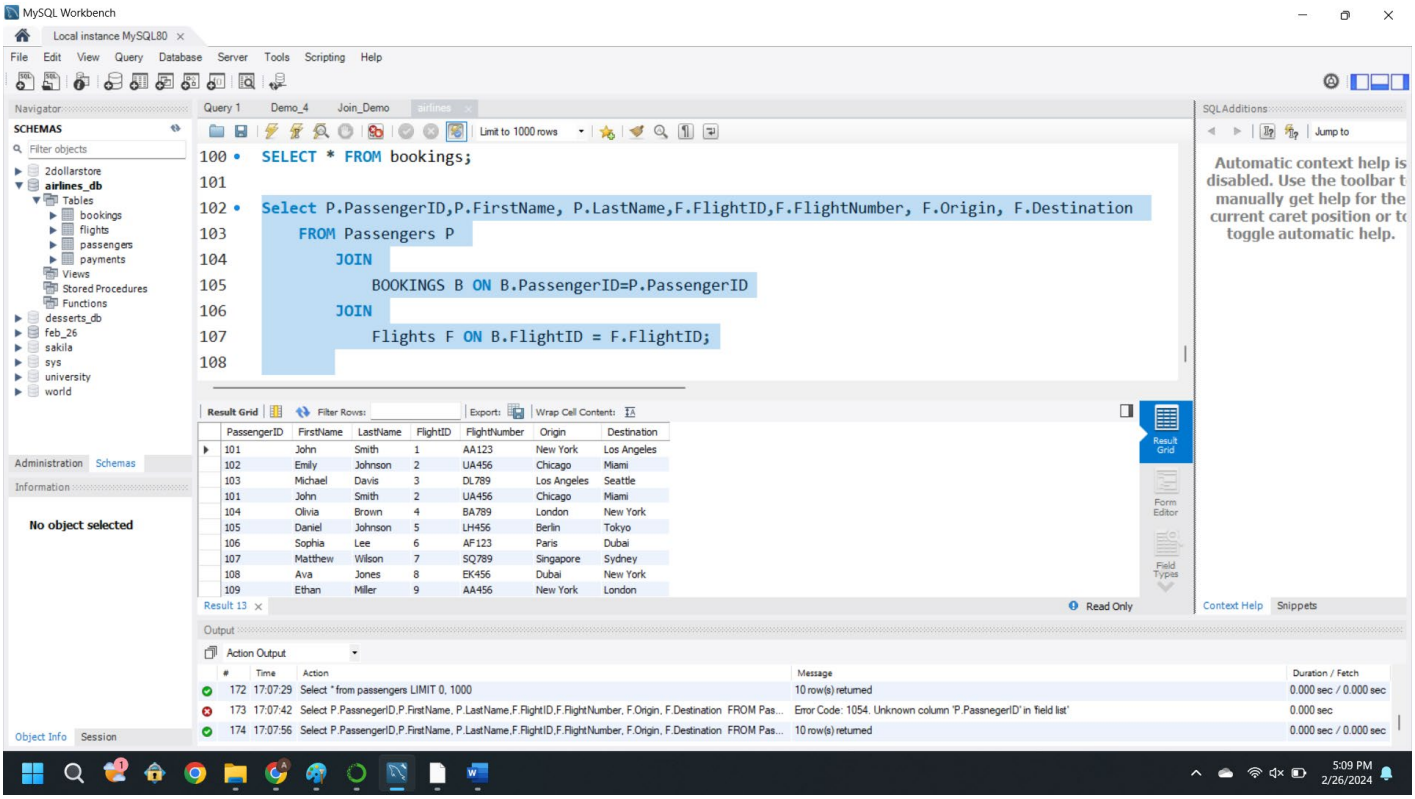
Question 1:

List all passengers and the flights they have booked.

Screenshot 1:



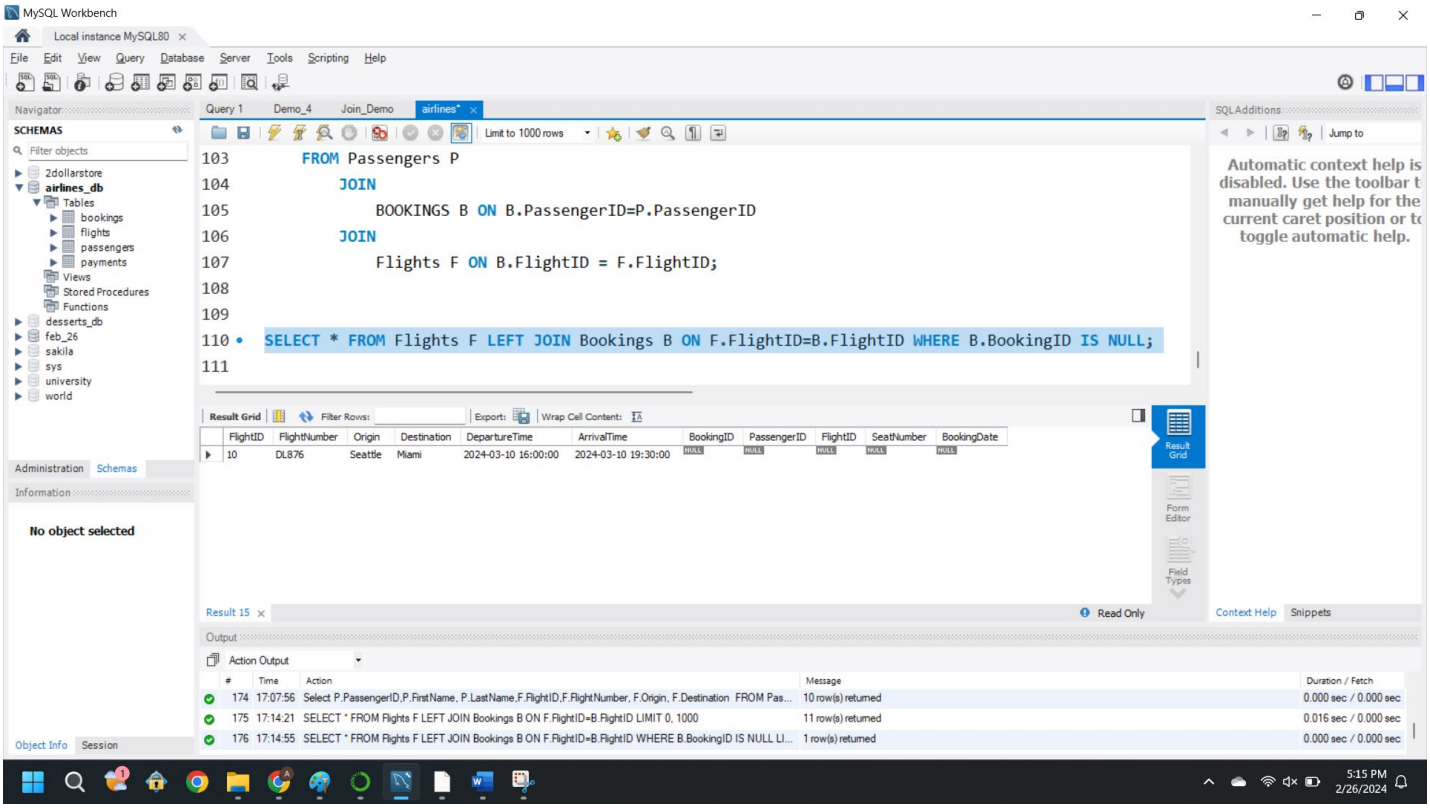
Query 1:

```
SELECT P.PassengerID,P.FirstName, P.LastName,F.FlightID,F.FlightNumber, F.Origin, F.Destination

FROM Passengers P

JOIN

BOOKINGS B ON B.PassengerID=P.PassengerID

JOIN

Flights F ON B.FlightID = F.FlightID;
```

Question 2:

Display the flights with no bookings made.

Screenshot 2:



Query 2:

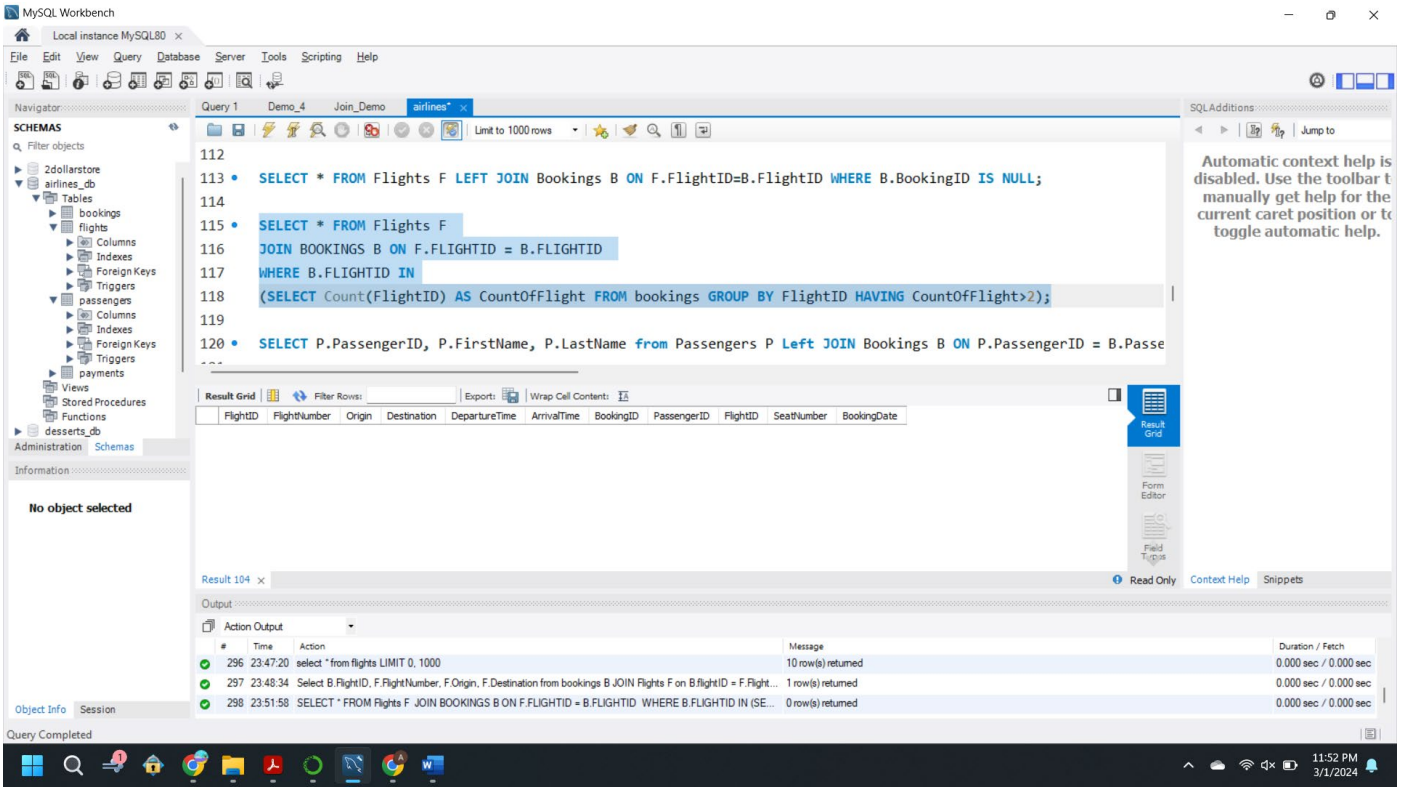
```
SELECT * FROM Flights F
LEFT JOIN
Bookings B ON F.FlightID=B.FlightID
WHERE
B.BookingID IS NULL;
```

Question 3:

Retrieve the flights with more than two bookings.

Screenshot 3:

No bookings were made more than 2 times for a flight.



Query 3:

SELECT * FROM Flights F

JOIN

BOOKINGS B ON F.FLIGHTID = B.FLIGHTID

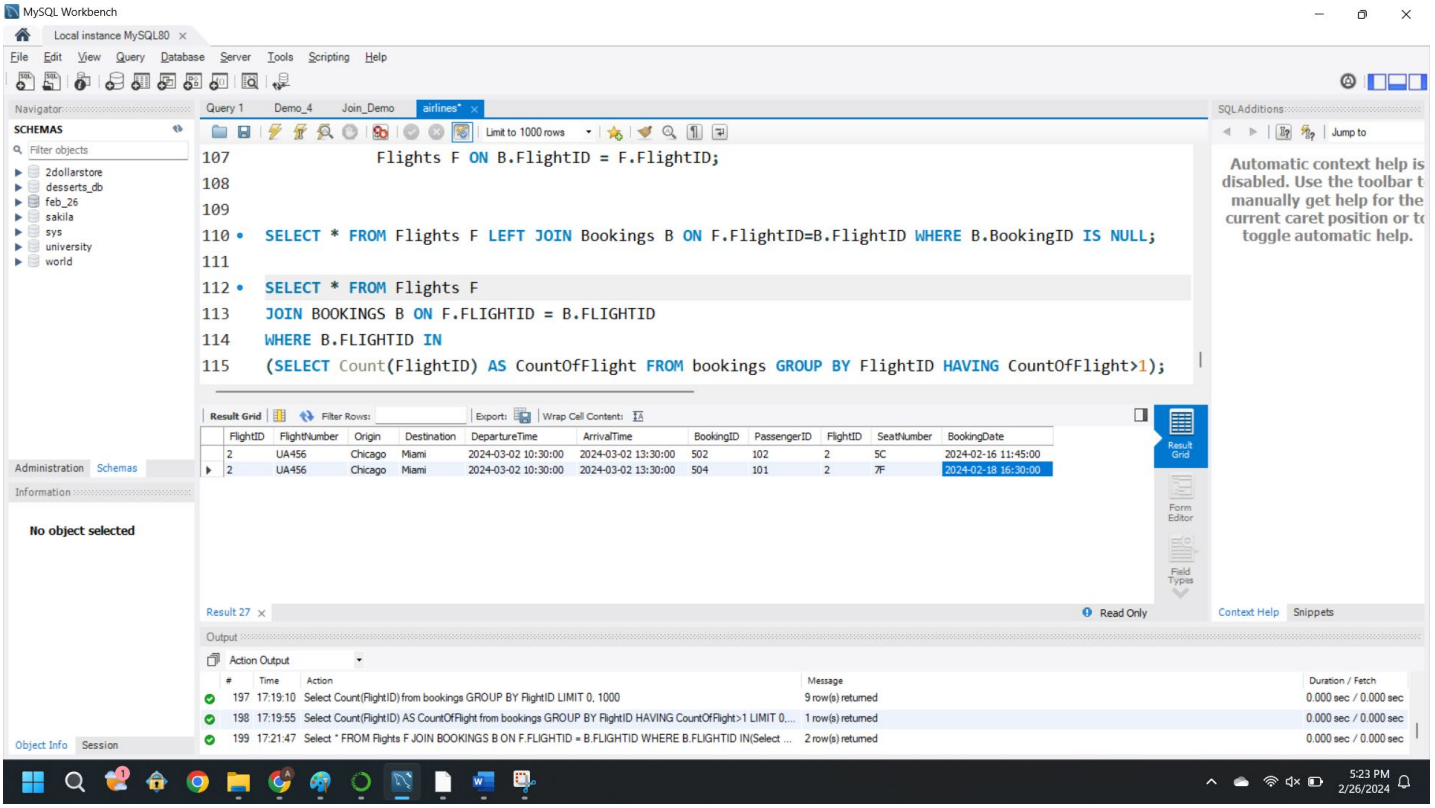
WHERE

B.FLIGHTID IN

(SELECT Count(FlightID) AS CountOfFlight FROM bookings GROUP BY FlightID HAVING CountOfFlight>2);

Question 3 – Instead of more than 2 flights, we can find more than 1 flight.

Screenshot 3:



Query 3:

SELECT * FROM Flights F

JOIN

BOOKINGS B ON F.FLIGHTID = B.FLIGHTID

WHERE

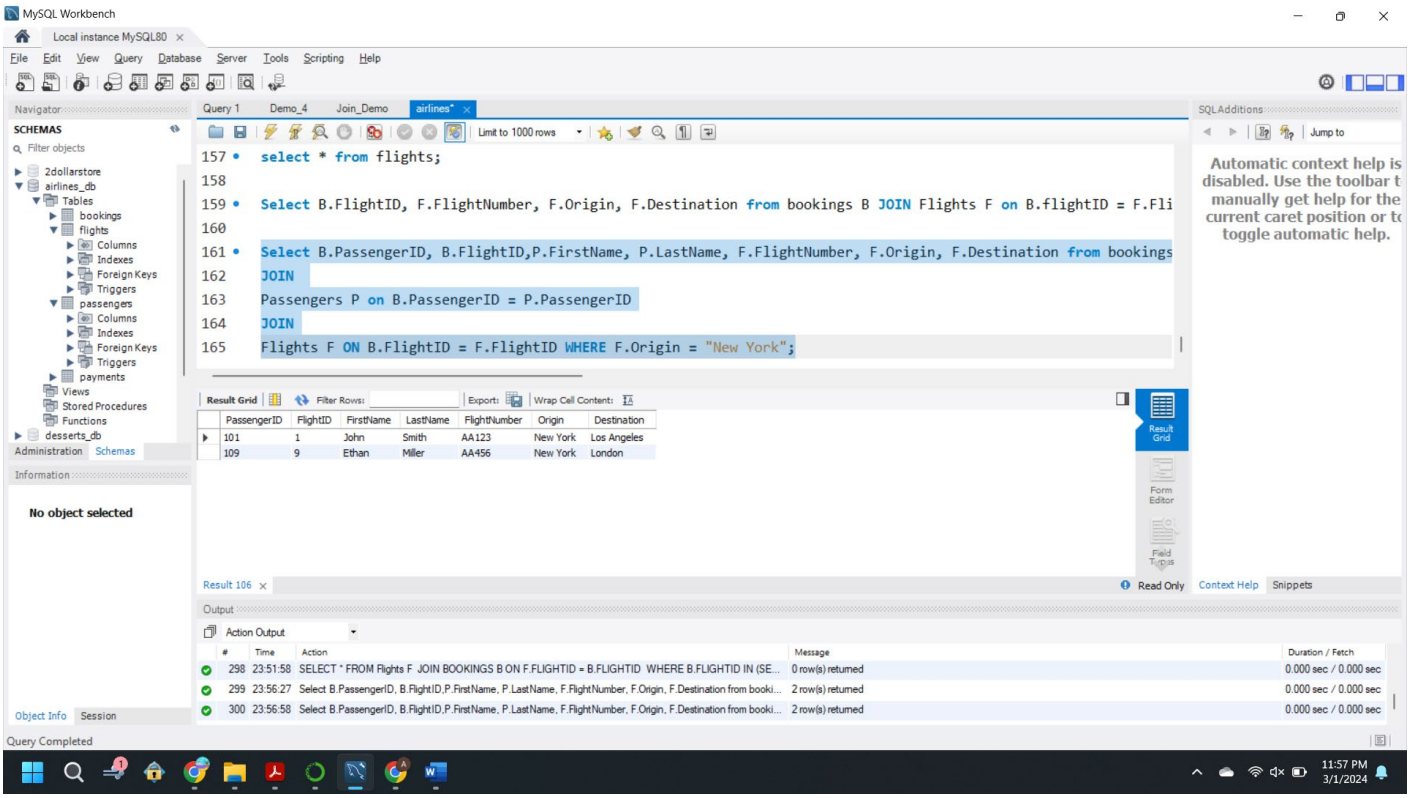
B.FLIGHTID IN

(SELECT Count(FlightID) AS CountOfFlight FROM bookings GROUP BY FlightID HAVING CountOfFlight>1);

Question 4:

List all passengers who booked a flight departing from New York.

Screenshot 4:



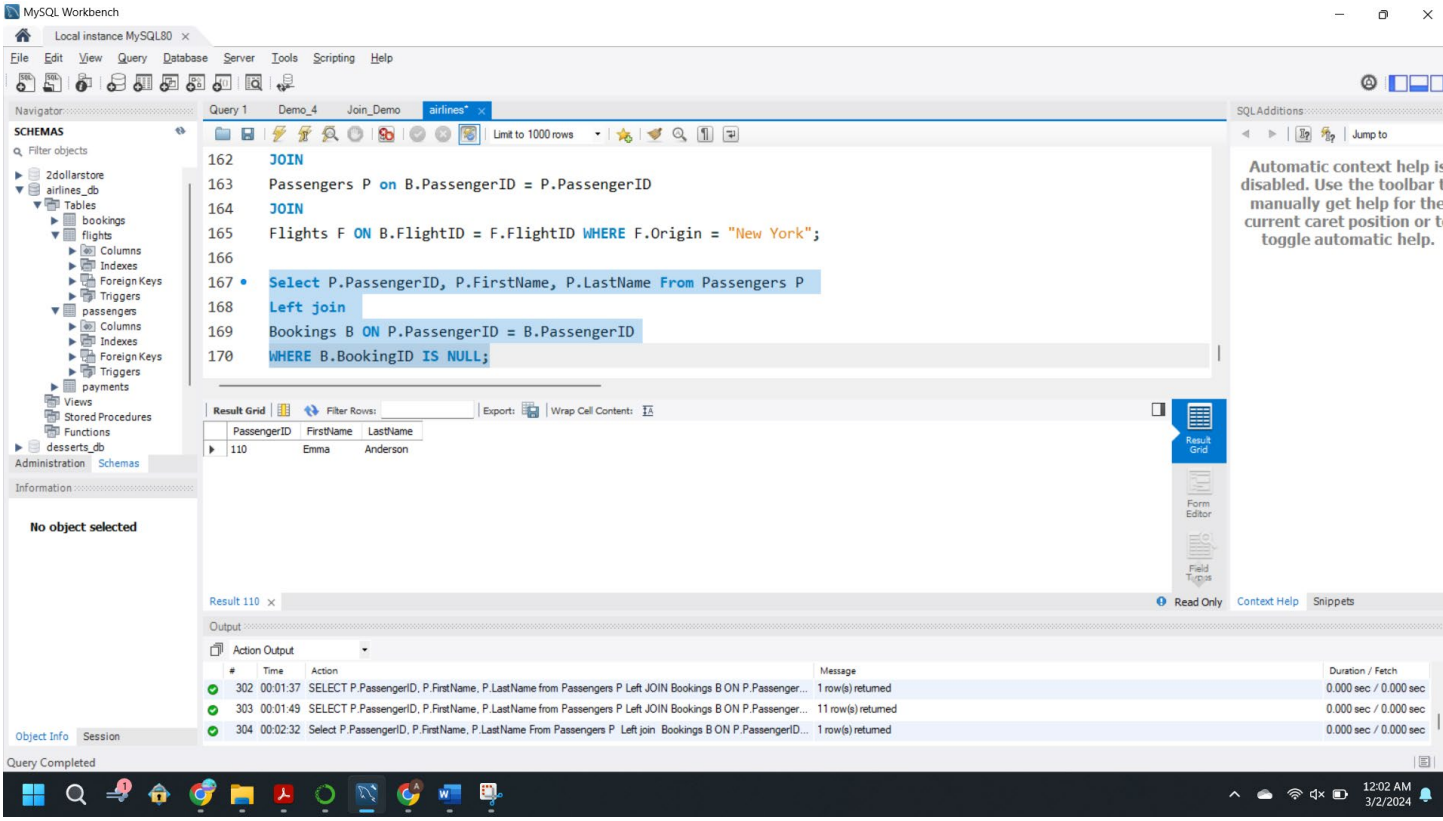
Query 4:

```
SELECT P.PassengerID,P.FirstName, P.LastName,F.FlightID,F.FlightNumber, F.Origin, F.Destination

FROM Passengers P

JOIN

BOOKINGS B ON B.PassengerID=P.PassengerID

JOIN

Flights F ON B.FlightID = F.FlightID

WHERE

F.Origin="New York";
```

Question 5:

Display the passengers who have not made any bookings.

Screenshot 5:



Query 5:

SELECT P.PassengerID, P.FirstName, P.LastName FROM Passengers P

LEFT JOIN

Bookings B ON P.PassengerID = B.PassengerID

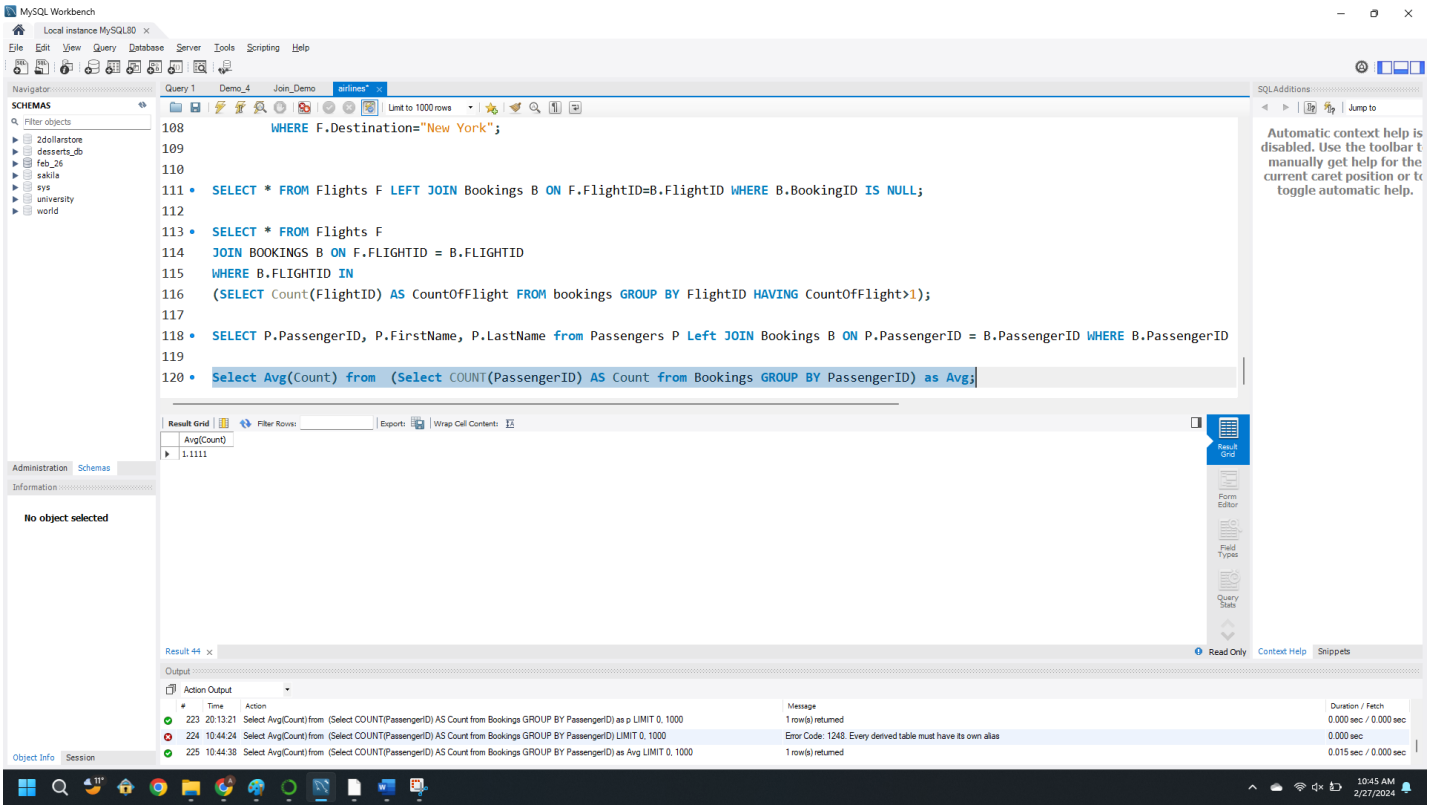
WHERE

B.BookingID IS NULL;

Question 6:

Retrieve the average number of bookings per passenger

Screenshot 6:



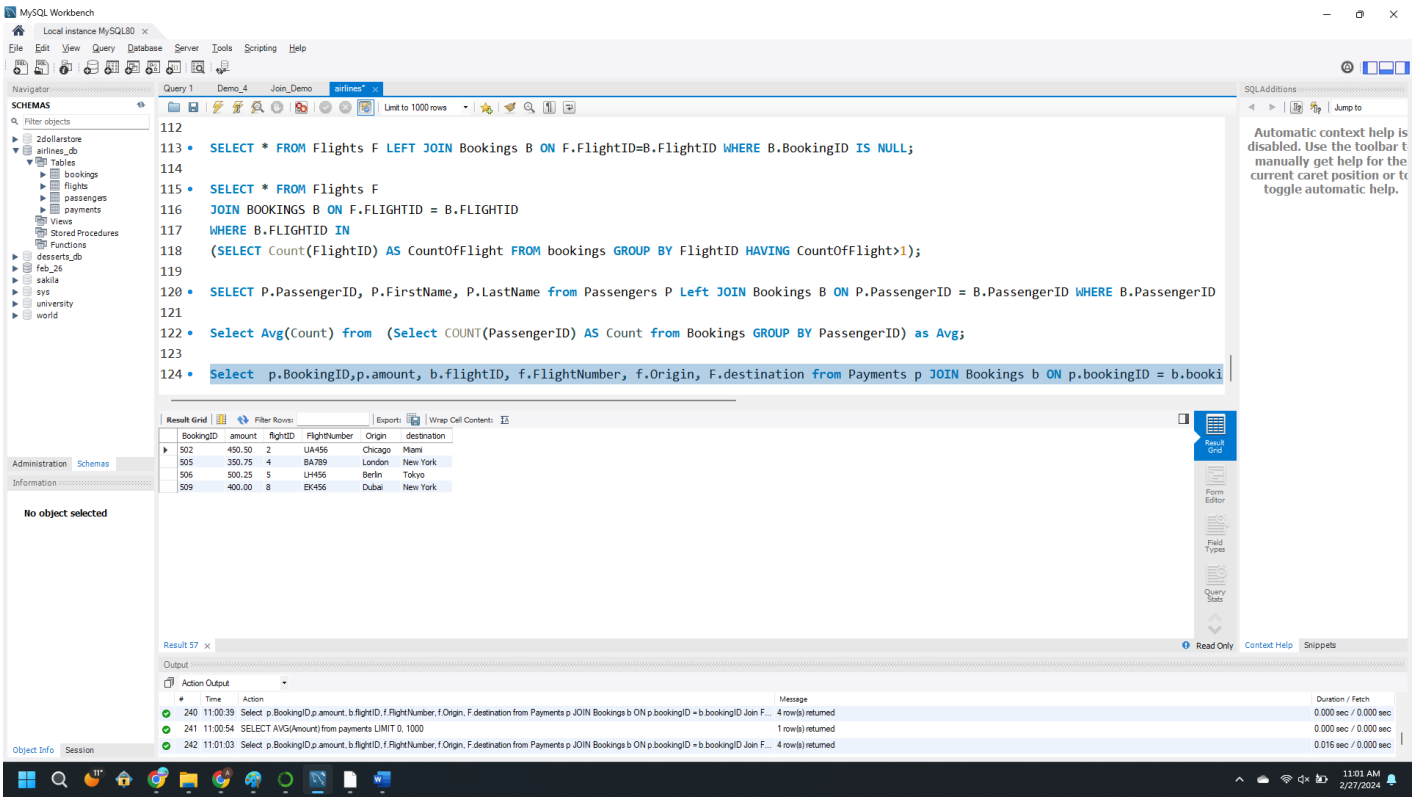
Query 6:

SELECT Avg(Count) FROM (SELECT COUNT(PassengerID) AS Count FROM Bookings GROUP BY PassengerID) as Avg;

Question 7:

List the flights where the total payment amount exceeds the average payment amount.

Screenshot 7:



Query 7:

SELECT p.BookingID,p.amount, b.flightID, f.FlightNumber, f.Origin, F.destination FROM Payments p

JOIN

Bookings b ON p.bookingID = b.bookingID

JOIN

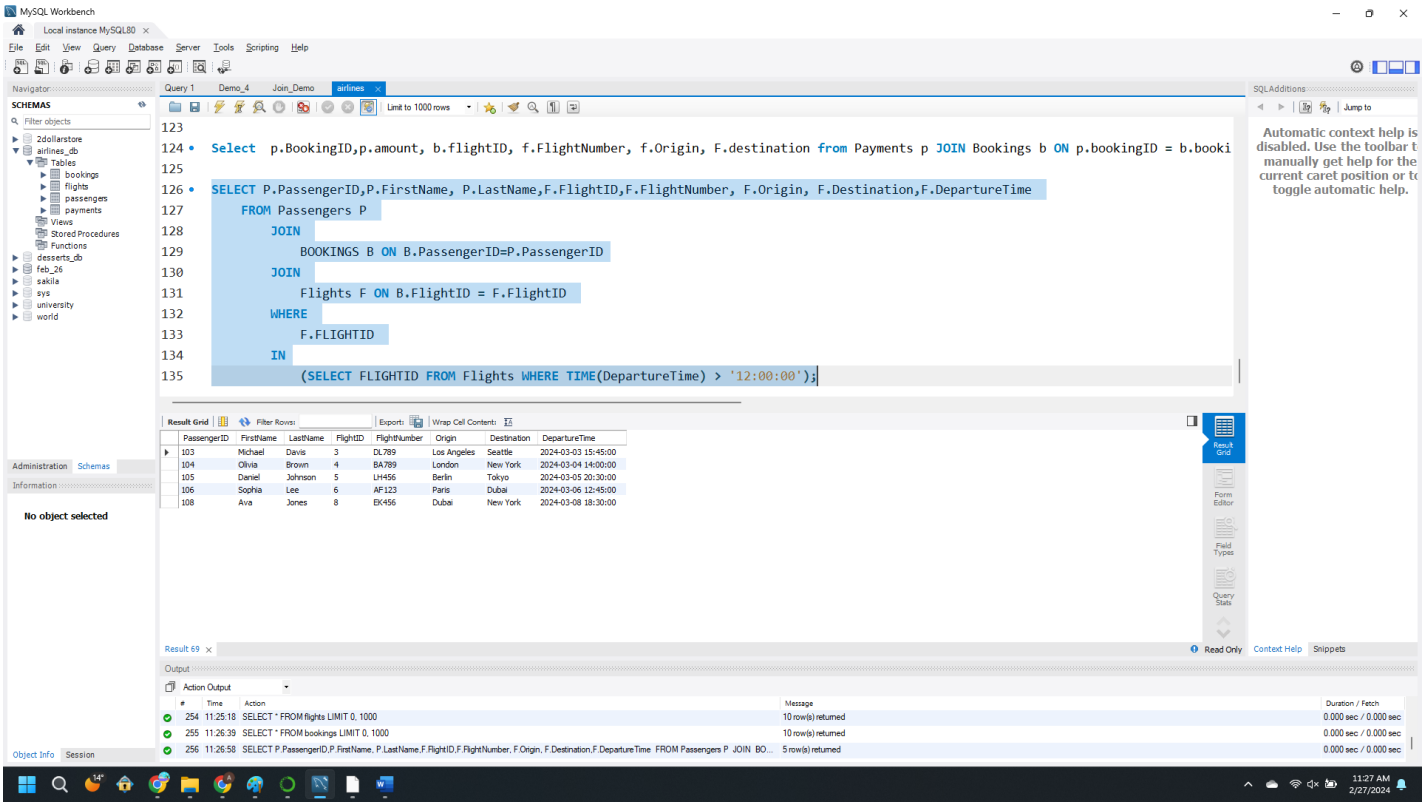
Flights f ON b.flightID = f.FlightID

WHERE p.amount>(SELECT AVG(Amount) FROM payments);

Question 8:

Retrieve the details of passengers who booked flights with a departure time later than 12:00 PM.

Screenshot 8:



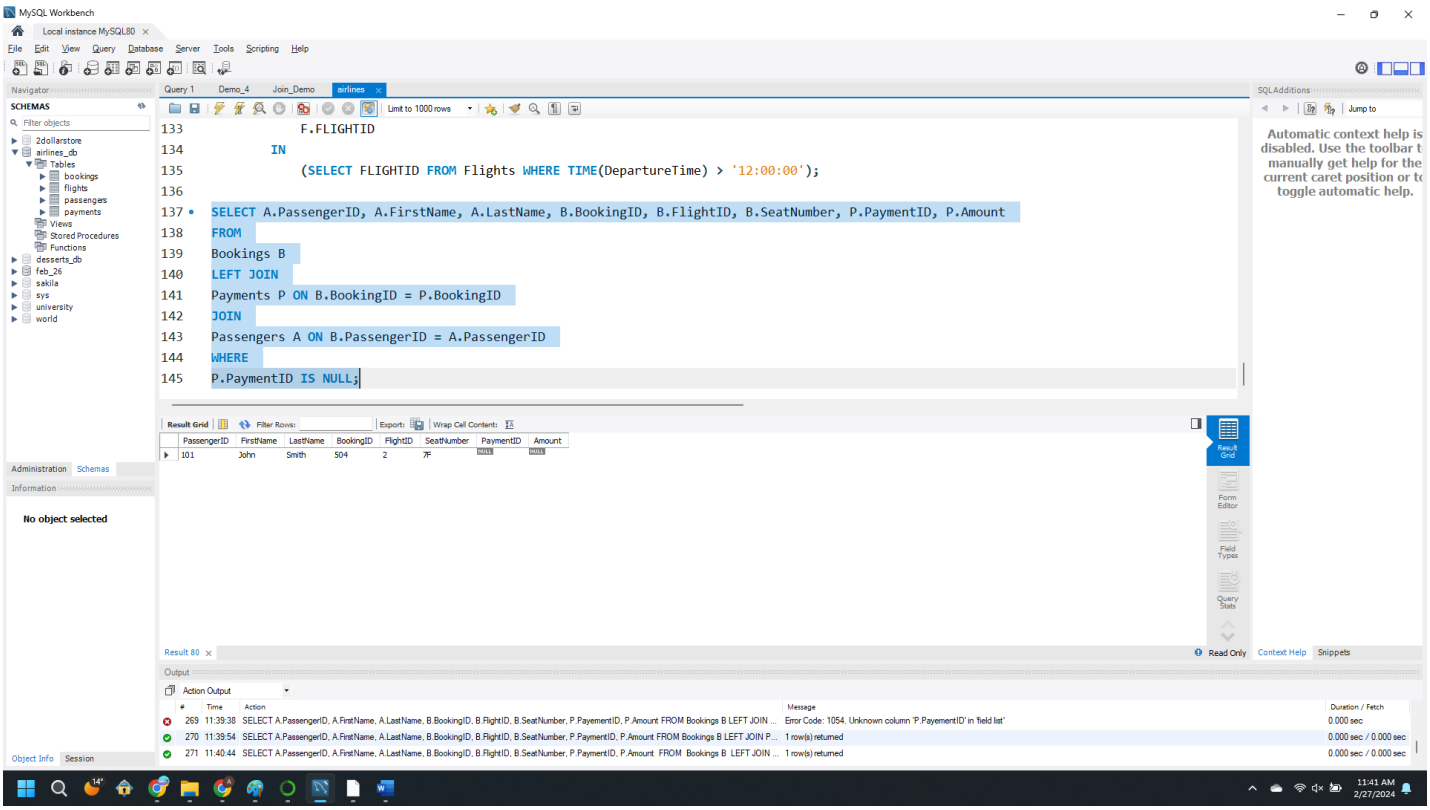
Query 8:

```
SELECT P.PassengerID,P.FirstName, P.LastName,F.FlightID,F.FlightNumber, F.Origin,
F.Destination,F.DepartureTime
FROM Passengers P
JOIN
BOOKINGS B ON B.PassengerID=P.PassengerID
JOIN
Flights F ON B.FlightID = F.FlightID
WHERE
F.FLIGHTID
IN
(SELECT FLIGHTID FROM Flights WHERE TIME(DepartureTime) > '12:00:00');
```

Question 9:

Find the passengers who booked a seat on a flight with no associated payment.

Screenshot 9:



Query 9:

SELECT A.PassengerID, A.FirstName, A.LastName, B.BookingID, B.FlightID, B.SeatNumber, P.PaymentID, P.Amount FROM

Bookings B

LEFT JOIN

Payments P ON B.BookingID = P.BookingID

JOIN

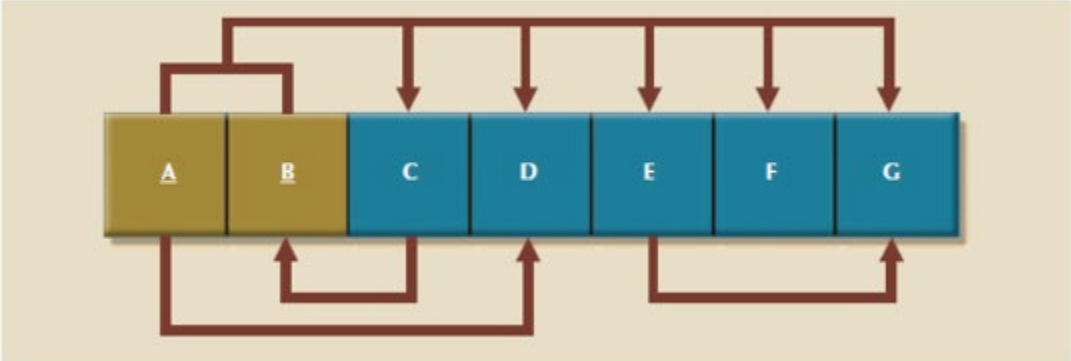
Passengers A ON B.PassengerID = A.PassengerID

WHERE

P.PaymentID IS NULL;

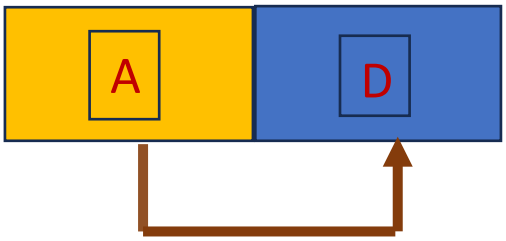
Question 10:

Break up the dependency diagram below to 3NF:

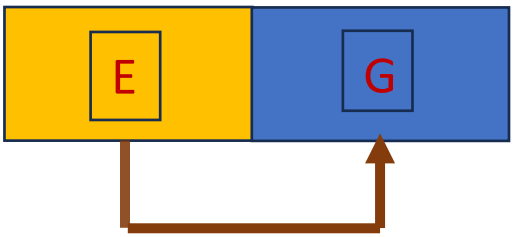


Answer:

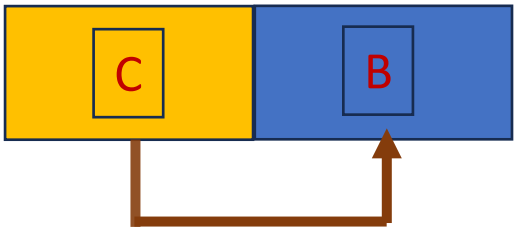
A and D has partial dependency, so they both form a new table.



E and G has transitive dependency, in 3NF transitive dependency should be removed, so E and G form a separate table.



C determines the value of the key attribute B. So C and B form a separate table.



A, C and E together can be used to determine F, so they all form a separate table.