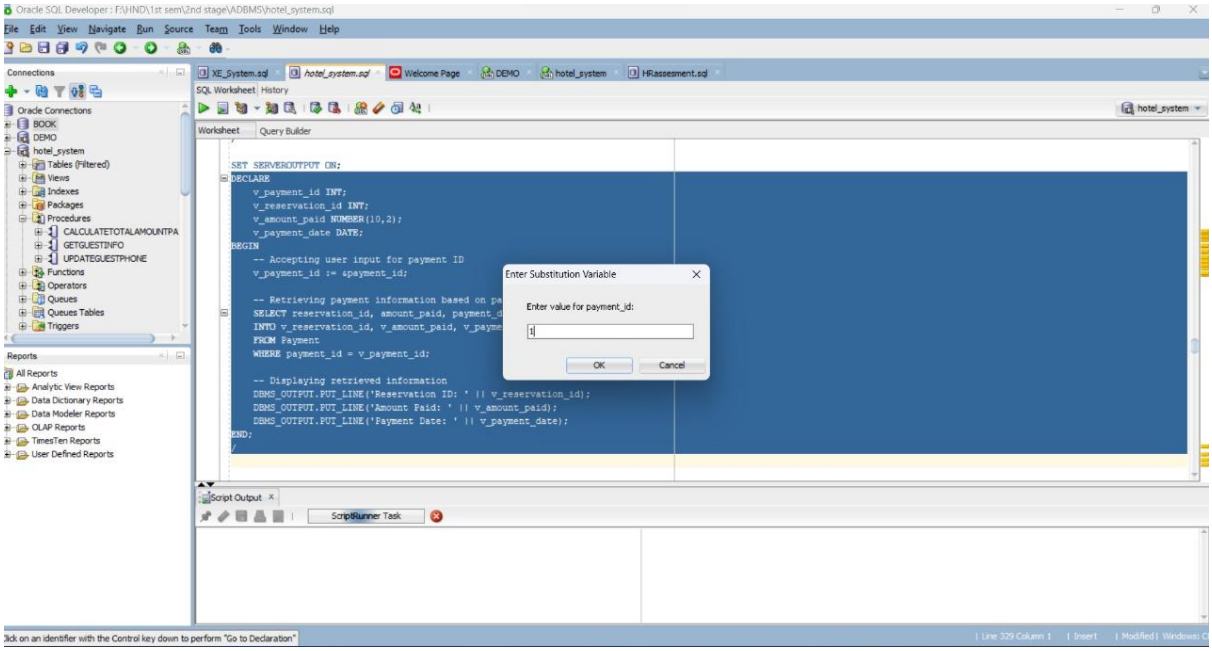


Figure 48:User Given Inputs

This code purpose is payment details based on the provided payment id.it include reservation id,amount paid and payment date from the database and display them.

…….

6) Write PL/SQL programs to cover following control structures. • If/Else • Case • For Loop • While Loop

• If/Else

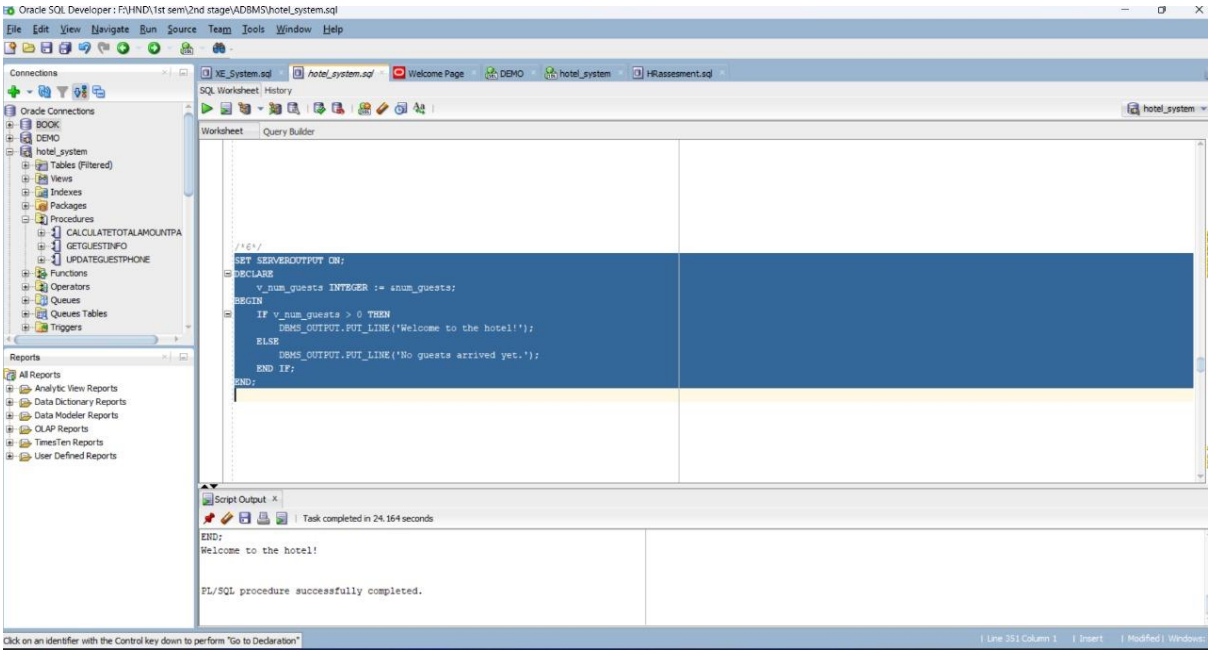


Figure 49:IF ELSE

Check the number of visitors and if it is more than 0 then show a welcome message and otherwise show another message.

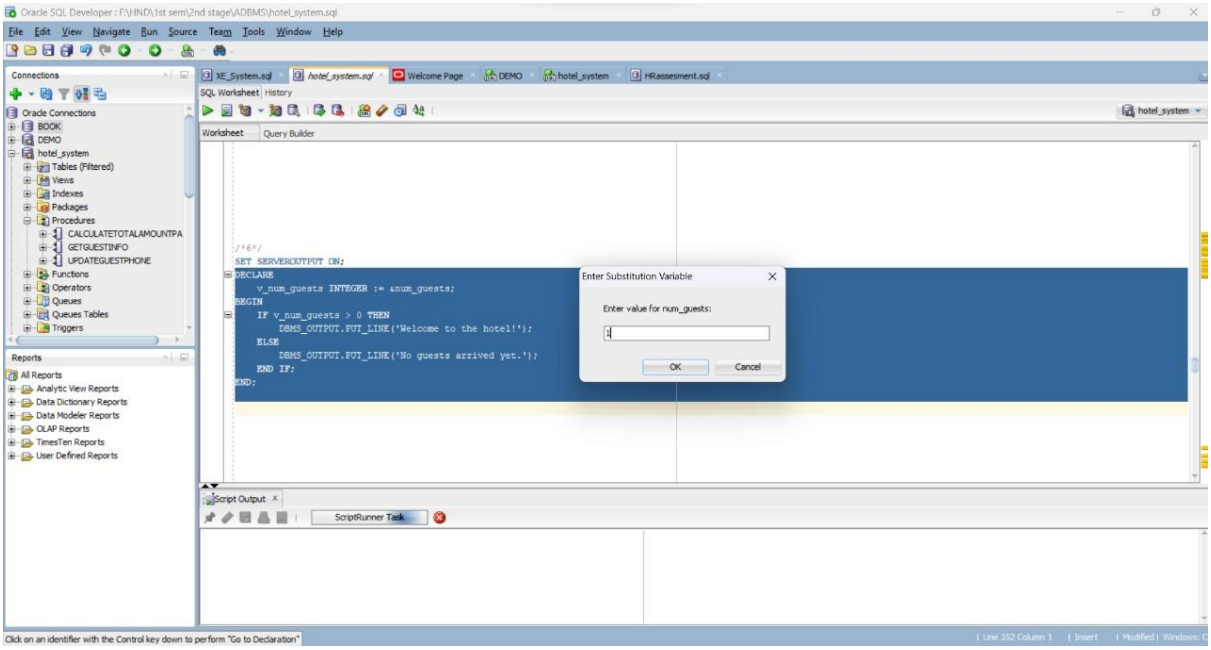


Figure 50:IF ELSE

• Case

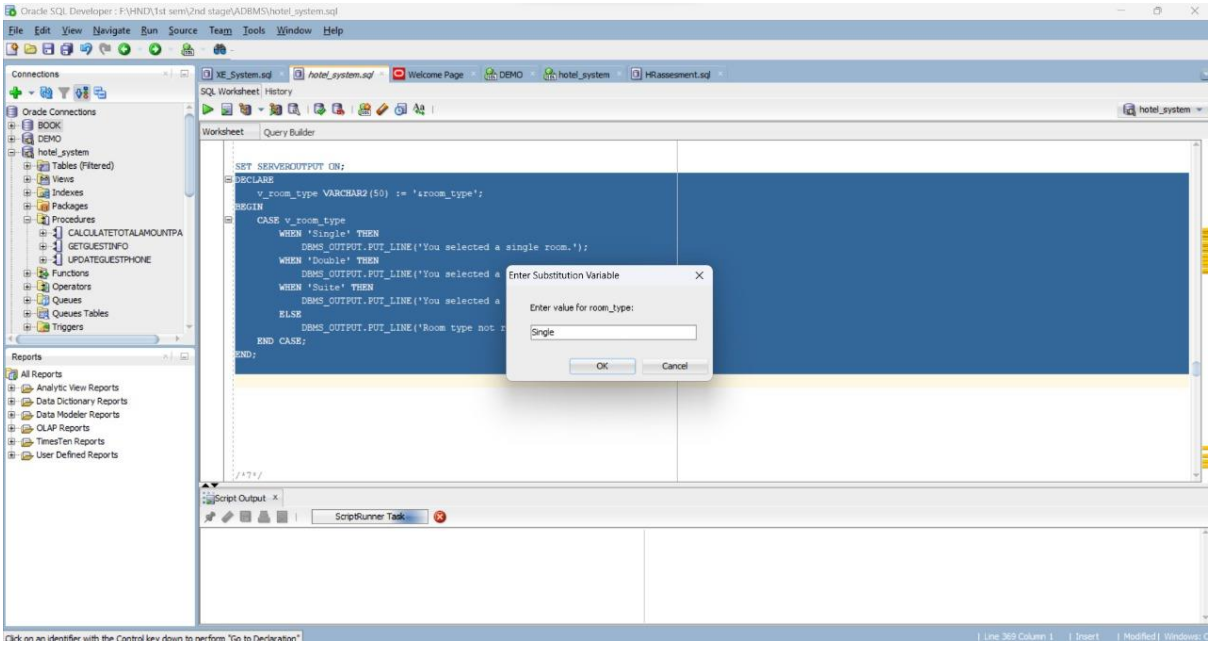


Figure 51:Case

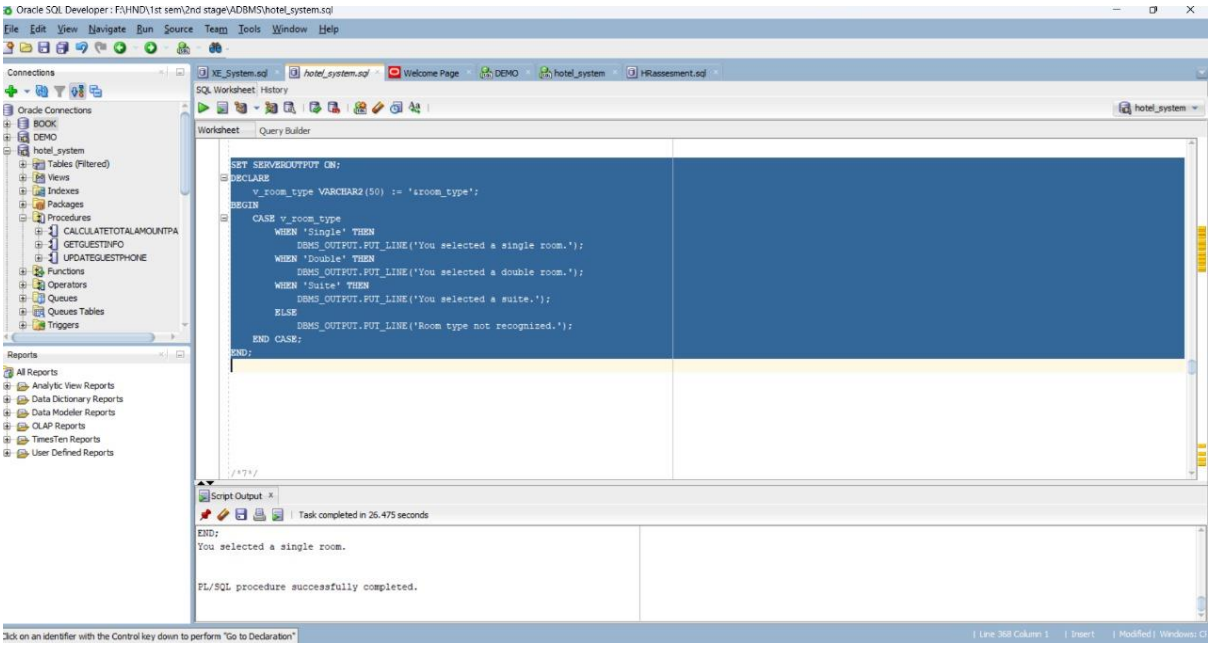


Figure 52:Case

This code purpose is a message indicating the room type is displayed related to what the guest input

• For Loop

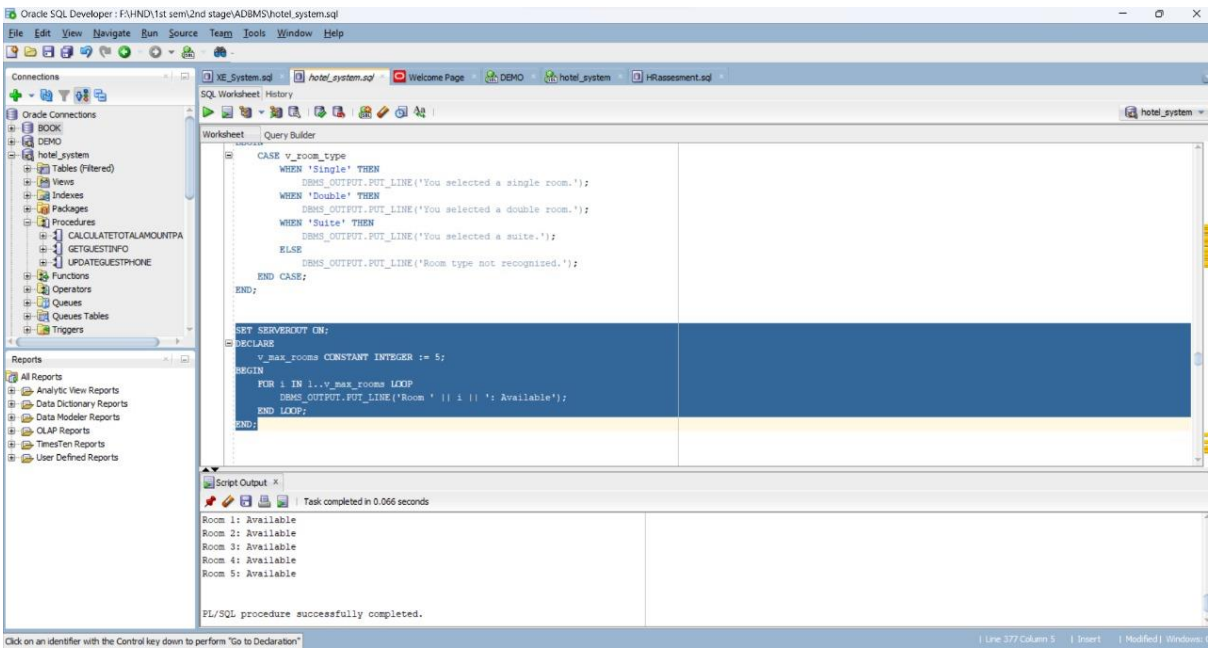


Figure 53:For Loop

This code purpose the room number indicates their availability. Here a for loop is used and the maximum number of rooms is assumed to be 5.

While Loop

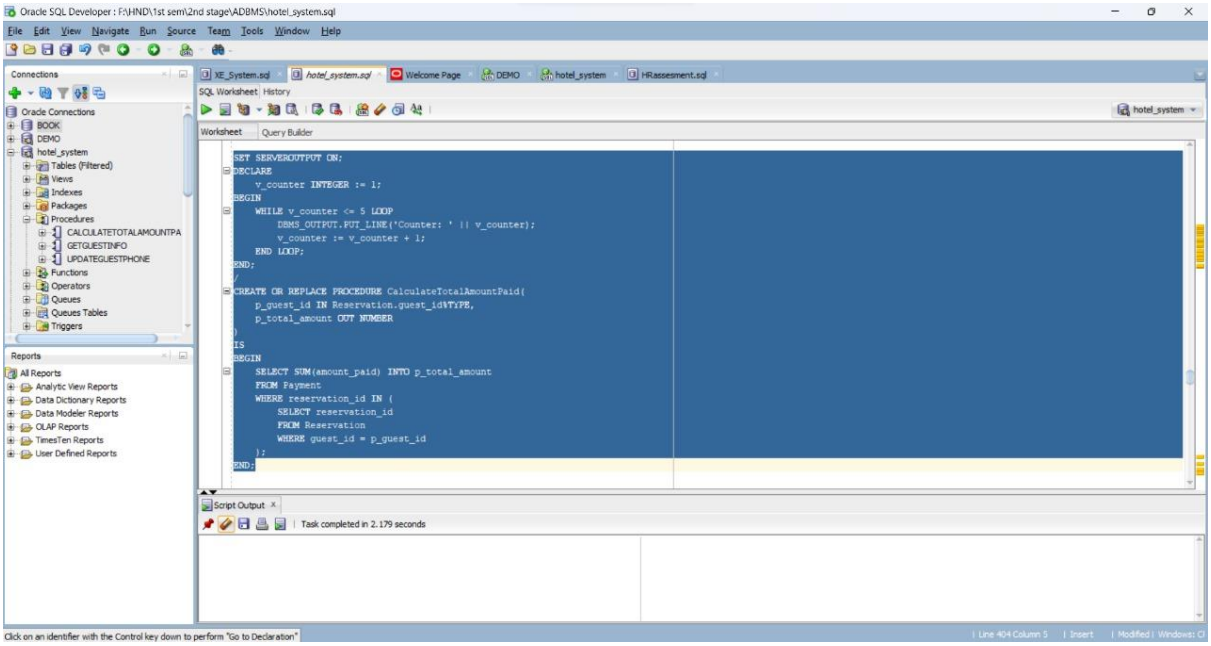


Figure 54:While Loop

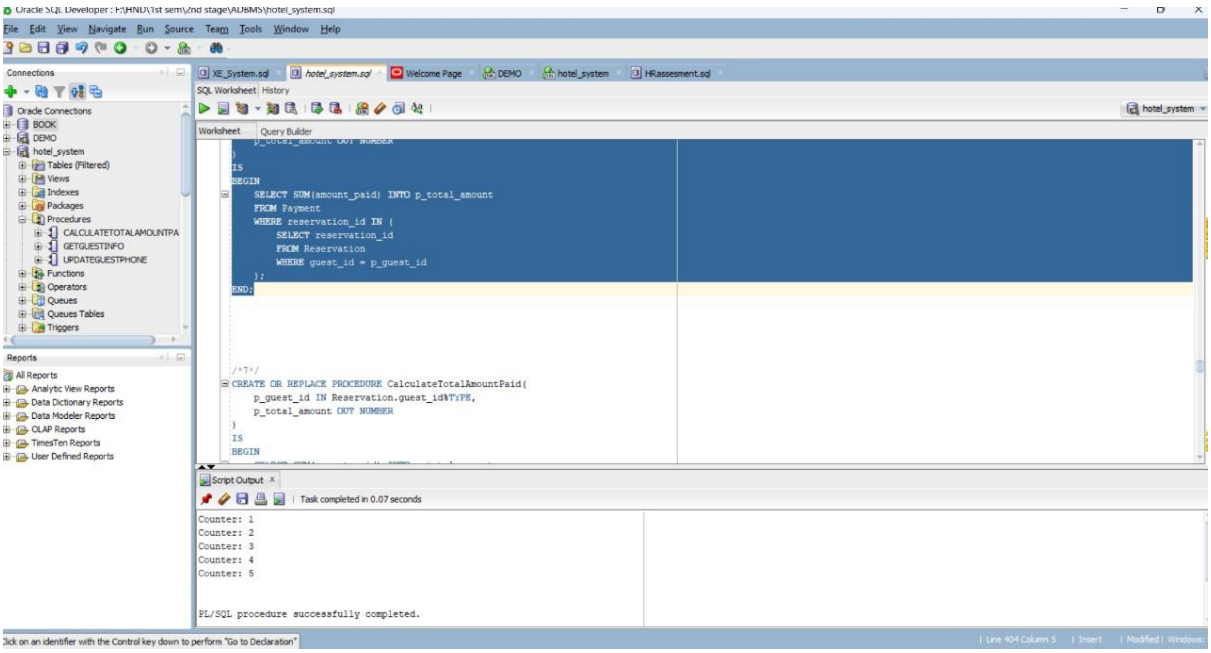


Figure 55:While Loop

This code shows a while loop that increments from 1 to 5 and displays the current value of the counter at each iteration.

7) Create three PL/SQL procedures with in and out parameters.

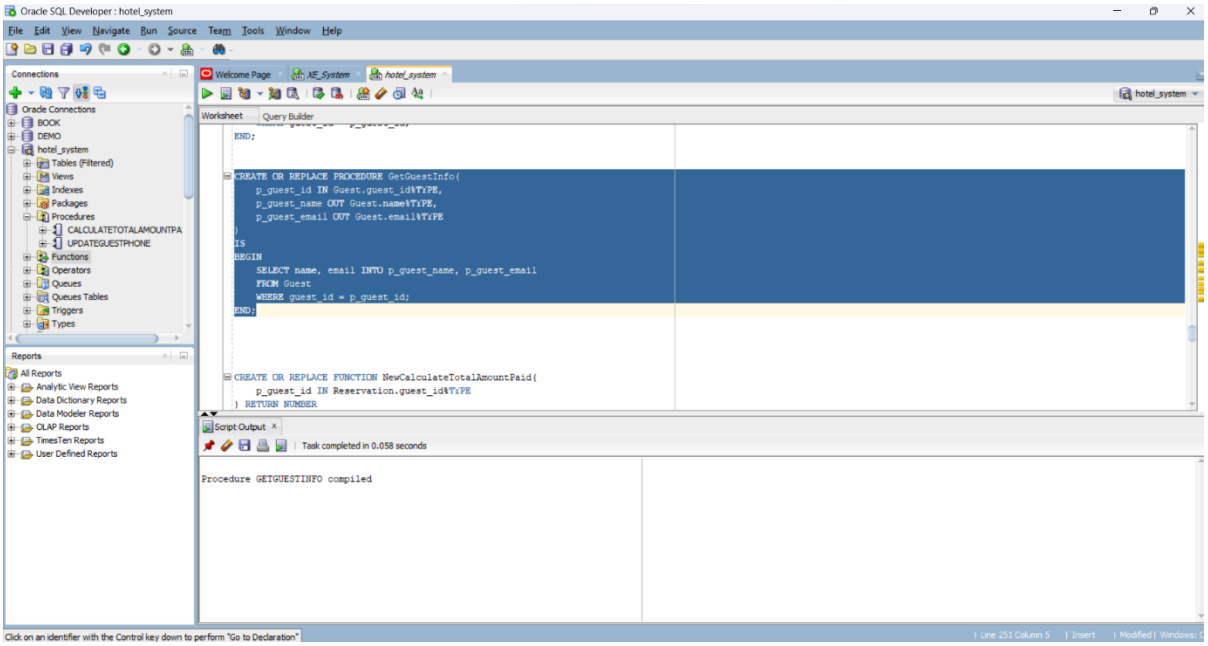


Figure 56:Create Procedure

GetGuestInfo

This procedure retrieves the guest name and guest email based on their guest id.get guest id its inout and return the guest name and email as output parameters.

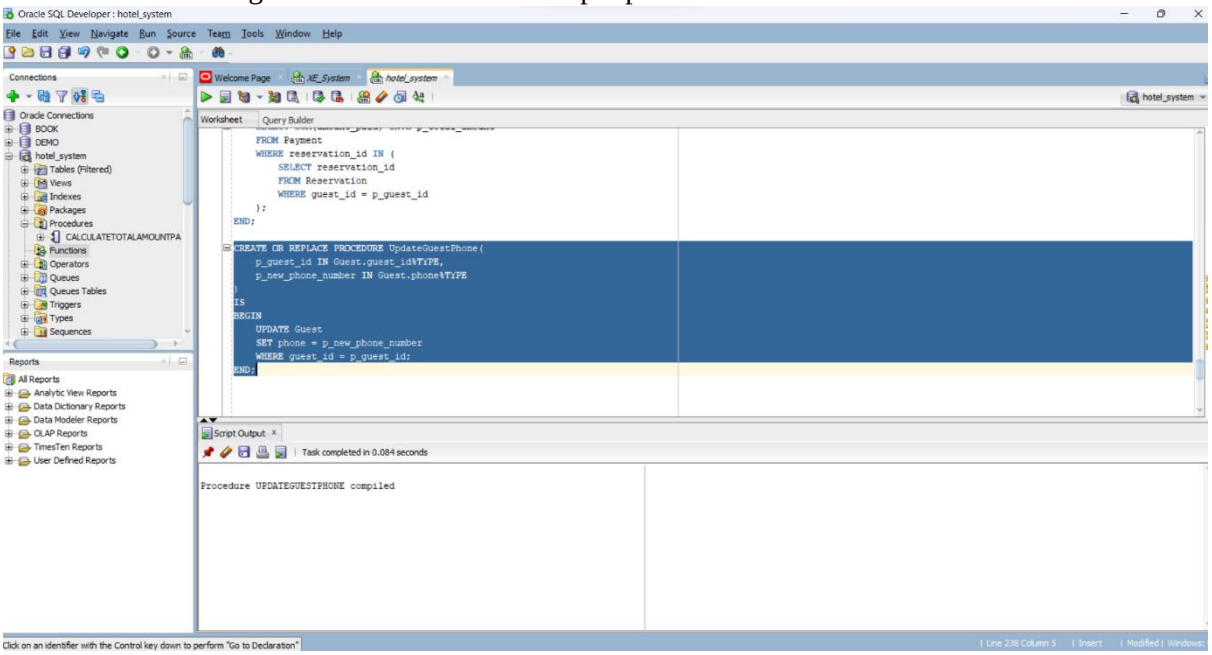


Figure 57:Create Procedure

UpdateGuestPhone

This procedure update the guest phone number based on their guest id.it take guest id and set the new phone number.new phone number is input parameter and update phone number in guest table

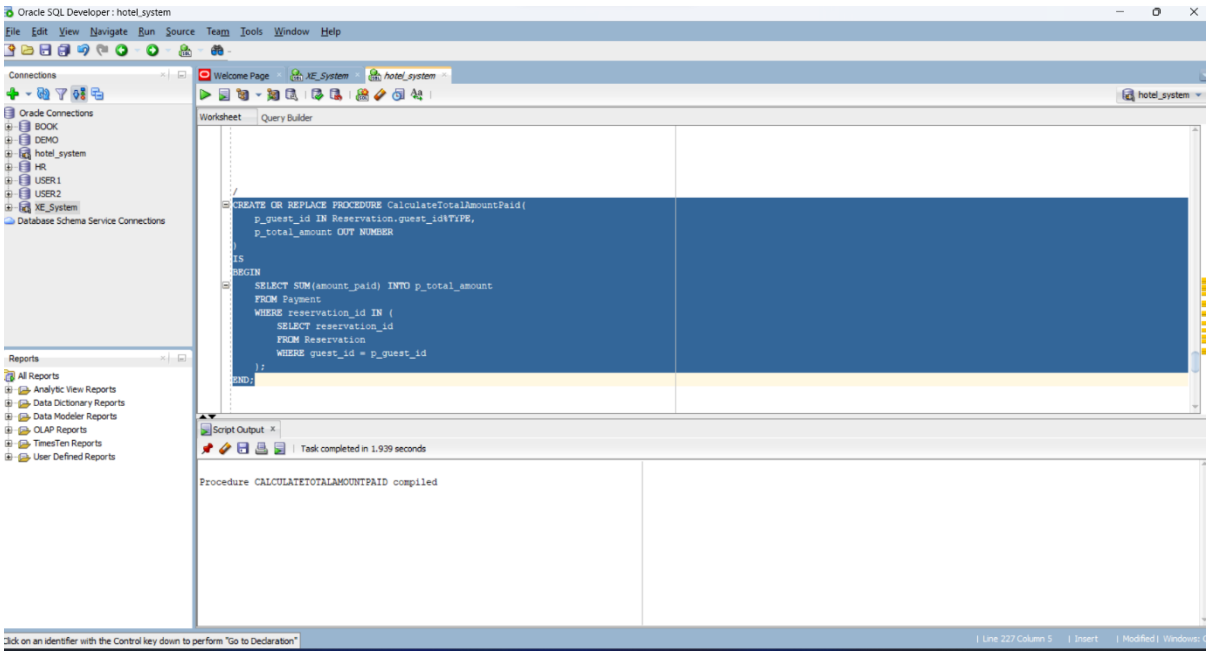


Figure 58:Create Procedure

CalculateTotalAmountPaid

This procedure calculate amount paid by a guest is calculated based on his guest id. it takes the guest id as input and res thetital amount paid as an output parameter.

7) Create three PL/SQL Function.

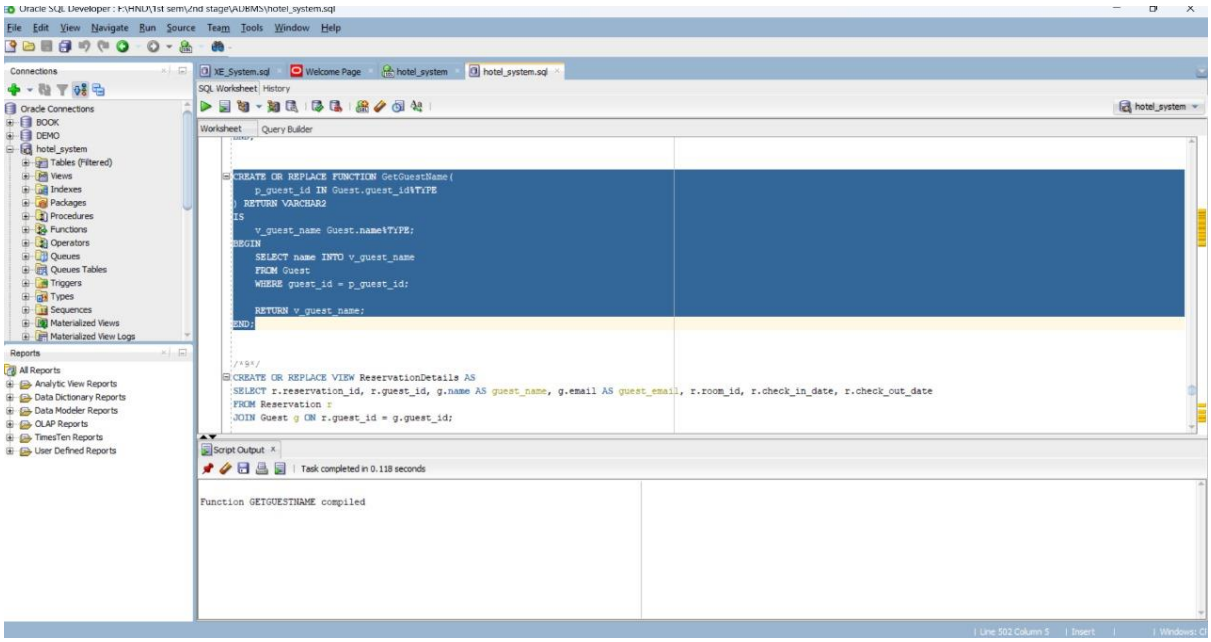


Figure 59:Function

GetGuestName

This function retrieve the The guest name is retrieved based on that guest's ID

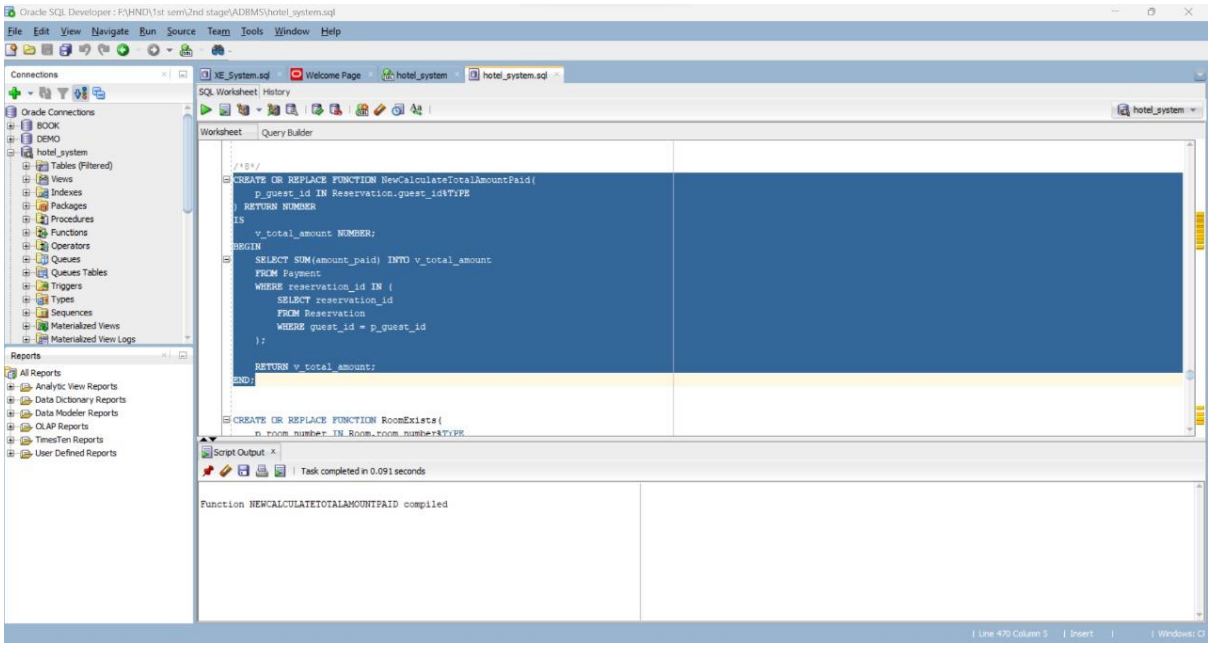


Figure 60:Function

NewCalculateTotalAmountPaid

This function calculate total amount paid by a guest is calculated based on his guest id.

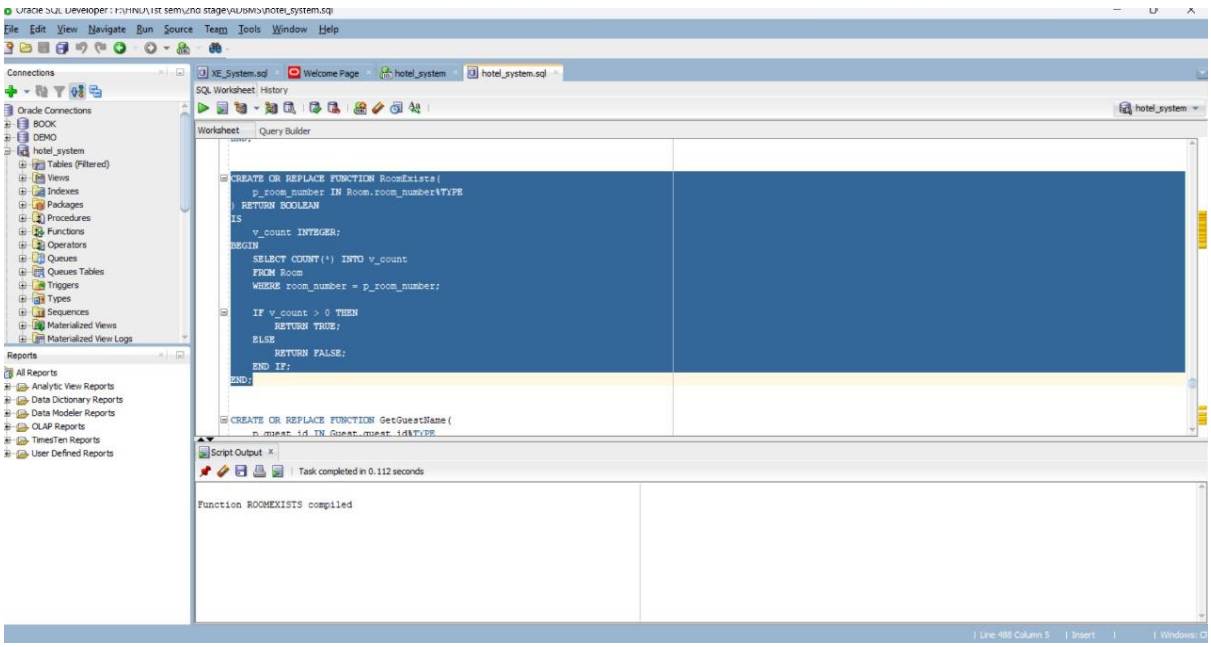


Figure 61:Function

RoomExists

This function Checks for a room with a unique room number and Returns true if present and false otherwise

8) Create three PL/SQL view.

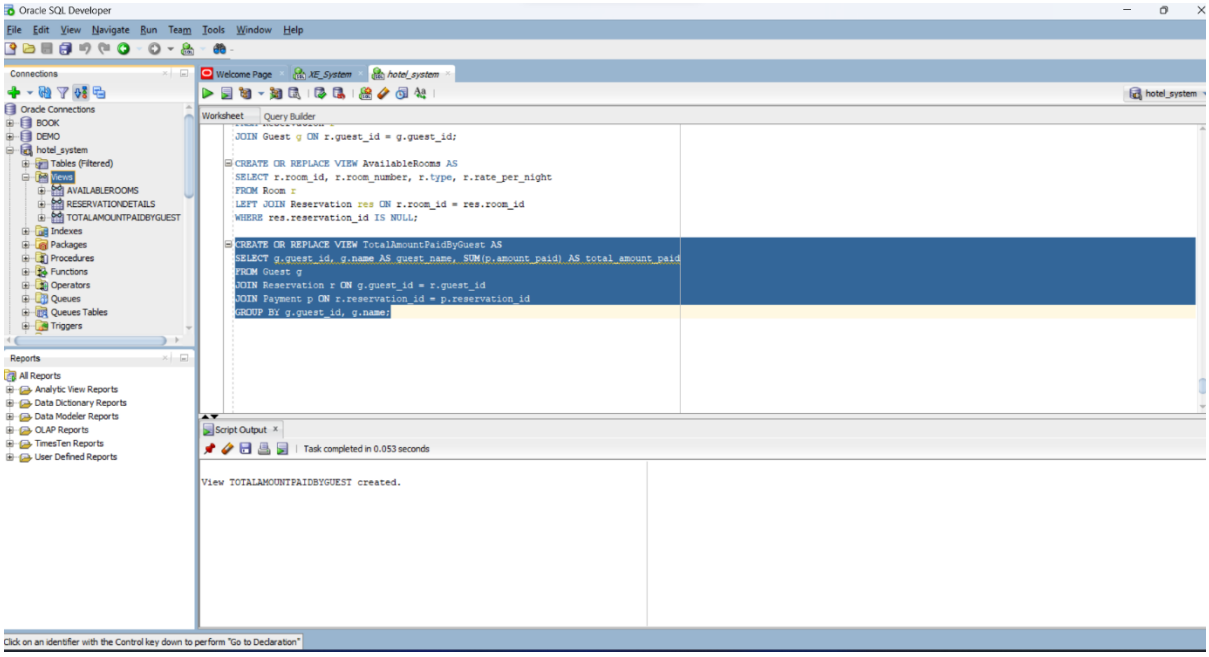


Figure 62:View

This view shows us summary of the amount paid by each visitor is displayed . it includes guest id,guest name and the total amount they have paid across all their reservation

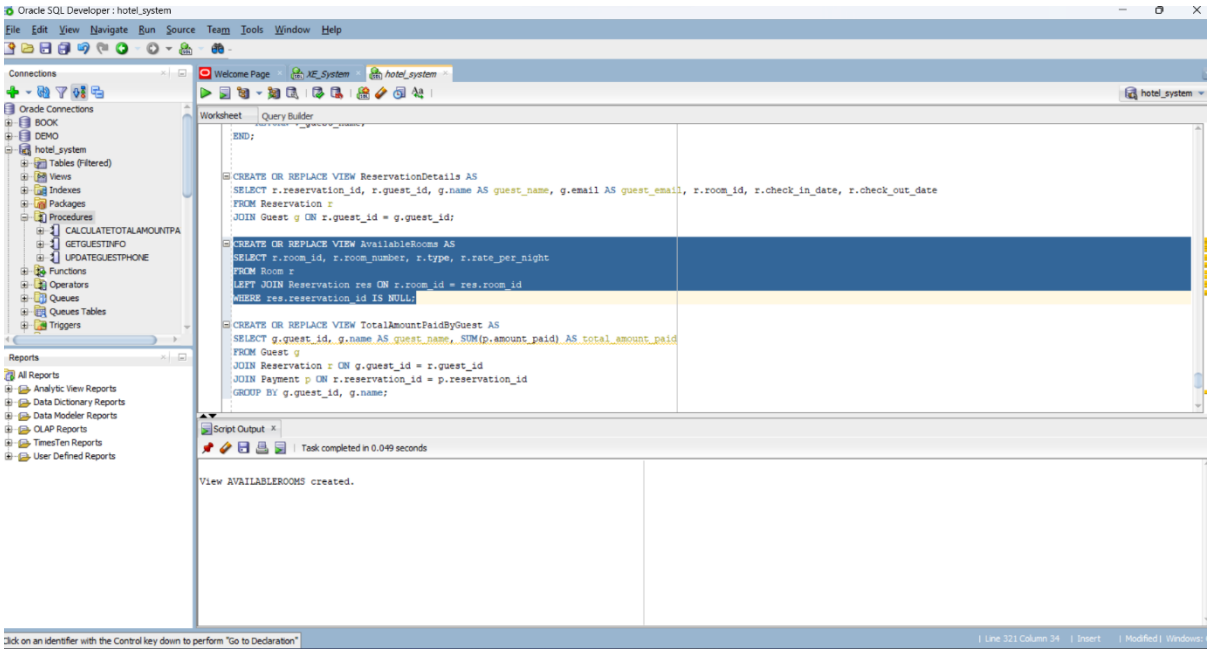


Figure 63:View

This view retrieves information about rooms that are currently available for reservation. It selects columns such as room ID, room number, room type, and rate per night from the "Room" table. It utilizes a LEFT JOIN operation with the "Reservation" table on the room ID, filtering out rooms that have no associated reservations.

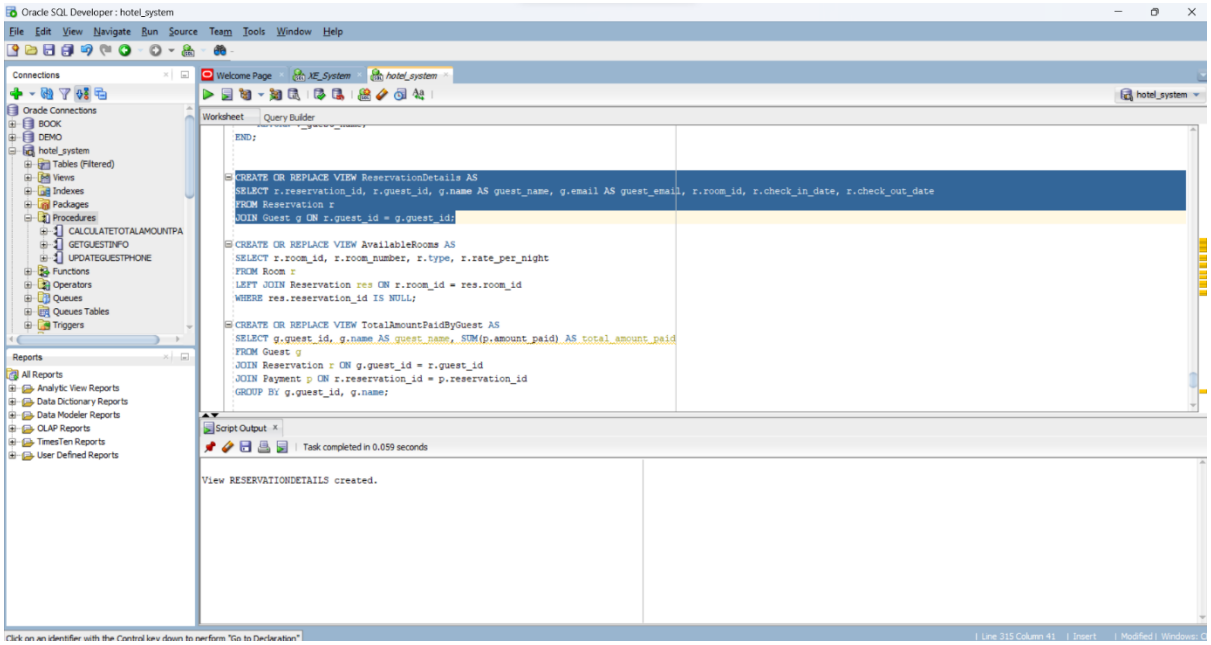


Figure 64:View

This Oracle SQL code creates or replaces a view named "ReservationDetails". This view retrieves reservation details along with guest information. It includes columns such as reservation ID, guest ID, guest name, guest email, room ID, check-in date, and check-out date. The view is created by joining the "Reservation" table with the "Guest" table based on the guest ID.

9) Create three PL/SQL materialized view.

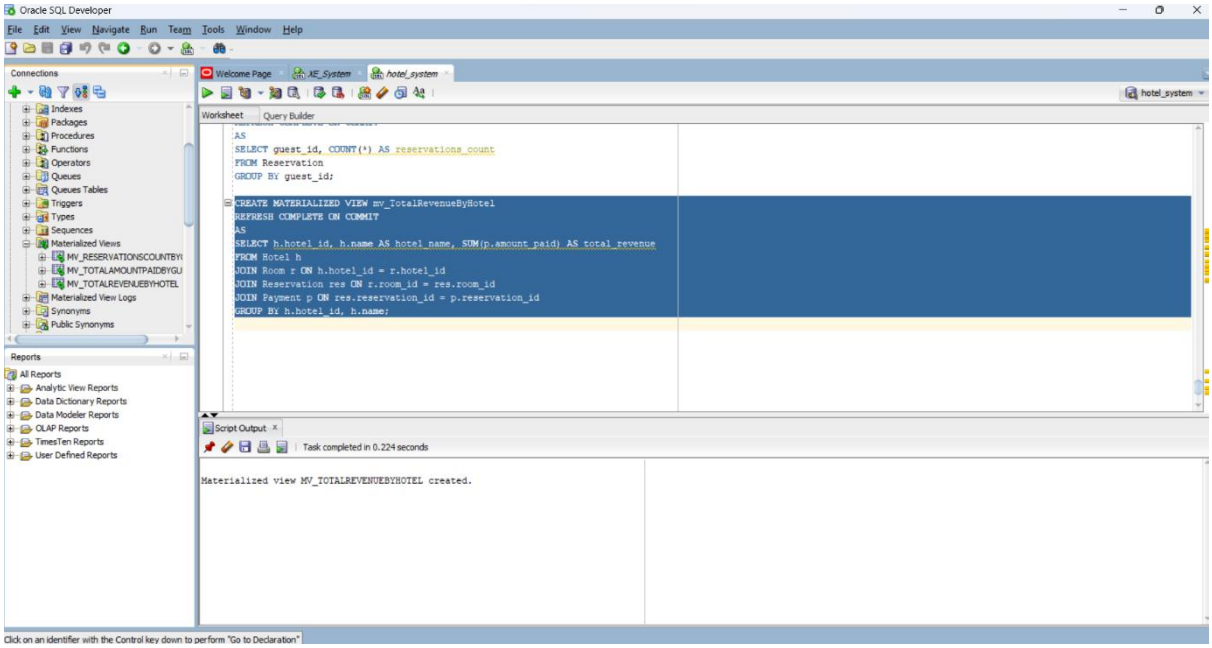


Figure 65:materialized view

mv_TotalRevenueByHotel

Stores the total revenue earned by each hotel by summing up payments for hotel rooms

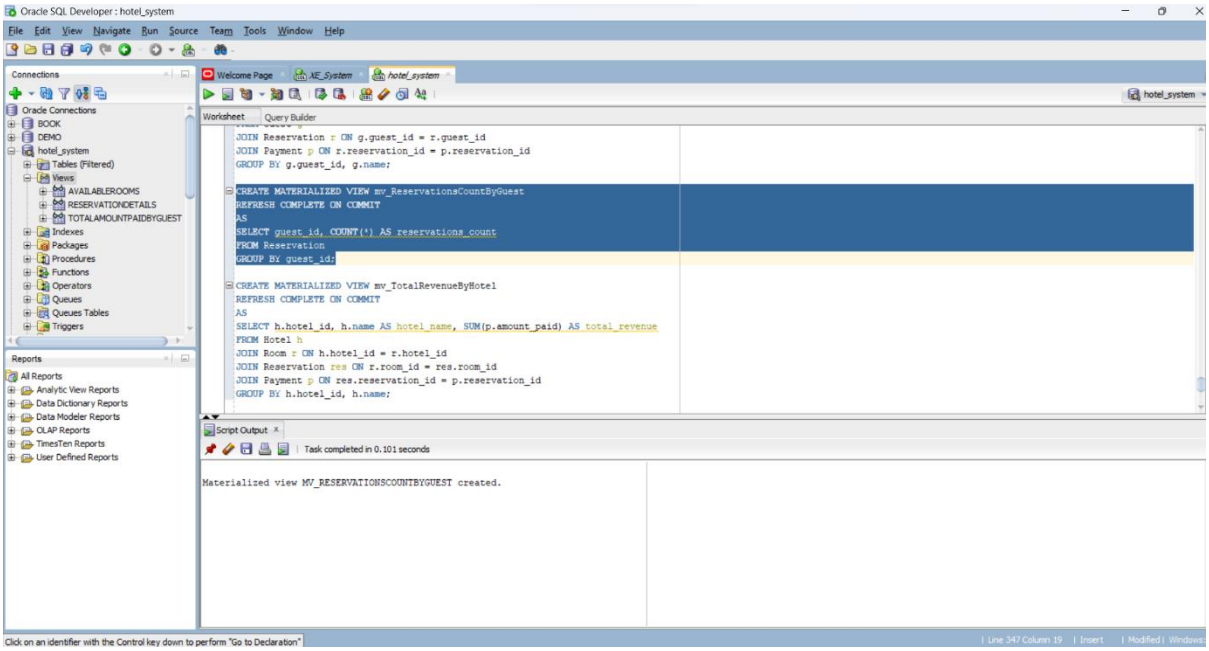


Figure 66:materialized view

mv_ReservationCounitByGuest

Guest store their respective reservation count here

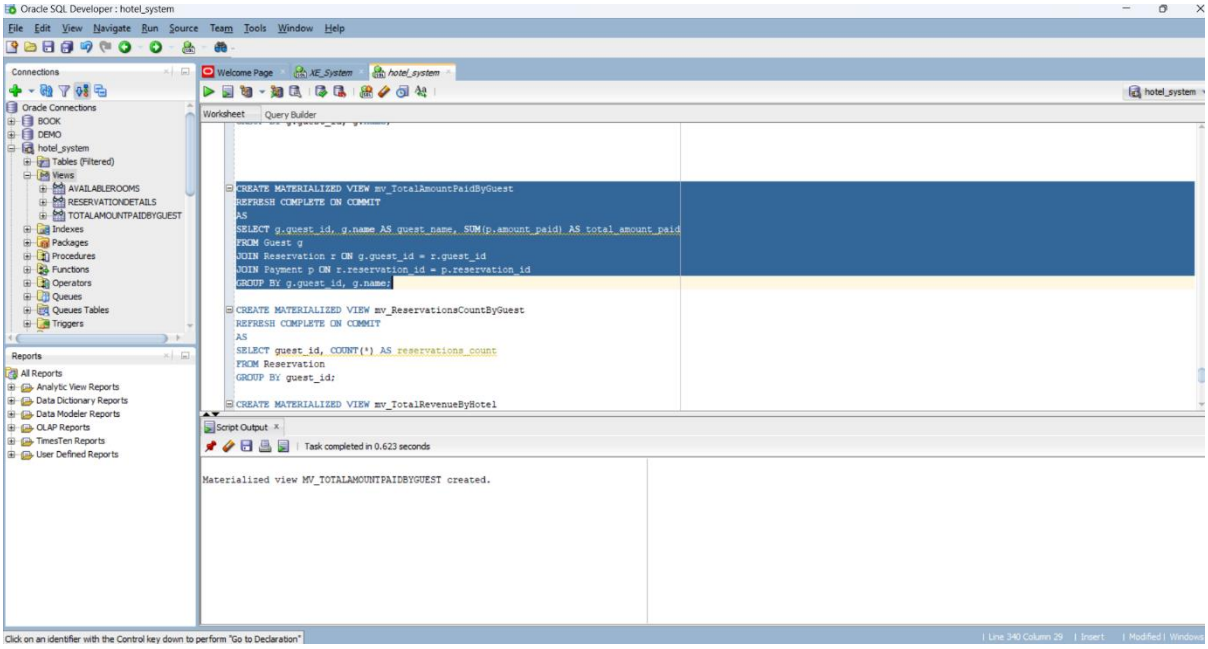


Figure 67:materialized view

mv_TotalAmountPaidByGuest

The purpose of This shows the total amount paid for each guest by summing up the payments related to their reservation