|  |  |  |
| --- | --- | --- |
| **Name** | **ID** | **STUDENT SIGN** |
| FARJANA YESMIN OPI | 22-47018-1 |  |

**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.**

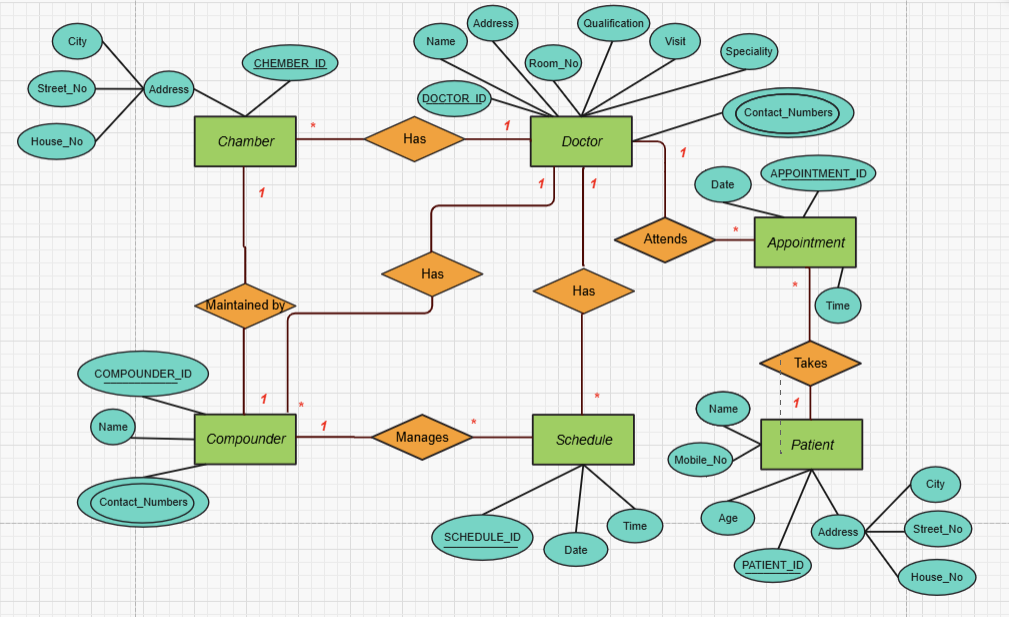
**Midterm Assignment**

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:

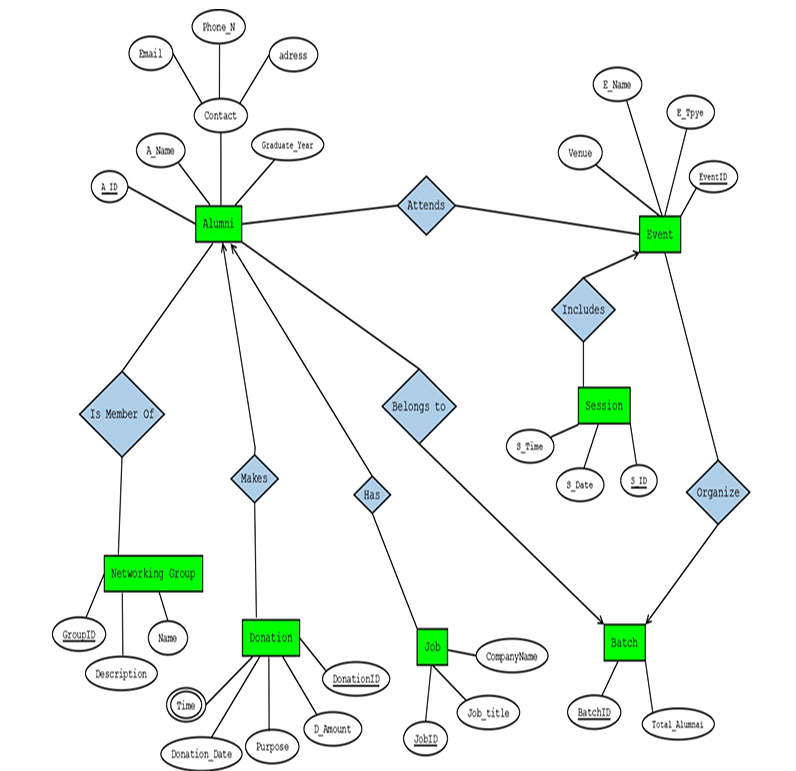


1. **Below an ER Diagram has been given write the scenario.**

***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 designed to manage alumni details, their participation in events, donations, networking groups, jobs, and batch records. An alumni member is identified by an alumni ID. The system also stores alumni name, graduation year, contact details, which include email, phone number, and address. An alumni member belongs to exactly one batch, and a batch consists of many alumni members. A batch is identified by a batch ID, and the system also stores the total number of alumni in the batch. A batch organizes multiple events, and an event is associated with multiple batches. Alumni attends events. An event is identified by an event ID. The system also stores the event name, event type, and venue. An alumni member may attend many events, and an event may be attended by many alumni members. An event includes one or more sessions, and each session has a session ID, session date, and session time. Session is identified by a session ID. A session belongs to exactly one event. An alumni member may make multiple donations, and a donation is uniquely identified by a donation ID. The system also stores the donation amount, donation date, time and purpose. An alumni member is member of multiple networking groups, and a networking group consists of multiple alumni members. A networking group is identified by a group ID, and the system stores the group name and description. An alumni member has jobs, and a job is uniquely identified by a job ID. The system stores the job title and company name. An alumni member may have multiple jobs.

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):

**Answer:**

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, N\_Id
2. N\_Id, First\_Name, Last\_Name
3. Bus\_Number, Type, Model, Capacity

Follows:

**UNF:**

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

**2NF:**

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

**3NF:**

Destination is transitive attribute.

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

**Table Creation:**

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

Conducts

**UNF:**

Conducts( 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, N\_Id
3. N\_Id, 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, N\_Id
2. National\_Id, Gender, Phone\_Number
3. N\_Id, 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. National\_Id, Gender, Phone\_Number
2. First\_Name, Last\_Name,
3. Transaction\_Id, Date, Total\_Amount

**Table Creation:**

1. National\_Id, Gender, Phone\_Number
2. N\_Id, First\_Name, Last\_Name
3. Transaction\_Id, Date, Total\_Amount, National\_Id, N\_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
2. Payment\_Code, Payment\_Method, Payment\_Type, Transaction\_Id

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. N\_Id, First\_Name, Last\_Name.
3. Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number, Supervisor\_Id, N\_Id

**Temporary Tables:**

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

**Final Tables:**

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

**Table Creation:**

**1. Driver Table:**

CREATE TABLE Driver (

Driver\_Id INT PRIMARY KEY,

License\_Number VARCHAR(50),

Experience\_Year INT,

Bus\_Number VARCHAR(20),

N\_Id VARCHAR(20)

);

**2. Person Table**

CREATE TABLE Person (

N\_Id VARCHAR(20) PRIMARY KEY,

First\_Name VARCHAR(50),

Last\_Name VARCHAR(50)

);

**3. Bus Table**

CREATE TABLE Bus (

Bus\_Number VARCHAR(20) PRIMARY KEY,

Type VARCHAR(30),

Model VARCHAR(50),

Capacity INT

);

**4. Route Table**

CREATE TABLE Route (

Route\_Id NUMBER PRIMARY KEY,

Distance NUMBER,

Departure\_Time DATE,

Bus\_Number VARCHAR2(20),

L\_Id NUMBER,

FOREIGN KEY (L\_Id) REFERENCES Location(L\_Id)

);

**5.Location Table**

CREATE TABLE Location (

L\_Id INT PRIMARY KEY,

Start\_Location VARCHAR(100),

End\_Location VARCHAR(100)

);

**6. Supervisor Table**

CREATE TABLE Supervisor (

Supervisor\_Id NUMBER PRIMARY KEY,

Experience\_Year NUMBER,

Bus\_Number VARCHAR2(20),

N\_Id VARCHAR2(20),

FOREIGN KEY (N\_Id) REFERENCES Person(N\_Id)

);

**7. Ticket Table**

CREATE TABLE Ticket (

Ticket\_Number INT PRIMARY KEY,

Ticket\_Status VARCHAR(20),

Ticket\_Price DECIMAL(10, 2),

Seat\_Number INT,

Bus\_Number VARCHAR(20)

);

**8. Ticket\_Person Table**

CREATE TABLE Ticket\_Person (

Ticket\_Number NUMBER,

Ticket\_Status VARCHAR2(20),

Ticket\_Price NUMBER(10, 2),

Seat\_Number NUMBER,

National\_Id VARCHAR2(20),

N\_Id VARCHAR2(20),

FOREIGN KEY (N\_Id) REFERENCES Person(N\_Id),

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

);

**9. Nationality Table**

CREATE TABLE National (

National\_Id VARCHAR(20) PRIMARY KEY,

Gender VARCHAR(10),

Phone\_Number VARCHAR(15)

);

**10. Transaction\_Record Table**

CREATE TABLE Transaction\_Record (

Transaction\_Id NUMBER PRIMARY KEY,

Transaction\_Date DATE,

Total\_Amount NUMBER(10, 2),

National\_Id VARCHAR2(20),

N\_Id VARCHAR2(20),

FOREIGN KEY (N\_Id) REFERENCES Person(N\_Id),

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

);

**11. Ticket\_Transaction Table**

CREATE TABLE Ticket\_Transaction (

Ticket\_Number NUMBER,

Ticket\_Status VARCHAR2(20),

Ticket\_Price NUMBER(10, 2),

Seat\_Number NUMBER,

Transaction\_Id NUMBER,

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

);

**12. Transaction\_Info Table**

CREATE TABLE Transaction\_Info (

Transaction\_Id NUMBER PRIMARY KEY,

Transaction\_Date DATE,

Total\_Amount NUMBER(10, 2)

);

**13. Payment Table**

CREATE TABLE Payment (

Payment\_Code NUMBER PRIMARY KEY,

Payment\_Method VARCHAR2(50),

Payment\_Type VARCHAR2(50),

Transaction\_Id NUMBER,

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

);

**14. Supervisor\_Info Table**

CREATE TABLE Supervisor\_Info (

Supervisor\_Id INT PRIMARY KEY,

Experience\_Year INT

);

**15. Ticket\_Supervisor Table**

CREATE TABLE Ticket\_Supervisor (

Ticket\_Number NUMBER,

Ticket\_Status VARCHAR2(20),

Ticket\_Price NUMBER(10, 2),

Seat\_Number NUMBER,

Supervisor\_Id NUMBER,

N\_Id VARCHAR2(20),

FOREIGN KEY (N\_Id) REFERENCES Person(N\_Id),

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

);

**Value Insert:**

**1. Driver Table**

INSERT INTO Driver (Driver\_Id, License\_Number, Experience\_Year, Bus\_Number, N\_Id)

VALUES (1, 'LIC1001', 5, 'BUS101', 'P001');

INSERT INTO Driver (Driver\_Id, License\_Number, Experience\_Year, Bus\_Number, N\_Id)

VALUES (2, 'LIC1002', 7, 'BUS102', 'P002');

INSERT INTO Driver (Driver\_Id, License\_Number, Experience\_Year, Bus\_Number, N\_Id)

VALUES (3, 'LIC1003', 4, 'BUS103', 'P003');

**2. Person Table**

INSERT INTO Person (N\_Id, First\_Name, Last\_Name) VALUES ('P001', 'Ali', 'Hassan');

INSERT INTO Person (N\_Id, First\_Name, Last\_Name) VALUES ('P002', 'Sara', 'Rahman');

INSERT INTO Person (N\_Id, First\_Name, Last\_Name) VALUES ('P003', 'Kamal', 'Uddin');

**3. Bus Table**

INSERT INTO Bus (Bus\_Number, Type, Model, Capacity) VALUES ('BUS101', 'AC', 'Volvo', 40);

INSERT INTO Bus (Bus\_Number, Type, Model, Capacity) VALUES ('BUS102', 'Non-AC', 'Isuzu', 35);

INSERT INTO Bus (Bus\_Number, Type, Model, Capacity) VALUES ('BUS103', 'AC', 'Hyundai', 45);

**4. Route Table**

INSERT INTO Route (Route\_Id, Distance, Departure\_Time, Bus\_Number, L\_Id)

VALUES (101, 250, TO\_DATE('2025-04-10 08:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'BUS101', 1);

INSERT INTO Route (Route\_Id, Distance, Departure\_Time, Bus\_Number, L\_Id)

VALUES (102, 300, TO\_DATE('2025-04-11 07:30:00', 'YYYY-MM-DD HH24:MI:SS'), 'BUS102', 2);

INSERT INTO Route (Route\_Id, Distance, Departure\_Time, Bus\_Number, L\_Id)

VALUES (103, 180, TO\_DATE('2025-04-12 09:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'BUS103', 3);

**5.Location Table**

INSERT INTO Location (L\_Id, Start\_Location, End\_Location) VALUES (1, 'Dhaka', 'Chittagong');

INSERT INTO Location (L\_Id, Start\_Location, End\_Location) VALUES (2, 'Sylhet', 'Rajshahi');

INSERT INTO Location (L\_Id, Start\_Location, End\_Location) VALUES (3, 'Khulna', 'Barisal');

**6. Supervisor Table**

INSERT INTO Supervisor (Supervisor\_Id, Experience\_Year, Bus\_Number, N\_Id)

VALUES (201, 8, 'BUS101', 'P001');

INSERT INTO Supervisor (Supervisor\_Id, Experience\_Year, Bus\_Number, N\_Id)

VALUES (202, 5, 'BUS102', 'P002');

INSERT INTO Supervisor (Supervisor\_Id, Experience\_Year, Bus\_Number, N\_Id)

VALUES (203, 10, 'BUS103', 'P003');

**7. Ticket Table**

INSERT INTO Ticket (Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number, Bus\_Number)

VALUES (1001, 'Confirmed', 500.00, 1, 'BUS101');

INSERT INTO Ticket (Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number, Bus\_Number)

VALUES (1002, 'Pending', 450.00, 2, 'BUS102');

INSERT INTO Ticket (Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number, Bus\_Number)

VALUES (1003, 'Cancelled', 600.00, 3, 'BUS103');

**8. Ticket\_Person Table**

INSERT INTO Ticket\_Person (Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number, National\_Id, N\_Id)

VALUES (1001, 'Confirmed', 500.00, 1, 'NAT001', 'P001');

INSERT INTO Ticket\_Person (Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number, National\_Id, N\_Id)

VALUES (1002, 'Pending', 450.00, 2, 'NAT002', 'P002');

INSERT INTO Ticket\_Person (Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number, National\_Id, N\_Id)

VALUES (1003, 'Cancelled', 600.00, 3, 'NAT003', 'P003');

**9. Nationality Table**

INSERT INTO Nationality (National\_Id, Gender, Phone\_Number) VALUES ('NAT001', 'Male', '0123456789');

INSERT INTO Nationality (National\_Id, Gender, Phone\_Number) VALUES ('NAT002', 'Female', '0987654321');

INSERT INTO Nationality (National\_Id, Gender, Phone\_Number) VALUES ('NAT003', 'Male', '0112233445');

**10. Transaction\_Record Table**

INSERT INTO Transaction\_Record (Transaction\_Id, Transaction\_Date, Total\_Amount, National\_Id, N\_Id)

VALUES (301, TO\_DATE('2025-04-10', 'YYYY-MM-DD'), 500.00, 'NAT001', 'P001');

INSERT INTO Transaction\_Record (Transaction\_Id, Transaction\_Date, Total\_Amount, National\_Id, N\_Id)

VALUES (302, TO\_DATE('2025-04-11', 'YYYY-MM-DD'), 450.00, 'NAT002', 'P002');

INSERT INTO Transaction\_Record (Transaction\_Id, Transaction\_Date, Total\_Amount, National\_Id, N\_Id)

VALUES (303, TO\_DATE('2025-04-12', 'YYYY-MM-DD'), 600.00, 'NAT003', 'P003');

**11. Ticket\_Transaction Table**

INSERT INTO Ticket\_Transaction (Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number, Transaction\_Id)

VALUES (1001, 'Confirmed', 500.00, 1, 301);

INSERT INTO Ticket\_Transaction (Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number, Transaction\_Id)

VALUES (1002, 'Pending', 450.00, 2, 302);

INSERT INTO Ticket\_Transaction (Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number, Transaction\_Id)

VALUES (1003, 'Cancelled', 600.00, 3, 303);

**12. Transaction\_Info Table**

INSERT INTO Transaction\_Info (Transaction\_Id, Transaction\_Date, Total\_Amount)

VALUES (301, TO\_DATE('2025-04-10', 'YYYY-MM-DD'), 500.00);

INSERT INTO Transaction\_Info (Transaction\_Id, Transaction\_Date, Total\_Amount)

VALUES (302, TO\_DATE('2025-04-11', 'YYYY-MM-DD'), 450.00);

INSERT INTO Transaction\_Info (Transaction\_Id, Transaction\_Date, Total\_Amount)

VALUES (303, TO\_DATE('2025-04-12', 'YYYY-MM-DD'), 600.00);

**13. Payment Table**

INSERT INTO Payment (Payment\_Code, Payment\_Method, Payment\_Type, Transaction\_Id)

VALUES (401, 'Bkash', 'Mobile', 301);

INSERT INTO Payment (Payment\_Code, Payment\_Method, Payment\_Type, Transaction\_Id)

VALUES (402, 'Card', 'Credit', 302);

INSERT INTO Payment (Payment\_Code, Payment\_Method, Payment\_Type, Transaction\_Id)

VALUES (403, 'Cash', 'Manual', 303);

**14. Supervisor\_Info Table**

INSERT INTO Supervisor\_Info (Supervisor\_Id, Experience\_Year)

VALUES (201, 8);

INSERT INTO Supervisor\_Info (Supervisor\_Id, Experience\_Year)

VALUES (202, 5);

INSERT INTO Supervisor\_Info (Supervisor\_Id, Experience\_Year)

VALUES (203, 10);

**15. Ticket\_Supervisor Table**

INSERT INTO Ticket\_Supervisor (Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number, Supervisor\_Id, N\_Id)

VALUES (1001, 'Confirmed', 500.00, 1, 201, 'P001');

INSERT INTO Ticket\_Supervisor (Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number, Supervisor\_Id, N\_Id)

VALUES (1002, 'Pending', 450.00, 2, 202, 'P002');

INSERT INTO Ticket\_Supervisor (Ticket\_Number, Ticket\_Status, Ticket\_Price, Seat\_Number, Supervisor\_Id, N\_Id)

VALUES (1003, 'Cancelled', 600.00, 3, 203, 'P003');

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**

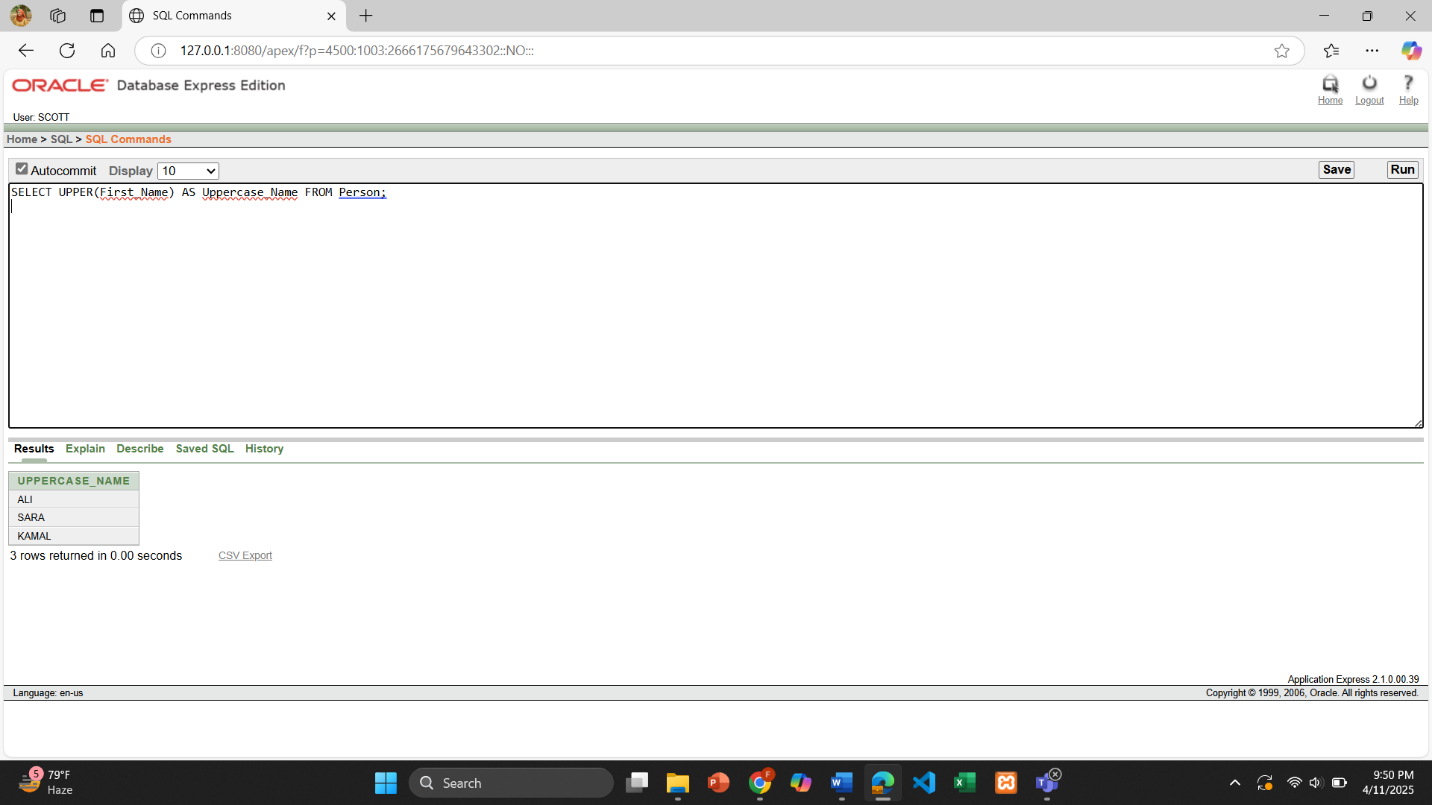
***For reference see BasicSQLTutorial and AdvanceSQLTutorial.***

Answer 4:

**-2 single-row function**

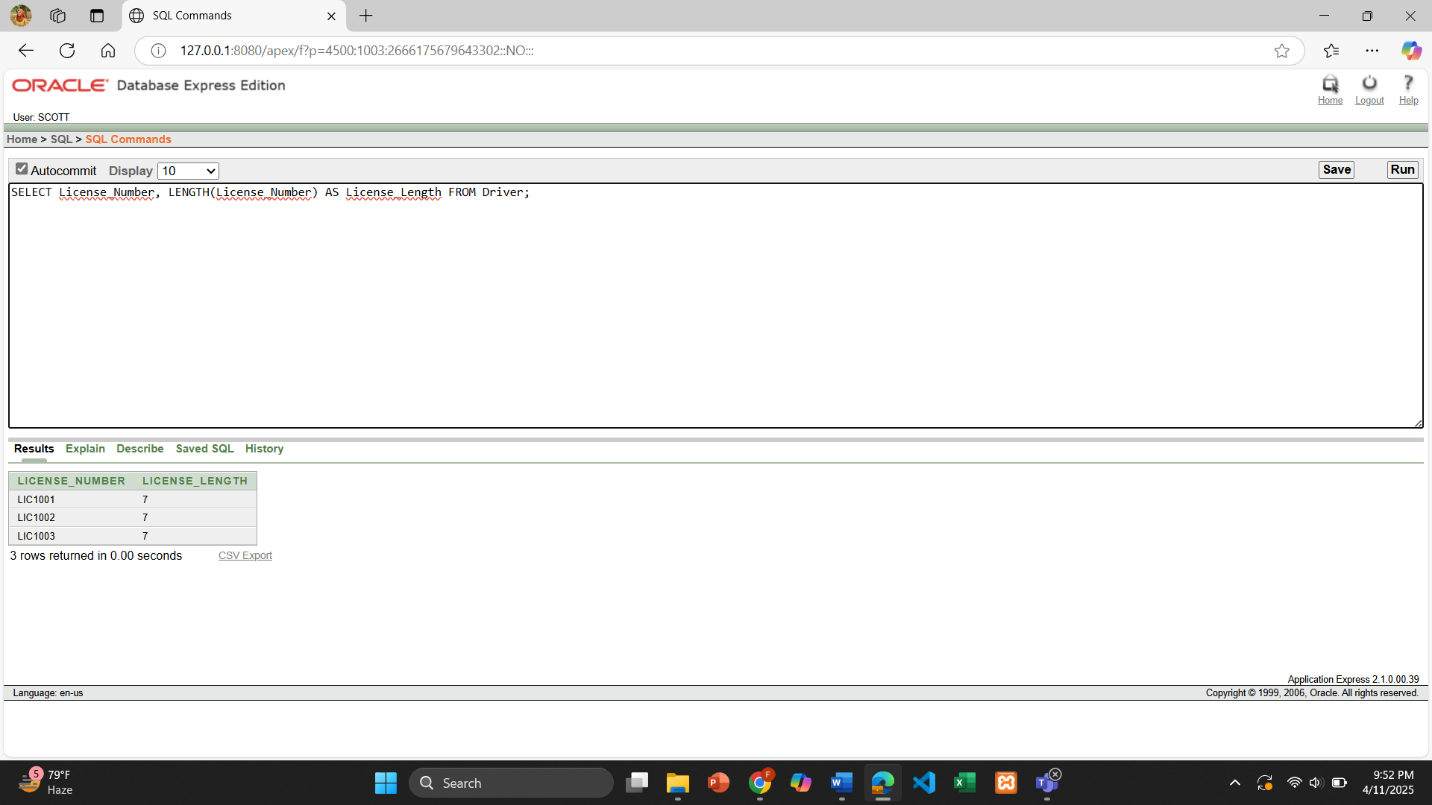
**Question:** Display the first name of all persons in uppercase.

1. SELECT UPPER(First\_Name) AS Uppercase\_Name FROM Person;



**Question:** Show the length of each driver’s license number.

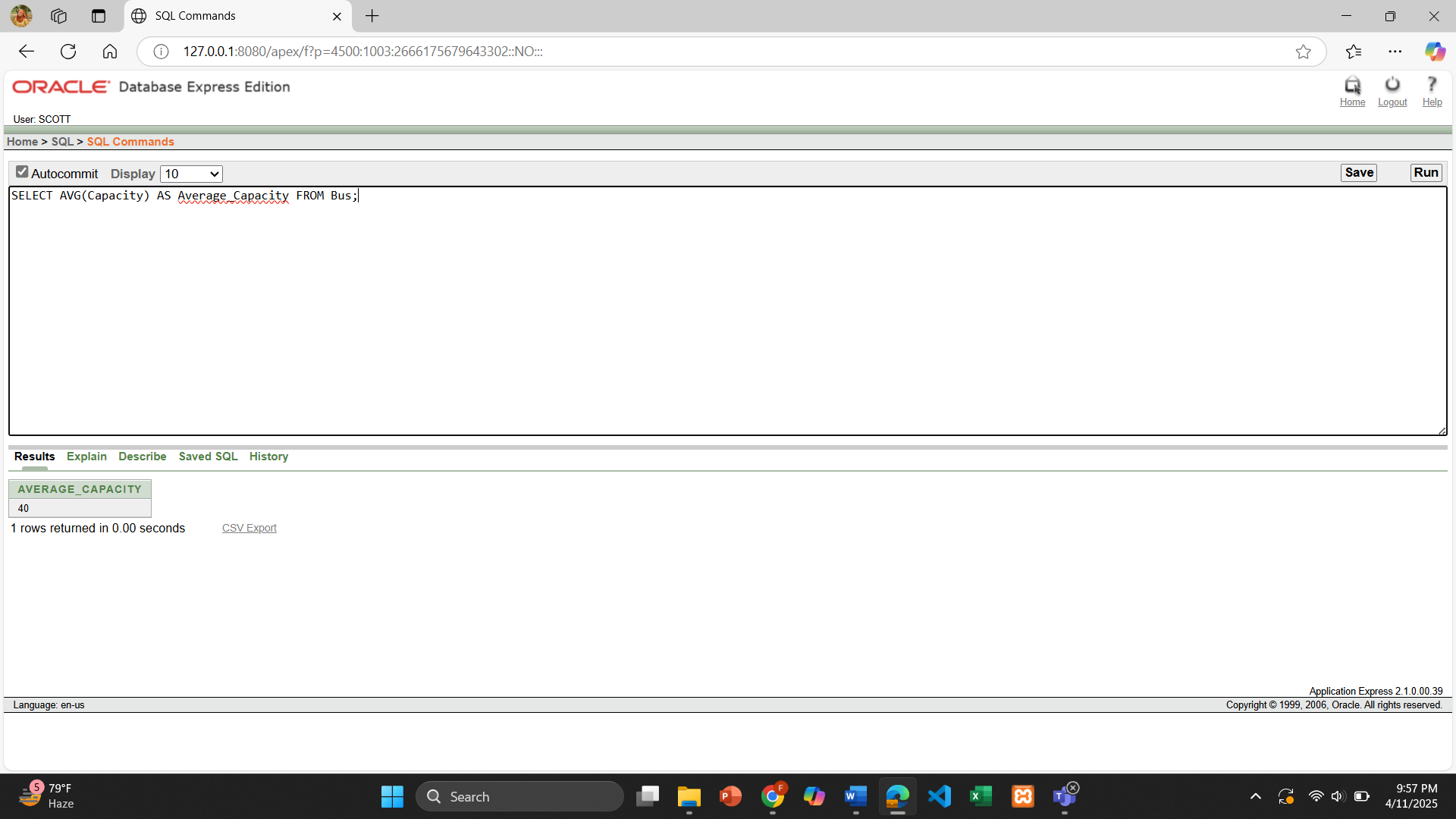
2. SELECT License\_Number, LENGTH(License\_Number) AS License\_Length FROM Driver;



**2 group function**

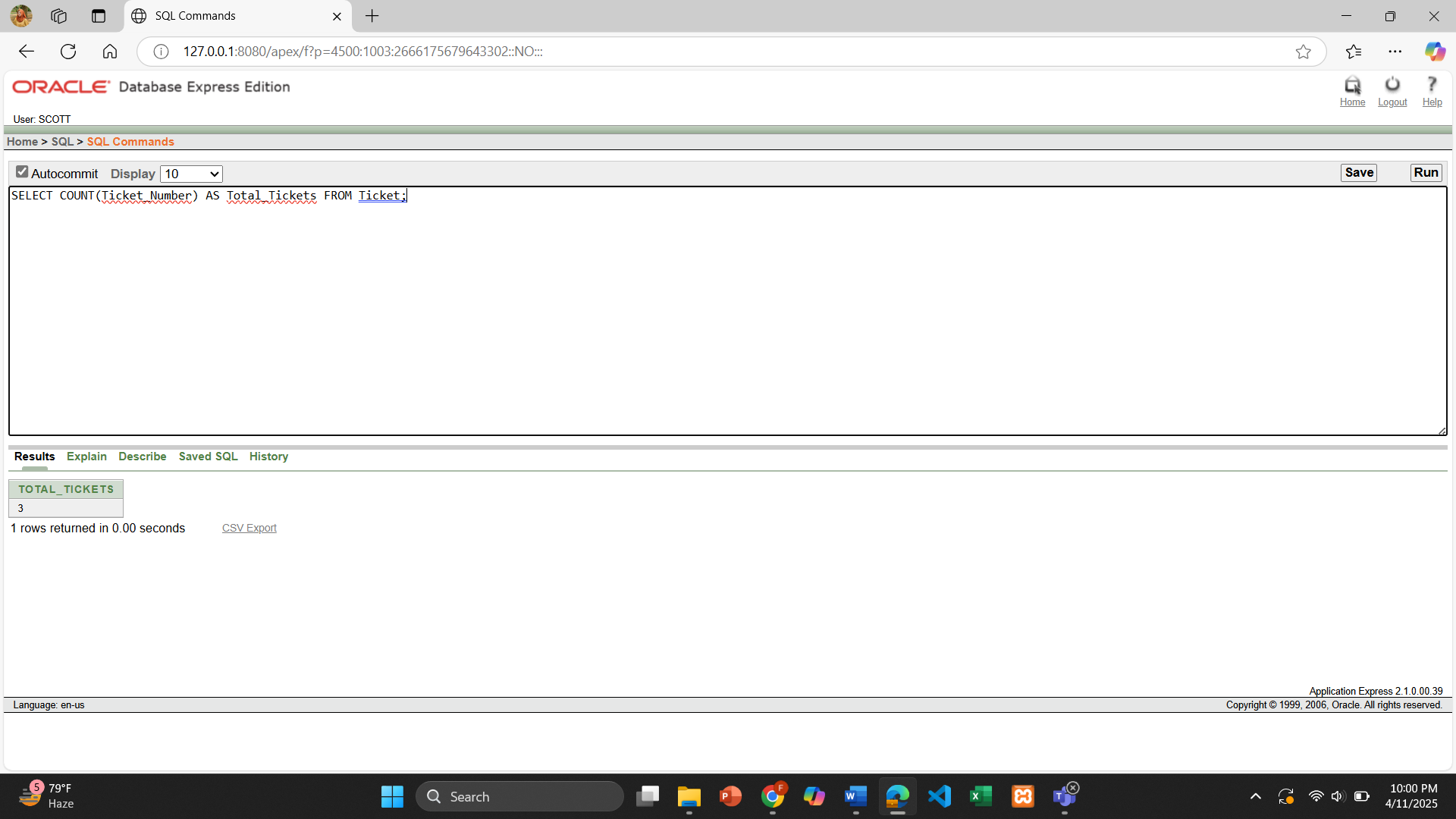
**Question:** Find the average capacity of all buses.

1. SELECT AVG(Capacity) AS Average\_Capacity FROM Bus;



**Question:** Count the total number of tickets issued.

2. SELECT COUNT(Ticket\_Number) AS Total\_Tickets FROM Ticket;



**2 subquery**

**Question :** Find the name of the person who paid the highest amount in a transaction.

1. SELECT First\_Name, Last\_Name

FROM Person

WHERE N\_Id = (

SELECT N\_Id

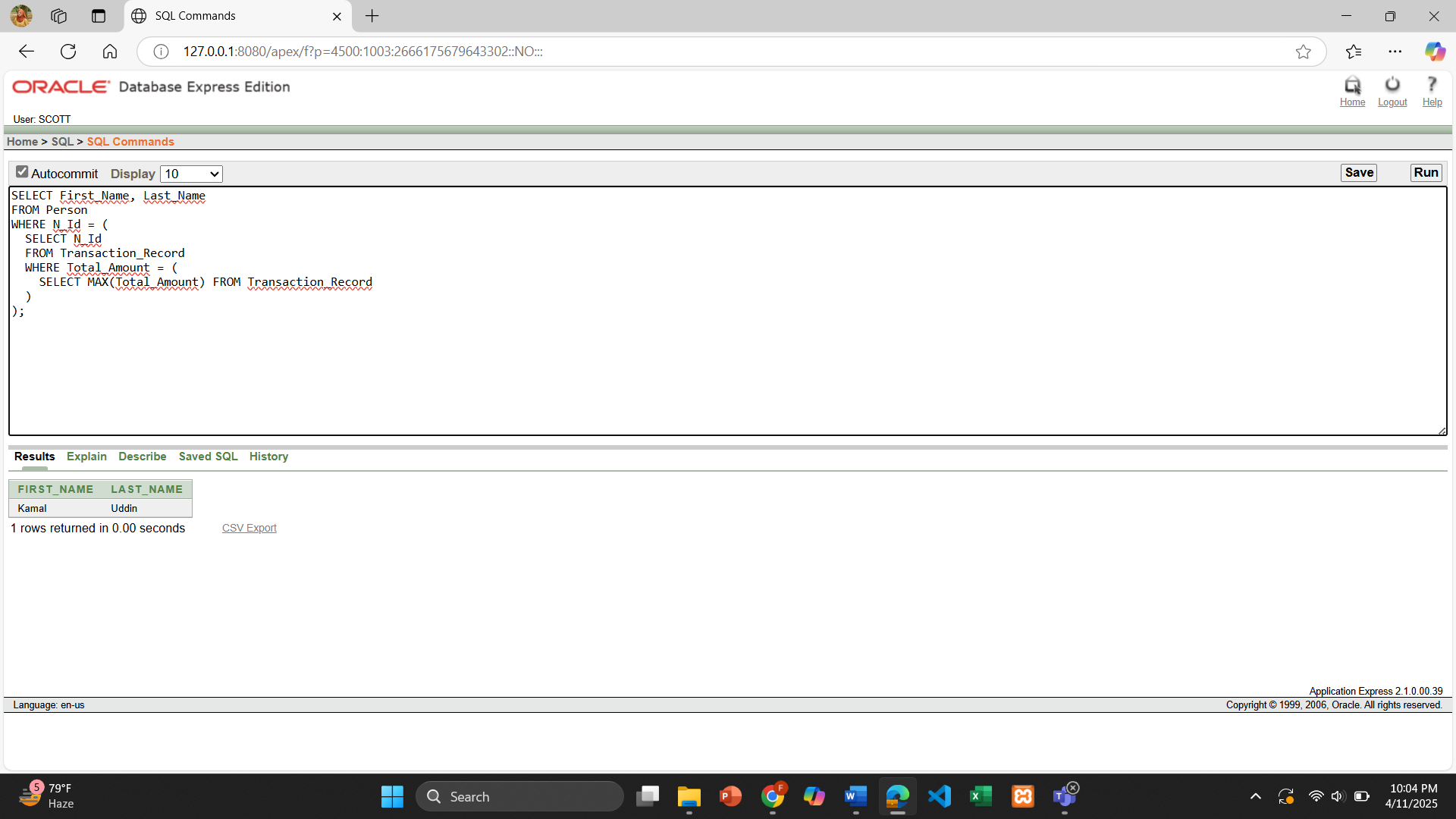
FROM Transaction\_Record

WHERE Total\_Amount = (

SELECT MAX(Total\_Amount) FROM Transaction\_Record

)

);



**Question:** Display the bus number and type of buses that have more capacity than the average bus capacity.

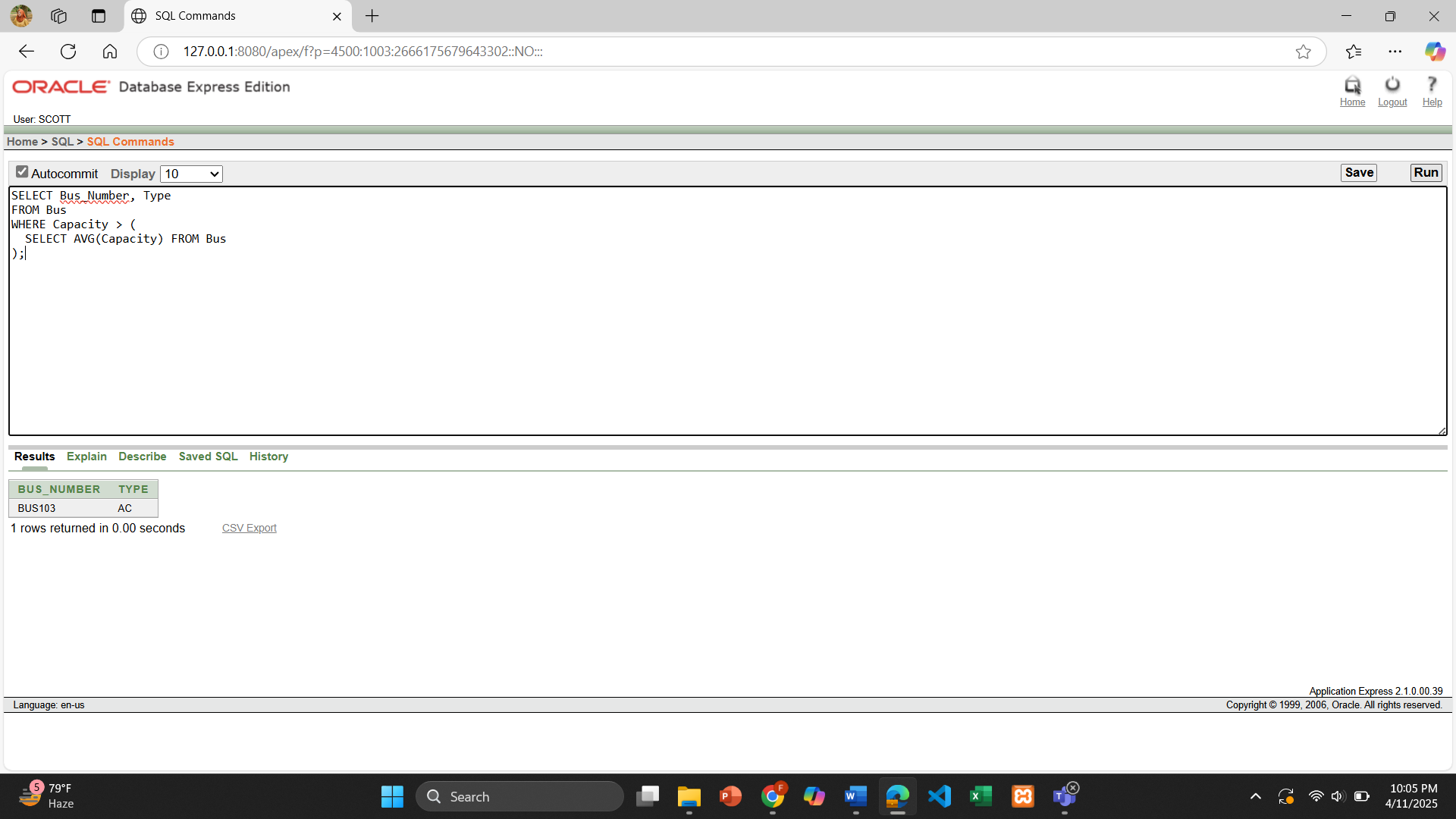
2. SELECT Bus\_Number, Type

FROM Bus

WHERE Capacity > (

SELECT AVG(Capacity) FROM Bus

);



**2 joining**

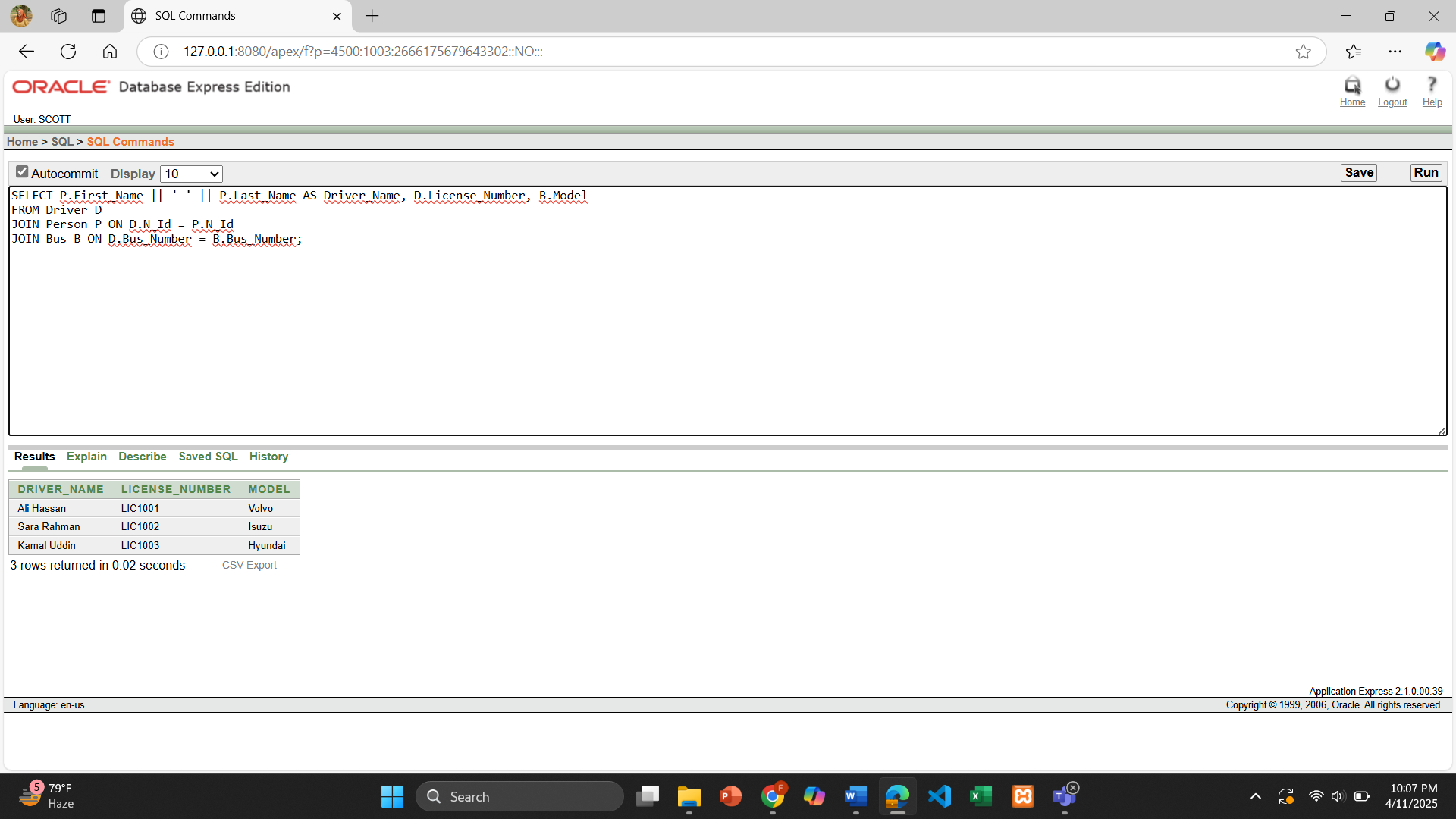
**Question:** Show the driver name, license number, and the bus model they are assigned to.

1. SELECT P.First\_Name || ' ' || P.Last\_Name AS Driver\_Name, D.License\_Number, B.Model

FROM Driver D

JOIN Person P ON D.N\_Id = P.N\_Id

JOIN Bus B ON D.Bus\_Number = B.Bus\_Number;



**Question:** Display ticket number, passenger name, and phone number.

2. SELECT TP.Ticket\_Number, P.First\_Name || ' ' || P.Last\_Name AS Passenger\_Name, N.Phone\_Number

FROM Ticket\_Person TP

JOIN Person P ON TP.N\_Id = P.N\_Id

JOIN Nationality N ON TP.National\_Id = N.National\_Id;

