

Milestone 2: Database Design and Queries for Travel Agency System

1. **Entity-Relationship Diagram (ERD)**

Entities and Attributes:

- **Clients**:
 - ClientID (Primary Key)
 - Name
 - Email
 - Phone
 - Address
- **Trips**:
 - TripID (Primary Key)
 - Destination
 - Date
 - Price
 - Description
- **Bookings**:
 - BookingID (Primary Key)
 - ClientID (Foreign Key)
 - TripID (Foreign Key)
 - BookingDate
 - Status
- **Admins**:
 - AdminID (Primary Key)
 - Username
 - Password
- **Notifications**:
 - NotificationID (Primary Key)
 - ClientID (Foreign Key)
 - Message
 - SentDate
- **Reviews**:
 - ReviewID (Primary Key)
 - ClientID (Foreign Key)
 - TripID (Foreign Key)
 - Rating
 - Comment

Relationships:

- Clients to Bookings: One-to-Many
- Trips to Bookings: One-to-Many
- Clients to Reviews: One-to-Many
- Trips to Reviews: One-to-Many
- Clients to Notifications: One-to-Many

2. **Relational Database Schema**

Clients Table:

...

```
CREATE TABLE Clients (  
    ClientID INT PRIMARY KEY,  
    Name VARCHAR(100),  
    Email VARCHAR(100),  
    Phone VARCHAR(15),  
    Address TEXT  
);  
...
```

Trips Table:

...

```
CREATE TABLE Trips (  
    TripID INT PRIMARY KEY,  
    Destination VARCHAR(100),  
    Date DATE,  
    Price DECIMAL(10, 2),  
    Description TEXT  
);  
...
```

Bookings Table:

...

```
CREATE TABLE Bookings (  
    BookingID INT PRIMARY KEY,  
    ClientID INT,  
    TripID INT,  
    BookingDate DATE,  
    Status VARCHAR(20),  
    FOREIGN KEY (ClientID) REFERENCES Clients(ClientID),  
    FOREIGN KEY (TripID) REFERENCES Trips(TripID)  
);
```

...

Admins Table:

...

```
CREATE TABLE Admins (  
    AdminID INT PRIMARY KEY,  
    Username VARCHAR(50),  
    Password VARCHAR(100)  
);  
...
```

Notifications Table:

...

```
CREATE TABLE Notifications (  
    NotificationID INT PRIMARY KEY,  
    ClientID INT,  
    Message TEXT,  
    SentDate DATE,  
    FOREIGN KEY (ClientID) REFERENCES Clients(ClientID)  
);  
...
```

Reviews Table:

...

```
CREATE TABLE Reviews (  
    ReviewID INT PRIMARY KEY,  
    ClientID INT,  
    TripID INT,  
    Rating INT,  
    Comment TEXT,  
    FOREIGN KEY (ClientID) REFERENCES Clients(ClientID),  
    FOREIGN KEY (TripID) REFERENCES Trips(TripID)  
);  
...
```

3. **SQL Queries**

Insert Queries:

1. Add a new client:

```sql

```
INSERT INTO Clients (ClientID, Name, Email, Phone, Address)
VALUES (1, 'John Doe', 'john.doe@example.com', '1234567890', '123 Elm Street');
```

...

2. Add a new trip:

```
```sql
```

```
INSERT INTO Trips (TripID, Destination, Date, Price, Description)
```

```
VALUES (1, 'Paris', '2024-06-15', 1500.00, 'A week-long trip to Paris including sightseeing and accommodations.');
```

```
```
```

#### Update Queries:

1. Update a trip's price:

```
```sql
```

```
UPDATE Trips
```

```
SET Price = 1600.00
```

```
WHERE TripID = 1;
```

```
```
```

2. Update client details:

```
```sql
```

```
UPDATE Clients
```

```
SET Phone = '0987654321'
```

```
WHERE ClientID = 1;
```

```
```
```

#### Delete Queries:

1. Delete a review:

```
```sql
```

```
DELETE FROM Reviews
```

```
WHERE ReviewID = 1;
```

```
```
```

2. Delete a trip:

```
```sql
```

```
DELETE FROM Trips
```

```
WHERE TripID = 1;
```

```
```
```

#### Select Queries:

1. Fetch all trips for a specific destination:

```
```sql
```

```
SELECT *
```

```
FROM Trips
```

```
WHERE Destination = 'Paris';
```

```
```
```

2. Fetch all bookings for a specific client:

```
```sql
SELECT *
FROM Bookings
WHERE ClientID = 1;
```
```

3. Fetch the average rating for a trip:

```
```sql
SELECT AVG(Rating) AS AverageRating
FROM Reviews
WHERE TripID = 1;
```
```

4. Fetch all notifications sent to a client:

```
```sql
SELECT *
FROM Notifications
WHERE ClientID = 1;
```
```

5. Fetch client details along with their bookings:

```
```sql
SELECT Clients.Name, Clients.Email, Bookings.BookingID, Bookings.BookingDate,
Bookings.Status
FROM Clients
JOIN Bookings ON Clients.ClientID = Bookings.ClientID;
```
```

6. Fetch trips with their average ratings:

```
```sql
SELECT Trips.TripID, Trips.Destination, Trips.Price, AVG(Reviews.Rating) AS AverageRating
FROM Trips
LEFT JOIN Reviews ON Trips.TripID = Reviews.TripID
GROUP BY Trips.TripID, Trips.Destination, Trips.Price;
```
```

7. Fetch bookings including trip details for a client:

```
```sql
SELECT Bookings.BookingID, Bookings.BookingDate, Bookings.Status, Trips.Destination,
Trips.Date, Trips.Price
FROM Bookings
JOIN Trips ON Bookings.TripID = Trips.TripID
```
```

WHERE Bookings.ClientID = 1;

---

---

#### ### 4. \*\*Requirement-Query Matrix\*\*

| Requirement | Query Type          | Example Query                                  |  |
|-------------|---------------------|------------------------------------------------|--|
| REQ001      | Select Query        | Fetch all trips for a destination              |  |
| REQ002      | Insert Query        | Add a new booking                              |  |
| REQ003      | Insert Query        | Add a new client                               |  |
| REQ005      | Update/Delete Query | Update a trip's price or delete a trip         |  |
| REQ007      | Insert Query        | Add a new review                               |  |
| REQ009      | Select Query        | Fetch all notifications for a client           |  |
| REQ010      | Join Query          | Fetch client details along with their bookings |  |
| REQ011      | Join with Aggregate | Fetch trips with their average ratings         |  |