use myrestaurent ; #use the database

#retrive all the information from any table .-select statement

select \* from country;

#sorting the data in asc and descending order using order by clause by default asc order of order by

select Hotel\_ID from specific\_hotel\_feature order by Hotel\_ID desc;

#Filtering the data

#select distinct data :remove duplicates data from the table

select distinct Standard\_Room\_Rate from room\_types;

#LIMIT-fetch limited number of rows.

select distinct Standard\_Room\_Rate from room\_types limit 3;

select \* from hotel\_features;

#where caluse is use to select the data based on where condition

select Feature\_Code\_Description from hotel\_features where

Feature\_Code\_Description='gym' and 'Swimming Pool';

select Feature\_Code\_Description from hotel\_features where

Feature\_Code\_Description='gym' or 'Swimming Pool';

#between operator :select data within a range of values.

#check 5 most booked hotels in the chain in a period

select count(ra.Room\_ID) as number\_of\_rooms\_booked,Hotel\_Name

from room\_availability ra join rooms rs on ra.ROOM\_ID=rs.ROOM\_ID

join hotels ht on ht.Hotel\_ID=rs.Hotel\_ID

where Room\_Availability\_Status='Booked' and

Day\_Date between '2019/02/01' and '2019/02/07'

group by Hotel\_Name

order by number\_of\_rooms\_booked desc

limit 5;

#case 2 -the managers can check the number of rooms available by hotel in specific days

select count(ra.Room\_ID) as number\_of\_rooms\_availbale

,hotel\_name from room\_availability ra join rooms rm on

ra.Room\_ID=rm.Room\_ID

join hotels ht on ht.Hotel\_ID=rm.Hotel\_ID

where room\_availability\_status='available'

and day\_date ='2019/02/07'

group by hotel\_name

order by hotel\_name;

#case 3 check number of bookings in a period and by hotel

select count(Booking\_ID) as number\_of\_bookings,hotel\_name

from booking\_insert bik join rooms rm on

bik.Room\_ID=rm.Room\_ID

join hotels ht on rm.Room\_ID=ht.Hotel\_ID

and date\_From BETWEEN '2019/02/01' and '2019/02/07'

group by hotel\_name

order by hotel\_name;

#queries for customer

#case 1 Id is not auto\_increment(will give error if the customer\_id alreday exists

#want to register customer to system

INSERT INTO customers(customer\_lastname,customer\_firstname,address,country\_code,city\_code,age)

VALUES('james','bond','cherry avenue','usa','sc','25');

select \* from customers;

#case 2 want to update customer information want to age of customer with id 19 to 26

UPDATE customers SET age=26 where Customer\_ID=19;

SELECT \* FROM customers ;

# case 3 check booking history of a customer

select \* from booking\_insert Where Booking\_ID=1 ;

#case 4 select those data where description ='processing '

select Booking\_status\_code from booking\_status where

Booking\_status\_description="Processing";

#queries for Reservation staff(salesperson)

#case 1. reservation staff can check how many rooms are avilable for specific period by in the date and hotel id .

select count(room\_availability.Room\_ID), room\_availability.Day\_Date

from rooms inner join room\_availability on rooms.Room\_ID = room\_availability.Room\_ID

where (room\_availability.Room\_Availability\_Status='Available') and (rooms.Hotel\_ID = '1') and

(room\_availability.Day\_Date = '2019-02-07';

# Reservation staff can check the room rates in asc or desc order lower to higher rate

select Standard\_Room\_Rate

from room\_types order by Standard\_Room\_Rate ;

#want to update the table name

ALTER TABLE booking\_insert RENAME TO booking;

#case 3 Front can use the customer reservation number and customer id to check -in the customers to correct room

select rm.Room\_Number from booking bik join rooms rm on

bik.Room\_ID=rm.Room\_ID

WHERE bik.booking\_ID='10' AND rm.Room\_ID='4';

#CREATE TRIGGERS

#Create a new table Booking\_insert ;

drop table if exists Booking\_Insert1;

CREATE TABLE Booking\_Insert1 (

Booking\_ID INT NOT NULL,

Room\_ID VARCHAR(45) NULL,

Date\_From DATE NULL,

Date\_To DATE NULL);

Drop trigger if exists Bookings\_AFTER\_INSERT;

CREATE DEFINER = CURRENT\_USER TRIGGER Bookings\_AFTER\_INSERT

AFTER INSERT ON Bookings

FOR EACH ROW

INSERT INTO Booking\_Insert1 SELECT Booking\_ID, Room\_ID, Date\_From, Date\_To

FROM Bookings WHERE Booking\_ID = NEW.Booking\_ID;

# Check if Booking\_ID doesn't exists in Bookings, insert new values

INSERT INTO Booking(Booking\_ID, Room\_ID, Date\_From, Date\_To)

SELECT \* FROM (SELECT 25,2001,'2019/02/01','2019/02/03') as b1

WHERE NOT EXISTS (

SELECT Booking\_ID FROM Booking WHERE Booking\_ID = 25

) LIMIT 1;

INSERT INTO Booking(Booking\_ID, Room\_ID, Date\_From, Date\_To)

SELECT \* FROM (SELECT 6,1002,'2019/02/05','2019/02/06') as b1

WHERE NOT EXISTS (

SELECT Booking\_ID FROM Booking WHERE Booking\_ID = 27

) LIMIT 1;

select \* from Booking;

select \* from Booking\_Insert1; # check whether the trigger works