

# **INFO 605: Information Systems Analysis and Design**

*Spring 2020*

**Final Project**

## **INTDB Co. Hotel Chain Management**

**Due Date: June 6, 2020**

**Group Member**  
**Kiran Zacharia – 14306687**

## Table of Contents

<b>1. Requirements .....</b>	<b>4</b>
<b>2. ER Model.....</b>	<b>5</b>
<b>3. Relational Schema .....</b>	<b>6</b>
<b>4. Data Dictionary .....</b>	<b>7</b>
<b>Hotel Data Dictionary .....</b>	<b>7</b>
<b>Staff Data Dictionary .....</b>	<b>7</b>
<b>Customer Data Dictionary .....</b>	<b>8</b>
<b>Customer_Phone Data Dictionary .....</b>	<b>9</b>
<b>Room Data Dictionary .....</b>	<b>9</b>
<b>Amenities Data Dictionary .....</b>	<b>10</b>
<b>Booking_Info Data Dictionary .....</b>	<b>10</b>
<b>Amenity_Requests Data Dictionary .....</b>	<b>12</b>
<b>Staff_Handles Data Dictionary .....</b>	<b>12</b>
<b>5. DDL.....</b>	<b>14</b>
<b>Hotel Create DDL .....</b>	<b>14</b>
<b>Staff Create DDL .....</b>	<b>14</b>
<b>Customer Create DDL.....</b>	<b>14</b>
<b>Customer_Phone Create DDL .....</b>	<b>14</b>
<b>Room Create DDL .....</b>	<b>15</b>
<b>Amenities Create DDL .....</b>	<b>15</b>
<b>Booking_Info Create DDL .....</b>	<b>15</b>
<b>Amenity_Requests Create DDL.....</b>	<b>16</b>
<b>Staff_Handles Create DDL .....</b>	<b>16</b>
<b>6. DML.....</b>	<b>17</b>
<b>Hotel Insert DML.....</b>	<b>17</b>
<b>Staff Insert DML .....</b>	<b>17</b>
<b>Customer Insert DML .....</b>	<b>18</b>
<b>Customer_Phone Insert DML .....</b>	<b>18</b>
<b>Room Insert DML.....</b>	<b>19</b>
<b>Amenities Insert DML .....</b>	<b>19</b>
<b>Booking_Info Insert DML.....</b>	<b>20</b>
<b>Amenity_Requests Insert DML .....</b>	<b>21</b>
<b>Staff_Handles Insert DML .....</b>	<b>22</b>

	<b>Customer Update DML .....</b>	<b>22</b>
	<b>Booking_Info Delete DML .....</b>	<b>23</b>
	<b>Booking_Info Update DML .....</b>	<b>23</b>
<b>7.</b>	<b>Queries .....</b>	<b>24</b>
	<b>Query Report 1.....</b>	<b>24</b>
	<b>Query Report 2.....</b>	<b>24</b>
	<b>Query Report 3.....</b>	<b>24</b>
	<b>Query Report 4.....</b>	<b>25</b>
	<b>Query Report 5.....</b>	<b>25</b>
	<b>Query Report 6.....</b>	<b>26</b>
	<b>Query Report 7.....</b>	<b>26</b>

## 1. Requirements

INTDB Co. owns a chain of hotels and want to create a database system to manage these hotels. They wish to create a database system to handle each of the chain hotel's room, staff, customer, amenities, and various relation details between them. The hotel entity has a unique hotel ID which is used to identify the hotel and a unique hotel name and address which is broken down to street, state, and zip, none of which are allowed to be empty.

The details of the staffs working contains a unique staff id, first name, last name and the salary of the staff. A hotel has at least one staff working per hotel and as many staff as possible while a staff must work at only one hotel at a time.

A customer is recognized by their unique customer id and stores their first and last names and all possible phone numbers of the customers. A staff handles payment details of at least one customer and can handle as many customers as possible, while a customer's details are also handled by at least one staff and can be assisted by multiple staff.

The room for each hotel has a room number which is used as a discriminator and contains the room cost for the room.

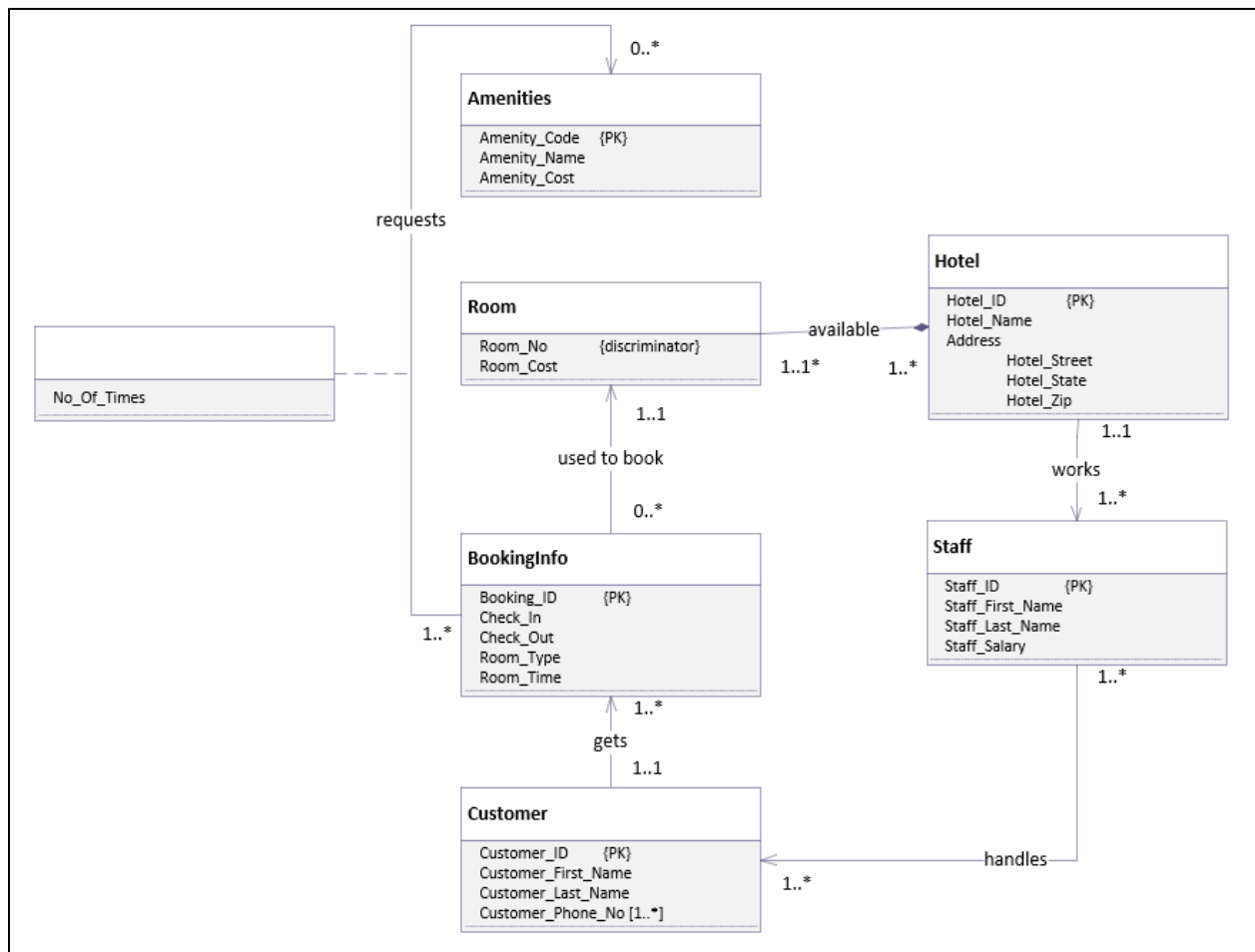
The different amenities which can be availed is stored using a unique code, name, and cost of the amenity.

The booking information for each customer has their unique booking id, check-in and check-out date, room rental type which can be hourly or daily and the time duration for when the room is rented on an hourly basis.

A customer can make multiple bookings and a customer has at least one booking and there is only one customer per booking. The booking information is used to book a room and there can be many or no bookings for a room, and only one room per booking.

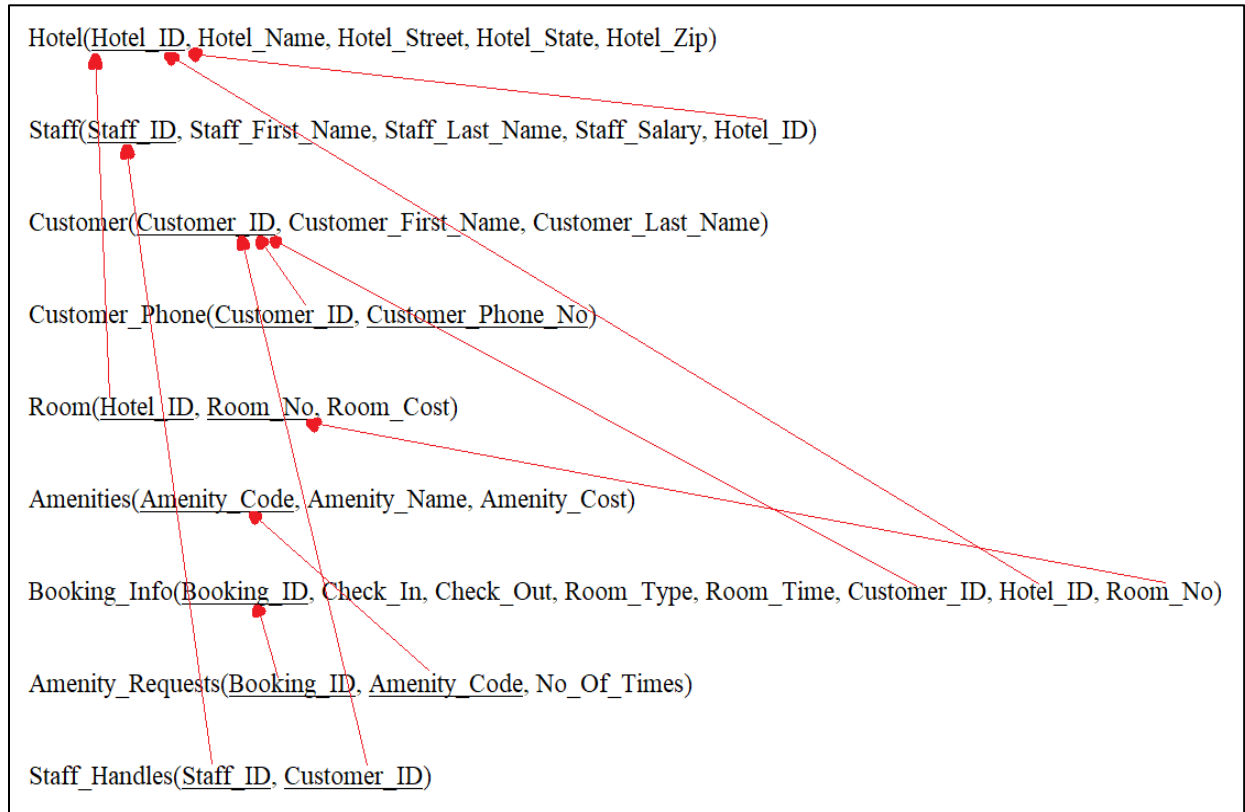
The booking information is used to request for amenities and the number of times an amenity is requested on a booking is tracked. At least one amenity request is to be made by the booking and a booking can be made to request as many amenities as they wish. An amenity needs not be requested by a booking but can be requested multiple times.

## 2. ER Model



### 3. Relational Schema

Normalization was applied on the ER diagram to get the following relational schema



## 4. Data Dictionary

### Hotel Data Dictionary

Hotel : Contains information regarding all the hotels chains owned by INTDB Co.						
AttributeName	Description	DataType	Domain	Nullable	PK	FK
Hotel_ID	Unique identifier for hotel which is part of INTDB Co. hotel chain	CHAR(8)	All	No	Yes	No
Hotel_Name	Name of the hotel which is part of INTDB Co. hotel chain	VARCHAR(50)	All, UNIQUE	No	No	No
Hotel_Street	Street address of the Hotel which is part of INTDB Co. hotel chain	VARCHAR(50)	All	No	No	No
Hotel_State	State where hotel is located	CHAR(2)	All	No	No	No
Hotel_Zip	Zip code of the hotel address	CHAR(5)	All	No	No	No

### Staff Data Dictionary

Staff : Contains information regarding the staff who work at a hotel						
AttributeName	Description	DataType	Domain	Nullable	PK	FK
Staff_ID	Unique identifier for staff who is part of INTDB Co. hotel chain	CHAR(8)	All	No	Yes	No
Staff_First_Name	First Name of Staff who is part of INTDB Co. hotel chain	VARCHAR(50)	All	No	No	No
Staff_Last_Name	Last Name of Staff who is part of	VARCHAR(50)	All	No	No	No

	INTDB Co. hotel chain					
Staff_Salary	Salary Amount of the Staff who is part of INTDB Co. hotel chain	NUMBER(10,2)	All	No	No	No
Hotel_ID	Unique identifier for hotel which is part of INTDB Co. hotel chain	CHAR(8)	All	No	No	Yes

### Customer Data Dictionary

Customer : Contains information regarding the customer who stay at any of the hotels						
AttributeName	Description	Data Type	Domain	Nullable	PK	FK
Customer_ID	Unique identifier for customers of INTDB Co. hotel chain	CHAR(8)	All	No	Yes	No
Customer_First_Name	First Name of customer who used one of the hotels of the INTDB Co. hotel chain	VARCHAR(50)	All	No	No	No
Customer_Last_Name	Last Name of Customer who used one of the hotels of the INTDB Co. hotel chain	VARCHAR(50)	All	No	No	No



### Customer\_Phone Data Dictionary

Customer_Phone : Contains information of all the phone numbers owned by the customer						
AttributeName	Description	DataType	Domain	Nullable	PK	FK
Customer_ID	Unique identifier for customers of INTDB Co. hotel chain	CHAR(8)	All	No	Yes	Yes
Customer_Phone_No	Phone numbers of the customers who used one of the hotels of the INTDB Co. hotel chain	CHAR(10)	All	No	Yes	No

### Room Data Dictionary

Room : Contains information regarding the hotel rooms						
AttributeName	Description	DataType	Domain	Nullable	PK	FK
Hotel_ID	Unique identifier for hotel which is part of the INTDB Co. hotel chain	CHAR(8)	All	No	Yes	Yes
Room_No	Number of the room of a hotel which is part of the INTDB Co. hotel chain	CHAR(4)	All	No	Yes	No
Room_Cost	Cost of the room of a hotel which is part of the INTDB Co. hotel chain	NUMBER(10,2)	All	No	No	No

### Amenities Data Dictionary

Amenities : Contains information regarding all the hotel amenities available						
AttributeName	Description	DataType	Domain	Nullable	PK	FK
Amenity_Code	Unique identifier for amenities that are available at all hotels which is part of the INTDB Co. hotel chain	CHAR(8)	All	No	Yes	No
Amenity_Name	Name of the amenities that are available at all hotels which is part of the INTDB Co. hotel chain	VARCHAR(50)	All	No	No	No
Amenity_Cost	Cost of the amenities that are available at all hotels which is part of the INTDB Co. hotel chain	NUMBER(10,2)	All	No	No	No

### Booking\_Info Data Dictionary

Booking_Info : Contains information regarding a hotel room booking information of a customer						
AttributeName	Description	DataType	Domain	Nullable	PK	FK
Booking_ID	Unique identifier for booking of one of the hotel rooms of the INTDB Co. hotel chain	CHAR(8)	All	No	Yes	No
Check_In	Date on which the customer is checking in for a specific booking at one of the hotel	DATE	All	No	No	No

	rooms of the INTDB Co. hotel chain					
Check_Out	Date on which the customer is checking out for a specific booking from one of the hotel rooms of the INTDB Co. hotel chain	DATE	All	No	No	No
Room_Type	Flag indicating whether the room is being rented on a daily rate or an hourly rate at one of the hotel rooms of the INTDB Co. hotel chain	CHAR(1)	“D”, “H”	No	No	No
Room_Time	Hours the room was rented for when the room is rented on an hourly basis at one of the hotel rooms of the INTDB Co. hotel chain	NUMBER(10,0)	All, defaults to 24 hours	No	No	No
Customer_ID	Unique identifier for customers of INTDB Co. hotel chain	CHAR(8)	All	No	No	Yes
Hotel_ID	Unique identifier for hotel which is part of the INTDB Co. hotel chain	CHAR(8)	All	No	No	Yes
Room_No	Number of the room of a hotel which is part of	CHAR(4)	All	No	No	Yes

	the INTDB Co. hotel chain					
--	------------------------------	--	--	--	--	--

### **Amenity\_Requests Data Dictionary**

Amenity_Requests : Contains information regarding amenities used by a customer for a specific booking						
AttributeName	Description	DataType	Domain	Nullable	PK	FK
Booking_ID	Unique identifier for booking of one of the hotel rooms of the INTDB Co. hotel chain	CHAR(8)	All	No	Yes	Yes
Amenity_Code	Unique identifier for amenities that are available at all hotels which is part of the INTDB Co. hotel chain	CHAR(8)	All	No	Yes	Yes
No_Of_Times	Number of times the amenity was requested for a certain booking at one of the hotel rooms which is a part of the INTDB Co. hotel chain	NUMBER(10,0)	All	No	No	No

### **Staff\_Handles Data Dictionary**

Staff_Handles : Contains information of staff that worked with a customer to handle payment						
AttributeName	Description	DataType	Domain	Nullable	PK	FK
Staff_ID	Unique identifier for staff who is	CHAR(8)	All	No	Yes	Yes

	part of INTDB Co. hotel chain					
Customer_ID	Unique identifier for customers of INTDB Co. hotel chain	CHAR(8)	All	No	Yes	Yes

## 5. DDL

### Hotel Create DDL

```
CREATE TABLE Hotel
(
    Hotel_ID CHAR(8),
    Hotel_Name VARCHAR(50) CONSTRAINT Hotel_nm_nn NOT NULL UNIQUE,
    Hotel_Street VARCHAR(50) CONSTRAINT Hotel_str_nn NOT NULL,
    Hotel_State CHAR(2) CONSTRAINT Hotel_st_nn NOT NULL,
    Hotel_Zip CHAR(5) CONSTRAINT Hotel_zip_nn NOT NULL,
    CONSTRAINT Hotel_pk PRIMARY KEY(Hotel_ID)
);
```

### Staff Create DDL

```
CREATE TABLE Staff
(
    Staff_ID CHAR(8),
    Staff_First_Name VARCHAR(50) CONSTRAINT staff_fn_nn NOT NULL,
    Staff_Last_Name VARCHAR(50) CONSTRAINT staff_ln_nn NOT NULL,
    Staff_Salary NUMBER(10,2) CONSTRAINT staff_sal_nn NOT NULL,
    Hotel_ID CHAR(8) CONSTRAINT Hotel_staff_id_nn NOT NULL,
    CONSTRAINT Staff_pk PRIMARY KEY(Staff_ID),
    CONSTRAINT Hotel_Staff_fk FOREIGN KEY(Hotel_ID) REFERENCES
Hotel(Hotel_ID) ON DELETE CASCADE
);
```

### Customer Create DDL

```
CREATE TABLE Customer
(
    Customer_ID CHAR(8),
    Customer_First_Name VARCHAR(50) CONSTRAINT Customer_fn_nn NOT NULL,
    Customer_Last_Name VARCHAR(50) CONSTRAINT Customer_ln_nn NOT NULL,
    CONSTRAINT Customer_pk PRIMARY KEY(Customer_ID)
);
```

### Customer\_Phone Create DDL

```
CREATE TABLE Customer_Phone
(
    Customer_ID CHAR(8),
    Customer_Phone_No CHAR(10) CONSTRAINT Customer_Phone_N_nn_unq NOT NULL
UNIQUE,
    CONSTRAINT Customer_Phone_pk PRIMARY KEY(Customer_ID, Customer_Phone_No),
    CONSTRAINT Customer_Phone_fk FOREIGN KEY(Customer_ID) REFERENCES
Customer(Customer_ID) ON DELETE CASCADE
);
```

## Room Create DDL

```
CREATE TABLE Room
(
    Hotel_ID CHAR(8),
    Room_No CHAR(4),
    Room_Cost NUMBER(10,2) CONSTRAINT Room_cost_nn NOT NULL,
    CONSTRAINT Room_pk PRIMARY KEY(Hotel_ID,Room_No),
    CONSTRAINT Hotel_Room_fk FOREIGN KEY(Hotel_ID) REFERENCES Hotel(Hotel_ID)
ON DELETE CASCADE
);
```

## Amenities Create DDL

```
CREATE TABLE Amenities
(
    Amenity_Code CHAR(8),
    Amenity_Name VARCHAR(50) CONSTRAINT Amenity_nm_nn NOT NULL,
    Amenity_Cost NUMBER(10,2) CONSTRAINT Amenity_cost_nn NOT NULL,
    CONSTRAINT Amenity_pk PRIMARY KEY(Amenity_Code)
);
```

## Booking\_Info Create DDL

```
CREATE TABLE Booking_Info
(
    Booking_ID CHAR(8),
    Check_In DATE CONSTRAINT check_in_nn NOT NULL,
    Check_Out DATE CONSTRAINT check_out_nn NOT NULL,
    Room_Type CHAR(1) CONSTRAINT room_type_nchk NOT NULL CHECK (Room_Type IN
('H', 'D')),
    Room_Time NUMBER(10,0) DEFAULT 24 CONSTRAINT room_time_nn NOT NULL,
    Customer_ID CHAR(8) CONSTRAINT cust_book_nn NOT NULL,
    Hotel_ID CHAR(8) CONSTRAINT hotel_book_nn NOT NULL,
    Room_No CHAR(4) CONSTRAINT room_book_nn NOT NULL,
    CONSTRAINT booking_pk PRIMARY KEY(Booking_ID),
    CONSTRAINT Customer_Booking_fk FOREIGN KEY(Customer_ID) REFERENCES
Customer(Customer_ID) ON DELETE CASCADE,
    CONSTRAINT Hotel_Room_Booking_fk FOREIGN KEY(Hotel_ID, Room_No)
REFERENCES Room(Hotel_ID, Room_No) ON DELETE CASCADE
);
```

## Amenity\_Requests Create DDL

```
CREATE TABLE Amenity_Requests
(
    Booking_ID CHAR(8) CONSTRAINT Booking_Request_id NOT NULL,
    Amenity_Code CHAR(8) CONSTRAINT Amenity_Request_code NOT NULL,
    No_Of_Times NUMBER(10,0) CONSTRAINT No_Of_Times_null NOT NULL,
    CONSTRAINT Amenity_Booking_Request_pk PRIMARY
KEY (Booking_ID,Amenity_Code),
    CONSTRAINT Booking_Request_fk FOREIGN KEY (Booking_ID) REFERENCES
Booking_Info(Booking_ID) ON DELETE CASCADE,
    CONSTRAINT Amenity_Request_fk FOREIGN KEY (Amenity_Code) REFERENCES
Amenities(Amenity_Code) ON DELETE CASCADE
);
```

## Staff\_Handles Create DDL

```
CREATE TABLE Staff_Handles
(
    Staff_ID CHAR(8) CONSTRAINT Staff_Handle_null NOT NULL,
    Customer_ID CHAR(8) CONSTRAINT Customer_Handle_null NOT NULL,
    CONSTRAINT Staff_Customer_Handle_pk PRIMARY KEY (Staff_ID,Customer_ID),
    CONSTRAINT Staff_Handle_fk FOREIGN KEY (Staff_ID) REFERENCES
Staff(Staff_ID) ON DELETE CASCADE,
    CONSTRAINT Customer_Handle_fk FOREIGN KEY (Customer_ID) REFERENCES
Customer(Customer_ID) ON DELETE CASCADE
);
```



## 6. DML

### Hotel Insert DML

```
-- Hotel(Hotel_ID, Hotel_Name, Hotel_Street, Hotel_State, Hotel_Zip)
--1
INSERT INTO Hotel(Hotel_ID, Hotel_Name, Hotel_Street, Hotel_State, Hotel_Zip)
VALUES ('11111111','Hotel_1','Street1','VA','11111');
--2
INSERT INTO Hotel(Hotel_ID, Hotel_Name, Hotel_Street, Hotel_State, Hotel_Zip)
VALUES ('11111112','Hotel_2','Street2','PA','11112');
--3
INSERT INTO Hotel(Hotel_ID, Hotel_Name, Hotel_Street, Hotel_State, Hotel_Zip)
VALUES ('11111113','Hotel_3','Street3','VA','11113');
SELECT * FROM Hotel;
```

Output:

	HOTEL_ID	HOTEL_NAME	HOTEL_STREET	HOTEL_STATE	HOTEL_ZIP
1	11111111	Hotel_1	Street1	VA	11111
2	11111112	Hotel_2	Street2	PA	11112
3	11111113	Hotel_3	Street3	VA	11113

### Staff Insert DML

```
--Staff(Staff_ID, Staff_First_Name, Staff_Last_Name, Staff_Salary, Hotel_ID)
--1
INSERT INTO Staff(Staff_ID, Staff_First_Name, Staff_Last_Name, Staff_Salary,
Hotel_ID) VALUES ('E1111111','FStaffone','LStaffone',407.00,'11111111');
--2
INSERT INTO Staff(Staff_ID, Staff_First_Name, Staff_Last_Name, Staff_Salary,
Hotel_ID) VALUES ('E1111112','FStafftwo','LStafftwo',207.00,'11111111');
--3
INSERT INTO Staff(Staff_ID, Staff_First_Name, Staff_Last_Name, Staff_Salary,
Hotel_ID) VALUES ('E1111113','FStaffthree','LStaffthree',77.00,'11111111');
--4
INSERT INTO Staff(Staff_ID, Staff_First_Name, Staff_Last_Name, Staff_Salary,
Hotel_ID) VALUES ('E2111111','FNextone','LNextone',107.00,'11111112');
--5
INSERT INTO Staff(Staff_ID, Staff_First_Name, Staff_Last_Name, Staff_Salary,
Hotel_ID) VALUES ('E3111111','FLastone','LLastone',307.00,'11111113');
--Show Staff Table
SELECT * FROM Staff;
```

Output:

	STAFF_ID	STAFF_FIRST_NAME	STAFF_LAST_NAME	STAFF_SALARY	HOTEL_ID
1	E1111111	FStaffone	LStaffone	407	11111111
2	E1111112	FStafftwo	LStafftwo	207	11111111
3	E1111113	FStaffthree	LStaffthree	77	11111111
4	E2111111	FNextone	LNextone	107	11111112
5	E3111111	FLastone	LLastone	307	11111113

## Customer Insert DML

```
--Customer(Customer_ID, Customer_First_Name, Customer_Last_Name)
--1
INSERT INTO Customer(Customer_ID, Customer_First_Name, Customer_Last_Name)
VALUES ('C1111111', 'CustOne', 'OneLast');
--2
INSERT INTO Customer(Customer_ID, Customer_First_Name, Customer_Last_Name)
VALUES ('C1111112', 'CustTwo', 'TwoLast');
--3
INSERT INTO Customer(Customer_ID, Customer_First_Name, Customer_Last_Name)
VALUES ('C1111113', 'ThreeCust', 'ThreeCust');
--Show Customer Table
SELECT * FROM Customer;
```

Output:

	CUSTOMER_ID	CUSTOMER_FIRST_NAME	CUSTOMER_LAST_NAME
1	C1111111	CustOne	OneLast
2	C1111112	CustTwo	TwoLast
3	C1111113	ThreeCust	ThreeCust

## Customer\_Phone Insert DML

```
--Customer_Phone(Customer_ID, Customer_Phone_No)
--1
INSERT INTO Customer_Phone(Customer_ID, Customer_Phone_No) VALUES
('C1111111', '1213441245');
--2
INSERT INTO Customer_Phone(Customer_ID, Customer_Phone_No) VALUES
('C1111111', '1213441246');
--3
INSERT INTO Customer_Phone(Customer_ID, Customer_Phone_No) VALUES
('C1111112', '2213441245');
--4
INSERT INTO Customer_Phone(Customer_ID, Customer_Phone_No) VALUES
('C1111113', '1213341245');
--Show Customer_Phone Table
SELECT * FROM Customer_Phone;
```

Output:

	CUSTOMER_ID	CUSTOMER_PHONE_NO
1	C1111111	1213441245
2	C1111111	1213441246
3	C1111112	2213441245
4	C1111113	1213341245

## Room Insert DML

```
--Room(Hotel_ID, Room_No, Room_Cost)
--1
INSERT INTO Room(Hotel_ID, Room_No, Room_Cost) VALUES ('11111111', 'A111',
1500.56);
--2
INSERT INTO Room(Hotel_ID, Room_No, Room_Cost) VALUES ('11111111', 'A211',
1500.56);
--3
INSERT INTO Room(Hotel_ID, Room_No, Room_Cost) VALUES ('11111111', 'A311',
1500.56);
--4
INSERT INTO Room(Hotel_ID, Room_No, Room_Cost) VALUES ('11111112', 'A111',
2500.56);
--5
INSERT INTO Room(Hotel_ID, Room_No, Room_Cost) VALUES ('11111113', 'A111',
6000.00);
--Show Room Table
SELECT * FROM Room;
```

Output:

	HOTEL_ID	ROOM_NO	ROOM_COST
1	11111111	A111	1500.56
2	11111111	A211	1500.56
3	11111111	A311	1500.56
4	11111112	A111	2500.56
5	11111113	A111	6000

## Amenities Insert DML

```
--Amenities(Amenity_Code, Amenity_Name, Amenity_Cost)
--1
INSERT INTO Amenities(Amenity_Code, Amenity_Name, Amenity_Cost) VALUES
('A1111111', 'Dry Cleaning', 12.00);
--2
INSERT INTO Amenities(Amenity_Code, Amenity_Name, Amenity_Cost) VALUES
('A1111112', 'Gym', 3.00);
--3
INSERT INTO Amenities(Amenity_Code, Amenity_Name, Amenity_Cost) VALUES
('A1111113', 'Laundry', 4.00);
--Show Amenities Table
SELECT * FROM Amenities;
```

Output:

	AMENITY_CODE	AMENITY_NAME	AMENITY_COST
1	A1111111	Dry Cleaning	12
2	A1111112	Gym	3
3	A1111113	Laundry	4

## Booking\_Info Insert DML

```
--Booking_Info(Booking_ID, Check_In, Check_Out, Room_Type, Room_Time,
Customer_ID, Hotel_ID, Room_No)
--1
INSERT INTO Booking_Info(Booking_ID, Check_In, Check_Out, Room_Type,
Room_Time, Customer_ID, Hotel_ID, Room_No) VALUES ('B1111111', '13-MAY-2020',
'13-MAY-2020', 'H', '14', 'C1111111', '11111111', 'A111');
--2
INSERT INTO Booking_Info(Booking_ID, Check_In, Check_Out, Room_Type,
Customer_ID, Hotel_ID, Room_No) VALUES ('B1111112', '14-MAY-2020', '23-MAY-
2020', 'D', 'C1111111', '11111111', 'A111');
--3
INSERT INTO Booking_Info(Booking_ID, Check_In, Check_Out, Room_Type,
Customer_ID, Hotel_ID, Room_No) VALUES ('B1111113', '13-MAY-2020', '19-MAY-
2020', 'D', 'C1111112', '11111111', 'A211');
--4
INSERT INTO Booking_Info(Booking_ID, Check_In, Check_Out, Room_Type,
Customer_ID, Hotel_ID, Room_No) VALUES ('B1111114', '1-MAY-2020', '15-MAY-
2020', 'D', 'C1111113', '11111111', 'A311');
--5
INSERT INTO Booking_Info(Booking_ID, Check_In, Check_Out, Room_Type,
Room_Time, Customer_ID, Hotel_ID, Room_No) VALUES ('B1111115', '16-MAY-2020',
'01-JUNE-2020', 'H', '14', 'C1111111', '11111113', 'A111');
--6
INSERT INTO Booking_Info(Booking_ID, Check_In, Check_Out, Room_Type,
Room_Time, Customer_ID, Hotel_ID, Room_No) VALUES ('B1111116', '16-MAY-2020',
'01-MAY-2020', 'H', '14', 'C1111111', '11111111', 'A211');
--7
INSERT INTO Booking_Info(Booking_ID, Check_In, Check_Out, Room_Type,
Room_Time, Customer_ID, Hotel_ID, Room_No) VALUES ('B1111117', '16-MAY-2020',
'16-MAY-2020', 'H', '14', 'C1111113', '11111111', 'A111');
--Show Booking_Info Table
SELECT * FROM Booking_Info;
```

Output:

	BOOKING_ID	CHECK_IN	CHECK_OUT	ROOM_TYPE	ROOM_TIME	CUSTOMER_ID	HOTEL_ID	ROOM_NO
1	B1111111	13-MAY-20	13-MAY-20	H	14	C1111111	11111111	A111
2	B1111112	14-MAY-20	23-MAY-20	D	24	C1111111	11111111	A111
3	B1111113	13-MAY-20	19-MAY-20	D	24	C1111112	11111111	A211
4	B1111114	01-MAY-20	15-MAY-20	D	24	C1111113	11111111	A311
5	B1111115	16-MAY-20	01-JUN-20	H	14	C1111111	11111113	A111
6	B1111116	16-MAY-20	01-MAY-20	H	14	C1111111	11111111	A211
7	B1111117	16-MAY-20	16-MAY-20	H	14	C1111113	11111111	A111

## Amenity\_Requests Insert DML

```
--Amenity_Requests(Booking_ID, Amenity_Code, No_Of_Times)
--1
INSERT INTO Amenity_Requests(Booking_ID, Amenity_Code, No_Of_Times) VALUES
('B1111112','A1111113', 3);
--2
INSERT INTO Amenity_Requests(Booking_ID, Amenity_Code, No_Of_Times) VALUES
('B1111111','A1111112', 1);
--3
INSERT INTO Amenity_Requests(Booking_ID, Amenity_Code, No_Of_Times) VALUES
('B1111114','A1111111', 1);
--4
INSERT INTO Amenity_Requests(Booking_ID, Amenity_Code, No_Of_Times) VALUES
('B1111115','A1111111', 2);
--5
INSERT INTO Amenity_Requests(Booking_ID, Amenity_Code, No_Of_Times) VALUES
('B1111113','A1111111', 1);
--6
INSERT INTO Amenity_Requests(Booking_ID, Amenity_Code, No_Of_Times) VALUES
('B1111112','A1111112', 1);
--7
INSERT INTO Amenity_Requests(Booking_ID, Amenity_Code, No_Of_Times) VALUES
('B1111112','A1111111', 1);
--Show Amenity_Requests Table
SELECT * FROM Amenity_Requests;
```

Output:

	BOOKING_ID	AMENITY_CODE	NO_OF_TIMES
1	B1111112	A1111113	3
2	B1111111	A1111112	1
3	B1111114	A1111111	1
4	B1111115	A1111111	2
5	B1111113	A1111111	1
6	B1111112	A1111112	1
7	B1111112	A1111111	1

## Staff\_Handles Insert DML

```
--Staff_Handles(Staff_ID, Customer_ID)
--1
INSERT INTO Staff_Handles(Staff_ID, Customer_ID) VALUES ('E1111112',
'C1111111');
--2
INSERT INTO Staff_Handles(Staff_ID, Customer_ID) VALUES ('E1111111',
'C1111111');
--3
INSERT INTO Staff_Handles(Staff_ID, Customer_ID) VALUES ('E2111111',
'C1111112');
--4
INSERT INTO Staff_Handles(Staff_ID, Customer_ID) VALUES ('E3111111',
'C1111113');
--5
INSERT INTO Staff_Handles(Staff_ID, Customer_ID) VALUES ('E1111113',
'C1111111');
--6
INSERT INTO Staff_Handles(Staff_ID, Customer_ID) VALUES ('E1111111',
'C1111112');
--Show Staff_Handles Table
SELECT * FROM Staff_Handles;
```

Output:

	STAFF_ID	CUSTOMER_ID
1	E1111111	C1111111
2	E1111111	C1111112
3	E1111112	C1111111
4	E1111113	C1111111
5	E2111111	C1111112
6	E3111111	C1111113

## Customer Update DML

```
--Change customer id is 'C1111113' First name to Alpha and Last name to Beta.
UPDATE Customer
SET Customer_First_Name = 'Alpha', Customer_Last_Name = 'Beta'
WHERE Customer_id = 'C1111113';
--Show Updated Customer Table
SELECT * FROM Customer;
```

Output:

	CUSTOMER_ID	CUSTOMER_FIRST_NAME	CUSTOMER_LAST_NAME
1	C1111111	CustOne	OneLast
2	C1111112	CustTwo	TwoLast
3	C1111113	Alpha	Beta

## Booking\_Info Delete DML

--Delete the record for booking information where check in date is before check out date

```
DELETE FROM Booking_Info
WHERE Check_in > Check_Out;
--Show Booking_Info Table after deletion
SELECT * FROM Booking_Info;
```

Output:

	BOOKING_ID	CHECK_IN	CHECK_OUT	ROOM_TYPE	ROOM_TIME	CUSTOMER_ID	HOTEL_ID	ROOM_NO
1	B1111111	13-MAY-20	13-MAY-20	H	14	C1111111	11111111	A111
2	B1111112	14-MAY-20	23-MAY-20	D	24	C1111111	11111111	A111
3	B1111113	13-MAY-20	19-MAY-20	D	24	C1111112	11111111	A211
4	B1111114	01-MAY-20	15-MAY-20	D	24	C1111113	11111111	A311
5	B1111115	16-MAY-20	01-JUN-20	H	14	C1111111	11111113	A111
6	B1111117	16-MAY-20	16-MAY-20	H	14	C1111113	11111111	A111

## Booking\_Info Update DML

--Set the room type to 'D' when check-in date is not same as check-out date

```
UPDATE Booking_Info
SET Room_Type = 'D'
WHERE Check_in != Check_Out;
--Show Updated Booking_Info Table
SELECT * FROM Booking_Info;
```

Output:

	BOOKING_ID	CHECK_IN	CHECK_OUT	ROOM_TYPE	ROOM_TIME	CUSTOMER_ID	HOTEL_ID	ROOM_NO
1	B1111111	13-MAY-20	13-MAY-20	H	14	C1111111	11111111	A111
2	B1111112	14-MAY-20	23-MAY-20	D	24	C1111111	11111111	A111
3	B1111113	13-MAY-20	19-MAY-20	D	24	C1111112	11111111	A211
4	B1111114	01-MAY-20	15-MAY-20	D	24	C1111113	11111111	A311
5	B1111115	16-MAY-20	01-JUN-20	D	14	C1111111	11111113	A111
6	B1111117	16-MAY-20	16-MAY-20	H	14	C1111113	11111111	A111

## 7. Queries

### Query Report 1

- 1.) Show all first name, last name and phone numbers of customers

```
SELECT C.Customer_First_Name, C.Customer_Last_Name, P.Customer_Phone_No
FROM Customer C
      JOIN Customer_Phone P ON C.Customer_id = P.Customer_id;
```

Output:

	CUSTOMER_FIRST_NAME	CUSTOMER_LAST_NAME	CUSTOMER_PHONE_NO
1	CustOne	OneLast	1213441245
2	CustOne	OneLast	1213441246
3	CustTwo	TwoLast	2213441245
4	Alpha	Beta	1213341245

### Query Report 2

- 2.) Show the total cost of the amenities requested for each booking ordered according to the booking id

```
SELECT AR.Booking_ID, SUM(A.amenity_cost * AR.no_of_times ) AS
Amenity_Cost_Total
FROM Amenities A
      JOIN Amenity_Requests AR ON A.Amenity_Code = AR.Amenity_Code
GROUP BY AR.Booking_ID
ORDER BY AR.Booking_ID;
```

Output:

	BOOKING_ID	AMENITY_COST_TOTAL
1	B1111111	3
2	B1111112	27
3	B1111113	12
4	B1111114	12
5	B1111115	24

### Query Report 3

- 3.) Show all staff information whose first name contains 'one' in lowercase

```
SELECT *
FROM Staff
WHERE Staff_First_Name LIKE '%one%';
```



Output:

	STAFF_ID	STAFF_FIRST_NAME	STAFF_LAST_NAME	STAFF_SALARY	HOTEL_ID
1	E11111111	FStaffone	LStaffone	407	11111111
2	E21111111	FNextone	LNextone	107	11111112
3	E31111111	FLastone	LLastone	307	11111113

#### Query Report 4

- 4.) Show the customer, id, first name, last name and count of bookings of customers who has more than one booking done at the hotel chains

```
SELECT B.Customer_Id,C.Customer_First_Name,C.Customer_Last_Name,
COUNT(B.Booking_id) AS Cust_Booking_Count
FROM Booking_Info B
JOIN Customer C ON B.Customer_id = C.Customer_id
GROUP BY B.Customer_Id,C.Customer_First_Name,C.Customer_Last_Name
HAVING COUNT(B.Booking_id)>1;
```

Output:

	CUSTOMER_ID	CUSTOMER_FIRST_NAME	CUSTOMER_LAST_NAME	CUST_BOOKING_COUNT
1	C11111111	CustOne	OneLast	3
2	C11111113	Alpha	Beta	2

#### Query Report 5

- 5.) Shows the total amenity cost, room cost and the total booking cost of a booking

```
SELECT B.Booking_Id,(A.Amenity_Cost*AR.No_Of_Times) AS "Total Amenity Cost",
CASE WHEN B.Room_Type = 'D' THEN ROUND(((B.Check_Out-
B.Check_In)*R.Room_Cost),2) WHEN B.Room_Type = 'H' THEN
ROUND(((B.Room_Time)*(R.Room_Cost/24)),2) END AS "Total Room Cost", CASE
WHEN B.Room_Type = 'D' THEN ROUND(((B.Check_Out-
B.Check_In)*R.Room_Cost),2)+(A.Amenity_Cost*AR.No_Of_Times) WHEN B.Room_Type
= 'H' THEN
ROUND(((B.Room_Time)*(R.Room_Cost/24)),2)+(A.Amenity_Cost*AR.No_Of_Times) END
AS "Total Booking Cost"
FROM Room R, Booking_Info B, Amenity_Requests AR,Amenities A
WHERE R.Hotel_Id = B.Hotel_Id AND R.Room_No = B.Room_No AND B.Booking_ID =
AR.Booking_ID AND AR.Amenity_Code = A.Amenity_Code;
```

Output:

	BOOKING_ID	Total Amenity Cost	Total Room Cost	Total Booking Cost
1	B11111114	12	21007.84	21019.84
2	B11111115	24	96000	96024
3	B11111113	12	9003.36	9015.36
4	B11111112	12	13505.04	13517.04
5	B11111111	3	875.33	878.33
6	B11111112	3	13505.04	13508.04
7	B11111112	12	13505.04	13517.04

## Query Report 6

- 6.) Show the staff's first name, last name, salary, commission (salary\*0.05\*number of customers handled) and total salary due which is the sum of salary and commission

```
SELECT S.Staff_Id, S.Staff_First_Name, S.Staff_Last_Name, S.Staff_Salary,  
ROUND(S.Staff_Salary*0.05*COUNT(SH.Customer_Id),2) AS Commision,  
ROUND(S.Staff_Salary*0.05*COUNT(SH.Customer_Id),2)+S.Staff_Salary AS "Total  
Salary"  
FROM Staff S, Staff_Handles SH  
WHERE S.Staff_Id = SH.Staff_Id  
GROUP BY S.Staff_Id, S.Staff_First_Name, S.Staff_Last_Name, S.Staff_Salary;
```

Output:

	STAFF_ID	STAFF_FIRST_NAME	STAFF_LAST_NAME	STAFF_SALARY	COMMISSION	Total Salary
1	E1111112	FStafftwo	LStafftwo	207	10.35	217.35
2	E2111111	FNextone	LNextone	107	5.35	112.35
3	E1111111	FStaffone	LStaffone	407	40.7	447.7
4	E3111111	FLastone	LLastone	307	15.35	322.35
5	E1111113	FStaffthree	LStaffthree	77	3.85	80.85

## Query Report 7

- 7.) Show the staff's first name, last name and salary of the staffs that make more than the average salary ordered according to their salary

```
SELECT Staff_First_Name,Staff_Last_Name, Staff_Salary  
FROM Staff  
WHERE Staff_Salary < (SELECT AVG(Staff_Salary)  
FROM Staff)  
ORDER BY Staff_Salary DESC;
```

Output:

	STAFF_FIRST_NAME	STAFF_LAST_NAME	STAFF_SALARY
1	FStafftwo	LStafftwo	207
2	FNextone	LNextone	107
3	FStaffthree	LStaffthree	77