Table of Contents

Sr no.	Topic	Page no.
1	Storyline	4
2	Components of Database Design	5
3	Entity Relationship Diagram	9
4	Relational Model	10
5	Normalization	11
6	SQL Queries	12
7	Project Demonstration	18
8	Self-learning beyond classroom	28
9	Learning from the project	29
10	Challenges faced	30
11	Conclusion	31

I. Storyline

Travel Booking Management

Introduction:
In the bustling world of travel, efficient booking management is essential for ensuring seamless journeys and satisfied customers. To address the complexities of this industry, we've developed a comprehensive travel booking management system leveraging the power of a relational database. Our system caters to various modes of transportation, including flights, trains, buses, cars, and cruises, providing a one-stop solution for travelers' diverse needs.

Audience:
Seasoned travelers who frequently embark on adventures around the globe. Frustrated by the cumbersome process of booking trips across different platforms, we've provided a simpler solution. Enter our travel booking management system – a user-friendly platform designed to streamline your booking experience and enhance your journey from start to finish.

Database Components:
1. Customer Management: The system should maintain comprehensive records of customers, including their personal details and preferred modes of travel.
2. Employee Management: Efficient workforce management is crucial, necessitating a database to store employee information, such as roles, departments, and salaries.
3. Destination Information: A repository of destination details, including names, distances, and countries, is essential for planning travel routes and itineraries.
4. Transportation Options: The system should support various modes of transportation, such as flights, trains, buses, cars, and cruises, along with relevant scheduling and availability data.
5. Payment Processing: Seamless payment processing capabilities, including different payment methods and transaction records, are vital for completing bookings securely.
6. Booking Management: The system should facilitate the booking process, managing reservations, tracking booking statuses, and generating booking confirmations.

Implementation:
Our system is built on a robust relational database, featuring tables for customer details, employee information, destinations, transportation options, payment records, and bookings. Using SQL queries, we can efficiently retrieve, manipulate, and analyze data to provide a seamless booking experience for users.

Conclusion
With our travel booking management system, one can now effortlessly plan adventures, book transportation, and manage bookings from a single platform. By leveraging the power of data and technology, we're revolutionizing the way travellers navigate the world, making their journeys smoother, more enjoyable, and truly unforgettable.

II. Components of Database Design

Tables & attributes

1. CustomerDetails_reneeka:

- CustomerID INT PRIMARY KEY,
- FullName VARCHAR(50),
- Email VARCHAR(50),
- Phone VARCHAR(10),
- Identity_proof VARCHAR(100)

2. Employee_reneeka:

- EmployeeID INT PRIMARY KEY,
- Emp_name VARCHAR(50),
- Department VARCHAR(50),
- Emp_salary INT(10),
- Emp_Position VARCHAR(50)

3. Destination:

- DestinationID INT PRIMARY KEY,
- DestinationName VARCHAR(50),
- Distance INT,
- Country VARCHAR(50)

4. Trains_reneeka:

- TrainID INT PRIMARY KEY,
- TrainName VARCHAR(50),
- T_DepartureTime TIME,
- T_ArrivalTime TIME,
- T_Origin VARCHAR(50),
- T_Destination VARCHAR(50),
- DestinationID INT,

- FOREIGN KEY (DestinationID) REFERENCES Destination(DestinationID)

5. Flights_amishi:

- FlightID INT PRIMARY KEY,
- FlightNumber VARCHAR(10),
- Airline VARCHAR(50),
- FL_DepartureTime DATETIME,
- FL_ArrivalTime DATETIME,
- FL_Origin VARCHAR(50),
- FL_Destination VARCHAR(50),
- DestinationID INT,
- FOREIGN KEY (DestinationID) REFERENCES Destination(DestinationID)

6. Buses_amishi:

- BusID INT PRIMARY KEY,
- BusNumber VARCHAR(10),
- Operator VARCHAR(50),
- B_DepartureTime DATETIME,
- B_ArrivalTime DATETIME,
- B_Origin VARCHAR(50),
- B_Destination VARCHAR(50),
- DestinationID INT,
- FOREIGN KEY (DestinationID) REFERENCES Destination(DestinationID)

7. Cruise_amishi:

- CruiseID INT PRIMARY KEY,
- CruiseName VARCHAR(50),
- CR_DepartureTime DATETIME,
- CR_ArrivalTime DATETIME,
- CR_Origin VARCHAR(50),
- CR_Destination VARCHAR(50),

- DestinationID INT,
- FOREIGN KEY (DestinationID) REFERENCES Destination(DestinationID)

8. Car_chahel:

- CarID INT PRIMARY KEY,
- CarModel VARCHAR(50),
- PlateNumber VARCHAR(20),
- Capacity INT,
- C_Origin VARCHAR(50),
- C_Destination VARCHAR(50),
- DestinationID INT,
- FOREIGN KEY (DestinationID) REFERENCES Destination(DestinationID)

9. Payment_chahel:

- PaymentID INT PRIMARY KEY,
- PaymentMethod VARCHAR(50),
- PaymentAmount DECIMAL(10, 2),
- PaymentDate DATE,
- CustomerID INT,
- FOREIGN KEY (CustomerID) REFERENCES CustomerDetails_reneeka(CustomerID)

10. Booking_chahel:

- BookingID INT PRIMARY KEY,
- CustomerID INT,
- Booking_status VARCHAR(20),
- PaymentID INT,
- Booking_time DATETIME,
- Transport VARCHAR(50) NOT NULL,
- Origin VARCHAR(50) NOT NULL,
- Destination VARCHAR(50) NOT NULL,

- FOREIGN KEY (CustomerID) REFERENCES CustomerDetails_reneeka(CustomerID),
- FOREIGN KEY (PaymentID) REFERENCES Payment_chahel(PaymentID)

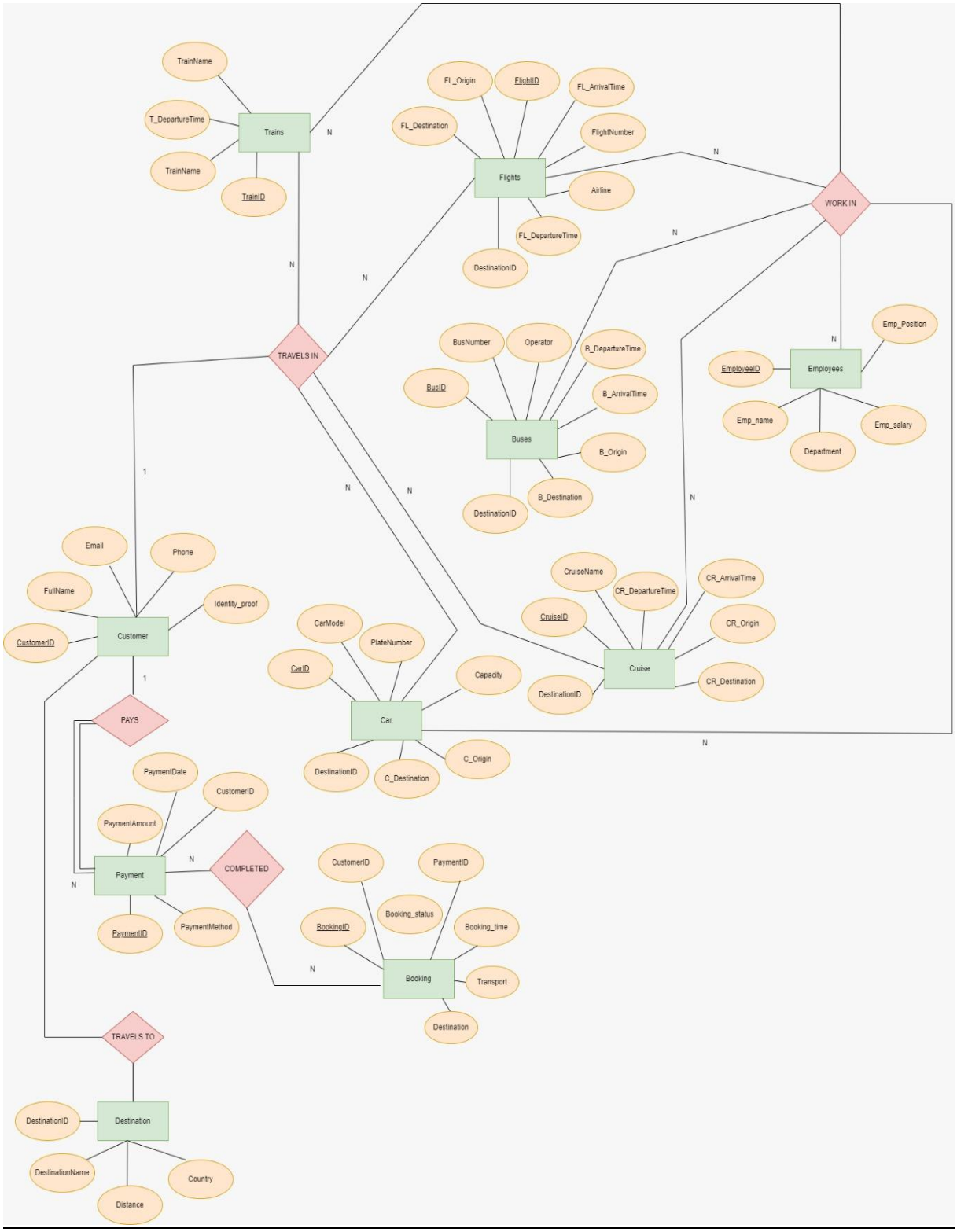
Relationships:

- CustomerDetails_reneeka and Payment_chahel:
    - One-to-Many relationship (One customer can make multiple payments)
- CustomerDetails_reneeka and Booking_chahel:
    - One-to-Many relationship (One customer can make multiple bookings)
- Employee_reneeka and Booking_chahel:
    - One-to-Many relationship (One employee can be associated with multiple bookings)
- Destination and (Trains_reneeka, Flights_amishi, Buses_amishi, Cruise_amishi, Car_chahel):
    - One-to-Many relationships (Each destination can have multiple transport options)

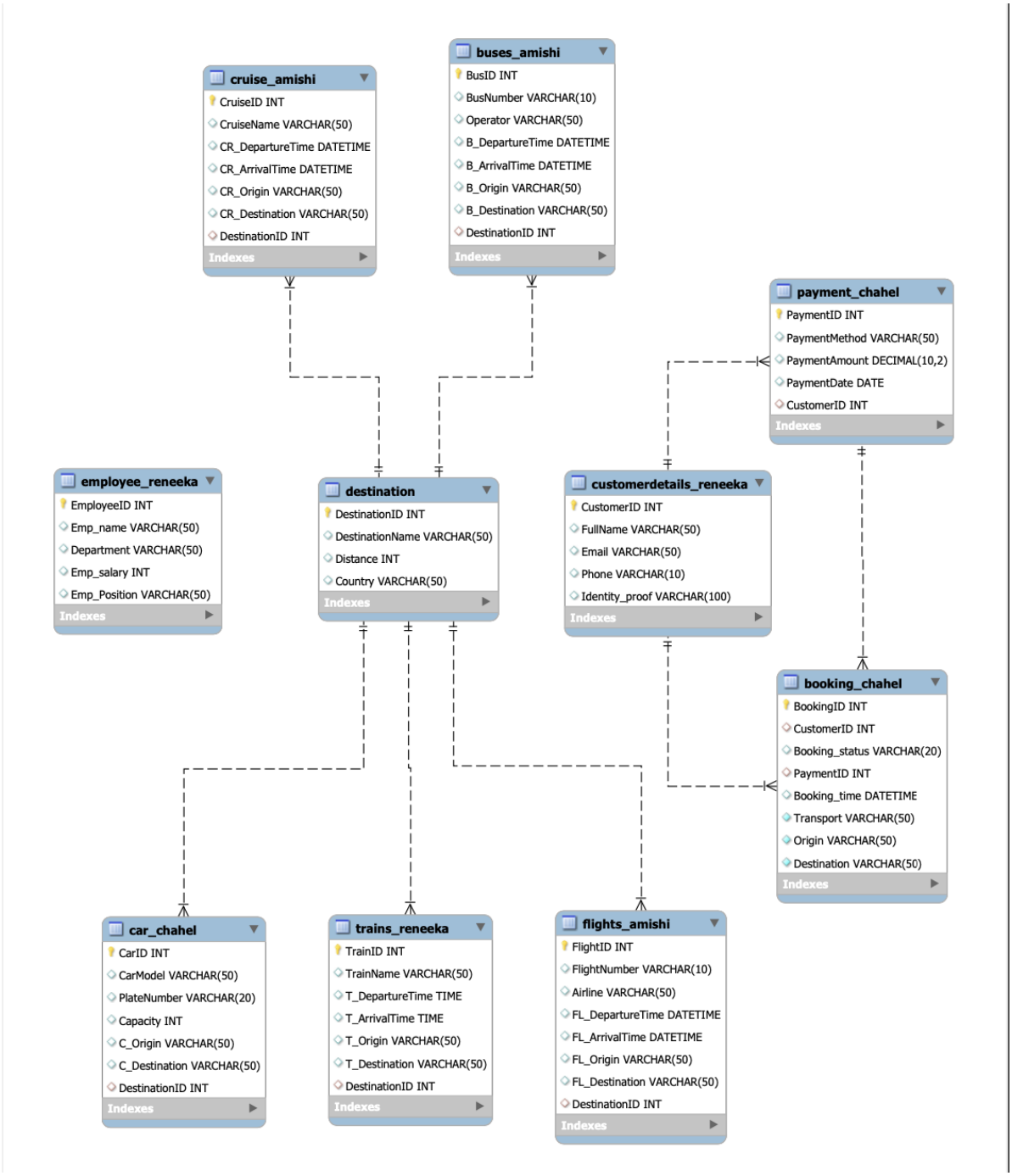
Cardinality and Participation:

- CustomerDetails_reneeka to Payment_chahel: Mandatory Participation on CustomerDetails_reneeka side, Optional Participation on Payment_chahel side. One-to-Many cardinality.
- Payment_chahel to Booking_chahel: Partial Participation on Payment_chahel side, Optional Participation on Booking_chahel side. One-to-Many cardinality.
- CustomerDetails_reneeka to Destination: Partial Participation on both sides. One-to-Many cardinality.
- Employee_reneeka to Trains_reneeka, Flights_amishi, Buses_amishi, Cruise_amishi, Car_chahel: Partial Participation on both sides. Many-to-Many cardinality.

III. Entity Relationship Diagram



IV. Relational Diagram



V. Normalization

BCNF (Boyce-Codd Normal Form):

- All tables are in 3NF, meeting the requirements for BCNF. There are no non-trivial functional dependencies on candidate keys, ensuring that each determinant is either a candidate key or a super key.

- We've meticulously designed our database to ensure that there are no non-trivial dependencies, minimizing the risk of data redundancy and anomalies.

3NF (Third Normal Form):

- Each table in our travel booking database is in 2NF, ensuring that there are no partial dependencies on the primary key. For example, in the CustomerDetails_reneeka table, attributes such as FullName, Email, Phone, and Identity_proof are directly dependent on the CustomerID primary key.

- We've eliminated transitive dependencies across all tables, ensuring that all non-prime attributes are directly dependent on the primary key. For instance, in the Booking_chahel table, Booking_status is directly dependent on BookingID, and Transport is directly dependent on the CustomerID.

2NF (Second Normal Form):

- Every table in our travel booking database is in 1NF, ensuring that each cell contains atomic values without any repeating groups. For instance, in the Destination table, each row contains atomic values for DestinationID, DestinationName, Distance, and Country, without any repeating groups.

- There are no partial dependencies across tables, and each non-prime attribute is fully functionally dependent on the entire primary key. For example, in the Flights_amishi table, FlightNumber, Airline, FL_DepartureTime, FL_ArrivalTime, FL_Origin, and FL_Destination are fully functionally dependent on the FlightID primary key.

1NF (First Normal Form):

- Our travel booking database ensures that each table maintains atomic values in every cell, adhering to the principles of 1NF. This means that there are no composite or multivalued attributes within any table.

- Repeating groups are strictly avoided, ensuring that each table represents a single entity with unique primary keys and no redundant data.

VI. SQL Queries

```
create database grp3_DBMSProject;

use grp3_DBMSProject;

CREATE TABLE IF NOT EXISTS CustomerDetails_reneeka (
   CustomerID INT PRIMARY KEY,
   FullName VARCHAR(50),
   Email VARCHAR(50),
   Phone VARCHAR(10),
   Identity_proof VARCHAR(100)
);

INSERT INTO CustomerDetails_reneeka (CustomerID, FullName, Email, Phone,
Identity_proof)
VALUES
   (1, 'Amishi Desai', 'desai014@gmail.com', '9345698768', 'Aadhar'),
   (2, 'Chahel Gupta', 'chx2904@gmail.com', '9876543210', 'Passport'),
   (3, 'Aneri Patel', 'ap2404@gmail.com', '7890123456', 'Driver License'),
   (4, 'Demira Ramnani', 'demram23@gmail.com', '2345678901', 'Aadhar'),
   (5, 'Shloka Arya', 'shloksrain@gmail.com', '8901234567', 'Passport'),
   (6, 'Vansh Mehta', 'vm567@gmail.com', '3456789012', 'Driver License'),
   (7, 'Shital Nagvenkar', 'shtlnad@gmail.com', '9012345678', 'Aadhar'),
   (8, 'Vikram Nadkarni', 'viks01@gmail.com', '4567890123', 'Passport'),
   (9, 'Aryan Shah', 'arshah56@gmail.com', '0123456789', 'Driver License'),
   (10, 'Veer Patil', 'vp3til@gmail.com', '9398631146', 'Aadhar');

CREATE TABLE IF NOT EXISTS Employee_reneeka (
   EmployeeID INT PRIMARY KEY,
   Emp_name VARCHAR(50),
   Department VARCHAR(50),
   Emp_salary INT(10),
   Emp_Position VARCHAR(50)
);

INSERT INTO Employee_reneeka (EmployeeID, Emp_name, Department, Emp_salary,
Emp_Position)
VALUES
   (1, 'Vivean Arya', 'Bus', 54000, 'Driver'),
   (2, 'Sonia Mascerehnas', 'Car', 55000, 'Guide'),
   (3, 'Vansh Dhoka', 'Train', 60000, 'Supervisor'),
   (4, 'Pratham Vasa', 'Flight', 65000, 'Booking Agent'),
   (5, 'Amit Singh', 'Cruise', 70000, 'Guide'),
   (6, 'Sunita Reddy', 'Bus', 50000, 'Driver'),
   (7, 'Rahul Verma', 'Car', 60000, 'Supervisor'),
   (8, 'Vivek Mishra', 'Flight', 70000, 'Guide'),
   (9, 'Anjali Shah', 'Train', 65000, 'Booking Agent'),
   (10, 'Pooja Gupta', 'Cruise', 75000, 'Supervisor');

CREATE TABLE IF NOT EXISTS Destination (
   DestinationID INT PRIMARY KEY,
```

```
    DestinationName VARCHAR(50),
    Distance INT,
    Country VARCHAR(50)
);

INSERT INTO Destination (DestinationID, DestinationName, Distance, Country)
VALUES
    (201, 'Mumbai', 0, 'India'),
    (202, 'Pune', 200, 'India'),
    (203, 'Chennai', 439, 'India'),
    (204, 'Bangalore', 5674, 'India'),
    (205, 'Hyderabad', 7656, 'India'),
    (206, 'Delhi', 987, 'India'),
    (207, 'Kolkata', 0, 'India'),
    (301, 'London', 12768, 'United Kingdom'),
    (302, 'Singapore', 5567, 'Singapore'),
    (303, 'Dubai', 767, 'United Arab Emirates'),
    (304, 'Frankfurt', 7898, 'Germany'),
    (305, 'Sydney', 9783, 'Australia'),
    (101, 'Ahmedabad', 0, 'India'),
    (102, 'Jaipur', 838, 'India'),
    (103, 'Indore', 595, 'India'),
    (104, 'Lucknow',895,'India');

CREATE TABLE IF NOT EXISTS Trains_reneeka (
    TrainID INT PRIMARY KEY,
    TrainName VARCHAR(50),
    T_DepartureTime TIME,
    T_ArrivalTime TIME,
    T_Origin VARCHAR(50),
    T_Destination VARCHAR(50),
    DestinationID INT,
    FOREIGN KEY (DestinationID) REFERENCES Destination(DestinationID)
);

INSERT INTO Trains_reneeka (TrainID, TrainName, T_DepartureTime, T_ArrivalTime,
T_Origin, T_Destination, DestinationID)
VALUES
    (1, 'Rajdhani Express', '08:00:00', '15:00:00', 'Delhi', 'Mumbai', 201),
    (2, 'Shatabdi Express', '09:00:00', '14:00:00', 'Mumbai', 'Pune', 202),
    (3, 'Duronto Express', '10:00:00', '17:00:00', 'Delhi', 'Chennai', 203),
    (4, 'Garib Rath Express', '11:00:00', '18:00:00', 'Kolkata', 'Bangalore', 204),
    (5, 'Shatabdi Express', '12:00:00', '19:00:00', 'Chennai', 'Hyderabad', 205),
    (6, 'Rajdhani Express', '13:00:00', '20:00:00', 'Bangalore', 'Delhi', 206),
    (7, 'Duronto Express', '14:00:00', '21:00:00', 'Mumbai', 'Kolkata', 207),
    (8, 'Garib Rath Express', '15:00:00', '22:00:00', 'Pune', 'Chennai', 203),
    (9, 'Shatabdi Express', '16:00:00', '23:00:00', 'Delhi', 'Hyderabad', 205),
    (10, 'Rajdhani Express', '17:00:00', '00:00:00', 'Kolkata', 'Mumbai', 201);

CREATE TABLE IF NOT EXISTS Flights_amishi (
    FlightID INT PRIMARY KEY,
    FlightNumber VARCHAR(10),
```

```
    Airline VARCHAR(50),
    FL_DepartureTime DATETIME,
    FL_ArrivalTime DATETIME,
    FL_Origin VARCHAR(50),
    FL_Destination VARCHAR(50),
    DestinationID INT,
    FOREIGN KEY (DestinationID) REFERENCES Destination(DestinationID)
);

INSERT INTO Flights_amishi (FlightID, FlightNumber, Airline, FL_DepartureTime,
FL_ArrivalTime, FL_Origin, FL_Destination, DestinationID)
VALUES
    (1, 'AI101', 'Air India', '2024-03-18 08:00:00', '2024-03-18 10:00:00', 'Delhi', 'Mumbai',
201),
    (2, 'AI102', 'Air India', '2024-03-18 10:00:00', '2024-03-18 12:00:00', 'Mumbai', 'Pune',
202),
    (3, 'AI103', 'Air India', '2024-03-18 12:00:00', '2024-03-18 15:00:00', 'Delhi', 'Chennai',
203),
    (4, 'AI104', 'Air India', '2024-03-18 14:00:00', '2024-03-18 17:00:00', 'Mumbai',
'Bangalore', 204),
    (5, 'BA101', 'British Airways', '2024-03-18 16:00:00', '2024-03-18 18:00:00', 'Delhi',
'London', 301),
    (6, 'SQ101', 'Singapore Airlines', '2024-03-18 18:00:00', '2024-03-18 20:00:00',
'Mumbai', 'Singapore', 302),
    (7, 'EK101', 'Emirates', '2024-03-18 20:00:00', '2024-03-19 02:00:00', 'Delhi', 'Dubai',
303),
    (8, 'LH101', 'Lufthansa', '2024-03-18 22:00:00', '2024-03-19 01:00:00', 'Mumbai',
'Frankfurt', 304),
    (9, 'QF101', 'Qantas', '2024-03-19 01:00:00', '2024-03-19 06:00:00', 'Delhi', 'Sydney',
305),
    (10, 'SQ104', 'Singapore Airlines', '2024-03-19 03:00:00', '2024-03-19 05:00:00',
'Mumbai', 'Singapore', 302);

CREATE TABLE IF NOT EXISTS Buses_amishi (
    BusID INT PRIMARY KEY,
    BusNumber VARCHAR(10),
    Operator VARCHAR(50),
    B_DepartureTime DATETIME,
    B_ArrivalTime DATETIME,
    B_Origin VARCHAR(50),
    B_Destination VARCHAR(50),
      DestinationID INT,
    FOREIGN KEY (DestinationID) REFERENCES Destination(DestinationID)
);

INSERT INTO Buses_amishi (BusID, BusNumber, Operator, B_DepartureTime,
B_ArrivalTime, B_Origin, B_Destination, DestinationID)
VALUES
    (1, 'B001', 'ABC Bus Services', '2024-03-18 08:00:00', '2024-03-18 10:00:00', 'Delhi',
'Mumbai', 201),
    (2, 'B002', 'XYZ Travels', '2024-03-18 10:00:00', '2024-03-18 12:00:00', 'Mumbai',
'Pune', 202),
```

```
    (3, 'B003', 'RST Tours', '2024-03-18 12:00:00', '2024-03-18 15:00:00', 'Delhi',
'Chennai', 203),
    (4, 'B004', 'PQR Roadways', '2024-03-18 14:00:00', '2024-03-18 17:00:00', 'Mumbai',
'Bangalore', 204),
    (5, 'B005', 'LMN Travels', '2024-03-18 16:00:00', '2024-03-18 18:00:00', 'Delhi',
'Hyderabad', 205),
    (6, 'B006', 'DEF Bus Lines', '2024-03-18 18:00:00', '2024-03-18 20:00:00', 'Mumbai',
'Kolkata', 206),
    (7, 'B007', 'GHI Tours', '2024-03-18 20:00:00', '2024-03-19 02:00:00', 'Delhi',
'Lucknow',104),
    (8, 'B008', 'JKL Bus Services', '2024-03-18 22:00:00', '2024-03-19 01:00:00', 'Mumbai',
'Ahmedabad', 101),
    (9, 'B009', 'MNO Roadways', '2024-03-19 01:00:00', '2024-03-19 06:00:00', 'Delhi',
'Jaipur', 102),
    (10, 'B010', 'PQR Travels', '2024-03-19 03:00:00', '2024-03-19 05:00:00', 'Mumbai',
'Indore', 103);

CREATE TABLE IF NOT EXISTS Cruise_amishi (
    CruiseID INT PRIMARY KEY,
    CruiseName VARCHAR(50),
    CR_DepartureTime DATETIME,
    CR_ArrivalTime DATETIME,
    CR_Origin VARCHAR(50),
    CR_Destination VARCHAR(50),
      DestinationID INT,
    FOREIGN KEY (DestinationID) REFERENCES Destination(DestinationID)
);

INSERT INTO Cruise_amishi (CruiseID, CruiseName, CR_DepartureTime,
CR_ArrivalTime, CR_Origin, CR_Destination, DestinationID)
VALUES
    (1, 'London to Sydney Cruise', '2024-03-18 08:00:00', '2024-03-18 20:00:00', 'London',
'Sydney', 305),
    (2, 'Singapore to Dubai Cruise', '2024-03-18 10:00:00', '2024-03-18 22:00:00',
'Singapore', 'Dubai', 303),
    (3, 'Frankfurt to Sydney Cruise', '2024-03-18 12:00:00', '2024-03-18 23:00:00',
'Frankfurt', 'Sydney', 305),
    (4, 'Dubai to London Cruise', '2024-03-18 14:00:00', '2024-03-18 23:00:00', 'Dubai',
'London', 301),
    (5, 'Sydney to Singapore Cruise', '2024-03-18 16:00:00', '2024-03-19 01:00:00',
'Sydney', 'Singapore', 302),
    (6, 'London to Dubai Cruise', '2024-03-18 18:00:00', '2024-03-19 04:00:00', 'London',
'Dubai', 303),
    (7, 'Singapore to Sydney Cruise', '2024-03-18 20:00:00', '2024-03-19 08:00:00',
'Singapore', 'Sydney', 305),
    (8, 'Frankfurt to Dubai Cruise', '2024-03-18 22:00:00', '2024-03-19 10:00:00',
'Frankfurt', 'Dubai', 303),
    (9, 'Dubai to Singapore Cruise', '2024-03-19 01:00:00', '2024-03-19 10:00:00', 'Dubai',
'Singapore', 302),
    (10, 'Sydney to London Cruise', '2024-03-19 03:00:00', '2024-03-19 16:00:00',
'Sydney', 'London', 301);
```

```
CREATE TABLE IF NOT EXISTS Car_chahel (
   CarID INT PRIMARY KEY,
   CarModel VARCHAR(50),
   PlateNumber VARCHAR(20),
   Capacity INT,
   C_Origin VARCHAR(50),
   C_Destination VARCHAR(50),
     DestinationID INT,
   FOREIGN KEY (DestinationID) REFERENCES Destination(DestinationID)
);

INSERT INTO Car_chahel (CarID, CarModel, PlateNumber, Capacity, C_Origin,
C_Destination, DestinationID)
VALUES
   (101, 'Toyota Innova', 'MH01AB1234', 7, 'Mumbai', 'Pune', 202),
   (102, 'Maruti Swift', 'MH02CD5678', 5, 'Pune', 'Mumbai', 201),
   (103, 'Honda City', 'MH03EF9012', 5, 'Chennai', 'Bangalore', 204),
   (104, 'Hyundai i20', 'MH04GH3456', 5, 'Bangalore', 'Hyderabad', 205),
   (105, 'Ford EcoSport', 'MH05IJ6789', 5, 'Hyderabad', 'Chennai', 203),
   (106, 'Volkswagen Polo', 'MH06KL0123', 5, 'Delhi', 'Kolkata', 207),
   (107, 'Toyota Corolla', 'MH07MN3456', 5, 'Kolkata', 'Delhi', 206),
   (108, 'Renault Kwid', 'MH08OP6789', 5, 'Ahmedabad', 'Jaipur', 102),
   (109, 'Honda Civic', 'MH09QR0123', 5, 'Jaipur', 'Indore', 103),
   (110, 'Mahindra Scorpio', 'MH10ST3456', 7, 'Indore', 'Ahmedabad', 101);


CREATE TABLE IF NOT EXISTS Payment_chahel (
   PaymentID INT PRIMARY KEY,
   PaymentMethod VARCHAR(50),
   PaymentAmount DECIMAL(10, 2),
   PaymentDate DATE,
   CustomerID INT,
   FOREIGN KEY (CustomerID) REFERENCES
CustomerDetails_reneeka(CustomerID)
);
INSERT INTO Payment_chahel (PaymentID, PaymentMethod, PaymentAmount,
PaymentDate, CustomerID)
VALUES
   (101, 'Credit Card', 5000.00, '2023-01-15', 1),
   (102, 'Debit Card', 3000.00, '2023-02-20', 2),
   (103, 'Net Banking', 2000.00, '2023-03-25', 3),
   (104, 'UPI', 4000.00, '2023-04-30', 4),
   (105, 'Cash', 2500.00, '2023-05-05', 5),
   (106, 'Credit Card', 3500.00, '2023-06-10', 6),
   (107, 'Debit Card', 2800.00, '2023-07-15', 7),
   (108, 'Net Banking', 1800.00, '2023-08-20', 8),
   (109, 'UPI', 4200.00, '2023-09-25', 9),
   (110, 'Cash', 2300.00, '2023-10-30', 10);


CREATE TABLE IF NOT EXISTS Booking_chahel (
   BookingID INT PRIMARY KEY,
```

```
    CustomerID INT,
    Booking_status VARCHAR(20),
    PaymentID INT,
    Booking_time DATETIME,
    Transport VARCHAR(50) NOT NULL,
    Origin VARCHAR(50) NOT NULL,
    Destination VARCHAR(50) NOT NULL,
    FOREIGN KEY (CustomerID) REFERENCES
CustomerDetails_reneeka(CustomerID),
    FOREIGN KEY (PaymentID) REFERENCES Payment_chahel(PaymentID)
);

INSERT INTO Booking_chahel (BookingID, CustomerID, Booking_status, PaymentID,
Booking_time, Transport, Origin, Destination)
VALUES
    (1, 1, 'Confirmed', 101, '2024-03-10 08:00:00', (SELECT Operator FROM
Buses_amishi WHERE DestinationID = 201), 'Delhi','Mumbai'),
    (2, 2, 'Confirmed', 102, '2024-03-10 10:00:00', (SELECT PlateNumber FROM
Car_chahel WHERE DestinationID = 202), 'Mumbai', 'Pune'),
    (3, 3, 'Confirmed', 103, '2024-03-10 12:00:00', (SELECT FlightNumber FROM
Flights_amishi WHERE DestinationID = 203), 'Delhi', 'Chennai'),
    (4, 4, 'Confirmed', 104, '2024-03-10 14:00:00', (SELECT FlightNumber FROM
Flights_amishi WHERE DestinationID = 204), 'Mumbai', 'Bangalore'),
    (5, 5, 'Confirmed', 105, '2024-03-10 16:00:00', (SELECT CruiseName FROM
Cruise_amishi WHERE DestinationID = 303 AND CR_Origin='Singapore'), 'Singapore',
'Dubai'),
    (6, 6, 'Confirmed', 106, '2024-03-10 18:00:00', (SELECT Operator FROM
Buses_amishi WHERE DestinationID = 206), 'Mumbai', 'Kolkata'),
    (7, 7, 'Confirmed', 107, '2024-03-10 20:00:00', (SELECT PlateNumber FROM
Car_chahel WHERE DestinationID = 103), 'Jaipur', 'Indore' ),
    (8, 8, 'Confirmed', 108, '2024-03-10 22:00:00', (SELECT TrainName FROM
Trains_reneeka WHERE DestinationID = 204), 'Kolkata', 'Bangalore'),
    (9, 9, 'Confirmed', 109, '2024-03-11 01:00:00', (SELECT FlightNumber FROM
Flights_amishi WHERE DestinationID = 203), 'Delhi', 'Chennai'),
    (10, 10, 'Confirmed', 110, '2024-03-11 03:00:00', (SELECT CruiseName FROM
Cruise_amishi WHERE DestinationID = 301 AND CR_Origin='Sydney'), 'Sydney',
'London');
```

VII. Project Demonstration

Tables

Result Grid Filter Rows: Search Edit:

CustomerID	FullName	Email	Phone	Identity_proof
1	Amishi Desai	desai014@gmail.com	9345698768	Aadhar
2	Chahel Gupta	chx2904@gmail.com	9876543210	Passport
3	Aneri Patel	ap2404@gmail.com	7890123456	Driver License
4	Demira Ramnani	demram23@gmail.com	2345678901	Aadhar
5	Shloka Arya	shloksrain@gmail.com	8901234567	Passport
6	Vansh Mehta	vm567@gmail.com	3456789012	Driver License
7	Shital Nagvenkar	shtlnad@gmail.com	9012345678	Aadhar
8	Vikram Nadkarni	viks01@gmail.com	4567890123	Passport
9	Aryan Shah	arshah56@gmail.com	0123456789	Driver License
10	Veer Patil	vp3til@gmail.com	9398631146	Aadhar
NULL	NULL	NULL	NULL	NULL

CustomerDetails_reneeka 55

EmployeeID	Emp_name	Department	Emp_salary	Emp_Position
1	Vivean Arya	Bus	54000	Driver
2	Sonia Mascerehnas	Car	55000	Guide
3	Vansh Dhoka	Train	60000	Supervisor
4	Pratham Vasa	Flight	65000	Booking Agent
5	Amit Singh	Cruise	70000	Guide
6	Sunita Reddy	Bus	50000	Driver
7	Rahul Verma	Car	60000	Supervisor
8	Vivek Mishra	Flight	70000	Guide
9	Anjali Shah	Train	65000	Booking Agent
10	Pooja Gupta	Cruise	75000	Supervisor
NULL	NULL	NULL	NULL	NULL

mployee_reneeka 56

TrainID	TrainName	T_DepartureTime	T_ArrivalTime	T_Origin	T_Destination	DestinationID
1	Rajdhani Express	08:00:00	15:00:00	Delhi	Mumbai	201
2	Shatabdi Express	09:00:00	14:00:00	Mumbai	Pune	202
3	Duronto Express	10:00:00	17:00:00	Delhi	Chennai	203
4	Garib Rath Express	11:00:00	18:00:00	Kolkata	Bangalore	204
5	Shatabdi Express	12:00:00	19:00:00	Chennai	Hyderabad	205
6	Rajdhani Express	13:00:00	20:00:00	Bangalore	Delhi	206
7	Duronto Express	14:00:00	21:00:00	Mumbai	Kolkata	207
8	Garib Rath Express	15:00:00	22:00:00	Pune	Chennai	203
9	Shatabdi Express	16:00:00	23:00:00	Delhi	Hyderabad	205
10	Rajdhani Express	17:00:00	00:00:00	Kolkata	Mumbai	201
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Trains_reneeka 58

FlightID	FlightNumber	Airline	FL_DepartureTime	FL_ArrivalTime	FL_Origin	FL_Destination	DestinationID
1	AI101	Air India	2024-03-18 08:00:00	2024-03-18 10:00:00	Delhi	Mumbai	201
2	AI102	Air India	2024-03-18 10:00:00	2024-03-18 12:00:00	Mumbai	Pune	202
3	AI103	Air India	2024-03-18 12:00:00	2024-03-18 15:00:00	Delhi	Chennai	203
4	AI104	Air India	2024-03-18 14:00:00	2024-03-18 17:00:00	Mumbai	Bangalore	204
5	BA101	British Airways	2024-03-18 16:00:00	2024-03-18 18:00:00	Delhi	London	301
6	SQ101	Singapore Airlines	2024-03-18 18:00:00	2024-03-18 20:00:00	Mumbai	Singapore	302
7	EK101	Emirates	2024-03-18 20:00:00	2024-03-19 02:00:00	Delhi	Dubai	303
8	LH101	Lufthansa	2024-03-18 22:00:00	2024-03-19 01:00:00	Mumbai	Frankfurt	304
9	QF101	Qantas	2024-03-19 01:00:00	2024-03-19 06:00:00	Delhi	Sydney	305
10	SQ104	Singapore Airlines	2024-03-19 03:00:00	2024-03-19 05:00:00	Mumbai	Singapore	302
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Flights_amishi 59

BusID	BusNumber	Operator	B_DepartureTime	B_ArrivalTime	B_Origin	B_Destination	DestinationID
1	B001	ABC Bus Services	2024-03-18 08:00:00	2024-03-18 10:00:00	Delhi	Mumbai	201
2	B002	XYZ Travels	2024-03-18 10:00:00	2024-03-18 12:00:00	Mumbai	Pune	202
3	B003	RST Tours	2024-03-18 12:00:00	2024-03-18 15:00:00	Delhi	Chennai	203
4	B004	PQR Roadways	2024-03-18 14:00:00	2024-03-18 17:00:00	Mumbai	Bangalore	204
5	B005	LMN Travels	2024-03-18 16:00:00	2024-03-18 18:00:00	Delhi	Hyderabad	205
6	B006	DEF Bus Lines	2024-03-18 18:00:00	2024-03-18 20:00:00	Mumbai	Kolkata	206
7	B007	GHI Tours	2024-03-18 20:00:00	2024-03-19 02:00:00	Delhi	Lucknow	104
8	B008	JKL Bus Services	2024-03-18 22:00:00	2024-03-19 01:00:00	Mumbai	Ahmedabad	101
9	B009	MNO Roadways	2024-03-19 01:00:00	2024-03-19 06:00:00	Delhi	Jaipur	102
10	B010	PQR Travels	2024-03-19 03:00:00	2024-03-19 05:00:00	Mumbai	Indore	103
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Buses_amishi 60

CruiseID	CruiseName	CR_DepartureTime	CR_ArrivalTime	CR_Origin	CR_Destination	DestinationID
1	London to Sydney Cruise	2024-03-18 08:00:00	2024-03-18 20:00:00	London	Sydney	305
2	Singapore to Dubai Cruise	2024-03-18 10:00:00	2024-03-18 22:00:00	Singapore	Dubai	303
3	Frankfurt to Sydney Cruise	2024-03-18 12:00:00	2024-03-18 23:00:00	Frankfurt	Sydney	305
4	Dubai to London Cruise	2024-03-18 14:00:00	2024-03-18 23:00:00	Dubai	London	301
5	Sydney to Singapore Cruise	2024-03-18 16:00:00	2024-03-19 01:00:00	Sydney	Singapore	302
6	London to Dubai Cruise	2024-03-18 18:00:00	2024-03-19 04:00:00	London	Dubai	303
7	Singapore to Sydney Cruise	2024-03-18 20:00:00	2024-03-19 08:00:00	Singapore	Sydney	305
8	Frankfurt to Dubai Cruise	2024-03-18 22:00:00	2024-03-19 10:00:00	Frankfurt	Dubai	303
9	Dubai to Singapore Cruise	2024-03-19 01:00:00	2024-03-19 10:00:00	Dubai	Singapore	302
10	Sydney to London Cruise	2024-03-19 03:00:00	2024-03-19 16:00:00	Sydney	London	301
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Cruise_amishi 61

PaymentID	PaymentMethod	PaymentAmount	PaymentDate	CustomerID
101	Credit Card	5000.00	2023-01-15	1
102	Debit Card	3000.00	2023-02-20	2
103	Net Banking	2000.00	2023-03-25	3
104	UPI	4000.00	2023-04-30	4
105	Cash	2500.00	2023-05-05	5
106	Credit Card	3500.00	2023-06-10	6
107	Debit Card	2800.00	2023-07-15	7
108	Net Banking	1800.00	2023-08-20	8
109	UPI	4200.00	2023-09-25	9
110	Cash	2300.00	2023-10-30	10
NULL	NULL	NULL	NULL	NULL

Payment_chahel 63

CarID	CarModel	PlateNumber	Capacity	C_Origin	C_Destination	DestinationID
101	Toyota Innova	MH01AB1234	7	Mumbai	Pune	202
102	Maruti Swift	MH02CD5678	5	Pune	Mumbai	201
103	Honda City	MH03EF9012	5	Chennai	Bangalore	204
104	Hyundai i20	MH04GH3456	5	Bangalore	Hyderabad	205
105	Ford EcoSport	MH05IJ6789	5	Hyderabad	Chennai	203
106	Volkswagen Polo	MH06KL0123	5	Delhi	Kolkata	207
107	Toyota Corolla	MH07MN3456	5	Kolkata	Delhi	206
108	Renault Kwid	MH08OP6789	5	Ahmedabad	Jaipur	102
109	Honda Civic	MH09QR0123	5	Jaipur	Indore	103
110	Mahindra Scorpio	MH10ST3456	7	Indore	Ahmedabad	101
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Car_chahel 62

DestinationID	DestinationName	Distance	Country
101	Ahmedabad	0	India
102	Jaipur	838	India
103	Indore	595	India
104	Lucknow	895	India
201	Mumbai	0	India
202	Pune	200	India
203	Chennai	439	India
204	Bangalore	5674	India
205	Hyderabad	7656	India
206	Delhi	987	India
207	Kolkata	0	India
301	London	12768	United

Destination 65

BookingID	CustomerID	Booking_stat...	PaymentID	Booking_time	Transport	Origin	Destination
1	1	Confirmed	101	2024-03-10 08:00:00	ABC Bus Services	Delhi	Mumbai
2	2	Confirmed	102	2024-03-10 10:00:00	MH01AB1234	Mumbai	Pune
3	3	Confirmed	103	2024-03-10 12:00:00	AI103	Delhi	Chennai
4	4	Confirmed	104	2024-03-10 14:00:00	AI104	Mumbai	Bangalore
5	5	Confirmed	105	2024-03-10 16:00:00	Singapore to Dubai Cruise	Singapore	Dubai
6	6	Confirmed	106	2024-03-10 18:00:00	DEF Bus Lines	Mumbai	Kolkata
7	7	Confirmed	107	2024-03-10 20:00:00	MH09QR0123	Jaipur	Indore
8	8	Confirmed	108	2024-03-10 22:00:00	Garib Rath Express	Kolkata	Bangalore
9	9	Confirmed	109	2024-03-11 01:00:00	AI103	Delhi	Chennai
10	10	Confirmed	110	2024-03-11 03:00:00	Sydney to London Cruise	Sydney	London
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Booking_chahel 67

SQL Select Queries:

1. **Update employee ID 2's position to Driver**

   UPDATE Employee_reneeka SET Emp_Position = 'Driver' WHERE EmployeeID = 2;

Result Grid | Filter Rows: Search | Edit:

EmployeeID	Emp_name	Department	Emp_salary	Emp_Position
1	Vivean Arya	Bus	54000	Driver
2	Sonia Mascerehnas	Car	55000	Driver
3	Vansh Dhoka	Train	60000	Supervisor
4	Pratham Vasa	Flight	65000	Booking Agent
5	Amit Singh	Cruise	70000	Guide
6	Sunita Reddy	Bus	50000	Driver
7	Rahul Verma	Car	60000	Supervisor
8	Vivek Mishra	Flight	70000	Guide
9	Anjali Shah	Train	65000	Booking Agent
10	Pooja Gupta	Cruise	75000	Supervisor
NULL	NULL	NULL	NULL	NULL

2. **Write a query to create new table CustomerTravelDoc with columns CustomerID, Full Name and Identity_proof selected from table CustomerDetails_reneeka.**

   CREATE TABLE CustomerTravelDoc AS SELECT CustomerID, FullName, Identity_proof FROM CustomerDetails_reneeka;

CustomerID	FullName	Identity_proof
1	Amishi Desai	Aadhar
2	Chahel Gupta	Passport
3	Aneri Patel	Driver License
4	Demira Ramnani	Aadhar
5	Shloka Arya	Passport
6	Vansh Mehta	Driver License
7	Shital Nagvenkar	Aadhar
8	Vikram Nadkarni	Passport
9	Aryan Shah	Driver License
10	Veer Patil	Aadhar

CustomerTravelDoc 13

3. **Rename CustomerTravelDoc to Customer_documents**

   ALTER TABLE CustomerTravelDoc RENAME TO Customer_documents;
   SELECT * FROM Customer_documents;

Result Grid | Filter Rows: Search

CustomerID	FullName	Identity_proof
1	Amishi Desai	Aadhar
2	Chahel Gupta	Passport
3	Aneri Patel	Driver License
4	Demira Ramnani	Aadhar
5	Shloka Arya	Passport
6	Vansh Mehta	Driver License
7	Shital Nagvenkar	Aadhar
8	Vikram Nadkarni	Passport
9	Aryan Shah	Driver License
10	Veer Patil	Aadhar

Customer_documents 14

4. **Add check constraint such that email of customers should end with @gmail.com**

   ALTER TABLE CustomerDetails_reneeka ADD CONSTRAINT CK_email_format CHECK (Email LIKE '%@gmail.com');

5. **Set default value of booking status to pending**

   ALTER TABLE Booking_chahel MODIFY COLUMN Booking_status VARCHAR(20) NOT NULL DEFAULT 'Pending';

6. **Add booking with following details; BookingID: 11, CustomerID: 7, PaymentID: 111, Booking_time: 11th March 2024 1am, Transport: AI103, Origin: Delhi, Destination:Chennai**

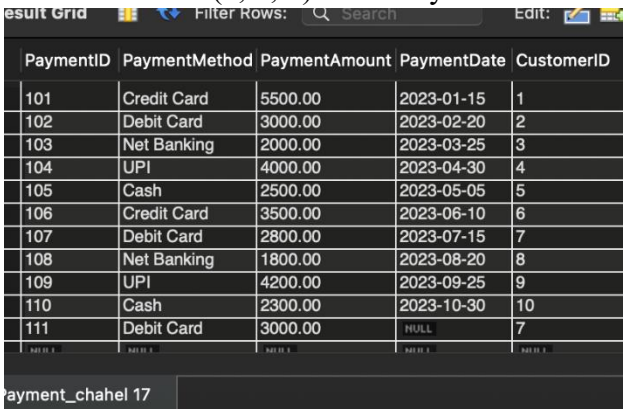
   INSERT INTO Booking_chahel (BookingID, CustomerID, PaymentID, Booking_time, Transport, Origin, Destination) VALUES (11, 7, 111, '2024-03-11 01:00:00', 'AI103', 'Delhi', 'Chennai');

BookingID	CustomerID	Booking_stat...	PaymentID	Booking_time	Transport	Origin	Destination
1	1	Confirmed	101	2024-03-10 08:00:00	ABC Bus Services	Delhi	Mumbai
2	2	Confirmed	102	2024-03-10 10:00:00	MH01AB1234	Mumbai	Pune
3	3	Confirmed	103	2024-03-10 12:00:00	AI103	Delhi	Chennai
4	4	Confirmed	104	2024-03-10 14:00:00	AI104	Mumbai	Bangalore
5	5	Confirmed	105	2024-03-10 16:00:00	Singapore to Dubai Cruise	Singapore	Dubai
6	6	Confirmed	106	2024-03-10 18:00:00	DEF Bus Lines	Mumbai	Kolkata
7	7	Confirmed	107	2024-03-10 20:00:00	MH09QR0123	Jaipur	Indore
8	8	Confirmed	108	2024-03-10 22:00:00	Garib Rath Express	Kolkata	Bangalore
9	9	Confirmed	109	2024-03-11 01:00:00	AI103	Delhi	Chennai
10	10	Confirmed	110	2024-03-11 03:00:00	Sydney to London Cruise	Sydney	London
11	7	Pending	111	2024-03-11 01:00:00	AI103	Delhi	Chennai

Booking_chahel 16

7. **Update the payment amounts for customers who have made payments using a credit card by increasing their payment amount by 10%. Include only those customers whose IDs are in the list (1, 3, 5).**

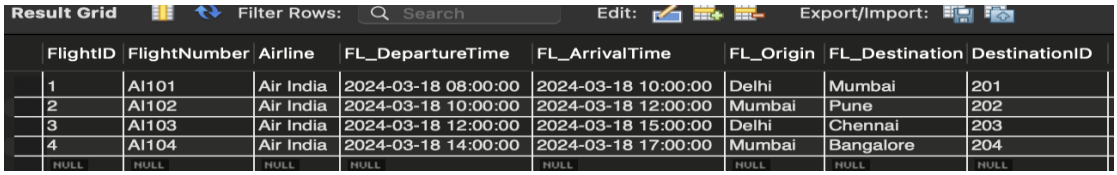
UPDATE Payment_chahel SET PaymentAmount = PaymentAmount * 1.1 WHERE CustomerID IN (1, 3, 5) AND PaymentMethod = 'Credit Card';



PaymentID	PaymentMethod	PaymentAmount	PaymentDate	CustomerID
101	Credit Card	5500.00	2023-01-15	1
102	Debit Card	3000.00	2023-02-20	2
103	Net Banking	2000.00	2023-03-25	3
104	UPI	4000.00	2023-04-30	4
105	Cash	2500.00	2023-05-05	5
106	Credit Card	3500.00	2023-06-10	6
107	Debit Card	2800.00	2023-07-15	7
108	Net Banking	1800.00	2023-08-20	8
109	UPI	4200.00	2023-09-25	9
110	Cash	2300.00	2023-10-30	10
111	Debit Card	3000.00	NULL	7

8. Display flight details of airline that starts with an A and ends with an A

SELECT * FROM Flights_amishi WHERE Airline LIKE 'A%A'



FlightID	FlightNumber	Airline	FL_DepartureTime	FL_ArrivalTime	FL_Origin	FL_Destination	DestinationID
1	AI101	Air India	2024-03-18 08:00:00	2024-03-18 10:00:00	Delhi	Mumbai	201
2	AI102	Air India	2024-03-18 10:00:00	2024-03-18 12:00:00	Mumbai	Pune	202
3	AI103	Air India	2024-03-18 12:00:00	2024-03-18 15:00:00	Delhi	Chennai	203
4	AI104	Air India	2024-03-18 14:00:00	2024-03-18 17:00:00	Mumbai	Bangalore	204

9. Update distance to 500 where distance = 0 and display destinations in india with distance between 500 and 1500.

SET SQL_SAFE_UPDATES = 0; UPDATE Destination SET Distance = 500 WHERE Distance = 0; SET SQL_SAFE_UPDATES = 1; SELECT * FROM Destination WHERE Country = 'India' AND Distance BETWEEN 500 AND 1500;

DestinationID	DestinationName	Distance	Country
101	Ahmedabad	500	India
102	Jaipur	838	India
103	Indore	595	India
104	Lucknow	895	India
201	Mumbai	500	India
206	Delhi	987	India
207	Kolkata	500	India

10. Display each destination country.

SELECT DISTINCT Country FROM Destination;

Country
India
United Kingdom
Singapore
United Arab Emirates
Germany
Australia

11. Describe structure of booking_chahel table.

DESC Booking_chahel;

Field	Type	Null	Key	Default	Extra
BookingID	int	NO	PRI	NULL	
CustomerID	int	YES	MUL	NULL	
Booking_status	varchar(20)	NO		Pending	
PaymentID	int	YES	MUL	NULL	
Booking_time	datetime	YES		NULL	
Transport	varchar(50)	NO		NULL	
Origin	varchar(50)	NO		NULL	
Destination	varchar(50)	NO		NULL	

12. Find out details of those bookings for which origin of one booking is same as destination of another booking.

SELECT b1.BookingID AS BookingID1, b1.CustomerID AS CustomerID1, b1.Booking_status AS BookingStatus1, b2.BookingID AS BookingID2, b2.CustomerID AS CustomerID2, b2.Booking_status AS BookingStatus2 FROM Booking_chahel b1, Booking_chahel b2 WHERE b1.Origin = b2.Destination;

BookingID1	CustomerID1	BookingStatus1	BookingID2	CustomerID2	BookingStatus2
6	6	Confirmed	1	1	Confirmed
4	4	Confirmed	1	1	Confirmed
2	2	Confirmed	1	1	Confirmed
8	8	Confirmed	6	6	Confirmed

13. Display details of various modes of transport available to go to Singapore.

SELECT DISTINCT d.DestinationName, t.TrainName AS Train, f.FlightNumber AS Flight, b.BusNumber AS Bus, c.CruiseName AS Cruise FROM Destination d
LEFT JOIN Trains_reneeka t ON d.DestinationID = t.DestinationID
LEFT JOIN Flights_amishi f ON d.DestinationID = f.DestinationID
LEFT JOIN Buses_amishi b ON d.DestinationID = b.DestinationID
LEFT JOIN Cruise_amishi c ON d.DestinationID = c.DestinationID
WHERE d.DestinationName = ‘Singapore';

DestinationName	Train	Flight	Bus	Cruise
Singapore	NULL	SQ101	NULL	Sydney to Singapore Cruise
Singapore	NULL	SQ101	NULL	Dubai to Singapore Cruise
Singapore	NULL	SQ104	NULL	Sydney to Singapore Cruise
Singapore	NULL	SQ104	NULL	Dubai to Singapore Cruise

14. Display booking and customer details of customers who paid via credit card.

SELECT B.BookingID, B.CustomerID, B.Booking_status,B.PaymentID, B.Booking_time, B.Transport, B.Origin, B.Destination, C.FullName,C.Email, C.Phone, C.Identity_proof FROM Booking_chahel AS B JOIN CustomerDetails_reneeka AS C ON B.CustomerID = C.CustomerID JOIN

Payment_chahel AS P ON B.PaymentID = P.PaymentID WHERE P.PaymentMethod = 'Credit Card';

BookingID	CustomerID	Booking_stat...	PaymentID	Booking_time	Transport	Origin	Destination	FullName	Email	Phone	Identity_proof
1	1	Confirmed	101	2024-03-10 08:00:00	ABC Bus Services	Delhi	Mumbai	Amishi Desai	desai014@gmail.com	9345698768	Aadhar
6	6	Confirmed	106	2024-03-10 18:00:00	DEF Bus Lines	Mumbai	Kolkata	Vansh Mehta	vm567@gmail.com	3456789012	Driver License

15. Display payment and customer details in order of most expensive trip booked to cheapest trip.

SELECT C.CustomerID, C.FullName, C.Email, C.Phone, C.Identity_proof, P.PaymentAmount FROM Payment_chahel AS P JOIN CustomerDetails_reneeka AS C ON P.CustomerID = C.CustomerID ORDER BY P.PaymentAmount DESC;

CustomerID	FullName	Email	Phone	Identity_proof	PaymentAmount
1	Amishi Desai	desai014@gmail.com	9345698768	Aadhar	5500.00
9	Aryan Shah	arshah56@gmail.com	0123456789	Driver License	4200.00
4	Demira Ramnani	demram23@gmail.com	2345678901	Aadhar	4000.00
6	Vansh Mehta	vm567@gmail.com	3456789012	Driver License	3500.00
2	Chahel Gupta	chx2904@gmail.com	9876543210	Passport	3000.00
7	Shital Nagvenkar	shtlnad@gmail.com	9012345678	Aadhar	3000.00
7	Shital Nagvenkar	shtlnad@gmail.com	9012345678	Aadhar	2800.00
5	Shloka Arya	shloksrain@gmail.com	8901234567	Passport	2500.00
10	Veer Patil	vp3til@gmail.com	9398631146	Aadhar	2300.00
3	Aneri Patel	ap2404@gmail.com	7890123456	Driver License	2000.00
8	Vikram Nadkarni	viks01@gmail.com	4567890123	Passport	1800.00

16. How many flights fly in and out of Mumbai?

SELECT COUNT(*) AS Total_Mumbai_flights FROM Flights_amishi WHERE FL_Origin = 'Mumbai' OR FL_Destination = ‘Mumbai';

Total_Mumbai_flights
6

17. Retrieve maximum and minimum salary of employee for each department.

SELECT Department, MAX(Emp_salary) AS Max_EmpSalary, MIN(Emp_salary) AS Min_EmpSalary FROM Employee_reneeka GROUP BY Department;

Department	Max_EmpSalary	Min_EmpSalary
Bus	54000	50000
Car	60000	55000
Train	65000	60000
Flight	70000	65000
Cruise	75000	70000

18. Find destinations that have both train and flight services. Also, find destinations that have bus services but not car services.

SELECT T_Destination AS Destination FROM Trains_reneeka INTERSECT SELECT FL_Destination AS Destination FROM Flights_amishi;

Destination
Mumbai
Pune
Chennai
Bangalore

SELECT B_Destination FROM Buses_amishi WHERE B_Destination NOT IN (SELECT C_Destination FROM Car_chahel);

B_Destination
Lucknow

19. Find the total number of bookings made by each customer and display only those customers who made 2 or more bookings.

SELECT CustomerID, COUNT(*) AS CustomerBookings FROM Booking_chahel GROUP BY CustomerID HAVING COUNT(*) >= 2;

CustomerID	CustomerBookings
7	2

20. Retrieve income incurred by Travel Booking.

SELECT SUM(PaymentAmount) AS Total_Income FROM Payment_chahel;

Total_Income
34600.00

21. Create a view that provides a summary of booking statistics, such as the total number of bookings, the total amount paid, and the average booking amount.

CREATE VIEW Booking_Summary AS SELECT COUNT(*) AS TotalBookings,SUM(p.PaymentAmount) AS TotalAmountPaid, AVG(p.PaymentAmount) AS AverageAmount FROM Booking_chahel b JOIN Payment_chahel p ON b.PaymentID = p.PaymentID; SELECT * FROM Booking_Summary;

TotalBookings	TotalAmountPaid	AverageAmount
11	34600.00	3145.454545

22. Create a view showing the number of bookings made for each destination.

CREATE VIEW DestinationS_Booked AS SELECT Destination, COUNT(*) AS BookingCount FROM Booking_chahel GROUP BY Destination;

Destination	BookingCount
Mumbai	1
Pune	1
Chennai	3
Bangalore	2
Dubai	1
Kolkata	1
Indore	1
London	1

23. Are there any destinations for which no bookings have been made by any customers?

SELECT Destination.DestinationID, Destination.DestinationName FROM Destination LEFT JOIN Booking_chahel ON Destination.DestinationID = Booking_chahel.Destination WHERE Booking_chahel.Destination IS NULL;

DestinationID	DestinationName
101	Ahmedabad
102	Jaipur
103	Indore
104	Lucknow
201	Mumbai
202	Pune
203	Chennai
204	Bangalore
205	Hyderabad
206	Delhi
207	Kolkata
301	London

24. Delete booking where transport is a TrainName and then display booking table starting with null values.

DELETE FROM Booking_chahel WHERE Transport IN (SELECT TrainName FROM Trains_reneeka);

25. Which mode of transport is most frequently used for bookings?

SELECT Transport, COUNT(*) AS BookingCount FROM Booking_chahel GROUP BY Transport ORDER BY COUNT(*) DESC LIMIT 1;

Transport	BookingCount
AI103	3

VIII. Self-Learning Beyond Classroom

Self-learning involves taking the initiative to explore topics independently, utilizing a range of resources such as online courses, documentation, tutorials, and practical projects. For instance, in the context of database management and software development, self-learning can involve experimenting with complex queries like this one

```
SELECT d.DestinationName,
   GROUP_CONCAT(DISTINCT t.TrainName) AS Train,
   GROUP_CONCAT(DISTINCT f.FlightNumber) AS Flight,
   GROUP_CONCAT(DISTINCT b.BusNumber) AS Bus,
   GROUP_CONCAT(DISTINCT c.CruiseName) AS Cruise
FROM
   Destination d
LEFT JOIN
   Trains_reneeka t ON d.DestinationID = t.DestinationID
LEFT JOIN
   Flights_amishi f ON d.DestinationID = f.DestinationID
LEFT JOIN
   Buses_amishi b ON d.DestinationID = b.DestinationID
LEFT JOIN
   Cruise_amishi c ON d.DestinationID = c.DestinationID
WHERE
   d.DestinationName = 'Singapore'
GROUP BY
   d.DestinationName;
```

In this query, the use of aggregate function `GROUP_CONCAT` alongside `LEFT JOIN` demonstrates a deeper understanding of SQL beyond basic querying. We needed to remove duplicate values from result, while we had understanding relying solely on traditional JOIN operations.

Through self-learning, we explored alternative techniques like using aggregate functions to address such challenges. By working on practical projects or real-world scenarios, we are sure we will gain valuable hands-on experience that reinforces theoretical knowledge. In this case, the query addresses modes of transport available to reach Singapore, illustrating the practical application of SQL in a relevant context.
In conclusion, self-learning empowers us to overcome challenges, excel in chosen fields, unlock new opportunities and achieve goals.

IX. Learning From The Project

This project has been instrumental in deepening our understanding of database management systems (DBMS) and applying theoretical concepts into practical scenarios. Through the development of the travel booking management system, we gained invaluable insights into various aspects of database design, normalization, and query optimization. Here's how this project has helped us:

- By actively engaging in the design and implementation of the travel booking database, we gained hands-on experience in creating normalized database schemas, defining relationships between entities, and ensuring data integrity through constraints.

- Working through the project challenges provided a platform to apply theoretical knowledge acquired during DBMS lectures and labs into real-world scenarios.

- The project served as a culmination of our learning in DBMS labs, where we familiarized ourselves with SQL queries, normalization techniques, and database design principles.

- Through iterative refinement of the database schema and query optimization, we reinforced our understanding of normalization forms, ensuring that the database adheres to 3NF and BCNF standards.

- Hands-on experience in debugging errors, resolving data inconsistencies, and optimizing query performance provided practical insights that complemented theoretical knowledge gained in labs.

- The travel booking management system offered a diverse range of scenarios that enabled us to explore various aspects of database design, including handling different modes of transportation, managing customer details, and integrating payment systems.

- Dealing with the complexity of travel bookings, including flights, trains, buses, and cruises, broadened our perspective on database design and management, challenging us to devise efficient schemas capable of handling diverse data requirements.

X. Challenges Faced

During the development of the travel booking system project, we encountered several challenges that tested our problem-solving abilities and teamwork. Some include:

- Database Design Complexity: Designing a database schema that effectively models the various entities and their relationships in the travel booking domain was a significant challenge. We needed to ensure that the schema was scalable, efficient, and capable of accommodating future modifications.
- Query Optimization: Crafting efficient SQL queries to retrieve relevant information from the database was another challenge. We needed to optimize queries to minimize response times and ensure smooth performance, especially when dealing with large datasets.
- User Interface Design: Designing an intuitive and user-friendly interface for the travel booking system was essential but challenging. We needed to consider factors such as usability, accessibility, and responsiveness for a seamless booking experience for users.
- Data Integration: Integrating data from multiple sources, such as customer details, transportation options (trains, flights, buses, cruises), destinations, and payments, posed a challenge. Ensuring data consistency and accuracy across different tables required careful planning and execution.
- Real life business logic: We had to ensure that the system accurately reflected the business rules and requirements while maintaining data integrity.
- Testing Quality: Thorough testing and quality assurance were crucial to identify and resolve bugs, errors, and inconsistencies in the system.

Despite these challenges, our team effectively collaborated, communicated, and problem-solved to overcome obstacles and deliver a simple travel booking system that met the project objectives and requirements.

XI. Conclusion

The experience of building a travel booking system allowed us to gain insights into the complexities of the travel industry, including the dynamic nature of bookings, the importance of real-time data updates, and the need for seamless customer experiences. Understanding these intricacies broadened our perspective and equipped us with valuable knowledge that can be applied to various domains beyond travel, such as e-commerce, hospitality, and logistics.

The project's scope exposed us to real-world challenges faced in the travel industry, such as managing bookings across multiple destinations and ensuring integration between different modes of transportation. This practical exposure was instrumental in enhancing our problem-solving skills and preparing us for future endeavors in database management and software development. It also highlighted the importance of collaboration and effective communication within a team. Working together to brainstorm solutions, address issues, and make decisions collectively, improved our project outcomes.

Overall, the project served as a valuable learning experience, bridging the gap between theoretical knowledge gained in classrooms and practical application in real-world scenarios. It provided us with a solid foundation in database management, software development methodologies, and problem-solving techniques.