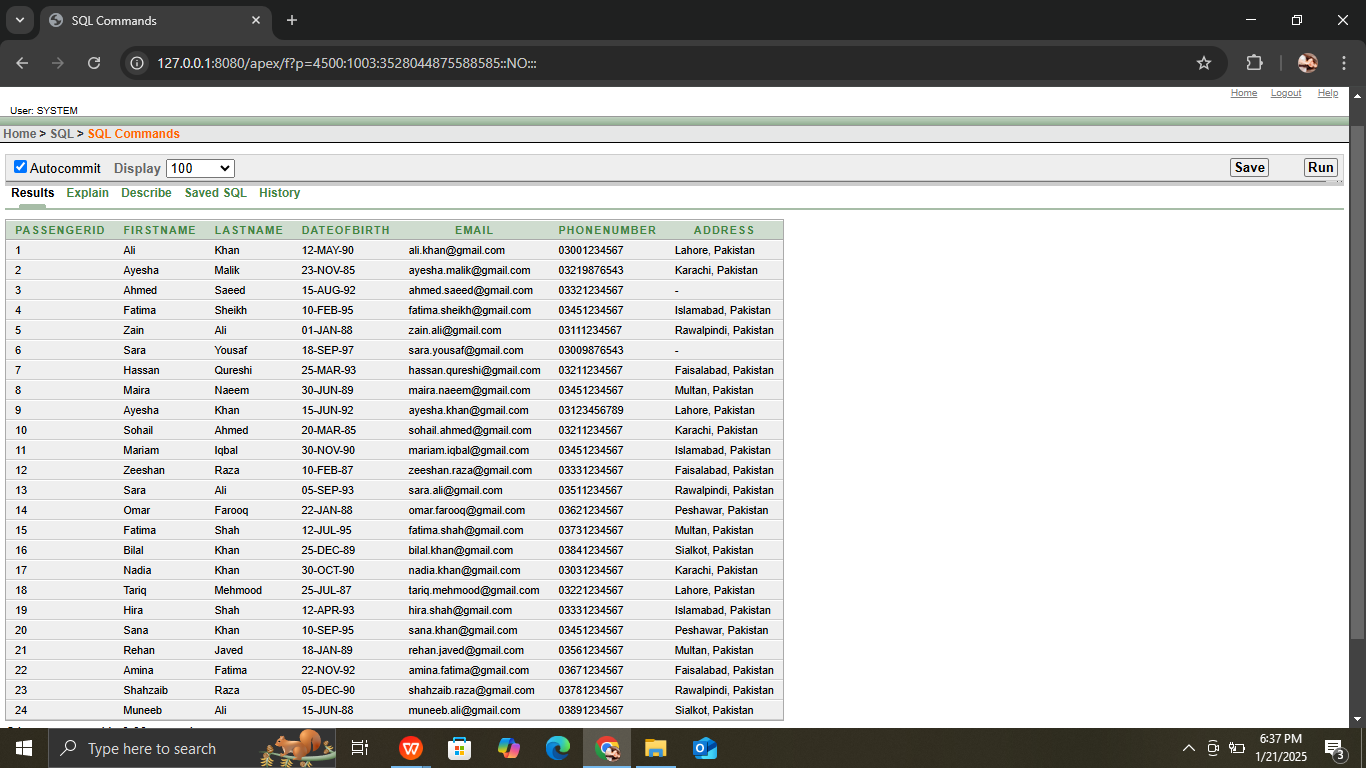
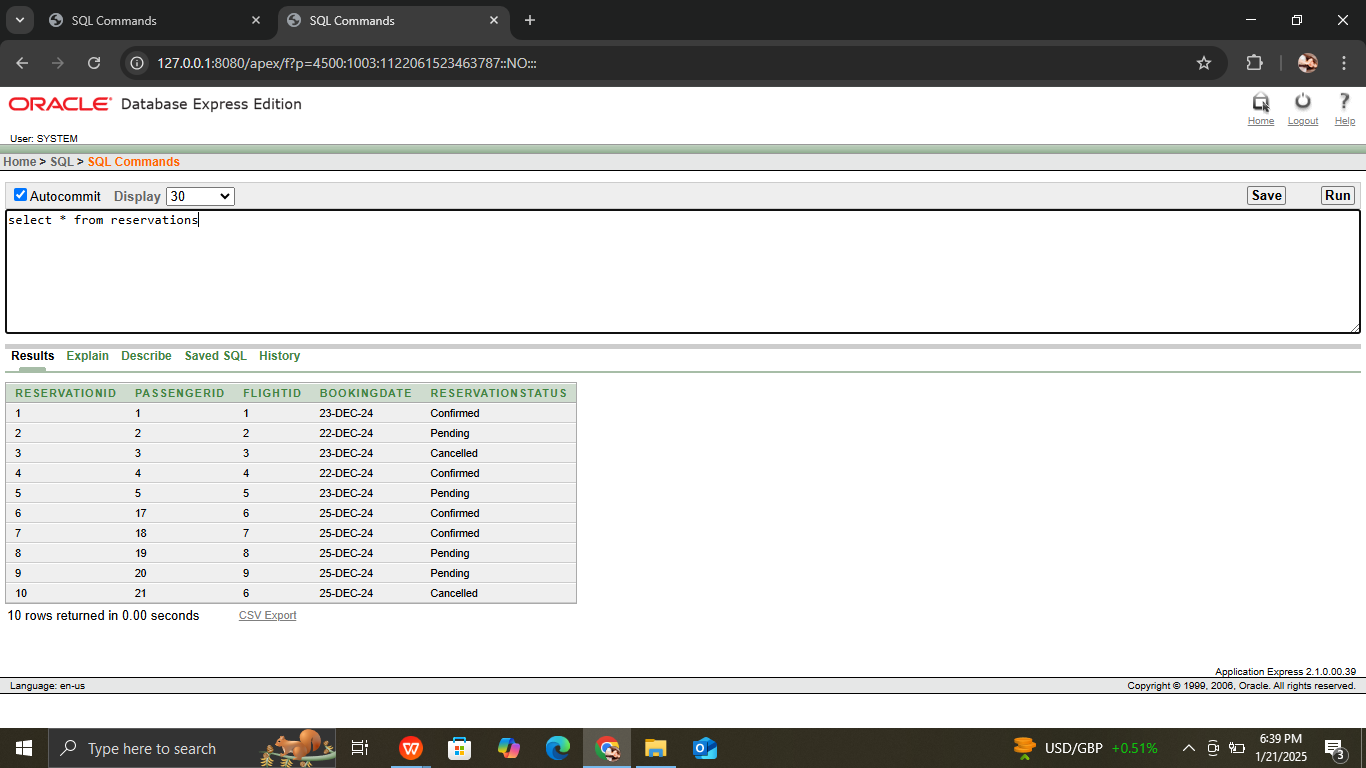
**Tables**

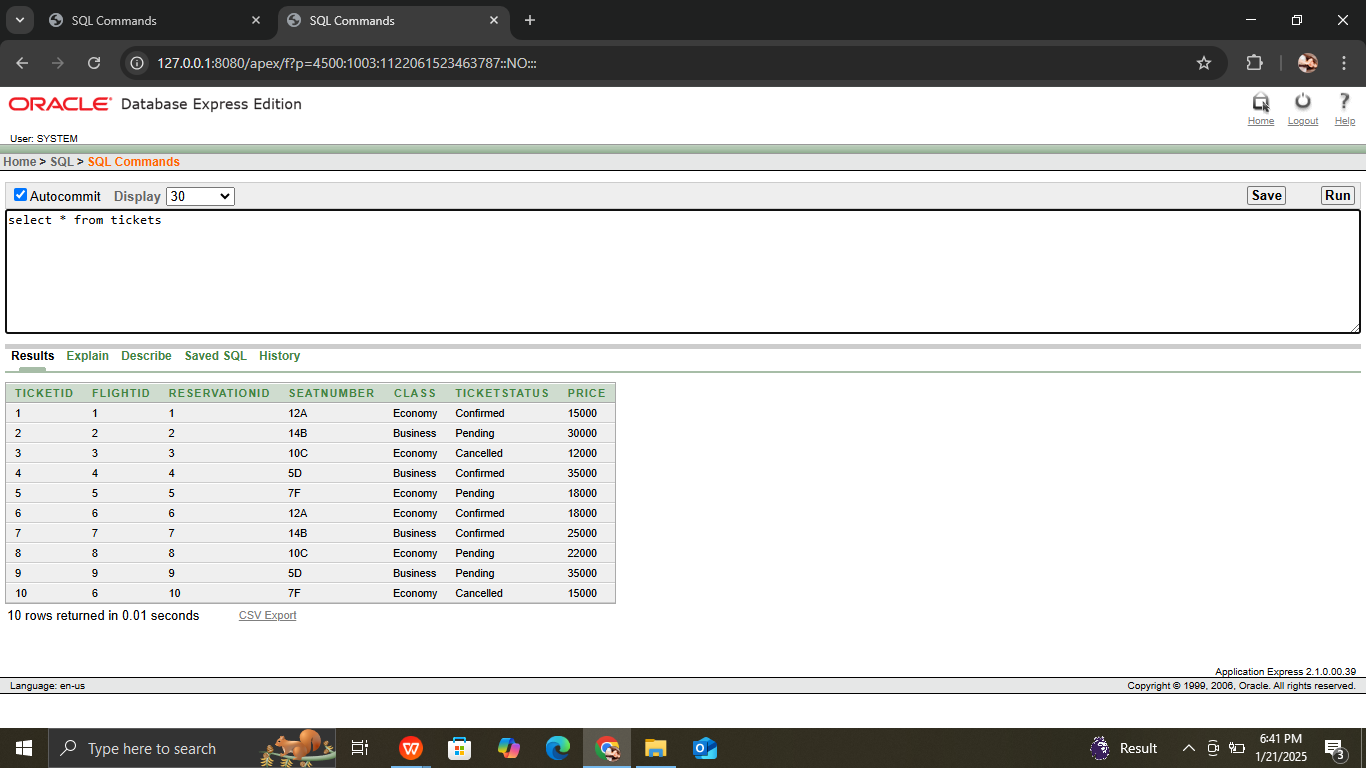
**Passenger table:**



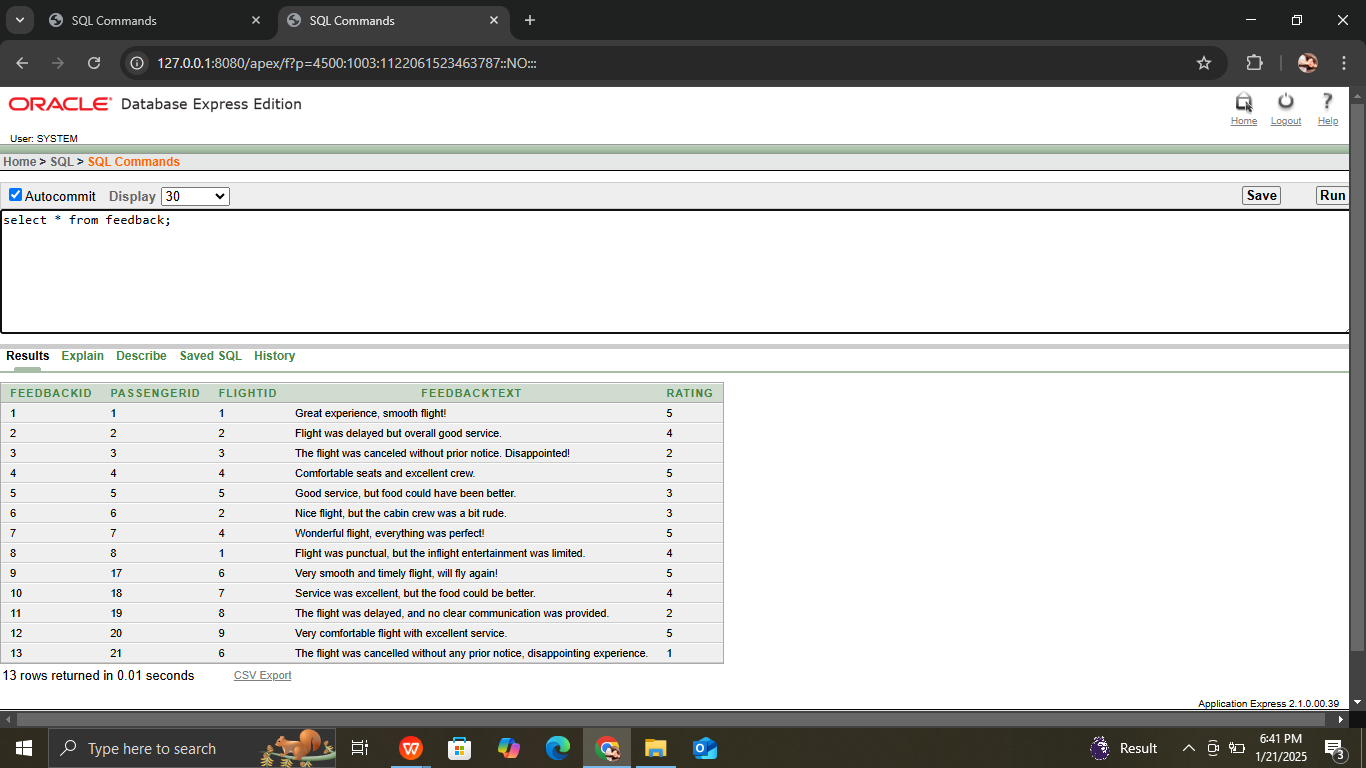
**Reservation table**



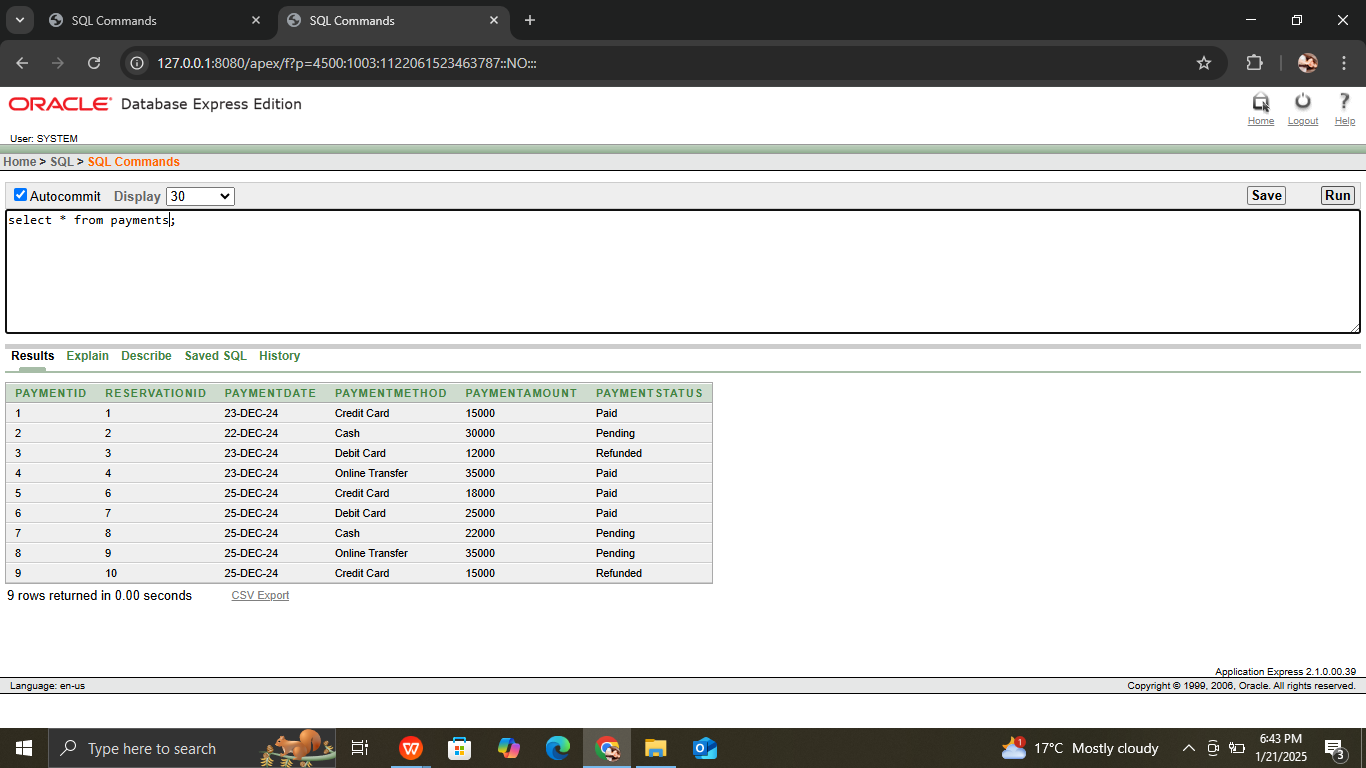
**Ticket table**



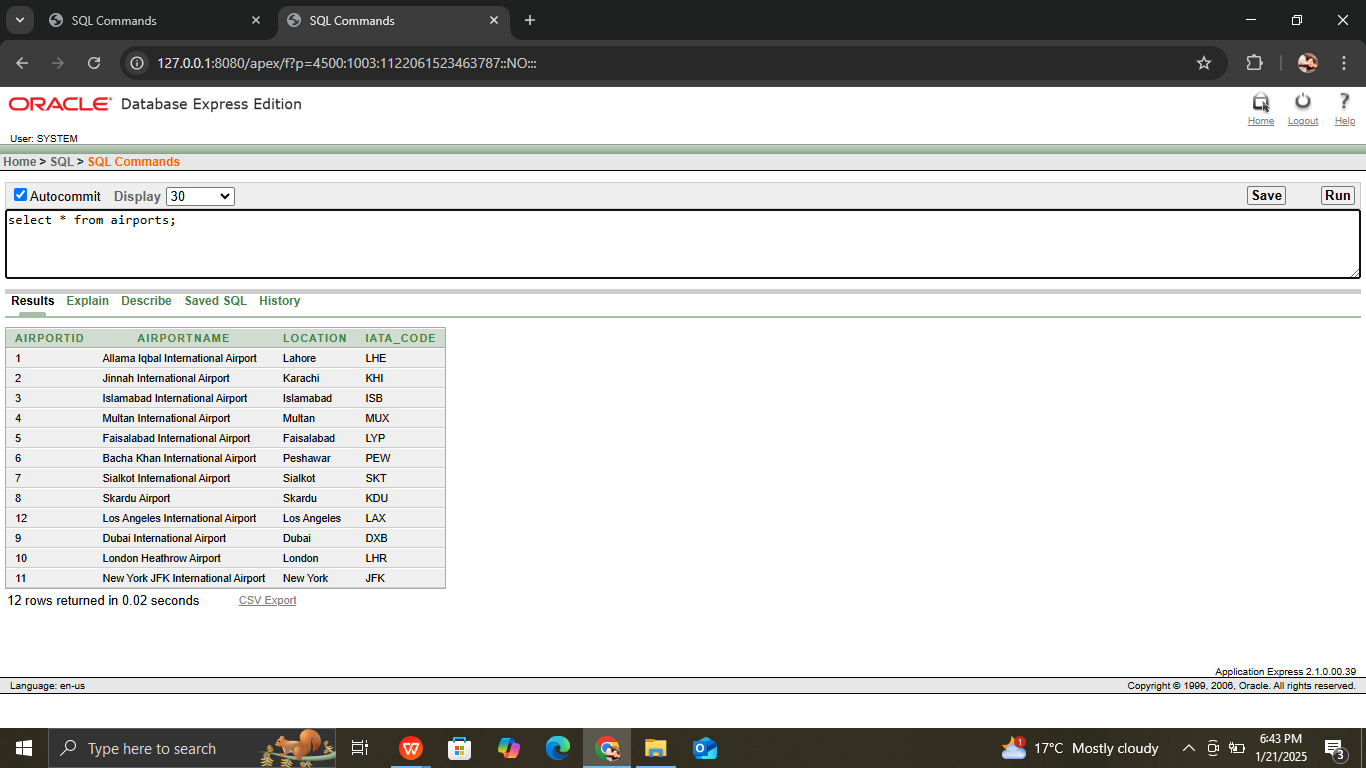
**Feedback table**



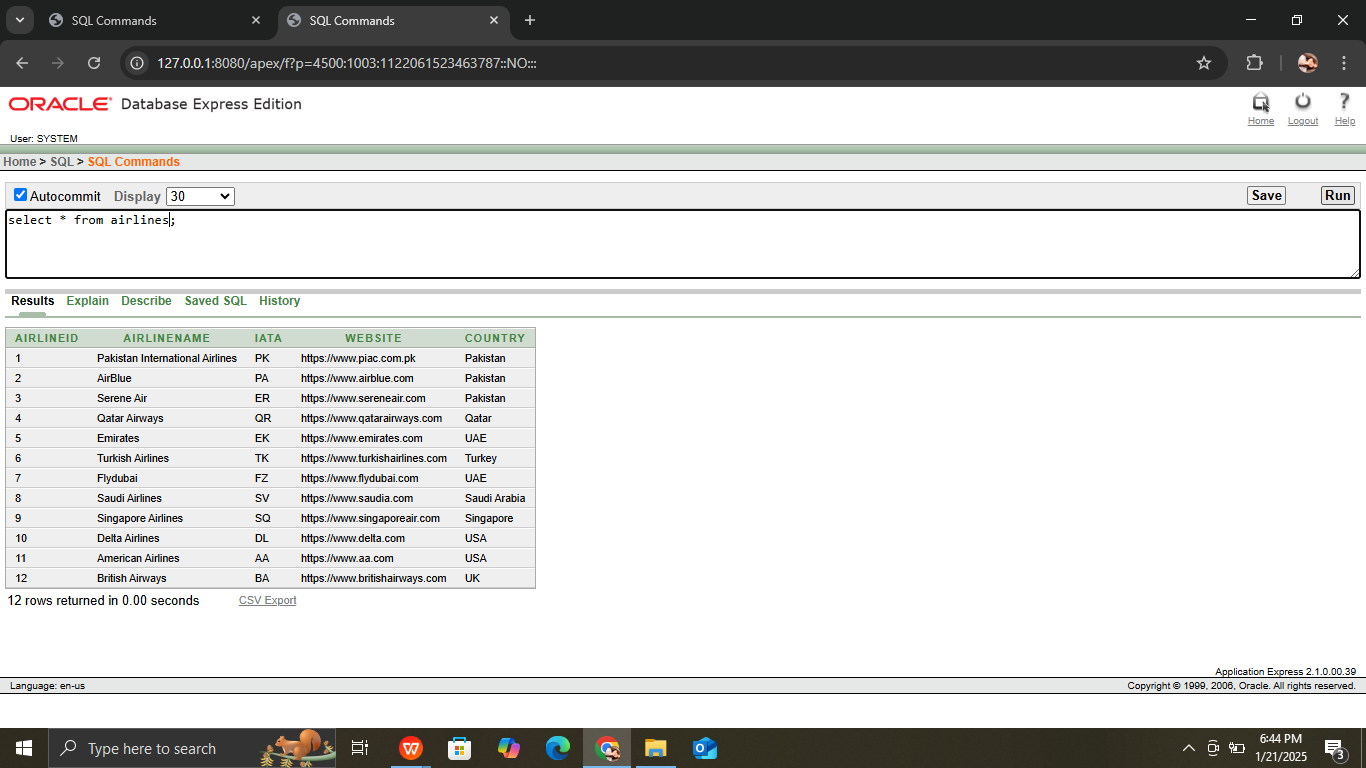
**Payment table:**



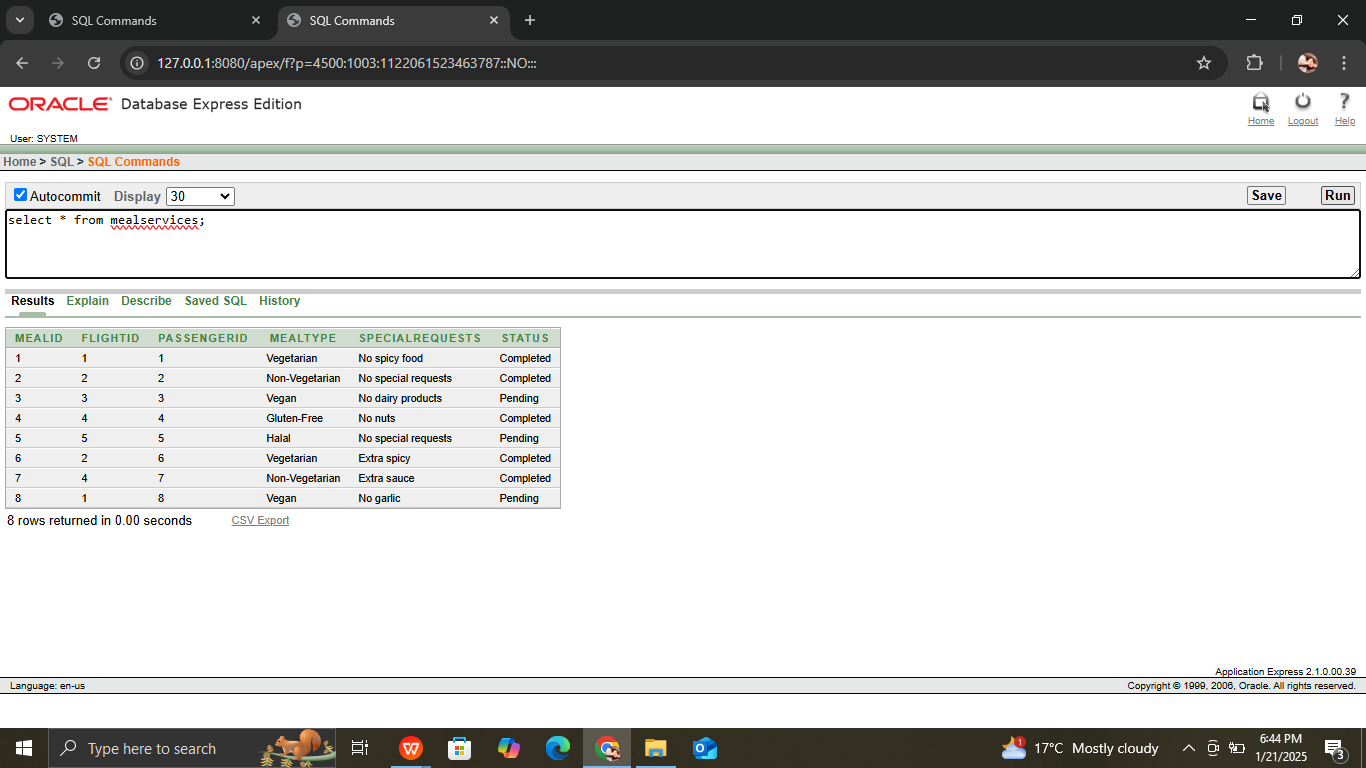
**Airport table**



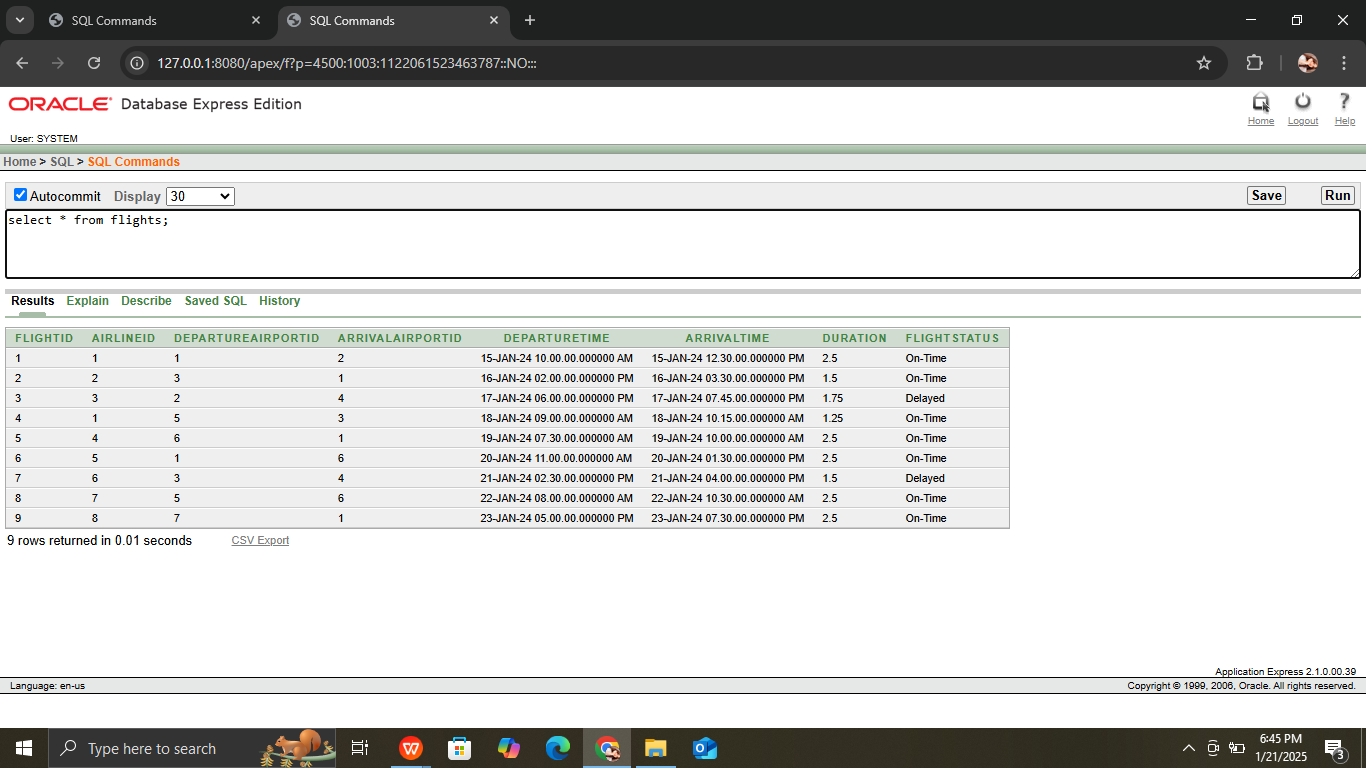
**Airlines table**



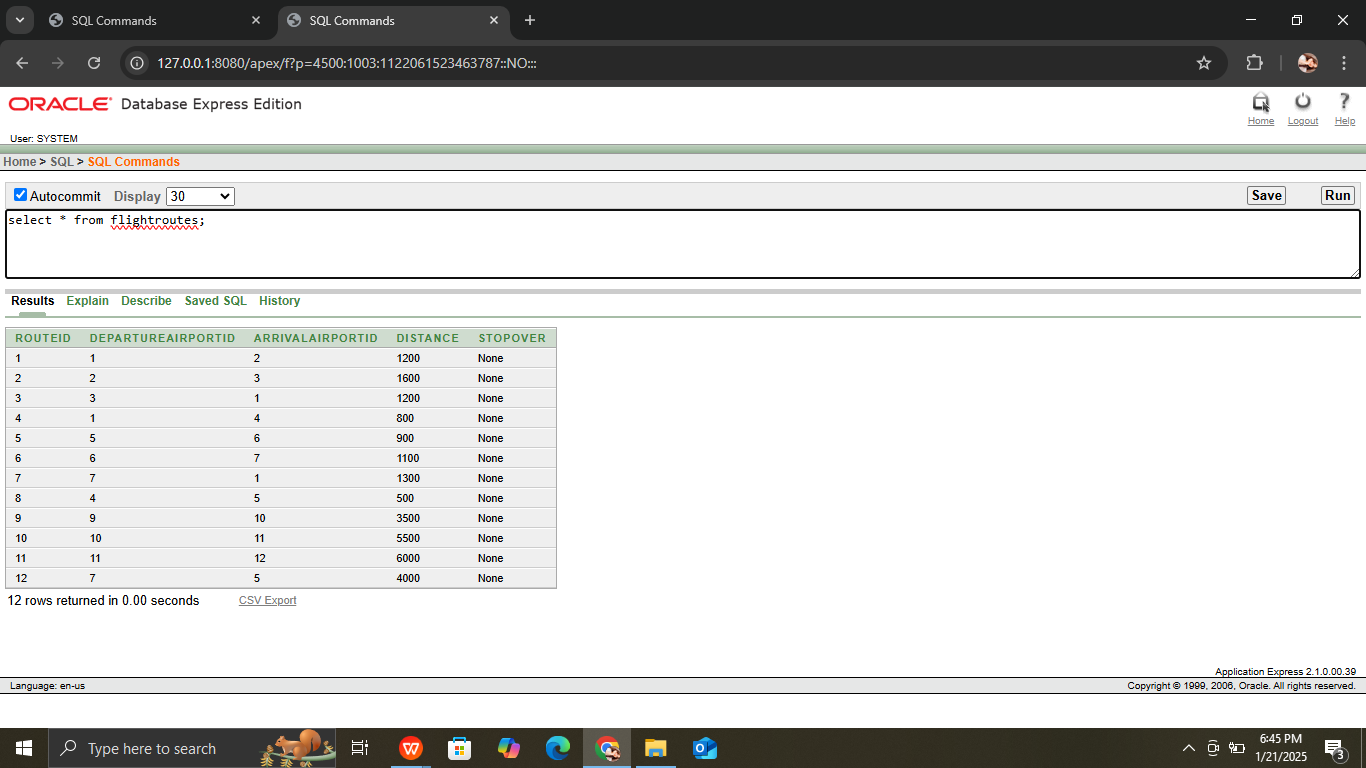
**Meal services:**



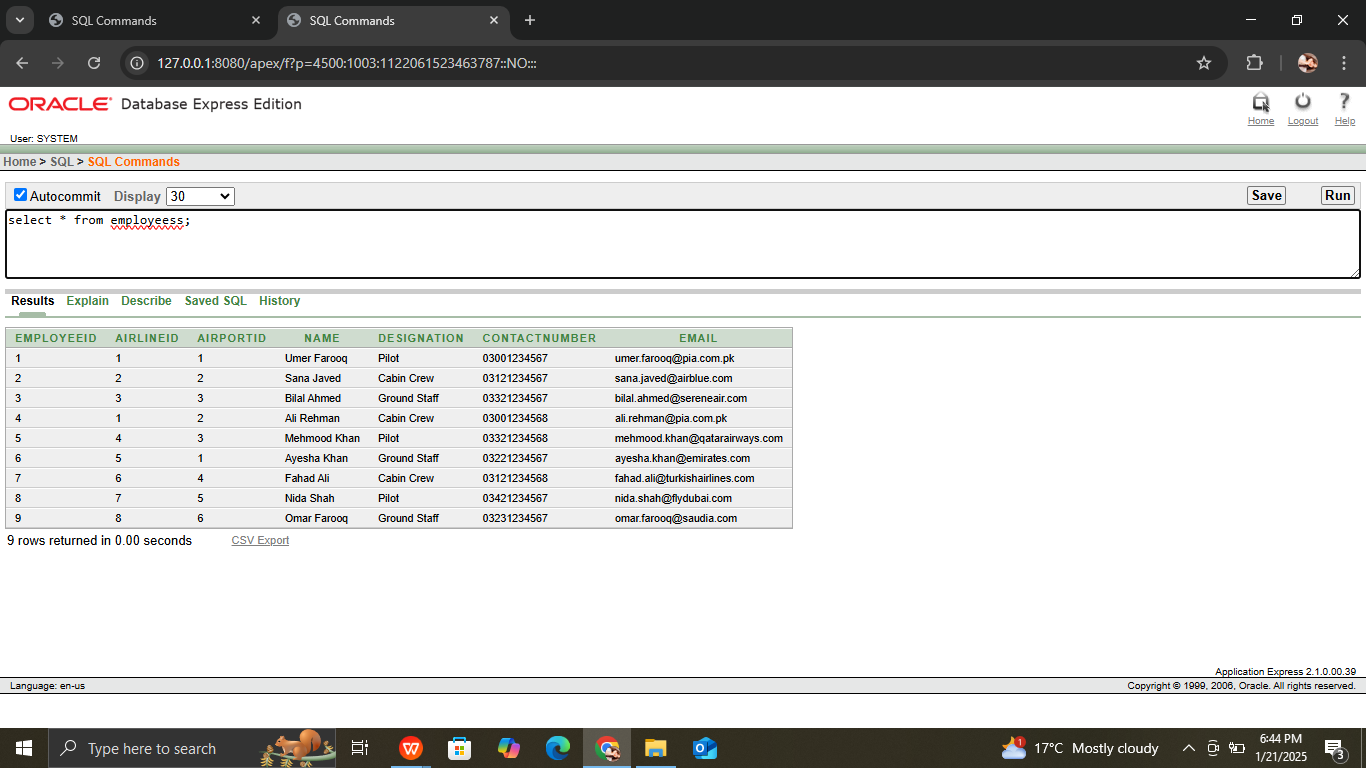
**Flights table:**



**Flight routes table**



**Employee table:**



**Luggage table:**



**QUERIES:**

**1. Join: List All Flights with Their Passengers and Reservations**

This query retrieves all flight details, along with the passenger information and reservation status for each flight.

SELECT

f.FlightID, f.DepartureTime, f.ArrivalTime, f.FlightStatus,

p.FirstName, p.LastName, p.Email,

r.ReservationStatus, r.BookingDate

FROM

Flights f

JOIN Reservations r ON f.FlightID = r.FlightID

JOIN Passengers p ON r.PassengerID = p.PassengerID;

A screenshot of a computer

Description automatically generated

**2. Join with Aggregate Function: Count the Number of Passengers per Flight**

This query counts the number of passengers who have reservations for each flight.

SELECT

f.FlightID,

f.DepartureTime,

f.ArrivalTime,

COUNT(r.ReservationID) AS NumberOfPassengers

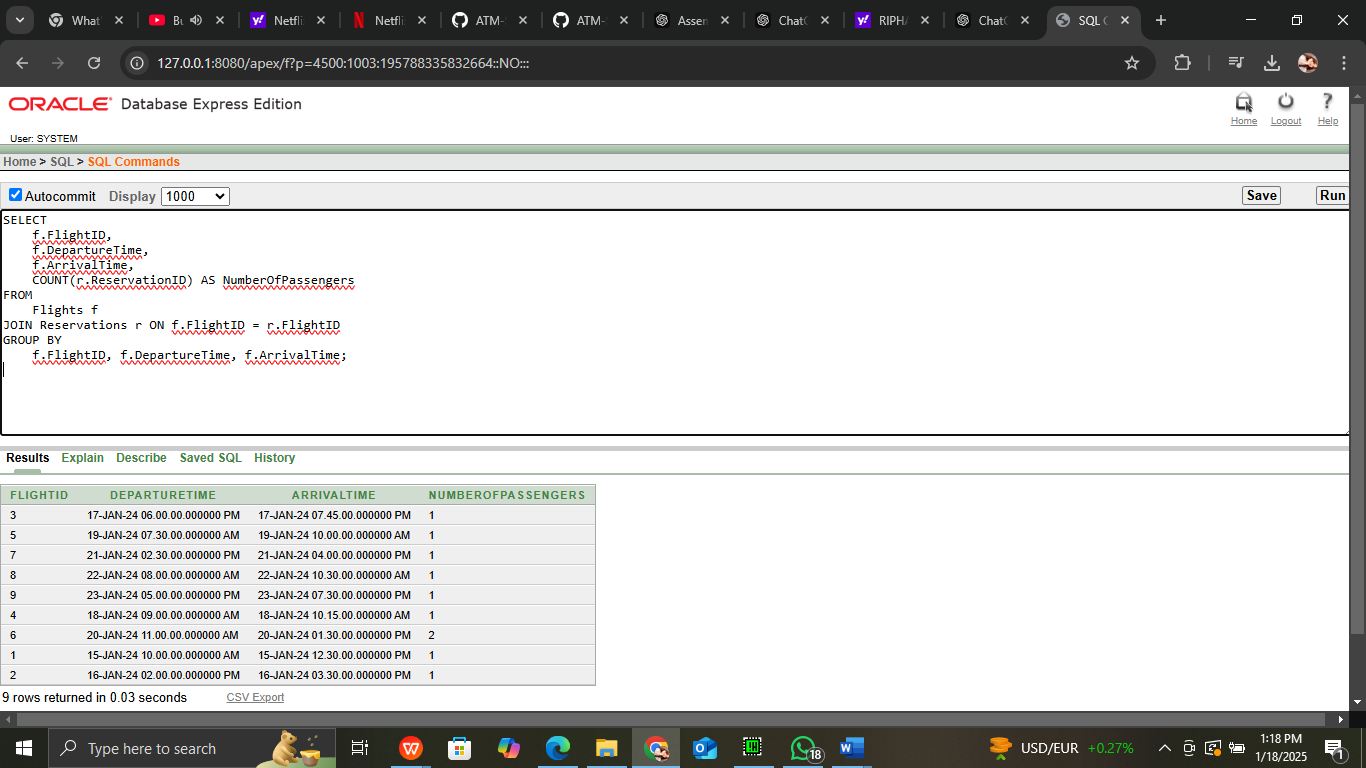
FROM

Flights f

JOIN Reservations r ON f.FlightID = r.FlightID

GROUP BY

f.FlightID, f.DepartureTime, f.ArrivalTime;



**3. Join with Aggregate Function and Group By: Total Revenue per Flight**

This query calculates the total revenue for each flight based on the ticket prices, grouped by flight.

SELECT

f.FlightID,

f.DepartureTime,

f.ArrivalTime,

SUM(t.Price) AS TotalRevenue

FROM

Flights f

JOIN Tickets t ON f.FlightID = t.FlightID

WHERE

t.TicketStatus = 'Confirmed'

GROUP BY

f.FlightID, f.DepartureTime, f.ArrivalTime;

A screenshot of a computer

Description automatically generated

**4. Subquery with IN: Find Passengers Who Have Reserved Flights with a Specific Airline**

This query finds passengers who have reserved flights operated by a specific airline.

SELECT

p.PassengerID, p.FirstName, p.LastName

FROM

Passengers p

WHERE

p.PassengerID IN (

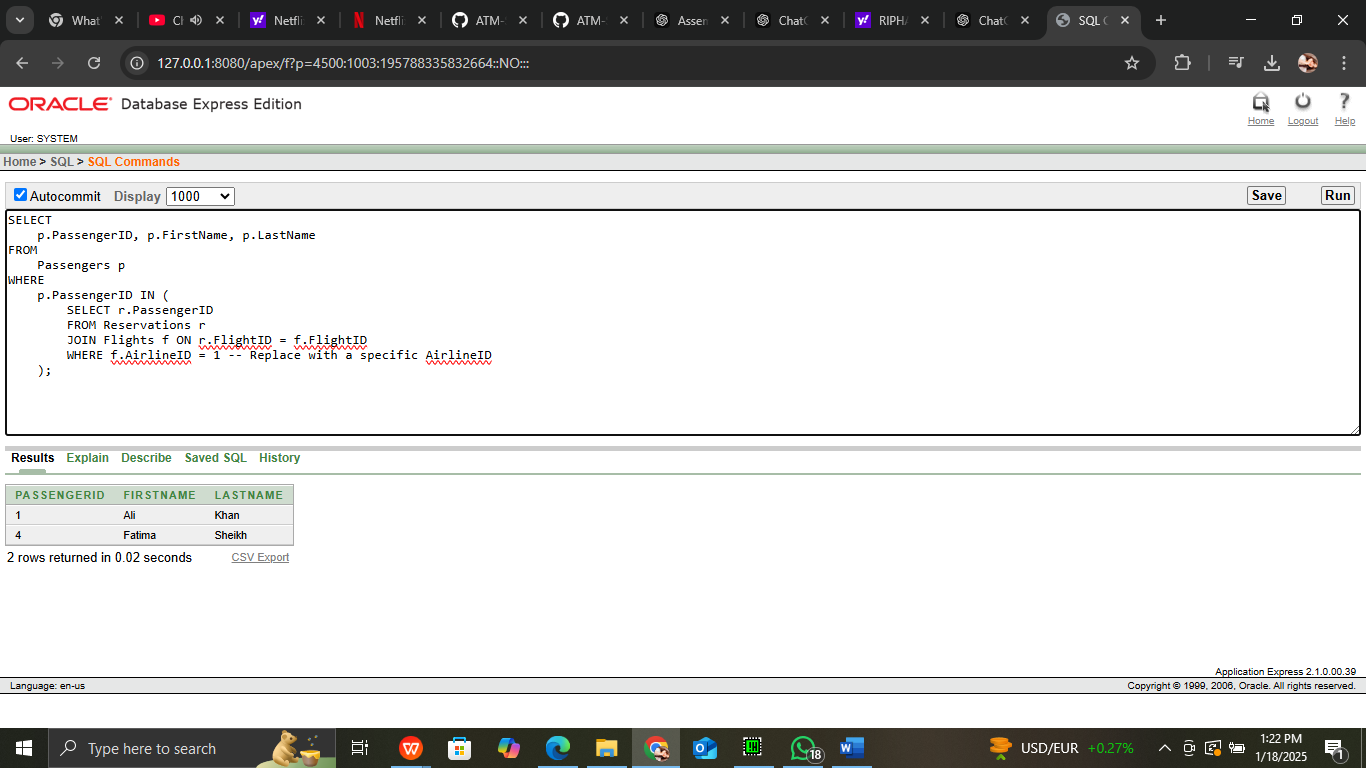
SELECT r.PassengerID

FROM Reservations r

JOIN Flights f ON r.FlightID = f.FlightID

WHERE f.AirlineID = 1 -- Replace with a specific AirlineID

);



**5. Join with Multiple Tables: List of Flights, Passengers, and Meal Services**

This query provides a list of all flights, along with passenger details and the meals requested for each flight.

SELECT

f.FlightID,

f.DepartureTime,

f.ArrivalTime,

p.FirstName,

p.LastName,

ms.MealType,

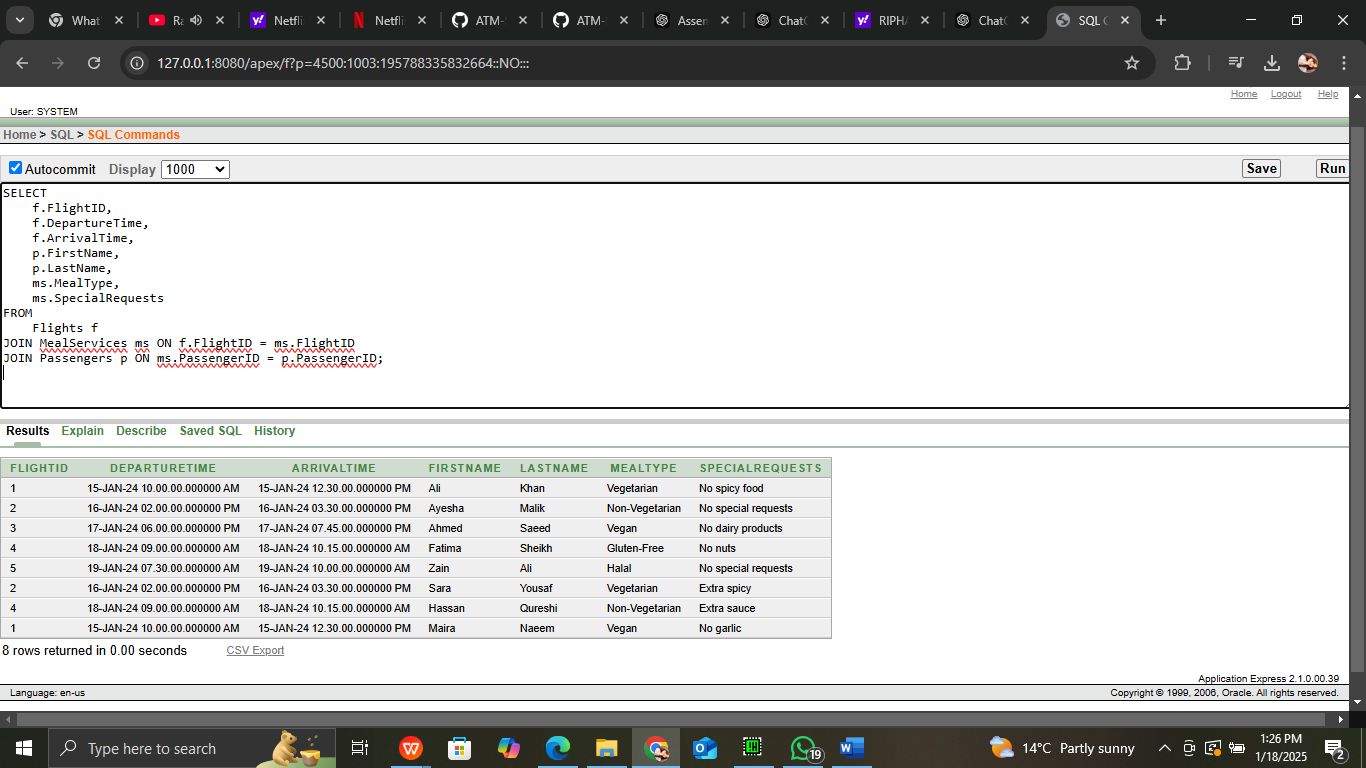
ms.SpecialRequests

FROM

Flights f

JOIN MealServices ms ON f.FlightID = ms.FlightID

JOIN Passengers p ON ms.PassengerID = p.PassengerID;



**6. Join with Payment Information: Flights and Payment Status**

This query shows flight details along with the payment status and payment amount for each reservation.

SELECT

f.FlightID,

f.DepartureTime,

f.ArrivalTime,

p.PaymentStatus,

p.PaymentAmount

FROM

Flights f

JOIN Reservations r ON f.FlightID = r.FlightID

JOIN Payments p ON r.ReservationID = p.ReservationID;

A screenshot of a computer

Description automatically generated

**7. Subquery with a Correlated Join: Passengers Who Have Luggage**

This query finds passengers who have checked luggage for their flights.

SELECT

p.PassengerID,

p.FirstName,

p.LastName

FROM

Passengers p

WHERE

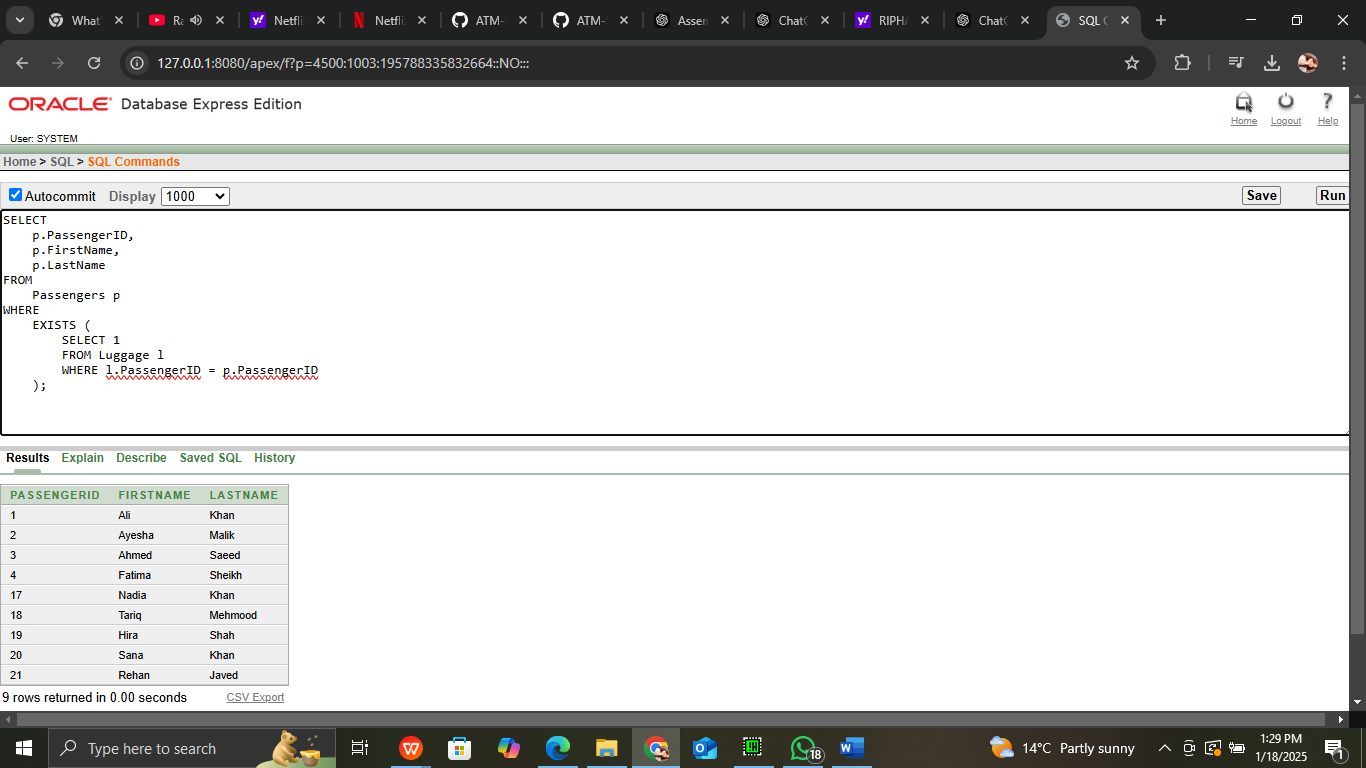
EXISTS (

SELECT 1

FROM Luggage l

WHERE l.PassengerID = p.PassengerID

);



**8. Nested Subquery: Find Flights with Reservations**

This query identifies flights that have been reserved by any passengers.

SELECT

f.FlightID,

f.DepartureTime,

f.ArrivalTime

FROM

Flights f

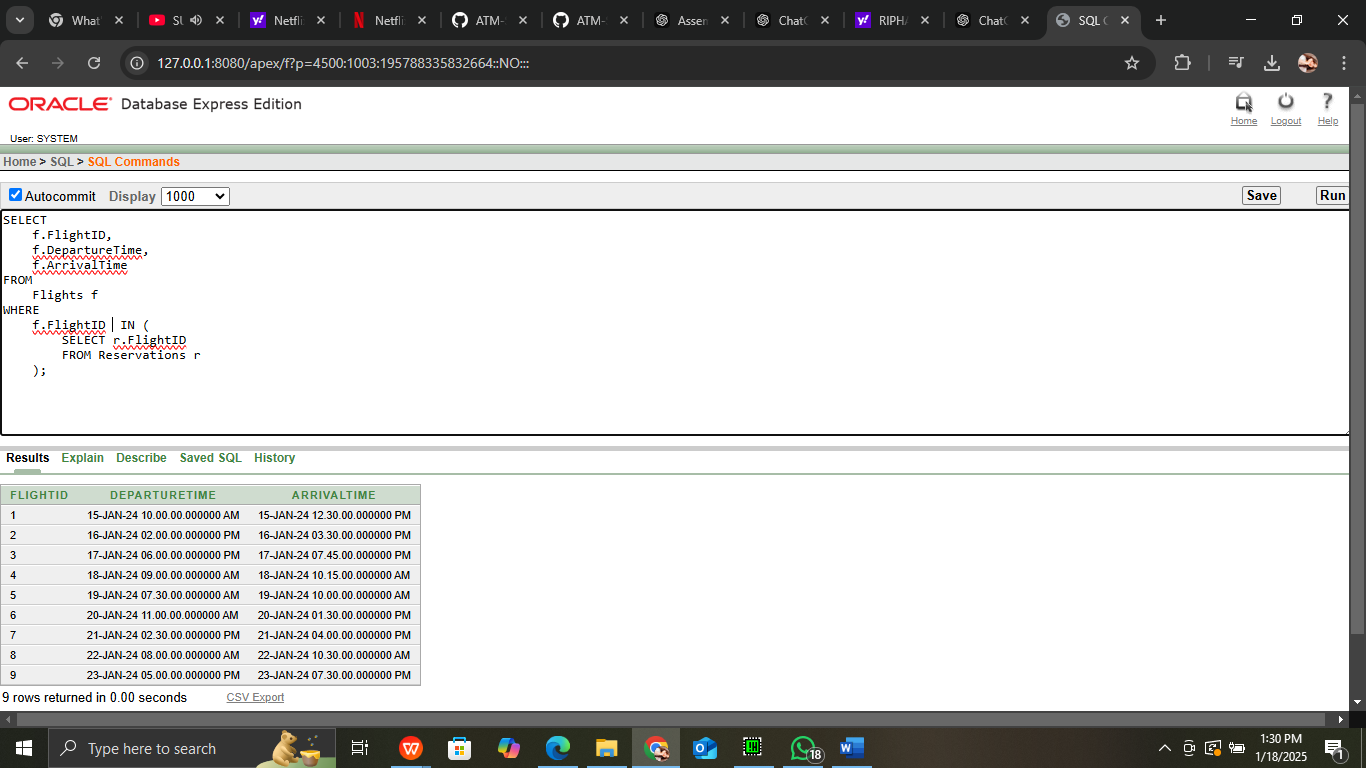
WHERE

f.FlightID NOT IN (

SELECT r.FlightID

FROM Reservations r

);



**9. Join with Feedback: Flights with Their Passenger Feedback**

This query lists flights and the feedback provided by passengers, including the rating and feedback text.

SELECT

f.FlightID,

f.DepartureTime,

f.ArrivalTime,

p.FirstName,

p.LastName,

fb.Rating,

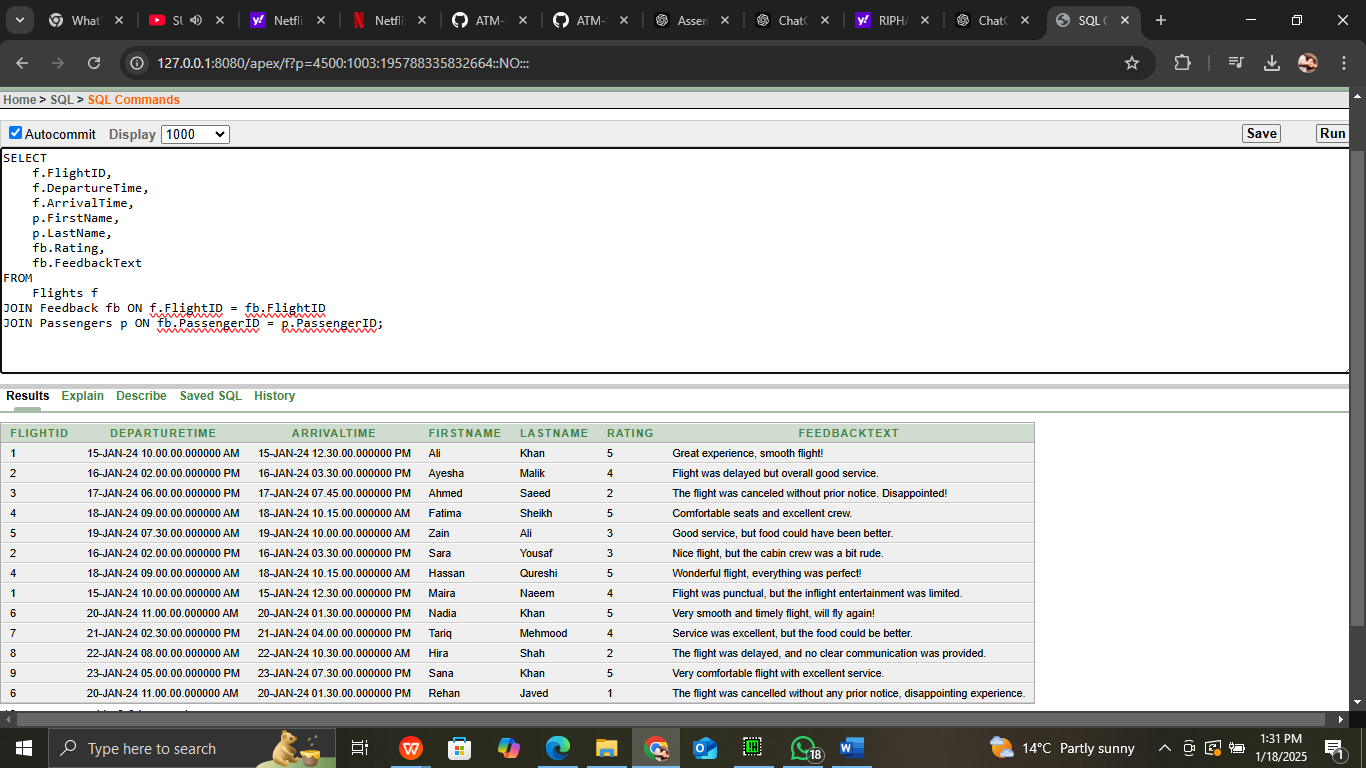
fb.FeedbackText

FROM

Flights f

JOIN Feedback fb ON f.FlightID = fb.FlightID

JOIN Passengers p ON fb.PassengerID = p.PassengerID;



**11. Subquery with Aggregate Function: Find the Flight with the Highest Revenue**

This query finds the flight with the highest total revenue based on ticket sales.

SELECT

f.FlightID,

f.DepartureTime,

f.ArrivalTime,

(SELECT SUM(t.Price)

FROM Tickets t

WHERE t.FlightID = f.FlightID

AND t.TicketStatus = 'Confirmed') AS TotalRevenue

FROM

Flights f

WHERE

(SELECT SUM(t.Price)

FROM Tickets t

WHERE t.FlightID = f.FlightID

AND t.TicketStatus = 'Confirmed') =

(SELECT MAX(TotalRevenue)

FROM (SELECT SUM(t.Price) AS TotalRevenue

FROM Tickets t

GROUP BY t.FlightID));

A screenshot of a computer

Description automatically generated

**12.Join with Multiple Aggregations: Average Ticket Price per Flight**

This query finds the average ticket price for each flight, grouped by flight ID.

SELECT

f.FlightID,

f.DepartureTime,

f.ArrivalTime,

AVG(t.Price) AS AverageTicketPrice

FROM

Flights f

JOIN Tickets t ON f.FlightID = t.FlightID

GROUP BY

f.FlightID, f.DepartureTime, f.ArrivalTime;

A screenshot of a computer

Description automatically generated

**13. Correlated Subquery: Find Passengers Who Have Luggage with Extra Charges**

This query finds passengers who have luggage that has extra charges.

SELECT

p.PassengerID,

p.FirstName,

p.LastName

FROM

Passengers p

WHERE

EXISTS (

SELECT 1

FROM Luggage l

WHERE l.PassengerID = p.PassengerID

AND l.ExtraCharges > 0

);

A screenshot of a computer

Description automatically generated

**14. List All Employees with Their Airlines and Airports**

This query lists all employees along with the name of the airline they work for and the airport they are associated with.

SELECT

e.EmployeeID,

e.Name AS EmployeeName,

a.AirlineName,

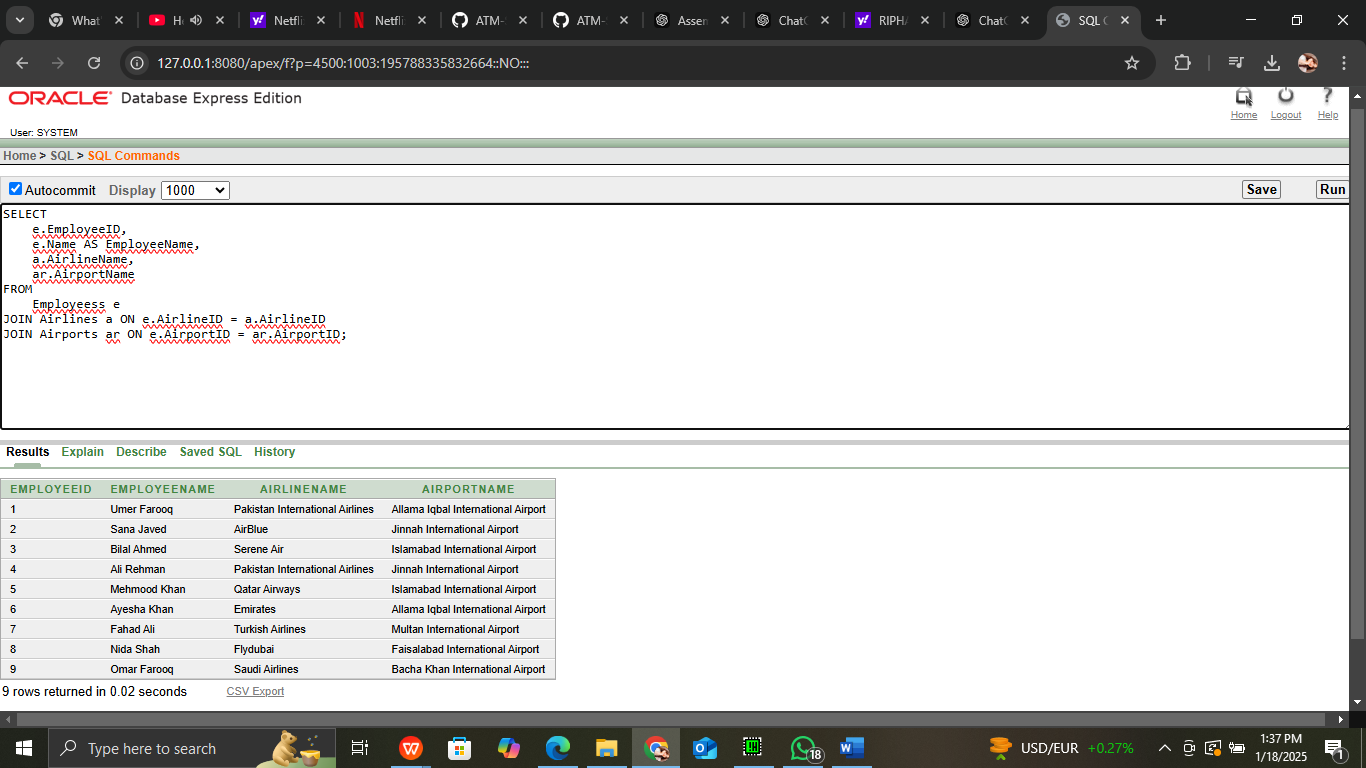
ar.AirportName

FROM

Employees e

JOIN Airlines a ON e.AirlineID = a.AirlineID

JOIN Airports ar ON e.AirportID = ar.AirportID;



**15 Count the Number of Employees Working for Each Airline**

This query counts the number of employees working for each airline.

SELECT

a.AirlineName,

COUNT(e.EmployeeID) AS NumberOfEmployees

FROM

Employees e

JOIN Airlines a ON e.AirlineID = a.AirlineID

GROUP BY

A screenshot of a computer

Description automatically generated

**16. Find the Number of Employees per Airport**

This query counts the number of employees working at each airport.

SELECT

ar.AirportName,

COUNT(e.EmployeeID) AS EmployeeCount

FROM

Employees e

JOIN Airports ar ON e.AirportID = ar.AirportID

GROUP BY

ar.AirportName;

A screenshot of a computer

Description automatically generated

***Task 5: Generate Reports***

**1. Monthly Sales Report: Total Sales per Month**

SELECT

TO\_CHAR(PaymentDate, 'YYYY-MM') AS Month,

SUM(PaymentAmount) AS TotalSales

FROM

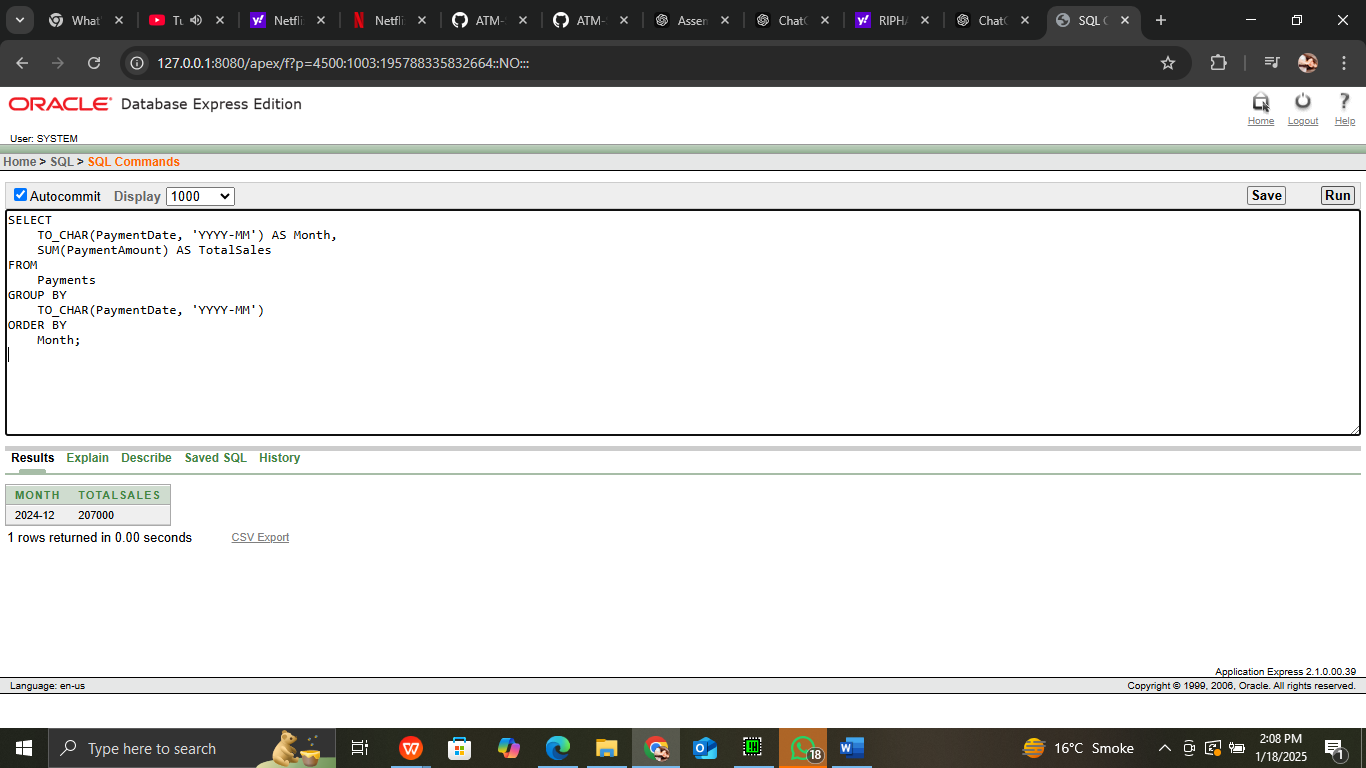
Payments

GROUP BY

TO\_CHAR(PaymentDate, 'YYYY-MM')

ORDER BY

Month;



1. **Top Users**: Customers with the highest purchase amounts.

SELECT \*

FROM (

SELECT

p.PassengerID,

p.FirstName || ' ' || p.LastName AS FullName,

SUM(py.PaymentAmount) AS TotalSpent

FROM

Passengers p

JOIN

Reservations r ON p.PassengerID = r.PassengerID

JOIN

Payments py ON r.ReservationID = py.ReservationID

GROUP BY

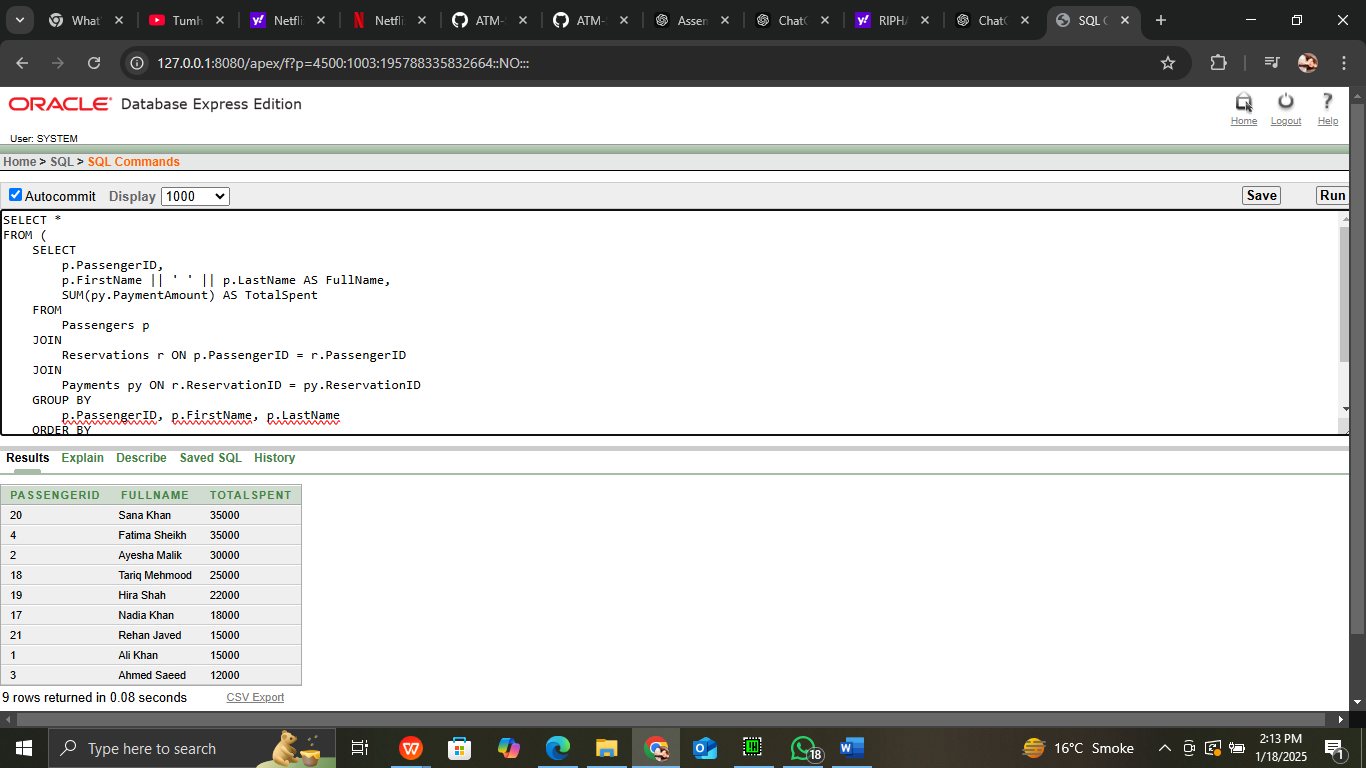
p.PassengerID, p.FirstName, p.LastName

ORDER BY

TotalSpent DESC

)

WHERE ROWNUM <= 10;



**VIEWS**

**FLIGHT RESERVATION DETAIL**

CREATE VIEW FlightDetails AS

SELECT

f.FlightID,

f.DepartureTime,

f.ArrivalTime,

f.Duration,

f.FlightStatus,

a.AirlineName,

d.AirportName AS DepartureAirport,

r.AirportName AS ArrivalAirport

FROM

Flights f

JOIN

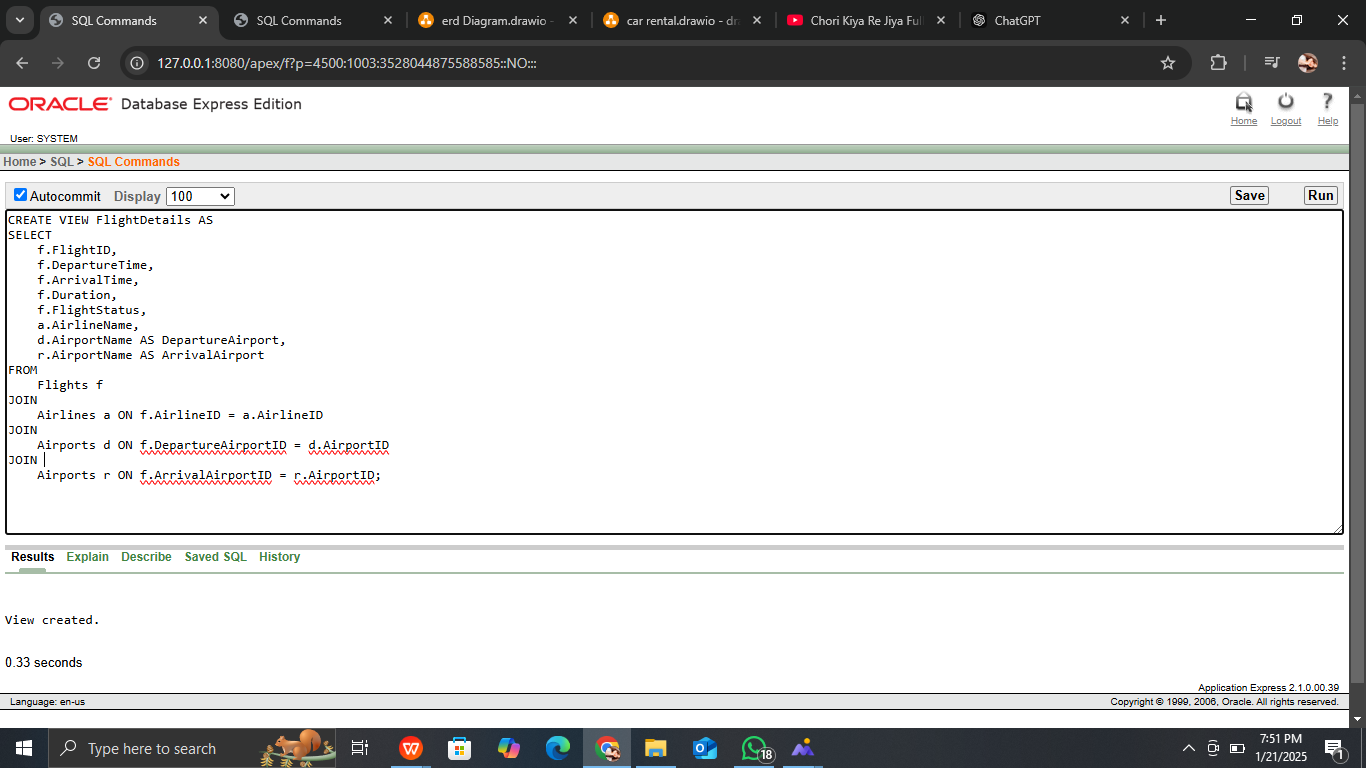
Airlines a ON f.AirlineID = a.AirlineID

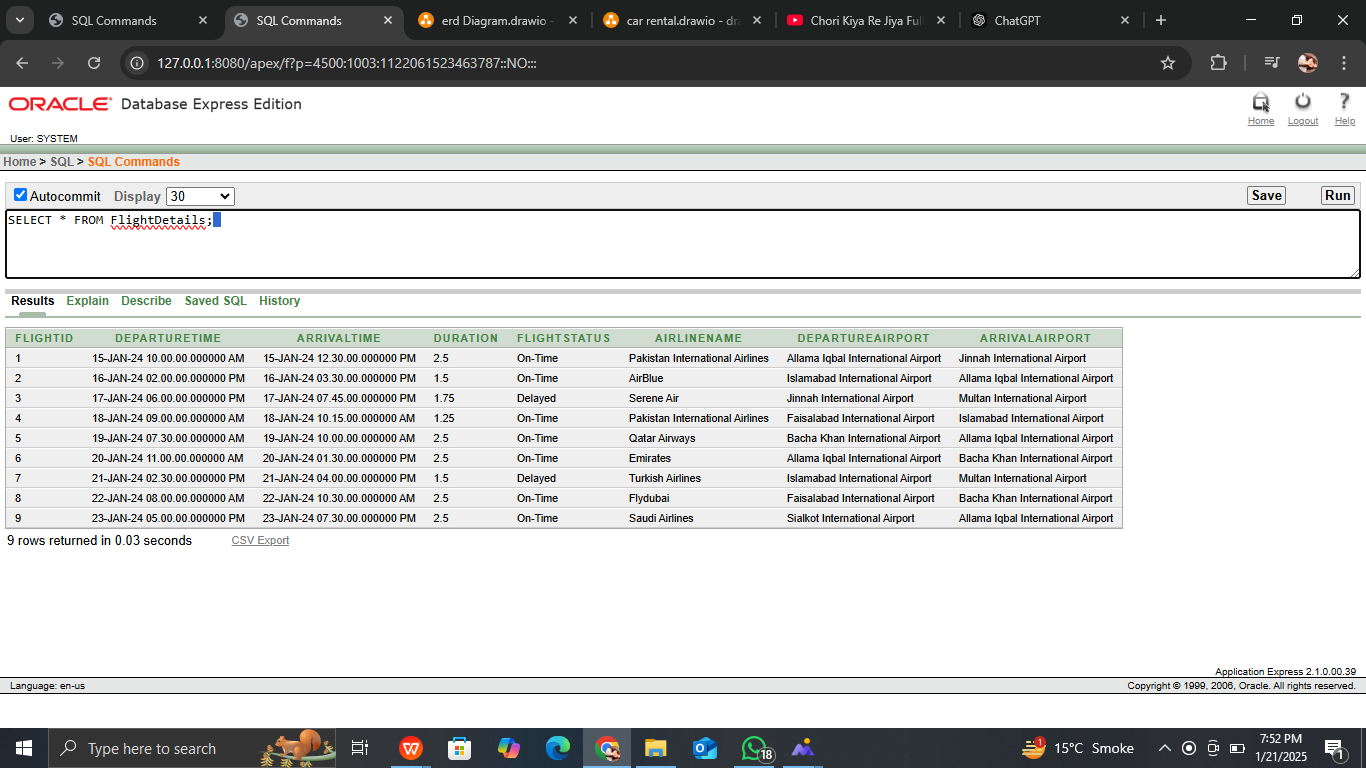
JOIN

Airports d ON f.DepartureAirportID = d.AirportID

JOIN

Airports r ON f.ArrivalAirportID = r.AirportID;





**PASSENGER RESERVATION**

CREATE VIEW PassengerReservations AS

SELECT

p.FirstName || ' ' || p.LastName AS PassengerName,

r.ReservationID,

r.BookingDate,

r.ReservationStatus,

f.FlightID,

f.DepartureTime,

f.ArrivalTime

FROM

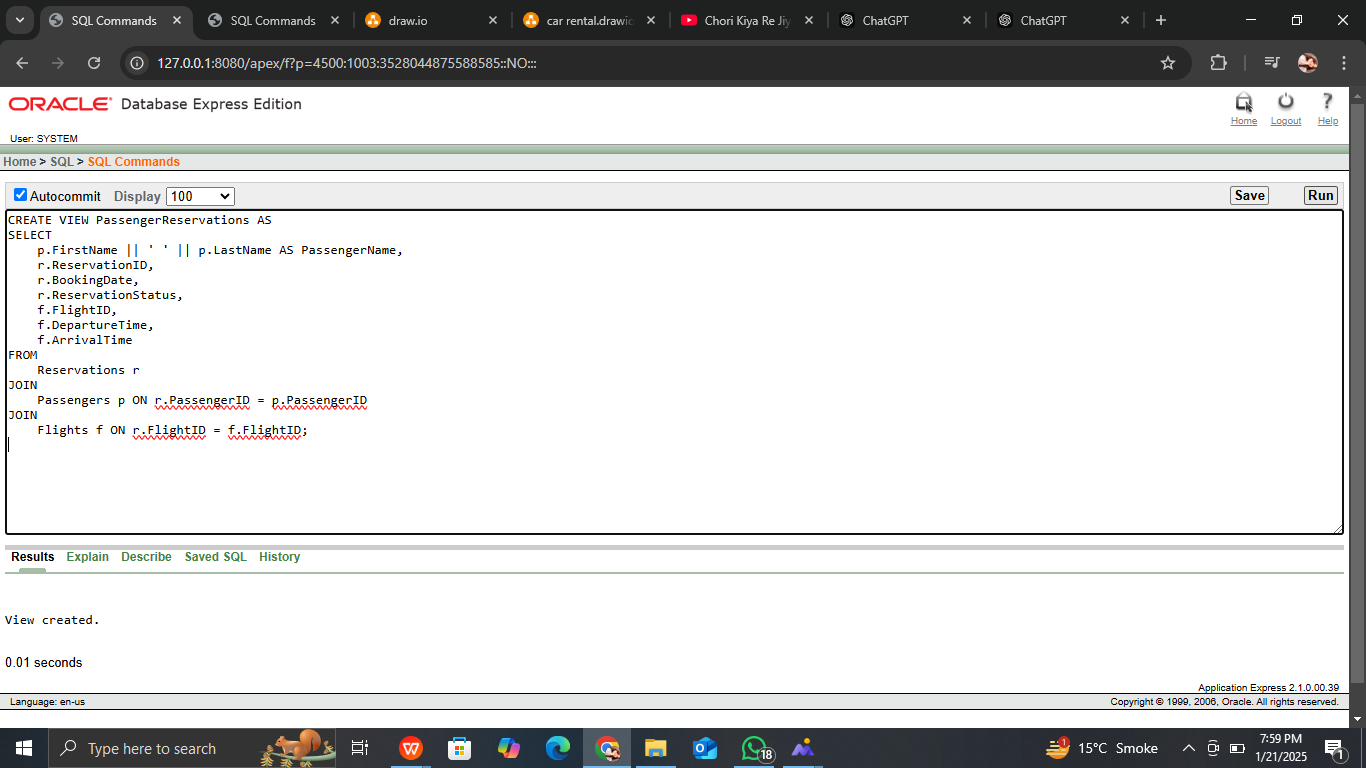
Reservations r

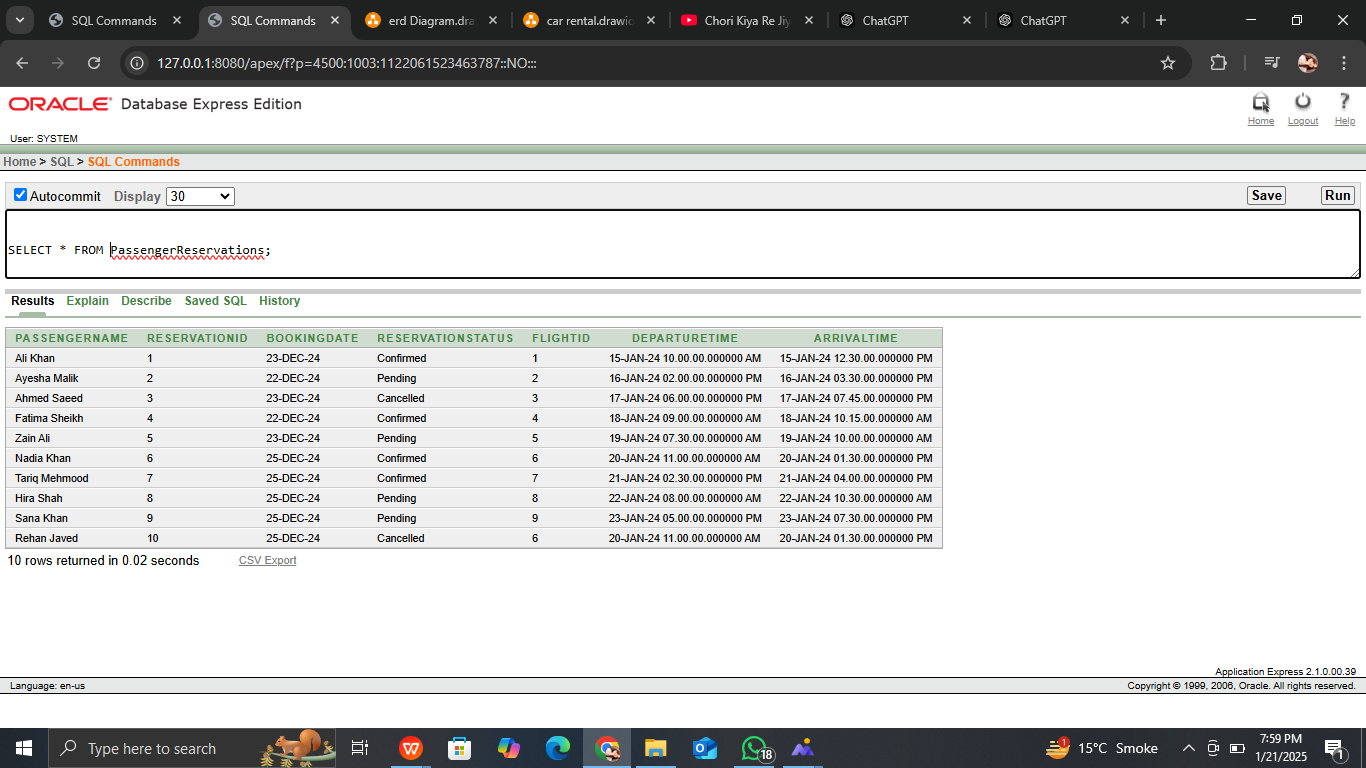
JOIN

Passengers p ON r.PassengerID = p.PassengerID

JOIN

Flights f ON r.FlightID = f.FlightID;





**PAYMENTSDETAILS**

CREATE VIEW PaymentDetails AS

SELECT

r.ReservationID,

r.ReservationStatus,

p.PaymentID,

p.PaymentAmount,

p.PaymentDate,

p.PaymentStatus,

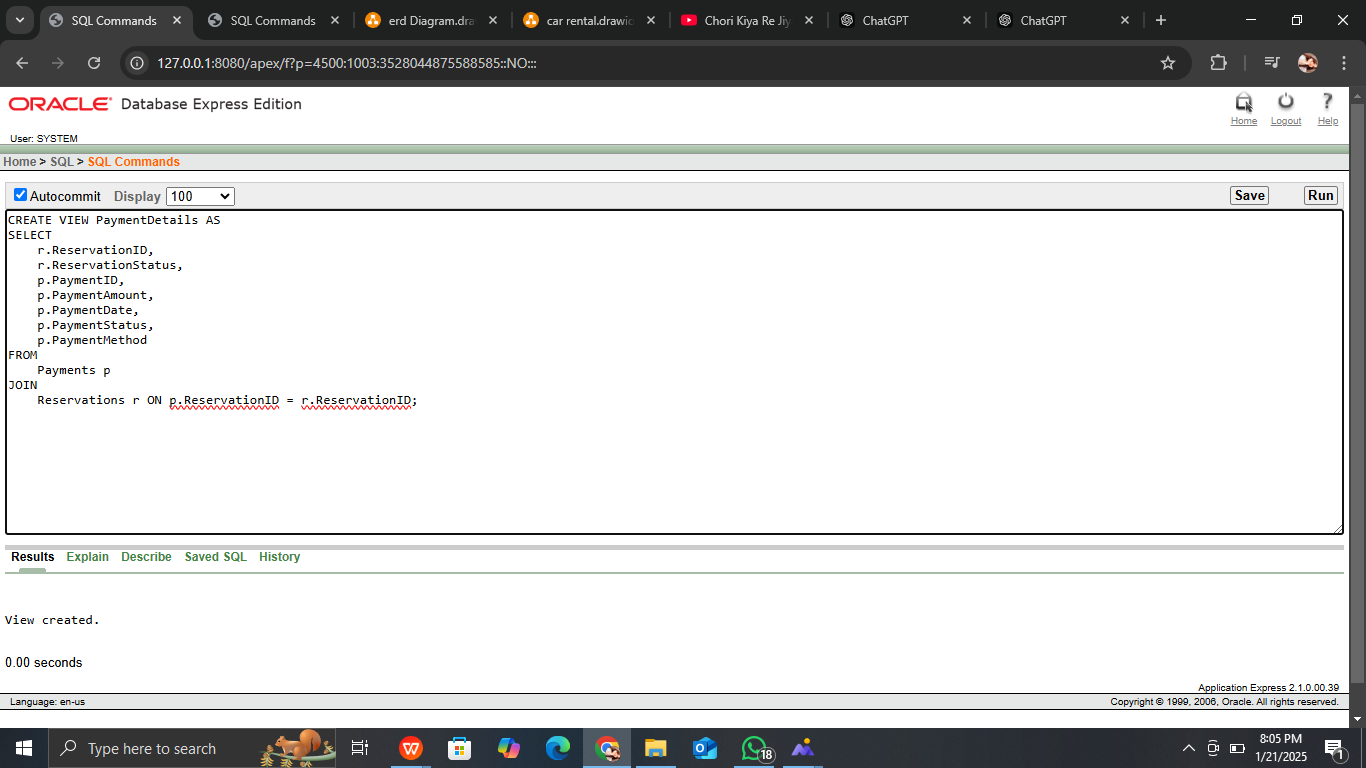
p.PaymentMethod

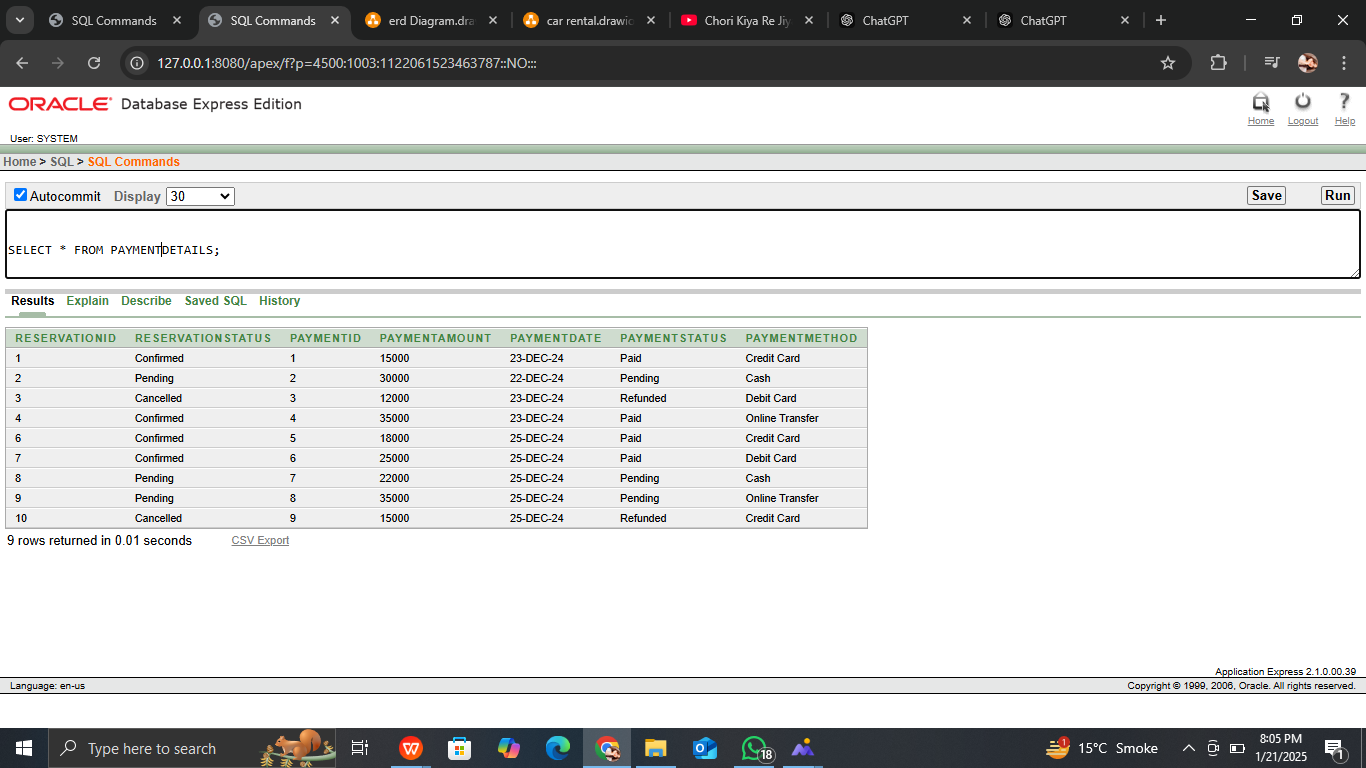
FROM

Payments p

JOIN

Reservations r ON p.ReservationID = r.ReservationID;





***PROCEDURES:***

**ReserveTicket**

CREATE OR REPLACE PROCEDURE ReserveTicket(

p\_PassengerID IN NUMBER,

p\_FlightID IN NUMBER,

p\_Class IN VARCHAR2,

p\_SeatNumber IN VARCHAR2,

p\_Price IN NUMBER,

p\_ReservationID OUT NUMBER

) IS

BEGIN

-- Insert into Reservations table

INSERT INTO Reservations (PassengerID, FlightID, ReservationStatus)

VALUES (p\_PassengerID, p\_FlightID, 'Pending')

RETURNING ReservationID INTO p\_ReservationID;

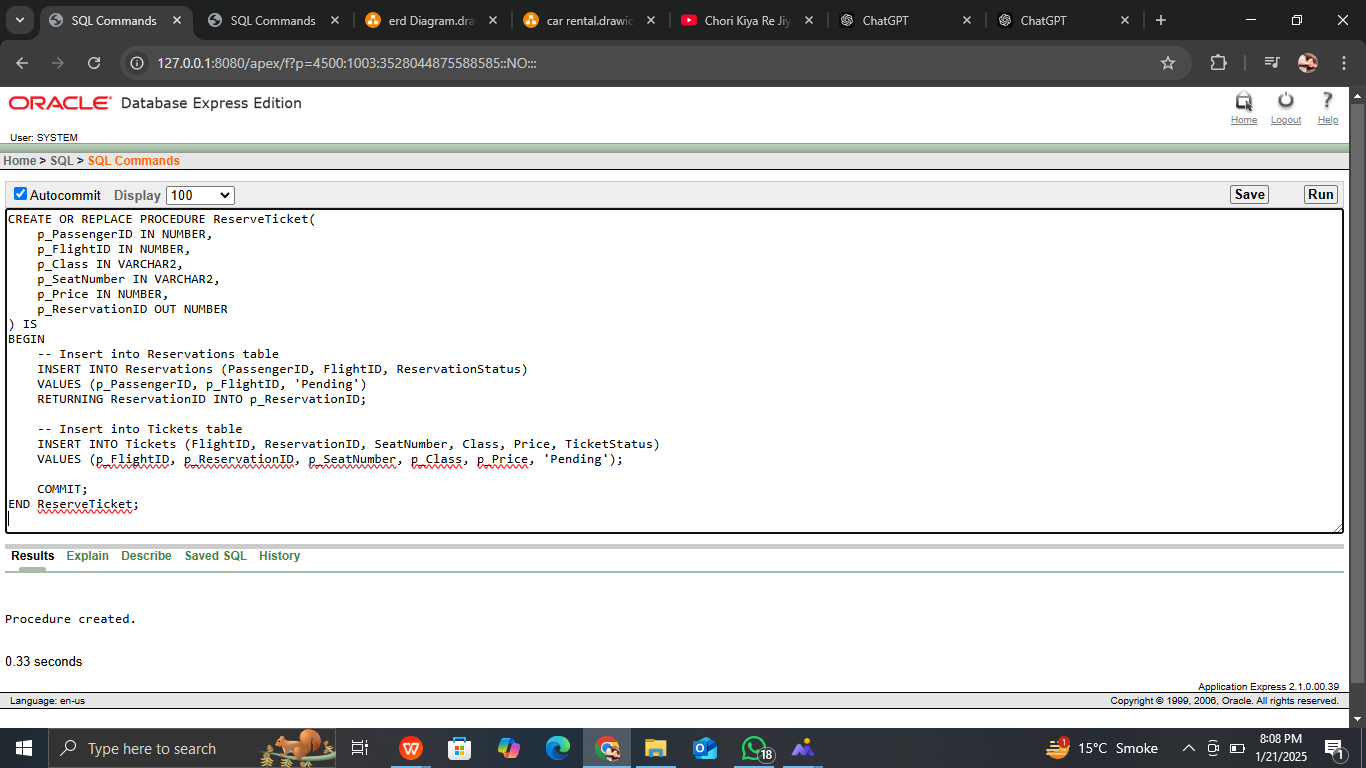
-- Insert into Tickets table

INSERT INTO Tickets (FlightID, ReservationID, SeatNumber, Class, Price, TicketStatus)

VALUES (p\_FlightID, p\_ReservationID, p\_SeatNumber, p\_Class, p\_Price, 'Pending');

COMMIT;

END ReserveTicket;



**CancelReservation**

CREATE OR REPLACE PROCEDURE CancelReservation(

p\_ReservationID IN NUMBER

) IS

BEGIN

-- Update reservation status to 'Cancelled'

UPDATE Reservations

SET ReservationStatus = 'Cancelled'

WHERE ReservationID = p\_ReservationID;

-- Update ticket status to 'Cancelled'

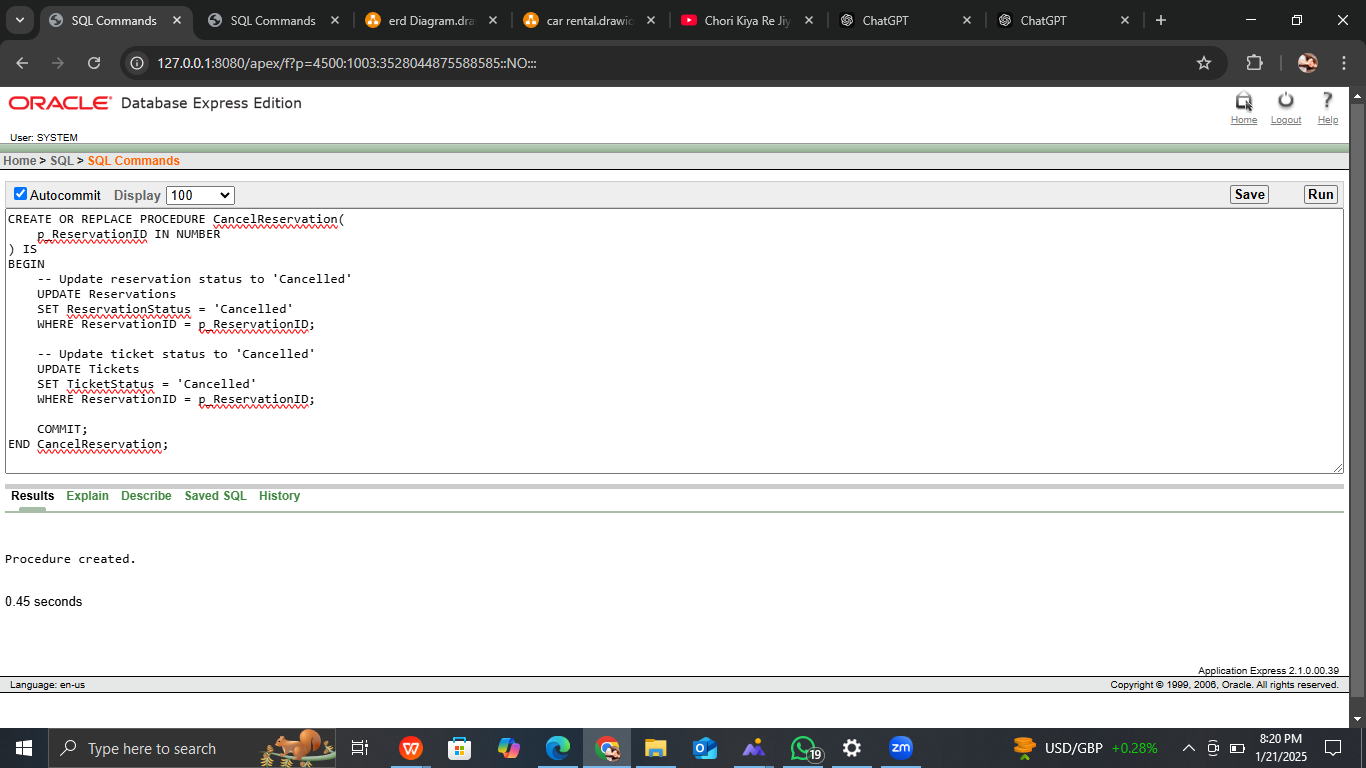
UPDATE Tickets

SET TicketStatus = 'Cancelled'

WHERE ReservationID = p\_ReservationID;

COMMIT;

END CancelReservation;



**ProcessPayment**

CREATE OR REPLACE PROCEDURE ProcessPayment(

p\_ReservationID IN NUMBER,

p\_PaymentAmount IN NUMBER,

p\_PaymentMethod IN VARCHAR2

) IS

BEGIN

-- Insert into Payments table

INSERT INTO Payments (ReservationID, PaymentAmount, PaymentMethod, PaymentStatus)

VALUES (p\_ReservationID, p\_PaymentAmount, p\_PaymentMethod, 'Completed');

-- Update reservation status to 'Confirmed' after payment

UPDATE Reservations

SET ReservationStatus = 'Confirmed'

WHERE ReservationID = p\_ReservationID;

COMMIT;

END ProcessPayment;

