|  |  |  |
| --- | --- | --- |
| **Name** | **ID** | **STUDENT SIGN** |
| **Sumaiya Tasnim** | **23-50014-1** | **Sign** |

**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.**
* **Submission Deadline: 18th August 2025, 11:59pm.**

**Midterm Assignment**

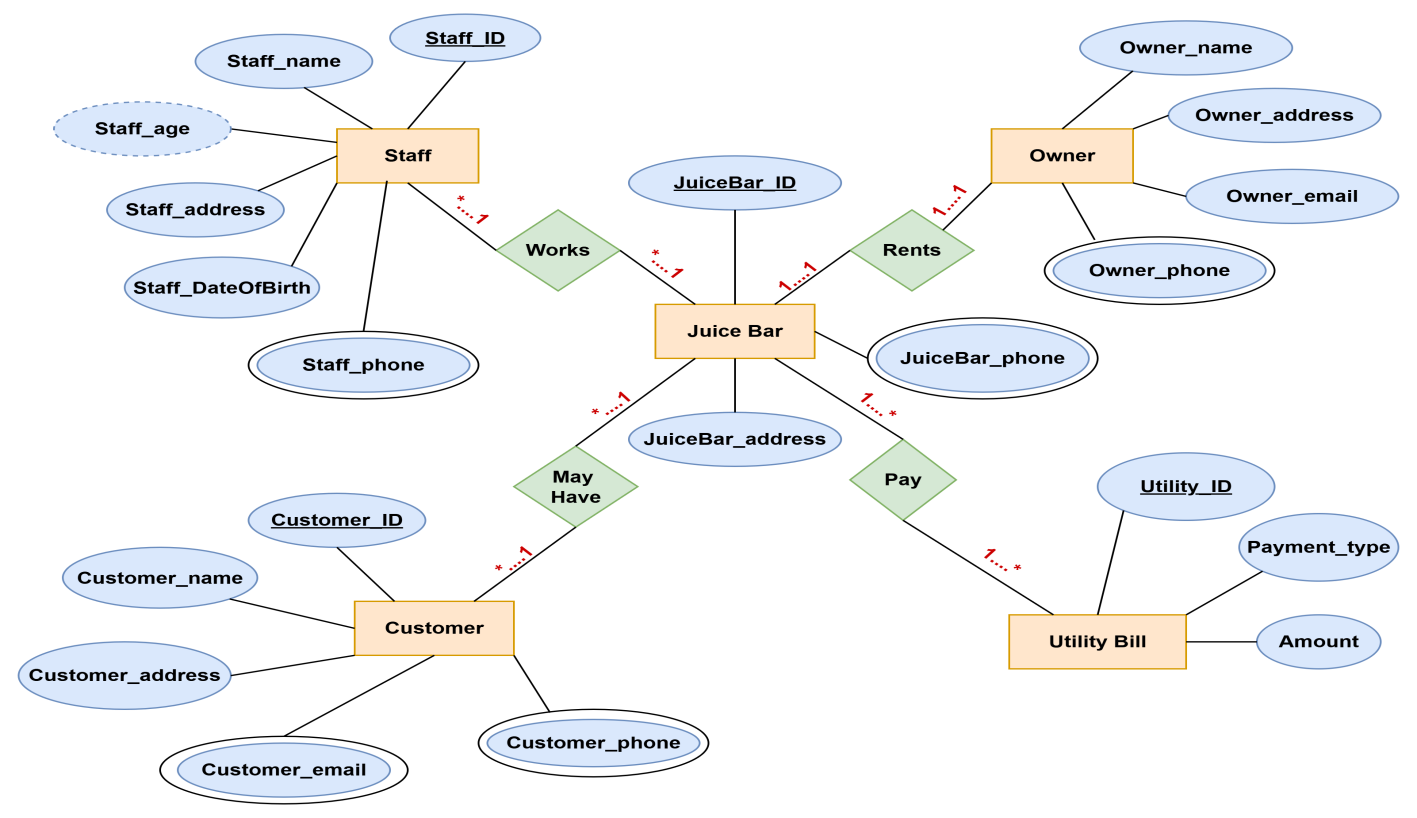
1. **Below a scenario has been given draw the ER Diagram.**

***Draw with proper annotations (use DIA, VISIO, MS WORD etc.).***

***For reference see ERDiagramTutorial.***

In a Juice Bar Management System, one Juice Bar may have many staff. But one staff can work in one Juice Bar only. Each staff has a unique identification number, name, age, address, date of birth and phone number. Juice Bar has a unique identification number, address, and phone. One Juice Bar maybe rented by exactly one owner. One owner may rent exactly one Juice Bar. Owner is defined by name, address, email and phone. A Juice Bar may have many customers. Each customer has a unique identification number, name, address, email, phone. Each customer can have more than one email address and phone number. A Juice Bar must pay the utility bill which has a unique identification number, payment\_type and amount.

Answer Box 1: ( I have used “draw.io” as tool to create the ER diagram)



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

***For reference see ERDiagramTutorial.***

A diagram of a company

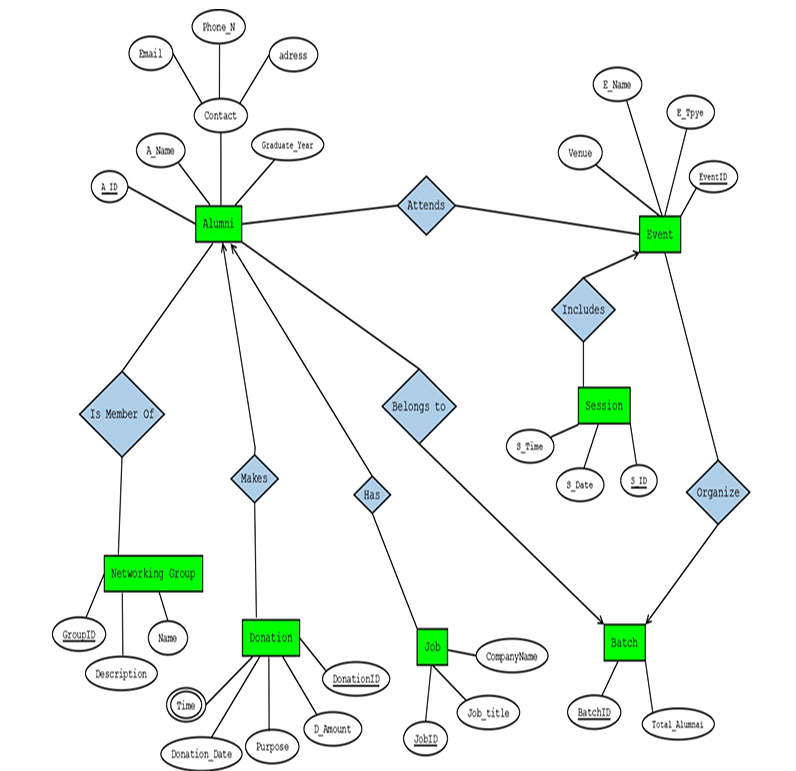
Description automatically generated

Answer Box 2:

In a Bus Ticketing Management System, one **bus** is operated by exactly one **driver**, while one **driver** can operate only one **bus**. Each driver has a unique driver ID, name, license number, first name, last name, and years of experience. Each bus has a unique bus number and is defined by its type, model, and capacity. A **bus** follows exactly one **route**, and a **route** can be followed by buses. A **route** has a unique route ID and is defined by start location, end location, destination, distance, and departure time.Each **bus** is conducted by exactly one **supervisor**, and one supervisor can conduct only one bus. A **supervisor** is uniquely identified by supervisor ID and is described by name, first name, last name, and experience year. A **supervisor** issues many **tickets**, but each **ticket** is issued by only one supervisor. A **ticket** is assigned to exactly one bus and contains details such as ticket number, ticket price, ticket status, and seat number.A **ticket** is bought by one or more **passengers**, and each **passenger** can buy multiple **tickets**. Each passenger is uniquely identified by their national ID and is described by name, gender, phone number, first name, and last name.A **passenger** makes one or more **transactions**, and each transaction is made by one passenger only. A **transaction** is uniquely identified by a transaction ID and is described by date and ticket amount. Each transaction uses exactly one **payment type**, but a payment type can be used in many transactions. A **payment type** is defined by its payment code, method, and type.

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.***



Answer Box 3 (Normalization steps in detail as shown in Normalization Tutorial Slide + all the queries required to create the tables and insert data after Normalization):

**Attends**

**UNF**

Alumni(A\_ID ,A\_name, Contact, address, Phone\_N, Email, Graduate\_Year, Venue, E\_Name, E\_Type, Event\_ID )

**1NF**

There is no multi valued attribute. Relation is already in 1NF.

1. A\_ID ,A\_name, Contact, address, Phone\_N, Email, Graduate\_Year, Venue, E\_Name, E\_Type, Event\_ID

**2NF**

1. A\_ID ,A\_name, Contact, address, Phone\_N, Email, Graduate\_Year
2. Venue, E\_Name, E\_Type, Event\_ID

**3NF**

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

1. A\_ID ,A\_name, Contact, address, Phone\_N, Email, Graduate\_Year
2. Venue, E\_Name, E\_Type, Event\_ID

**Table Creation**

1. A\_ID ,A\_name, Contact, address, Phone\_N, Email, Graduate\_Year
2. Venue, E\_Name, E\_Type, Event\_ID, **A\_ID**

### ****Is Member Of****

**UNF**  
Alumni(A\_ID, A\_Name, Contact, Address, Phone\_N, Email, Graduate\_Year, GroupID, Group\_Name, Group\_Description)

**1NF**  
There is no multi-valued attribute. Relation is already in 1NF.

1. A\_ID, A\_Name, Contact, Address, Phone\_N, Email, Graduate\_Year, GroupID, Group\_Name, Group\_Description

**2NF**

1. A\_ID, A\_Name, Contact, Address, Phone\_N, Email, Graduate\_Year

2. GroupID, Group\_Name, Group\_Description

**3NF**  
There is no transitive dependency. Relation already in 3NF.

1. A\_ID, A\_Name, Contact, Address, Phone\_N, Email, Graduate\_Year

2. GroupID, Group\_Name, Group\_Description

**Table Creation**

1. A\_ID, A\_Name, Contact, Address, Phone\_N, Email, Graduate\_Year

2. GroupID, Group\_Name, Group\_Description, **A\_ID**

### ****Makes****

**UNF**  
Alumni(A\_ID, A\_Name, Contact, Address, Phone\_N, Email, Graduate\_Year, DonationID, Donation\_Date, Time, Purpose, D\_Amount)

**1NF**  
Time is a multi valued attribute.

1. A\_ID, A\_Name, Contact, Address, Phone\_N, Email, Graduate\_Year, DonationID, Donation\_Date, Time, Purpose, D\_Amount

**2NF**

1. A\_ID, A\_Name, Contact, Address, Phone\_N, Email, Graduate\_Year

2. DonationID, Donation\_Date, Time, Purpose, D\_Amount

**3NF**  
There is no transitive dependency. Relation already in 3NF.

1. A\_ID, A\_Name, Contact, Address, Phone\_N, Email, Graduate\_Year
2. DonationID, Donation\_Date, Time, Purpose, D\_Amount

**Table Creation**

1. A\_ID, A\_Name, Contact, Address, Phone\_N, Email, Graduate\_Year

2. DonationID, Donation\_Date, Time, Purpose, D\_Amount, **A\_ID**

### ****Has****

**UNF**  
Alumni(A\_ID, A\_Name, Contact, Address, Phone\_N, Email, Graduate\_Year, JobID, Job\_Title, CompanyName)

**1NF**  
There is no multi-valued attribute. Relation is already in 1NF.

1. A\_ID, A\_Name, Contact, Address, Phone\_N, Email, Graduate\_Year, JobID, Job\_Title, CompanyName

**2NF**

1. A\_ID, A\_Name, Contact, Address, Phone\_N, Email, Graduate\_Year
2. JobID, Job\_Title, CompanyName

**3NF**  
There is no transitive dependency. Relation already in 3NF.

1. A\_ID, A\_Name, Contact, Address, Phone\_N, Email, Graduate\_Year
2. JobID, Job\_Title, CompanyName

**Table Creation**

1. A\_ID, A\_Name, Contact, Address, Phone\_N, Email, Graduate\_Year
2. JobID, Job\_Title, CompanyName, **A\_ID**

### ****Belongs To****

**UNF**  
Alumni(A\_ID, A\_Name, Contact, Address, Phone\_N, Email, Graduate\_Year, BatchID, Total\_Alumni)

**1NF**  
There is no multi-valued attribute. Relation is already in 1NF.

1.A\_ID, A\_Name, Contact, Address, Phone\_N, Email, Graduate\_Year, BatchID, Total\_Alumni

**2NF**

1. A\_ID, A\_Name, Contact, Address, Phone\_N, Email, Graduate\_Year
2. BatchID, Total\_Alumni

**3NF**  
There is no transitive dependency. Relation already in 3NF.

1. A\_ID, A\_Name, Contact, Address, Phone\_N, Email, Graduate\_Year
2. BatchID, Total\_Alumni

**Table Creation**

1. A\_ID, A\_Name, Contact, Address, Phone\_N, Email, Graduate\_Year, BatchID
2. BatchID, Total\_Alumni, **A\_ID**

### ****Includes****

**UNF**  
Event(EventID, E\_Name, E\_Type, Venue, S\_ID, S\_Date, S\_Time)

**1NF**  
There is no multi-valued attribute. Relation is already in 1NF.

1. EventID, E\_Name, E\_Type, Venue, S\_ID, S\_Date, S\_Time

**2NF**

1. EventID, E\_Name, E\_Type, Venue
2. S\_ID, S\_Date, S\_Time

**3NF**  
There is no transitive dependency. Relation already in 3NF.

1. EventID, E\_Name, E\_Type, Venue
2. S\_ID, S\_Date, S\_Time

**Table Creation**

1. EventID, E\_Name, E\_Type, Venue
2. S\_ID, S\_Date, S\_Time, **EventID**

### ****Organize****

**UNF**  
Event(EventID, E\_Name, E\_Type, Venue, BatchID, Total\_Alumni)

**1NF**  
There is no multi-valued attribute. Relation is already in 1NF.

1. EventID, E\_Name, E\_Type, Venue, BatchID, Total\_Alumni

**2NF**

1. EventID, E\_Name, E\_Type, Venue
2. BatchID, Total\_Alumni

**3NF**  
There is no transitive dependency. Relation already in 3NF.

1. EventID, E\_Name, E\_Type, Venue
2. BatchID, Total\_Alumni

**Table Creation**

1. EventID, E\_Name, E\_Type, Venue
2. BatchID, Total\_Alumni, **EventID**

**Temporary Tables**

1. A\_ID ,A\_name, Contact, address, Phone\_N, Email, Graduate\_Year

2. Venue, E\_Name, E\_Type, Event\_ID, **A\_ID**

~~3. A\_ID, A\_Name, Contact, Address, Phone\_N, Email, Graduate\_Year~~

4. GroupID, Group\_Name, Group\_Description, **A\_ID**

~~5. A\_ID, A\_Name, Contact, Address, Phone\_N, Email, Graduate\_Year~~

6. DonationID, Donation\_Date, Time, Purpose, D\_Amount, **A\_ID**

~~7. A\_ID, A\_Name, Contact, Address, Phone\_N, Email, Graduate\_Year~~

8. JobID, Job\_Title, CompanyName, **A\_ID**

9. ~~A\_ID, A\_Name, Contact, Address, Phone\_N, Email, Graduate\_Year, BatchID~~

10. BatchID, Total\_Alumni, **A\_ID**

~~11. EventID, E\_Name, E\_Type, Venue~~

12. S\_ID, S\_Date, S\_Time, **EventID**

~~13. EventID, E\_Name, E\_Type, Venue~~

~~14.BatchID, Total\_Alumni,~~ **~~EventID~~**

**Final Tables**

1. A\_ID ,A\_name, Contact, address, Phone\_N, Email, Graduate\_Year
2. Venue, E\_Name, E\_Type, Event\_ID, **A\_ID**
3. GroupID, Group\_Name, Group\_Description, **A\_ID**

4. DonationID, Donation\_Date, Time, Purpose, D\_Amount, **A\_ID**

5. JobID, Job\_Title, CompanyName, **A\_ID**

6. BatchID, Total\_Alumni, **A\_ID**

7. S\_ID, S\_Date, S\_Time, **EventID**

**(In Oracle using SQL) writing the queries below that are required to create all the tables with necessary constraints :**

**-- 1. Alumni Table**

CREATE TABLE Alumni (

A\_ID NUMBER PRIMARY KEY,

A\_name VARCHAR2(100) NOT NULL,

Contact VARCHAR2(50),

Address VARCHAR2(200),

Phone\_N VARCHAR2(15) UNIQUE,

Email VARCHAR2(100) UNIQUE,

Graduate\_Year NUMBER(4) );

**-- 2. Event Table**

CREATE TABLE Event (

Event\_ID NUMBER PRIMARY KEY,

Venue VARCHAR2(100),

E\_Name VARCHAR2(100),

E\_Type VARCHAR2(50),

A\_ID NUMBER,

FOREIGN KEY (A\_ID) REFERENCES Alumni(A\_ID)

);

**-- 3. Networking Group Table**

CREATE TABLE Networking\_Group (

GroupID NUMBER PRIMARY KEY,

Group\_Name VARCHAR2(100),

Group\_Description VARCHAR2(200),

A\_ID NUMBER,

FOREIGN KEY (A\_ID) REFERENCES Alumni(A\_ID)

);

**-- 4. Donation Table**

CREATE TABLE Donation (

DonationID NUMBER PRIMARY KEY,

Donation\_Date DATE,

Time VARCHAR2(10),

Purpose VARCHAR2(200),

D\_Amount NUMBER(10,2),

A\_ID NUMBER,

FOREIGN KEY (A\_ID) REFERENCES Alumni(A\_ID)

);

**-- 5. Job Table**

CREATE TABLE Job (

JobID NUMBER PRIMARY KEY,

Job\_Title VARCHAR2(100),

CompanyName VARCHAR2(100),

A\_ID NUMBER,

FOREIGN KEY (A\_ID) REFERENCES Alumni(A\_ID)

);

**-- 6. Batch Table**

CREATE TABLE Batch (

BatchID NUMBER PRIMARY KEY,

Total\_Alumni NUMBER,

A\_ID NUMBER,

FOREIGN KEY (A\_ID) REFERENCES Alumni(A\_ID)

);

**-- 7. Session Table**

CREATE TABLE Event\_Session (

S\_ID NUMBER PRIMARY KEY,

S\_Date DATE,

S\_Time VARCHAR2(10),

EventID NUMBER,

FOREIGN KEY (EventID) REFERENCES Event(Event\_ID)

);

**Inserting at least 3 rows of data in each created table:**

**-- Alumni**

INSERT INTO Alumni VALUES (1, 'John Smith', 'Facebook', 'NY, USA', '1234567890', 'john@example.com', 2015);

INSERT INTO Alumni VALUES (2, 'Sara Khan', 'Twitter', 'London, UK', '2345678901', 'sara@example.com', 2016);

INSERT INTO Alumni VALUES (3, 'David Lee', 'LinkedIn', 'Toronto, Canada', '3456789012', 'david@example.com', 2017);

**-- Event**

INSERT INTO Event VALUES (101, 'Auditorium', 'Alumni Meet', 'Social', 1);

INSERT INTO Event VALUES (102, 'Hall B', 'Career Fair', 'Professional', 2);

INSERT INTO Event VALUES (103, 'Conference Room', 'Workshop', 'Educational', 3);

**-- Networking Group**

INSERT INTO Networking\_Group VALUES (201, 'Tech Group', 'Focus on IT networking', 1);

INSERT INTO Networking\_Group VALUES (202, 'Business Leaders', 'Entrepreneurship and startups', 2);

INSERT INTO Networking\_Group VALUES (203, 'Research Circle', 'Scientific research collaboration', 3);

**-- Donation**

INSERT INTO Donation VALUES (301, TO\_DATE('2024-05-10','YYYY-MM-DD'), '10:00AM', 'Library Renovation', 5000, 1);

INSERT INTO Donation VALUES (302, TO\_DATE('2024-06-15','YYYY-MM-DD'), '02:00PM', 'Scholarship Fund', 3000, 2);

INSERT INTO Donation VALUES (303, TO\_DATE('2024-07-20','YYYY-MM-DD'), '11:00AM', 'Sports Complex', 7000, 3);

**-- Job**

INSERT INTO Job VALUES (401, 'Software Engineer', 'Google', 1);

INSERT INTO Job VALUES (402, 'Marketing Manager', 'Amazon', 2);

INSERT INTO Job VALUES (403, 'Data Scientist', 'Microsoft', 3);

**-- Batch**

INSERT INTO Batch VALUES (501, 150, 1);

INSERT INTO Batch VALUES (502, 200, 2);

INSERT INTO Batch VALUES (503, 180, 3);

**-- Session**

INSERT INTO Event\_Session VALUES (601, TO\_DATE('2024-08-01','YYYY-MM-DD'), '09:00AM', 101);

INSERT INTO Event\_Session VALUES (602, TO\_DATE('2024-08-05','YYYY-MM-DD'), '01:00PM', 102);

INSERT INTO Event\_Session VALUES (603, TO\_DATE('2024-08-10','YYYY-MM-DD'), '03:00PM', 103);

**4.Query Writing (continuation of Question 3) (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**

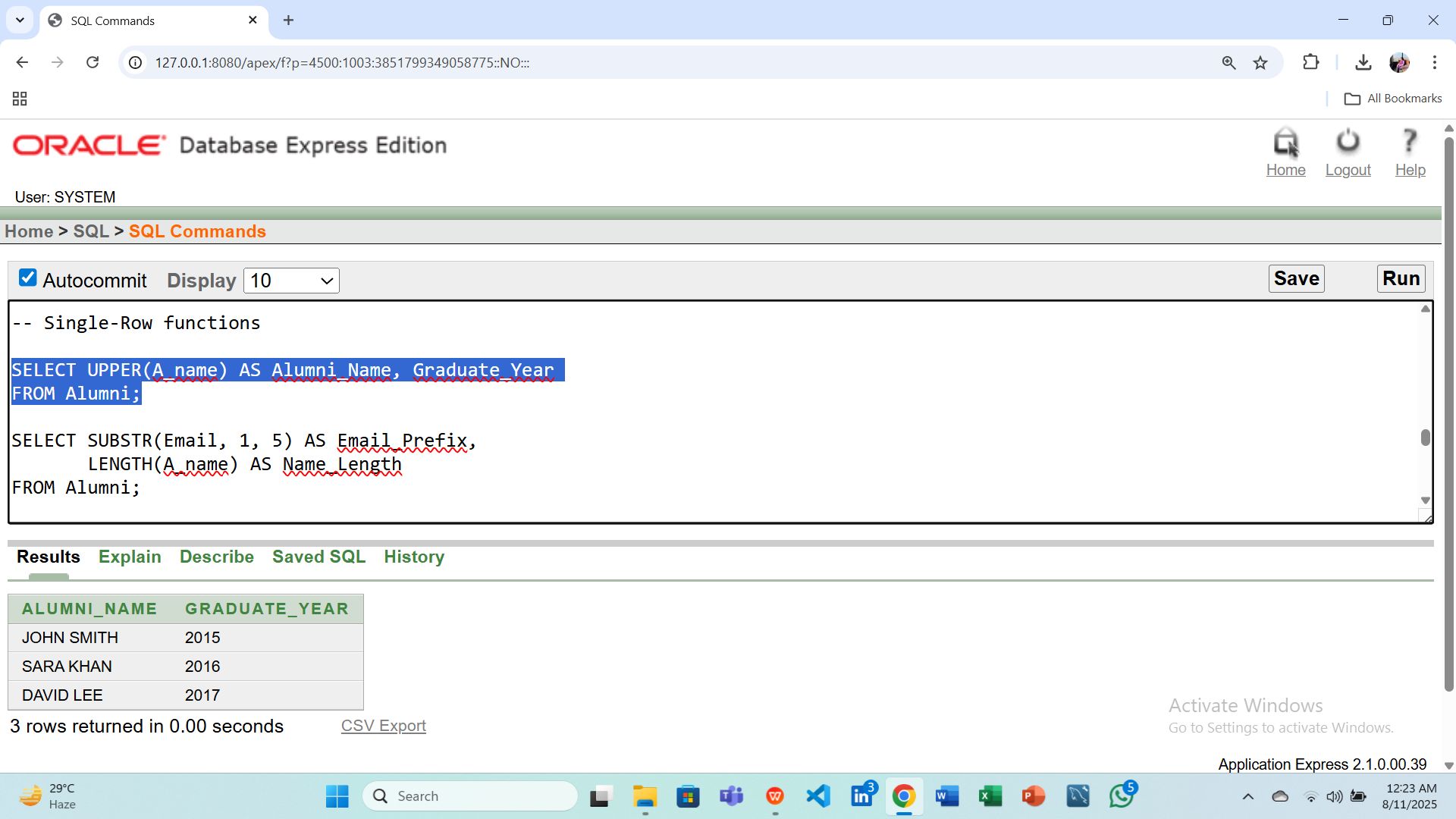
***For reference see BasicSQLTutorial and AdvanceSQLTutorial.***

Answer Box 4:

**1. Single-Row Functions**

**Question(1): Display the names of alumni in uppercase and show their graduation year.  
Answer:**

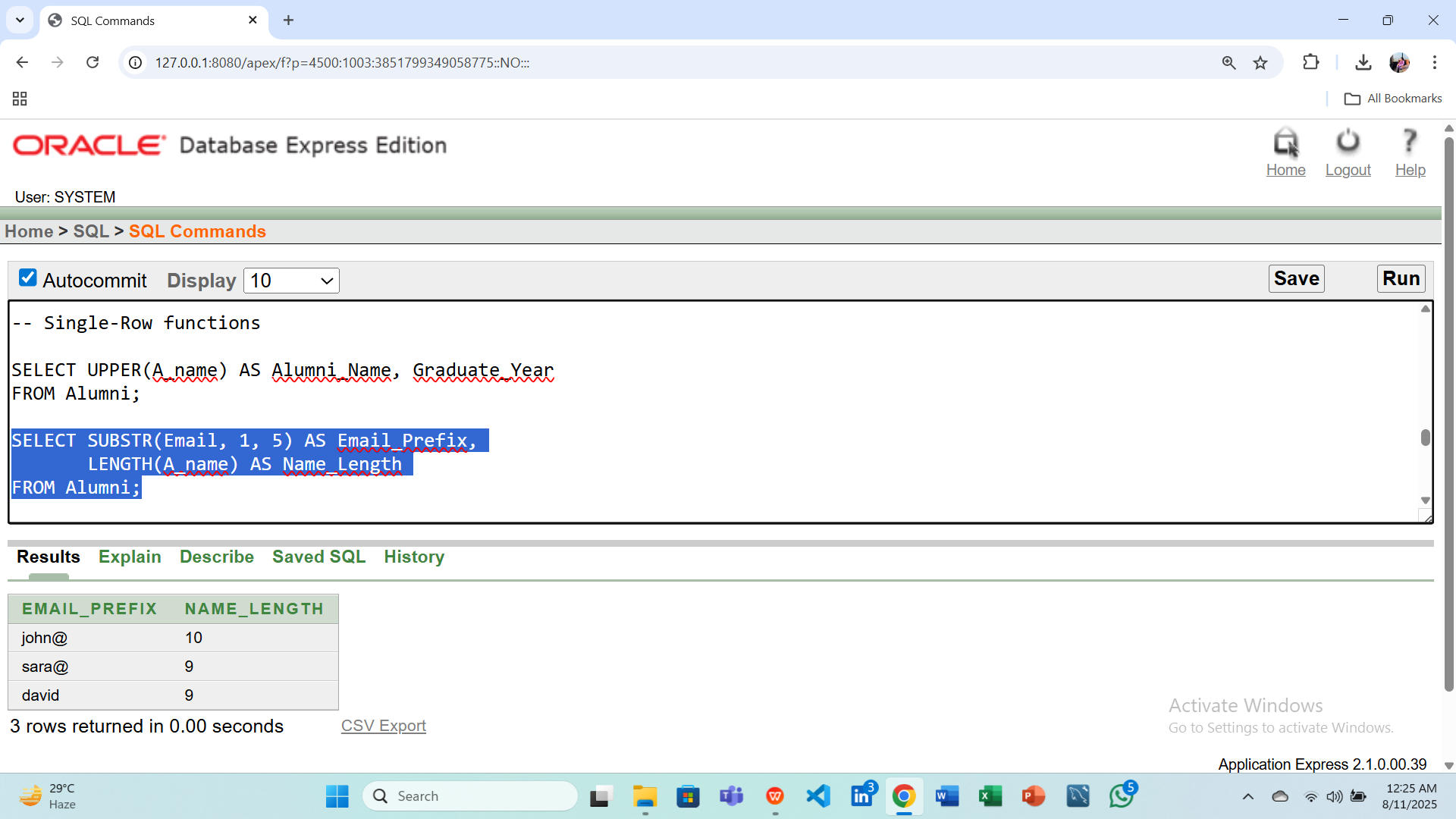
SELECT UPPER(A\_name) AS Alumni\_Name, Graduate\_Year FROM Alumni;



**Question(2): Display the first 5 characters of each alumni's email and the length of their name.  
Answer:**

SELECT SUBSTR(Email, 1, 5) AS Email\_Prefix,

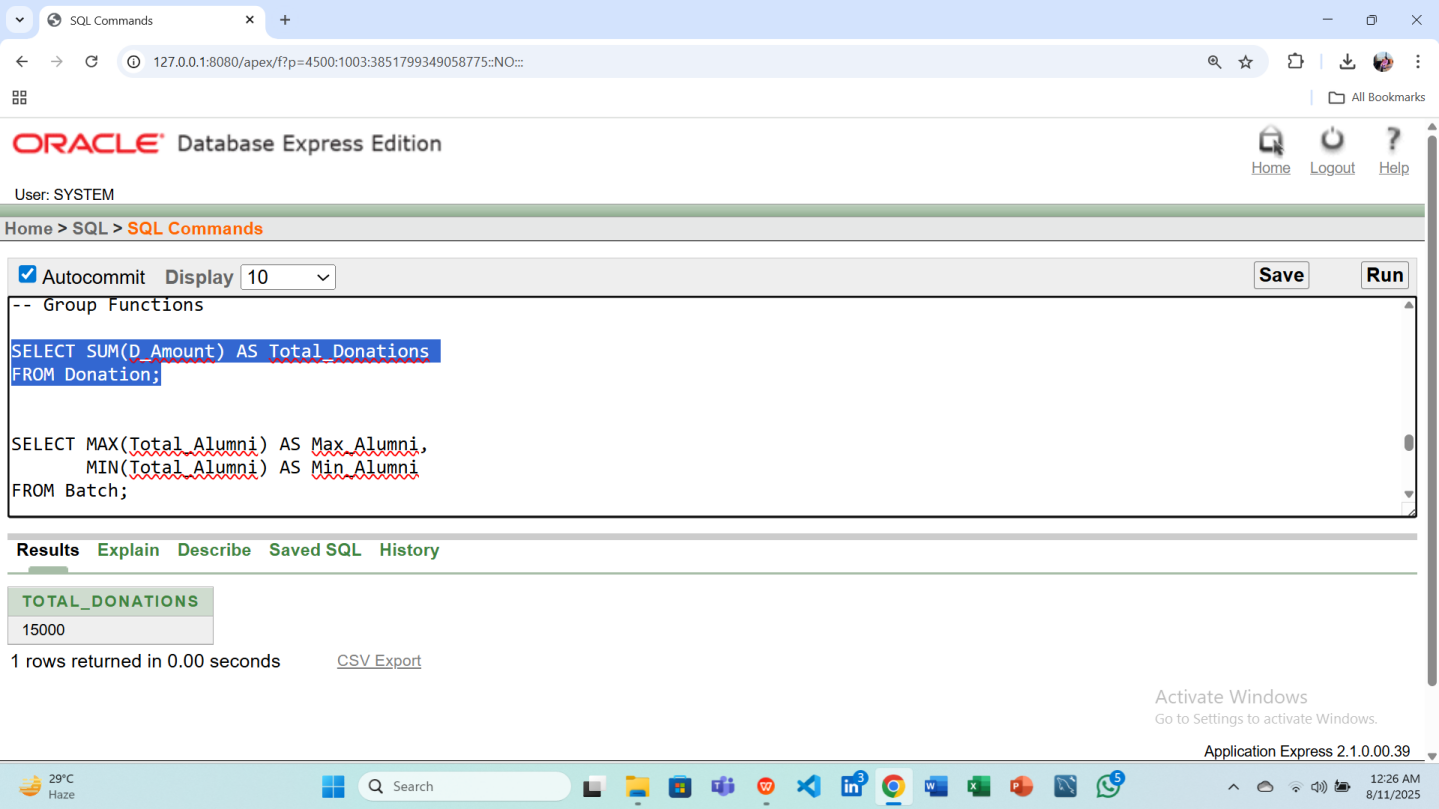
LENGTH(A\_name) AS Name\_LengthFROM Alumni;



**2. Group Functions**

**Question(1): Find the total donation amount received from all alumni.  
Answer:**

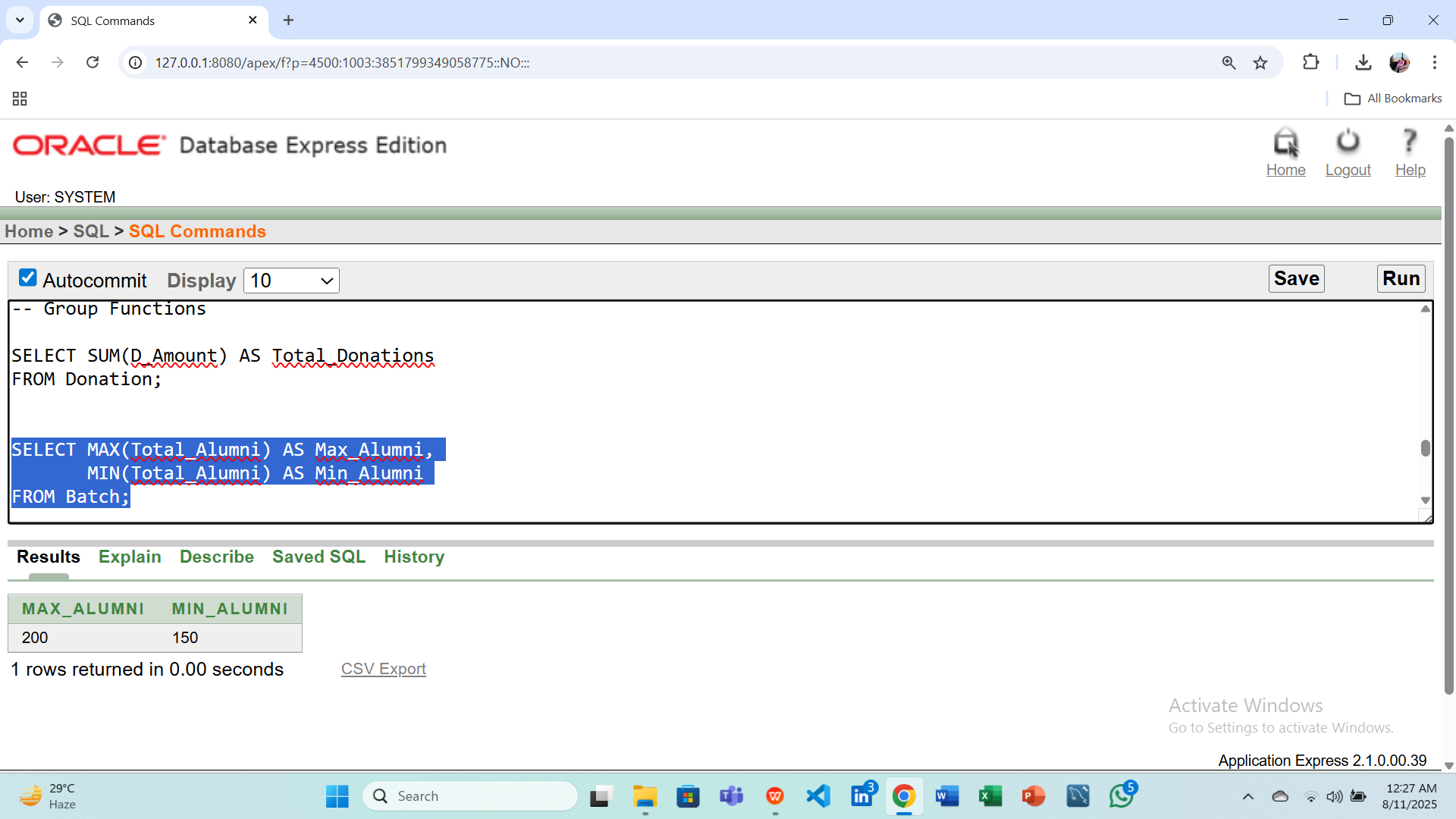
SELECT SUM(D\_Amount) AS Total\_DonationsFROM Donation;



**Question(2): Find the maximum and minimum total alumni count among all batches.  
Answer:**

SELECT MAX(Total\_Alumni) AS Max\_Alumni,

MIN(Total\_Alumni) AS Min\_AlumniFROM Batch;



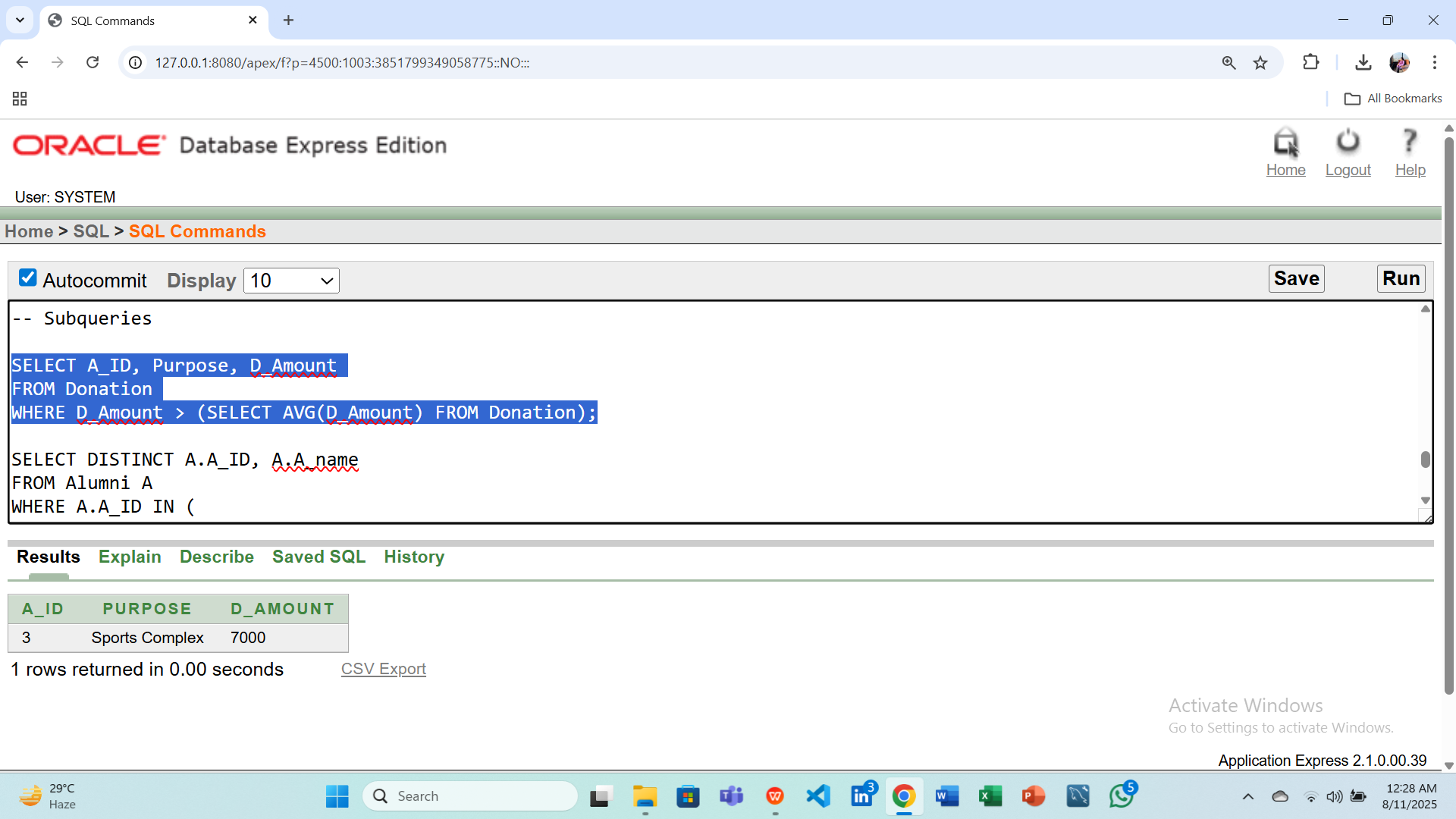
**3. Subqueries**

**Question(1): Display the alumni who donated more than the average donation amount.  
Answer:**

SELECT A\_ID, Purpose, D\_Amount

FROM Donation

WHERE D\_Amount > (SELECT AVG(D\_Amount) FROM Donation);



**Question(2): Display alumni who participated in events with the same venue as event ID 101.  
Answer:**

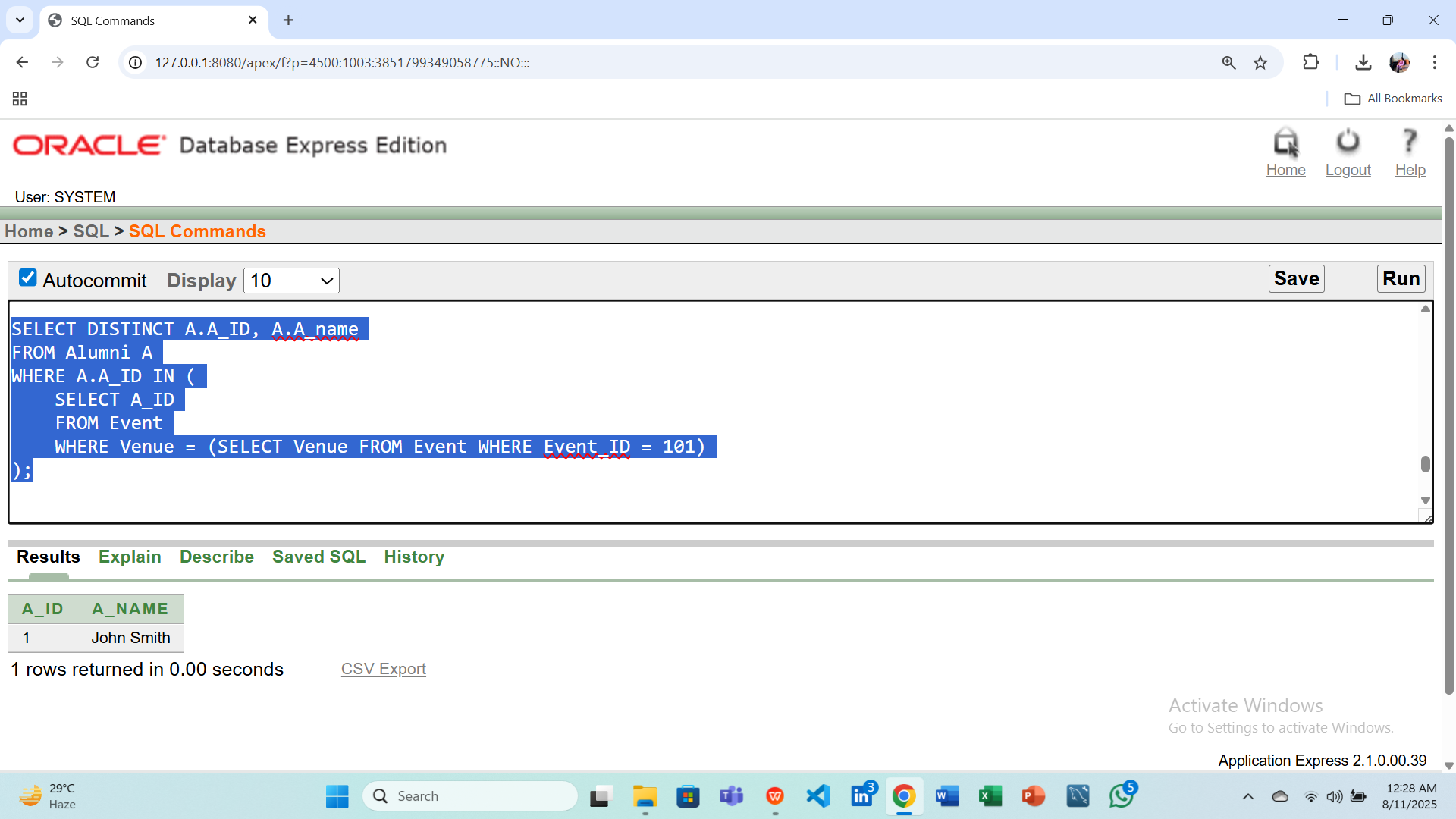
SELECT DISTINCT A.A\_ID, A.A\_nameFROM Alumni AWHERE A.A\_ID IN (

SELECT A\_ID

FROM Event

WHERE Venue = (SELECT Venue FROM Event WHERE Event\_ID = 101)

);



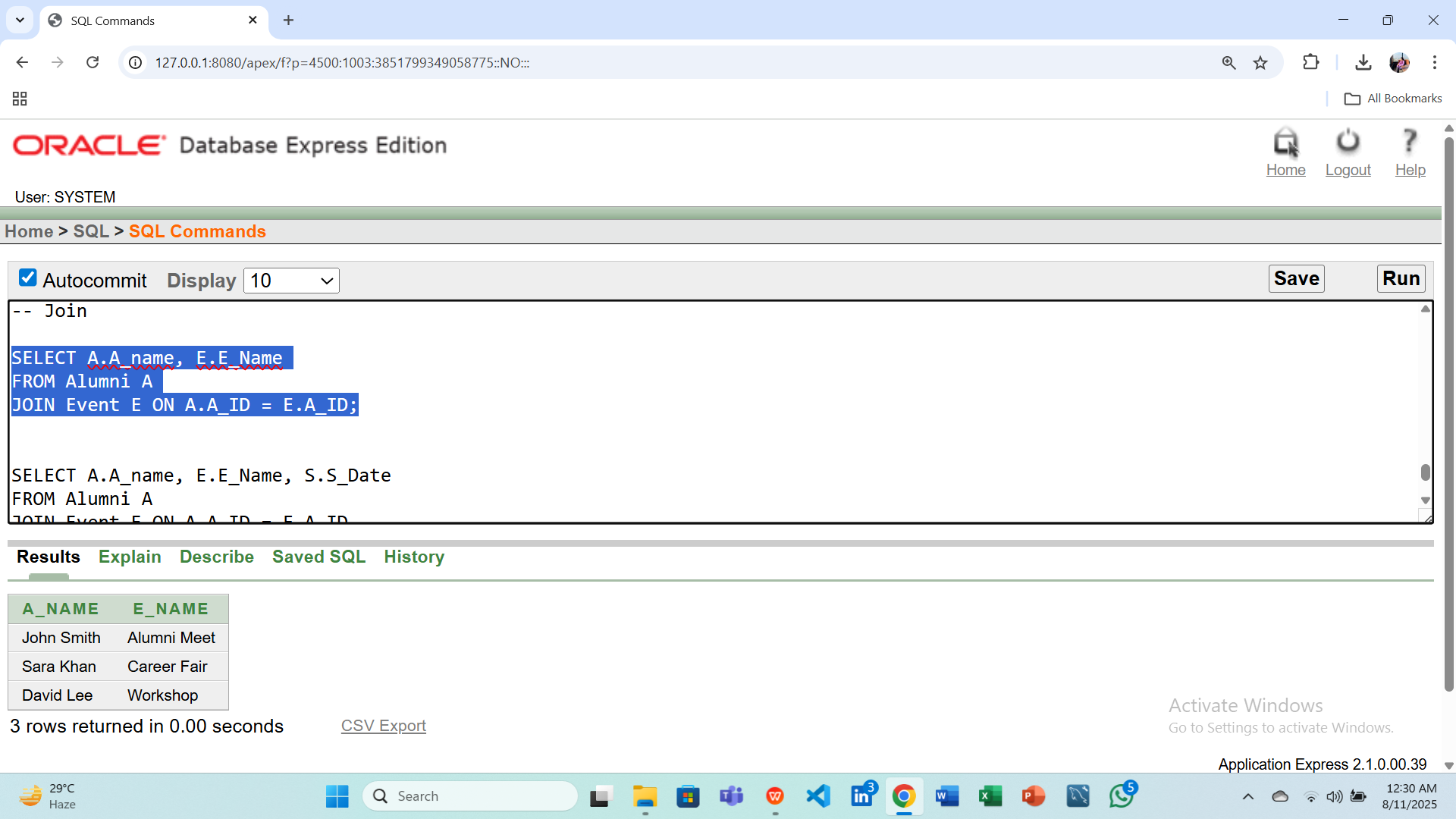
**4. Joins**

**Question(1): Display all alumni names along with the event names they are associated with.  
Answer:**

SELECT A.A\_name, E.E\_Name

FROM Alumni A

JOIN Event E ON A.A\_ID = E.A\_ID;



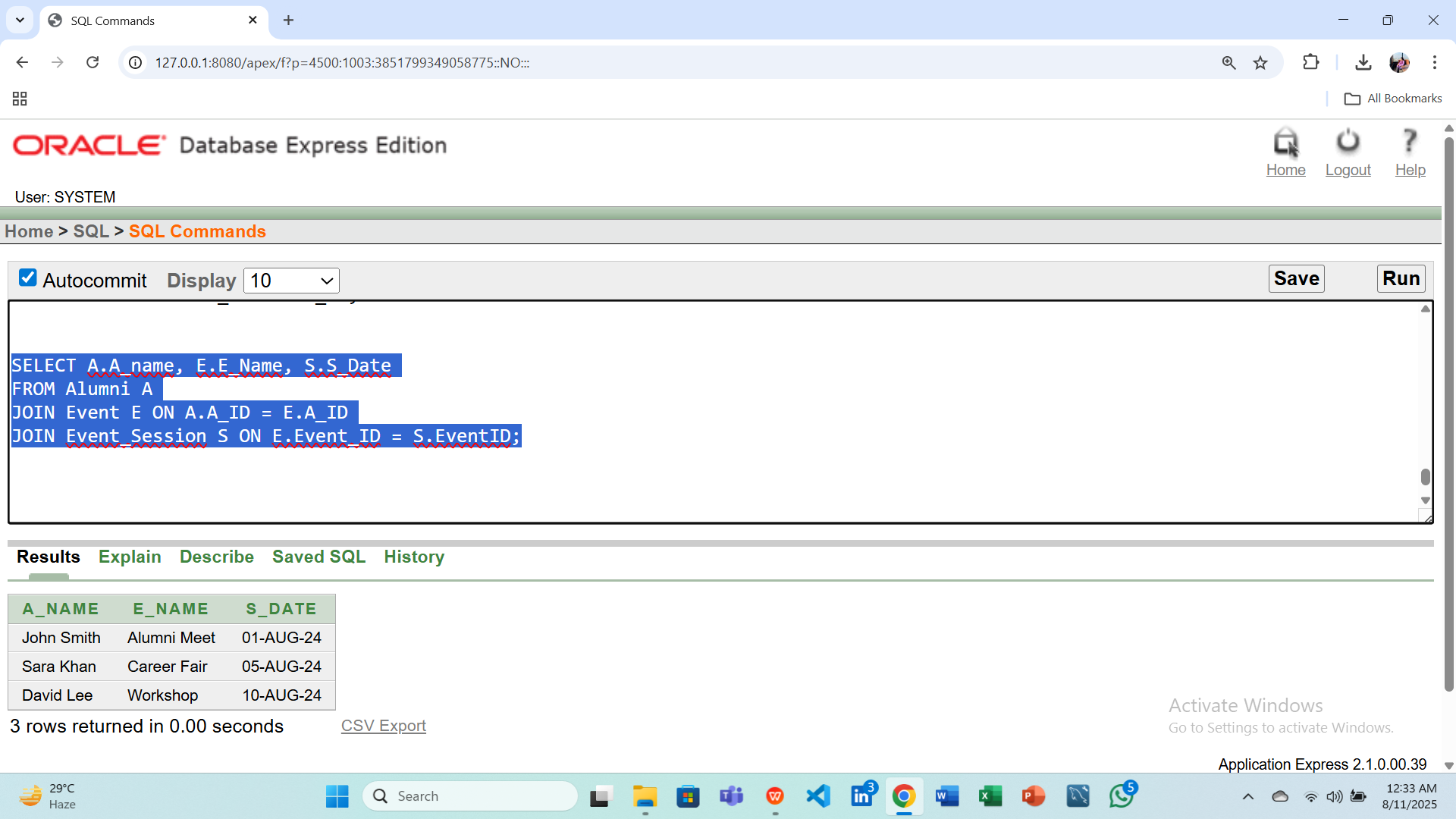
**Question(2): Display alumni names, event names, and session dates for all scheduled sessions.  
Answer:**

SELECT A.A\_name, E.E\_Name, S.S\_Date

FROM Alumni A

JOIN Event E ON A.A\_ID = E.A\_ID

JOIN Event\_Session S ON E.Event\_ID = S.EventID;



1. **Query Writing (continuation of Question 4) (Write down the answer only. 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**

**PL/SQL**

1. **Convert the SQLs of Question 4 into equivalent PL/SQL code**
2. **For this part, 8 PL/SQL code must be submitted**

Answer Box 5:

**1. Single-Row Functions**

**Question(1): Display the names of alumni in uppercase and show their graduation year.  
Answer:**

BEGIN

FOR rec IN (

SELECT \* FROM (

SELECT UPPER(A\_name) AS Alumni\_Name, Graduate\_Year

FROM Alumni

ORDER BY A\_name

)

WHERE ROWNUM <= 3

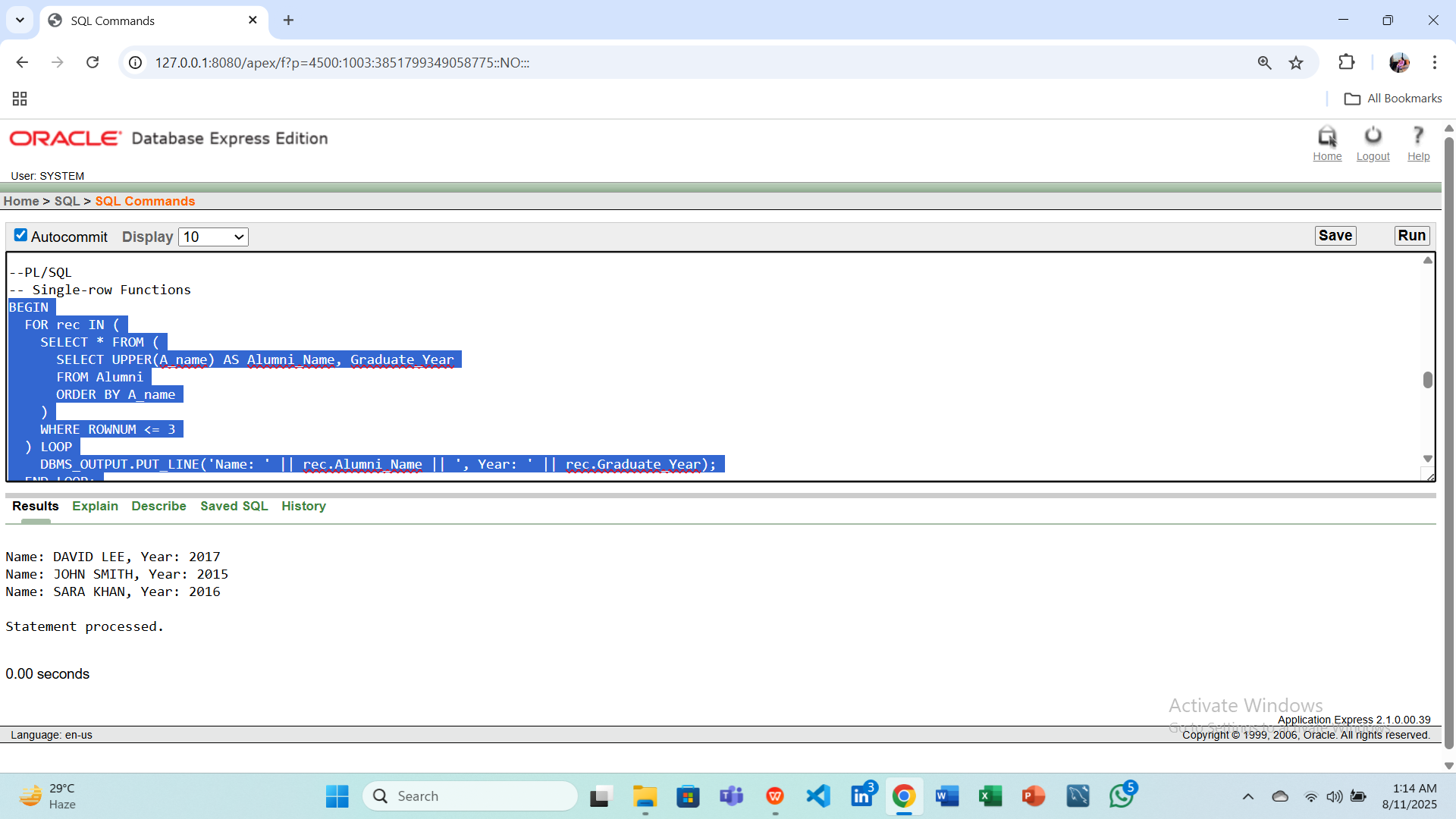
) LOOP

DBMS\_OUTPUT.PUT\_LINE('Name: ' || rec.Alumni\_Name || ', Year: ' || rec.Graduate\_Year);

END LOOP;

END;

/



**Question(2): Display the first 5 characters of each alumni's email and the length of their name.  
Answer:**

BEGIN

FOR rec IN (

SELECT SUBSTR(Email, 1, 5) AS Email\_Prefix,

LENGTH(A\_name) AS Name\_Length

FROM Alumni

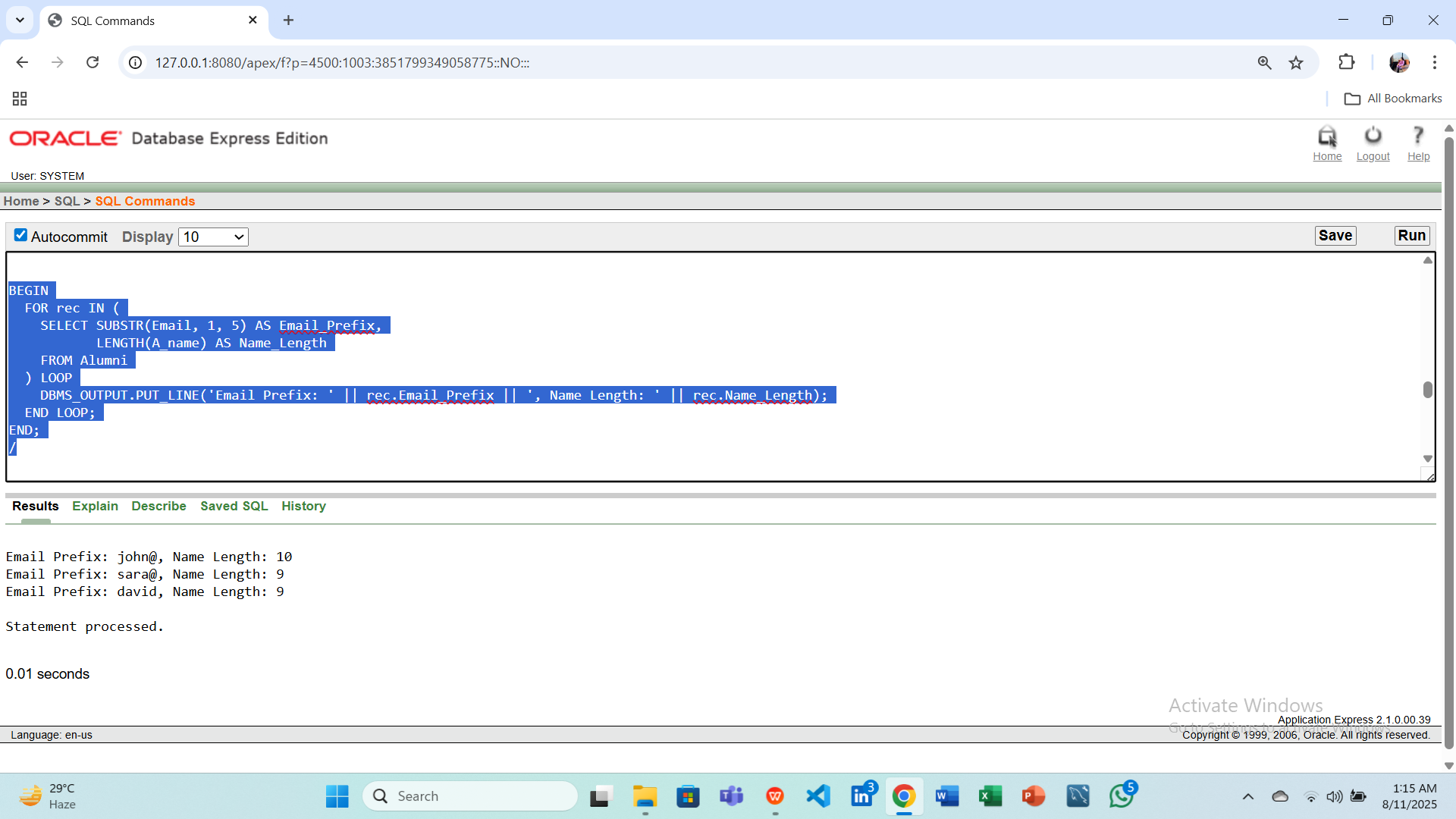
) LOOP

DBMS\_OUTPUT.PUT\_LINE('Email Prefix: ' || rec.Email\_Prefix || ', Name Length: ' || rec.Name\_Length);

END LOOP;

END;

/



**2. Group Functions**

**Question(1): Find the total donation amount received from all alumni.  
Answer:**

DECLARE

v\_total NUMBER;

BEGIN

SELECT SUM(D\_Amount)

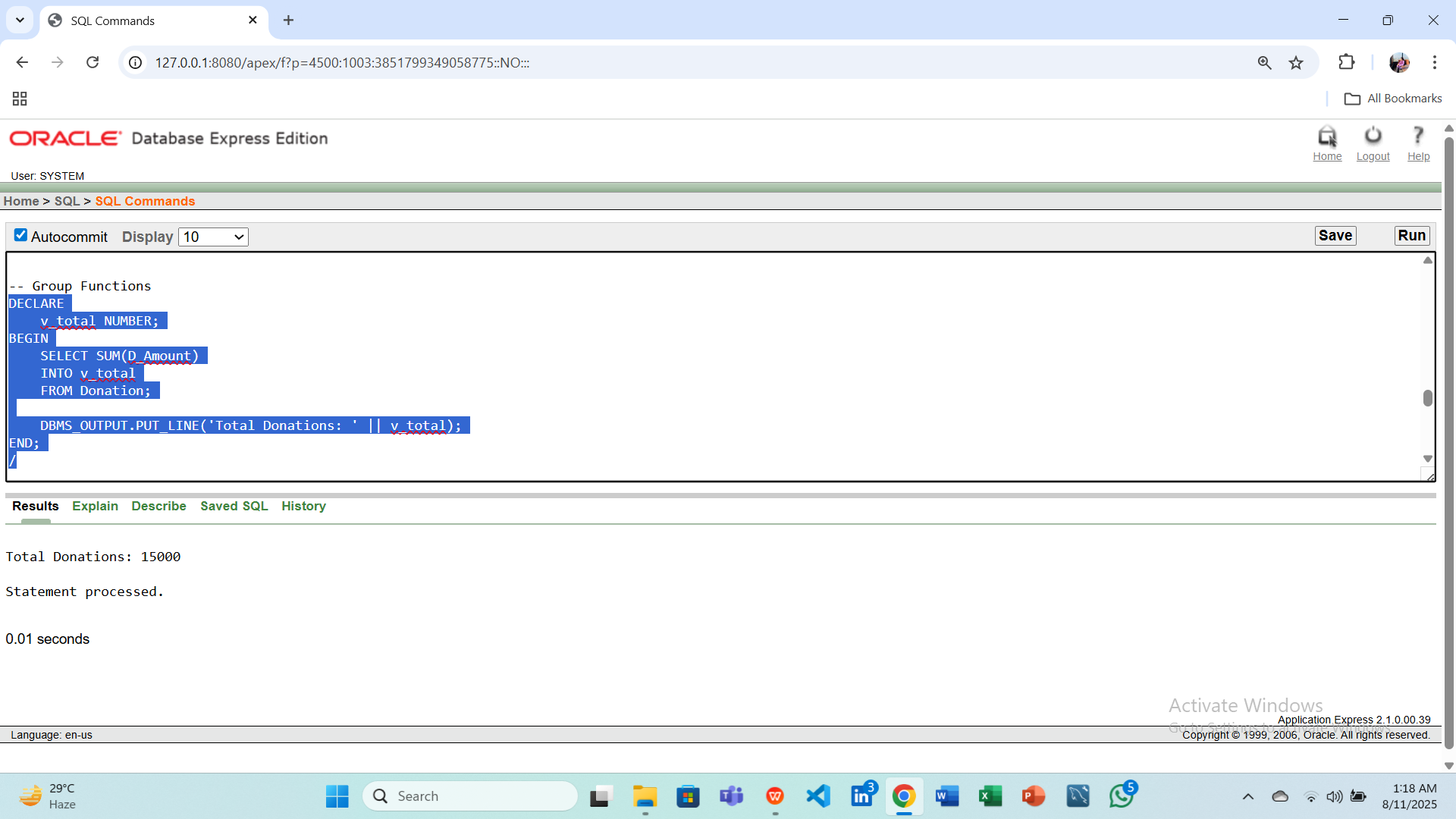
INTO v\_total

FROM Donation;

DBMS\_OUTPUT.PUT\_LINE('Total Donations: ' || v\_total);

END;

/



**Question(2): Find the maximum and minimum total alumni count among all batches.  
Answer:**

DECLARE

v\_max NUMBER;

v\_min NUMBER;

BEGIN

SELECT MAX(Total\_Alumni), MIN(Total\_Alumni)

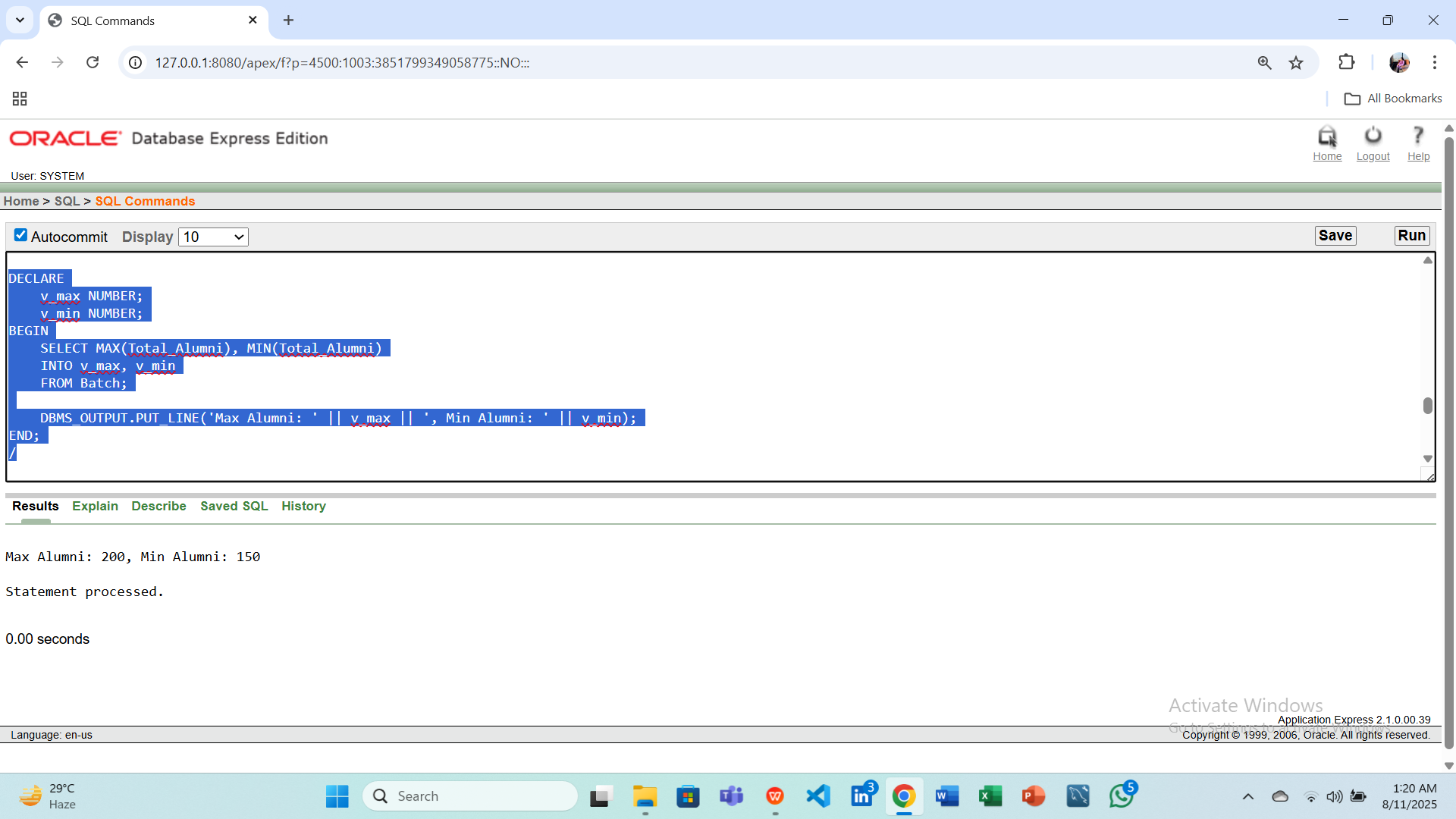
INTO v\_max, v\_min

FROM Batch;

DBMS\_OUTPUT.PUT\_LINE('Max Alumni: ' || v\_max || ', Min Alumni: ' || v\_min);

END;

/



**3. Subqueries**

**Question(1): Display the alumni who donated more than the average donation amount.  
Answer:**

BEGIN

FOR rec IN (

SELECT A\_ID, Purpose, D\_Amount

FROM Donation

WHERE D\_Amount > (SELECT AVG(D\_Amount) FROM Donation)

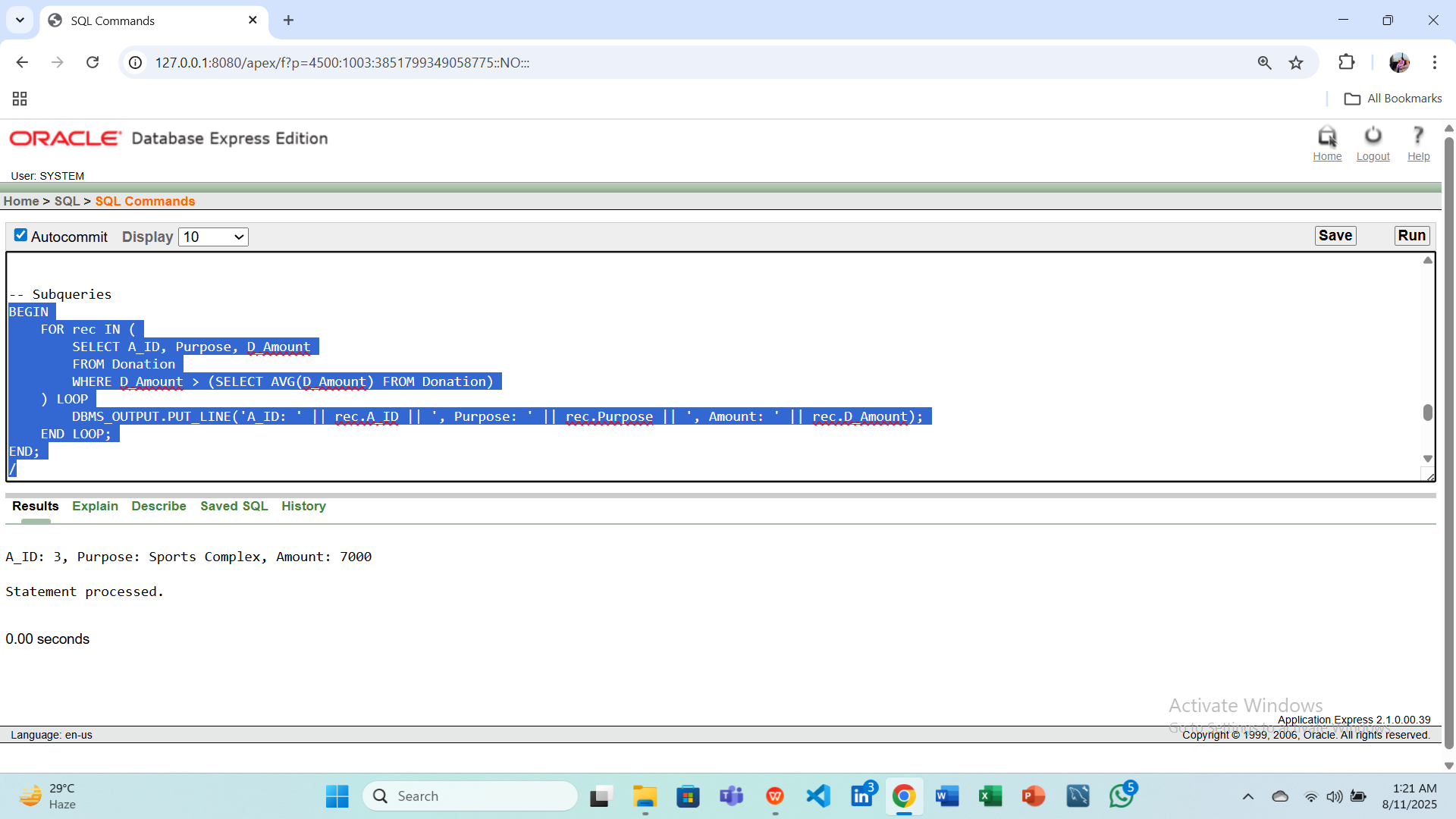
) LOOP

DBMS\_OUTPUT.PUT\_LINE('A\_ID: ' || rec.A\_ID || ', Purpose: ' || rec.Purpose || ', Amount: ' || rec.D\_Amount);

END LOOP;

END;

/



**Question(2): Display alumni who participated in events with the same venue as event ID 101.  
Answer:**

BEGIN

FOR rec IN (

SELECT DISTINCT A.A\_ID, A.A\_name

FROM Alumni A

WHERE A.A\_ID IN (

SELECT A\_ID

FROM Event

WHERE Venue = (SELECT Venue FROM Event WHERE Event\_ID = 101)

)

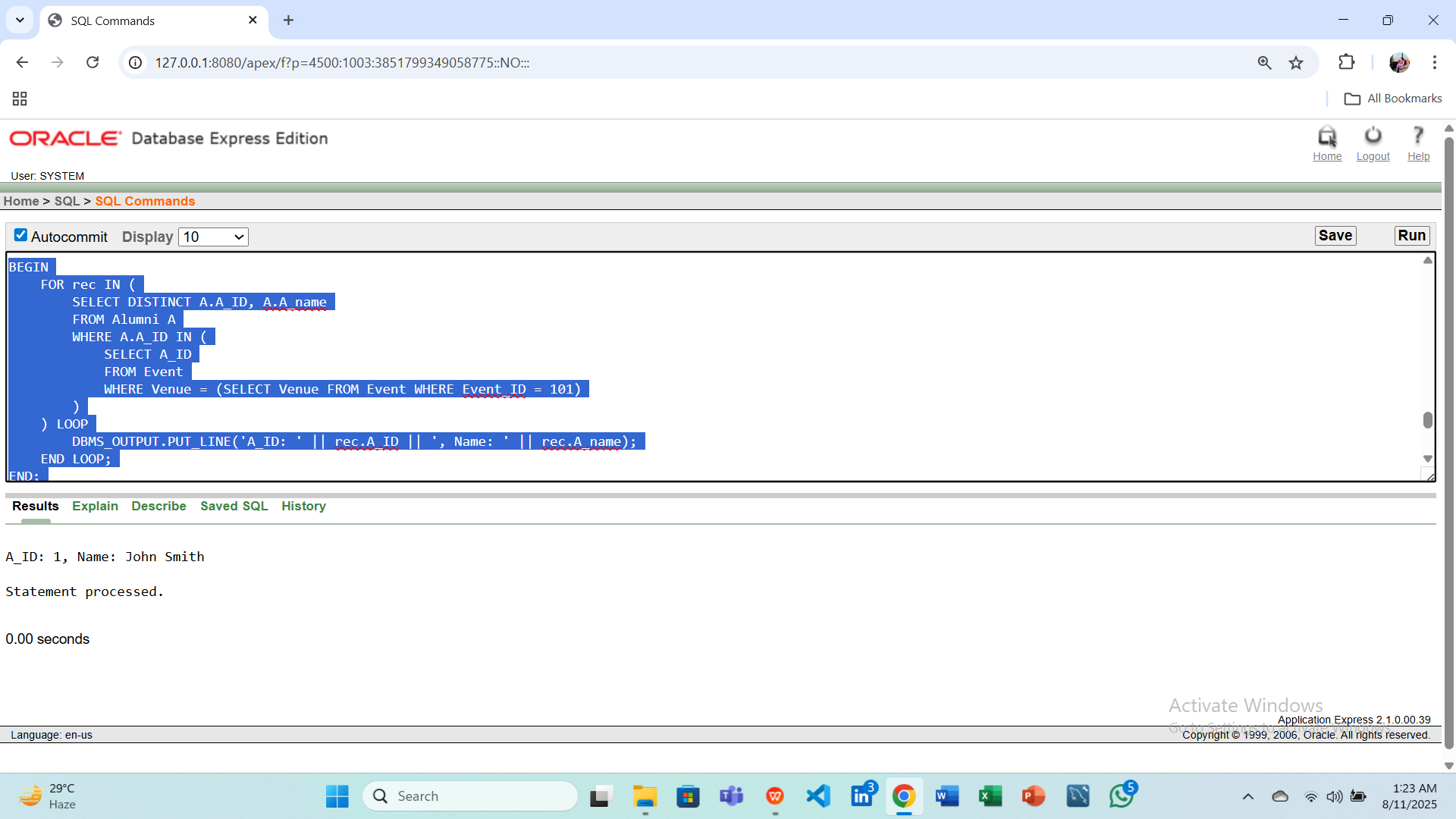
) LOOP

DBMS\_OUTPUT.PUT\_LINE('A\_ID: ' || rec.A\_ID || ', Name: ' || rec.A\_name);

END LOOP;

END;

/



**4. Joins**

**Question(1): Display all alumni names along with the event names they are associated with.  
Answer:**

BEGIN

FOR rec IN (

SELECT A.A\_name, E.E\_Name

FROM Alumni A

JOIN Event E ON A.A\_ID = E.A\_ID

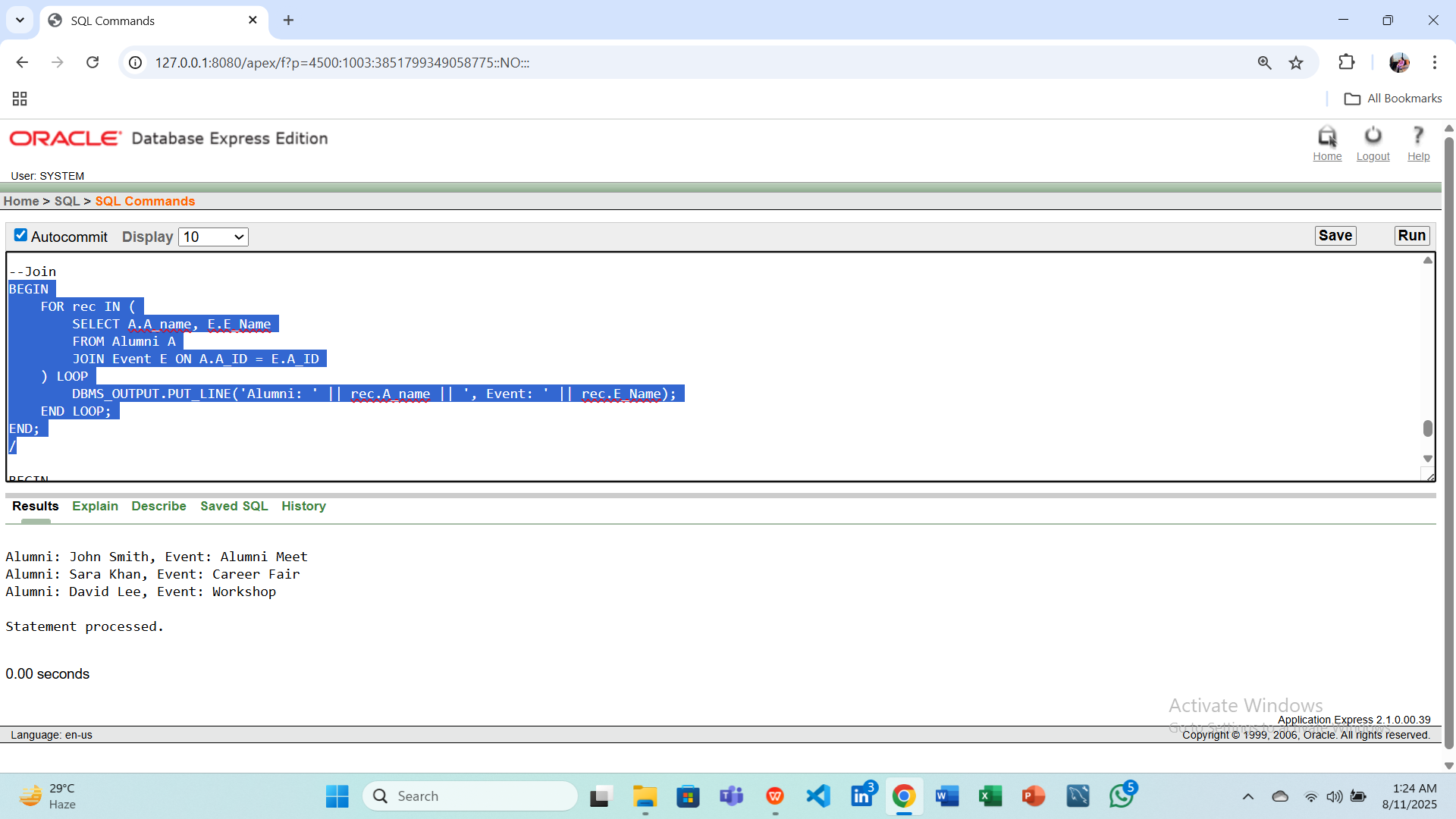
) LOOP

DBMS\_OUTPUT.PUT\_LINE('Alumni: ' || rec.A\_name || ', Event: ' || rec.E\_Name);

END LOOP;

END;

/



**Question(2): Display alumni names, event names, and session dates for all scheduled sessions.  
Answer:**

BEGIN

FOR rec IN (

SELECT A.A\_name, E.E\_Name, S.S\_Date

FROM Alumni A

JOIN Event E ON A.A\_ID = E.A\_ID

JOIN Event\_Session S ON E.Event\_ID = S.EventID

) LOOP

DBMS\_OUTPUT.PUT\_LINE('Alumni: ' || rec.A\_name || ', Event: ' || rec.E\_Name || ', Date: ' || TO\_CHAR(rec.S\_Date, 'DD-MON-YYYY'));

END LOOP;

END;

/

