INTRODUCTION

Welcome to our Railway Reservation System project! This system is designed to facilitate the booking of train tickets for passengers. It provides a user-friendly interface for passengers to search for trains, make bookings, and manage their reservations.

Our system consists of five main tables: Passengers, Trains, Stations, Bookings, and Seats. Passengers can create bookings for available trains, and each booking is associated with a specific passenger and train. The Seats table keeps track of seat availability for each train on different dates.

With our system, passengers can easily search for trains based on their preferences, such as departure and destination stations, departure time, and available seats. They can view detailed information about each train, including its schedule and available seats, before making a booking. Once a booking is made, passengers can manage their reservations, including canceling or modifying bookings if needed.

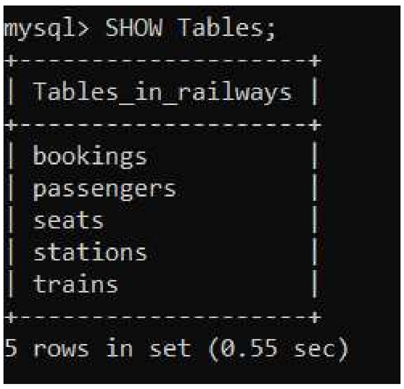
Furthermore, our system provides administrators with tools to manage train schedules, update seat availability, and monitor bookings. Administrators can ensure smooth operation of the railway system by overseeing train operations and resolving any issues that may arise.

Let's embark on a journey together with our Railway Reservation System!

LIST OF ENTITIES & ATTRIBUTES:

->ENTITIES:

SHOW Tables;

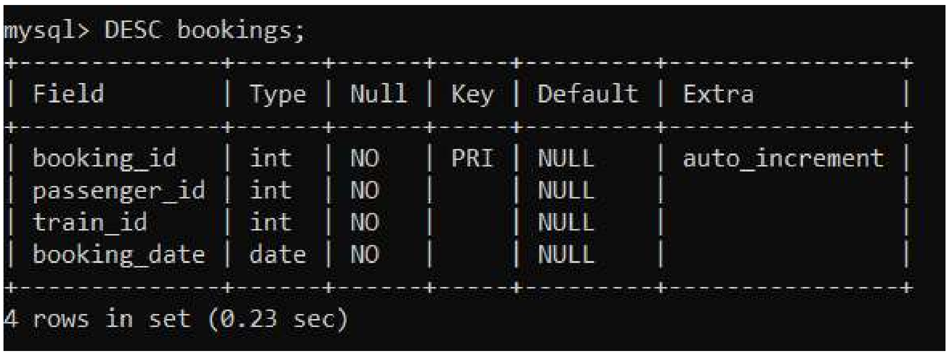


```
mysql> SHOW Tables;
+--------------------+
| Tables_in_railways |
+--------------------+
| bookings           |
| passengers         |
| seats              |
| stations           |
| trains             |
+--------------------+
5 rows in set (0.55 sec)
```

->ATTRIBUTES:

1.bookings

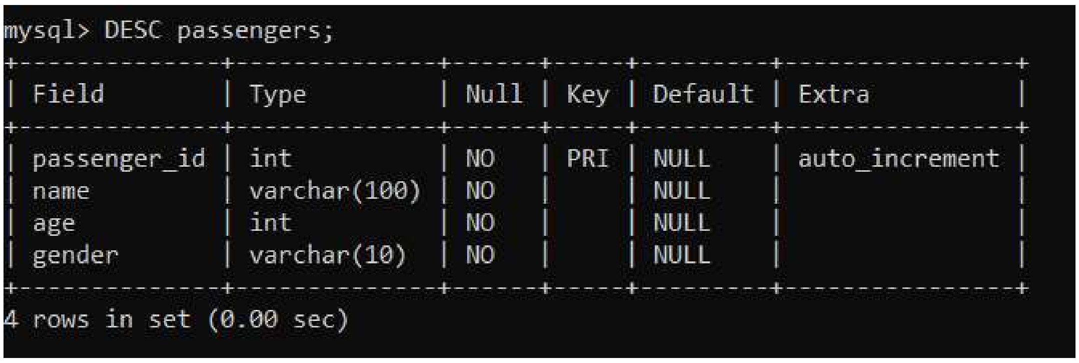
DESC bookings;



```
mysql> DESC bookings;
+--------------+------+------+-----+---------+----------------+
| Field        | Type | Null | Key | Default | Extra          |
+--------------+------+------+-----+---------+----------------+
| booking_id   | int  | NO   | PRI | NULL    | auto_increment |
| passenger_id | int  | NO   |     | NULL    |                |
| train_id     | int  | NO   |     | NULL    |                |
| booking_date | date | NO   |     | NULL    |                |
+--------------+------+------+-----+---------+----------------+
4 rows in set (0.23 sec)
```

2.passengers

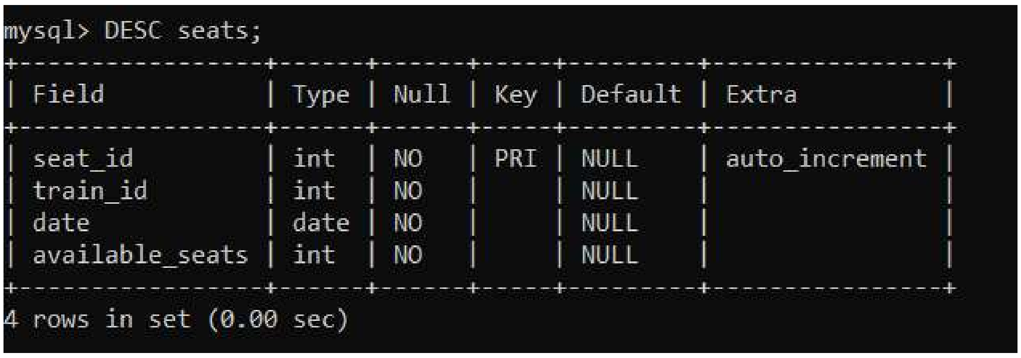
DESC passengers;



```
mysql> DESC passengers;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| passenger_id | int          | NO   | PRI | NULL    | auto_increment |
| name         | varchar(100) | NO   |     | NULL    |                |
| age          | int          | NO   |     | NULL    |                |
| gender       | varchar(10)  | NO   |     | NULL    |                |
+--------------+--------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

3.seats

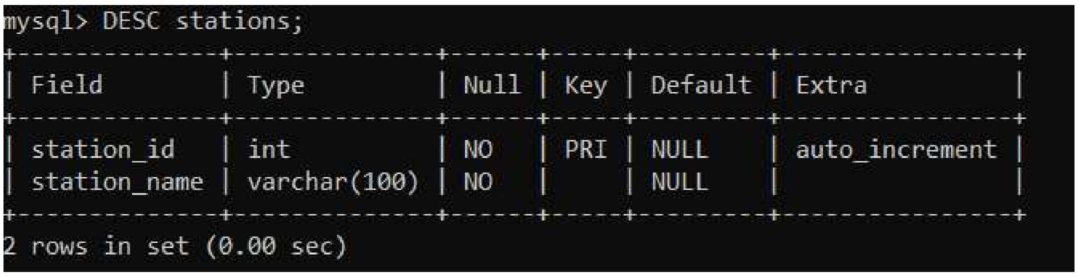
DESC seats;



```
mysql> DESC seats;
+-----------------+------+------+-----+---------+----------------+
| Field           | Type | Null | Key | Default | Extra          |
+-----------------+------+------+-----+---------+----------------+
| seat_id         | int  | NO   | PRI | NULL    | auto_increment |
| train_id        | int  | NO   |     | NULL    |                |
| date            | date | NO   |     | NULL    |                |
| available_seats | int  | NO   |     | NULL    |                |
+-----------------+------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

4.stations

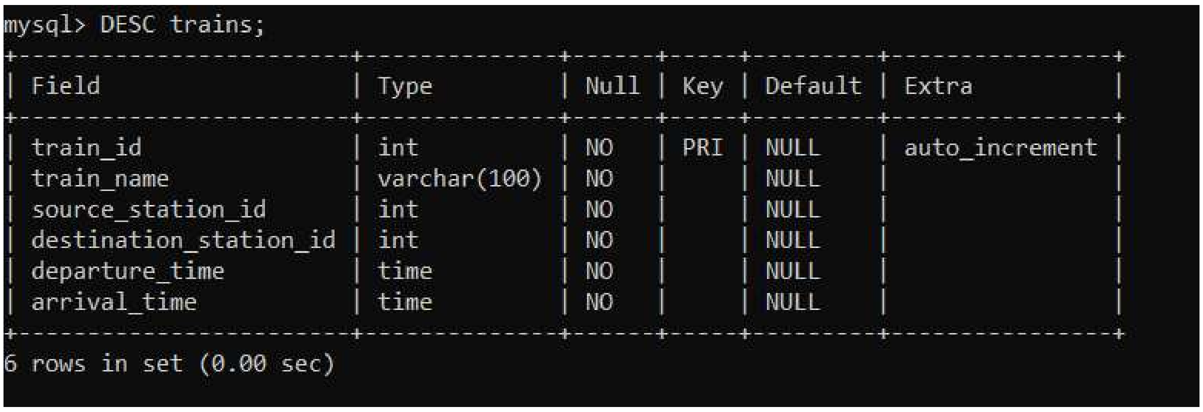
DESC stations;



```
mysql> DESC stations;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| station_id   | int          | NO   | PRI | NULL    | auto_increment |
| station_name | varchar(100) | NO   |     | NULL    |                |
+--------------+--------------+------+-----+---------+----------------+
2 rows in set (0.00 sec)
```

5.trains

DESC trains;



```
mysql> DESC trains;
+------------------------+--------------+------+-----+---------+----------------+
| Field                  | Type         | Null | Key | Default | Extra          |
+------------------------+--------------+------+-----+---------+----------------+
| train_id               | int          | NO   | PRI | NULL    | auto_increment |
| train_name             | varchar(100) | NO   |     | NULL    |                |
| source_station_id      | int          | NO   |     | NULL    |                |
| destination_station_id | int          | NO   |     | NULL    |                |
| departure_time         | time         | NO   |     | NULL    |                |
| arrival_time           | time         | NO   |     | NULL    |                |
+------------------------+--------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)
```

LIST OF RELATIONSHIPS Between Tables:

1. **Passengers to Bookings:** One-to-Many relationship. One passenger can make multiple bookings, but each booking belongs to only one passenger.

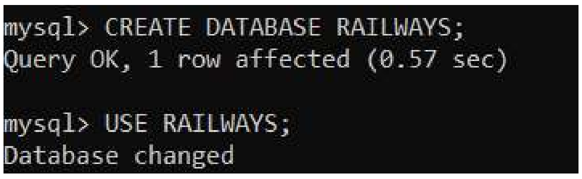
2. **Trains to Bookings:** One-to-Many relationship. One train can have multiple bookings, but each booking is associated with only one train.

3. **Trains to Seats:** One-to-Many relationship. One train can have multiple seats, but each seat is associated with only one train.

4. **Stations to Trains (source_station_id):** One-to-Many relationship. One station can be the source station for multiple trains, but each train has only one source station.

5. **Stations to Trains (destination_station_id):** One-to-Many relationship. One station can be the destination station for multiple trains, but each train has only one destination station.

CREATE TABLE SQL QUERIES:



```
mysql> CREATE DATABASE RAILWAYS;
Query OK, 1 row affected (0.57 sec)

mysql> USE RAILWAYS;
Database changed
```

```
mysql> CREATE TABLE Passengers (
    ->     passenger_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     name VARCHAR(100) NOT NULL,
    ->     age INT NOT NULL,
    ->     gender VARCHAR(10) NOT NULL
    -> );
```

```
mysql> CREATE TABLE Trains (
    ->     train_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     train_name VARCHAR(100) NOT NULL,
    ->     source_station_id INT NOT NULL,
    ->     destination_station_id INT NOT NULL,
    ->     departure_time TIME NOT NULL,
    ->     arrival_time TIME NOT NULL
    -> );
```

```
mysql> CREATE TABLE Stations (
    ->     station_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     station_name VARCHAR(100) NOT NULL
    -> );
```

```
mysql> CREATE TABLE Bookings (
    ->     booking_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     passenger_id INT NOT NULL,
    ->     train_id INT NOT NULL,
    ->     booking_date DATE NOT NULL
    -> );

mysql> CREATE TABLE Seats (
    ->     seat_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     train_id INT NOT NULL,
    ->     date DATE NOT NULL,
    ->     available_seats INT NOT NULL
    -> );
```

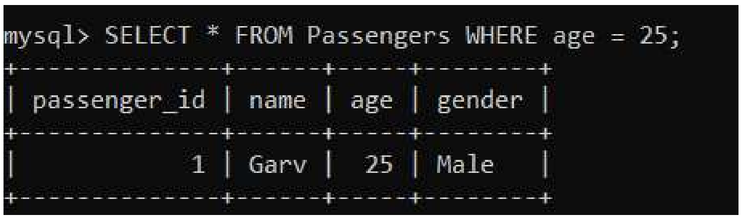
INSERT VALUES INTO TABLE SQL QUERIES:

```
mysql> INSERT INTO Passengers (name, age, gender) VALUES
    -> ('Garv', 25, 'Male'),
    -> ('Advik', 30, 'Male'),
    -> ('Aarav', 28, 'Male');

mysql> INSERT INTO Trains (train_name, source_station_id,
destination_station_id, departure_time, arrival_time) VALUES
    -> ('Delhi Express', 1, 4, '08:00:00', '16:00:00'),
    -> ('Mumbai Superfast', 2, 3, '10:00:00', '14:00:00'),
    -> ('Chennai Mail', 3, 1, '09:00:00', '17:00:00');

mysql> INSERT INTO Stations (station_name) VALUES
    -> ('Delhi'),
    -> ('Mumbai'),
    -> ('Chennai'),
    -> ('Kolkata');

mysql> INSERT INTO Bookings (passenger_id, train_id, booking_date) VALUES
    -> (1, 1, '2024-05-15'),
    -> (2, 2, '2024-05-16'),
    -> (3, 3, '2024-05-17');

mysql> INSERT INTO Seats (train_id, date, available_seats) VALUES
    -> (1, '2024-05-15', 150),
    -> (2, '2024-05-16', 200),
    -> (3, '2024-05-17', 100);
```

SQL QUERIES TO RETRIEVE DATA:

1.Retrieve details of passengers aged 25.

SELECT * FROM Passengers WHERE age = 25;



```
mysql> SELECT * FROM Passengers WHERE age = 25;
+--------------+------+-----+--------+
| passenger_id | name | age | gender |
+--------------+------+-----+--------+
|            1 | Garv |  25 | Male   |
+--------------+------+-----+--------+
```

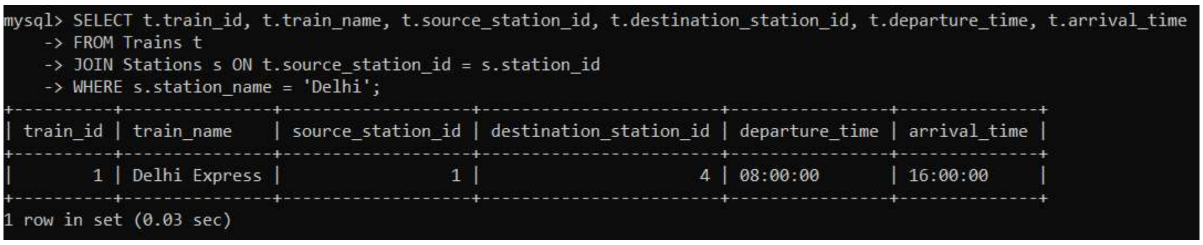
2. Retrieve the train details including train ID, train name, source station ID, destination station ID, departure time, and arrival time for trains departing from the station 'Delhi'.

SELECT t.train_id, t.train_name, t.source_station_id, t.destination_station_id, t.departure_time, t.arrival_time

-> FROM Trains t

-> JOIN Stations s ON t.source_station_id = s.station_id

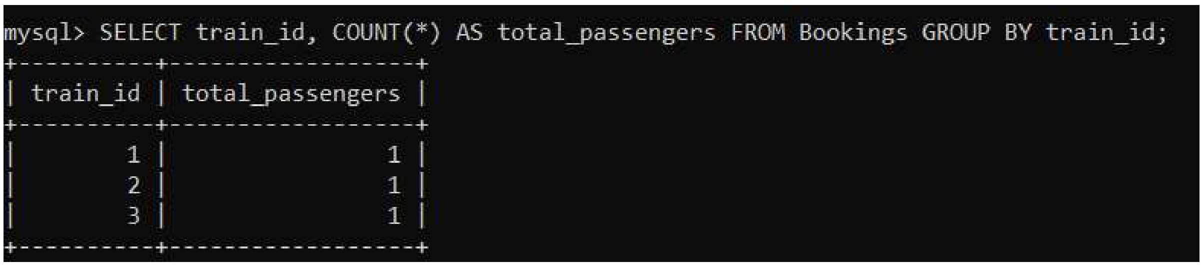
-> WHERE s.station_name = 'Delhi';



```
mysql> SELECT t.train_id, t.train_name, t.source_station_id, t.destination_station_id, t.departure_time, t.arrival_time
    -> FROM Trains t
    -> JOIN Stations s ON t.source_station_id = s.station_id
    -> WHERE s.station_name = 'Delhi';
+----------+---------------+-------------------+------------------------+----------------+--------------+
| train_id | train_name    | source_station_id | destination_station_id | departure_time | arrival_time |
+----------+---------------+-------------------+------------------------+----------------+--------------+
|        1 | Delhi Express |                 1 |                      4 | 08:00:00       | 16:00:00     |
+----------+---------------+-------------------+------------------------+----------------+--------------+
1 row in set (0.03 sec)
```

3. Calculate the total number of passengers booked on each train by counting the number of bookings for each train and grouping the results by train ID.

SELECT train_id, COUNT(*) AS total_passengers FROM Bookings GROUP BY train_id;



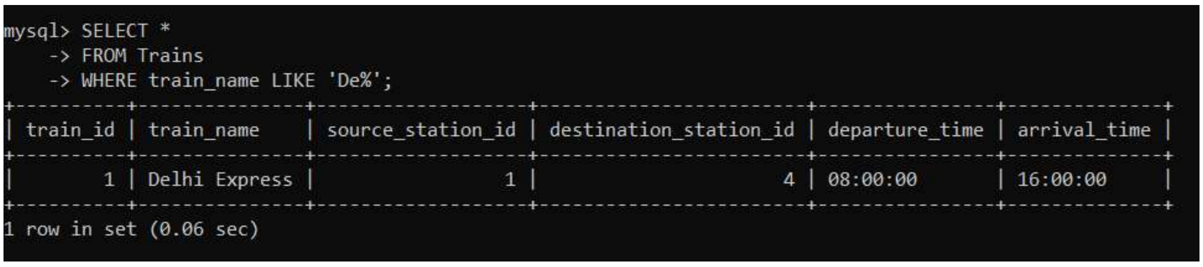
```
mysql> SELECT train_id, COUNT(*) AS total_passengers FROM Bookings GROUP BY train_id;
+----------+------------------+
| train_id | total_passengers |
+----------+------------------+
|        1 |                1 |
|        2 |                1 |
|        3 |                1 |
+----------+------------------+
```

4. Retrieve all details of trains where the train name starts with 'De'.

SELECT *

-> FROM Trains

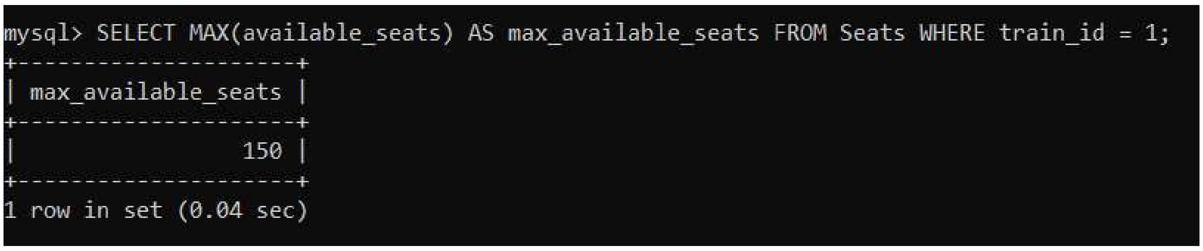
-> WHERE train_name LIKE 'De%';



```
mysql> SELECT *
    -> FROM Trains
    -> WHERE train_name LIKE 'De%';
+----------+---------------+-------------------+------------------------+----------------+--------------+
| train_id | train_name    | source_station_id | destination_station_id | departure_time | arrival_time |
+----------+---------------+-------------------+------------------------+----------------+--------------+
|        1 | Delhi Express |                 1 |                      4 | 08:00:00       | 16:00:00     |
+----------+---------------+-------------------+------------------------+----------------+--------------+
1 row in set (0.06 sec)
```

5. Retrieve the maximum available seats for train ID 1 from the Seats table.

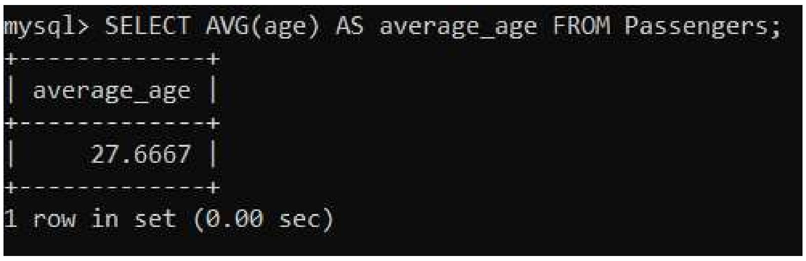
SELECT MAX(available_seats) AS max_available_seats FROM Seats WHERE train_id = 1;



```
mysql> SELECT MAX(available_seats) AS max_available_seats FROM Seats WHERE train_id = 1;
+---------------------+
| max_available_seats |
+---------------------+
|                 150 |
+---------------------+
1 row in set (0.04 sec)
```

6. Calculate the average age of passengers from the Passengers table.

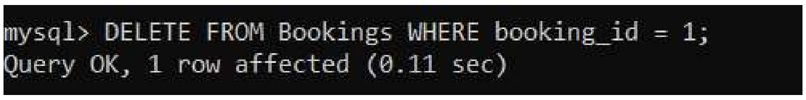
SELECT AVG(age) AS average_age FROM Passengers;



```
mysql> SELECT AVG(age) AS average_age FROM Passengers;
+-------------+
| average_age |
+-------------+
|     27.6667 |
+-------------+
1 row in set (0.00 sec)
```

7. Delete the booking with booking ID 1 from the Bookings table.

DELETE FROM Bookings WHERE booking_id = 1;



```
mysql> DELETE FROM Bookings WHERE booking_id = 1;
Query OK, 1 row affected (0.11 sec)
```

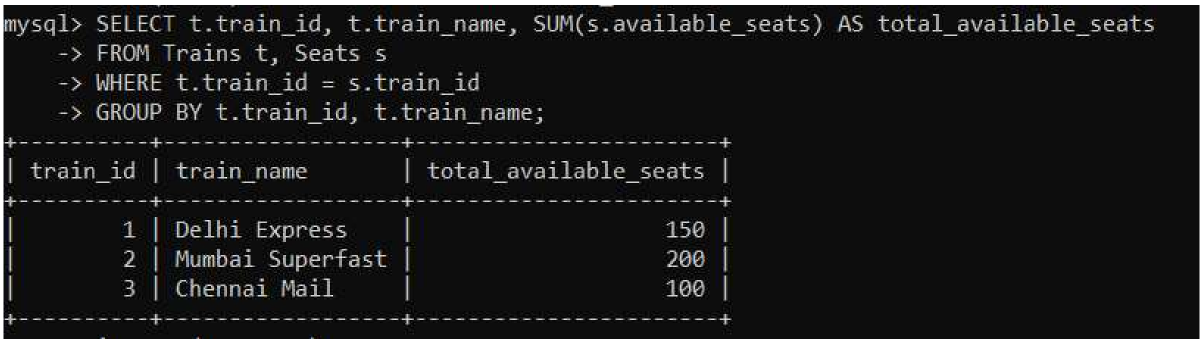
8. Retrieve the train ID, train name, and total available seats by summing up the available seats for each train from the Trains and Seats tables, grouped by train ID and train name.

SELECT t.train_id, t.train_name, SUM(s.available_seats) AS total_available_seats

-> FROM Trains t, Seats s

-> WHERE t.train_id = s.train_id

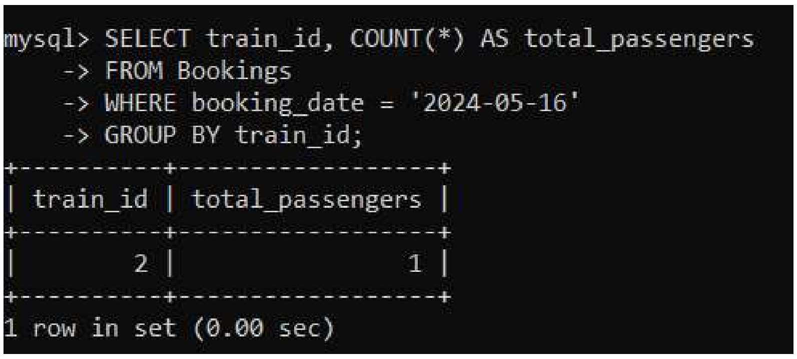
-> GROUP BY t.train_id, t.train_name;



```
mysql> SELECT t.train_id, t.train_name, SUM(s.available_seats) AS total_available_seats
    -> FROM Trains t, Seats s
    -> WHERE t.train_id = s.train_id
    -> GROUP BY t.train_id, t.train_name;
+----------+------------------+-----------------------+
| train_id | train_name       | total_available_seats |
+----------+------------------+-----------------------+
|        1 | Delhi Express    |                   150 |
|        2 | Mumbai Superfast |                   200 |
|        3 | Chennai Mail     |                   100 |
+----------+------------------+-----------------------+
```

9. Count the total number of passengers booked on each train for the date '2024-05-16', grouping the results by train ID.

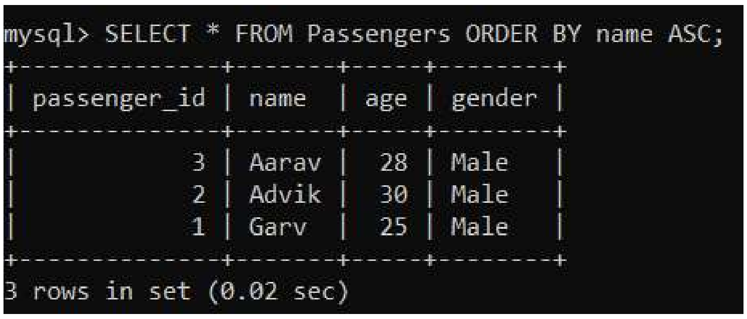
```
SELECT train_id, COUNT(*) AS total_passengers
    -> FROM Bookings
    -> WHERE booking_date = '2024-05-16'
    -> GROUP BY train_id;
```



```
mysql> SELECT train_id, COUNT(*) AS total_passengers
    -> FROM Bookings
    -> WHERE booking_date = '2024-05-16'
    -> GROUP BY train_id;
+----------+------------------+
| train_id | total_passengers |
+----------+------------------+
|        2 |                1 |
+----------+------------------+
1 row in set (0.00 sec)
```

10. Retrieve all details of passengers from the Passengers table, ordered by name in ascending order.

```
SELECT * FROM Passengers ORDER BY name ASC;
```



```
mysql> SELECT * FROM Passengers ORDER BY name ASC;
+--------------+-------+-----+--------+
| passenger_id | name  | age | gender |
+--------------+-------+-----+--------+
|            3 | Aarav |  28 | Male   |
|            2 | Advik |  30 | Male   |
|            1 | Garv  |  25 | Male   |
+--------------+-------+-----+--------+
3 rows in set (0.02 sec)
```

CONCLUSION

Thank you for exploring our Railway Reservation System project! We have developed this system to provide a seamless and convenient experience for passengers to book train tickets and manage their reservations. Throughout this project, we have implemented a robust database schema and utilized SQL queries to perform various operations, such as retrieving passenger details, finding available trains, and calculating total passengers per train.

By leveraging modern technologies and best practices in database management, we have created a reliable and efficient platform for both passengers and administrators. Passengers can easily plan their journeys, book tickets hassle-free, and enjoy a smooth travel experience. Administrators, on the other hand, can efficiently manage train schedules, update seat availability, and monitor bookings to ensure the smooth operation of the railway system.

We are committed to continuously improving and enhancing our Railway Reservation System to meet the evolving needs of our users. Your feedback and suggestions are invaluable to us as we strive to provide the best possible service. Thank you for your support, and we look forward to serving you in your future travels!

REFERENCES

1. Database Design Tutorial:

https://www.tutorialspoint.com/dbms/index.htm

2. SQL Tutorial:

https://www.geeksforgeeks.org/sql-tutorial/

3. Library Management System Project Idea:
https://www.geeksforgeeks.org/library-management-system-project-in-c-sharp/

4. Library Management System GitHub Repository:
https://github.com/topics/library-management-system