Project – Database Foundation

Project Title: DBF 12

Tool – MySQL Workbench

DBF 12 - Technical Report

Table of Contents

I. Introduction

II. Requirements Definition Document

a. Entity and Attribute Description

b. Relationship, Cardinality Description and Business Rules

c. Assumptions and Special Considerations

III. Detailed Database Design – Entity Relationship Diagram (ERD)
IV. DDL Source Code
V. DML Source Code
VI. Query Source Code
VII. DDL Output
VIII. DML Output
IX. Query Output

I. Introduction

In this Database Design Document, we will see the design and implementation of a database for the front desk of our new hotel that has a theme park on the hotel premises itself. The Hotel is named as DBF 12 in loving memory of our database foundations class taught by Prof. Gasan Elkhodari. The Number 12 is derived from Group 12 that we are a part of and was generated by an auto-generator. As this document progresses, we will see the design and implementation of the database for the front desk of hotel DBF 12 that will store all the guest information with details, the rooms and the amenities they offer including the rates for it, reservation management system, pricing policies and finally computing total payment for our guests based on their selections, any discounts offered to them and storing the payment details. So, check out our new hotel DBF 12 that has its own theme park for your next vacation!

II. Requirements Definition Document

a. Entity and Attribute Description

Entity Name: Guest
Entity Description: Guest is the primary entity around whom all the other entities would revolve. It includes details on all guests.
Main Attributes of Guest:

Guest_ID: (Primary Key) A unique identifier of the guest's account in Hotel DBF 12's database.
L_Name: Last Name of the guest who is the primary guest under whose name booking is made and who is responsible for clearing payment upon checkout
F_Name: First Name of the guest who is the primary guest under whose name booking is made and who is responsible for clearing payment upon checkout
Age: Age of the primary guest under whose name booking is made and who is responsible for clearing payment upon checkout. It is important to know the age as the hotel requires an adult to check-in. Also, the hotel offers a discount to senior citizens who are of the age 65 years and above.
Email_ID: Email_ID of the guest who is the primary guest under whose name booking is made and who is responsible for clearing payment upon checkout. Email_ID is used for the purposes of sending reservation confirmation, discount coupons and a copy of payment receipt.
Phone: It identifies the 10 digit phone number of the guest who is the primary guest under whose name booking is made and who is responsible for clearing payment upon checkout
Platinum_Member: Identifier of whether a guest has a platinum membership at Hotel DBF 12 or not. It is identified as 'Yes' if the guest is a platinum member of Hotel DBF 12 and 'No' if the guest is not a platinum member of Hotel DBF 12.

Entity Name: Reservation
Entity Description: Reservation Entity reflects the reservation details of the guest like their planned stay at the hotel, purchase of theme park tickets and how far in advanced they made the reservation.
Main Attributes of Reservation:

Res_No: (Primary Key) A unique identifier of the guest's reservation in Hotel DBF 12's database.

Guest_ID: (Foreign Key) References the Guest table and connects each unique reservation to one guest.
Checkin_Dt: Date when the guest checks in at Hotel DBF 12.
Checkout_Dt: Date when the guest checks out of Hotel DBF 12.
No_of_Nights: The total stay a guest is at the hotel for. It is computed through subtracting the Checkin_Dt from the Checkout_Dt
Res_Status: It reflects the status of the guest's reservation. Whether the reservation is confirmed or is it in waiting status for a room to become available for the dates requested.
No_of_Guests : This is the total number of guests on a particular reservation that includes the primary guest who checked-in and all the other guests accompanying him/her.
No_of_TP_Tickets: It gives the count of DBF 12 theme park tickets a guest purchased along with their room reservation.
Res_Date: Date when the guest made the reservation. It is an important factor in deciding the discount for the guest. If the guest made the reservation 30 days before their check-in date, they would get an early bird discount of 10%.

Entity Name: Rooms
Entity Description: It describes the various features of a room that is reserved by a guest.
Main Attributes of Rooms:

Room_No: (Primary Key) A unique identifier of each room. It is not auto generated but manually assigned by the Hotel Staff based on the attributes and amenities a customer selects while making reservations.
Res_No:(Foreign Key) References the Reservation table and connects each unique reservation to the room reserved under that reservation.
Room_Category: It categorizes a room into either Standard King, Standard Queen, Standard Twin or an Executive Suite.
Room_Type: It indicates whether a room is a smoking room or a non-smoking room
Room_View: It indicates whether a room has a Pool View, Street View, Themepark View or a Garden View
Breakfast: It is a 'Yes' or 'No' indicator which indicates whether a buffet breakfast is included with the room reservation or not.
Room_Rate: It is a rate predetermined by the owner of the Hotel DBF 12 who has provided a specific non-changeable room rate card to the hotel front desk staff that is to be used for making any reservation and collection of final payment.

Entity Name: Pricing
Entity Description: Pricing table plays a crucial role in determining the final payment by indicating how much to charge each customer for their hotel stay and theme park tickets and what discounts to apply.
Main Attributes of Pricing:

Pricing_ID: (Primary Key) A unique identifier that indicates what price to charge based on the discounts a guest is eligible for. It follows the DBF 12 clause that only one type of discount can be offered to a guest even if they are eligible for more than one type of discount. The discount

with the maximum percentage off will be offered to the guest and the Pricing ID associated with that discount will be used to indicate the payment they should be charged.
Discount_Code: A unique identifier that identifies the discount offered to a guest.
Discount_Type: It indicates the type of discount offered to a guest based on their eligibility.
Discount_percent: It indicates the percentage off a guest would receive on their total bill based on their eligibility.
Theme_Park_Price: Kids under the age of 6 get a free theme park entry. A theme park entry ticket is required for anyone age of 6 and above who wishes to enjoy theme park rides and is sold at a flat rate of $55. A hotel guest is however not required to purchase the theme park ticket. It is their choice if they purchase the theme park ticket along with their stay at the hotel. If they purchase more than two theme park tickets, they get a discount of 5%. If they reserve a room and purchase a theme park ticket they get a discount of 20%. Again, as per DBF 12 clause, if a guest is eligible for more than one discount, only one discount would be applicable, whichever is higher.
Room_Tax_Rate: It indicates the total of city and state tax rate on hotel rooms which is 13% for Richardson where Hotel DBF 12 is located at. The only ones exempt from Room_Tax_Rate are guests with a state of Texas tax exempt certificate. In that event, the front desk would use the Pricing ID of 6 to indicate that they are Tax Exempt.
TP_Ticket_Tax_Rate: It indicates the sales tax of 8.25% to be charged on the purchase of each theme park ticket. The only ones exempt from TP_Ticket_Tax_Rate are guests with a state of Texas tax exempt certificate. In that event, the front desk would use the Pricing ID of 6 to indicate that they are Tax Exempt.

Entity Name: Payment
Entity Description: The Payment entity indicates the payment amount due for a guest for their stay at the hotel and purchase of theme park tickets and the payment details.
Main Attributes of Payment:

Pmt_ID: (Primary Key) : A unique identifier that indicates the Payment ID associated with a particular reservation for a guest.
Res_No (Foreign Key): It references to the Reservation table and indicates the choices of the guest made under this reservation, the length of their stay and the number of theme park tickets purchased by them(if applicable).
Pricing_ID(Foreign Key): It references to the Pricing table. The Pricing ID is determined, based on the discount a guest is eligible for. It follows the DBF 12 clause that only one type of discount can be offered to a guest even if they are eligible for more than one type of discount. The discount with the maximum percentage off will be offered to the guest and the Pricing ID associated with that discount will be used to indicate the payment they should be charged.
Pmt_Method: It would indicate whether a guest would be using a credit card or a debit card for the payment of their hotel room stay and theme park ticket purchase if applicable.
Card_Number: A 16 digit number unique to each credit or debit card used to make the payment.
Type_of_Card: It would indicate whether the card is a Visa or a Mastercard.
Name_on_Card: This would indicate the First name and Last name of the guest generated by concatenating the First Name and Last Name from the Guest Table
Exp_Date: This would indicate the expiration date of the credit or debit card used to make the payment by the guest.

Pmt_Status: This would indicate whether a guest who made a reservation at the hotel has paid in full and cleared their invoice or do they have a balance due. The balance due amount will be listed under the Final Payment Amount. The Final Payment Amount is subject to change based on the guest's addition to additional theme park tickets or their extending the stay at the hotel. Hence, the Final Payment Amount is calculated for the guest during the check-out process.

b. Relationship, Cardinality Description and Business Rules

Relationship: has between GUEST and RESERVATION.
Cardinality: 1:M between GUEST and RESERVATION.
Business rule: A GUEST may have zero, one or two RESERVATION at a given point in time; One RESERVATION must be associated with only one GUEST.

Relationship: shows between RESERVATION and ROOMS.
Cardinality: 1:1 between RESERVATION and ROOMS.
Business rule: Each RESERVATION shows details on one ROOM associated with it. Each ROOM is associated with only one RESERVATION. Thus, only one ROOM can be booked per RESERVATION identified by a unique Reservation Number. If multiple ROOMS need to be booked, multiple RESERVATION under different Reservation Number should be made by the GUEST.

Relationship: associates between RESERVATION and PAYMENT.
Cardinality: 1:1 between RESERVATION and PAYMENT.
Business rule: Each RESERVATION is associated with a unique PAYMENT. Each PAYMENT is associated with only one RESERVATION.

Relationship: determines between PRICING and PAYMENT.
Cardinality: 1:1 between PRICING and PAYMENT.
Business rule: Each PRICING ID determines a unique PAYMENT. Each PAYMENT is associated with only one PRICING ID. This is possible through the hotel DBF 12 clause that only one Discount Code can be used at a time. If a guest is eligible for more than one type of Discount Code, the one with the higher percentage will be considered. Thus, their PAYMENT with only be determined by one PRICING ID that is associated with the higher discount code.

BUSINESS RULES:

1. A GUEST may have zero, one or two RESERVATIONS at a given point in time
2. Each RESERVATION must be associated with only one GUEST.
3. Each RESERVATION shows details on one ROOM associated with it.
4. Each ROOM is associated with only one RESERVATION.
5. Each RESERVATION is associated with a unique PAYMENT.
6. Each PAYMENT is associated with only one RESERVATION.
7. Each PRICING ID determines a unique PAYMENT.
8. Each PAYMENT is associated with only one PRICING ID.

c. Assumptions and Special Considerations

1. Hotel DBF 12 clause is that only one Discount Code can be used at a time. If a guest is eligible for more than one type of Discount Code, the one with the higher percentage will be

considered. Thus, their PAYMENT will only be determined by one PRICING ID that is associated with the higher discount code.

2. Kids under the age of 6 get a free theme park entry. A theme park entry ticket is required for anyone age of 6 and above and is sold at a flat rate of $55.

3. Platinum Membership at DBF 12 is earned by accumulating points. When a Guest accumulates 1000 points, they are awarded platinum membership. Each stay at Hotel DBF 12 leads to accumulation of 100 points. Once a guest is a platinum member, they are eligible for the Platinum member discount of 15% and access to the Hotel's VIP lounge on the Roof top. The Platinum Membership points and further details are maintained by our marketing staff in their database. The Guest table in the front desk database would only have an indication whether they are a platinum member or not.

4. Guest records remain on DBF 12's database for 3 years. After that they are purged. So, a guest who has been to the hotel in last three years will show up even if they don't have a current reservation at the hotel. Also, DBF 12 accepts walk-ins, so we will let a guest check-in instantly without a prior reservation on record. Upon check-in, they will be assigned an auto-generated sequential Res_No that will be associated with their account.

5. Hotel DBF 12's owner is very particular and has pre-determined the nightly rate for each room based on the Room Category, Type of Room, Type of View it Offers, whether it is a Smoking or Non-Smoking Room and whether a guest chooses to include Buffet Breakfast or Not as a part of their reservation. The Front Desk is expected to enter this amount for the Room Rate. The Owner has shared her Room Rate Card below:

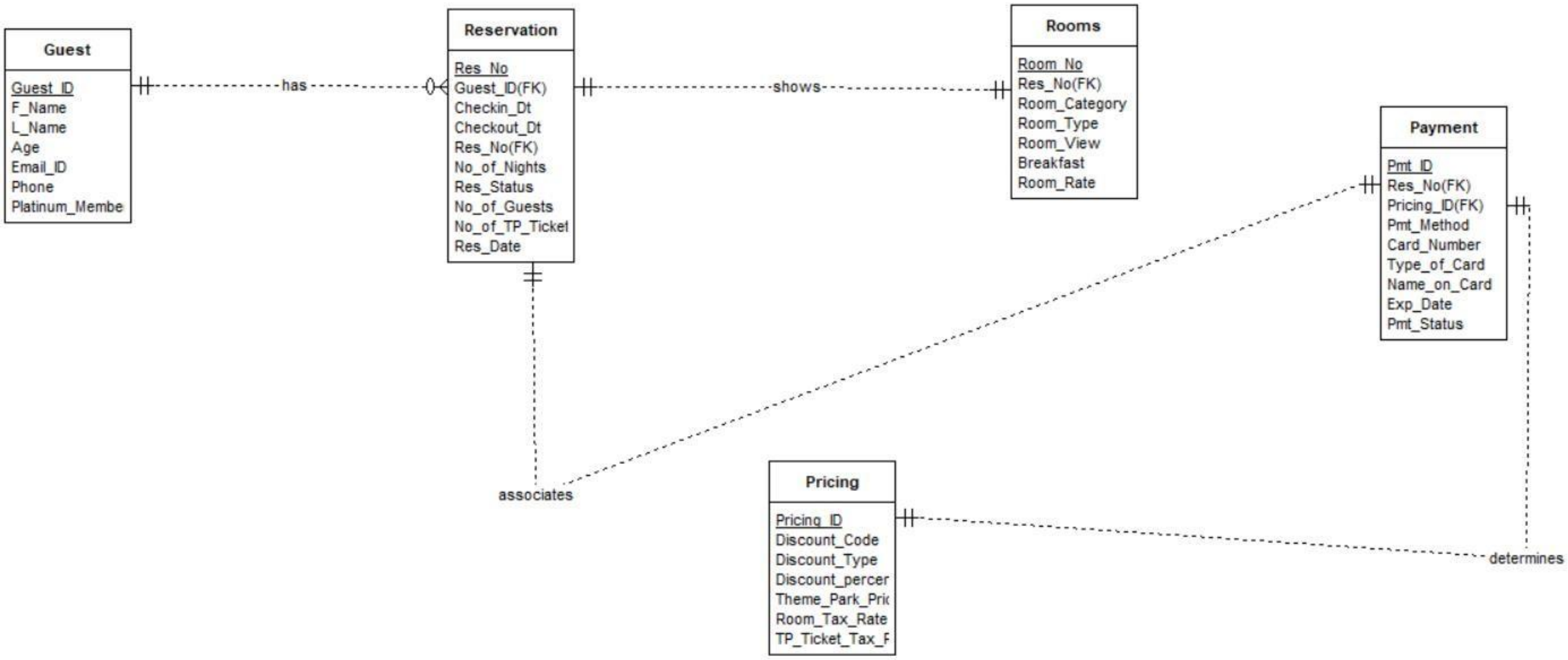
Category	View	Type	Breakfast	Room Rate/night
Standard King	Street View	Non-Smoking	Yes	$170
Standard King	Street View	Non-Smoking	No	$160
Standard King	Street View	Smoking	Yes	$175
Standard King	Street View	Smoking	No	$165
Standard King	Themepark View	Non-Smoking	Yes	$185
Standard King	Themepark View	Non-Smoking	No	$175
Standard King	Themepark View	Smoking	Yes	$190
Standard King	Themepark View	Smoking	No	$180
Standard King	Pool View	Non-Smoking	Yes	$180

Standard King	Pool View	Non-Smoking	No	$170
Standard King	Pool View	Smoking	Yes	$185
Standard King	Pool View	Smoking	No	$175
Standard King	Garden View	Non-Smoking	Yes	$175
Standard King	Garden View	Non-Smoking	No	$165
Standard King	Garden View	Smoking	Yes	$180
Standard King	Garden View	Smoking	No	$170
Standard Queen	Street View	Non-Smoking	Yes	$150
Standard Queen	Street View	Non-Smoking	No	$140
Standard Queen	Street View	Smoking	Yes	$155
Standard Queen	Street View	Smoking	No	$145
Standard Queen	Themepark View	Non-Smoking	Yes	$165
Standard Queen	Themepark View	Non-Smoking	No	$155
Standard Queen	Themepark View	Smoking	Yes	$170
Standard Queen	Themepark View	Smoking	No	$160
Standard Queen	Pool View	Non-Smoking	Yes	$160
Standard Queen	Pool View	Non-Smoking	No	$150
Standard Queen	Pool View	Smoking	Yes	$165
Standard Queen	Pool View	Smoking	No	$155
Standard Queen	Garden View	Non-Smoking	Yes	$155
Standard Queen	Garden View	Non-Smoking	No	$145
Standard Queen	Garden View	Smoking	Yes	$160
Standard Queen	Garden View	Smoking	No	$150
Standard Twin	Street View	Non-Smoking	Yes	$130
Standard Twin	Street View	Non-Smoking	No	$120
Standard Twin	Street View	Smoking	Yes	$135
Standard Twin	Street View	Smoking	No	$125

Standard Twin	Themepark View	Non-Smoking	Yes	$145
Standard Twin	Themepark View	Non-Smoking	No	$135
Standard Twin	Themepark View	Smoking	Yes	$150
Standard Twin	Themepark View	Smoking	No	$140
Standard Twin	Pool View	Non-Smoking	Yes	$140
Standard Twin	Pool View	Non-Smoking	No	$130
Standard Twin	Pool View	Smoking	Yes	$145
Standard Twin	Pool View	Smoking	No	$135
Standard Twin	Garden View	Non-Smoking	Yes	$135
Standard Twin	Garden View	Non-Smoking	No	$125
Standard Twin	Garden View	Smoking	Yes	$140
Standard Twin	Garden View	Smoking	No	$130
Executive Suite	Street View	Non-Smoking	Yes	$210
Executive Suite	Street View	Non-Smoking	No	$200
Executive Suite	Street View	Smoking	Yes	$215
Executive Suite	Street View	Smoking	No	$205
Executive Suite	Themepark View	Non-Smoking	Yes	$225
Executive Suite	Themepark View	Non-Smoking	No	$215
Executive Suite	Themepark View	Smoking	Yes	$230
Executive Suite	Themepark View	Smoking	No	$220
Executive Suite	Pool View	Non-Smoking	Yes	$220
Executive Suite	Pool View	Non-Smoking	No	$210
Executive Suite	Pool View	Smoking	Yes	$225
Executive Suite	Pool View	Smoking	No	$215
Executive Suite	Garden View	Non-Smoking	Yes	$215

Executive Suite	Garden View	Non-Smoking	No	$205
Executive Suite	Garden View	Smoking	Yes	$220
Executive Suite	Garden View	Smoking	No	$210

III. Detailed Database Design – Entity Relationship Diagram (ERD)



IV. DDL Source Code

```
/*Project DBF 12- DDL*/


/* Create tables based on entities */
CREATE TABLE Guest
(
Guest_ID INTEGER
AUTO_INCREMENT NOT
NULL,
F_Name VARCHAR(25) NOT
NULL,
L_Name VARCHAR(25) NOT
NULL,
AGE INTEGER NOT NULL,
Email_ID VARCHAR(40) NOT
NULL,
Phone VARCHAR(20) NOT
NULL,
Platinum_Member VARCHAR(3)
NOT NULL,
CONSTRAINT PK_Guest
PRIMARY KEY (Guest_ID)
)AUTO_INCREMENT = 101;

CREATE TABLE Reservation
(
   Res_no INTEGER
AUTO_INCREMENT NOT
NULL,
   Guest_ID INTEGER NOT
NULL,
   Checkin_Dt DATE NOT NULL,
   Checkout_Dt DATE NOT
NULL,
   No_of_Nights INTEGER
GENERATED ALWAYS AS
(DATEDIFF(Checkout_Dt,
Checkin_Dt)) STORED,
   Res_Status VARCHAR(50)
NOT NULL,
   No_of_Guests INTEGER NOT
NULL,
   No_of_TP_Tickets INTEGER
NOT NULL,
   Res_Date DATE NOT NULL,
   CONSTRAINT PK_Reservation
PRIMARY KEY (Res_no),
   CONSTRAINT FK_Reservation
FOREIGN KEY (Guest_ID)
REFERENCES Guest(Guest_ID)
) AUTO_INCREMENT = 10025;



Create TABLE Rooms
(
Room_No INTEGER NOT NULL,
Res_No INTEGER NOT NULL,
Room_Category VARCHAR(30)
NOT NULL,
Room_Type VARCHAR(30) NOT
NULL,
Room_View VARCHAR(30) NOT
NULL,
Breakfast VARCHAR(5) NOT
NULL,
Room_Rate VARCHAR(15) NOT
NULL,
CONSTRAINT PK_Rooms
PRIMARY KEY (Room_No),
CONSTRAINT FK_Rooms
FOREIGN KEY (Res_No)
REFERENCES Reservation
(Res_No)
);

CREATE TABLE Pricing
```

```
(
Pricing_ID INTEGER
AUTO_INCREMENT NOT
NULL,
Discount_Code VARCHAR(4)
NOT NULL,
Discount_Type VARCHAR(35)
NOT NULL,
Discount_percent INTEGER NOT
NULL,
Theme_Park_Price INTEGER
NOT NULL,
Room_Tax_Rate INTEGER NOT
NULL,
TP_Ticket_Tax_Rate INTEGER
NOT NULL,
CONSTRAINT PK_Pricing
PRIMARY KEY (Pricing_ID)
)AUTO_INCREMENT = 1;

CREATE TABLE Payment
(
Pmt_ID INTEGER
AUTO_INCREMENT NOT
NULL,
Res_No INTEGER NOT NULL,
Pricing_ID INTEGER NOT
NULL,
Pmt_Method VARCHAR(35)
NOT NULL,
Card_Number VARCHAR(16)
NOT NULL,
Type_of_Card VARCHAR(16)
NOT NULL,
Name_on_Card VARCHAR(50)
NOT NULL,
Exp_Date VARCHAR(7) NOT
NULL,
Pmt_Status VARCHAR(20) NOT
NULL,
CONSTRAINT PK_Payment
PRIMARY KEY (Pmt_ID),
CONSTRAINT FK_Payment
FOREIGN KEY (Res_No)
REFERENCES Reservation
(Res_No),
CONSTRAINT FK2_Payment
FOREIGN KEY (Pricing_ID)
REFERENCES Pricing
(Pricing_ID)
)AUTO_INCREMENT = 130045;


/* Create indices for natural keys,
foreign keys, and frequently-
queried columns */
-- Guest
-- Natural keys
CREATE INDEX
IDX_Guest_FName ON Guest
(F_Name);
```

```
-- Frequently-queried columns
CREATE INDEX
IDX_Guest_Age ON Guest (Age);
CREATE INDEX
IDX_Guest_Platinum_Membershi
p ON Guest (Platinum_Member);
-- Reservation
-- Frequently-queried columns
CREATE INDEX
IDX_Reservation_No_of_Nights
ON Reservation (No_of_Nights);
CREATE INDEX
IDX_Reservation_No_of_TP_Tick
ets ON Reservation
(No_of_TP_Tickets);
-- Foreign Key
CREATE INDEX IDX_Guest_ID
ON Reservation (Guest_ID);
-- Rooms
-- Frequently-queried columns
CREATE INDEX
IDX_Room_View ON Rooms
(Room_View);
CREATE INDEX
IDX_Room_Rate ON Rooms
(Room_Rate);
-- Pricing
-- Frequently-queried columns
CREATE INDEX
IDX_Pricing_Discount ON Pricing
(Discount_percent);
```

```
CREATE INDEX
IDX_Pricing_Discount_Type ON
Pricing (Discount_Type);
-- Payment
-- Natural keys
CREATE INDEX
IDX_Payment_Name_on_Card
ON Payment (Name_on_Card);
-- Frequently-queried columns
CREATE INDEX
IDX_Payment_Card_number ON
Payment (Card_Number);
-- Foreign Keys
CREATE INDEX
IDX_Payment_Res_No ON
Payment (Res_No);
CREATE INDEX
IDX_Paymnet_Pricing_ID ON
Payment (Pricing_ID);

/* Alter Tables by adding Audit
Columns */

ALTER TABLE Guest
ADD (
created_by VARCHAR(30),
date_created DATE, modified_by
VARCHAR(30), date_modified
DATE
);
```

```
ALTER TABLE Reservation
ADD (
created_by VARCHAR(30),
date_created DATE, modified_by
VARCHAR(30), date_modified
DATE
);

ALTER TABLE Rooms
ADD (
created_by VARCHAR(30),
date_created DATE, modified_by
VARCHAR(30), date_modified
DATE
);

ALTER TABLE Pricing
ADD (
created_by VARCHAR(30),
date_created DATE, modified_by
VARCHAR(30), date_modified
DATE
);

ALTER TABLE Payment
ADD (
created_by VARCHAR(30),
date_created DATE, modified_by
VARCHAR(30), date_modified
DATE
);
```

```
/* Create Views */
-- Business purpose: The GuestInfo
view will be used primarily for
quickly searching information
about individual guests for
populating their Age and whether
they are a Platinum member or not
for computing discounts.
CREATE OR REPLACE VIEW
GuestInfo AS
SELECT Guest_ID, F_Name,
L_Name, Age, Platinum_Member
FROM Guest;
-- Business purpose: The
ReservationInfo view will be used
primarily for quickly searching
information about individual guests
for populating the number of theme
park tickets they purchased and the
number of nights they would be
staying at the hotel to compute the
payment amount for them
CREATE OR REPLACE VIEW
ReservationInfo AS
SELECT Res_no, Guest_ID,
No_of_Nights,No_of_TP_Tickets
FROM Reservation;
-- Business purpose: The
RoomsInfo view will be used
primarily for quickly searching
```

```
about individual guests for
populating the room type and room
view and the price of the room
CREATE OR REPLACE VIEW
RoomsInfo AS
SELECT Room_No,
Room_Category, Room_Type,
Room_View, Room_Rate
FROM Rooms;
-- Business purpose: The
PricingInfo view will be used
primarily for quickly searching
information about the Discount
Code and Discount Type to derive
the Pricing_ID.
CREATE OR REPLACE VIEW
PricingInfo AS
SELECT Pricing_ID,
Discount_Type, Discount_percent
FROM Pricing;
-- Business purpose: The
PaymentInfo view will be used
primarily for quickly searching
information about the name of card
holder and respective Pmt_ID and
Res_No associated with it.
CREATE OR REPLACE VIEW
PaymentInfo AS
SELECT Pmt_ID, Res_No,
Name_on_Card
FROM Payment;
```

```
/* Create Triggers */
-- Business purpose: The
TR_Guest trigger automatically
assigns a sequential Guest ID to a
newly-inserted row in the Guest
table, assigning appropriate values
to the created_by and date_created
fields. If the record is being
inserted or updated, appropriate
values are assigned to the
modified_by and modified_date
fields.
DELIMITER //
CREATE TRIGGER TR_Guest
BEFORE INSERT ON Guest
FOR EACH ROW
BEGIN
    IF NEW.Guest_ID IS NULL
THEN
        SET NEW.Guest_ID =
(SELECT AUTO_INCREMENT
FROM
information_schema.TABLES
WHERE TABLE_NAME = 'Guest'
AND TABLE_SCHEMA =
DATABASE()) + 1;
    END IF;

    IF NEW.created_by IS NULL
THEN
        SET NEW.created_by =
USER();
    END IF;

    IF NEW.date_created IS NULL
THEN
        SET NEW.date_created =
NOW();
    END IF;

    SET NEW.modified_by =
USER();
    SET NEW.date_modified =
NOW();
END;
//
DELIMITER ;
-- Business purpose: The TR_Res
trigger automatically assigns a
sequential Reservation Number to
a newly-inserted row in the
Reservation table, assigning
appropriate values to the
created_by and date_created fields.
If the record is being inserted or
updated, appropriate values are
assigned to the modified_by and
modified_date fields.
DELIMITER //

CREATE TRIGGER TR_Res
BEFORE INSERT ON
```

```
Reservation FOR EACH ROW
BEGIN
    IF NEW.Res_No IS NULL
THEN
        SET NEW.Res_No =
(SELECT
NEXTVAL('SEQ_Res_No'));
    END IF;

    IF NEW.created_by IS NULL
THEN
        SET NEW.created_by =
CURRENT_USER();
    END IF;

    IF NEW.date_created IS NULL
THEN
        SET NEW.date_created =
NOW();
    END IF;

    SET NEW.modified_by =
CURRENT_USER();
    SET NEW.date_modified =
NOW();
END;
//

DELIMITER ;
-- Business purpose: The
TR_Rooms trigger automatically
```

assigns appropriate values to the created_by and date_created fields. If the record is being inserted or updated, appropriate values are assigned to the modified_by and modified_date fields.

```
DELIMITER //

CREATE TRIGGER TR_Rooms
BEFORE INSERT ON Rooms
FOR EACH ROW
BEGIN
   IF NEW.created_by IS NULL
THEN
       SET NEW.created_by =
CURRENT_USER();
   END IF;

   IF NEW.date_created IS NULL
THEN
       SET NEW.date_created =
NOW();
   END IF;

   SET NEW.modified_by =
CURRENT_USER();
   SET NEW.date_modified =
NOW();
END;
//
```

```
DELIMITER ;
-- Business purpose: The
TR_Pricing trigger automatically
assigns a sequential Pricing ID to a
newly-inserted row in the Pricing
table, assigning appropriate values
to the created_by and date_created
fields. If the record is being
inserted or updated, appropriate
values are assigned to the
modified_by and modified_date
fields.
DELIMITER //

CREATE TRIGGER TR_Pricing
BEFORE INSERT ON Pricing
FOR EACH ROW
BEGIN
    IF NEW.Pricing_ID IS NULL
THEN
        SET NEW.Pricing_ID =
(SELECT
NEXTVAL('SEQ_Pricing_ID'));
    END IF;

    IF NEW.created_by IS NULL
THEN
        SET NEW.created_by =
CURRENT_USER();
    END IF;
    IF NEW.date_created IS NULL
THEN
        SET NEW.date_created =
NOW();
    END IF;

    SET NEW.modified_by =
CURRENT_USER();
    SET NEW.date_modified =
NOW();
END;
//

DELIMITER ;

-- Business purpose: The
TR_Payment trigger automatically
assigns a sequential Pmt ID to a
newly-inserted row in the Payment
table, assigning appropriate values
to the created_by and date_created
fields. If the record is being
inserted or updated, appropriate
values are assigned to the
modified_by and modified_date
fields.
DELIMITER //

CREATE TRIGGER TR_Payment
BEFORE INSERT ON Payment
FOR EACH ROW
```

```
BEGIN
    IF NEW.Pmt_ID IS NULL
THEN
        SET NEW.Pmt_ID =
(SELECT
NEXTVAL('SEQ_Pmt_ID'));
    END IF;

    IF NEW.created_by IS NULL
THEN
        SET NEW.created_by =
CURRENT_USER();
    END IF;

    IF NEW.date_created IS NULL
THEN
        SET NEW.date_created =
NOW();
    END IF;

    SET NEW.modified_by =
CURRENT_USER();
    SET NEW.date_modified =
NOW();
END;
//

DELIMITER ;

/* Populate all tables */
-- Guest table
```

```
INSERT INTO Guest (F_Name,
L_Name, Age, Email_ID, Phone,
Platinum_Member)
VALUES ('Gasan', 'Elkhodari', 42,
'gasan.elkhodari@utdallas.edu',
'(972)883-4779', 'Yes');
INSERT INTO Guest (F_Name,
L_Name, Age, Email_ID, Phone,
Platinum_Member)
VALUES ('Sai Suma', 'Maguluri',
24,
'saisuma.maguluri@utdallas.edu',
'(972)883-2705', 'Yes');
INSERT INTO Guest (F_Name,
L_Name, Age, Email_ID, Phone,
Platinum_Member)
VALUES ('John', 'Doe', 69,
'john.doe@gmail.com', '(469)850-
5189', 'No');
INSERT INTO Guest (F_Name,
L_Name, Age, Email_ID, Phone,
Platinum_Member)
VALUES ('Jacob', 'Foster', 31,
'jacob.foster@gmail.com',
'(214)573-7890', 'Yes');
INSERT INTO Guest (F_Name,
L_Name, Age, Email_ID, Phone,
Platinum_Member)
VALUES ('John', 'Smith',
26,'john.smith@yahoo.com',
'(469)650-9879', 'No');
```

```
INSERT INTO Guest (F_Name,
L_Name, Age, Email_ID, Phone,
Platinum_Member)
VALUES ('Andrea', 'Jones', 55,
'andreajones@hotmail.com',
'(630)639-2914', 'No');
INSERT INTO Guest (F_Name,
L_Name, Age, Email_ID, Phone,
Platinum_Member)
VALUES ('Taylor', 'Anderson',
74,'taylor.anderson@gmail.com',
'(469)879-0770', 'No');
INSERT INTO Guest (F_Name,
L_Name, Age, Email_ID, Phone,
Platinum_Member)
VALUES ('Aman', 'Sharma',
23,'sharmaaman@yahoo.com',
'(847)245-0990', 'Yes');
INSERT INTO Guest (F_Name,
L_Name, Age, Email_ID, Phone,
Platinum_Member)
VALUES ('Kyle', 'Cunningham',
45,'cunninghamk@gmail.com',
'(202)230-8970', 'Yes');
INSERT INTO Guest (F_Name,
L_Name, Age, Email_ID, Phone,
Platinum_Member)
VALUES ('Mehnaz', 'Mahmood',
51, 'mmahmood@yahoo.com',
'(449)517-9809', 'No');
```

```
-- Reservation table
INSERT INTO Reservation
(Guest_ID, Checkin_Dt,
Checkout_Dt, Res_Status,
No_of_Guests, No_of_TP_Tickets,
Res_Date)
VALUES (101, '2022-10-24',
'2022-10-27', 'Confirmed', 1, 0,
'2022-10-14');
INSERT INTO Reservation
(Guest_ID, Checkin_Dt,
Checkout_Dt, Res_Status,
No_of_Guests, No_of_TP_Tickets,
Res_Date)
VALUES (102, '2022-09-29',
'2022-10-03', 'Confirmed', 5, 5,
'2022-09-10');
INSERT INTO Reservation
(Guest_ID, Checkin_Dt,
Checkout_Dt, Res_Status,
No_of_Guests, No_of_TP_Tickets,
Res_Date)
VALUES (103, '2022-10-21',
'2022-10-26', 'Confirmed', 4, 4,
'2022-10-11');
INSERT INTO Reservation
(Guest_ID, Checkin_Dt,
Checkout_Dt, Res_Status,
No_of_Guests, No_of_TP_Tickets,
Res_Date)
VALUES (104, '2022-11-14',
'2022-11-19', 'Confirmed', 2, 2,
'2022-10-30');
INSERT INTO Reservation
(Guest_ID, Checkin_Dt,
Checkout_Dt, Res_Status,
No_of_Guests, No_of_TP_Tickets,
Res_Date)
VALUES (105, '2022-11-27',
'2022-12-01', 'Confirmed', 2, 2,
'2022-11-20');
INSERT INTO Reservation
(Guest_ID, Checkin_Dt,
Checkout_Dt, Res_Status,
No_of_Guests, No_of_TP_Tickets,
Res_Date)
VALUES (106, '2022-12-08',
'2022-12-12', 'Confirmed', 2, 2,
'2022-11-15');
INSERT INTO Reservation
(Guest_ID, Checkin_Dt,
Checkout_Dt, Res_Status,
No_of_Guests, No_of_TP_Tickets,
Res_Date)
VALUES (107, '2022-12-12',
'2022-12-15', 'Confirmed', 3, 2,
'2022-11-10');
INSERT INTO Reservation
(Guest_ID, Checkin_Dt,
Checkout_Dt, Res_Status,
No_of_Guests, No_of_TP_Tickets,
Res_Date)
```

```
VALUES (108, '2022-12-15',
'2022-12-18', 'Waiting_for_Room
Availability', 4, 3, '2022-11-14');
INSERT INTO Reservation
(Guest_ID, Checkin_Dt,
Checkout_Dt, Res_Status,
No_of_Guests, No_of_TP_Tickets,
Res_Date)
VALUES (109, '2022-12-23',
'2022-12-25', 'Waiting_for_Room
Availability', 3, 3, '2022-11-22');
INSERT INTO Reservation
(Guest_ID, Checkin_Dt,
Checkout_Dt, Res_Status,
No_of_Guests, No_of_TP_Tickets,
Res_Date)
VALUES (110, '2022-12-25',
'2022-12-30', 'Waiting_for_Room
Availability', 2, 2, '2022-12-15');

-- Rooms Table
INSERT INTO Rooms (Room_No,
Res_No, Room_Category,
Room_Type, Room_View,
Breakfast, Room_Rate)
VALUES ('401', '10025', 'Standard
King', 'Non-Smoking', 'Street
View', 'No', '$160');
INSERT INTO Rooms(Room_No,
Res_No, Room_Category,
Room_Type, Room_View,
Breakfast, Room_Rate)
VALUES('302', '10026', 'Standard
Queen', 'Smoking', 'Themepark
View', 'Yes', '$170');
INSERT INTO Rooms(Room_No,
Res_No, Room_Category,
Room_Type, Room_View,
Breakfast, Room_Rate)
VALUES('203', '10027', 'Standard
Twin', 'Non-Smoking', 'Pool View',
'Yes', '$140');
INSERT INTO Rooms(Room_No,
Res_No, Room_Category,
Room_Type, Room_View,
Breakfast, Room_Rate)
VALUES('504', '10028', 'Executive
Suite', 'Smoking', 'Pool View',
'Yes', '$225');
INSERT INTO Rooms(Room_No,
Res_No, Room_Category,
Room_Type, Room_View,
Breakfast, Room_Rate)
VALUES('305', '10029', 'Standard
Queen', 'Smoking', 'Pool View',
'Yes', '$165');
INSERT INTO Rooms(Room_No,
Res_No, Room_Category,
Room_Type, Room_View,
Breakfast, Room_Rate)
VALUES('206', '10030', 'Standard
Twin', 'Non-Smoking', 'Themepark
```

```
View', 'No', '$135');
INSERT INTO Rooms(Room_No,
Res_No, Room_Category,
Room_Type, Room_View,
Breakfast, Room_Rate)
VALUES('507', '10031', 'Executive
Suite', 'Smoking', 'Themepark
View', 'Yes', '$230');
INSERT INTO Rooms(Room_No,
Res_No, Room_Category,
Room_Type, Room_View,
Breakfast, Room_Rate)
VALUES('408', '10032', 'Standard
King', 'Smoking', 'Garden View',
'Yes', '$180');
INSERT INTO Rooms(Room_No,
Res_No, Room_Category,
Room_Type, Room_View,
Breakfast, Room_Rate)
VALUES('209', '10033', 'Standard
Twin', 'Non-Smoking', 'Pool View',
'Yes', '$140');
INSERT INTO Rooms(Room_No,
Res_No, Room_Category,
Room_Type, Room_View,
Breakfast, Room_Rate)
VALUES('510', '10034', 'Executive
Suite', 'Smoking', 'Themepark
View', 'Yes', '$230');

-- Pricing Table
```

```
INSERT INTO Pricing
(Discount_Code, Discount_Type,
Discount_percent,
Theme_Park_Price,
Room_Tax_Rate,
TP_Ticket_Tax_Rate)
VALUES ('EB', 'Early Bird', 10,
55, 13, 8.25);
INSERT INTO Pricing
(Discount_Code, Discount_Type,
Discount_percent,
Theme_Park_Price,
Room_Tax_Rate,
TP_Ticket_Tax_Rate)
VALUES ('RT', 'Room Theme
Park Combo', 20, 55, 13, 8.25);
INSERT INTO Pricing
(Discount_Code, Discount_Type,
Discount_percent,
Theme_Park_Price,
Room_Tax_Rate,
TP_Ticket_Tax_Rate)
VALUES ('SC', 'Senior Citizen
Discount', 10, 55, 13, 8.25);
INSERT INTO Pricing
(Discount_Code, Discount_Type,
Discount_percent,
Theme_Park_Price,
Room_Tax_Rate,
TP_Ticket_Tax_Rate)
VALUES ('PM', 'Platinum
Member', 15, 55, 13, 8.25);
INSERT INTO Pricing
(Discount_Code, Discount_Type,
Discount_percent,
Theme_Park_Price,
Room_Tax_Rate,
TP_Ticket_Tax_Rate)
VALUES ('T3', 'More than 2
Theme Park Tickets', 5, 55, 13,
8.25);
INSERT INTO Pricing
(Discount_Code, Discount_Type,
Discount_percent,
Theme_Park_Price,
Room_Tax_Rate,
TP_Ticket_Tax_Rate)
VALUES ('EXEM', 'Tax Exempt',
0, 55, 0, 0);
INSERT INTO Pricing
(Discount_Code, Discount_Type,
Discount_percent,
Theme_Park_Price,
Room_Tax_Rate,
TP_Ticket_Tax_Rate)
VALUES ('VET', 'Veteran', 7, 55,
13, 8.25);
INSERT INTO Pricing
(Discount_Code, Discount_Type,
Discount_percent,
Theme_Park_Price,
Room_Tax_Rate,
TP_Ticket_Tax_Rate)
VALUES ('RES', 'Local Texas
Resident', 2, 55, 13, 8.25);
INSERT INTO Pricing
(Discount_Code, Discount_Type,
Discount_percent,
Theme_Park_Price,
Room_Tax_Rate,
TP_Ticket_Tax_Rate)
VALUES ('RSF', 'Room Service
Food Order', 2, 55, 13, 8.25);
INSERT INTO Pricing
(Discount_Code, Discount_Type,
Discount_percent,
Theme_Park_Price,
Room_Tax_Rate,
TP_Ticket_Tax_Rate)
VALUES ('CS', 'College Student',
3, 55, 13, 8.25);

-- Payment table
INSERT INTO Payment (Res_No,
Pricing_ID, Pmt_Method,
Card_Number, Type_of_Card,
Name_on_Card, Exp_Date,
Pmt_Status)
VALUES (10025,6,'Credit
Card','4044120045673450','Visa','
Gasan Elkhodari','04/2026','Paid');
INSERT INTO Payment (Res_No,
Pricing_ID, Pmt_Method,
Card_Number, Type_of_Card,
Name_on_Card, Exp_Date,
Pmt_Status)
VALUES (10026,2,'Debit
Card','5002341245783458','Master
card','Sai Suma
Maguluri','05/2025','Paid');
INSERT INTO Payment (Res_No,
Pricing_ID, Pmt_Method,
Card_Number, Type_of_Card,
Name_on_Card, Exp_Date,
Pmt_Status)
VALUES (10027,2,'Credit
Card','4232456789762345','Visa','J
ohn Doe','04/2023','Paid');
INSERT INTO Payment (Res_No,
Pricing_ID, Pmt_Method,
Card_Number, Type_of_Card,
Name_on_Card, Exp_Date,
Pmt_Status)
VALUES (10028,2,'Credit
Card','4908789067534573','Visa','J
acob Foster','09/2026','Balance
Due');
INSERT INTO Payment (Res_No,
Pricing_ID, Pmt_Method,
Card_Number, Type_of_Card,
Name_on_Card, Exp_Date,
Pmt_Status)
VALUES (10029,2,'Credit
Card','5590898765456723','Master
```

```
card','John
Smith','07/2025','Balance Due');
INSERT INTO Payment (Res_No,
Pricing_ID, Pmt_Method,
Card_Number, Type_of_Card,
Name_on_Card, Exp_Date,
Pmt_Status)
VALUES (10030,2,'Credit
Card','4890654323455789','Visa','
Andrea Jones','12/2026','Balance
Due');
INSERT INTO Payment (Res_No,
Pricing_ID, Pmt_Method,
Card_Number, Type_of_Card,
Name_on_Card, Exp_Date,
Pmt_Status)
VALUES (10031,2,'Credit
Card','4509897654231257','Visa','T
aylor Anderson','11/2025','Balance
Due');
INSERT INTO Payment (Res_No,
Pricing_ID, Pmt_Method,
Card_Number, Type_of_Card,
Name_on_Card, Exp_Date,
Pmt_Status)
VALUES (10032,2,'Credit
Card','4123578974562345','Visa','
Aman Sharma','03/2024','Balance
Due');
INSERT INTO Payment (Res_No,
Pricing_ID, Pmt_Method,
Card_Number, Type_of_Card,

Name_on_Card, Exp_Date,

Pmt_Status)

VALUES (10033,2,'Debit

Card','4678657089234452','Visa','

Kyle

Cunningham','05/2024','Balance

Due');

INSERT INTO Payment (Res_No,

Pricing_ID, Pmt_Method,

Card_Number, Type_of_Card,

Name_on_Card, Exp_Date,

Pmt_Status)

VALUES (10034,2,'Credit

Card','5890786543218764','Master

card','Mehnaz

Mahmood','12/2023','Balance

Due');
```

VII. Query Source Code

```
 -- 20 SQL Queries
-- Q1. Select all columns and all rows from one table
SELECT * FROM Guest;

-- Q2. Select five columns and all rows from one table
SELECT Room_Category, Room_View, Room_Type, Breakfast, Room_Rate FROM Rooms;

-- Q3. Select all columns from all rows from one view
SELECT * FROM ReservationInfo;

-- Q4. Using a join on 2 tables, select all columns and all rows
SELECT * FROM Reservation
LEFT OUTER JOIN Guest ON Reservation.Guest_ID = Guest.Guest_ID;

-- Q5. Select and order data retrieved from one table
SELECT * FROM Rooms
ORDER BY Room_No;
```

```
-- Q6. Using a join on 3 tables, select 5 columns from the 3 tables. Use syntax that would limit the output to 10 rows
SELECT Guest.F_Name, Guest.L_Name, Reservation.Res_Status, Rooms.Room_Type, Rooms.Room_Rate
FROM Reservation
INNER JOIN Guest ON Reservation.Guest_ID = Guest.Guest_ID
INNER JOIN Rooms ON Rooms.Res_No = Reservation.Res_No
LIMIT 10;

-- Q7. Select distinct rows using joins on 3 tables
SELECT DISTINCT Rooms.Room_Category, Rooms.Room_View, Rooms.Room_Type, Rooms.Room_Rate, Payment.Pricing_ID
FROM Rooms
INNER JOIN Reservation ON Rooms.Res_no = Reservation.Res_no
INNER JOIN Payment ON Reservation.Res_no = Payment.Res_no;

-- Q8. Use GROUP BY and HAVING in a select statement using one or more tables
SELECT Room_Category, Room_View, Room_Type, Breakfast, Room_Rate
FROM Rooms
GROUP BY Room_Rate, Room_Category, Room_View, Room_Type, Breakfast
HAVING Room_View = 'Themepark View'
ORDER BY Room_Rate;

-- Q9.Use IN clause to select data from one or more tables
SELECT * FROM Pricing
WHERE Discount_percent IN (10,15,20);

-- Q10.Select length of one column from one table
SELECT L_Name, Length(L_Name) from Guest;

-- Q11.Delete one record from one table. Use select statements to demonstrate the table contents before and after the DELETE statement. Make sure you use ROLLBACK afterwards so that the data will not be physically removed
COMMIT;
SELECT * FROM Payment;
DELETE FROM Payment WHERE Pricing_ID = 6;
SELECT * FROM Payment;
ROLLBACK;

-- Q12.Update one record from one table. Use select statements to demonstrate the table contents before and after the UPDATE statement. Make sure you use ROLLBACK afterwards so that the data will not be physically removed
START TRANSACTION;
-- Your UPDATE statement
UPDATE Pricing
SET Discount_Type = 'Military Service Discount'
WHERE Pricing_ID = '7';
-- Optionally, you can SELECT to verify changes
SELECT * FROM Pricing WHERE Pricing_ID = '7';
SELECT * FROM Pricing;
-- If everything is correct, you can COMMIT the changes
COMMIT;
```

```sql
-- If there's an issue and you want to discard changes, you can ROLLBACK
ROLLBACK;

-- Q13. List all guests whose Reservation is in Confirmed Status. List their IDs, Reservation Number, First Name, Last Name, Room Number, Payment Status and Reservation Status
SELECT Guest.Guest_ID, Reservation.Res_No, Guest.F_Name, Guest.L_Name, Rooms.Room_No, Payment.Pmt_Status, Reservation.Res_Status
FROM Reservation
INNER JOIN Rooms ON Rooms.Res_No = Reservation.Res_No
INNER JOIN Guest ON Guest.Guest_ID = Reservation.Guest_ID
INNER JOIN Payment ON Reservation.Res_No = Payment.Res_No
WHERE Reservation.Res_Status = 'Confirmed';

-- Q14. Write a query to assist the front desk in order to prepare a list of all guests whose room rate is at least $180. The front desk would like to leave a small complimentary gift with a customized note in their room. Display their Room Number, First Name, Last Name, Room Rate, Payment Status
SELECT Rooms.Room_No, Guest.F_Name, Guest.L_Name, Rooms.Room_Rate, Payment.Pmt_Status from Rooms
INNER JOIN Reservation ON Rooms.Res_No = Reservation.Res_No
INNER JOIN Payment ON Reservation.Res_No = payment.Res_No
INNER JOIN Guest on Guest.Guest_ID = Reservation.Guest_ID
WHERE Rooms.Room_Rate >= '$180'
ORDER BY Rooms.Room_No;

-- Q15. Hotel DBF 12's Marketing Team reached out to get a list of most popular room categories that get booked the most. Write a query to assist the marketing team. List the Room Category and the Count of the Ones that get booked the most.
SELECT q1.Room_Category, q1.num_occurrences
FROM (
SELECT Rooms.Room_Category, COUNT(*) AS num_occurrences
FROM Rooms
GROUP BY Rooms.Room_Category
) q1, (
SELECT MAX(q2.num_occurrences) AS max_occurrences
FROM (
SELECT Rooms.Room_Category, COUNT(*) AS num_occurrences
FROM Rooms
GROUP BY Rooms.Room_Category
) q2
) q3
WHERE q1.num_occurrences = q3.max_occurrences;

-- Q16. List all reservations where the Room Theme Park Combo Discount has been applied and a Mastercard will be used for Payment
SELECT Reservation.Res_No, Guest.F_Name, Guest.L_Name, Reservation.Res_Status, Payment.Type_of_Card, Pricing.Discount_Code
FROM Guest
INNER JOIN Reservation ON Guest.Guest_ID = Reservation.Guest_ID
INNER JOIN Payment ON Reservation.Res_no = Payment.Res_No
INNER JOIN Pricing on Payment.Pricing_ID = Pricing.Pricing_ID
WHERE Payment.Type_Of_Card = 'Mastercard'
```

```
AND Pricing.Discount_Code = 'RT'
ORDER BY Res_No;

-- Q17. List all the platinum members who have made reservation for atleast 3 people and have purchased at
least 2 Theme Park tickets
Select Guest.Guest_ID, Guest.F_Name, Guest.L_Name, Reservation.No_of_Guests,
Reservation.No_of_TP_Tickets, Guest.Platinum_Member
FROM Guest
INNER JOIN Reservation ON Guest.Guest_ID = Reservation.Guest_ID
WHERE Guest.Platinum_Member = 'Yes'
AND Reservation.No_of_Guests >= 3
AND Reservation.No_of_TP_Tickets >= 2
ORDER BY Guest_ID;

-- Q18. Write a query to assist the front desk staff of DBF 12 to calculate room total, theme park total and final
amt before and after discount for all the hotel guests.
CREATE TABLE payment_temp AS
SELECT
    a.*,
    (b.No_of_Nights) * CAST(SUBSTRING(c.Room_Rate, 2, 3) AS DECIMAL) AS room_total,
    (b.No_of_TP_Tickets * d.Theme_Park_Price) AS Theme_park_total
FROM
    payment a
    LEFT OUTER JOIN reservation b ON a.RES_NO = b.RES_NO
    LEFT OUTER JOIN rooms c ON b.RES_NO = c.RES_NO
    LEFT OUTER JOIN pricing d ON a.pricing_id = d.pricing_id;

SELECT
    a.PMT_ID,
    a.RES_NO,
    a.PRICING_ID,
    a.ROOM_TOTAL,
    a.THEME_PARK_TOTAL,
    Room_Tax_Rate,
    TP_Ticket_Tax_Rate,
    (a.room_total * (1 + (b.Room_Tax_Rate / 100))) + (a.Theme_park_total * (1 + (b.TP_Ticket_Tax_Rate /
100))) AS final_amt_bef_disc,
    ((a.room_total * (1 + (b.Room_Tax_Rate / 100))) + (a.Theme_park_total * (1 + (b.TP_Ticket_Tax_Rate /
100)))) * (1 - (b.Discount_percent / 100)) AS final_amt_after_disc
FROM
    payment_temp a
    LEFT OUTER JOIN pricing b ON a.pricing_id = b.pricing_id;

-- Q19. One of the hotel guest Mr. Gasan Elkhodari showed up to the hotel front desk with a tax exempt
certificate mentioning that the purpose of his stay at the hotel is for attending a conference nearby and as a state
employee of University of Texas at Dallas, he should be exempted from taxes. Please write a query to assist the
front desk staff for calculating Room total for Mr. Elkhodari and more guests like him in the future who would
present a valid tax exempt certificate to the hotel front desk.
SELECT A.res_no, A.f_name, A.l_name, B.discount_code, A.room_rate, B.room_total
FROM (
    SELECT *
    FROM (
        SELECT guest.guest_id AS g1_id, guest.f_name, guest.l_name
        FROM guest
        JOIN reservation ON guest.guest_id = reservation.guest_id
    ) AS A1,
    (
        SELECT reservation.guest_id AS g2_id, reservation.res_no, rooms.room_rate
        FROM reservation
        JOIN rooms ON reservation.res_no = rooms.res_no
    ) AS A2
    WHERE A1.g1_id = A2.g2_id
) AS A,
(
    SELECT T.res_no, pt.room_total, T.discount_code
    FROM payment_temp AS pt,
    (
        SELECT pricing_id, res_no, discount_code
        FROM (
            SELECT res_no, discount_code, pricing_id
            FROM pricing
            JOIN (
                SELECT reservation.res_no, pricing_id AS res_pid
                FROM reservation, payment
                WHERE reservation.res_no = payment.res_no
            ) AS P1 ON res_pid = pricing_id
        ) AS P2
        WHERE discount_code = 'EXEM'
    ) AS T
    WHERE pt.res_no = T.res_no
) AS B
WHERE A.res_no = B.res_no;

-- Q20. For people who availed Room theme park combo, find out how many platinum members are spending
more than the average amount
create table payment_temp1 as
select a.*,Room_Tax_Rate,TP_Ticket_Tax_Rate,(a.room_total *
(1+(b.Room_Tax_Rate/100)))+(a.Theme_park_total*(1+(b.TP_Ticket_Tax_Rate/100))) as final_amt_bef_disc,
((a.room_total * (1+(b.Room_Tax_Rate/100)))+(a.Theme_park_total*(1+(b.TP_Ticket_Tax_Rate/100))))*(1 -
(b.Discount_percent/100)) as final_amt_after_disc
from payment_temp a left outer join pricing b
on a.pricing_id = b.pricing_id;
select a.platinum_member,count(a.guest_id) as
COUNT_OF_CUSTOMERS_WHO_SPEND_ABOVE_AVERAGE
from guest a left outer join reservation b
on a.guest_id = b.guest_id
left outer join payment_temp1 c
on b.res_no = c.res_no
left outer join pricing d
on c.pricing_id = d.pricing_id
where d.discount_type = 'Room Theme Park Combo' AND
c.FINAL_AMT_AFTER_DISC > (select avg(FINAL_AMT_AFTER_DISC) from payment_temp1 where
pricing_id in (select pricing_id from pricing where discount_type = 'Room Theme Park Combo'))
group by a.platinum_member;
DROP TABLE payment_temp;
DROP TABLE payment_temp1;
```

VIII. DDL Output

```
Trigger TR_GUEST dropped.


Trigger TR_RES dropped.


Trigger TR_ROOMS dropped.


Trigger TR_PRICING dropped.


Trigger TR_PAYMENT dropped.


Sequence SEQ_GUEST_ID dropped.


Sequence SEQ_RES_NO dropped.


Sequence SEQ_PRICING_ID dropped.


Sequence SEQ_PMT_ID dropped.


View GUESTINFO dropped.
View RESERVATIONINFO dropped.

View ROOMSINFO dropped.

View PRICINGINFO dropped.

View PAYMENTINFO dropped.

Index IDX_GUEST_FNAME dropped.

Index IDX_GUEST_AGE dropped.

Index IDX_GUEST_PLATINUM_MEMBERSHIP dropped.

Index IDX_RESERVATION_NO_OF_NIGHTS dropped.

Index IDX_RESERVATION_NO_OF_TP_TICKETS dropped.

Index IDX_ROOM_VIEW dropped.

Index IDX_ROOM_RATE dropped.

Index IDX_PRICING_DISCOUNT dropped.

Index IDX_PRICING_DISCOUNT_TYPE dropped.

Index IDX_PAYMENT_NAME_ON_CARD dropped.

Index IDX_PAYMENT_CARD_NUMBER dropped.
Index IDX_PAYMENT_RES_NO dropped.

Index IDX_PAYMNET_PRICING_ID dropped.

Table PAYMENT dropped.

Table PRICING dropped.

Table ROOMS dropped.

Table RESERVATION dropped.

Table GUEST dropped.

Table GUEST created.

Table RESERVATION created.

Table ROOMS created.

Table PRICING created.

Table PAYMENT created.

Index IDX_GUEST_FNAME created.

Index IDX_GUEST_AGE created.

Index IDX_GUEST_PLATINUM_MEMBERSHIP created.

Index IDX_RESERVATION_NO_OF_NIGHTS created.
Index IDX_RESERVATION_NO_OF_TP_TICKETS created.

Index IDX_GUEST_ID created.

Index IDX_ROOM_VIEW created.

Index IDX_ROOM_RATE created.

Index IDX_PRICING_DISCOUNT created.

Index IDX_PRICING_DISCOUNT_TYPE created.

Index IDX_PAYMENT_NAME_ON_CARD created.

Index IDX_PAYMENT_CARD_NUMBER created.

Index IDX_PAYMENT_RES_NO created.

Index IDX_PAYMNET_PRICING_ID created.

Table GUEST altered.

Table RESERVATION altered.

Table ROOMS altered.

Table PRICING altered.

Table PAYMENT altered.
```

```
View GUESTINFO created.

View RESERVATIONINFO created.

View ROOMSINFO created.

View PRICINGINFO created.

View PAYMENTINFO created.

Sequence SEQ_GUEST_ID created.

Sequence SEQ_RES_NO created.

Sequence SEQ_PRICING_ID created.

Sequence SEQ_PMT_ID created.

Trigger TR_GUEST compiled

Trigger TR_RES compiled

Trigger TR_ROOMS compiled

Trigger TR_PRICING compiled

Trigger TR_PAYMENT compiled
```

VIII. DML Output

```
1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.
```

```
1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.
```

```
1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.
```

1 row inserted.

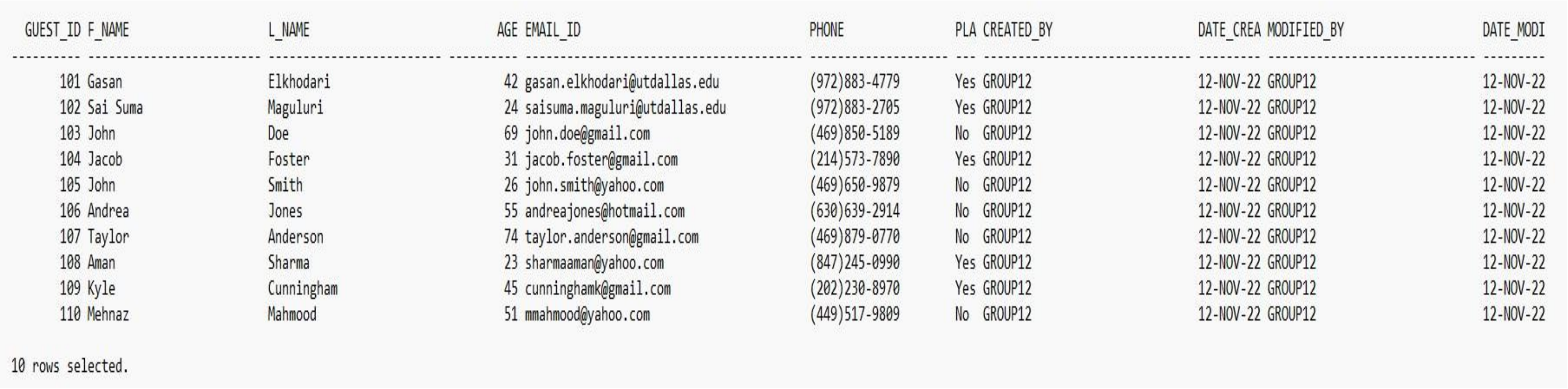
1 row inserted.

1 row inserted.

1 row inserted.

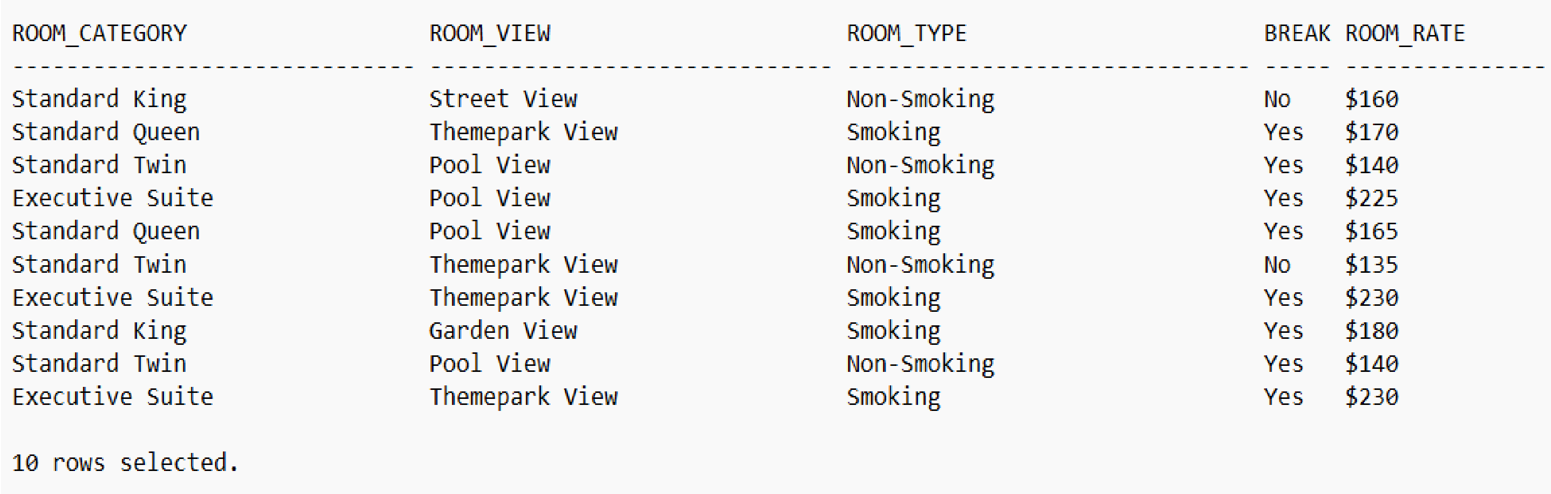
IX. Query Output

Q1:



```
 GUEST_ID F_NAME                    L_NAME                          AGE EMAIL_ID                            PHONE                PLA CREATED_BY                     DATE_CREA MODIFIED_BY                    DATE_MODI
--------- ------------------------- ------------------------- --------- ----------------------------------- -------------------- --- ------------------------------ --------- ------------------------------ ---------
      101 Gasan                     Elkhodari                        42 gasan.elkhodari@utdallas.edu        (972)883-4779        Yes GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
      102 Sai Suma                  Maguluri                         24 saisuma.maguluri@utdallas.edu       (972)883-2705        Yes GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
      103 John                      Doe                              69 john.doe@gmail.com                  (469)850-5189        No  GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
      104 Jacob                     Foster                           31 jacob.foster@gmail.com              (214)573-7890        Yes GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
      105 John                      Smith                            26 john.smith@yahoo.com                (469)650-9879        No  GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
      106 Andrea                    Jones                            55 andreajones@hotmail.com             (630)639-2914        No  GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
      107 Taylor                    Anderson                         74 taylor.anderson@gmail.com           (469)879-0770        No  GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
      108 Aman                      Sharma                           23 sharmaaman@yahoo.com                (847)245-0990        Yes GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
      109 Kyle                      Cunningham                       45 cunninghamk@gmail.com               (202)230-8970        Yes GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
      110 Mehnaz                    Mahmood                          51 mmahmood@yahoo.com                  (449)517-9809        No  GROUP12                        12-NOV-22 GROUP12                        12-NOV-22

10 rows selected.
```

Q2:



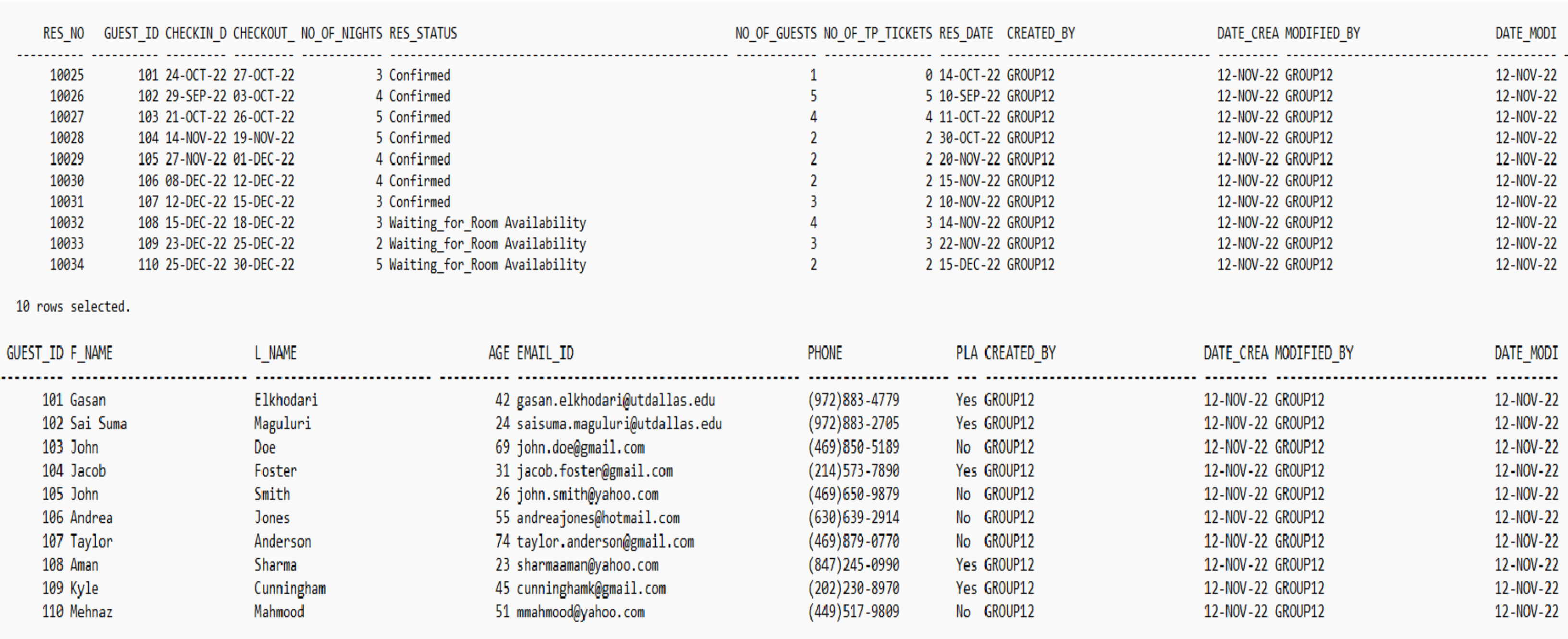
```
ROOM_CATEGORY                  ROOM_VIEW                      ROOM_TYPE                      BREAK ROOM_RATE
------------------------------ ------------------------------ ------------------------------ ----- ---------------
Standard King                  Street View                    Non-Smoking                    No    $160
Standard Queen                 Themepark View                 Smoking                        Yes   $170
Standard Twin                  Pool View                      Non-Smoking                    Yes   $140
Executive Suite                Pool View                      Smoking                        Yes   $225
Standard Queen                 Pool View                      Smoking                        Yes   $165
Standard Twin                  Themepark View                 Non-Smoking                    No    $135
Executive Suite                Themepark View                 Smoking                        Yes   $230
Standard King                  Garden View                    Smoking                        Yes   $180
Standard Twin                  Pool View                      Non-Smoking                    Yes   $140
Executive Suite                Themepark View                 Smoking                        Yes   $230

10 rows selected.
```

Q3:

```
 RES_NO  GUEST_ID NO_OF_NIGHTS NO_OF_TP_TICKETS

-------  --------- --------    -----------

 10025    101      3               0

 10026    102      4               5

 10027    103      5               4

 10028    104      5               2

 10029    105      4               2

 10030    106      4               2

 10031    107      3               2

 10032    108      3               3

 10033    109      2               3

 10034    110      5               2
```

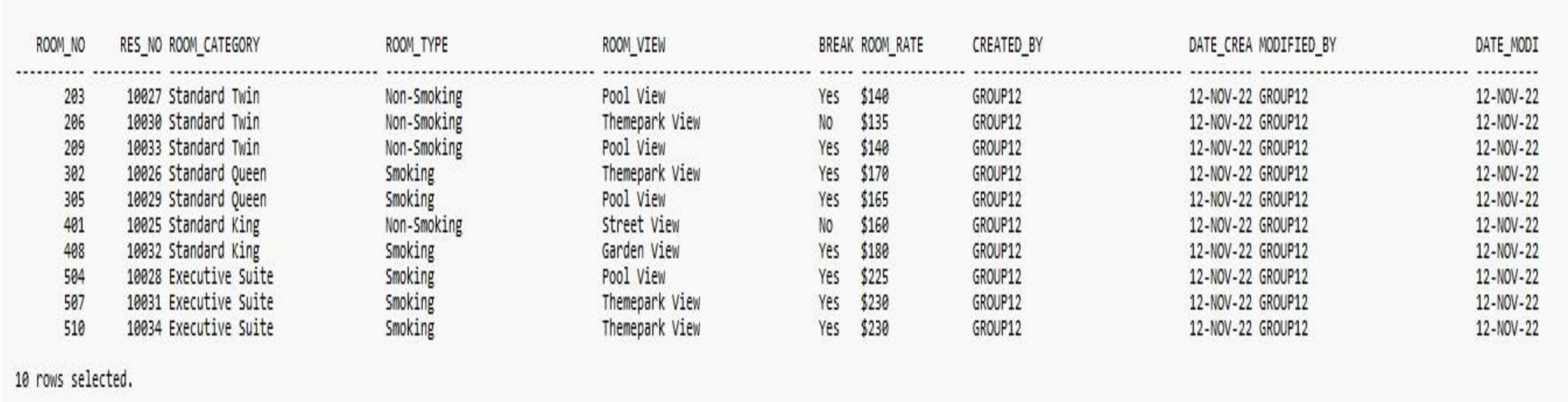
10 rows selected.

Q4:



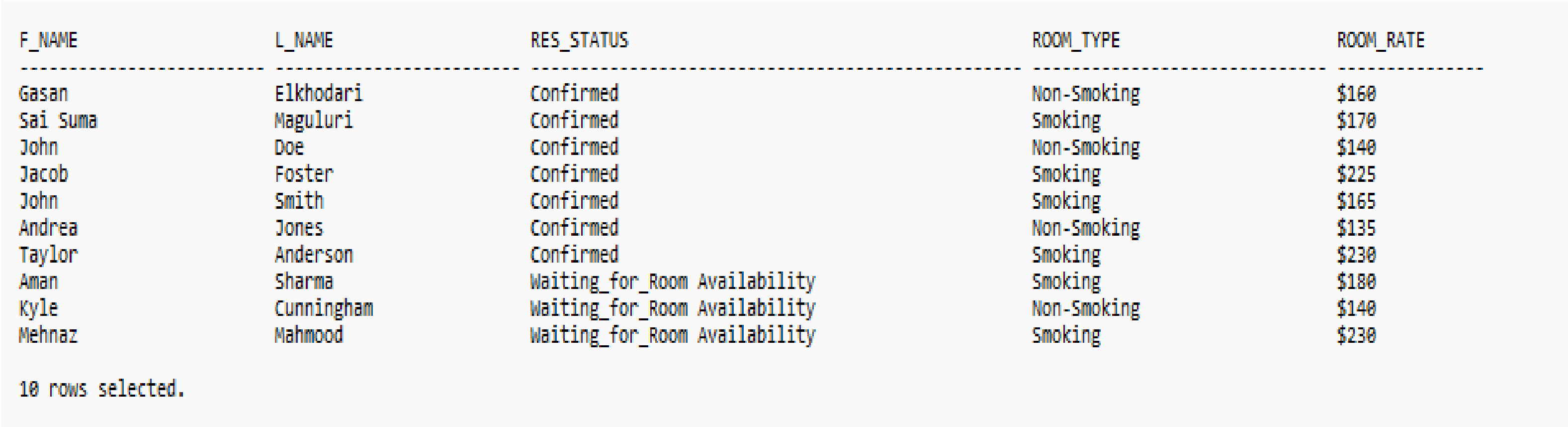
```
    RES_NO   GUEST_ID CHECKIN_D CHECKOUT_ NO_OF_NIGHTS RES_STATUS                                         NO_OF_GUESTS NO_OF_TP_TICKETS RES_DATE  CREATED_BY                     DATE_CREA MODIFIED_BY                    DATE_MODI
---------- ---------- --------- --------- ------------ -------------------------------------------------- ------------ ---------------- --------- ------------------------------ --------- ------------------------------ ---------
     10025        101 24-OCT-22 27-OCT-22            3 Confirmed                                                     1                0 14-OCT-22 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
     10026        102 29-SEP-22 03-OCT-22            4 Confirmed                                                     5                5 10-SEP-22 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
     10027        103 21-OCT-22 26-OCT-22            5 Confirmed                                                     4                4 11-OCT-22 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
     10028        104 14-NOV-22 19-NOV-22            5 Confirmed                                                     2                2 30-OCT-22 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
     10029        105 27-NOV-22 01-DEC-22            4 Confirmed                                                     2                2 20-NOV-22 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
     10030        106 08-DEC-22 12-DEC-22            4 Confirmed                                                     2                2 15-NOV-22 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
     10031        107 12-DEC-22 15-DEC-22            3 Confirmed                                                     3                2 10-NOV-22 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
     10032        108 15-DEC-22 18-DEC-22            3 Waiting_for_Room Availability                                 4                3 14-NOV-22 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
     10033        109 23-DEC-22 25-DEC-22            2 Waiting_for_Room Availability                                 3                3 22-NOV-22 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
     10034        110 25-DEC-22 30-DEC-22            5 Waiting_for_Room Availability                                 2                2 15-DEC-22 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22

10 rows selected.

GUEST_ID F_NAME                    L_NAME                           AGE EMAIL_ID                                 PHONE                PLA CREATED_BY                     DATE_CREA MODIFIED_BY                    DATE_MODI
-------- ------------------------- ------------------------- ---------- ---------------------------------------- -------------------- --- ------------------------------ --------- ------------------------------ ---------
     101 Gasan                     Elkhodari                         42 gasan.elkhodari@utdallas.edu             (972)883-4779        Yes GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
     102 Sai Suma                  Maguluri                          24 saisuma.maguluri@utdallas.edu            (972)883-2705        Yes GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
     103 John                      Doe                               69 john.doe@gmail.com                       (469)850-5189        No  GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
     104 Jacob                     Foster                            31 jacob.foster@gmail.com                   (214)573-7890        Yes GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
     105 John                      Smith                             26 john.smith@yahoo.com                     (469)650-9879        No  GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
     106 Andrea                    Jones                             55 andreajones@hotmail.com                  (630)639-2914        No  GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
     107 Taylor                    Anderson                          74 taylor.anderson@gmail.com                (469)879-0770        No  GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
     108 Aman                      Sharma                            23 sharmaaman@yahoo.com                     (847)245-0990        Yes GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
     109 Kyle                      Cunningham                        45 cunninghamk@gmail.com                    (202)230-8970        Yes GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
     110 Mehnaz                    Mahmood                           51 mmahmood@yahoo.com                       (449)517-9809        No  GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
```

Q5.



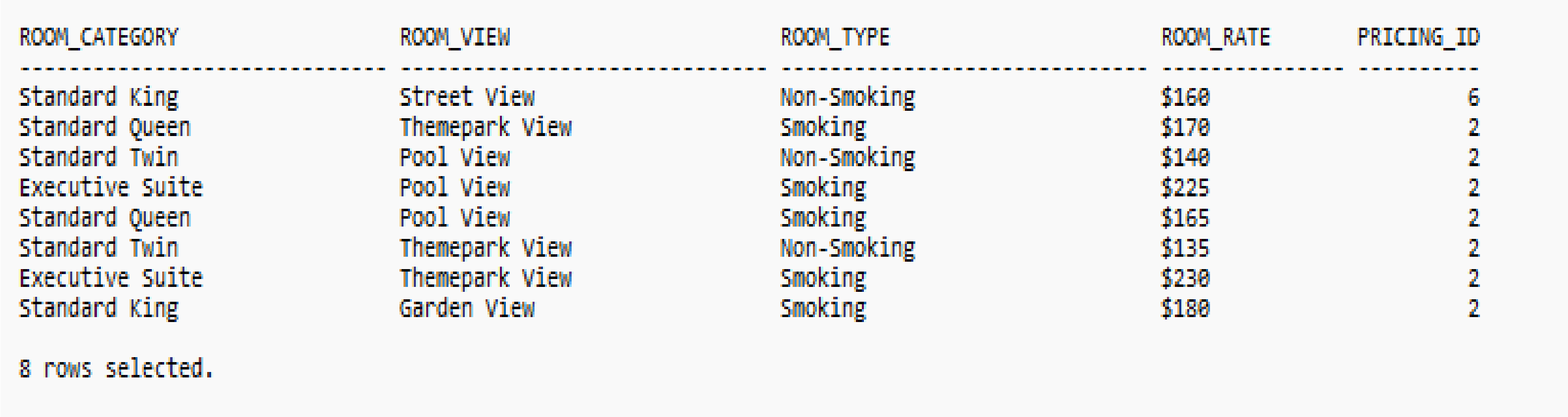
```
   ROOM_NO     RES_NO ROOM_CATEGORY                  ROOM_TYPE                      ROOM_VIEW                      BREAK ROOM_RATE       CREATED_BY                     DATE_CREA MODIFIED_BY                    DATE_MODI
---------- ---------- ------------------------------ ------------------------------ ------------------------------ ----- --------------- ------------------------------ --------- ------------------------------ ---------
       203      10027 Standard Twin                  Non-Smoking                    Pool View                      Yes   $140            GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
       206      10030 Standard Twin                  Non-Smoking                    Themepark View                 No    $135            GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
       209      10033 Standard Twin                  Non-Smoking                    Pool View                      Yes   $140            GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
       302      10026 Standard Queen                 Smoking                        Themepark View                 Yes   $170            GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
       305      10029 Standard Queen                 Smoking                        Pool View                      Yes   $165            GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
       401      10025 Standard King                  Non-Smoking                    Street View                    No    $160            GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
       408      10032 Standard King                  Smoking                        Garden View                    Yes   $180            GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
       504      10028 Executive Suite                Smoking                        Pool View                      Yes   $225            GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
       507      10031 Executive Suite                Smoking                        Themepark View                 Yes   $230            GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
       510      10034 Executive Suite                Smoking                        Themepark View                 Yes   $230            GROUP12                        12-NOV-22 GROUP12                        12-NOV-22

10 rows selected.
```

Q6.



```
F_NAME                    L_NAME                    RES_STATUS                                         ROOM_TYPE                      ROOM_RATE
------------------------- ------------------------- -------------------------------------------------- ------------------------------ ---------------
Gasan                     Elkhodari                 Confirmed                                          Non-Smoking                    $160
Sai Suma                  Maguluri                  Confirmed                                          Smoking                        $170
John                      Doe                       Confirmed                                          Non-Smoking                    $140
Jacob                     Foster                    Confirmed                                          Smoking                        $225
John                      Smith                     Confirmed                                          Smoking                        $165
Andrea                    Jones                     Confirmed                                          Non-Smoking                    $135
Taylor                    Anderson                  Confirmed                                          Smoking                        $230
Aman                      Sharma                    Waiting_for_Room Availability                      Smoking                        $180
Kyle                      Cunningham                Waiting_for_Room Availability                      Non-Smoking                    $140
Mehnaz                    Mahmood                   Waiting_for_Room Availability                      Smoking                        $230

10 rows selected.
```

Q7.

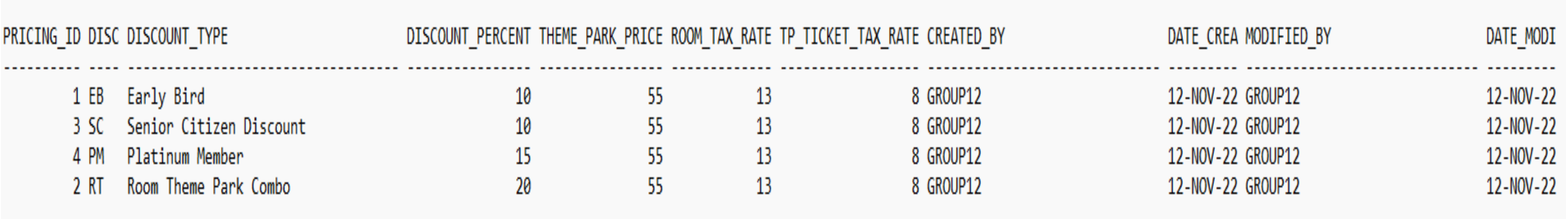


```
ROOM_CATEGORY                  ROOM_VIEW                      ROOM_TYPE                      ROOM_RATE       PRICING_ID
------------------------------ ------------------------------ ------------------------------ --------------- ----------
Standard King                  Street View                    Non-Smoking                    $160                     6
Standard Queen                 Themepark View                 Smoking                        $170                     2
Standard Twin                  Pool View                      Non-Smoking                    $140                     2
Executive Suite                Pool View                      Smoking                        $225                     2
Standard Queen                 Pool View                      Smoking                        $165                     2
Standard Twin                  Themepark View                 Non-Smoking                    $135                     2
Executive Suite                Themepark View                 Smoking                        $230                     2
Standard King                  Garden View                    Smoking                        $180                     2

8 rows selected.
```

Q8.

```
ROOM_CATEGORY   ROOM_VIEW            ROOM_TYPE            BREAK ROOM_RATE
---------------------------------------------------------- ------  ----------------
Standard Twin          Themepark View      Non-Smoking          No         $135
Standard Queen         Themepark View      Smoking              Yes        $170
Executive Suite        Themepark View      Smoking              Yes        $230
```

Q9.

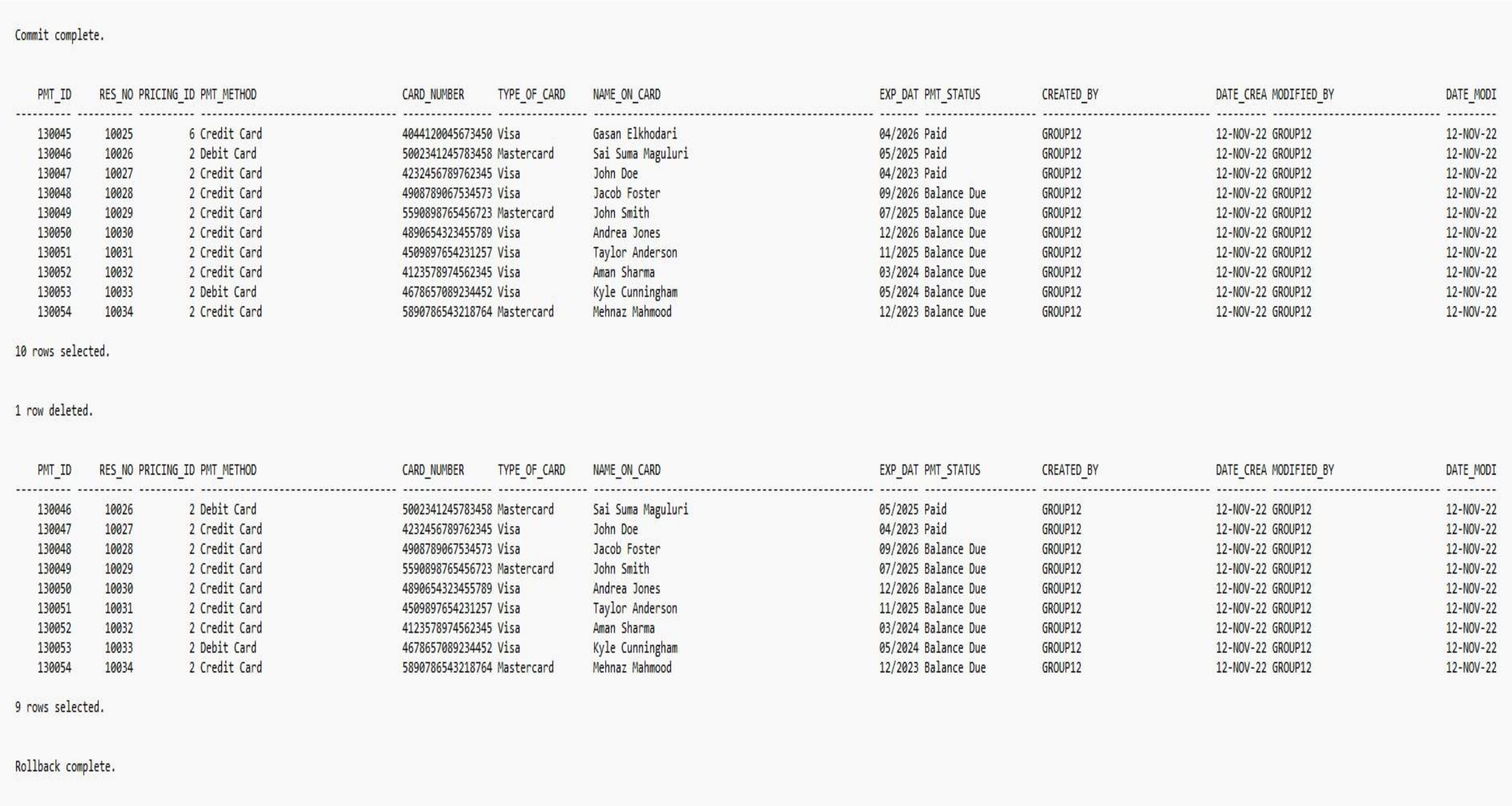


```
PRICING_ID DISC DISCOUNT_TYPE                       DISCOUNT_PERCENT THEME_PARK_PRICE ROOM_TAX_RATE TP_TICKET_TAX_RATE CREATED_BY                     DATE_CREA MODIFIED_BY                    DATE_MODI
---------- ---- ---------------------------------- ---------------- ---------------- ------------- ------------------ ------------------------------ --------- ------------------------------ ---------
         1 EB   Early Bird                                       10               55            13                  8 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
         3 SC   Senior Citizen Discount                          10               55            13                  8 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
         4 PM   Platinum Member                                  15               55            13                  8 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
         2 RT   Room Theme Park Combo                            20               55            13                  8 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
```

Q10.

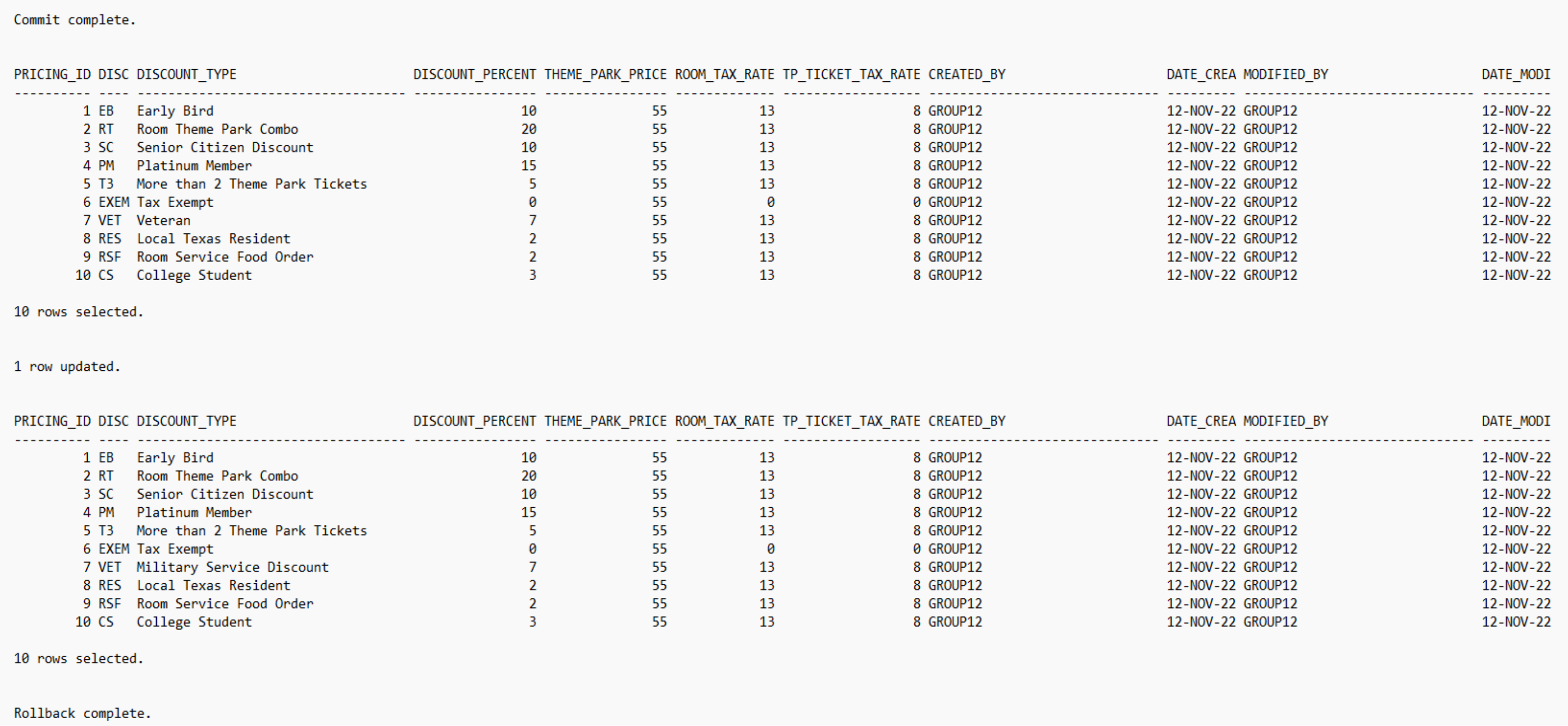
```
L_NAME                LENGTH(L_NAME)
-------------------------
Elkhodari                  9
Maguluri                   8
Doe                        3
Foster                     6
Smith                      5
Jones                      5
Anderson                   8
Sharma                     6
Cunningham                10
Mahmood                    7

10 rows selected.
```

Q11.



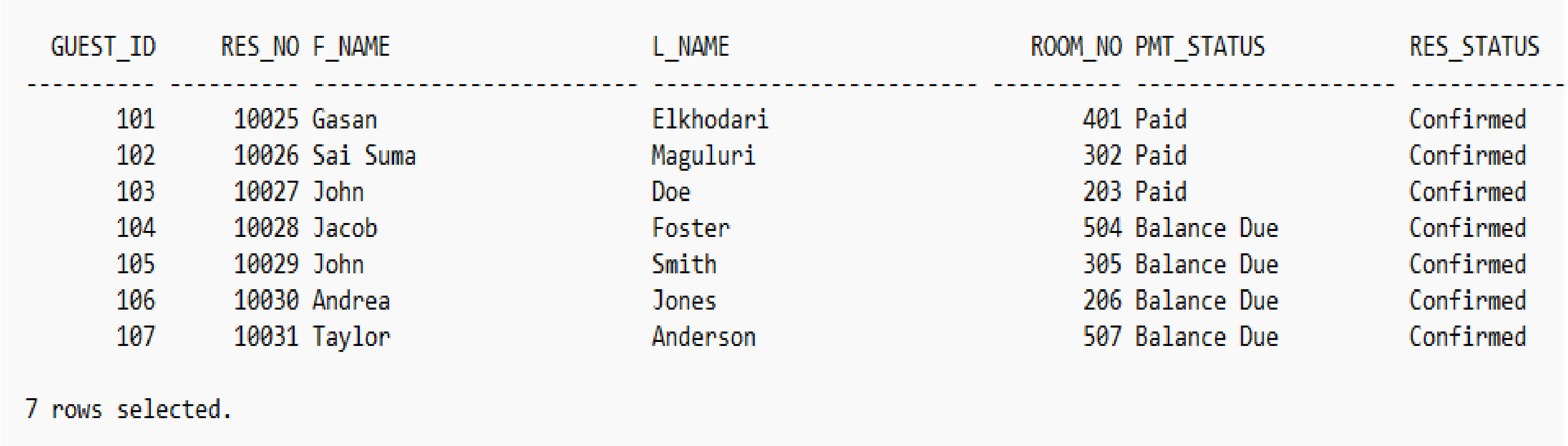
```
Commit complete.


    PMT_ID     RES_NO PRICING_ID PMT_METHOD                          CARD_NUMBER      TYPE_OF_CARD    NAME_ON_CARD                             EXP_DAT PMT_STATUS           CREATED_BY                     DATE_CREA MODIFIED_BY                    DATE_MODI
---------- ---------- ---------- ----------------------------------- ---------------- --------------- ---------------------------------------- ------- -------------------- ------------------------------ --------- ------------------------------ ---------
    130045      10025          6 Credit Card                         4044120045673450 Visa            Gasan Elkhodari                          04/2026 Paid                 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
    130046      10026          2 Debit Card                          5002341245783458 Mastercard      Sai Suma Maguluri                        05/2025 Paid                 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
    130047      10027          2 Credit Card                         4232456789762345 Visa            John Doe                                 04/2023 Paid                 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
    130048      10028          2 Credit Card                         4908789067534573 Visa            Jacob Foster                             09/2026 Balance Due          GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
    130049      10029          2 Credit Card                         5590898765456723 Mastercard      John Smith                               07/2025 Balance Due          GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
    130050      10030          2 Credit Card                         4890654323455789 Visa            Andrea Jones                             12/2026 Balance Due          GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
    130051      10031          2 Credit Card                         4509897654231257 Visa            Taylor Anderson                          11/2025 Balance Due          GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
    130052      10032          2 Credit Card                         4123578974562345 Visa            Aman Sharma                              03/2024 Balance Due          GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
    130053      10033          2 Debit Card                          4678657089234452 Visa            Kyle Cunningham                          05/2024 Balance Due          GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
    130054      10034          2 Credit Card                         5890786543218764 Mastercard      Mehnaz Mahmood                           12/2023 Balance Due          GROUP12                        12-NOV-22 GROUP12                        12-NOV-22

10 rows selected.


1 row deleted.


    PMT_ID     RES_NO PRICING_ID PMT_METHOD                          CARD_NUMBER      TYPE_OF_CARD    NAME_ON_CARD                             EXP_DAT PMT_STATUS           CREATED_BY                     DATE_CREA MODIFIED_BY                    DATE_MODI
---------- ---------- ---------- ----------------------------------- ---------------- --------------- ---------------------------------------- ------- -------------------- ------------------------------ --------- ------------------------------ ---------
    130046      10026          2 Debit Card                          5002341245783458 Mastercard      Sai Suma Maguluri                        05/2025 Paid                 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
    130047      10027          2 Credit Card                         4232456789762345 Visa            John Doe                                 04/2023 Paid                 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
    130048      10028          2 Credit Card                         4908789067534573 Visa            Jacob Foster                             09/2026 Balance Due          GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
    130049      10029          2 Credit Card                         5590898765456723 Mastercard      John Smith                               07/2025 Balance Due          GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
    130050      10030          2 Credit Card                         4890654323455789 Visa            Andrea Jones                             12/2026 Balance Due          GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
    130051      10031          2 Credit Card                         4509897654231257 Visa            Taylor Anderson                          11/2025 Balance Due          GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
    130052      10032          2 Credit Card                         4123578974562345 Visa            Aman Sharma                              03/2024 Balance Due          GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
    130053      10033          2 Debit Card                          4678657089234452 Visa            Kyle Cunningham                          05/2024 Balance Due          GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
    130054      10034          2 Credit Card                         5890786543218764 Mastercard      Mehnaz Mahmood                           12/2023 Balance Due          GROUP12                        12-NOV-22 GROUP12                        12-NOV-22

9 rows selected.


Rollback complete.
```

Q12.



```
Commit complete.


PRICING_ID DISC DISCOUNT_TYPE                       DISCOUNT_PERCENT THEME_PARK_PRICE ROOM_TAX_RATE TP_TICKET_TAX_RATE CREATED_BY                     DATE_CREA MODIFIED_BY                    DATE_MODI
---------- ---- ----------------------------------- ---------------- ---------------- ------------- ------------------ ------------------------------ --------- ------------------------------ ---------
         1 EB   Early Bird                                        10               55            13                  8 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
         2 RT   Room Theme Park Combo                             20               55            13                  8 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
         3 SC   Senior Citizen Discount                           10               55            13                  8 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
         4 PM   Platinum Member                                   15               55            13                  8 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
         5 T3   More than 2 Theme Park Tickets                     5               55            13                  8 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
         6 EXEM Tax Exempt                                         0               55             0                  0 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
         7 VET  Veteran                                            7               55            13                  8 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
         8 RES  Local Texas Resident                               2               55            13                  8 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
         9 RSF  Room Service Food Order                            2               55            13                  8 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
        10 CS   College Student                                    3               55            13                  8 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22

10 rows selected.


1 row updated.


PRICING_ID DISC DISCOUNT_TYPE                       DISCOUNT_PERCENT THEME_PARK_PRICE ROOM_TAX_RATE TP_TICKET_TAX_RATE CREATED_BY                     DATE_CREA MODIFIED_BY                    DATE_MODI
---------- ---- ----------------------------------- ---------------- ---------------- ------------- ------------------ ------------------------------ --------- ------------------------------ ---------
         1 EB   Early Bird                                        10               55            13                  8 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
         2 RT   Room Theme Park Combo                             20               55            13                  8 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
         3 SC   Senior Citizen Discount                           10               55            13                  8 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
         4 PM   Platinum Member                                   15               55            13                  8 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
         5 T3   More than 2 Theme Park Tickets                     5               55            13                  8 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
         6 EXEM Tax Exempt                                         0               55             0                  0 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
         7 VET  Military Service Discount                          7               55            13                  8 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
         8 RES  Local Texas Resident                               2               55            13                  8 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
         9 RSF  Room Service Food Order                            2               55            13                  8 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22
        10 CS   College Student                                    3               55            13                  8 GROUP12                        12-NOV-22 GROUP12                        12-NOV-22

10 rows selected.


Rollback complete.
```

Q13.



```
  GUEST_ID     RES_NO F_NAME                   L_NAME                    ROOM_NO PMT_STATUS          RES_STATUS
---------- ---------- ------------------------ ------------------------- ---------- -------------------- ------------
       101      10025 Gasan                    Elkhodari                        401 Paid                 Confirmed
       102      10026 Sai Suma                 Maguluri                         302 Paid                 Confirmed
       103      10027 John                     Doe                              203 Paid                 Confirmed
       104      10028 Jacob                    Foster                           504 Balance Due          Confirmed
       105      10029 John                     Smith                            305 Balance Due          Confirmed
       106      10030 Andrea                   Jones                            206 Balance Due          Confirmed
       107      10031 Taylor                   Anderson                         507 Balance Due          Confirmed

7 rows selected.
```

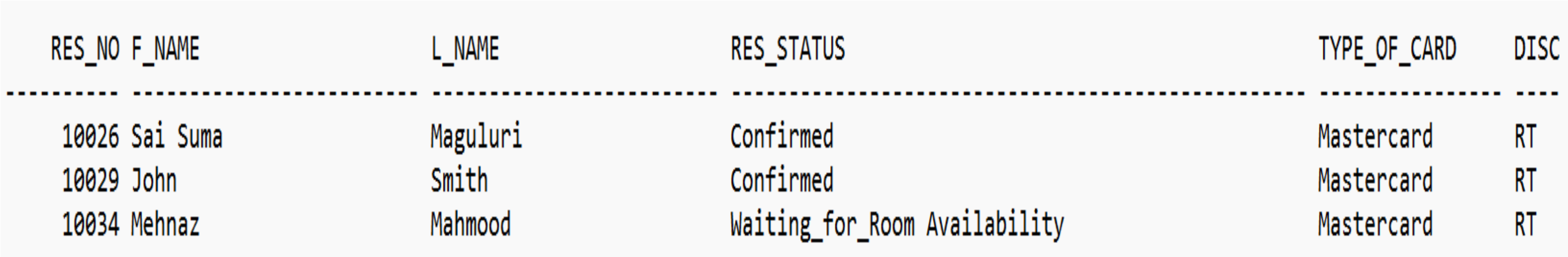
Q14.

```
  ROOM_NO F_NAME                L_NAME    ROOM_RATE PMT_STATUS
------- ------------------------- ------------------------- ---------- ----------------------
      408 Aman                  Sharma          $180        Balance Due
      504 Jacob                 Foster          $225        Balance Due
      507 Taylor                Anderson        $230        Balance Due
      510 Mehnaz                Mahmood         $230        Balance Due
```

Q15.

```
ROOM_CATEGORY              NUM_OCCURRENCES
--------------------       ----------
Executive Suite              3
Standard Twin                3
```

Q16.

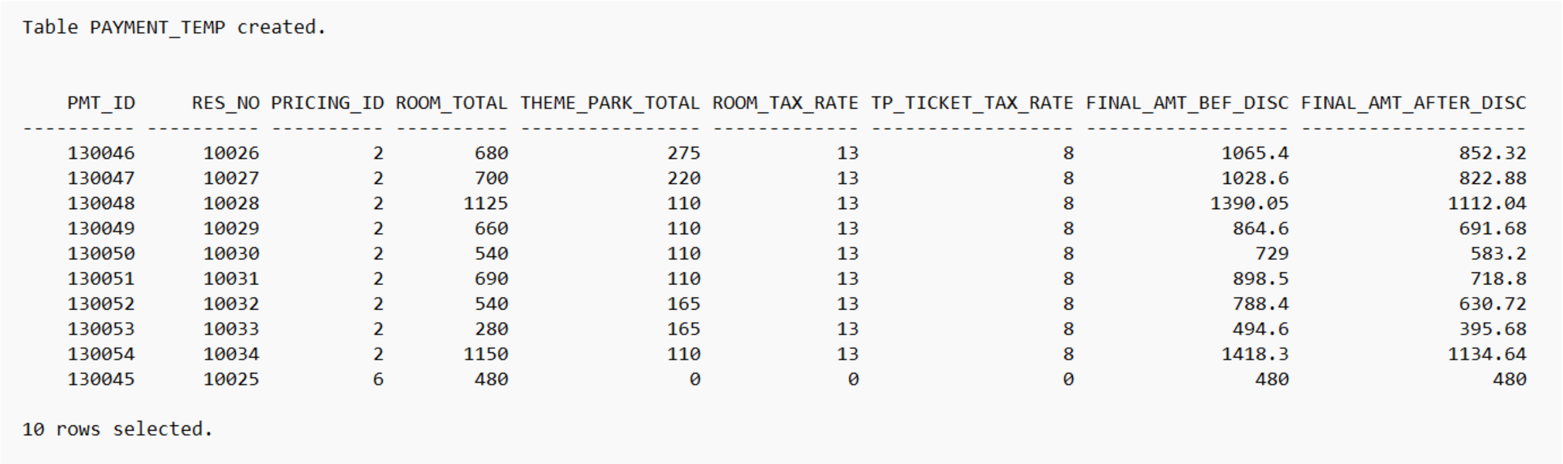


```
    RES_NO F_NAME                   L_NAME                   RES_STATUS                                         TYPE_OF_CARD    DISC
---------- ------------------------ ------------------------ -------------------------------------------------- --------------- ----
     10026 Sai Suma                 Maguluri                 Confirmed                                          Mastercard      RT
     10029 John                     Smith                    Confirmed                                          Mastercard      RT
     10034 Mehnaz                   Mahmood                  Waiting_for_Room Availability                      Mastercard      RT
```

Q17.

```
GUEST_ID F_NAME          L_NAME      NO_OF_GUESTS NO_OF_TP_TICKETS PLA
-------    -------------------- ------------- --------------           -----------------               --
      102 Sai Suma             Maguluri      5                 5                           Yes
108                   Aman            Sharma            4                    3      Yes
109                  Kyle            Cunningham        3                    3      Yes
```

Q18.



```
Table PAYMENT_TEMP created.


    PMT_ID     RES_NO PRICING_ID ROOM_TOTAL THEME_PARK_TOTAL ROOM_TAX_RATE TP_TICKET_TAX_RATE FINAL_AMT_BEF_DISC FINAL_AMT_AFTER_DISC
---------- ---------- ---------- ---------- ---------------- ------------- ------------------ ------------------ --------------------
    130046      10026          2        680              275            13                  8             1065.4               852.32
    130047      10027          2        700              220            13                  8             1028.6               822.88
    130048      10028          2       1125              110            13                  8            1390.05              1112.04
    130049      10029          2        660              110            13                  8              864.6               691.68
    130050      10030          2        540              110            13                  8                729                583.2
    130051      10031          2        690              110            13                  8              898.5                718.8
    130052      10032          2        540              165            13                  8              788.4               630.72
    130053      10033          2        280              165            13                  8              494.6               395.68
    130054      10034          2       1150              110            13                  8             1418.3              1134.64
    130045      10025          6        480                0             0                  0                480                  480

10 rows selected.
```

Q19.

```
   RES_NO F_NAME            L_NAME            DISC ROOM_RATE       ROOM_TOTAL
  -------  -------------------------- ----------------------------- -----      -----------------          -------
    10025 Gasan                 Elkhodari              EXEM $160               480
```

Q20.

Table PAYMENT_TEMP1 created.

```
PLA      COUNT_OF_CUSTOMERS_WHO_SPEND_ABOVE_AVERAGE
--          -----------------------------------------------
Yes                                         2
No                                          2
```