|  |  |  |
| --- | --- | --- |
| **Name** | **ID** | **STUDENT SIGN** |
| **MOST. SAYMA KHATUN** | **22-47035-1** |  |

**Instructions:**

* **Make sure to write your Name, ID and Signature on this document.**
* **First write your signature on a paper then take photo of that signature and use it for signing this document.**
* **After completing the requirements of the midterm assignment by editing this document, upload this document in the link provided in your VUES Student Account.**

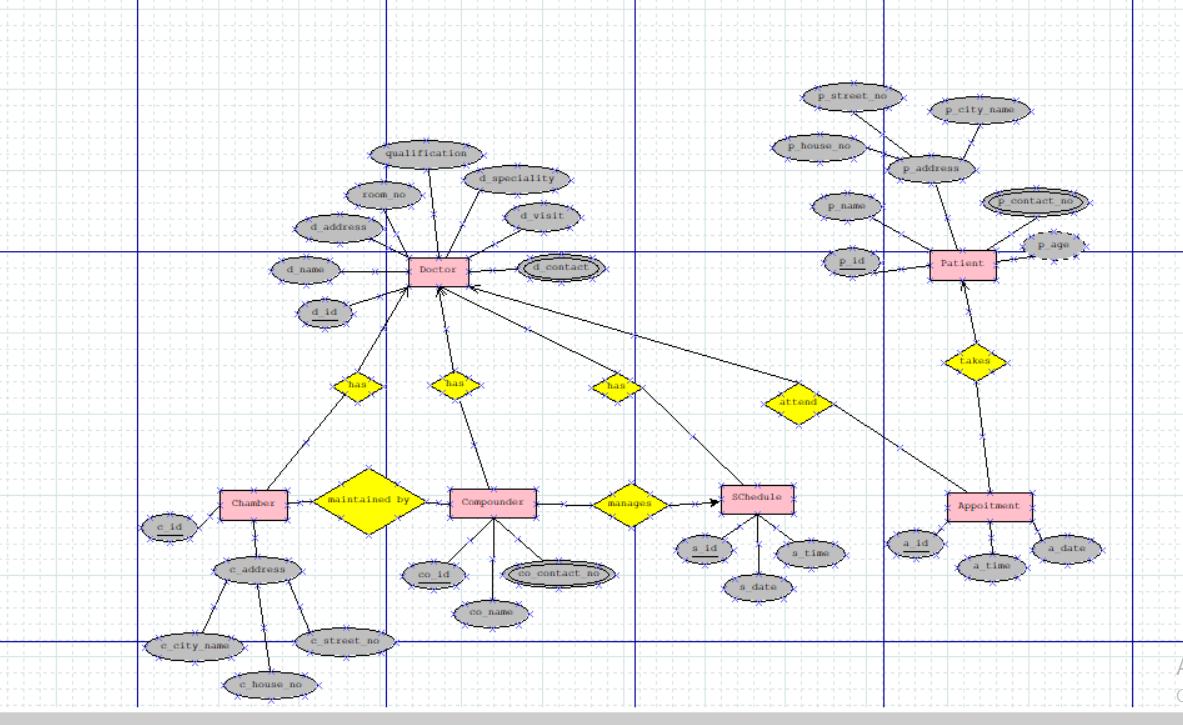
**Midterm Assignment**

1. **Below a scenario has been given draw the ER Diagram.**

***Draw with proper annotations (use DIA, VISIO, MS WORD etc.).*** ***For reference see ERDiagramTutorial.***

In a doctor appointment management system, a patient takes appointment from a doctor and a doctor attend appointment of many patients. A doctor is identified by doctor id, name, address, room no, qualification, specialty, visit, and contact no. There may be multiple contact no of a doctor. A Patient is identified by their unique ID and the system stores their name, mobile number, age and address. A patient address is composed of city name, street number, and house number. One doctor can be associated with many chambers, but one chamber is associated with only one doctor and the system stores the id and address of each chamber, including the city name, street number, and house number. Additionally, one doctor has many compounders, but a compounder is associated with only one doctor and a compounder is identified by their ID, name, and contact numbers. A chamber is maintained by only one compounder and a compounder maintain only one chamber. While taking appointment, the date and time of appointment for each patient is also stored. Furthermore, one doctor can have many schedules, but each schedule is associated with only one doctor and the system stores the id, time and date for each schedule and a compounder can manage many schedules for doctor, but a schedule is managed by one compounder.

Answer 1:

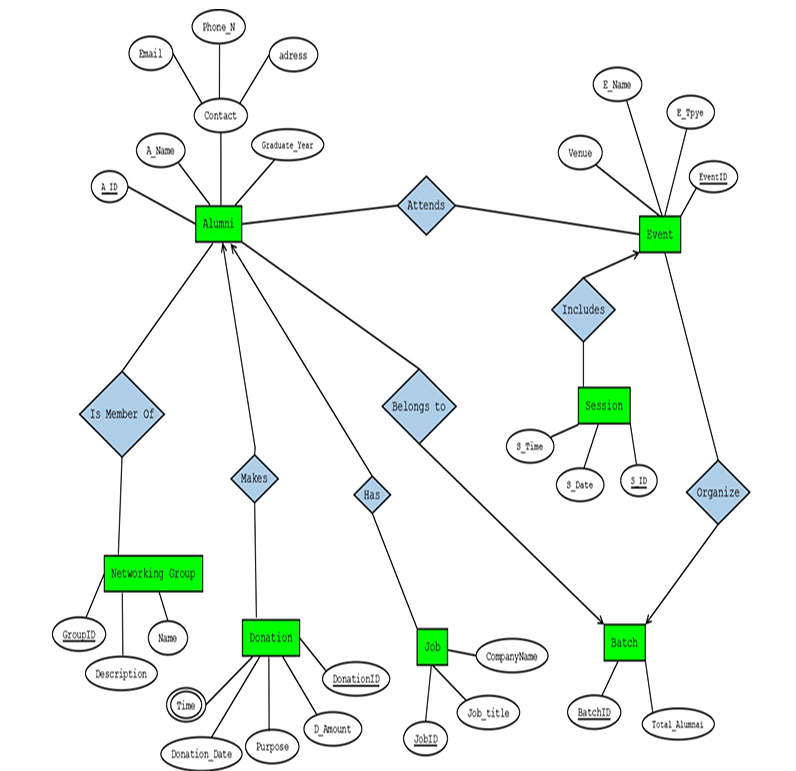


1. **Below an ER Diagram has been given write the scenario.**

***For reference see ERDiagramTutorial.***

1. **Below an ER Diagram has been given write the scenario.**

***For reference see ERDiagramTutorial.***



Answer 2:

An Alumni Management System is created to manage alumni profiles, their participation in events, donations they make, networking groups they belong to, jobs they have, and batch records. Each alumni is identified by an Alumni ID. The system stores alumni name, graduation year, and contact information which includes email, phone number, and address. Each alumni belongs to one batch. A batch can have many alumni. A batch is identified by a batch ID, and also stores the total number of alumni. A batch can organize multiple events, and an event can be organized by many batches. An event is identified by an event ID and includes event name, type, and venue. Alumni can attend multiple events, and each event can be attended by multiple alumni. Events include one or more session. Each session has a session ID and stores session date and time. A session always belongs to one specific event. An alumni can make multiple donations. A donation is identified by a Donation ID, and stores the amount, date, time, and purpose of the donation. Alumni can also be members of multiple networking groups, and each group can have multiple alumni. A networking group is identified by Group ID, and also stores the group name and description. An alumni can have several jobs, and each job is identified by a Job ID. The system also keeps job title and company name for each job.

1. **Normalize the ER Diagram given below up to 3rd Normal Form and finalize the tables that needs to be created. Then (in Oracle using SQL) write down the queries that are required to create all the tables with necessary constraints. Also insert at least 3 rows of data in each created table.**

***For reference see NormalizationTutorial and BasicSQLTutorial.***

A diagram of a company

Description automatically generated

Answer Box (Normalization steps in detail as shown in Normalization Tutorial Slide + all the queries required to create the tables and insert data after Normalization):

**Step 1: Analysing the ER Diagram**

From the diagram, we can identify the following entities and their attributes:

* **Driver**: Driver\_Id (PK), First\_Name, Last\_Name, License\_Number, and Experience\_Year

* **Bus**: Bus\_Number (PK), Type, Model, and Capacity

* **Route**: Route\_Id (PK), Start\_Location, End\_Location, Distance, and Departure\_Time

* **Supervisor**: Supervisor\_Id (PK), First\_Name, Last\_Name, and Experience\_Year

* **Ticket**: Ticket\_Number (PK), Ticket\_Status, Ticket\_Price, and Seat\_Number
* **Passenger**: National\_Id (PK), First\_Name, Last\_Name, Phone\_Number, and Gender
* **Transaction**: Transaction\_Id (PK), Date, and Total\_Amount
* **Payment\_Type**: Payment\_Code (PK), Payment\_Method, and Payment\_Type

**Step-2: Normalization:**

Operate by

**UNF:**

Operate by(Driver\_Id, First\_Name, Last\_Name, License\_Number, Experience\_Year, Bus\_Number, Type, Model, Capacity,)

**1NF:**

There is no multivalued attribute. Relation is already in 1NF.

1. Driver\_Id, First\_Name, Last\_Name, License\_Number, Experience\_Year,

Bus\_Number, Type, Model, Capacity.

**2NF:**

1. Driver\_Id, First\_Name, Last\_Name, License\_Number, Experience\_Year.
2. Bus\_Number, Type, Model, Capacity.

**3NF:**

Name is transitive attribute.

1. Driver\_Id, License\_Number, Experience\_Year.
2. First\_Name, Last\_Name.
3. Bus\_Number, Type, Model, Capacity.

**Table Creation:**

1. Driver\_Id, License\_Number, Experience\_Year, Bus\_Number
2. Name , First\_Name, Last\_Name.
3. Bus\_Number, Type, Model, Capacity.

Follows:

**UNF:**

Follows( Bus\_Number, Type, Model, Capacity, Rute\_Id , Start\_Location, End\_Location, Distance, Departure\_Time)

**1NF:**

There is no multivalued attribute. Relation is already in 1NF.

1. Bus\_Number, Type, Model, Capacity, Rute\_Id , Start\_Location, End\_Location, Distance, Departure\_Time.

**2NF:**

1. Bus\_Number, Type, Model, Capacity.
2. Rute\_Id , Start\_Location, End\_Location, Distance, Departure\_Time.

**3NF:**

Destination is transitive attribute.

1. Bus\_Number, Type, Model, Capacity.
2. Rute\_Id , Distance, Departure\_Time.
3. Start\_Location, End\_Location.

**Table Creation:**

1. Bus\_Number, Type, Model, Capacity, Rute\_Id.
2. Rute\_Id , Distance, Departure\_Time.
3. Destination, Start\_Location, End\_Location.

Contacts

**UNF:**

Contacts( Bus\_Number, Type, Model, Capacity, Supervisor\_Id, First\_Name, Last\_Name, Experience\_Year)

**1NF:**

There is no multivalued attribute. Relation is already in 1NF.

1. Bus\_Number, Type, Model, Capacity, Supervisor\_Id, First\_Name, Last\_Name, Experience\_Year.

**2NF:**

1. Bus\_Number, Type, Model, Capacity.
2. Supervisor\_Id, First\_Name, Last\_Name, Experience\_Year

**3NF:**

Name is Transitive attribute.

1. Bus\_Number, Type, Model, Capacity.
2. Supervisor\_Id, Experience\_Year.
3. First\_Name, Last\_Name.

**Table Creation:**

1. Bus\_Number, Type, Model, Capacity.
2. Supervisor\_Id, Experience\_Year, Bus\_Number.
3. Name, First\_Name, Last\_Name.

Assign

**UNF:**

Assign (Bus\_Number, Type, Model, Capacity, Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number)

**1NF:**

There is no multivalued attribute. Relation is already in 1NF.

1. Bus\_Number, Type, Model, Capacity, Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number.

**2NF:**

1. Bus\_Number, Type, Model, Capacity,.
2. Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number.

**3NF:**

There is no transitive attribute. Relation is already in 3NF.

1. Bus\_Number, Type, Model, Capacity,.
2. Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number.

**Table Creation:**

1. Bus\_Number, Type, Model, Capacity,.
2. Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number, Bus\_Number.

Buy

**UNF:**

Buy(Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number, National\_Id, First\_Name, Last\_Name, Gender, Phone\_Number)

**1NF:**

Phone\_Number is multivalued attribute.

1. Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number, National\_Id, First\_Name, Last\_Name, Gender, Phone\_Number.

**2NF:**

1. Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number.
2. National\_Id, First\_Name, Last\_Name, Gender, Phone\_Number.

**3NF:**

Name is transitive attribute.

1. Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number.
2. National\_Id, Gender, Phone\_Number.
3. First\_Name, Last\_Name.

**Table Creation:**

1. Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number, National\_Id.
2. National\_Id, Gender, Phone\_Number.
3. Name, First\_Name, Last\_Name.

Makes

**UNF:**

Makes(National\_Id, First\_Name, Last\_Name, Gender, Phone\_Number, Transaction\_Id, Date, Total\_Amount)

**1NF:**

Phone\_Number is multivalued attribute.

1. National\_Id, First\_Name, Last\_Name, Gender, Phone\_Number, Transaction\_Id, Date, Total\_Amount.

**2NF:**

1. National\_Id, First\_Name, Last\_Name, Gender, Phone\_Number.
2. Transaction\_Id, Date, Total\_Amount.

**3NF:**

Name is transitive attribute.

1. Nationalr\_Id, Gender, Phone\_Number
2. First\_Name, Last\_Name,
3. Transaction\_Id, Date, Total\_Amount.

**Table Creation:**

1. National\_Id, Gender, Phone\_Number
2. Name , First\_Name, Last\_Name,
3. Transaction\_Id, Date, Total\_Amount, National\_Id.

Contains

**UNF:**

Contains( Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number, Transaction\_Id, Date, Total\_Amount.)

**1NF:**

There is no multivalued attribute. Relation is already in 1NF.

1. Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number, Transaction\_Id, Date, Total\_Amount.

**2NF:**

1. Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number.
2. Transaction\_Id, Date, Total\_Amount.

**3NF:**

There is no transitive attribute.

1. Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number.
2. Transaction\_Id, Date, Total\_Amount.

**Table Creation:**

1. Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number, Transaction\_Id.
2. Transaction\_Id, Date, Total\_Amount.

Uses

**UNF:**

Uses(Transaction\_Id, Date, Total\_Amount, Payment\_Code, Payment\_Method, Payment\_Type)

**1NF:**

There is no multivalued attribute. Relation is already in 1NF.

1. Transaction\_Id, Date, Total\_Amount, Payment\_Code, Payment\_Method, Payment\_Type.

**2NF:**

1. Transaction\_Id, Date, Total\_Amount.
2. Payment\_Code, Payment\_Method, Payment\_Type.

**3NF:**

There is no transitive attribute. Relation is already in 3NF.

1. Transaction\_Id, Date, Total\_Amount.
2. Payment\_Code, Payment\_Method, Payment\_Type.

**Table Creation:**

1. Transaction\_Id, Date, Total\_Amount, Payment\_Code.
2. Payment\_Code, Payment\_Method, Payment\_Type.

Issues

**UNF:**

Issues(Supervisor\_Id, First\_Name, Last\_Name, Experience\_Year, Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number)

**1NF:**

There is no multivalued attribute. Relation is already in 1NF.

1. Supervisor\_Id, First\_Name, Last\_Name, Experience\_Year, Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number.

**2NF:**

1. Supervisor\_Id, First\_Name, Last\_Name, Experience\_Year.
2. Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number.

**3NF:**

Name is transitive attribute.

1. Supervisor\_Id, Experience\_Year.
2. First\_Name, Last\_Name.
3. Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number.

**Table Creation:**

1. Supervisor\_Id, Experience\_Year.
2. Name , First\_Name, Last\_Name.
3. Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number, Supervisor\_Id.

**Temporary Table:**

1. Driver\_Id, License\_Number, Experience\_Year, Bus\_Number
2. Name , First\_Name, Last\_Name.
3. Bus\_Number, Type, Model, Capacity.
4. Bus\_Number, Type, Model, Capacity, Rute\_Id.
5. Rute\_Id , Distance, Departure\_Time.
6. Destination, Start\_Location, End\_Location.
7. Bus\_Number, Type, Model, Capacity.
8. Supervisor\_Id, Experience\_Year, Bus\_Number.
9. Name, First\_Name, Last\_Name.
10. Bus\_Number, Type, Model, Capacity,.
11. Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number, Bus\_Number.
12. Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number, National\_Id.
13. National\_Id, Gender, Phone\_Number.
14. Name, First\_Name, Last\_Name.
15. National\_Id, Gender, Phone\_Number
16. Name , First\_Name, Last\_Name,
17. Transaction\_Id, Date, Total\_Amount, National\_Id.
18. Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number, Transaction\_Id.
19. Transaction\_Id, Date, Total\_Amount.
20. Transaction\_Id, Date, Total\_Amount, Payment\_Code.
21. Payment\_Code, Payment\_Method, Payment\_Type.
22. Supervisor\_Id, Experience\_Year.
23. Name , First\_Name, Last\_Name.
24. Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number, Supervisor\_Id.

Final table Table:

1. Driver\_Id, License\_Number, Experience\_Year, Bus\_Number
2. Name , First\_Name, Last\_Name.
3. Bus\_Number, Type, Model, Capacity, Rute\_Id.
4. Rute\_Id , Distance, Departure\_Time.
5. Destination, Start\_Location, End\_Location.
6. Supervisor\_Id, Experience\_Year, Bus\_Number.
7. Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number, Bus\_Number.
8. Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number, National\_Id.
9. National\_Id, Gender, Phone\_Number
10. Transaction\_Id, Date, Total\_Amount, National\_Id.
11. Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number, Transaction\_Id.
12. Transaction\_Id, Date, Total\_Amount, Payment\_Code.
13. Payment\_Code, Payment\_Method, Payment\_Type.
14. Supervisor\_Id, Experience\_Year.
15. Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number, Supervisor\_Id.

**Table Creation Query:**

1. Driver

CREATE TABLE Driver (

Driver\_Id NUMBER PRIMARY KEY,

License\_Number VARCHAR2(20) UNIQUE NOT NULL,

Experience\_Year NUMBER,

Bus\_Number VARCHAR2(10)

);

2. Name

CREATE TABLE Name (

Name\_Id NUMBER PRIMARY KEY,

First\_Name VARCHAR2(50),

Last\_Name VARCHAR2(50)

);

3. Bus

CREATE TABLE Bus (

Bus\_Number VARCHAR2(10) PRIMARY KEY,

Type VARCHAR2(20),

Model VARCHAR2(30),

Capacity NUMBER,

Rute\_Id NUMBER

);

4. Rute

CREATE TABLE Rute (

Rute\_Id NUMBER PRIMARY KEY,

Distance NUMBER,

Departure\_Time TIMESTAMP

);

5. Destination

CREATE TABLE Destination (

Destination\_Id NUMBER PRIMARY KEY,

Start\_Location VARCHAR2(100),

End\_Location VARCHAR2(100)

);

6. Supervisor

CREATE TABLE Supervisor (

Supervisor\_Id NUMBER PRIMARY KEY,

Experience\_Year NUMBER,

Bus\_Number VARCHAR2(10),

FOREIGN KEY (Bus\_Number) REFERENCES Bus(Bus\_Number)

);

7. Ticket\_Bus

CREATE TABLE Ticket\_Bus (

Ticket\_Number NUMBER PRIMARY KEY,

Ticket\_Status VARCHAR2(20),

Ticket\_Price NUMBER,

Seat\_Number VARCHAR2(5),

Bus\_Number VARCHAR2(10),

FOREIGN KEY (Bus\_Number) REFERENCES Bus(Bus\_Number)

);

8. Ticket\_Customer

CREATE TABLE Ticket\_Customer (

Ticket\_Number NUMBER PRIMARY KEY,

Ticket\_Status VARCHAR2(20),

Ticket\_Price NUMBER,

Seat\_Number VARCHAR2(5),

National\_Id VARCHAR2(20),

FOREIGN KEY (National\_Id) REFERENCES Passenger(National\_Id)

);

);

9. Passenger

CREATE TABLE Passenger (

National\_Id VARCHAR2(20) PRIMARY KEY,

Gender VARCHAR2(10),

Phone\_Number VARCHAR2(15)

);

10. Transaction\_Customer

CREATE TABLE Transaction\_Customer (

Transaction\_Id NUMBER PRIMARY KEY,

Date DATE,

Total\_Amount NUMBER,

National\_Id VARCHAR2(20),

FOREIGN KEY (National\_Id) REFERENCES Passenger(National\_Id)

);

11. Ticket\_Transaction

CREATE TABLE Ticket\_Transaction (

Ticket\_Number NUMBER PRIMARY KEY,

Ticket\_Status VARCHAR2(20),

Ticket\_Price NUMBER,

Seat\_Number VARCHAR2(5),

Transaction\_Id NUMBER,

FOREIGN KEY (Transaction\_Id) REFERENCES Transaction\_Customer(Transaction\_Id)

);

12. Transaction\_Payment

CREATE TABLE Transaction\_Payment (

Transaction\_Id NUMBER PRIMARY KEY,

Date DATE,

Total\_Amount NUMBER,

Payment\_Code VARCHAR2(20),

FOREIGN KEY (Transaction\_Id) REFERENCES Transaction\_Customer(Transaction\_Id)

);

13. Payment

CREATE TABLE Payment (

Payment\_Code VARCHAR2(20) PRIMARY KEY,

Payment\_Method VARCHAR2(20),

Payment\_Type VARCHAR2(20)

);

14. Supervisor\_Info

CREATE TABLE Supervisor\_Info (

Supervisor\_Id NUMBER PRIMARY KEY,

Experience\_Year NUMBER

);

15. Ticket\_Supervisor

CREATE TABLE Ticket\_Supervisor (

Ticket\_Number NUMBER PRIMARY KEY,

Ticket\_Status VARCHAR2(20),

Ticket\_Price NUMBER,

Seat\_Number VARCHAR2(5),

Supervisor\_Id NUMBER,

FOREIGN KEY (Supervisor\_Id) REFERENCES Supervisor(Supervisor\_Id)

);

**Data Insertion Query:**

Table 1

INSERT INTO Driver VALUES (1, 'DL1001', 5, 'BUS101');

INSERT INTO Driver VALUES (2, 'DL1002', 10, 'BUS102');

INSERT INTO Driver VALUES (3, 'DL1003', 3, 'BUS103');

Table 2

INSERT INTO Name VALUES (1, 'John', 'Doe');

INSERT INTO Name VALUES (2, 'Alice', 'Smith');

INSERT INTO Name VALUES (3, 'Bob', 'Brown');

Table 3

INSERT INTO Bus VALUES ('BUS101', 'AC', 'Volvo', 45, 201);

INSERT INTO Bus VALUES ('BUS102', 'Non-AC', 'Tata', 50, 202);

INSERT INTO Bus VALUES ('BUS103', 'AC', 'AshokLeyland', 40, 203);

Table 4

INSERT INTO Rute VALUES (201, 300, TO\_TIMESTAMP('2025-04-12 08:00:00', 'YYYY-MM-DD HH24:MI:SS'));

INSERT INTO Rute VALUES (202, 150, TO\_TIMESTAMP('2025-04-12 10:00:00', 'YYYY-MM-DD HH24:MI:SS'));

INSERT INTO Rute VALUES (203, 200, TO\_TIMESTAMP('2025-04-12 12:00:00', 'YYYY-MM-DD HH24:MI:SS'));

Table 5

INSERT INTO Destination VALUES (1, 'City A', 'City B');

INSERT INTO Destination VALUES (2, 'City B', 'City C');

INSERT INTO Destination VALUES (3, 'City A', 'City C');

Table 6

INSERT INTO Supervisor VALUES (101, 7, 'BUS101');

INSERT INTO Supervisor VALUES (102, 5, 'BUS102');

INSERT INTO Supervisor VALUES (103, 8, 'BUS103');

Table 7

INSERT INTO Ticket\_Bus VALUES (1001, 'Confirmed', 500, 'A1', 'BUS101');

INSERT INTO Ticket\_Bus VALUES (1002, 'Pending', 400, 'B2', 'BUS102');

INSERT INTO Ticket\_Bus VALUES (1003, 'Cancelled', 450, 'C3', 'BUS103');

Table 9

INSERT INTO Passenger VALUES ('NID001', 'Male', '01711111111');

INSERT INTO Passenger VALUES ('NID002', 'Female', '01822222222');

INSERT INTO Passenger VALUES ('NID003', 'Male', '01933333333');

Table 8

INSERT INTO Ticket\_Customer VALUES (1001, 'Confirmed', 500, 'A1', 'NID001');

INSERT INTO Ticket\_Customer VALUES (1002, 'Confirmed', 400, 'B2', 'NID002');

INSERT INTO Ticket\_Customer VALUES (1003, 'Cancelled', 450, 'C3', 'NID003');

Table 10

INSERT INTO Transaction\_Customer VALUES (501, TO\_DATE('2025-04-10', 'YYYY-MM-DD'), 500, 'NID001');

INSERT INTO Transaction\_Customer VALUES (502, TO\_DATE('2025-04-11', 'YYYY-MM-DD'), 400, 'NID002');

INSERT INTO Transaction\_Customer VALUES (503, TO\_DATE('2025-04-12', 'YYYY-MM-DD'), 450, 'NID003');

Table 11

INSERT INTO Ticket\_Transaction VALUES (1001, 'Confirmed', 500, 'A1', 501);

INSERT INTO Ticket\_Transaction VALUES (1002, 'Confirmed', 400, 'B2', 502);

INSERT INTO Ticket\_Transaction VALUES (1003, 'Cancelled', 450, 'C3', 503);

Table-12:

INSERT INTO Transaction\_Payment VALUES (501, TO\_DATE('2025-04-10', 'YYYY-MM-DD'), 500, 'PAY001');

INSERT INTO Transaction\_Payment VALUES (502, TO\_DATE('2025-04-11', 'YYYY-MM-DD'), 400, 'PAY002');

INSERT INTO Transaction\_Payment VALUES (503, TO\_DATE('2025-04-12', 'YYYY-MM-DD'), 450, 'PAY003');

Table 13

INSERT INTO Payment VALUES ('PAY001', 'Bkash', 'Mobile');

INSERT INTO Payment VALUES ('PAY002', 'Cash', 'Direct');

INSERT INTO Payment VALUES ('PAY003', 'Card', 'Visa');

INSERT INTO Supervisor\_Info VALUES (101, 7);

INSERT INTO Supervisor\_Info VALUES (102, 5);

INSERT INTO Supervisor\_Info VALUES (103, 8);

Table 15

INSERT INTO Ticket\_Supervisor VALUES (1001, 'Confirmed', 500, 'A1', 101);

INSERT INTO Ticket\_Supervisor VALUES (1002, 'Confirmed', 400, 'B2', 102);

INSERT INTO Ticket\_Supervisor VALUES (1003, 'Cancelled', 450, 'C3', 103);

1. **Query Writing (Write down the question and the answer. Give full screenshot of the Oracle 10g Homepage that contains the answer and result)**

**-All screenshots MUST include the DATE and TIME feature from the screen of the machine (PC, Laptop etc.) used**

**SQL**

**-2 single-row function**

**-2 group function**

**-2 subquery**

**-2 joining**

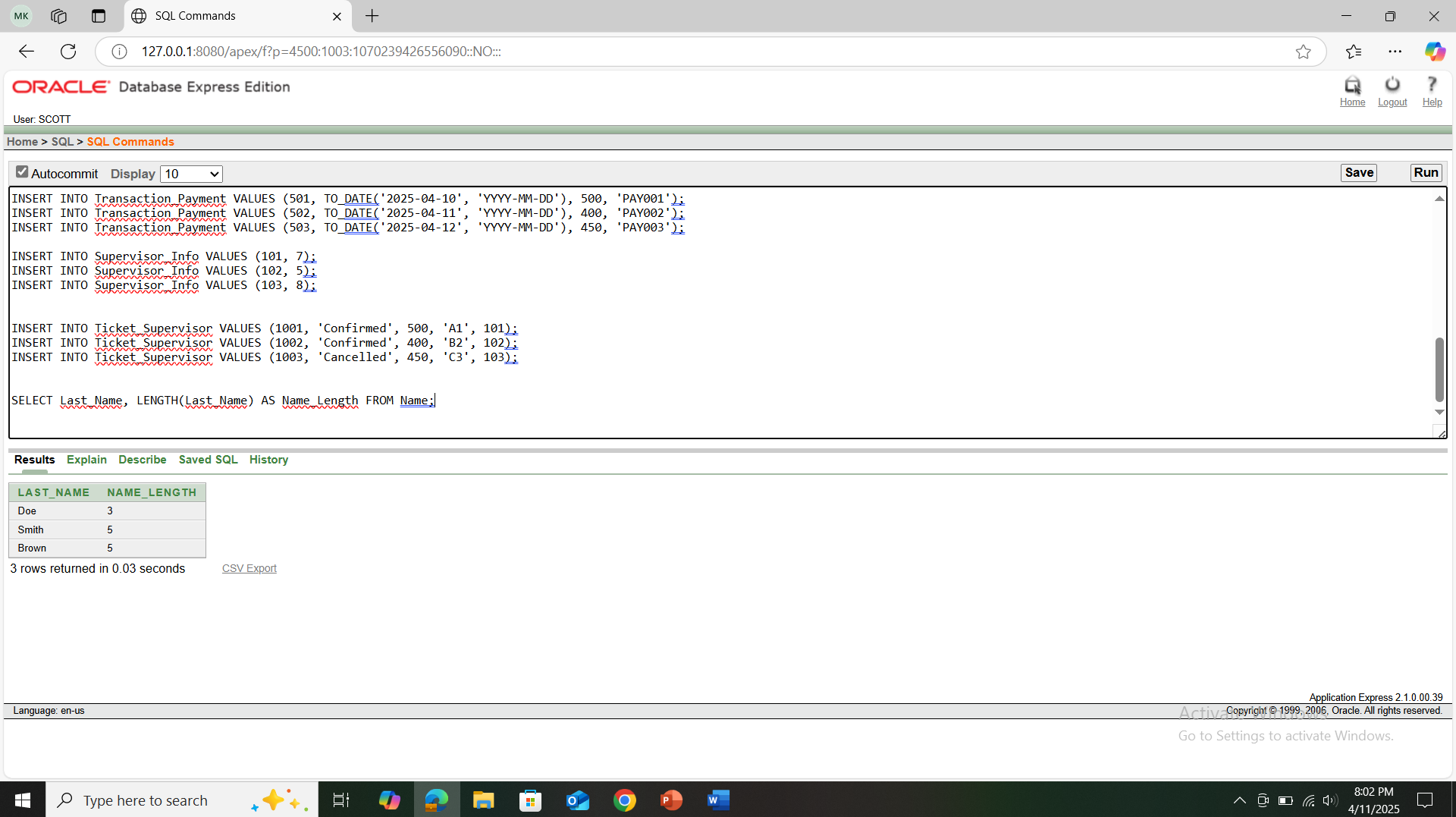
***For reference see BasicSQLTutorial and AdvanceSQLTutorial.***

Answer 4:

**-2 single-row function**

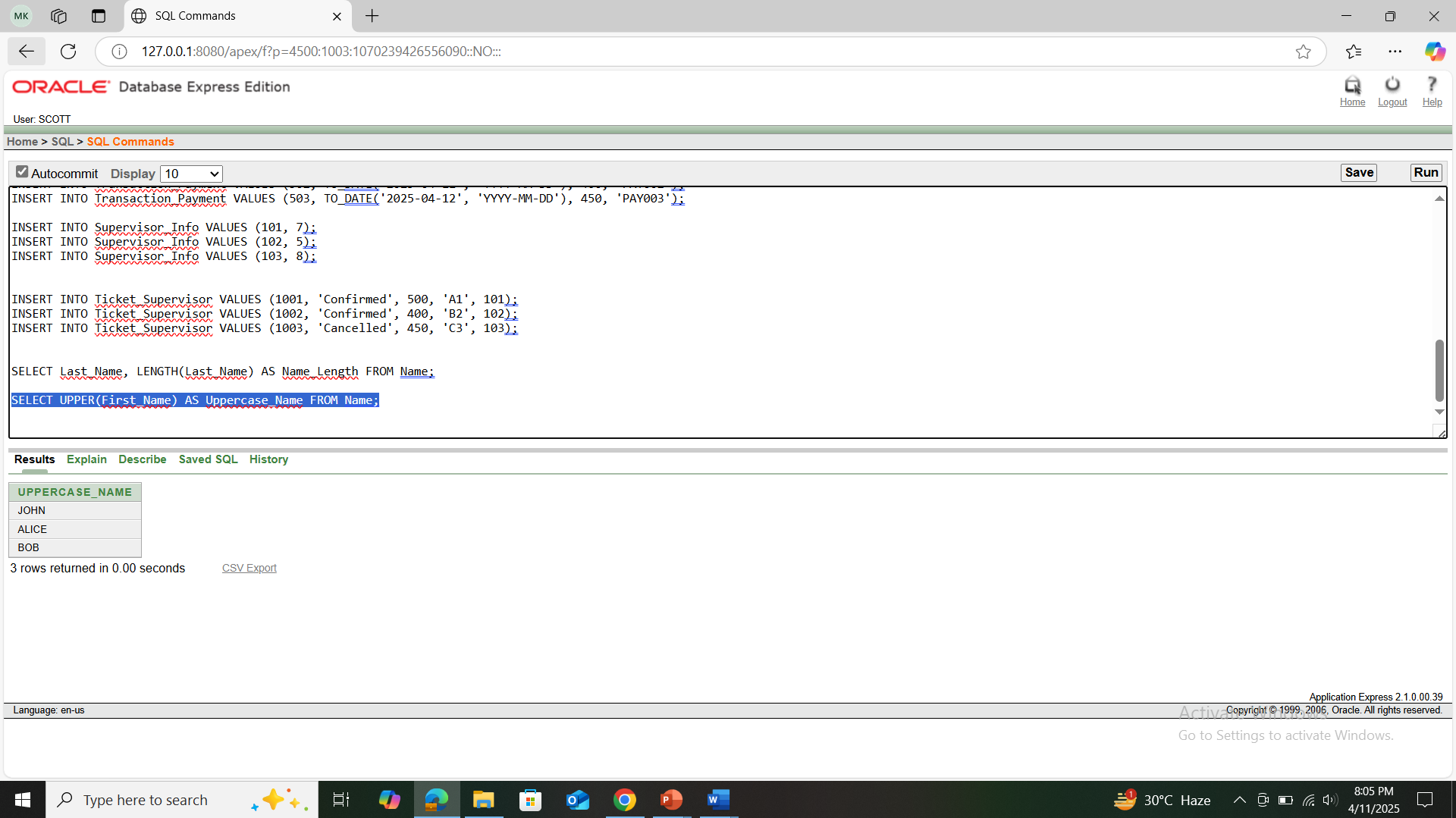
**Question-1:** Display the length of each last name from the Name table.

**Query**: SELECT Last\_Name, LENGTH(Last\_Name) AS Name\_Length FROM Name;



**Question-2:** Show the uppercase version of all first names from the Name table.

**Query:** SELECT UPPER(First\_Name) AS Uppercase\_Name FROM Name;



**-2 group function**

**Question-1:** Find the average ticket price from the Ticket\_Bus table.

**Query:** SELECT AVG(Ticket\_Price) AS Average\_Price FROM Ticket\_Bus;

A screenshot of a computer

AI-generated content may be incorrect.

**Question-2:** Count how many tickets have been issued in the Ticket\_Bus table.

**Query:** SELECT COUNT(\*) AS Total\_Tickets FROM Ticket\_Bus;

A screenshot of a computer

AI-generated content may be incorrect.

**-2 subquery**

**Question-1:** Show the Supervisor IDs from the Supervisor table whose experience is greater than the average experience from the Supervisor\_Info table.

**Query:** SELECT Supervisor\_Id FROM Supervisor WHERE Experience\_Year > (SELECT AVG(Experience\_Year)

FROM Supervisor\_Info);

A screenshot of a computer

AI-generated content may be incorrect.

**Question-2:** Display the Supervisor IDs and their experience from the Supervisor table where the Supervisor ID also exists in the Supervisor\_Info table.

**Query:** SELECT Supervisor\_Id, Experience\_Year FROM Supervisor WHERE Supervisor\_Id IN (SELECT Supervisor\_Id FROM Supervisor\_Info);

A screenshot of a computer

AI-generated content may be incorrect.

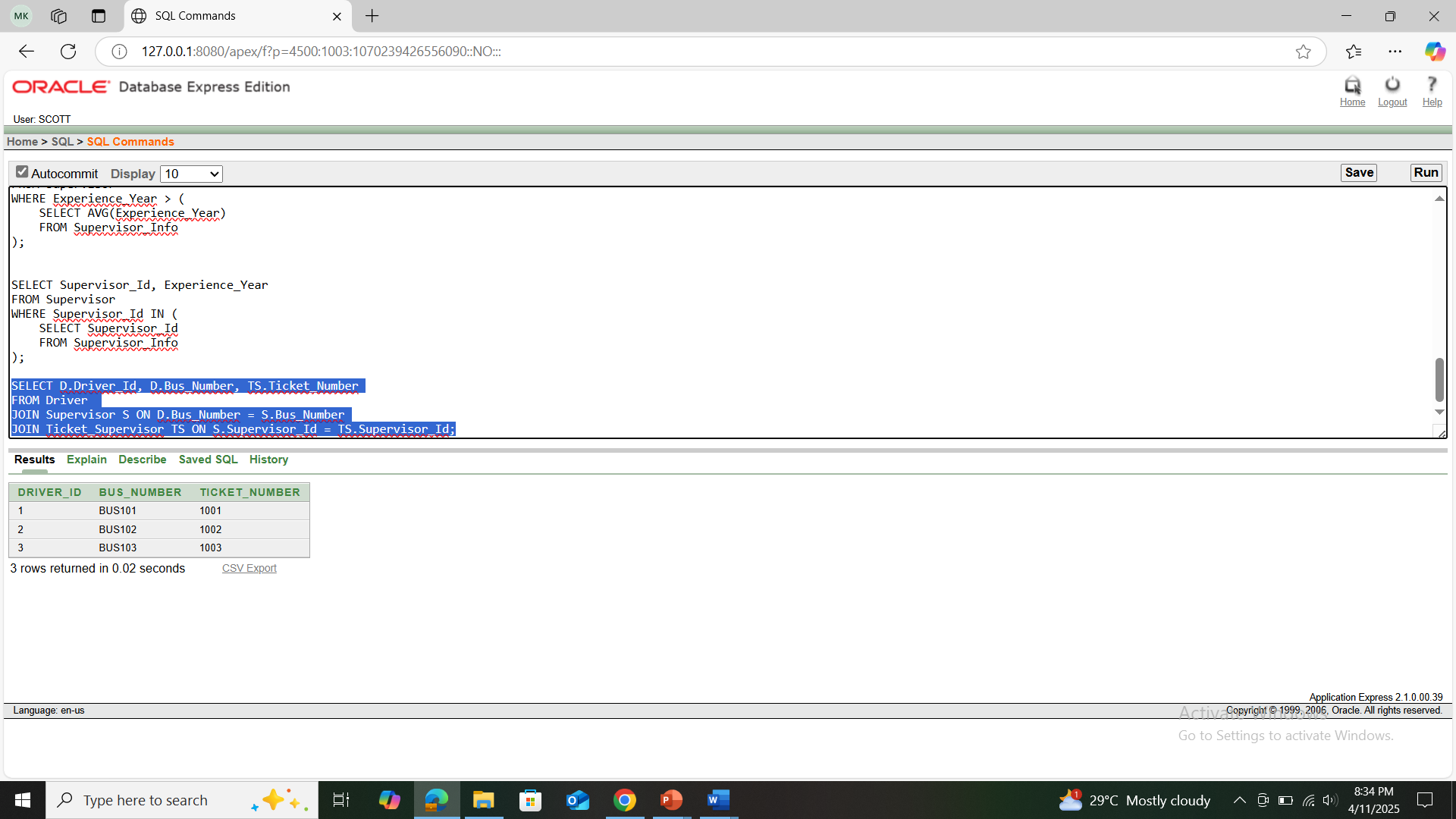
**-2 joining**

**Question-1:** Display the Driver ID, their Experience Year, and the Ticket Status from the Ticket\_Supervisor table.

**Query:** SELECT D.Driver\_Id, D.Bus\_Number, TS.Ticket\_Number

FROM Driver JOIN Supervisor S ON D.Bus\_Number = S.Bus\_Number

JOIN Ticket\_Supervisor TS ON S.Supervisor\_Id = TS.Supervisor\_Id;



**Question-2:** Show the Driver ID, License Number, Ticket Number, Ticket Price, and Seat Number.

**Query:** SELECT D.Driver\_Id, D.License\_Number, TS.Ticket\_Number, TS.Ticket\_Price, TS.Seat\_Number

FROM Driver D

JOIN Supervisor S ON D.Bus\_Number = S.Bus\_Number

JOIN Ticket\_Supervisor TS ON S.Supervisor\_Id = TS.Supervisor\_Id;