DBMS PROJECT

Dr. Ali Ouni

Done by:

1) Hamza Luai U22105870

2) Ahmad Al Dosari U22106543

3) Mohammad Raed U22105630

4) Mahaz Ishtiaq khan U22200217

Table of contents:

1.0) Restatement of the case study

Case Study:

The system will support multiple hotel operating companies, each managing several hotels. Each hotel will offer numerous rooms available for booking. Every hotel operating company will have a user account within the booking system. Users will enter initial hotel information and will be able to update room availability. Customers will register with the booking system to book rooms for specific periods. They will also have access to their previous bookings and the ability to cancel reservations.

System Facts and Policies:

• Each hotel operating company is identified by its name, while each hotel will be assigned a unique integer identifier. Other attributes for hotels include: Hotel Name, Hotel Type (e.g., business, resort, motel), Star Rating (e.g., 5-star), Contact Phone Number, Contact Email, and Address (including country, city, and street address).

• Each room have the following attributes: Daily Accommodation Price, Area (e.g., 25 m^2), View (e.g., sea view), and List of Amenities (e.g., hair dryer, iron, safe). • Users from hotel operating companies will have the ability to add or remove hotels and create rooms. They can also view and update the availability of their hotel rooms.

• Customers will be able to search for, book, and manage their reservations. Customer attributes include Social Security Number (SSN), Name, Email, Phone Number, and Gender.

Our Restatement:

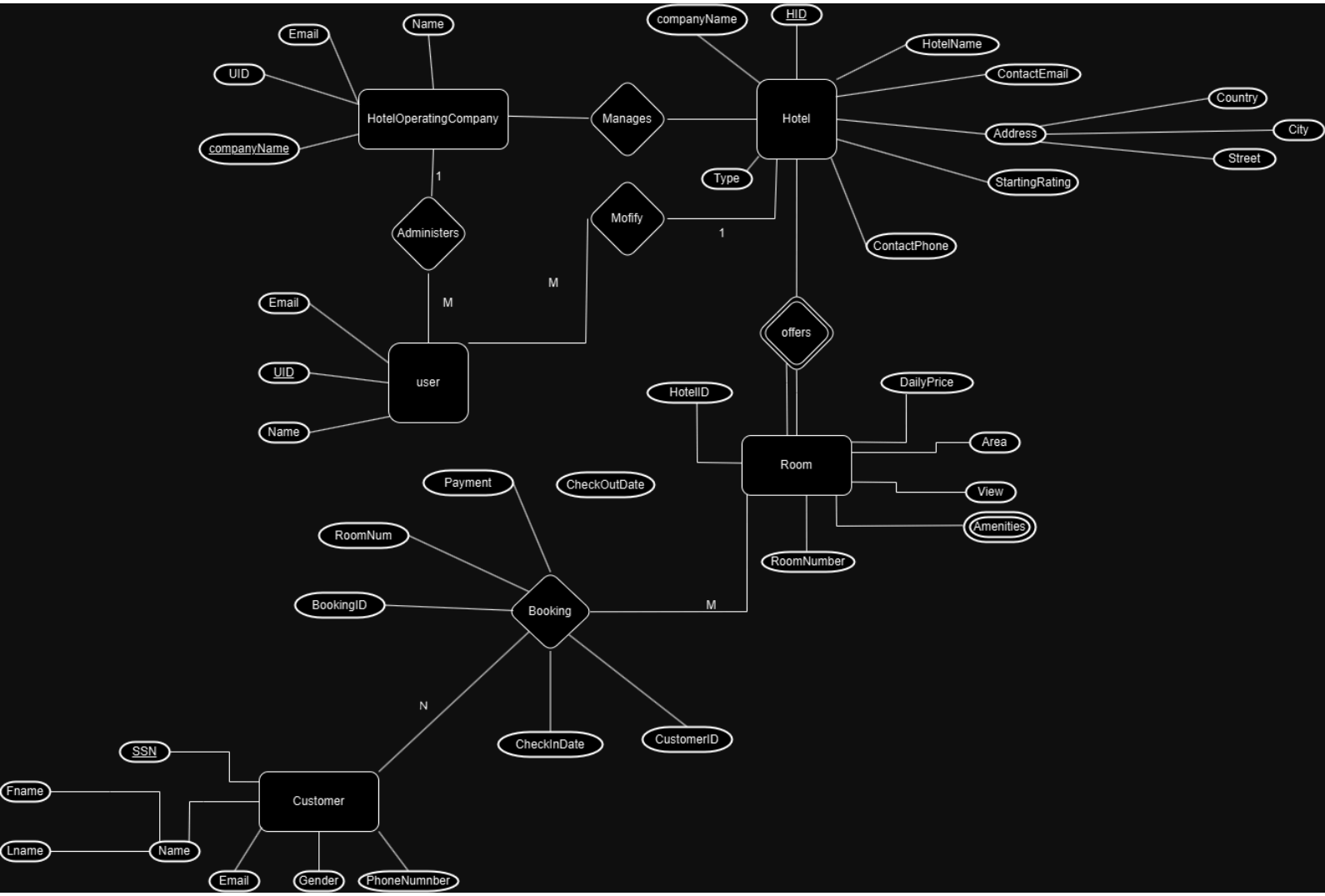
The purpose of this system database to provide and implement a relational model that serves the needs of an online hotel reservation system for an e-business company of medium size in scope. Such a system should enable several operating companies owning hotels to maintain their properties and also provide a channel through which clients can book hotel facilities.

Each operating company will administer various hotels and will be assigned a number of accounts on the system which will be secure. Via these accounts, company delegates will be able to input information relating to hotels such as rooms available for sale, rates, two and more, features, etc. During the interaction, they will also be able to refresh the available accommodation options in order for clients to always look for what is available.

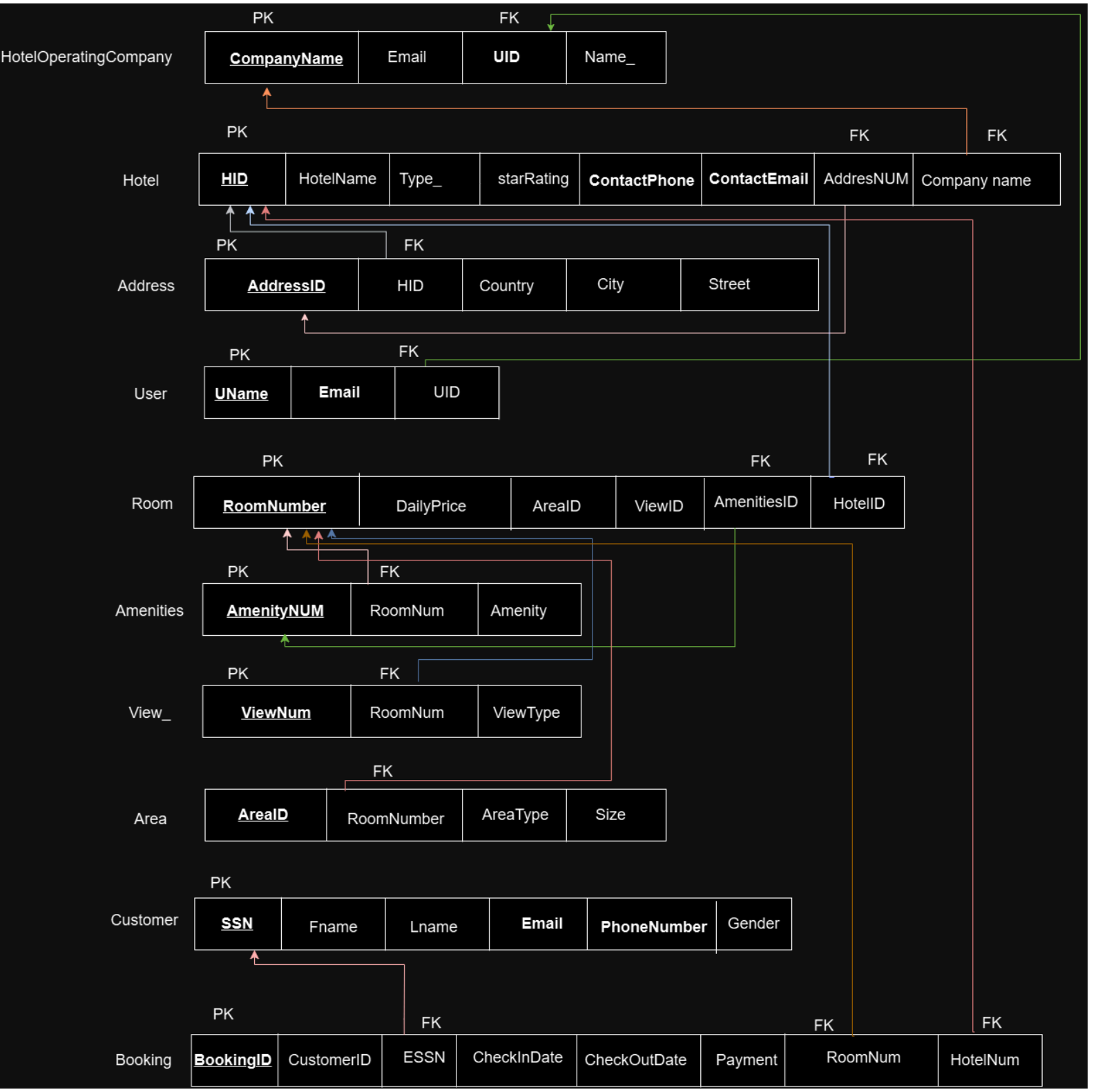
From the customer perspective, the system will make it easy to register and browse available rooms across various hotels for specific dates. Customers will be able to select and book rooms for their preferred stays . and To improve user convenience we can add a special types of booking means if you have a celebrations or similar stuff , and also they will have access to a personal booking history, which allows them to review past reservations and cancel or modify upcoming ones if necessary.

To equally match user expectations, the system must cope with different situations including but not limited to last minute booking of rooms, cancelation of room bookings, and multiple bookings.

2) An Entity–Relationship (ER) Diagram



3.0) Relational Model Diagram



4.0) Database Schema

4.1) Database Creation

```
DROP DATABASE IF EXISTS HotelOS;
CREATE DATABASE HotelOS;
USE HotelOS;
```

4.2.1) HotelOperatingCompany Table

This table stores the companies that manage hotels. Each company will have its own unique account to manage its hotels.

```
DROP TABLE IF EXISTS HotelOperatingCompany;
CREATE TABLE HotelOperatingCompany(
CompanyName VARCHAR(50)  NOT NULL,
UID         VARCHAR(50)  NOT NULL,
Email       VARCHAR(50)  NOT NULL,
Name_       VARCHAR(50)  NOT NULL,
--
PRIMARY KEY(CompanyName),
UNIQUE(UID)
);
```

Design Constraints:

* Non-null values enforced in the attributes:

  CompanyName,UID,Email,Name_.

4.2.2) Hotel Table

This table contains the details of individual hotels. Each hotel belongs to one operating

```
DROP TABLE IF EXISTS Hotel;
CREATE TABLE Hotel(
HID             VARCHAR(15)   NOT NULL,
ContactEmail    VARCHAR(50)   NOT NULL,
StarRating      FLOAT      ,
ContactPhone     VARCHAR(20)   NOT NULL,
HotelName       VARCHAR(50)   NOT NULL,
Type_           VARCHAR(15)   NOT NULL,
Address         VARCHAR(100)   NOT NULL,
CompanyName     VARCHAR(50)   NOT NULL,
--
PRIMARY KEY(HID),
UNIQUE(ContactEmail),
UNIQUE(ContactPhone),
FOREIGN KEY(CompanyName) REFERENCES
HotelOperatingCompany(CompanyName) ON DELETE CASCADE
);
```

Design Constraints:

* Non-null values enforced in the attributes:

HID,ContactEmail,ContactPhone,HotelName,Type_,Address,CompanyName.

* ContactEmail and ContactPhone is unique for every Hotel.
* Address has Multivalue :[country,street,city].
* Hotel Type_ has multivalue :[Resort,motel,Busniess].
* StarRating ranges from [0, 0.1 , 0.2 , …..,5].

4.2.3) Room Table

This table keeps track of individual rooms in each hotel, including their price, and features.

```
DROP TABLE IF EXISTS Room;
CREATE TABLE Room(
DailyPrice      DOUBLE       NOT NULL,
Area            VARCHAR(45)    NOT NULL,
View_           VARCHAR(45),
Amenities       VARCHAR(20),
RoomNumber      INT          NOT NULL,
HotelID       VARCHAR(15)    NOT NULL,
--
PRIMARY KEY(RoomNumber),
FOREIGN KEY(HotelID) REFERENCES Hotel(HID) ON DELETE CASCADE
);
```

Design Constraints:

* Non-non values enforced in the attributes:

  DailyPrice,Area,RoomNumber,HotelID.

* RoomNumber is unique.

* Room Views can be multiple like:

  [Forest,Woods,Beach,SeaSide,etc].

* Room Amenities can have: [Iron,Safe,Hair Drayer,etc].

* Referential integrity enforced the HotelID.

4.2.4) Customer Table

This table stores customer details, including their registration info .

```
DROP TABLE IF EXISTS Customer;
CREATE TABLE Customer(
SSN             INT         NOT NULL,
Fname           VARCHAR(20)   NOT NULL,
Lname           VARCHAR(20)   NOT NULL,
Email           VARCHAR(100)   NOT NULL,
Gender          CHAR,
PhoneNumber     VARCHAR(20)   NOT NULL,
--
PRIMARY KEY(SSN),
UNIQUE(Email),
UNIQUE(PhoneNumber)
);
```

Design Constraint:

* Non-null values are enforced for all attributes except Gender.

  Email and PhoneNumber are Unique.

4.2.5) Booking Table

This table records customer bookings for hotel rooms, including the dates.

```
DROP TABLE IF EXISTS Booking;
CREATE TABLE Booking(
BookingID        VARCHAR(45)   NOT NULL,
Payment          FLOAT      NOT NULL,
Check_IN         DATETIME   NOT NULL,
Check_OUT        DATETIME   NOT NULL,
RoomNum          INT        NOT NULL,
CustomerID       INT        NOT NULL,
HotelNum         VARCHAR(15)   NOT NULL,
ESSN             INT        NOT NULL,
--
PRIMARY KEY(BookingID),
UNIQUE(RoomNum),
FOREIGN KEY(HotelNum ) REFERENCES Hotel(HID) ON DELETE CASCADE,
FOREIGN KEY(ESSN) REFERENCES Customer(SSN) ON DELETE CASCADE,
FOREIGN KEY(RoomNum) REFERENCES Room(RoomNumber) ON DELETE CASCADE
);
```

Design Constraints:

* Non-null values are enforced for all attributes.
* RoomNum is unique.
* Referential integrity enforced on the

  ESSN,HotelNum,RoomNum.

4.2.6) User Table

This table is for admin accounts that belong to operating companies. Admins can manage hotels and room availability.

```
DROP TABLE IF EXISTS User_;
CREATE TABLE User_(
UID             VARCHAR(20)    NOT NULL,
Email           VARCHAR(100)    NOT NULL,
UName           VARCHAR(30)    NOT NULL,
--
FOREIGN KEY(UID) REFERENCES HotelOperatingCompany(UID) ON
DELETE CASCADE,
UNIQUE(Uname),
UNIQUE(Email)
);
Design Constraints:
*   Non-null values enforced for all attributes.
*   User name (Uname) and Email is unique for every user.
```

Constraints and Indexes

Preventing Duplicate Bookings for the Same Room and Date Range:

```
ALTER TABLE Booking
ALTER TABLE Booking
ADD CONSTRAINT UC_Booking UNIQUE (RoomNum, Check_In,
Check_Out);
```

Create Indexes for Better Performance:

```
CREATE INDEX IDX_Hotel_Location ON Hotel(Address);
```

4.2.7) Address Table

```
DROP TABLE IF EXISTS Address;
CREATE TABLE Address(
AddressID        INT   AUTO_INCREMENT  NOT NULL,
HID             VARCHAR(15)   NOT NULL,
Country         VARCHAR(15)   NOT NULL,
City            VARCHAR(15)   NOT NULL,
Street          VARCHAR(40)   NOT NULL,
--
PRIMARY KEY(AddressID),
FOREIGN KEY(HID) REFERENCES Hotel(HID) ON DELETE CASCADE

);
```

* Non-null values enforced to all attributes.

Referential integrity enforced the HID.

4.2.8) Amenities Table

Information about the Amenities that comes with the room

```
DROP TABLE IF EXISTS Amenities;
CREATE TABLE Amenities(
AmenityNum    VARCHAR(20),
RoomNumber    INT NOT NULL,
Amenity       VARCHAR(50) NOT NULL,
--
PRIMARY KEY(AmenityNum),
FOREIGN KEY(RoomNumber) REFERENCES Room(RoomNumber) ON DELETE
CASCADE
);
```

Design Constraints:

Non-non values enforced in the attributes:

RoomNumber,Amenity.

Referential integrity enforced the RoomNumber.

4.2.9) View Table

Information about the view that companies the room

```
DROP TABLE IF EXISTS View_;
CREATE TABLE View_(
ViewNum         INT   AUTO_INCREMENT  NOT NULL,
RoomNumber      INT        NOT NULL,
ViewType        TEXT,
--
PRIMARY KEY(ViewNum),
FOREIGN KEY(RoomNumber) REFERENCES Room(RoomNumber) ON DELETE
CASCADE
);
```

Design Constraints:

* Non-non values enforced in the attributes:

RoomNumber.

* Referential integrity enforced the RoomNumber.

* Auto increments for ViewNum.

4.2.10) Area Table

Area information regarding the room

```
DROP TABLE IF EXISTS Area;
CREATE TABLE Area(
AreaID          INT    AUTO_INCREMENT   NOT NULL,
RoomNumber      INT         NOT NULL,
AreaType        TEXT,
Size            VARCHAR(100)   NOT NULL,
--
PRIMARY KEY(AreaID),
FOREIGN KEY(RoomNumber) REFERENCES Room(RoomNumber) ON DELETE
CASCADE
);
```

Design Constraints:

* Non-non values enforced in the attributes:

  RoomNumber , Size , AreaID .

* Referential integrity enforced the RoomNumber.

* AreaType is text so the hotel can fill up some relative info about the area.

General explanation for what we did so far:

· **Database Name**: The database is called `HotelOperaingSystem` to reflect its purpose.

· **Relationships Between Tables**:

- `OperatingCompany` is the parent table for `Hotel`.
- `Hotel` is the parent table for `Room`.
- `Customer` and `Room` are linked through `Booking`.

· **Constraints**:

- Primary keys (`PK`) ensure each record is unique.
- Foreign keys (`FK`) link related tables and enforce referential integrity.
- Unique constraints (`UK`) prevent duplicate data, like bookings for the same room and date.

· **Indexes**: Indexes on commonly searched fields (like location and availability) make the system faster when handling a large amount of data.

5.0) Database population

5.1) Insertion for HotelOperatingCompany :

```
INSERT INTO HotelOperatingCompany (CompanyName, UID, Email, Name_)
VALUES
('LuxuryStay', 'UID001', 'luxury@luxury.com', 'John Doe'),
('EliteHotels', 'UID002', 'elite@elite.com', 'Jane Smith'),
('ComfortInn', 'UID003', 'comfort@comfort.com', 'Alice Brown'),
('UrbanSuites', 'UID004', 'urban@urban.com', 'Robert Johnson'),
('ParadiseHotels', 'UID005', 'paradise@paradise.com', 'Emily Davis'),
('CozyHotels', 'UID006', 'cozy@cozy.com', 'Sophia Wilson'),
('OceanicResorts', 'UID007', 'oceanic@oceanic.com', 'James Miller'),
('SkylineHotels', 'UID008', 'skyline@skyline.com', 'Charlotte Taylor');
```

CompanyName	UID	Email	Name_
ComfortInn	UID003	comfort@comfort.com	Alice Brown
CozyHotels	UID006	cozy@cozy.com	Sophia Wilson
EliteHotels	UID002	elite@elite.com	Jane Smith
LuxuryStay	UID001	luxury@luxury.com	John Doe
OceanicResorts	UID007	oceanic@oceanic.com	James Miller
ParadiseHotels	UID005	paradise@paradise.com	Emily Davis
SkylineHotels	UID008	skyline@skyline.com	Charlotte Taylor
UrbanSuites	UID004	urban@urban.com	Robert Johnson

5.2) Insertion for Hotel :

```
INSERT INTO Hotel (HID, ContactEmail, StarRating, ContactPhone,
HotelName, Type_, AddressNum, CompanyName) VALUES
('HID001', 'palm@luxury.com', 5, '1234567890', 'Palm Paradise',
'Resort', 1, 'LuxuryStay'),
('HID002', 'elite1@elite.com', 4, '1234567891', 'Elite Sharjah',
'Hotel', 2, 'EliteHotels'),
('HID003', 'comfort1@comfort.com', 3.5, '1234567892', 'Comfort
Corniche', 'Inn', 3, 'ComfortInn'),
('HID004', 'urban1@urban.com', 4.2, '1234567893', 'Urban Stay', 'Hotel',
4, 'UrbanSuites'),
('HID005', 'paradise1@paradise.com', 5, '1234567894', 'Paradise NYC',
'Resort', 5, 'ParadiseHotels'),
('HID006', 'cozy1@cozy.com', 3.8, '1234567895', 'Cozy Oxford', 'Hotel',
6, 'CozyHotels'),
('HID007', 'skyline1@skyline.com', 4.7, '1234567896', 'Skyline View',
'Hotel', 7, 'SkylineHotels'),
('HID008', 'elite2@elite.com', 4.5, '1234567897', 'Elite Al Buhaira',
'Hotel', 8, 'EliteHotels');
```

HID	ContactEmail	StarRating	ContactPhone	HotelName	Type_	AddressNum	CompanyName
HID001	palm@luxury.com	5	1234567890	Palm Paradise	Resort	1	LuxuryStay
HID002	elite1@elite.com	4	1234567891	Elite Sharjah	Hotel	2	EliteHotels
HID003	comfort1@comfort.com	3.5	1234567892	Comfort Corniche	Inn	3	ComfortInn
HID004	urban1@urban.com	4.2	1234567893	Urban Stay	Hotel	4	UrbanSuites
HID005	paradise1@paradise.com	5	1234567894	Paradise NYC	Resort	5	ParadiseHotels
HID006	cozy1@cozy.com	3.8	1234567895	Cozy Oxford	Hotel	6	CozyHotels
HID007	skyline1@skyline.com	4.7	1234567896	Skyline View	Hotel	7	SkylineHotels
HID008	elite2@elite.com	4.5	1234567897	Elite Al Buhaira	Hotel	8	EliteHotels

5.3)Insertion for Room table:

```
-- Insert data into Room
INSERT INTO Room (DailyPrice, AreaID, ViewID, AmenitiesID, RoomNumber,
HotelID) VALUES
(200, 1, 'V001', 'A001', 101, 'HID001'),
(150, 2, 'V002', 'A002', 102, 'HID002'),
(120, 3, 'V003', 'A003', 103, 'HID003'),
(180, 4, 'V004', 'A004', 104, 'HID004'),
(250, 5, 'V005', 'A005', 105, 'HID005'),
(100, 6, 'V006', 'A006', 106, 'HID006'),
(220, 7, 'V007', 'A007', 107, 'HID007'),
(190, 8, 'V008', 'A008', 108, 'HID008');
```

DailyPrice	AreaID	ViewID	AmenitiesID	RoomNumber	HotelID
200	1	V001	A001	101	HID001
150	2	V002	A002	102	HID002
120	3	V003	A003	103	HID003
180	4	V004	A004	104	HID004
250	5	V005	A005	105	HID005
100	6	V006	A006	106	HID006
220	7	V007	A007	107	HID007
190	8	V008	A008	108	HID008

5.4)Insertion for Customer table:

```
-- Insert data into Customer
INSERT INTO Customer (SSN, Fname, Lname, Email, Gender, PhoneNumber)
VALUES
(123456789, 'Michael', 'Scott', 'michael@dundermifflin.com', 'M',
'555-1111'),
(987654321, 'Dwight', 'Schrute', 'dwight@dundermifflin.com', 'M',
'555-2222'),
(456789123, 'Jim', 'Halpert', 'jim@dundermifflin.com', 'M', '555-3333'),
(321654987, 'Pam', 'Beesly', 'pam@dundermifflin.com', 'F', '555-4444'),
(789123456, 'Stanley', 'Hudson', 'stanley@dundermifflin.com', 'M',
'555-5555'),
(654321789, 'Angela', 'Martin', 'angela@dundermifflin.com', 'F',
'555-6666'),
(147258369, 'Kevin', 'Malone', 'kevin@dundermifflin.com', 'M',
'555-7777'),
(369258147, 'Oscar', 'Martinez', 'oscar@dundermifflin.com', 'M',
'555-8888');
```

SSN	Fname	Lname	Email	Gender	PhoneNumber
123456789	Michael	Scott	michael@dundermifflin.com	M	555-1111
147258369	Kevin	Malone	kevin@dundermifflin.com	M	555-7777
321654987	Pam	Beesly	pam@dundermifflin.com	F	555-4444
369258147	Oscar	Martinez	oscar@dundermifflin.com	M	555-8888
456789123	Jim	Halpert	jim@dundermifflin.com	M	555-3333
654321789	Angela	Martin	angela@dundermifflin.com	F	555-6666
789123456	Stanley	Hudson	stanley@dundermifflin.com	M	555-5555
987654321	Dwight	Schrute	dwight@dundermifflin.com	M	555-2222

5.5)Insertion for Booking table:

```
-- Insert data into Booking
INSERT INTO Booking (BookingID, Payment, Check_IN, Check_OUT, RoomNum, CustomerID, HotelNum, ESSN)
VALUES
('B001', 400, '2024-11-01 14:00:00', '2024-11-05 12:00:00', 101, 123456789, 'HID001', 123456789),
('B002', 300, '2024-11-03 14:00:00', '2024-11-07 12:00:00', 102, 987654321, 'HID002', 987654321),
('B003', 360, '2024-11-02 14:00:00', '2024-11-06 12:00:00', 103, 456789123, 'HID003', 456789123),
('B004', 720, '2024-11-04 14:00:00', '2024-11-08 12:00:00', 104, 321654987, 'HID004', 321654987),
('B005', 1000, '2024-11-05 14:00:00', '2024-11-10 12:00:00', 105, 789123456, 'HID005', 789123456),
('B006', 200, '2024-11-06 14:00:00', '2024-11-09 12:00:00', 106, 123456789, 'HID006', 123456789),
('B007', 440, '2024-11-07 14:00:00', '2024-11-11 12:00:00', 107, 147258369, 'HID005', 147258369),
('B008', 600, '2024-11-08 14:00:00', '2024-11-12 12:00:00', 108, 123456789, 'HID008', 123456789),
('B009', 500, '2024-11-09 14:00:00', '2024-11-13 12:00:00', 109, 456789123, 'HID001', 456789123),
('B010', 150, '2024-11-10 14:00:00', '2024-11-12 12:00:00', 110, 123456789, 'HID002', 123456789);
```

BookingID	Payment	Check_IN	Check_OUT	RoomNum	CustomerID	HotelNum	ESSN
B001	400	11/1/2024 14:00	11/5/2024 12:00	101	123456789	HID001	123456789
B002	300	11/3/2024 14:00	11/7/2024 12:00	102	987654321	HID002	987654321
B003	360	11/2/2024 14:00	11/6/2024 12:00	103	456789123	HID003	456789123
B004	720	11/4/2024 14:00	11/8/2024 12:00	104	321654987	HID004	321654987
B005	1000	11/5/2024 14:00	11/10/2024 12:00	105	789123456	HID005	789123456
B006	200	11/6/2024 14:00	11/9/2024 12:00	106	123456789	HID006	123456789
B007	440	11/7/2024 14:00	11/11/2024 12:00	107	147258369	HID005	147258369
B008	600	11/8/2024 14:00	11/12/2024 12:00	108	123456789	HID008	123456789
B009	500	11/9/2024 14:00	11/13/2024 12:00	109	456789123	HID001	456789123
B010	150	11/10/2024 14:00	11/12/2024 12:00	110	123456789	HID002	123456789

5.6) Insertion for User_:

```
-- Insert data into User_
INSERT INTO User_ (UID, Email, UName) VALUES
('UID001', 'admin1@luxury.com', 'admin1'),
('UID002', 'admin2@elite.com', 'admin2'),
('UID003', 'admin3@comfort.com', 'admin3'),
('UID004', 'admin4@urban.com', 'admin4'),
('UID005', 'admin5@paradise.com', 'admin5'),
('UID006', 'admin6@cozy.com', 'admin6'),
('UID007', 'admin7@oceanic.com', 'admin7'),
('UID008', 'admin8@skyline.com', 'admin8');
```

UID	Email	UName
UID001	admin1@luxury.com	admin1
UID002	admin2@elite.com	admin2
UID003	admin3@comfort.com	admin3
UID004	admin4@urban.com	admin4
UID005	admin5@paradise.com	admin5
UID006	admin6@cozy.com	admin6
UID007	admin7@oceanic.com	admin7
UID008	admin8@skyline.com	admin8

5.7)Insertion for Address table:

```
-- Insert data into Address
INSERT INTO Address (HID, Country, City, Street) VALUES
('HID001', 'UAE', 'Dubai', 'Palm Jumeirah St'),
('HID002', 'UAE', 'Sharjah', 'Al Majaz St'),
('HID003', 'UAE', 'Abu Dhabi', 'Corniche St'),
('HID004', 'UAE', 'Sharjah', 'University City Rd'),
('HID005', 'USA', 'New York', '5th Avenue'),
('HID006', 'UK', 'London', 'Oxford St'),
('HID007', 'UAE', 'Dubai', 'Sheikh Zayed Rd'),
('HID008', 'UAE', 'Sharjah', 'Al Buhaira St');
```

AddressID	HID	Country	City	Street
1	HID001	UAE	Dubai	Palm Jumeirah St
2	HID002	UAE	Sharjah	Al Majaz St
3	HID003	UAE	Abu Dhabi	Corniche St
4	HID004	UAE	Sharjah	University City Rd
5	HID005	USA	New York	5th Avenue
6	HID006	UK	London	Oxford St
7	HID007	UAE	Dubai	Sheikh Zayed Rd
8	HID008	UAE	Sharjah	Al Buhaira St

5.8) Insertion for Amenities table:

```
-- Insert data into Amenities
INSERT INTO Amenities (AmenityNum, RoomNumber, Amenity) VALUES
('A001', 101, 'Free WiFi'),
('A002', 102, 'Breakfast Included'),
('A003', 103, 'Gym Access'),
('A004', 104, 'Pool Access'),
('A005', 105, 'Ocean View'),
('A006', 106, 'Room Service'),
('A007', 107, 'Airport Shuttle'),
('A008', 108, 'Spa Access');
```

AmenityNum	RoomNumber	Amenity
A001	101	Free WiFi
A002	102	Breakfast Included
A003	103	Gym Access
A004	104	Pool Access
A005	105	Ocean View
A006	106	Room Service
A007	107	Airport Shuttle
A008	108	Spa Access

5.9) Insertion for View_ table:

```
-- Insert data into View_
INSERT INTO View_ (ViewNum, RoomNumber, ViewType) VALUES
(1, 101, 'Sea View'),
(2, 102, 'City View'),
(3, 103, 'Garden View'),
(4, 104, 'Skyline View'),
(5, 105, 'Oceanfront View'),
(6, 106, 'Courtyard View'),
(7, 107, 'Mountain View'),
(8, 108, 'Lake View');
```

ViewNum	RoomNumber	ViewType
1	101	Sea View
2	102	City View
3	103	Garden View
4	104	Skyline View
5	105	Oceanfront View
6	106	Courtyard View
7	107	Mountain View
8	108	Lake View

5.10) Insertion for Area table:

```
-- Insert data into Area
INSERT INTO Area (AreaID, RoomNumber, AreaType, Size) VALUES
(1, 101, 'Bedroom', '25 sqm'),
(2, 102, 'Bedroom', '20 sqm'),
(3, 103, 'Living Room', '30 sqm'),
(4, 104, 'Living Room', '35 sqm'),
(5, 105, 'Suite', '50 sqm'),
(6, 106, 'Standard', '15 sqm'),
(7, 107, 'Deluxe', '40 sqm'),
(8, 108, 'Executive', '45 sqm');
```

AreaID	RoomNumber	AreaType	Size
1	101	Bedroom	25 sqm
2	102	Bedroom	20 sqm
3	103	Living Room	30 sqm
4	104	Living Room	35 sqm
5	105	Suite	50 sqm
6	106	Standard	15 sqm
7	107	Deluxe	40 sqm
8	108	Executive	45 sqm

6.0) Queries / Reports

6.1) The name of companies that run at least 2 hotels.

```
SELECT CompanyName
FROM Hotel
GROUP BY CompanyName
HAVING COUNT(HID) >= 2;
```

Output:

CompanyName
EliteHotels

6.2) The name of the hotels that received no booking.

```
SELECT HotelName
FROM Hotel
WHERE HID NOT IN (SELECT HotelNum FROM Booking);
```

Output:

HotelName
Skyline View

6.3) The names of Customers booked with at least 3 distinct hotels.

```
SELECT C.Fname, C.Lname, C.Email
FROM Customer C
JOIN Booking B ON C.SSN = B.ESSN
GROUP BY C.SSN, C.Fname, C.Lname, C.Email
HAVING COUNT(DISTINCT B.HotelNum) >= 3;
```

Output:

Fname	Lname	Email
Michael	Scott	michael@dundermifflin.com

6.4) The name of companies that run some hotel located in Sharjah.

```
SELECT DISTINCT H.CompanyName
FROM Hotel H
JOIN Address A ON H.AddressNum = A.AddressID
WHERE A.City = 'Sharjah';
```

Output:

CompanyName
EliteHotels
UrbanSuites

6.5) The average number of bookings per customer.

```
SELECT ROUND(COUNT(*) / COUNT(DISTINCT ESSN), 2) AS
AvgBookingsPerCustomer
FROM Booking;
```

Output:

AvgBookingsPerCustomer
1.67

7) Group Member Contribution Form

No	Student Name	Student ID	Contribution
1	Hamza Luai	U22105870	25%
2	Ahmad Al Dosari	U22106543	25%
3	Mohammad Raed	U22105630	25%
4	Mahaz Ishtiaq khan	U22200217	25%