

Name: Shubham Dinde

## SQL Project

### -- Objectives:

The primary objective of this project is to design and implement a comprehensive relational database system for a train scheduling and booking application. This system aims to efficiently manage data related to trains, stations, schedules, passengers, and bookings, facilitating smooth operations and decision-making processes within a transportation service provider.

### -- Database

-- creating a database

"Railway\_Management\_Sytem"

CREATE DATABASE

Railway\_Management\_Sytem;

USE Railway\_Management\_Sytem;

### -- Creating the tables and inserting the values

```
CREATE TABLE Trains (  
    TrainID INT PRIMARY KEY,  
    TrainName VARCHAR(100) NOT NULL,  
    TrainType VARCHAR(50) NOT NULL,  
    Capacity INT NOT NULL,  
    SourceStationID INT,  
    DestinationStationID INT,  
    FOREIGN KEY (SourceStationID) REFERENCES Stations(StationID),  
    FOREIGN KEY (DestinationStationID) REFERENCES Stations(StationID)  
);
```

```
INSERT INTO Trains (TrainID, TrainName, TrainType, Capacity, SourceStationID,  
DestinationStationID) VALUES  
(1, 'Express 1', 'Express', 200, 1, 2),  
(2, 'Local 2', 'Local', 150, 1, 3),  
(3, 'Fast Track', 'Express', 250, 2, 4),  
(4, 'Night Rider', 'Overnight', 100, 3, 5),  
(5, 'Scenic Route', 'Scenic', 80, 4, 1),  
(6, 'Coastal Express', 'Express', 300, 6, 12),  
(7, 'Desert Special', 'Regional', 120, 13, 14),  
(8, 'Lake Shuttle', 'Local', 90, 7, 8),  
(9, 'Mountain Climber', 'Tourist', 60, 8, 9),  
(10, 'City Express', 'Express', 250, 10, 11);
```

```
SELECT * FROM Trains;
```

TrainID	TrainName	TrainType	Capacity	SourceStationID	DestinationStationID
1	Express 1	Express	200	1	2
2	Local 2	Local	150	1	3
3	Fast Track	Express	250	2	4
4	Night Rider	Overnight	100	3	5
5	Scenic Route	Scenic	80	4	1
6	Coastal Express	Express	300	6	12
7	Desert Special	Regional	120	13	14
8	Lake Shuttle	Local	90	7	8
9	Mountain Climber	Tourist	60	8	9
10	City Express	Express	250	10	11

```
CREATE TABLE Stations (
    StationID INT PRIMARY KEY,
    StationName VARCHAR(100) NOT NULL,
    Location VARCHAR(255) NOT NULL,
    State VARCHAR(50) NOT NULL
);
```

```
INSERT INTO Stations (StationID, StationName, Location, State) VALUES
(1, 'Central Station', 'Downtown', 'NY'),
(2, 'East Station', 'East Side', 'NY'),
(3, 'West Station', 'West Side', 'NY'),
(4, 'North Station', 'North End', 'NY'),
(5, 'South Station', 'South End', 'NY'),
(6, 'River Station', 'Riverside', 'CA'),
(7, 'Lake Station', 'Lakeside', 'CA'),
(8, 'Mountain Station', 'Mountainview', 'CA'),
(9, 'Valley Station', 'Valley View', 'CA'),
(10, 'City Station', 'City Center', 'IL'),
(11, 'Suburb Station', 'Suburb Area', 'IL'),
(12, 'Coastal Station', 'Coastal Region', 'FL'),
(13, 'Desert Station', 'Desert View', 'AZ'),
(14, 'Forest Station', 'Forest Area', 'OR'),
(15, 'Island Station', 'Island Region', 'HI');
```

```
SELECT * FROM Stations;
```

StationID	StationName	Location	State
1	Central Station	Downtown	NY
2	East Station	East Side	NY
3	West Station	West Side	NY
4	North Station	North End	NY
5	South Station	South End	NY
6	River Station	Riverside	CA
7	Lake Station	Lakeside	CA
8	Mountain Station	Mountainview	CA
9	Valley Station	Valley View	CA
10	City Station	City Center	IL
11	Suburb Station	Suburb Area	IL
12	Coastal Station	Coastal Region	FL
13	Desert Station	Desert View	AZ
14	Forest Station	Forest Area	OR
15	Island Station	Island Region	HI

```
CREATE TABLE Schedules (  
    ScheduleID INT PRIMARY KEY,  
    TrainID INT,  
    DepartureTime DATETIME NOT NULL,  
    ArrivalTime DATETIME NOT NULL,  
    SourceStationID INT,  
    DestinationStationID INT,  
    FOREIGN KEY (TrainID) REFERENCES Trains(TrainID),  
    FOREIGN KEY (SourceStationID) REFERENCES Stations(StationID),  
    FOREIGN KEY (DestinationStationID) REFERENCES Stations(StationID)  
);
```

```
INSERT INTO Schedules (ScheduleID, TrainID, DepartureTime, ArrivalTime, SourceStationID,  
DestinationStationID) VALUES  
(1, 1, '2024-09-20 08:00:00', '2024-09-20 09:00:00', 1, 2),  
(2, 2, '2024-09-20 09:30:00', '2024-09-20 10:15:00', 1, 3),  
(3, 3, '2024-09-20 11:00:00', '2024-09-20 12:00:00', 2, 4),  
(4, 4, '2024-09-20 13:00:00', '2024-09-20 15:00:00', 3, 5),  
(5, 5, '2024-09-20 16:00:00', '2024-09-20 17:30:00', 4, 1),  
(6, 6, '2024-09-20 18:00:00', '2024-09-20 20:00:00', 6, 12),  
(7, 7, '2024-09-20 21:00:00', '2024-09-20 22:30:00', 13, 14),  
(8, 8, '2024-09-20 07:00:00', '2024-09-20 08:30:00', 7, 8),  
(9, 9, '2024-09-20 08:45:00', '2024-09-20 10:00:00', 8, 9),  
(10, 10, '2024-09-20 09:00:00', '2024-09-20 09:45:00', 10, 11);
```

Select \* from Schedules;

ScheduleID	TrainID	DepartureTime	ArrivalTime	SourceStationID	DestinationStationID
1	1	2024-09-20 08:00:00.000	2024-09-20 09:00:00.000	1	2
2	2	2024-09-20 09:30:00.000	2024-09-20 10:15:00.000	1	3
3	3	2024-09-20 11:00:00.000	2024-09-20 12:00:00.000	2	4
4	4	2024-09-20 13:00:00.000	2024-09-20 15:00:00.000	3	5
5	5	2024-09-20 16:00:00.000	2024-09-20 17:30:00.000	4	1
6	6	2024-09-20 18:00:00.000	2024-09-20 20:00:00.000	6	12
7	7	2024-09-20 21:00:00.000	2024-09-20 22:30:00.000	13	14
8	8	2024-09-20 07:00:00.000	2024-09-20 08:30:00.000	7	8
9	9	2024-09-20 08:45:00.000	2024-09-20 10:00:00.000	8	9
10	10	2024-09-20 09:00:00.000	2024-09-20 09:45:00.000	10	11

```
CREATE TABLE Passengers (  
    PassengerID INT PRIMARY KEY,  
    FirstName VARCHAR(50) NOT NULL,  
    LastName VARCHAR(50) NOT NULL,  
    Age INT NOT NULL,  
    Gender VARCHAR(10) NOT NULL,  
    Email VARCHAR(100) UNIQUE  
);
```

```
INSERT INTO Passengers (PassengerID, FirstName, LastName, Age, Gender, Email) VALUES  
(1, 'John', 'Doe', 30, 'Male', 'john.doe@example.com'),  
(2, 'Jane', 'Smith', 25, 'Female', 'jane.smith@example.com'),  
(3, 'Mike', 'Johnson', 35, 'Male', 'mike.j@example.com'),  
(4, 'Emily', 'Davis', 28, 'Female', 'emily.d@example.com'),
```

```
(5, 'Chris', 'Brown', 40, 'Male', 'chris.brown@example.com'),
(6, 'Katie', 'Wilson', 22, 'Female', 'katie.w@example.com'),
(7, 'David', 'Lee', 33, 'Male', 'david.lee@example.com'),
(8, 'Sarah', 'Garcia', 29, 'Female', 'sarah.g@example.com'),
(9, 'Daniel', 'Martinez', 31, 'Male', 'daniel.m@example.com'),
(10, 'Jessica', 'Hernandez', 27, 'Female', 'jessica.h@example.com');
```

```
SELECT * FROM Passengers;
```

PassengerID	FirstName	LastName	Age	Gender	Email
1	John	Doe	30	Male	john.doe@example.com
2	Jane	Smith	25	Female	jane.smith@example.com
3	Mike	Johnson	35	Male	mike.j@example.com
4	Emily	Davis	28	Female	emily.d@example.com
5	Chris	Brown	40	Male	chris.brown@example.com
6	Katie	Wilson	22	Female	katie.w@example.com
7	David	Lee	33	Male	david.lee@example.com
8	Sarah	Garcia	29	Female	sarah.g@example.com
9	Daniel	Martinez	31	Male	daniel.m@example.com
10	Jessica	Hernandez	27	Female	jessica.h@example.com

```
CREATE TABLE Bookings (
    BookingID INT PRIMARY KEY,
    PassengerID INT,
    ScheduleID INT,
    BookingDate DATETIME NOT NULL,
    SeatNumber INT NOT NULL,
    Status VARCHAR(20) NOT NULL,
    FOREIGN KEY (PassengerID) REFERENCES Passengers(PassengerID),
    FOREIGN KEY (ScheduleID) REFERENCES Schedules(ScheduleID)
);

INSERT INTO Bookings (BookingID, PassengerID, ScheduleID, BookingDate, SeatNumber, Status)
VALUES
(1, 3, 1, '2024-09-19 14:00:00', 1, 'Confirmed'),
(2, 1, 2, '2024-09-19 14:30:00', 5, 'Confirmed'),
(3, 4, 3, '2024-09-19 15:00:00', 3, 'Cancelled'),
(4, 6, 4, '2024-09-19 15:30:00', 1, 'Confirmed'),
(5, 2, 5, '2024-09-19 16:00:00', 2, 'Confirmed'),
(6, 7, 6, '2024-09-19 16:30:00', 6, 'Pending'),
(7, 8, 7, '2024-09-19 17:00:00', 7, 'Confirmed'),
(8, 5, 8, '2024-09-19 17:30:00', 8, 'Confirmed'),
(9, 9, 9, '2024-09-19 18:00:00', 2, 'Cancelled'),
(10, 10, 10, '2024-09-19 18:30:00', 4, 'Confirmed');
```

```
SELECT * FROM Bookings;
```

1. Find all trains operating from Central Station.

```
SELECT Trains.*
FROM Trains
WHERE SourceStationID = (SELECT StationID FROM Stations WHERE StationName =
'Central Station');
```

TrainID	TrainName	TrainType	Capacity	SourceStationID	DestinationStationID
1	Express 1	Express	200	1	2
2	Local 2	Local	150	1	3

2. Get a list of all schedules for a specific train type.

```
SELECT Schedules.*
FROM Schedules
JOIN Trains ON Schedules.TrainID = Trains.TrainID
WHERE Trains.TrainType = 'Express';
```

ScheduleID	TrainID	DepartureTime	ArrivalTime	SourceStationID	DestinationStationID
1	1	2024-09-20 08:00:00.000	2024-09-20 09:00:00.000	1	2
3	3	2024-09-20 11:00:00.000	2024-09-20 12:00:00.000	2	4
6	6	2024-09-20 18:00:00.000	2024-09-20 20:00:00.000	6	12
10	10	2024-09-20 09:00:00.000	2024-09-20 09:45:00.000	10	11

3. Find the number of bookings for each train.

```
SELECT Schedules.TrainID, COUNT(Bookings.BookingID) AS BookingCount
FROM Schedules
LEFT JOIN Bookings ON Schedules.ScheduleID = Bookings.ScheduleID
GROUP BY Schedules.TrainID;
```

TrainID	BookingCount
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1

4. Get the total number of seats booked for each train type.

```
SELECT Trains.TrainType, SUM(Trains.Capacity) AS TotalSeatsBooked
FROM Trains
JOIN Schedules ON Trains.TrainID = Schedules.TrainID
JOIN Bookings ON Schedules.ScheduleID = Bookings.ScheduleID
GROUP BY Trains.TrainType;
```

TrainType	TotalSeatsBooked
Express	1000
Local	240
Overnight	100
Regional	120
Scenic	80
Tourist	60

5. Find the oldest passenger in the database.

```
SELECT * FROM Passengers
WHERE Age = (SELECT MAX(Age) FROM Passengers);
```

PassengerID	FirstName	LastName	Age	Gender	Email
5	Chris	Brown	40	Male	chris.brown@example.com

6. Find passengers who booked a train that departs from a Central Station.

```
SELECT DISTINCT Passengers.*
FROM Passengers
JOIN Bookings ON Passengers.PassengerID = Bookings.PassengerID
JOIN Schedules ON Bookings.ScheduleID = Schedules.ScheduleID
WHERE Schedules.SourceStationID = (SELECT StationID FROM Stations WHERE
StationName = 'Central Station');
```

PassengerID	FirstName	LastName	Age	Gender	Email
1	John	Doe	30	Male	john.doe@example.com
3	Mike	Johnson	35	Male	mike.j@example.com

7. Find all passengers who are adults and have booked a train.

```
SELECT DISTINCT Passengers.*
FROM Passengers
JOIN Bookings ON Passengers.PassengerID = Bookings.PassengerID
WHERE Age >= 18;
```

PassengerID	FirstName	LastName	Age	Gender	Email
1	John	Doe	30	Male	john.doe@example.com
2	Jane	Smith	25	Female	jane.smith@example.com
3	Mike	Johnson	35	Male	mike.j@example.com
4	Emily	Davis	28	Female	emily.d@example.com
5	Chris	Brown	40	Male	chris.brown@example.com
6	Katie	Wilson	22	Female	katie.w@example.com
7	David	Lee	33	Male	david.lee@example.com
8	Sarah	Garcia	29	Female	sarah.g@example.com
9	Daniel	Martinez	31	Male	daniel.m@example.com
10	Jessica	Hernandez	27	Female	jessica.h@example.com

8. Get the average capacity of trains by train type.

```
SELECT TrainType, AVG(Capacity) AS AverageCapacity
FROM Trains
GROUP BY TrainType;
```

TrainType	AverageCapacity
Express	250
Local	120
Overnight	100
Regional	120
Scenic	80
Tourist	60

9. Find the total number of passengers for each train.

```
SELECT Schedules.TrainID, COUNT(DISTINCT Bookings.PassengerID) AS  
PassengerCount  
FROM Schedules  
LEFT JOIN Bookings ON Schedules.ScheduleID = Bookings.ScheduleID  
GROUP BY Schedules.TrainID;
```

TrainID	PassengerCount
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1

10. Get the average booking count for each train type.

```
SELECT Trains.TrainType, AVG(BookingCount) AS AvgBookingCount  
FROM (SELECT Schedules.TrainID, COUNT(Bookings.BookingID) AS BookingCount  
FROM Schedules  
LEFT JOIN Bookings  
ON Schedules.ScheduleID = Bookings.ScheduleID  
GROUP BY Schedules.TrainID) AS BookingCounts  
JOIN Trains  
ON BookingCounts.TrainID = Trains.TrainID  
GROUP BY Trains.TrainType;
```

TrainType	AvgBookingCount
Express	1
Local	1
Overnight	1
Regional	1
Scenic	1
Tourist	1

11. Get the maximum and minimum ages of passengers who have booked a train.

```
SELECT MAX(Age) AS MaxAge, MIN(Age) AS MinAge  
FROM Passengers  
WHERE PassengerID IN (SELECT PassengerID FROM Bookings);
```

MaxAge	MinAge
40	22

12. Get the top 5 busiest stations based on bookings.

```
SELECT TOP(5)
Stations.StationName,
COUNT(Bookings.BookingID) AS BookingCount
FROM Stations
LEFT JOIN Trains
ON Stations.StationID = Trains.SourceStationID
LEFT JOIN Schedules
ON Trains.TrainID = Schedules.TrainID
LEFT JOIN Bookings
ON Schedules.ScheduleID = Bookings.ScheduleID
GROUP BY Stations.StationName
ORDER BY BookingCount DESC;
```

StationName	BookingCount
Central Station	2
City Station	1
East Station	1
Desert Station	1
Lake Station	1

13. Get the top 3 most popular trains based on the number of bookings.

```
SELECT TOP(3)
Schedules.TrainID, COUNT(Bookings.BookingID) AS BookingCount
FROM Schedules
LEFT JOIN Bookings ON Schedules.ScheduleID = Bookings.ScheduleID
GROUP BY Schedules.TrainID
ORDER BY BookingCount DESC;
```

TrainID	BookingCount
4	1
3	1
2	1

14. Find the percentage of confirmed vs. cancelled bookings.

```
SELECT
    (COUNT(CASE WHEN Status = 'Confirmed' THEN 1 END) * 100 / COUNT(*)) AS
ConfirmedPercentage,
    (COUNT(CASE WHEN Status = 'Cancelled' THEN 1 END) * 100 / COUNT(*)) AS
CancelledPercentage
FROM Bookings;
```

ConfirmedPercentage	CancelledPercentage
70	20

15. Get all trains that contain "Express" in their name.

```
SELECT * FROM Trains WHERE TrainName LIKE '%Express%';
```

TrainID	TrainName	TrainType	Capacity	SourceStationID	DestinationStationID
1	Express 1	Express	200	1	2
6	Coastal Express	Express	300	6	12
10	City Express	Express	250	10	11



16. Get a summary of bookings per train including the number of confirmed and cancelled.

```
SELECT Schedules.TrainID,  
       COUNT(CASE WHEN Bookings.Status = 'Confirmed' THEN 1 END) AS Confirmed,  
       COUNT(CASE WHEN Bookings.Status = 'Cancelled' THEN 1 END) AS Cancelled  
FROM Schedules  
LEFT JOIN Bookings ON Schedules.ScheduleID = Bookings.ScheduleID  
GROUP BY Schedules.TrainID;
```

TrainID	Confirmed	Cancelled
1	1	0
2	1	0
3	0	1
4	1	0
5	1	0
6	0	0
7	1	0
8	1	0
9	0	1
10	1	0

17. Identify the most frequently booked train.

```
SELECT TOP(1)  
Schedules.TrainID,  
COUNT(Bookings.BookingID) AS BookingCount  
FROM Schedules  
JOIN Bookings ON Schedules.ScheduleID = Bookings.ScheduleID  
GROUP BY Schedules.TrainID  
ORDER BY BookingCount DESC;
```

TrainID	BookingCount
2	1

18. Find the station with the highest number of departing trains.

```
SELECT TOP(1)  
Stations.StationName, COUNT(Trains.TrainID) AS TrainCount  
FROM Stations  
JOIN Trains ON Stations.StationID = Trains.SourceStationID  
GROUP BY Stations.StationName  
ORDER BY TrainCount DESC;
```

StationName	TrainCount
Central Station	2

19. Get a summary of the average age of passengers by train type.

```
SELECT Trains.TrainType, AVG(Passengers.Age) AS AverageAge  
FROM Passengers  
JOIN Bookings ON Passengers.PassengerID = Bookings.PassengerID  
JOIN Schedules ON Bookings.ScheduleID = Schedules.ScheduleID  
JOIN Trains ON Schedules.TrainID = Trains.TrainID  
GROUP BY Trains.TrainType;
```

TrainType	AverageAge
Express	30
Local	35
Overnight	22
Regional	29
Scenic	25
Tourist	31

20. Get the details of the last booking made by each passenger.

```
SELECT Passengers.*, Bookings.*
FROM Passengers
JOIN Bookings ON Passengers.PassengerID = Bookings.PassengerID
WHERE BookingDate IN (SELECT MAX(BookingDate) FROM Bookings GROUP BY
PassengerID);
```

PassengerID	FirstName	LastName	Age	Gender	Email	BookingID	PassengerID	ScheduleID	BookingDate	SeatNumber	Status
1	John	Doe	35	Male	john.d@example.com	2	1	2	2024-09-19 14:30:00.000	5	Confirmed
2	Jane	Smith	25	Female	jane.smith@example.com	5	2	5	2024-09-19 16:00:00.000	2	Confirmed
3	Mike	Johnson	35	Male	mike.j@example.com	1	3	1	2024-09-19 14:00:00.000	1	Confirmed
4	Emily	Davis	28	Female	emily.d@example.com	3	4	3	2024-09-19 15:00:00.000	3	Cancelled
5	Chris	Brown	40	Male	chris.brown@example.com	8	5	8	2024-09-19 17:30:00.000	8	Confirmed
6	Katie	Wilson	22	Female	katie.w@example.com	4	6	4	2024-09-19 15:30:00.000	1	Confirmed
7	David	Lee	33	Male	david.lee@example.com	6	7	6	2024-09-19 16:30:00.000	6	Pending
8	Sarah	Garcia	29	Female	sarah.g@example.com	7	8	7	2024-09-19 17:00:00.000	7	Confirmed
9	Daniel	Martinez	31	Male	daniel.m@example.com	9	9	9	2024-09-19 18:00:00.000	2	Cancelled
10	Jessica	Hernandez	27	Female	jessica.h@example.com	10	10	10	2024-09-19 18:30:00.000	4	Confirmed

21. Count the total number of unique trains used for bookings.

```
SELECT COUNT(DISTINCT TrainID) AS UniqueTrains
FROM Schedules
JOIN Bookings ON Schedules.ScheduleID = Bookings.ScheduleID;
```

UniqueTrains
10

22. Find the most popular train based on the percentage of total bookings.

```
SELECT TOP(1)
Schedules.TrainID,
(COUNT(Bookings.BookingID) * 100 / (SELECT COUNT(*) FROM Bookings)) AS
BookingPercentage
FROM Schedules
LEFT JOIN Bookings ON Schedules.ScheduleID = Bookings.ScheduleID
GROUP BY Schedules.TrainID
ORDER BY BookingPercentage DESC;
```

TrainID	BookingPercentage
2	10

23. Get a summary of the status of bookings, displaying a message based on the status.

```
SELECT
Status,
COUNT(*) AS TotalBookings,
CASE
    WHEN Status = 'Confirmed' THEN 'Confirmed bookings'
    WHEN Status = 'Cancelled' THEN 'Bookings cancelled'
    ELSE 'Other'
END AS StatusMessage
FROM Bookings
GROUP BY Status;
```

Status	TotalBookings	StatusMessage
Cancelled	2	Bookings cancelled
Confirmed	7	Confirmed bookings
Pending	1	Other

24. Get the average age of passengers who booked an express train.

```
SELECT AVG(Passengers.Age) AS AvgAge
FROM Passengers
JOIN Bookings ON Passengers.PassengerID = Bookings.PassengerID
JOIN Schedules ON Bookings.ScheduleID = Schedules.ScheduleID
JOIN Trains ON Schedules.TrainID = Trains.TrainID
WHERE Trains.TrainType = 'Express';
```

AvgAge
30

25. List all trains sorted by their capacity, then by name.

```
SELECT * FROM Trains
ORDER BY Capacity DESC, TrainName ASC;
```

TrainID	TrainName	TrainType	Capacity	SourceStationID	DestinationStationID
6	Coastal Express	Express	300	6	12
10	City Express	Express	250	10	11
3	Fast Track	Express	250	2	4
1	Express 1	Express	200	1	2
2	Local 2	Local	150	1	3
7	Desert Special	Regional	120	13	14
4	Night Rider	Overnight	100	3	5
8	Lake Shuttle	Local	90	7	8
5	Scenic Route	Scenic	80	4	1
9	Mountain Climber	Tourist	60	8	9

26. Find the top 10 stations with the most bookings.

```
SELECT TOP(10)
Stations.StationName, COUNT(Bookings.BookingID) AS BookingCount
FROM Stations
JOIN Trains ON Stations.StationID = Trains.SourceStationID
JOIN Schedules ON Trains.TrainID = Schedules.TrainID
LEFT JOIN Bookings ON Schedules.ScheduleID = Bookings.ScheduleID
GROUP BY Stations.StationName
ORDER BY BookingCount DESC;
```

StationName	BookingCount
Central Station	2
West Station	1
River Station	1
North Station	1
Mountain Station	1
Lake Station	1
East Station	1
Desert Station	1
City Station	1

27. Get bookings for each train type and their total revenue.

```
SELECT Trains.TrainType,
COUNT(Bookings.BookingID) AS TotalBookings,
COUNT(Bookings.BookingID) * 50 AS TotalRevenue
FROM Trains
JOIN Schedules ON Trains.TrainID = Schedules.TrainID
JOIN Bookings ON Schedules.ScheduleID = Bookings.ScheduleID
GROUP BY Trains.TrainType;
```

TrainType	TotalBookings	TotalRevenue
Express	4	200
Local	2	100
Overnight	1	50
Regional	1	50
Scenic	1	50
Tourist	1	50

28. Determine the correlation between train capacity and booking counts.

```
SELECT Trains.Capacity, COUNT(Bookings.BookingID) AS BookingCount
FROM Trains
LEFT JOIN Schedules ON Trains.TrainID = Schedules.TrainID
LEFT JOIN Bookings ON Schedules.ScheduleID = Bookings.ScheduleID
GROUP BY Trains.Capacity;
```

Capacity	BookingCount
60	1
80	1
90	1
100	1
120	1
150	1
200	1
250	2
300	1

29. Find the average age of passengers for each train type.

```
SELECT Trains.TrainType, AVG(Passengers.Age) AS AvgAge
FROM Passengers
JOIN Bookings ON Passengers.PassengerID = Bookings.PassengerID
JOIN Schedules ON Bookings.ScheduleID = Schedules.ScheduleID
JOIN Trains ON Schedules.TrainID = Trains.TrainID
GROUP BY Trains.TrainType;
```

TrainType	AvgAge
Express	30
Local	35
Overnight	22
Regional	29
Scenic	25
Tourist	31

30. Identify trains that have a higher cancellation rate than the average.

```
WITH CancellationRates AS (
    SELECT
        Trains.TrainID,
        COUNT(CASE WHEN Bookings.Status = 'Cancelled' THEN 1 END) * 100.0 /
        COUNT(Bookings.BookingID) AS CancellationRate
    FROM Trains
    JOIN Schedules
    ON Trains.TrainID = Schedules.TrainID
    LEFT JOIN Bookings
    ON Schedules.ScheduleID = Bookings.ScheduleID
    GROUP BY Trains.TrainID)
SELECT
    TrainID, CancellationRate
```

```

FROM CancellationRates
WHERE CancellationRate > (SELECT AVG(CancellationRate)
FROM (SELECT
COUNT(CASE WHEN Status = 'Cancelled' THEN 1 END) * 100.0 / COUNT(*) AS
CancellationRate
FROM Bookings
GROUP BY ScheduleID
) AS Rates
);

```

TrainID	CancellationRate
3	100.000000000000
9	100.000000000000