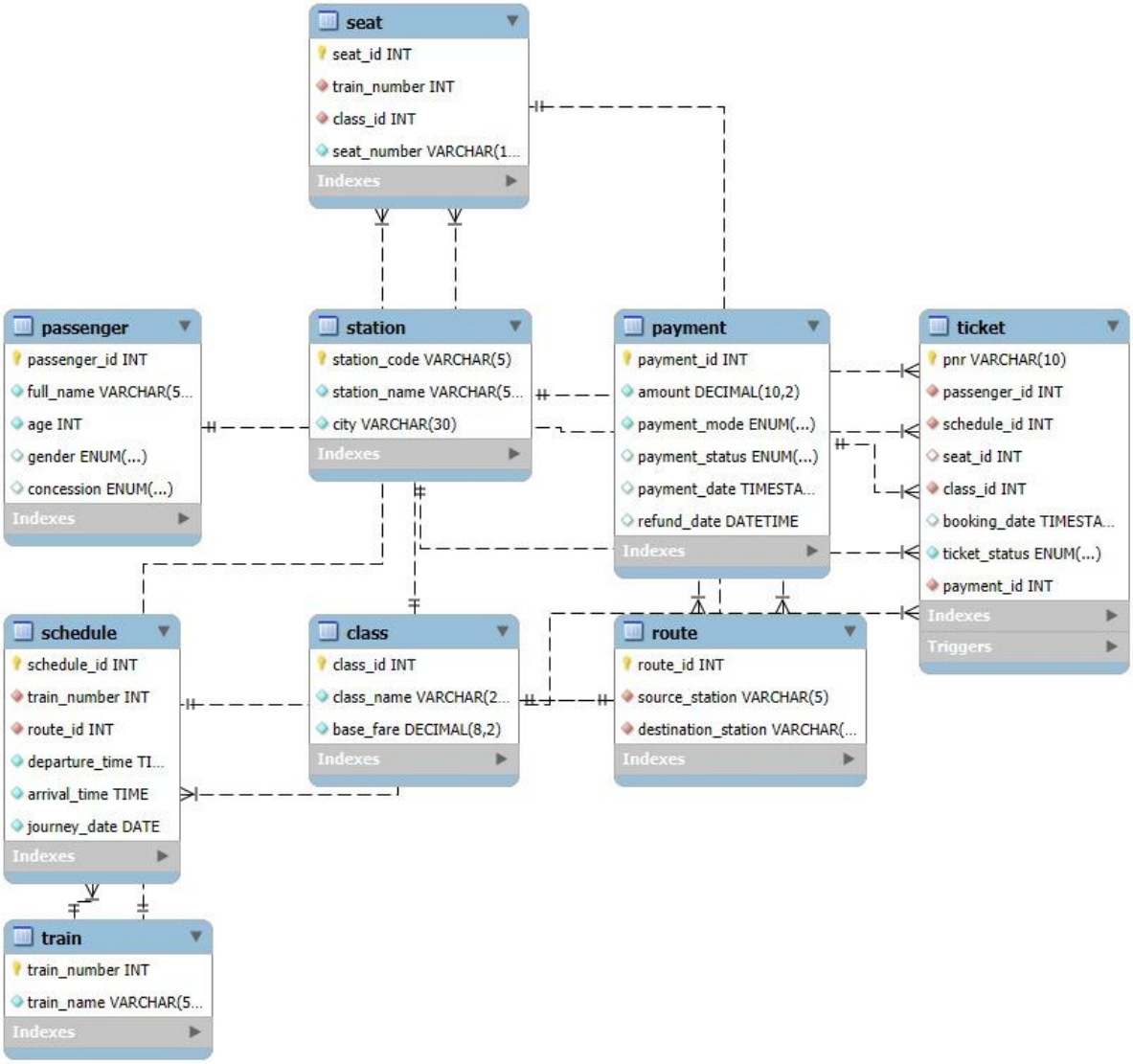
1.)ER Diagram


seat
seat_id INT
train_number INT
class_id INT
seat_number VARCHAR(1...
Indexes
passenger
passenger_id INT
full_name VARCHAR(5...
age INT
gender ENUM(...)
concession ENUM(...)
Indexes
station
station_code VARCHAR(5)
station_name VARCHAR(5...
city VARCHAR(30)
Indexes
payment
payment_id INT
amount DECIMAL(10,2)
payment_mode ENUM(...)
payment_status ENUM(...)
payment_date TIMESTA...
refund_date DATETIME
Indexes
ticket
pnr VARCHAR(10)
passenger_id INT
schedule_id INT
seat_id INT
class_id INT
booking_date TIMESTA...
ticket_status ENUM(...)
payment_id INT
Indexes
Triggers
schedule
schedule_id INT
train_number INT
route_id INT
departure_time TI...
arrival_time TIME
journey_date DATE
Indexes
class
class_id INT
class_name VARCHAR(2...
base_fare DECIMAL(8,2)
Indexes
route
route_id INT
source_station VARCHAR(5)
destination_station VARCHAR(...
Indexes
train
train_number INT
train_name VARCHAR(5...
Indexes

2.)**Relational Schema**

- Passenger

```
passenger(
  passenger_id INT PRIMARY KEY,
  full_name VARCHAR(50),
  age INT,
  gender ENUM('M', 'F', 'O'),
  concession ENUM('student', 'senior_citizen', 'military', 'none'))
```

- Station

```
station(
  station_code VARCHAR(5) PRIMARY KEY,
  station_name VARCHAR(50),
  city VARCHAR(30)
)
```

- Train

```
train(
  train_number INT PRIMARY KEY,
  train_name VARCHAR(50)
)
```

- Route

```
route(
  route_id INT PRIMARY KEY,
  source_station VARCHAR(5),
  destination_station VARCHAR(5),
  FOREIGN KEY (source_station) REFERENCES station(station_code),
  FOREIGN KEY (destination_station) REFERENCES station(station_code)
)
```

- Schedule

```
schedule(
  schedule_id INT PRIMARY KEY,
  train_number INT,
  route_id INT,
  departure_time TIME,
  arrival_time TIME,
  journey_date DATE,
  FOREIGN KEY (train_number) REFERENCES train(train_number),
  FOREIGN KEY (route_id) REFERENCES route(route_id)
```

)

- Class

class(
class_id INT PRIMARY KEY,
class_name VARCHAR(20),
base_fare DECIMAL(8,2)
)

- Seat

seat(
seat_id INT PRIMARY KEY,
train_number INT,
class_id INT,
seat_number VARCHAR(10),
FOREIGN KEY (train_number) REFERENCES train(train_number),
FOREIGN KEY (class_id) REFERENCES class(class_id)
)

- Payment

payment(
payment_id INT PRIMARY KEY,
amount DECIMAL(10,2),
payment_mode ENUM('UPI', 'CARD', 'NETBANKING'),
payment_status ENUM('SUCCESS', 'FAILED', 'PENDING'),
payment_date TIMESTAMP,
refund_date DATETIME
)

- Ticket

ticket(
pnr VARCHAR(10) PRIMARY KEY,
passenger_id INT,
schedule_id INT,
seat_id INT,
class_id INT,
booking_date TIMESTAMP,
ticket_status ENUM('CONFIRMED', 'CANCELLED', 'WAITING'),
payment_id INT,
FOREIGN KEY (passenger_id) REFERENCES passenger(passenger_id),
FOREIGN KEY (schedule_id) REFERENCES schedule(schedule_id),
FOREIGN KEY (seat_id) REFERENCES seat(seat_id),
FOREIGN KEY (class_id) REFERENCES class(class_id),
FOREIGN KEY (payment_id) REFERENCES payment(payment_id)
)

3.)Sample Data Summary:

```
mysql> select count(*) from class;
+----------+
| count(*) |
+----------+
|        3 |
+----------+
1 row in set (0.00 sec)

mysql> select count(*) from passenger;
+----------+
| count(*) |
+----------+
|       50 |
+----------+
1 row in set (0.01 sec)

mysql> select count(*) from payment;
+----------+
| count(*) |
+----------+
|      100 |
+----------+
1 row in set (0.00 sec)
```

```
mysql> select count(*) from station;
+----------+
| count(*) |
+----------+
|        5 |
+----------+
1 row in set (0.00 sec)

mysql> select count(*) from ticket;
+----------+
| count(*) |
+----------+
|      100 |
+----------+
1 row in set (0.00 sec)

mysql> select count(*) from train;
+----------+
| count(*) |
+----------+
|        3 |
+----------+
1 row in set (0.00 sec)
```

```
mysql> select count(*) from route;
+----------+
| count(*) |
+----------+
|        3 |
+----------+
1 row in set (0.00 sec)

mysql> select count(*) from schedule;
+----------+
| count(*) |
+----------+
|       30 |
+----------+
1 row in set (0.00 sec)

mysql> select count(*) from seat;
+----------+
| count(*) |
+----------+
|       45 |
+----------+
1 row in set (0.00 sec)
```

4.)Procedures:

1. GetPNRStatus

Checks status and journey details of a ticket using PNR.

2. GetTrainSchedule

Fetches schedule of a particular train.

3. CheckAvailability

Lists available seats on a train for a given date and class

4. ListPassengers

Lists passengers booked for a particular schedule.

5. GetWaitlist

Fetches waitlisted passengers for a particular train schedule.

6. CalculateTrainCancellationRefund

Computes total and refundable amount when a train is cancelled.

7. GetTotalRevenue

Returns the total payment collected between two dates.

8. GetCancellationRecords

Lists cancelled tickets and corresponding refund amounts.

9. FindBusiestRoute

Finds the most traveled route

10. GenerateItemizedBill

Generates a fare breakdown and discount applied based on concession.

11. CancelTicket

Cancels a ticket, handles seat reallocation, and updates payment.

12. BookTicket

Books a ticket, applies concession, and allocates seats based on availability

5.)Triggers

This SQL trigger named AfterTicketUpdate automatically executes after an update is made to the Ticket table.

To **automatically manage RAC (Reservation Against Cancellation)** bookings.
If a **confirmed ticket is cancelled**, the **next eligible RAC ticket** is:

- **Promoted to 'Confirmed'**
- **Assigned the freed seat**

This helps maintain fairness and real-time updates in the reservation system.

6.) Other Queries Implemented

- Listing all trains that are fully booked on a particular date.
- Finding the number of tickets booked under each concession category.
- Getting the most booked train on a particular date.