McCoy College of Business, Texas State University

Course: CIS.5355 Database Management Systems

Project Final Phase

Instructor: *Barbara Hewitt*

Students: Hafila Morais Max Morais, Nisarga Arsikere Chidananda, Rodolfo Olivares Tamayo, Vaishali Sunil Bhirud

Due date: *November 26, 2023*

Phase 2(80 points)

Continue working on the problem statement you created in phase 1 and turn in the following items (the following items are required for each team):

1. Business Case and Rules.
2. ERD.
3. SQL script to create the complete database.
4. SQL script to insert records into the tables (each table should have at least 10 rows of data).
5. SQL queries for generating reports (minimum 2 reports from each team member should be created).

1. The Problem Statement:

Design a database system for a train booking system that can handle the reservation of train tickets and store customer information. Consider the various entities involved, such as trains, stations, schedules, bookings, and passengers. Include the necessary attributes for each entity and establish the relationships between these entities. Also, consider how to handle seat availability for different classes on each train and how to track reservations made by customers.

Business Rules:

1. A train starts at a train station, ends at a train station, and can stop at different train stations along its journey.

2. A train has a planned start time for its journey and planned times that it arrives at each station. Multiple trains run on the same day.

3. The times for a train are not the same every day. For example, on weekends, the trains run less frequently.

4. Each train journey has a train with several different carriages, each of which has a different class, such as First, Second, and Economy.

5. Each class of carriage has a specific number of seats which defines the capacity of passengers.

6. Each class of carriage has a different price for booking a ticket on a train, and there is one price for weekdays and one for weekends.

7. When a passenger books a train ticket, they will need to provide their name, email address, and a password for their account.

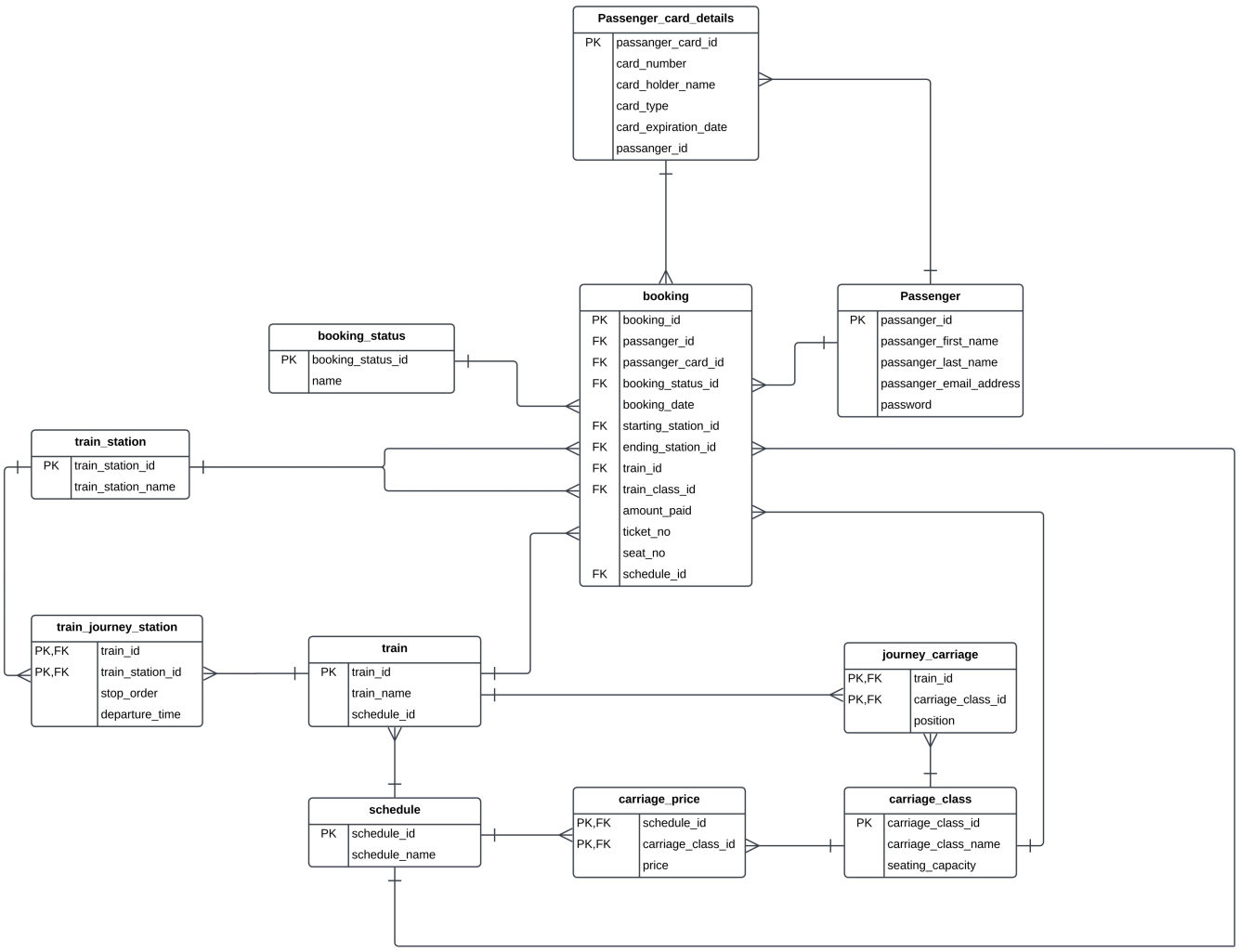
8. A Passenger can book the tickets with multiple cards.

9. A passenger can book train tickets on the website. The passenger will select the date they wish to travel, their starting station, their departing station, the time of the train they want to get, the class of ticket, and will be shown the price.

10. A train ticket has the details that the customer specified during the booking, along with a ticket number and their seat on the carriage.

11. A passenger can view their booking status and can cancel their booking at any time before the train has departed.

2. ERD


Passenger_card_details
PK passanger_card_id
card_number
card_holder_name
card_type
card_expiration_date
passanger_id
booking
PK booking_id
FK passanger_id
FK passanger_card_id
FK booking_status_id
booking_date
FK starting_station_id
FK ending_station_id
FK train_id
FK train_class_id
amount_paid
ticket_no
seat_no
FK schedule_id
Passenger
PK passanger_id
passanger_first_name
passanger_last_name
passanger_email_address
password
booking_status
PK booking_status_id
name
train_station
PK train_station_id
train_station_name
train_journey_station
PK,FK train_id
PK,FK train_station_id
stop_order
departure_time
train
PK train_id
train_name
schedule_id
journey_carriage
PK,FK train_id
PK,FK carriage_class_id
position
schedule
PK schedule_id
schedule_name
carriage_price
PK,FK schedule_id
PK,FK carriage_class_id
price
carriage_class
PK carriage_class_id
carriage_class_name
seating_capacity

3. SQL script to create the complete database.

```
-- CREATION OF DATABASE --

CREATE DATABASE train_booking;

USE train_booking;

-- CREATION OF TABLES --

CREATE TABLE schedule (
schedule_id VARCHAR(20) PRIMARY KEY,
schedule_name VARCHAR(200)
);

CREATE TABLE train (
train_id VARCHAR(20) PRIMARY KEY,
train_name VARCHAR(200),
schedule_id VARCHAR(20),
FOREIGN KEY (schedule_id) REFERENCES schedule (schedule_id) ON DELETE SET NULL
);

CREATE TABLE train_station (
train_station_id VARCHAR(20) PRIMARY KEY,
train_station_name VARCHAR(200)
);

CREATE TABLE train_journey_station (
train_id VARCHAR(20),
train_station_id VARCHAR(20),
stop_order INT,
departure_time DATETIME,
PRIMARY KEY (train_id,train_station_id),
FOREIGN KEY (train_id) REFERENCES train (train_id) ON DELETE CASCADE,
FOREIGN KEY (train_station_id) REFERENCES train_station (train_station_id) ON DELETE
CASCADE
);

CREATE TABLE carriage_class (
   carriage_class_id VARCHAR(20) PRIMARY KEY,
   carriage_class_name VARCHAR(200),
   seating_capacity INT
);
```

```
CREATE TABLE carriage_price (
   schedule_id VARCHAR(20),
   carriage_class_id VARCHAR(20),
   price MONEY,
   PRIMARY KEY (schedule_id, carriage_class_id),
   FOREIGN KEY (schedule_id) REFERENCES schedule (schedule_id) ON DELETE CASCADE,
   FOREIGN KEY (carriage_class_id) REFERENCES carriage_class (carriage_class_id) ON DELETE
CASCADE
);

CREATE TABLE journey_carriage (
   train_id VARCHAR(20),
   carriage_class_id VARCHAR(20),
   position INT,
   PRIMARY KEY (train_id, carriage_class_id),
   FOREIGN KEY (train_id) REFERENCES train (train_id) ON DELETE CASCADE,
   FOREIGN KEY (carriage_class_id) REFERENCES carriage_class (carriage_class_id) ON DELETE
CASCADE
);

CREATE TABLE booking_status (
   booking_status_id VARCHAR(20) PRIMARY KEY,
   name VARCHAR(200)
);


CREATE TABLE passenger (
passenger_id VARCHAR(20) PRIMARY KEY,
passenger_first_name VARCHAR(200),
passenger_last_name VARCHAR(200),
passenger_email_address VARCHAR(200),
password VARCHAR(200)
);


CREATE TABLE passenger_card_details (
   passenger_card_id VARCHAR(20) PRIMARY KEY,
   card_number BIGINT ,
   card_holder_name VARCHAR(100),
   card_type VARCHAR(50),
   card_expiration_date DATE,
   passenger_id VARCHAR(20),
   FOREIGN KEY (passenger_id) REFERENCES passenger (passenger_id) ON DELETE SET NULL
);
```

```
CREATE TABLE booking (
booking_id VARCHAR(20) PRIMARY KEY,
passenger_id VARCHAR(20),
passenger_card_id VARCHAR(20),
booking_status_id VARCHAR(20),
booking_date DATE,
starting_station_id VARCHAR(20),
ending_station_id VARCHAR(20),
train_id VARCHAR(20),
schedule_id VARCHAR(20),
ticket_class_id VARCHAR(20),
amount_paid INT,
ticket_no INT,
seat_no VARCHAR(10),
FOREIGN KEY (passenger_id) REFERENCES passenger (passenger_id) ON DELETE SET NULL,
FOREIGN KEY (passenger_card_id) REFERENCES passenger_card_details (passenger_card_id) ON
DELETE SET NULL,
FOREIGN KEY (booking_status_id) REFERENCES booking_status (booking_status_id) ON DELETE
SET NULL,
FOREIGN KEY (starting_station_id) REFERENCES train_station (train_station_id) ON DELETE SET
NULL,
FOREIGN KEY (ending_station_id) REFERENCES train_station (train_station_id) ON DELETE NO
ACTION,
FOREIGN KEY (train_id) REFERENCES train (train_id) ON DELETE SET NULL,
FOREIGN KEY (schedule_id) REFERENCES schedule (schedule_id) ON DELETE SET NULL,
FOREIGN KEY (ticket_class_id) REFERENCES carriage_class (carriage_class_id) ON DELETE SET
NULL
);
```

4. SQL script to insert records into the tables (each table should have at least 10 rows of data).

```
-- INSERTION OF VALUES --

INSERT INTO schedule (schedule_id, schedule_name) VALUES
('S1', 'Weekday Schedule'),
('S2', 'Weekend Schedule'),
('S3', 'Holiday Schedule'),
('S4', 'Special Event Schedule'),
('S5', 'Regular Schedule'),
('S6', 'Express Schedule'),
('S7', 'Local Schedule'),
('S8', 'Night Schedule'),
('S9', 'Morning Schedule'),
('S10', 'Afternoon Schedule');

INSERT INTO train (train_id, train_name, schedule_id) VALUES
('T1', '8:00 Express Train Montreal to Pittsburg', 'S6'),
('T2', '10:00 Local Train Raleigh to Miami', 'S7'),
('T3', '20:00 Night Train New Orleans to Austin', 'S8'),
('T4', '6:00 Morning Train Montreal to Pittsburg ', 'S9'),
('T5', '12:00 Afternoon Train Raleigh to Miami', 'S10'),
('T6', '14:00 Special Train New York to Atlanta', 'S4'),
('T7', '8:00 Regular Train New York to Washington', 'S5'),
('T8', '10:00 Weekend Train Washington to New York', 'S2'),
('T9', '12:00 Holiday Train Charleston to Atlanta', 'S3'),
('T10', '7:00 Weekday Train New Orleans to Charleston', 'S1');

INSERT INTO train_station (train_station_id, train_station_name) VALUES
('TS1', 'Montreal'),
('TS2', 'New York'),
('TS3', 'Pittsburg'),
('TS4', 'Washington'),
('TS5', 'Raleigh'),
('TS6', 'Charleston'),
('TS7', 'Miami'),
('TS8', 'New Orleans'),
('TS9', 'Atlanta'),
('TS10', 'Austin');
```

```
INSERT INTO train_journey_station (train_id, train_station_id, stop_order, departure_time)
VALUES
('T1', 'TS1', 1, '2024-01-01 08:00:00'),
('T1', 'TS2', 2, '2024-01-01 09:30:00'),
('T1', 'TS3', 3, '2024-01-01 11:00:00'),
('T2', 'TS5', 1, '2024-01-01 10:00:00'),
('T2', 'TS6', 2, '2024-01-01 11:30:00'),
('T2', 'TS7', 3, '2024-01-01 13:00:00'),
('T3', 'TS8', 1, '2024-01-01 20:00:00'),
('T3', 'TS9', 2, '2024-01-01 21:30:00'),
('T3', 'TS10', 3,'2024-01-01 23:00:00'),
('T4', 'TS1', 1, '2024-01-01 06:00:00'),
('T4', 'TS2', 2, '2024-01-01 07:30:00'),
('T4', 'TS3', 3, '2024-01-01 09:00:00'),
('T5', 'TS5', 1, '2024-01-01 12:00:00'),
('T5', 'TS6', 2, '2024-01-01 13:30:00'),
('T5', 'TS7', 3, '2024-01-01 14:30:00'),
('T6', 'TS2', 1, '2024-01-01 14:00:00'),
('T6', 'TS6', 2, '2024-01-01 17:30:00'),
('T6', 'TS9', 3, '2024-01-01 19:30:00'),
('T7', 'TS2', 1, '2024-01-01 08:00:00'),
('T7', 'TS3', 2, '2024-01-01 09:00:00'),
('T7', 'TS4', 3, '2024-01-01 10:00:00'),
('T8', 'TS4', 1, '2024-01-01 10:00:00'),
('T8', 'TS3', 2, '2024-01-01 11:00:00'),
('T8', 'TS2', 3, '2024-01-01 12:00:00'),
('T9', 'TS6', 1, '2024-01-01 12:00:00'),
('T9', 'TS8', 2, '2024-01-01 13:00:00'),
('T9', 'TS9', 3, '2024-01-01 14:00:00'),
('T10', 'TS8', 1, '2024-01-01 07:00:00'),
('T10', 'TS7', 2, '2024-01-01 08:00:00'),
('T10','TS6',3,'2024-01-01 09:00:00');

INSERT INTO carriage_class (carriage_class_id, carriage_class_name, seating_capacity) VALUES
   ('CC1', 'Economy Class', 150),
   ('CC2', 'Business Class', 50),
   ('CC3', 'First Class', 20);


INSERT INTO carriage_price (schedule_id, carriage_class_id, price) VALUES
   ('S1', 'CC1', 50.00),
   ('S1', 'CC2', 100.00),
```

('S1', 'CC3', 150.00),
('S2', 'CC1', 40.00),
('S2', 'CC2', 90.00),
('S2', 'CC3', 140.00),
('S3', 'CC1', 60.00),
('S3', 'CC2', 110.00),
('S3', 'CC3', 160.00),
('S4', 'CC1', 70.00),
('S4', 'CC2', 120.00),
('S4', 'CC3', 170.00),
('S5', 'CC1', 80.00),
('S5', 'CC2', 130.00),
('S5', 'CC3', 180.00),
('S6', 'CC1', 90.00),
('S6', 'CC2', 140.00),
('S6', 'CC3', 190.00),
('S7', 'CC1', 100.00),
('S7', 'CC2', 150.00),
('S7', 'CC3', 200.00),
('S8', 'CC1', 110.00),
('S8', 'CC2', 160.00),
('S8', 'CC3', 210.00),
('S9', 'CC1', 120.00),
('S9', 'CC2', 170.00),
('S9', 'CC3', 220.00),
('S10', 'CC1', 130.00),
('S10', 'CC2', 180.00),
('S10', 'CC3', 230.00);

INSERT INTO journey_carriage (train_id, carriage_class_id, position) VALUES
('T1', 'CC1', 1),
('T1', 'CC2', 2),
('T1', 'CC3', 3),
('T2', 'CC1', 1),
('T2', 'CC2', 2),
('T2', 'CC3', 3),
('T3', 'CC1', 1),
('T3', 'CC2', 2),
('T3', 'CC3', 3),
('T4', 'CC1', 1),
('T4', 'CC2', 2),
('T4', 'CC3', 3),
('T5', 'CC1', 1),

```
  ('T5', 'CC2', 2),
  ('T5', 'CC3', 3),
  ('T6', 'CC1', 1),
  ('T6', 'CC2', 2),
  ('T6', 'CC3', 3),
  ('T7', 'CC1', 1),
  ('T7', 'CC2', 2),
  ('T7', 'CC3', 3),
  ('T8', 'CC1', 1),
  ('T8', 'CC2', 2),
  ('T8', 'CC3', 3),
  ('T9', 'CC1', 1),
  ('T9', 'CC2', 2),
  ('T9', 'CC3', 3),
  ('T10', 'CC1', 1),
  ('T10', 'CC2', 2),
  ('T10', 'CC3', 3);

INSERT INTO booking_status (booking_status_id, name) VALUES
('BS1', 'Confirmed'),
('BS2', 'Pending'),
('BS3', 'Cancelled');


INSERT INTO passenger (passenger_id, passenger_first_name, passenger_last_name,
passenger_email_address, password) VALUES
 ('P1', 'John', 'Doe', 'john.doe@gmail.com', 'password1'),
 ('P2', 'Jane', 'Smith', 'jane.smith@outlook.com', 'password2'),
 ('P3', 'Michael', 'Johnson', 'michael.johnson@gmail.com', 'password3'),
 ('P4', 'Emily', 'Williams', 'emily.williams@gmail.com', 'password4'),
 ('P5', 'David', 'Brown', 'david.brown@gmail.com', 'password5'),
 ('P6', 'Olivia', 'Jones', 'olivia.jones@yahoo.com', 'password6'),
 ('P7', 'William', 'Taylor', 'william.taylor@yahoo.com', 'password7'),
 ('P8', 'Sophia', 'Anderson', 'sophia.anderson@gmail.com', 'password8'),
 ('P9', 'Matthew', 'Moore', 'matthew.moore@outlook.com', 'password9'),
 ('P10', 'Emma', 'Clark', 'emma.clark@gmail.com', 'password10');
```

```
 INSERT INTO passenger_card_details (passenger_card_id, card_number, card_holder_name,
card_type, card_expiration_date, passenger_id) VALUES
  ('PC1', 1234567890123456, 'John Doe', 'Visa', '2023-12-31', 'P1'),
  ('PC2', 2345678901234567, 'Jane Smith', 'MasterCard', '2024-06-30', 'P2'),
  ('PC3', 3456789012345678, 'Michael Johnson', 'American Express', '2023-09-15', 'P3'),
  ('PC4', 4567890123456789, 'Emily Williams', 'Discover', '2024-03-28', 'P4'),
  ('PC5', 5678901234567890, 'David Brown', 'Visa', '2023-11-30', 'P5'),
  ('PC6', 6789012345678901, 'Olivia Jones', 'MasterCard', '2024-02-15', 'P6'),
  ('PC7', 7890123456789012, 'William Taylor', 'American Express', '2023-08-22', 'P7'),
  ('PC8', 8901234567890123, 'Sophia Anderson', 'Discover', '2024-04-10', 'P8'),
  ('PC9', 9012345678901234, 'Matthew Moore', 'Visa', '2023-10-05', 'P9'),
  ('PC10', 1234901234567890, 'Emma Clark', 'MasterCard', '2024-01-18', 'P10');

INSERT INTO booking (
    booking_id,
    passenger_id,
    passenger_card_id,
    booking_status_id,
    booking_date,
    starting_station_id,
    ending_station_id,
    train_id,
    schedule_id,
    ticket_class_id,
    amount_paid,
    ticket_no,
    seat_no
) VALUES
('B1', 'P1', 'PC1', 'BS1', '2024-01-15', 'TS1', 'TS3', 'T1','S6', 'CC1', 90, 123456, 'A1'),
('B2', 'P2', 'PC2', 'BS2', '2024-02-20', 'TS5', 'TS7', 'T2','S7', 'CC2', 150, 789012, 'B3'),
('B3', 'P3', 'PC3', 'BS1', '2024-03-25', 'TS8', 'TS10', 'T3','S8', 'CC3', 210, 345678, 'C2'),
('B4', 'P4', 'PC4', 'BS2', '2024-04-10', 'TS1', 'TS3', 'T4','S9', 'CC1', 120, 901234, 'D4'),
('B5', 'P5', 'PC5', 'BS1', '2024-05-15', 'TS5', 'TS7', 'T5','S10', 'CC1', 130, 567890, 'E5'),
('B6', 'P6', 'PC6', 'BS2', '2024-06-20', 'TS2', 'TS9', 'T6','S4', 'CC3', 170, 234567, 'F1'),
('B7', 'P7', 'PC7', 'BS1', '2024-07-25', 'TS2', 'TS4', 'T7','S5', 'CC1', 80, 789012, 'G2'),
('B8', 'P8', 'PC8', 'BS2', '2024-08-30', 'TS4', 'TS2', 'T8','S2','CC2', 90, 456789, 'H3'),
('B9', 'P9', 'PC9', 'BS1', '2024-09-05', 'TS6', 'TS9', 'T9', 'S3','CC3', 160, 123490, 'I4'),
('B10', 'P10', 'PC10', 'BS2', '2024-10-10','TS8', 'TS6', 'T10','S1', 'CC2', 100, 987654, 'J5');
```

5. SQL queries for generating reports (minimum 2 reports from each team member should be created).

1. Give the details of the passenger who booked the train T1, T2 ,T9 and sort it by passanger_id in ascending order.

```
Select b.train_id ,b.passenger_ID , CONCAT(p.passenger_first_name,' ',p.passenger_last_name ) AS
Passanger_name
FROM booking b
INNER JOIN passenger p
ON b.passenger_id = p.passenger_id
where b.train_id IN ('T1','T2','T9')
order by b.passenger_id;
```

2. Write a SQL query to retrieve details of passenger along with their journey details, including the starting and ending stations, the total cost of the journey, and sort by the amount paid in descending order?

```
SELECT b.passenger_id ,
CONCAT(p.passenger_first_name,' ',p.passenger_last_name ) AS Passanger_name ,
b.booking_date,
s.schedule_name,
t.train_name,
ts_start.train_station_name as Journey_started_station,
ts_end.train_station_name as Journey_end_station,
b.amount_paid as Total_cost_of_Journey
FROM booking b
INNER JOIN passenger p
ON b.passenger_id = p.passenger_id
INNER JOIN schedule s
ON b.schedule_id = s.schedule_id
INNER JOIN train t
ON b.train_id = t.train_id
INNER JOIN train_station ts_start
ON b.starting_station_id = ts_start.train_station_id
INNER JOIN train_station ts_end
ON b.ending_station_id = ts_end.train_station_id
Order by b.amount_paid DESC;
```

3. Could you retrieve the top 3 bookings based on passenger details, booking date, schedule, train, carriage class, seat number, amount paid, and booking status, specifically for bookings where the amount paid is equal to or exceeds $150? and sort by amount paid in descending order.

```
SELECT TOP 3 b.passenger_id ,
CONCAT(p.passenger_first_name,' ',p.passenger_last_name ) AS Passanger_name ,
b.booking_date,
s.schedule_name,
t.train_name,
cc.carriage_class_name,
b.seat_no,
b.amount_paid,
bs.name as Booking_Status
FROM booking b
INNER JOIN passenger p
ON b.passenger_id = p.passenger_id
INNER JOIN schedule s
ON b.schedule_id = s.schedule_id
INNER JOIN train t
ON b.train_id = t.train_id
INNER JOIN carriage_class cc
ON b.ticket_class_id = cc.carriage_class_id
INNER JOIN booking_status bs
ON b.booking_status_id = bs.booking_status_id
WHERE b.amount_paid >= $150
ORDER BY b.amount_paid DESC;
```

4. Write a query to give the details of booking status of each passenger.

```
SELECT b.booking_id,CONCAT(p.passenger_first_name,' ',p.passenger_last_name ) as passenger_name
,bs.name as booking_status
FROM booking b
INNER JOIN passenger p
ON b.passenger_id = p.passenger_id
INNER JOIN booking_status bs
ON b.booking_status_id = bs.booking_status_id
Order By booking_id;
```

5. Provide a report showing the passenger details, booking date, schedule and train information, seat number, amount paid, carriage class, and booking status for each booking?

```
SELECT b.passenger_id ,
CONCAT(p.passenger_first_name,' ',p.passenger_last_name ) AS Passanger_name ,
b.booking_date,
s.schedule_name,
t.train_name,
cc.carriage_class_name,
b.seat_no,
b.amount_paid,
bs.name as Booking_Status
FROM booking b
INNER JOIN passenger p
ON b.passenger_id = p.passenger_id
INNER JOIN schedule s
ON b.schedule_id = s.schedule_id
INNER JOIN train t
ON b.train_id = t.train_id
INNER JOIN carriage_class cc
ON b.ticket_class_id = cc.carriage_class_id
INNER JOIN booking_status bs
ON b.booking_status_id = bs.booking_status_id
ORDER BY passenger_id;
```

6. Could you generate a report displaying the passenger details, booking date, train name, carriage class, and passenger card type used for each booking?

```
SELECT b.passenger_id ,
CONCAT(p.passenger_first_name,' ',p.passenger_last_name ) AS Passanger_name ,
b.booking_date,
t.train_name,
cc.carriage_class_name,
pc.card_type
FROM booking b
INNER JOIN passenger p
ON b.passenger_id = p.passenger_id
INNER JOIN train t
ON b.train_id = t.train_id
INNER JOIN carriage_class cc
ON b.ticket_class_id = cc.carriage_class_id
INNER JOIN passenger_card_details pc
ON b.passenger_card_id = pc.passenger_card_id
Order by passenger_id;
```

7. Write a query to retrieve total revenue generated per schedule, including the schedule ID, schedule name, and the amount paid and sort in descending order by the total amount paid?

```
SELECT b.schedule_id, s.schedule_name, b.amount_paid AS total_revenue
FROM booking b
JOIN schedule s ON b.schedule_id = s.schedule_id
ORDER BY b.amount_paid DESC;
```

8. Write a query to count the number of trains that use each station:

```
SELECT
  train_station.train_station_id,
  train_station.train_station_name,
  COUNT(train_journey_station.train_id) AS number_of_trains
FROM train_station
LEFT JOIN train_journey_station ON train_station.train_station_id =
train_journey_station.train_station_id
GROUP BY train_station.train_station_id, train_station.train_station_name
ORDER BY number_of_trains DESC;
```

9. Write a query to retrieve details of a train T1 journey with stations and departure times:

```
SELECT t.train_id, t.train_name, ts.train_station_name, tjs.stop_order, tjs.departure_time
FROM train t
INNER JOIN train_journey_station tjs
ON t.train_id = tjs.train_id
INNER JOIN train_station ts ON
tjs.train_station_id = ts.train_station_id
WHERE t.train_id = 'T1'
ORDER BY tjs.stop_order;
```