I) Project Name :

Travel Agency Database

II) Project Scope :

The Travel Agency System offers various options for the customers travel. A database structure should be designed for logging in for customer and agency, based on the flow of ,occupancy of a hotel's room capacity, booking rooms, and there should also be restaurant and flight reservation sections. The application has 4 main modules. Database design should be made in accordance with these modules.

1. Definitions
2. Introduction to the Application
3. Reservation Procedures
4. Post-Reservation Processes

1. DEFINITIONS

These are the preliminary definitions that must be made within the program in order for the application to work.

These:

- Hotel Description
- Room Definitions
- Room Type
- Room Capacity
- User Types
- Payment Types
- Personnel Definitions
- Personnel Types
- Restaurant Names
- Flight Destinations

• Vehicle Definitions

• Vehicle Types

1 a. Hotel Description

These are the areas where the definitions of information about the hotel will be made. Hotel's name, address, city, telephone, fax number, website address, the directory where the photos are saved are recorded with the data to be entered in the relevant fields. Definitions related to the hotel will be made in the hotel description table.

1b. Room Descriptions

It is the area where the room definitions of the hotel are made. It takes values according to the room numbering in the hotel. For example, it is named with room numbers such as 101,102. A room definition table is needed where information about the room can be recorded.

1c. Room Type

It is the area where the types of hotel rooms are recorded. It is recorded with room types such as single, double, suite, family room and their definitions.

1d. Room Capacity

Depending on the type of hotel room, its capacity will vary. While reservations can be made for 1 person for a single room, values such as the option to stay for 2 people for a double room, 4 people for a suite room, and 5 people for a family room.

1e. User Types

Different types of users such as agents, guests, registered users, and hotel personnel can be connected to the system. Depending on the types of users to be connected, the screens that can be seen will change and the authorization levels will differ. While entering the agency, you will be able to make reservations on behalf of the guest with special prices, while the guest will be able to see a limited number of areas with their own price. When the hotel staff is connected, all screens will be open to them and they will be able to log in to the application with full authorization. All this information is in the user table to be saved.

1f. Payment Types

Payment can be made after the room is booked. For this, various payment types should be defined. There should be payment types such as bank transfer, credit card payment. For this, the payment type table will be used.

1g. Staff Definitions

Hotel staff are required to deal with reservations, check and review records. For this reason, the relevant personnel should be defined in the table.

1h. Staff Types

Staff type of hotel staff should be defined. Personnel in departments such as the front desk and accounting will keep the reservations. For this reason, personnel type table will be stored.

1i. Vehicle Descriptions

After the reservation, the vehicle will be sent according to the size of the arriving group. For this reason, the vehicles belonging to the hotel should be defined together with their types.

1j. Vehicle Type Definitions

If the group that will make a reservation will come by bus, the definitions such as bus or automobile should be entered in the relevant table.

1k. Restaurant Names

In travel system a vistor can go to a restaurant and eat something. So we must keep the restaurant names.

1l. Destinations

Customers can travel with airlines and can go from one destination to other. So destination names must be identify in travel database. This destination names can be citties or countries.

2. APPLICATION WORKFLOW

There should be 2 types of login to the application.

Agency and Hotel Entrance

Guest login

2 a. Agency+Hotel Entry

Agency and Hotel Staff will enter the reservation application with similar authorizations. The only difference is that the hotel personnel have admin rights and can make all the definitions of the agency. The authority to record in the relevant agency table will be with the hotel staff. The agency official, on the other hand, will have the right to register on behalf of the guest when he logs into the system. Agency information will be logged at the reservation stage. The agency will log into the system with its own password.

2.b. Guest Login

When the guest logs in, they will be able to see the reservation and payment screens. The guest will define a password during the registration process and will log in and confirm with this password. If he is going to confirm his reservation and confirm his reservation, he will need to define his password and other personal information in advance. You will need to enter information such as name, surname, gender, phone, address. The person who enters the system as a guest can also start directly from the reservation stage. In other words, by entering the date range, it can also proceed by choosing how many rooms are needed for the relevant nights.

3. Reservation Transactions

The working logic of the reservation process will be as follows. After the person entering the application as a guest selects the start and end dates and enters how many rooms he needs or how many people there are, the system will run a coding on which rooms are available through the codes running on the database or the application. The coding will save the date range entered by

the guest and the room requirement in the database. This will be the pre-booking process. If a sufficient number of rooms are found as a result of the employee coding, the system will reserve the relevant rooms for the person with a process called last reservation or option. These rooms, which are optional in the system, will be registered to that person after the guest or the agency making the reservation in the system approves the transaction. For this, the Reservation End table will be used. In addition, relevant tables will be used for aircraft and restaurant operations.

4. Post-Booking Transactions

Customers will be invoiced after booking. Payment types will be determined and an invoice will be issued after the payment, with choices such as bank transfer or credit card, with pricing in accordance with the number of nights the customer has stayed. In addition, customers may want to enter the relevant application and enter their comments and give a grade. In this case, they will open the relevant screen of the application and give the necessary note. They will write their comments.

III) Entity Names

The database structure will be established in a relational design, with the primary keys to be stored in each table and the secondary key to refer to the other table.

1. Hotel_Guest

Information about the guest is stored. Name, surname, address etc. information is in this table.

Attributes:

Guest_Id , Guest_City, Guest_Password, Guest_Name , Guest_Lastname , Guest_Phone, Guest_Email,Guest_Active,Guest_Company_Name , Guest_Gender,

Guest_Address , Guest_System_Name

2. Hotel_Agent

Information about the agency is stored. Contains agency data.

Attributes:

Agency_Id , Agency_Company_Name , Agency_Email , Agency_System_Name ,

Agent_Address , Agent_Password ,Agency_Sehir , Agency_Type ,Agent_Phone, Agent_Active

3. Hotel_Description

Information about the hotel description is stored. It contains information such as name, address and phone.

Attributes:

Hotel_Id, Hotel_Name,Hotel_Firma_Name, Hotel_Address, Hotel_City,Hotel_Phone,

Hotel_Fax ,Hotel_Email,Hotel_Web_Adres , Hotel_Photo_Directory

4. Hotel_Room_Type

The room types in the hotel are stored.

Attributes:

Room_Type_Id, Room_Type, Room_Type_Description, Room_Type ,Photo_Directory

5. Hotel_Room

Room definitions in the hotel are stored in this table along with their types.

Attributes:

Room_Id, Hotel_Id, Room_Name, Room_Description, Room_Type_Id

6. Hotel_Reservation_On

Pre-reservation information is stored. The system runs the code according to the date range entered by the guest.

Attributes:

On_Rzr_Id , Hotel_Id, Guest_Id, Rzr_Start_Date, Rzr_End_Date , Rzr_Room_Number , Rzr_Active , Agency_id

7. Hotel_Reservation_Last

Room information coming from the system after reservation is stored in this table.

Attributes:

Last_Rzr_Id , On_Rzr_Id , Room_Id

8. Hotel_Payment_Type

It includes the payment types used in the hotel.

Attributes:

Payment_Type_Id , Payment_Type, Payment_Type_Description

9. Hotel_Room_Comment

Comments are entered after the stay and scoring is done.

Attributes:

Comment_Id, Comment_Value, End_Rzr_Id

10. Hotel_Room_Invoice

Reservation related invoices are stored

Attributes:

Invoice_Id , Invoice_Value , Invoice_Date , Last_Rzr_Id , Payment_Type_Id

11. Hotel_Staff_Tip

Hotel staff types are stored in this table. (Reservation staff, front desk etc.)

Attributes:

Staff_Type_Id, Staff_Type , Staff_Type_Description

12. Hotel_Staff

Hotel Personnel is defined in this table and stored with personnel type values.

Attributes:

Staff_Id ,Hotel_Id ,Personnel_Name ,Staff_Surname, Staff_Type_Id

13. Hotel_Vehicle_Type

After the reservation made to the hotel, the types of vehicles to meet the convoys are stored.

Attributes:

Vehicle_Type_Id ,Vehicle_Type , Vehicle_Type_Description

14. Hotel_Car

After the reservation made to the hotel, the vehicle definitions that will meet the convoys are made in this table.

Attributes:

Vehicle_Id , Hotel_Id , Vehicle_Name , Vehicle_Description, Vehicle_Type_Id

15. Hotel_Log

The logs of the entire system will be stored here with the triggers to be taken from the relevant tables.

Attributes:

Log_Id ,Log_Description , Log_Date

16. Flight Destination

The flight destination information is stored in this table.

Attributes:

Destination_Id , Destination_name

17. Airline Name

The flight information of the guests is stored in this table.

Attributes:

Airline_Id, Airline_Name

18. Flight

The flight information of the guests is stored in this table.

Attributes:

Flight_Id , Guest_Id , Airline_Id , Flight_Date , Flight_Name , Seat_Numbers , from_id , to_id

19. Restaurant Name

Includes restaurant service information to be offered to guests.

Attributes:

Reserve_Id , Restaurant_Name, Restaurant_Address

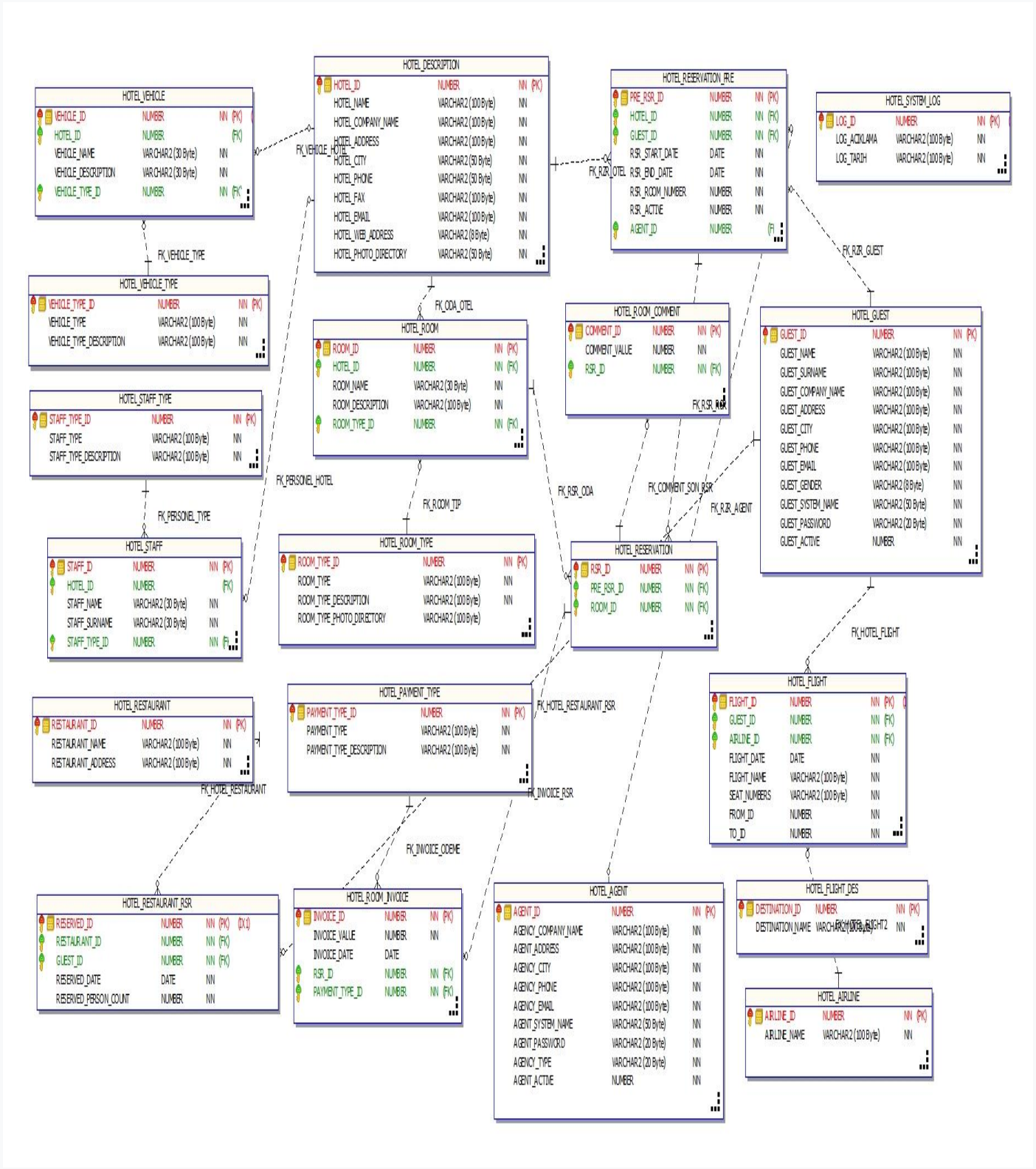
20. Restaurant Reservation

Includes restaurant service information to be offered to guests.

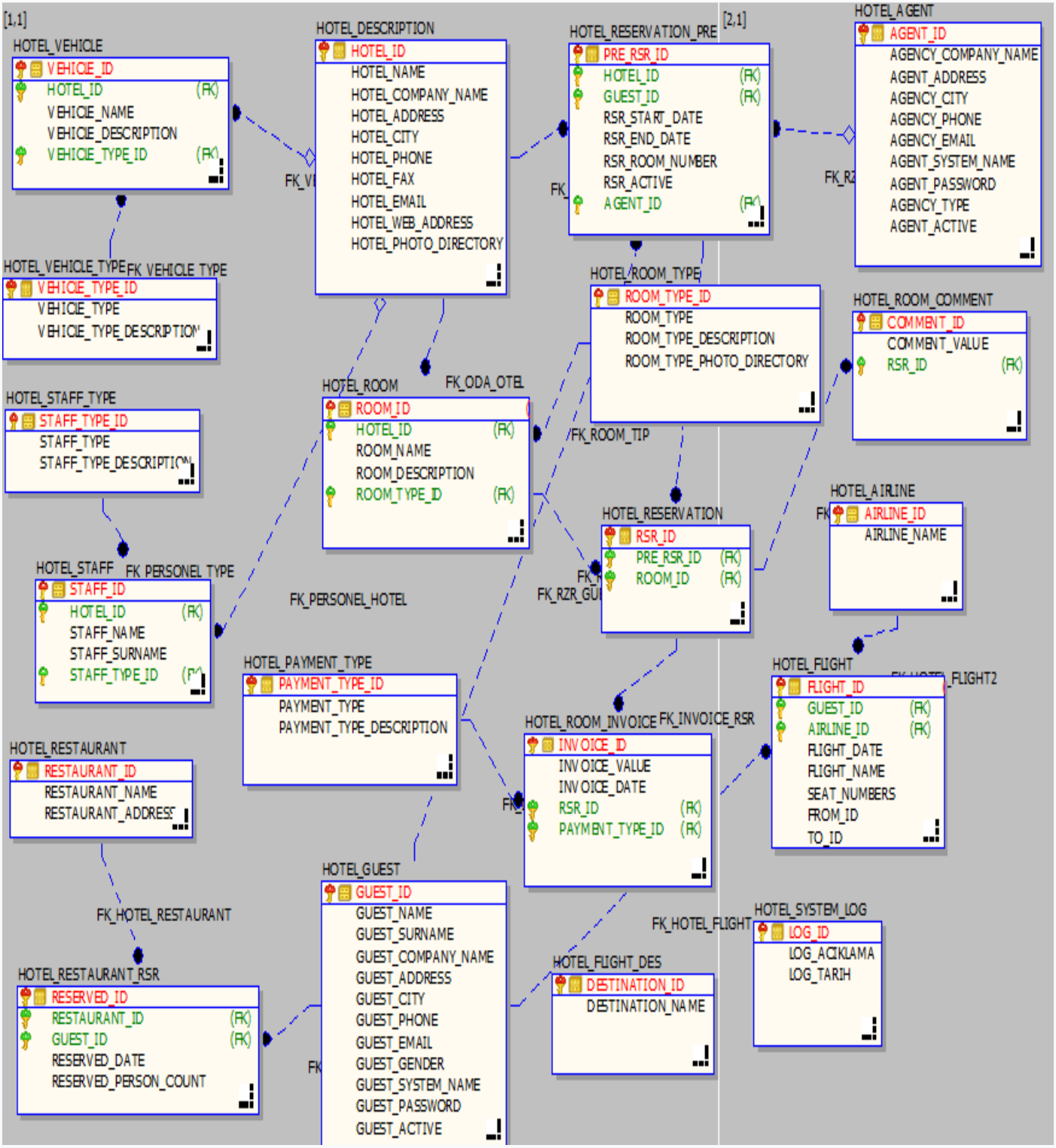
Attributes:

Reserve_Id , Guest_Id , Reserve_Date , Reserved_Person_Number

IV) Entity RelationShip Diagram



V) Entity RelationShip Diagram – ERWIN IDEF1X



VI) Database Selection – Oracle

We choose Oracle Database. Oracle Database is an advanced relational database system that enables large amounts of data to be stored and accessed in multi-user environments. Despite its large transaction size, Oracle Database meets the needs of businesses in the database management process with its features such as performing transactions quickly and securely, offering different version options suitable for needs, ensuring the security of stored data with various methods. In this article, we have explained the Oracle Database and its versions for you and talked about the features that make it stand out.

Oracle database is a relational database management system. As a relational database, Oracle Database is built on SQL, which is also used in the processes of managing, designing, preparing and querying data. Oracle Database not only stores data, but also offers a web-based software development environment.

Offering performance, scalability and security both in the enterprise and in the cloud, Oracle Database can achieve high performance for the most demanding analytics and operations workloads. In addition to these features, it stands out with its multi-model, multi-task, machine learning access and multi-user development capabilities.

VII) DDL Statements For Tables

```
CREATE TABLE Hotel_Guest
(
   Guest_Id              NUMBER NOT NULL,
   Guest_Name            VARCHAR2 (100) NOT NULL,
   Guest_Surname         VARCHAR2 (100) NOT NULL,
   Guest_Company_Name    VARCHAR2 (100) NOT NULL,
   Guest_Address         VARCHAR2 (100) NOT NULL,
   Guest_City            VARCHAR2 (100) NOT NULL,
   Guest_Phone           VARCHAR2 (100) NOT NULL,
   Guest_Email           VARCHAR2 (100) NOT NULL,
   Guest_Gender          VARCHAR2 (8) NOT NULL,
   Guest_System_Name     VARCHAR2 (50) NOT NULL,
   Guest_Password        VARCHAR2 (20) NOT NULL,
   Guest_Active          NUMBER DEFAULT ( (1)) NOT NULL,
   CONSTRAINT PK_Hotel_Guest PRIMARY KEY (Guest_Id)
);
```

```
CREATE TABLE Hotel_Agent
(
   Agent_Id              NUMBER NOT NULL,
   Agency_Company_Name   VARCHAR2 (100) NOT NULL,
   Agent_Address         VARCHAR2 (100) NOT NULL,
   Agency_City           VARCHAR2 (100) NOT NULL,
   Agency_Phone          VARCHAR2 (100) NOT NULL,
   Agency_Email          VARCHAR2 (100) NOT NULL,
   Agent_System_Name     VARCHAR2 (50) NOT NULL,
   Agent_Password        VARCHAR2 (20) NOT NULL,
   Agency_Type           VARCHAR2 (20) NOT NULL,
   Agent_Active          NUMBER DEFAULT ( (1)) NOT NULL,
   CONSTRAINT PK_Otel_Agent PRIMARY KEY (Agent_Id)
);

CREATE TABLE Hotel_Description
(
   Hotel_Id                NUMBER NOT NULL,
   Hotel_Name              VARCHAR2 (100) NOT NULL,
   Hotel_Company_Name      VARCHAR2 (100) NOT NULL,
   Hotel_Address           VARCHAR2 (100) NOT NULL,
   Hotel_City              VARCHAR2 (50) NOT NULL,
   Hotel_Phone             VARCHAR2 (50) NOT NULL,
   Hotel_Fax               VARCHAR2 (100) NOT NULL,
   Hotel_Email             VARCHAR2 (100) NOT NULL,
   Hotel_Web_Address       VARCHAR2 (50) NOT NULL,
   Hotel_Photo_Directory   VARCHAR2 (50) NOT NULL,
   CONSTRAINT PK_Otel_Tanim PRIMARY KEY (Hotel_Id)
);


CREATE TABLE Hotel_Room_Type
(
   Room_Type_Id                NUMBER NOT NULL,
   Room_Type                   VARCHAR2 (100) NOT NULL,
   Room_Type_Description       VARCHAR2 (100) NOT NULL,
   Room_Type_Photo_Directory   VARCHAR2 (100) NULL,
   CONSTRAINT PK_Room_Tip PRIMARY KEY (Room_Type_Id)
);


CREATE TABLE Hotel_Room
(
   Room_Id            NUMBER NOT NULL,
   Hotel_Id           NUMBER NOT NULL,
   Room_Name          VARCHAR2 (30) NOT NULL,
   Room_Description   VARCHAR2 (100) NOT NULL,
   Room_Type_Id       NUMBER NOT NULL,
   CONSTRAINT PK_Room PRIMARY KEY (Room_Id),
   CONSTRAINT FK_Oda_Otel FOREIGN KEY
      (Hotel_Id)
       REFERENCES Hotel_Description (Hotel_Id),
   CONSTRAINT FK_Room_Tip FOREIGN KEY
      (Room_Type_Id)
       REFERENCES Hotel_Room_Type (Room_Type_Id)
);


CREATE TABLE Hotel_Reservation_Pre
(
   Pre_Rsr_Id        NUMBER NOT NULL,
   Hotel_Id          NUMBER NOT NULL,
   Guest_Id          NUMBER NOT NULL,
   Rsr_Start_Date    DATE NOT NULL,
   Rsr_End_Date      DATE NOT NULL,
```

```
    Rsr_Room_Number    NUMBER NOT NULL,
    Rsr_Active         NUMBER DEFAULT ( (1)) NOT NULL,
    Agent_Id           NUMBER NULL,
    CONSTRAINT PK_Rzr_On PRIMARY KEY (Pre_Rsr_Id),
    CONSTRAINT FK_Rzr_Otel FOREIGN KEY
       (Hotel_Id)
        REFERENCES Hotel_Description (Hotel_Id),
    CONSTRAINT FK_Rzr_Guest FOREIGN KEY
       (Guest_Id)
        REFERENCES Hotel_Guest (Guest_Id),
    CONSTRAINT FK_Rzr_Agent FOREIGN KEY
       (Agent_Id)
        REFERENCES Hotel_Agent (Agent_Id)
);

CREATE TABLE Hotel_Reservation
(
    Rsr_Id        NUMBER NOT NULL,
    Pre_Rsr_Id    NUMBER NOT NULL,
    Room_Id       NUMBER NOT NULL,
    CONSTRAINT PK_Rsr PRIMARY KEY (Rsr_Id),
    CONSTRAINT FK_Rsr_Oda FOREIGN KEY
       (Room_Id)
        REFERENCES Hotel_Room (Room_Id),
    CONSTRAINT FK Rsr_Rsr FOREIGN KEY
       (Pre_Rsr_Id)
        REFERENCES Hotel_Reservation_Pre (Pre_Rsr_Id)
);

CREATE TABLE Hotel_Payment_Type
(
    Payment_Type_Id             NUMBER NOT NULL,
    Payment_Type                VARCHAR2 (100) NOT NULL,
    Payment_Type_Description    VARCHAR2 (100) NOT NULL,
    CONSTRAINT PK_Odeme_Tip PRIMARY KEY (Payment_Type_Id)
);

CREATE TABLE Hotel_Room_Comment
(
    Comment_Id       NUMBER NOT NULL,
    Comment_Value    NUMBER NOT NULL,
    Rsr_Id           NUMBER NOT NULL,
    CONSTRAINT PK_Room_Comment PRIMARY KEY (Comment_Id),
    CONSTRAINT FK_Comment_Son_Rsr FOREIGN KEY
       (Rsr_Id)
        REFERENCES Hotel_Reservation (Rsr_Id)
);

CREATE TABLE Hotel_Room_Invoice
(
    Invoice_Id         NUMBER NOT NULL,
    Invoice_Value      NUMBER NOT NULL,
    Invoice_Date       DATE,
    Rsr_Id             NUMBER NOT NULL,
    Payment_Type_Id    NUMBER NOT NULL,
    CONSTRAINT PK_Room_Invoice PRIMARY KEY (Invoice_Id),
    CONSTRAINT FK_Invoice_Rsr FOREIGN KEY
       (Rsr_Id)
        REFERENCES Hotel_Reservation (Rsr_Id),
    CONSTRAINT FK_Invoice_Odeme FOREIGN KEY
       (Payment_Type_Id)
        REFERENCES Hotel_Payment_Type (Payment_Type_Id)
);
```

```
CREATE TABLE Hotel_Staff_Type
(
   Staff_Type_Id            NUMBER NOT NULL,
   Staff_Type               VARCHAR2 (100) NOT NULL,
   Staff_Type_Description   VARCHAR2 (100) NOT NULL,
   CONSTRAINT PK_Personel_Tip PRIMARY KEY (Staff_Type_Id)
);


CREATE TABLE Hotel_Staff
(
   Staff_Id        NUMBER NOT NULL,
   Hotel_Id        NUMBER,
   Staff_Name      VARCHAR2 (30) NOT NULL,
   Staff_Surname   VARCHAR2 (30) NOT NULL,
   Staff_Type_Id   NUMBER NOT NULL,
   CONSTRAINT PK_Personel PRIMARY KEY (Staff_Id),
   CONSTRAINT FK_Personel_hotel FOREIGN KEY
      (Hotel_Id)
       REFERENCES Hotel_Description (Hotel_Id),
   CONSTRAINT FK_Personel_Type FOREIGN KEY
      (Staff_Type_Id)
       REFERENCES Hotel_Staff_Type (Staff_Type_Id)
);


CREATE TABLE Hotel_Vehicle_Type
(
   Vehicle_Type_Id            NUMBER NOT NULL,
   Vehicle_Type               VARCHAR2 (100) NOT NULL,
   Vehicle_Type_Description   VARCHAR2 (100) NOT NULL,
   CONSTRAINT PK_Otel_Tip PRIMARY KEY (Vehicle_Type_Id)
);

CREATE TABLE Hotel_Vehicle
(
   Vehicle_Id            NUMBER NOT NULL,
   Hotel_Id              NUMBER,
   Vehicle_NAme          VARCHAR2 (30) NOT NULL,
   Vehicle_Description   VARCHAR2 (30) NOT NULL,
   Vehicle_Type_Id       NUMBER NOT NULL,
   CONSTRAINT PK_Vehicle PRIMARY KEY (Vehicle_Id),
   CONSTRAINT FK_Vehicle_Hotel FOREIGN KEY
      (hotel_Id)
       REFERENCES Hotel_Description (hotel_Id),
   CONSTRAINT FK_Arac_Tip FOREIGN KEY
      (Vehicle_Type_Id)
       REFERENCES Hotel_Vehicle_Type (Vehicle_Type_Id)
);

CREATE TABLE Hotel_System_Log
(
   Log_Id         NUMBER NOT NULL,
   Log_Aciklama   VARCHAR2 (100) NOT NULL,
   Log_Tarih      VARCHAR2 (100) NOT NULL,
   CONSTRAINT PK_Otel_Log PRIMARY KEY (Log_Id));

CREATE TABLE Hotel_Flight_Des
(
   Destination_Id     NUMBER NOT NULL,
   Destination_name   VARCHAR2 (100) NOT NULL,
   CONSTRAINT PK_Flight_Destination PRIMARY KEY (Destination_Id)
);
```

```
CREATE TABLE Hotel_Airline
(
   Airline_Id     NUMBER NOT NULL,
   Airline_Name   VARCHAR2 (100) NOT NULL,
   CONSTRAINT PK_Hotel_Airline PRIMARY KEY (Airline_Id)
);


CREATE TABLE Hotel_Flight
(
   Flight_Id      NUMBER NOT NULL,
   Guest_Id       NUMBER NOT NULL,
   Airline_Id     NUMBER NOT NULL,
   Flight_Date    DATE NOT NULL,
   Flight_Name    VARCHAR2 (100) NOT NULL,
   Seat_Numbers   VARCHAR2 (100) NOT NULL,
   From_id        NUMBER NOT NULL,
   To_id          NUMBER NOT NULL,
   CONSTRAINT PK_Hotel_Flight PRIMARY KEY (Flight_Id),
   CONSTRAINT FK_Hotel_Flight FOREIGN KEY
      (Guest_Id)
       REFERENCES Hotel_Guest (Guest_Id),
   CONSTRAINT FK_Hotel_Flight2 FOREIGN KEY
      (Airline_Id)
       REFERENCES Hotel_Airline (Airline_Id)
);


CREATE TABLE Hotel_Restaurant
(
   Restaurant_Id        NUMBER NOT NULL,
   Restaurant_Name      VARCHAR2 (100) NOT NULL,
   Restaurant_Address   VARCHAR2 (100) NOT NULL,
   CONSTRAINT PK_Hotel_Restaurant PRIMARY KEY (Restaurant_Id)
);


CREATE TABLE Hotel_Restaurant_Rsr
(
   Reserved_Id             NUMBER NOT NULL,
   Restaurant_Id           NUMBER NOT NULL,
   Guest_Id                NUMBER NOT NULL,
   Reserved_Date           DATE NOT NULL,
   Reserved_Person_count   NUMBER NOT NULL,
   CONSTRAINT PK_Hotel_Restaurant_Rsr PRIMARY KEY (Reserved_Id),
   CONSTRAINT FK_Hotel_Restaurant_Rsr FOREIGN KEY
      (Guest_Id)
       REFERENCES Hotel_Guest (Guest_Id),
   CONSTRAINT FK_Hotel_Restaurant FOREIGN KEY
      (Restaurant_Id)
       REFERENCES Hotel_Restaurant (Restaurant_Id)
);
```

VIII) Additional DDL Statements

A) Create View

```
--View which lists information about The total number of nights that the guests booked
by the agencies stayed for more than 5 nights with their name, surname, phone and
gender information
CREATE OR REPLACE FORCE VIEW V_HOTEL_GUEST_REPORT
(GUEST_NAME, GUEST_SURNAME, GUEST_PHONE, GUEST_GENDER, TOTAL_NIGHT)
AS
  SELECT guest_name,
         guest_surname,
         guest_phone,
         guest_gender,
         SUM (rsr_end_date - rsr_start_date) total_night
    FROM hotel_guest mi, hotel_reservation_pre res_pre
   WHERE     mi.guest_Id = res_pre.guest_Id
         AND agent_Id IS NOT NULL
         AND rsr_active = 1
GROUP BY guest_name,
         guest_surname,
         guest_phone,
         guest gender
  HAVING SUM (rsr_end_date - rsr_start_date) > 5;


--View which lists the  name, surname and phone number of customers with a payment type
credit card and the total amount they paid with the room they stayed.
CREATE OR REPLACE FORCE VIEW V_HOTEL_INVOICE_LIST
(FULL_NAME, GUEST_PHONE, TOTAL_INVOICE_AMOUNT)
AS
  SELECT guest_name || ' ' || guest_surname full_name,
         guest_phone,
         SUM (invoice_value) total_invoice_amount
    FROM hotel_guest mi,
         hotel_reservation_pre res_pre,
         hotel_reservation res,
         hotel_room room,
         hotel_room_invoice invoice,
         hotel_payment_type TYPE
   WHERE     mi.guest_id = res_pre.guest_id
         AND res.rsr_id = invoice.rsr_id
         AND res.pre_rsr_id = res_pre.pre_rsr_id
         AND room.room_id = res.room_id
         AND TYPE.payment_type_id = invoice.payment_type_id
         AND payment_type = 'Credit Card'
         AND rsr_active = 1
GROUP BY guest_name || ' ' || guest_surname, guest_phone;
```

A2) Create Index

```
CREATE UNIQUE INDEX PK_FLIGHT_DESTINATION ON HOTEL_FLIGHT_DES
(DESTINATION_ID) LOGGING TABLESPACE USERS NOPARALLEL;

CREATE UNIQUE INDEX PK_HOTEL_AIRLINE ON HOTEL_AIRLINE
(AIRLINE_ID)LOGGING TABLESPACE USERS NOPARALLEL;

CREATE UNIQUE INDEX PK_HOTEL_FLIGHT ON HOTEL_FLIGHT
(FLIGHT_ID)LOGGING TABLESPACE USERS NOPARALLEL;

CREATE UNIQUE INDEX PK_HOTEL_GUEST ON HOTEL_GUEST
```

```
(GUEST_ID) LOGGING TABLESPACE USERS NOPARALLEL;

CREATE UNIQUE INDEX PK_HOTEL_RESTAURANT ON HOTEL_RESTAURANT
(RESTAURANT_ID) LOGGING TABLESPACE USERS NOPARALLEL;

CREATE UNIQUE INDEX PK_HOTEL_RESTAURANT_RSR ON HOTEL_RESTAURANT_RSR
(RESERVED_ID)LOGGING TABLESPACE USERS NOPARALLEL;

CREATE UNIQUE INDEX PK_ODEME_TIP ON HOTEL_PAYMENT_TYPE
(PAYMENT_TYPE_ID) LOGGING TABLESPACE USERS NOPARALLEL;

CREATE UNIQUE INDEX PK_OTEL_AGENT ON HOTEL_AGENT
(AGENT_ID) LOGGING TABLESPACE USERS NOPARALLEL;

CREATE UNIQUE INDEX PK_OTEL_LOG ON HOTEL_SYSTEM_LOG (LOG_ID)
LOGGING TABLESPACE USERS NOPARALLEL;

CREATE UNIQUE INDEX PK_OTEL_TANIM ON HOTEL_DESCRIPTION (HOTEL_ID)
LOGGING TABLESPACE USERS NOPARALLEL;

CREATE UNIQUE INDEX PK_OTEL_TIP ON HOTEL_VEHICLE_TYPE
(VEHICLE_TYPE_ID) LOGGING TABLESPACE USERS NOPARALLEL;
```

B)DML Statements

B1) Insert Into Statements

```
INSERT INTO Hotel_Guest VALUES (1,'Serkan','Zehir','Zehir
Firma','Ankara','Ankara','5537851367','serkanzehir1@hotmail.com','Erkek','serkanzehir',
'123', 1);
INSERT INTO Hotel_Guest  VALUES (2,'Onur','Demir','Demir
Firma','İstanbul','Ankara','5532227845','onurdemir1@hotmail.com','Erkek','onurdemir','2
34', 1);
INSERT INTO Hotel_Guest  VALUES (3,'Ayşe','Soyalan','Soyalan
Firma','İzmir','İzmir','5457851366','aysesoyalan@hotmail.com','Kadın','aysesoyalan','12
3', 1);
INSERT INTO Hotel_Guest  VALUES (4,'Kağan','Zehir','Zehir
Firma','Ankara','Ankara','5457861366','kağanzehir@hotmail.com','Erkek','kaganzehir','12
3', 1);
INSERT INTO Hotel_Guest VALUES (5,'Fatih','Apaydın','Apaydın
Firma','Zonguldak','Zonguldak','5531234567','fatihapaydın@hotmail.com','Erkek','fatihap
aydın','123', 1);


INSERT INTO Hotel_Agent VALUES (1,'Zehir
Acenteleri','Ankara','Ankara','3123307699','zehiracentesi@hotmail.com','zehiracente','1
23','seyahat', 1);
INSERT INTO Hotel_Agent VALUES (2,'Apaydın
Acente','Zonguldak','Zonguldak','3723124566','apaydınacentesi@hotmail.com','apaydınacen
te','123','sigorta', 1);
INSERT INTO Hotel_Agent VALUES (3,'Demir
Acente','Ankara','Ankara','3123304566','demiracente@hotmail.com','demiracente','123','s
eyahat', 1);
INSERT INTO Hotel_Agent VALUES (4,'Yüksel
Acente','İstanbul','İstanbul','2127564566','yükselacente@hotmail.cokm','yükselacente','
123','taşıma', 1);
INSERT INTO Hotel_Agent VALUES (5,'Uğraş
Acente','İzmir','İzmir','2127851367','ugrasacente@hotmail.com','ugrasacente','123','sig
orta', 1);
```

```
INSERT INTO Hotel_Description  VALUES (1,'Zehir Otel','Zehir A.Ş','Yukarı Sk.
No:4','Ankara','3122525256','03125247889','zehir@hotel.com','zehirotel.com','c:\foto');
INSERT INTO Hotel_Description  VALUES (2,'Demir Otel','Demir
A.Ş','Ankara','Ankara','3123307699','3123307698','demir@otel.com','demirotel.com','c:\f
oto');
INSERT INTO Hotel_Description  VALUES (3,'Ugras Otel','Ugras
A.Ş','İzmir','İzmir','3123304578','3123304577','ugras@otel.com','ugrasotel.cım','c:\fot
o');
INSERT INTO Hotel_Description VALUES (4,'Yüksel Otel','Yüksel
A.Ş','İstanbul','İstanbul','2124578888','2124587952','yuksel@otel.com','yukselotel.com'
,'c:\foto');
INSERT INTO Hotel_Description  VALUES (5,'Serkan Otel','Serkan
A.Ş','Zonguldak','Zonguldak','3721234578','3721234577','serkan@otel.com','serkanotel.co
m','c:\foto');


INSERT INTO Hotel_Reservation_Pre  VALUES (1, 1, 1, to_date('2021-02-01','YYYY-MM-DD'),
to_date('2021-01-02','YYYY-MM-DD'), 1, 1, 1);
INSERT INTO Hotel_Reservation_Pre VALUES (2, 2, 2, to_date('2021-02-02','YYYY-MM-DD'),
to_date('2021-03-02','YYYY-MM-DD'), 1, 1, 2);
INSERT INTO Hotel_Reservation_Pre VALUES (3, 3, 3, to_date('2021-05-01','YYYY-MM-DD'),
to_date('2021-05-07','YYYY-MM-DD'), 1, 1, 3);
INSERT INTO Hotel Reservation Pre VALUES (4, 4, 4, to_date('2021-01-02','YYYY-MM-DD'),
to_date('2021-02-01','YYYY-MM-DD'), 1, 1, 4);
INSERT INTO Hotel_Reservation_Pre  VALUES (5, 5, 5, to_date('2021-02-05','YYYY-MM-DD'),
to_date('2021-05-07','YYYY-MM-DD'), 1, 1, 1);


INSERT INTO Hotel_Room_Type  VALUES (1,'Tek Kişilik Oda','Tek kişi
kalabilir','c:\foto');
INSERT INTO Hotel_Room_Type VALUES (2,'Çift Kişilik Oda','Çift kişi
kalabilir','c:\foto');
INSERT INTO Hotel_Room_Type VALUES (3,'Aile Odası','aileler kalabilir','c:\foto');
INSERT INTO Hotel_Room_Type VALUES (4,'Suite Oda','Suite oda','c:\foto');
INSERT INTO Hotel_Room_Type VALUES (5,'Junior Suite','Junior Suite','c:\foto');


INSERT INTO Hotel_Room VALUES (1, 1,'Tek Kişilik','tek kişi kalabilir', 1);
INSERT INTO Hotel_Room VALUES (2, 2,'Çift Kişilik','çift kişi kalabilir', 2);
INSERT INTO Hotel_Room  VALUES (3, 3,'Aile','aileler kalabilir', 3);
INSERT INTO Hotel_Room VALUES (4, 4,'Suite','Suite', 4);
INSERT INTO Hotel_Room VALUES (5, 5,'Junior Suite','Junior Suite', 5);

INSERT INTO Hotel_Payment_Type  VALUES (1,'Nakit','Nakit Ödendi');
INSERT INTO Hotel_Payment_Type  VALUES (2,'Kredi Kartı','Kartla Ödendi');
INSERT INTO Hotel_Payment_Type  VALUES (3,'Taksit','5 Taksitte ödenecek');
INSERT INTO Hotel_Payment_Type  VALUES (4,'Nakit',' Ödendi');
INSERT INTO Hotel_Payment_Type  VALUES (5,'Kart','Kartla Ödendi');

INSERT INTO Hotel_Reservation  VALUES (1, 1, 1);
INSERT INTO Hotel_Reservation  VALUES (2, 2, 2);
INSERT INTO Hotel_Reservation  VALUES (3, 3, 3);
INSERT INTO Hotel_Reservation  VALUES (4, 4, 4);
INSERT INTO Hotel_Reservation  VALUES (5, 5, 5);

INSERT INTO Hotel_Room_Invoice  VALUES (1, 5000, to_date('2021-05-01','YYYY-MM-DD'), 1,
1);
INSERT INTO Hotel_Room_Invoice  VALUES (2, 700, to_date('2021-01-02','YYYY-MM-DD'), 2,
2);
INSERT INTO Hotel_Room_Invoice  VALUES (3, 300, to_date('2021-02-02','YYYY-MM-DD'), 3,
3);
INSERT INTO Hotel_Room_Invoice  VALUES (4, 400, to_date('2021-02-02','YYYY-MM-DD'), 4,
4);
```

```
INSERT INTO Hotel_Room_Invoice  VALUES (5, 500, to_date('2021-01-05','YYYY-MM-DD'), 5,
5);

INSERT INTO Hotel_Staff_Type  VALUES (1,'Banknot','Banknot');
INSERT INTO Hotel_Staff_Type VALUES (2,'SGK','SGK');
INSERT INTO Hotel_Staff_Type  VALUES (3,'SSK','SSK');
INSERT INTO Hotel_Staff_Type  VALUES (4,'Yevmiye','Yevmiye');
INSERT INTO Hotel_Staff_Type  VALUES (5,'Devamlı','Devamlı');

INSERT INTO Hotel_Staff  VALUES (1, 2,'Serkan','Zehir', 1);
INSERT INTO Hotel_Staff VALUES (2, 2,'Onur','Demir', 2);
INSERT INTO Hotel_Staff  VALUES (3, 3,'Ahmet','Keskin', 3);
INSERT INTO Hotel_Staff  VALUES (4, 4,'Ali','Taşdemir', 4);
INSERT INTO Hotel_Staff  VALUES (5, 5,'Ahmet','Ersoy', 5);


INSERT INTO Hotel_Vehicle_Type  VALUES (1,'Ticari','Ticari');
INSERT INTO Hotel_Vehicle_Type  VALUES (2,'Kamyonet','Kamyonet');
INSERT INTO Hotel_Vehicle_Type  VALUES (3,'Pick-up','Pick-up');
INSERT INTO Hotel_Vehicle_Type  VALUES (4,'Binek','Binek');
INSERT INTO Hotel_Vehicle_Type  VALUES (5,'Sedan','Sedan');


INSERT INTO Hotel_Vehicle  VALUES (1, 1,'Renault','Binek araç', 1);
INSERT INTO Hotel Vehicle  VALUES (2, 2,'Pick-up','Pick-up', 2);
INSERT INTO Hotel_Vehicle  VALUES (3, 3,'Kamyonet','Kamyonet', 3);
INSERT INTO Hotel_Vehicle  VALUES (4, 4,'Ticari','Ticari', 4);
INSERT INTO Hotel_Vehicle  VALUES (5, 5,'Sedan','Sedan', 5);


INSERT INTO Hotel_Room_Comment  VALUES (1, 10, 1);
INSERT INTO Hotel_Room_Comment  VALUES (2, 10, 2);
INSERT INTO Hotel_Room_Comment  VALUES (3, 10, 3);
INSERT INTO Hotel_Room_Comment VALUES (4, 10, 4);
INSERT INTO Hotel_Room_Comment  VALUES (5, 10, 5);
```

B2) Update Statements

```
--1 Writes NULL for the customers who make a reservation with an empty surname.
UPDATE Hotel_Guest
   SET Guest_Surname = 'EMPTY'
 WHERE Guest_id IN (SELECT Hotel_Reservation_Pre.Guest_id
                      FROM Hotel_Reservation_Pre, Hotel_Guest
                     WHERE Hotel_Reservation_Pre.Guest_Id =
                             Hotel_Guest.Guest_Id);



--2 Increases invoice value by 100
UPDATE Hotel_Room_Invoice
   SET invoice_value = invoice_value + 100;
```

```sql
--3-Increases the invoice amount of active reservations paid in cash by 150
UPDATE hotel_room_invoice
   SET invoice_value = invoice_value + 100
 WHERE invoice_id IN (SELECT invoice_id
                        FROM hotel_guest gs,
                             hotel_reservation_pre res_pre,
                             hotel_reservation res,
                             hotel_room room,
                             hotel_room_invoice invoice,
                             hotel_payment_type typ
                       WHERE     gs.guest_id = res_pre.guest_id
                             AND res.rsr_id = invoice.rsr_id
                             AND res.pre_rsr_id = res_pre.pre_rsr_id
                             AND room.room_id = res.room_id
                             AND typ.payment_type_id =
                                    invoice.payment_type_id
                             AND agent_id IS NULL
                             AND payment_type = 'Cash'
                             AND rsr_active = 1);

--4- Marks all customers who do not write their gender as male
UPDATE Hotel_Guest
   SET Guest_Gender = 'Male' WHERE Guest_Gender IS NULL;
```

B2) Delete Statements

```sql
--1-Deleting the records that are not assigned to any personnel from the Hotel
Personnel Type table
DELETE FROM Hotel_Staff_Type
      WHERE Staff_type_Id NOT IN (SELECT DISTINCT Staff_Type_Id
                                    FROM Hotel_Staff);

--2-Invoices after a week are deleted
DELETE FROM Hotel_Room_Invoice
      WHERE invoice_date = SYSDATE + 7;

--3-Deletes the value of all records whose comment value is empty
DELETE FROM Hotel_Room_Comment
      WHERE Comment_Value IS NULL;

--4-Pre-Reservation Id Delete the records that are equal to the last reservation ID
DELETE FROM Hotel_Reservation
      WHERE Rsr_Id = Pre_Rsr_Id;

--5-Deletes invoices less than 1000TL and issued after 15 days
DELETE FROM Hotel_Room_Invoice
      WHERE invoice_value < 1000 AND invoice_date = SYSDATE + 15
```

B2) Oracle Function

```
-- Function which calculates According to the entered year and room number
  -- Multiplying the dollar rate between 2010 and 2015 by 2 and taking the post 2015
dollar rate from the value
  --Direct booking guests whose payment type is cash
  -- and SQL function that returns the total amount they paid
CREATE OR REPLACE FUNCTION hotel_return_exchange (p_year           NUMBER,
                                                  p_room           NUMBER,
                                                  p_dollar_rate    NUMBER)
   RETURN NUMBER
IS
   result   NUMBER;
BEGIN
   IF p_year BETWEEN 2010 AND 2015
   THEN
      SELECT SUM (invoice_value) / (p_dollar_rate * 2) total_invoice_amount
        INTO result
        FROM hotel_guest mi,
             hotel_reservation_pre res_pre,
             hotel_reservation res,
             hotel_room room,
             hotel_room_invoice invoice,
             hotel_payment_type typ
       WHERE     mi.guest_id = res_pre.guest_id
             AND res.rsr_id = invoice.rsr_id
             AND res.pre_rsr_id = res_pre.pre_rsr_id
             AND room.room_id = res.room_id
             AND typ.payment_type_id = invoice.payment_type_id
             AND payment_type = 'Cash'
             AND agent_id IS NULL
             AND rsr_active = 1;
   ELSIF p_year > 2015
   THEN
      SELECT SUM (invoice_value) / p_dollar_rate total_invoice_amount
        INTO result
        FROM hotel_guest mi,
             hotel_reservation_pre res_pre,
             hotel_reservation res,
             hotel_room room,
             hotel_room_invoice invoice,
             hotel_payment_type typ
       WHERE     mi.guest_id = res_pre.guest_id
             AND res.rsr_id = invoice.rsr_id
             AND res.pre_rsr_id = res_pre.pre_rsr_id
             AND room.room_id = res.room_id
             AND typ.payment_type_id = invoice.payment_type_id
             AND agent_id IS NULL
             AND payment_type = 'Cash'
             AND rsr_active = 1;
   ELSE
      result := NULL;
   END IF;

   RETURN result;
END;
```

C) DQL Statements

C1) Simple Join

```
--Name, surname, company, address, city and telephone information of guests who have
booked more than 7 rooms

SELECT Guest_Name,
       Guest_Surname,
       Guest_Company_Name,
       Guest_Address,
       Guest_City,
       Guest_Phone
  FROM Hotel_Reservation_Pre, Hotel_Guest
 WHERE     Hotel_Reservation_pre.Guest_Id = Hotel_Guest.Guest_Id
           AND Rsr_Room_Number > 7
```

C2) Right Join

```
--Find  the guest Name, surname, company, address, city and telephone information who
save the profile but never reserved any room

SELECT Guest_Name,
       Guest_Surname,
       Guest_Company_Name,
       Guest Address,
       Guest_City,
       Guest_Phone
  FROM Hotel_Reservation_Pre
       RIGHT JOIN Hotel_Guest
          ON Hotel_Reservation_pre.Guest_Id = Hotel_Guest.Guest_Id
```

C3) Nested Query

```
--List of the types of rooms in the hotel never reserved any customer
  SELECT Hotel_Name, Room_Type
    FROM Hotel_Room, Hotel_Description d, hotel_room_type
   WHERE     hotel_Room.hotel_id = d.hotel_id
         AND hotel_room_type.room_type_id = hotel_room.room_type_id
         AND hotel_Room.room_id NOT IN (SELECT r.room_id
                                          FROM hotel_room r,
                                               hotel_reservation h
                                         WHERE h.room_id = r.room_id)
ORDER BY Hotel_Name, Room_Type ASC
```

C4) Order By

```
--Sorting the Male customers by name
SELECT Guest_Name,
       Guest_Surname,
       Guest_Company_Name,
       Guest_Address,
       Guest_City,
       Guest_Phone,
       Guest_Email
  FROM Hotel_Guest
 WHERE Guest_Gender = 'Erkek'
       order by Guest_Name
```

C5) Group By

```
--The total number of nights that the guests booked by the agencies stayed for more
than 2 nights and must be Male with their name, surname, phone and gender information
  SELECT guest_name,
         guest_name,
         guest_phone,
         guest_gender,
         SUM (rsr_end_date - rsr_start_date) total_night
    FROM hotel_guest mi, hotel_reservation_pre rez_pre
   WHERE     mi.guest_Id = rez_pre.guest_Id
         AND agent_Id IS NOT NULL
         AND rsr_active = 1
GROUP BY guest_name,
         guest_surname,
         guest_phone,
         guest_gender
        HAVING SUM (rsr_start_date - rsr_end_date) > 2
```

C6) Select with Function

```
--List the revenue of the room with dollar exchange and the with the rate  13
SELECT hotel_name,
       room_name,
       hotel_return_exchange (2022, room_id, 13) revenue
  FROM hotel_description d, hotel_room r
       WHERE d.hotel_id = r.hotel_id
```