Introduction to Database

Finalterm Assignment

|  |  |
| --- | --- |
| **Name** | **ID** |
| 1. MD. Mohituzzaman Bhuiyan | 20-42546-1 |
| 1. Mostaque Ahammed Naim | 20-42140-1 |
| 1. Joysree Dey Sukanna | 20-41946-1 |
| 1. Amit Podder | 20-42273-1 |
| 1. Shakib MD. Shahadot Hossain | 20-42385-1 |

**Name of the Assignment**

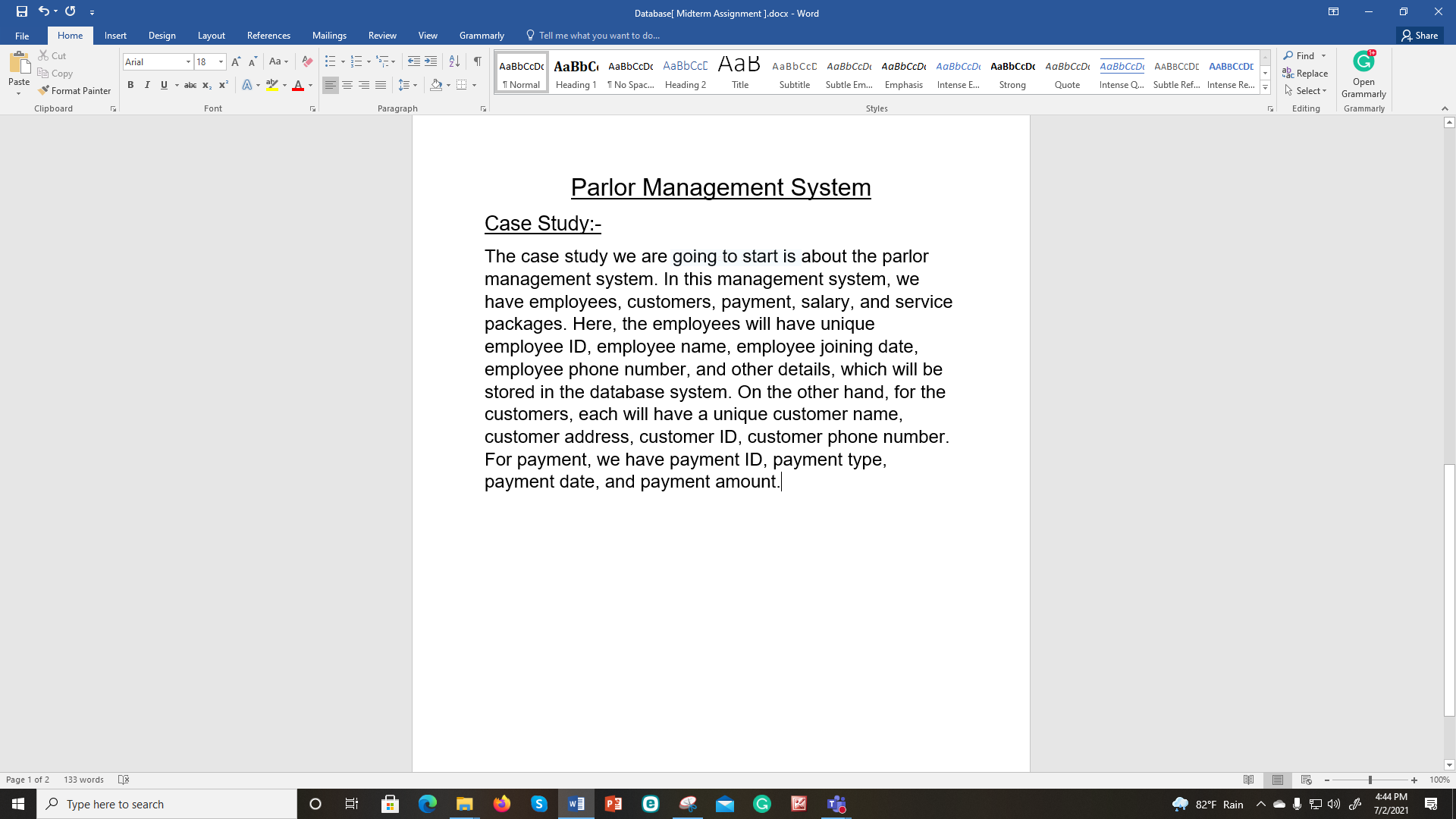
* Parlor Management System

**Parlor Management System**

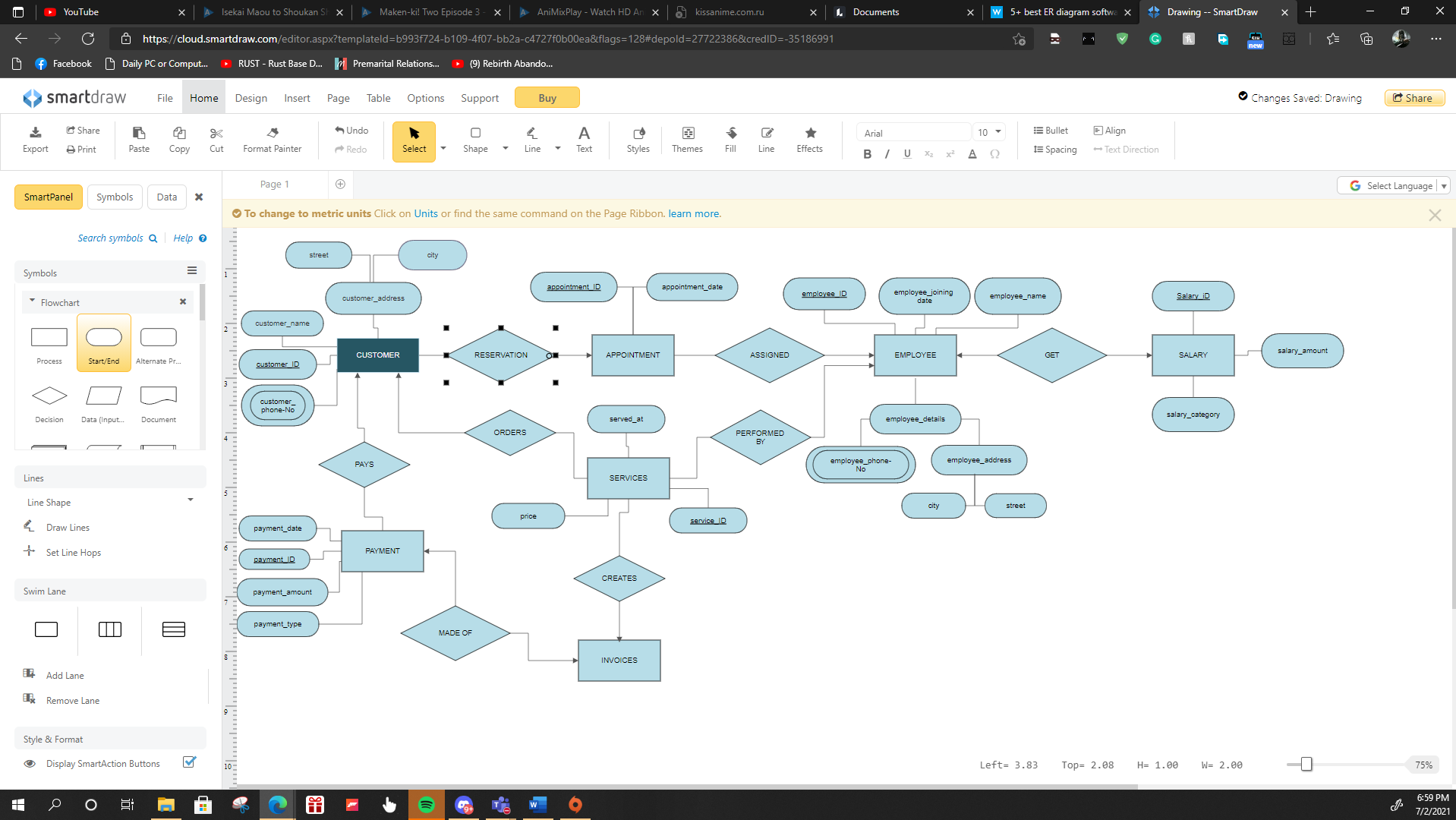
Case Study:-

The case study we are going to start is about the parlor management system. In this management system, we have employees, customers, payment, salary, and service packages. Here, the employees will have unique employee ID, employee name, employee joining date, employee phone number, and other details, which will be stored in the database system. On the other hand, for the customers, each will have a customer name, customer address, unqiue customer ID, customer phone number. In terms of payment, we have unique payment ID, payment type, payment date, and payment amount. Here, employees and customers can have multiple phone numbers. Here, the customers can book an appointment which will identify the unique appointment ID, appointment date, and the type of service the customer wants which will also include unique service ID and service price. A customer can have many services or we can also say that a service can be taken by many customers. When the customer decides it service, then the employee will be notified by invoice ID, invoice service, and invoice amount. Then the customer will pay the bill according to the service and it will identify by payment types, payment date, payment ID, and payment amount. After that, the employees will get their salary as a unique salary ID, salary category and, salary amount.

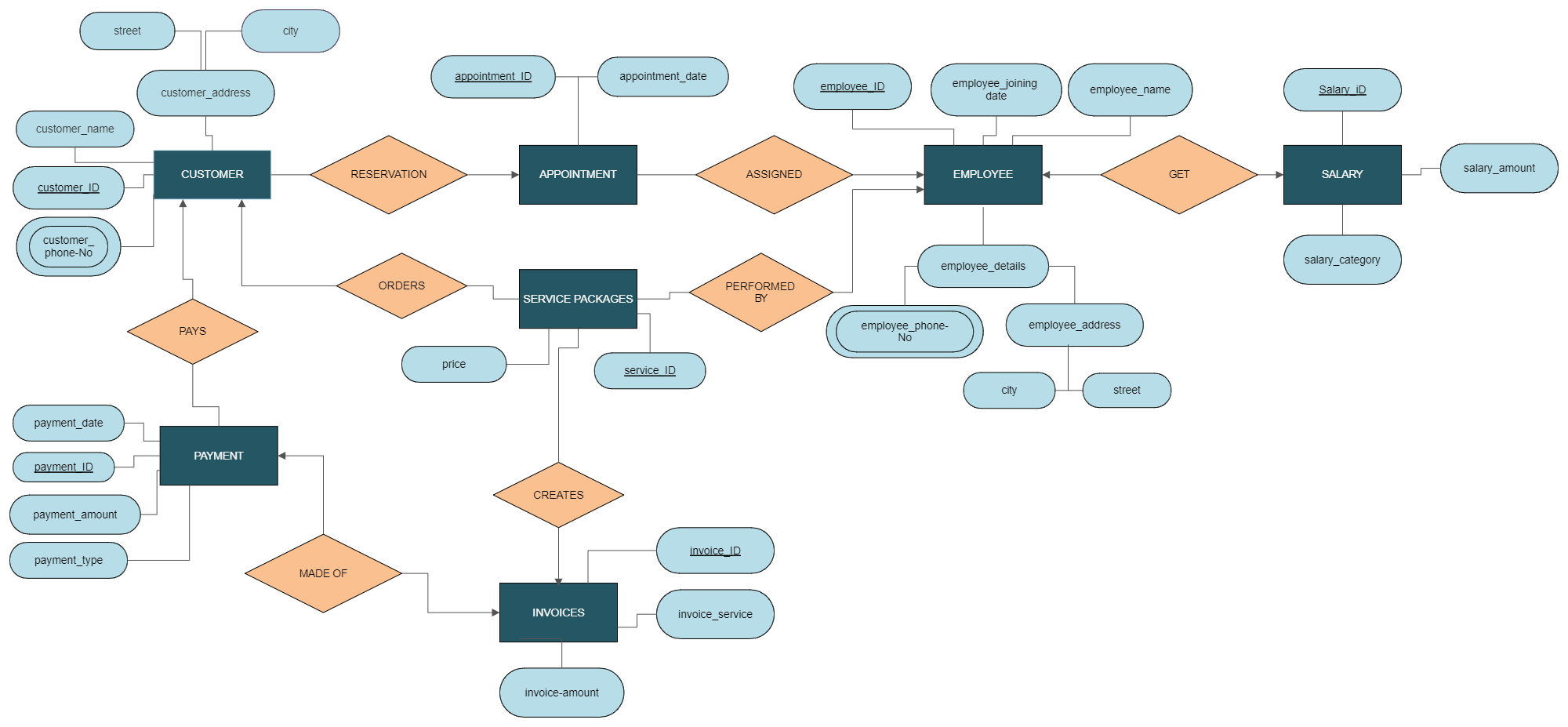
Starting Screenshot:-



Before Finishing Screenshot:-



ER Diagram:-



Normalization of Database and Functional Dependencies:-

**CUSTOMER:**

UNF:

customer (customer\_id, customer\_name, street, city, customer\_phone-No, payment\_id, employee\_id)

1NF:

customer\_phone-No is multivalued.

1. customer\_id, customer\_name, street, city, customer\_phone-No, payment\_id, employee\_id

2NF:

1. customer\_id, customer\_name
2. customer\_address, city, street
3. payment\_id, employee\_id, customer\_phone-No

3NF:

No transitive Dependency

**Services Packages:**

UNF:

services (service\_ID, price, employee\_ID, customer\_ID, invoice\_ID)

1NF:

1. service\_ID, price, employee\_ID, customer\_ID, invoice\_ID

2NF:

1. service\_ID, employee\_ID, customer\_ID, invoice\_ID

3NF:

No transitive dependency

**Invoice:**

UNF:

invoice (invoice\_ID, Invoice\_service, service\_ID, invoice\_amount)

1NF:

1. invoice\_ID, invoice\_service, service\_ID, invoice\_amount

2NF:

No partial dependency

3NF:

No transitive dependency

**Payment:**

UNF:

payment (payment\_ID, payment\_date, payment\_amount, payment\_type, customer\_ID, invoice\_ID)

1NF:

1. payment\_ID, payment\_date, payment\_amount, payment\_type, customer\_ID, invoice\_ID

2NF:

1. No partial dependency

3NF:

1. No transitive dependency

**Appointment:**

UNF:

appointments (appointment\_ID, appointment\_\_date, customer\_ID, employee\_ID)

1NF:

appointment\_ID, appointment\_\_date, customer\_ID, employee\_ID

2NF:

No partial dependency

3NF:

No transitive dependency

**Employee:**

UNF:

employee(employee\_ID, employee\_joining-details, employee\_name, employee\_phone-no, city, street)

1NF:

employee\_phone-no is multivalued

1. employee\_ID, employee\_joining-details, employee\_name, employee\_phone-no, city, street

2NF:

1. employee\_ID, employee\_details, employee\_joining-date
2. detail\_ID, employee\_ID, city, street, employee\_phone-no
3. service\_ID, employee\_ID, employee\_phone-no

3NF:

1. Employee\_ID, employee\_details, employee\_joining-date
2. detail\_ID, a\_ID
3. Service\_ID, employee\_ID, employee\_phone-no
4. a\_ID, street, city

**Salary:**

UNF:

salary(salary\_ID, employee\_ID, salary\_type, salary\_amount)

1NF:

1. Salary\_ID, employee\_ID, salary\_type, salary\_amount

2NF:

No dependency

3NF:

No dependency

**Final Table Creation:-**

1. customer\_id, customer\_name, customer\_address
2. a\_ID, street, city
3. service\_ID, employee\_ID, employee\_phone-no
4. employee\_ID, employee\_details, employee\_joining-date
5. salary\_ID, employee\_ID, salary\_type, salary\_amount
6. payment\_ID, payment\_date, payment\_amount, payment\_type, customer\_ID, invoice\_ID
7. appointment\_ID, customer\_ID, employee\_ID, appointment\_date
8. invoice\_ID, invoice\_service, service\_ID, invoice\_amount
9. service\_ID, employee\_ID, customer\_ID, invoice\_ID

Creating New User and Granting Permission:

create user parAD identified by parlor;

grant create view, connect, resource, unlimited tablespace to parAD;

Table Creation and Data Insertion:-

* create table address(

a\_ID number PRIMARY KEY,

street varchar2(20),

city varchar2(10)

);

INSERT INTO ADDRESS VALUES(1, 'MIRPUR DOHS', 'DHAKA');

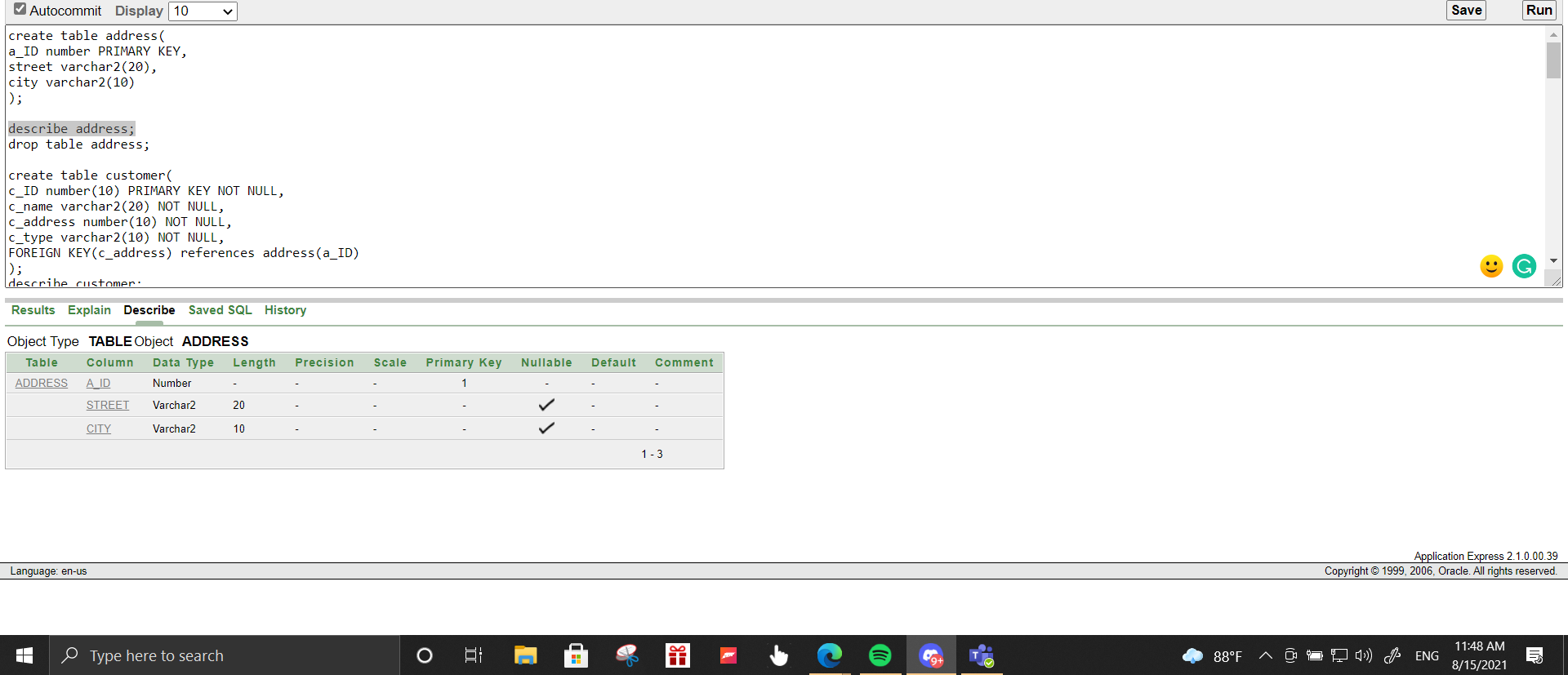
INSERT INTO ADDRESS VALUES(2, 'HALISAHAR', 'CHITTAGONG');

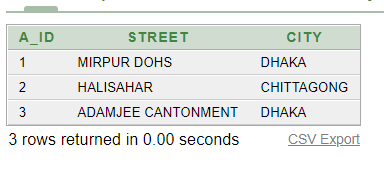
INSERT INTO ADDRESS VALUES(3, 'ADAMJEE CANTONMENT', 'DHAKA');

select \* from address;

describe address;

drop table address;





* create table customer(

c\_ID number(10) PRIMARY KEY NOT NULL,

c\_name varchar2(20) NOT NULL,

c\_address number(10) NOT NULL,

c\_type varchar2(10) NOT NULL,

FOREIGN KEY(c\_address) references address(a\_ID)

);

INSERT INTO CUSTOMER VALUES(1, 'MR.X', 2, 'PREMIUM');

INSERT INTO CUSTOMER VALUES(2, 'MR.Z', 3, 'REGULAR');

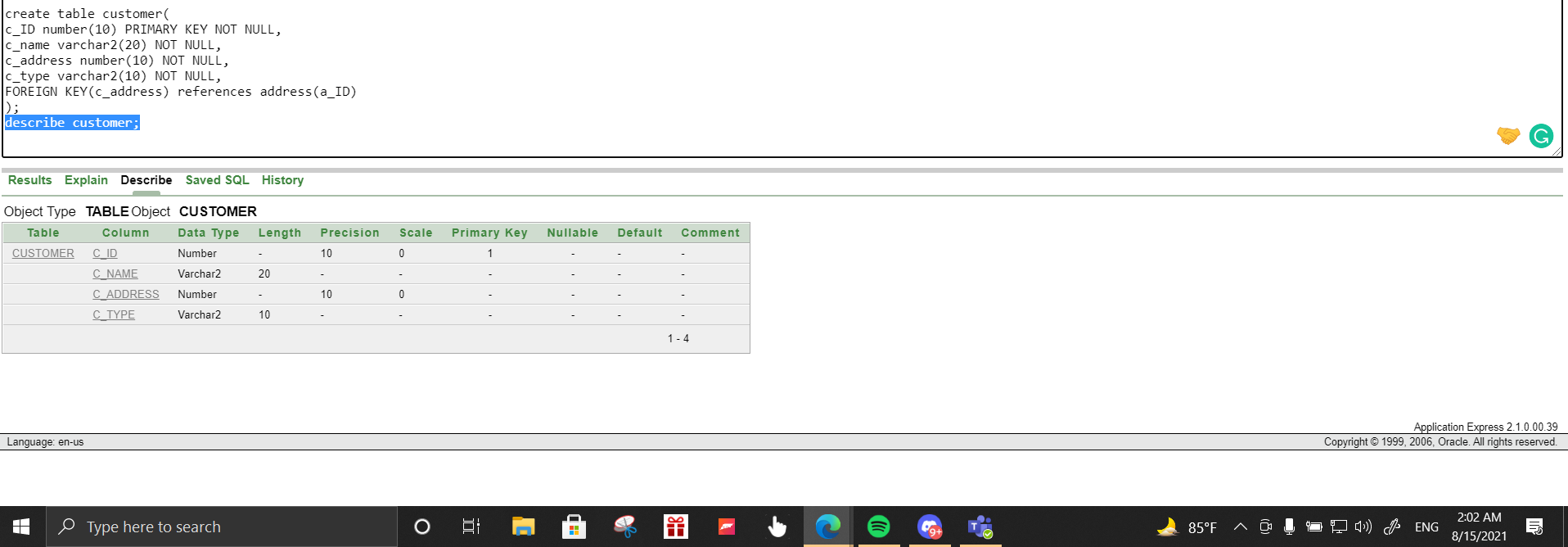
INSERT INTO CUSTOMER VALUES(3, 'MR.Y', 1, 'REGULAR');

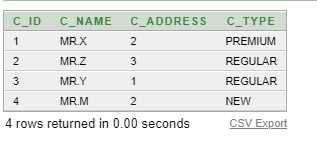
INSERT INTO CUSTOMER VALUES(4, 'MR.M', 2, 'NEW');

select \* from customer;

describe customer;

drop table customer;





* create table employee(

e\_ID number(10) PRIMARY KEY NOT NULL,

e\_name varchar2(20) NOT NULL,

e\_joining\_date date NOT NULL,

e\_address number(10) NOT NULL,

FOREIGN KEY(e\_address) references address(a\_ID)

);

INSERT INTO EMPLOYEE VALUES(101, 'RAFIK', '24-JAN-2021', 2);

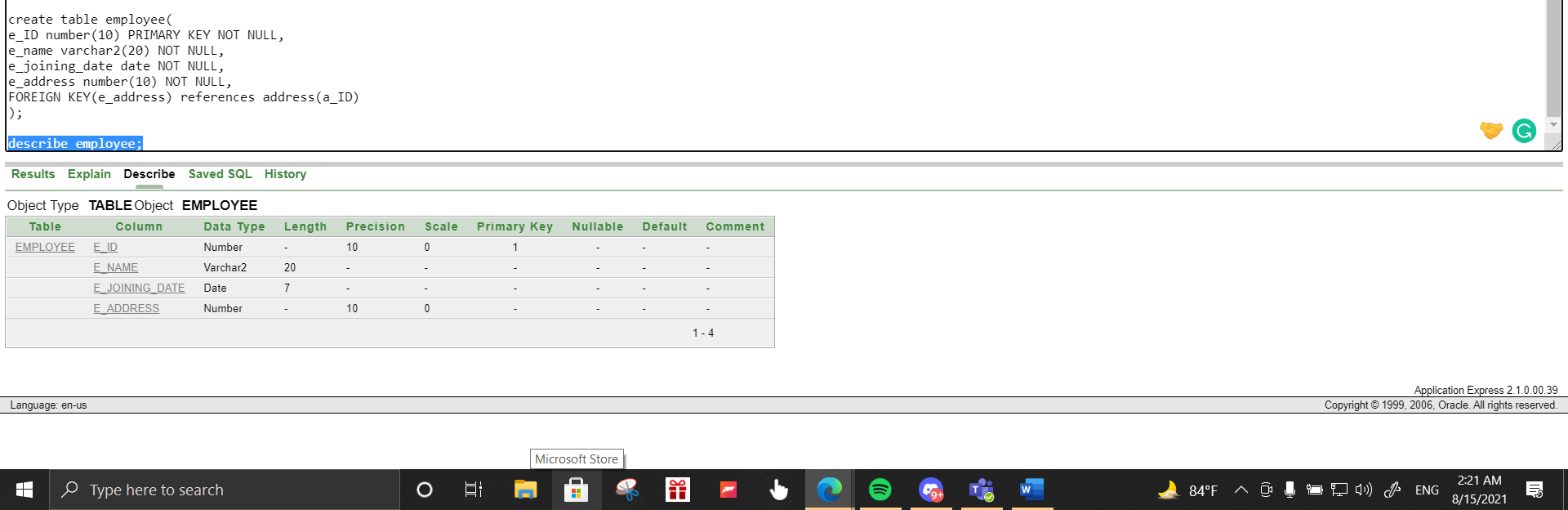
INSERT INTO EMPLOYEE VALUES(102, 'TASNIM', '24-JAN-2021', 1);

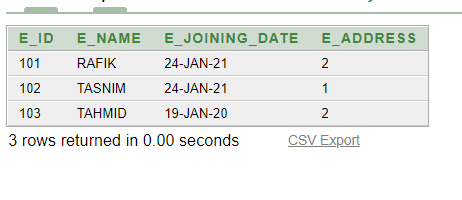
INSERT INTO EMPLOYEE VALUES(103, 'TAHMID', '19-JAN-2020', 2);

select \* from employee;

describe employee;

drop table employee;





* create table salary(

salary\_ID number(10),

e\_ID number(10),

salary\_cat varchar2(20),

salary\_amount number(20),

FOREIGN KEY(e\_ID) references employee(e\_ID)

);

INSERT INTO SALARY VALUES(11, 101, 'MONTHLY', 25000);

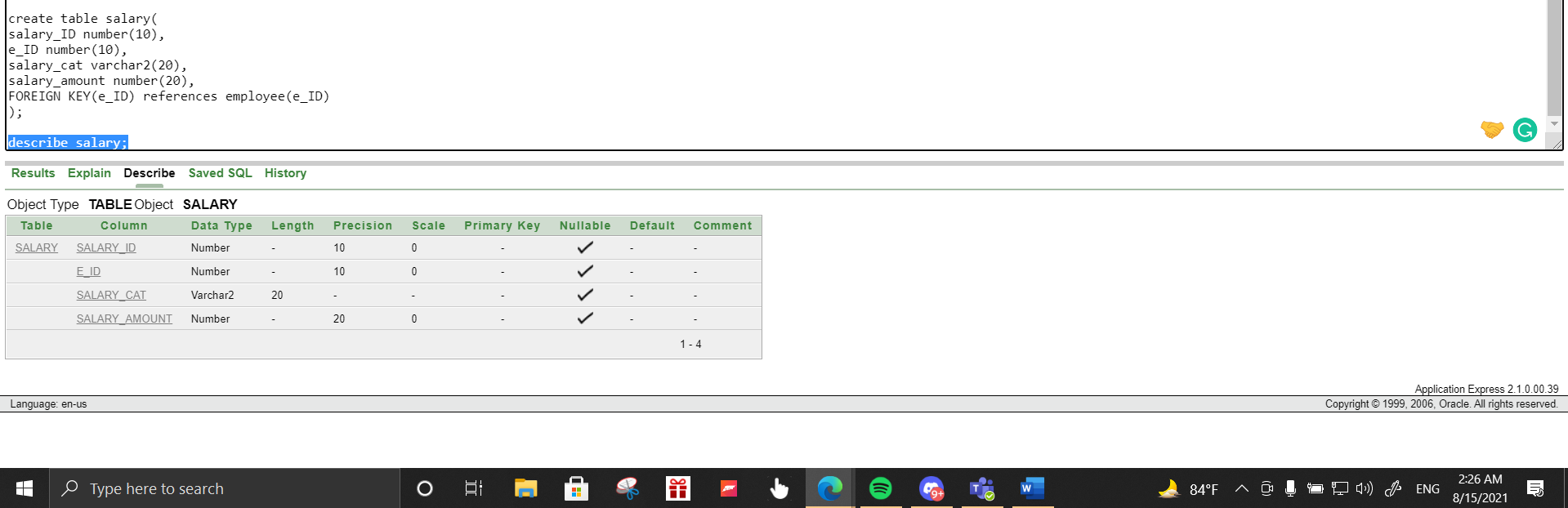
INSERT INTO SALARY VALUES(12, 102, 'MONTHLY', 30000);

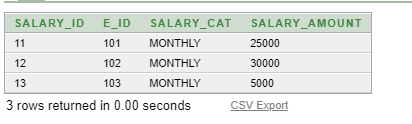
INSERT INTO SALARY VALUES(13, 103, 'MONTHLY', 5000);

select \* from salary;

describe salary;

drop table salary;





* create table appointment(

app\_ID number(10) PRIMARY KEY NOT NULL,

c\_ID number(10) NOT NULL,

e\_ID number(10) NOT NULL,

app\_date date NOT NULL,

FOREIGN KEY(e\_ID) references employee(e\_ID),

FOREIGN KEY(c\_ID) references customer(c\_ID)

);

INSERT INTO APPOINTMENT VALUES(1101, 2, 103, '7-AUG-2021');

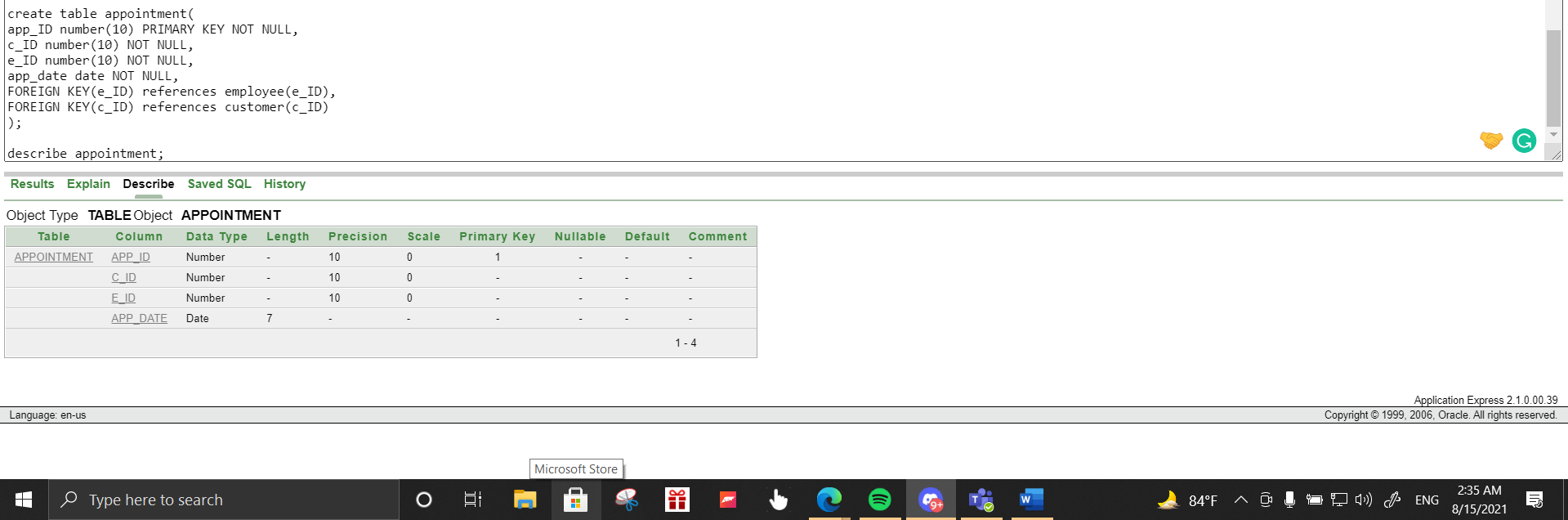
INSERT INTO APPOINTMENT VALUES(1102, 1, 101, '7-AUG-2021');

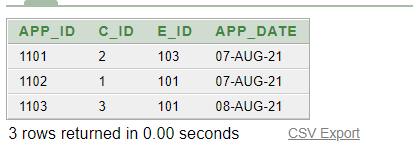
INSERT INTO APPOINTMENT VALUES(1103, 3, 101, '8-AUG-2021');

select \* from appointment;

describe appointment;

drop table appointment;





* create table service(

service\_ID number(10) PRIMARY KEY NOT NULL,

c\_ID number(10) NOT NULL,

e\_ID number(10) NOT NULL,

FOREIGN KEY (c\_ID) references customer(c\_ID),

FOREIGN KEY (e\_ID) references employee(e\_ID)

);

INSERT INTO SERVICE VALUES(1201, 2,103);

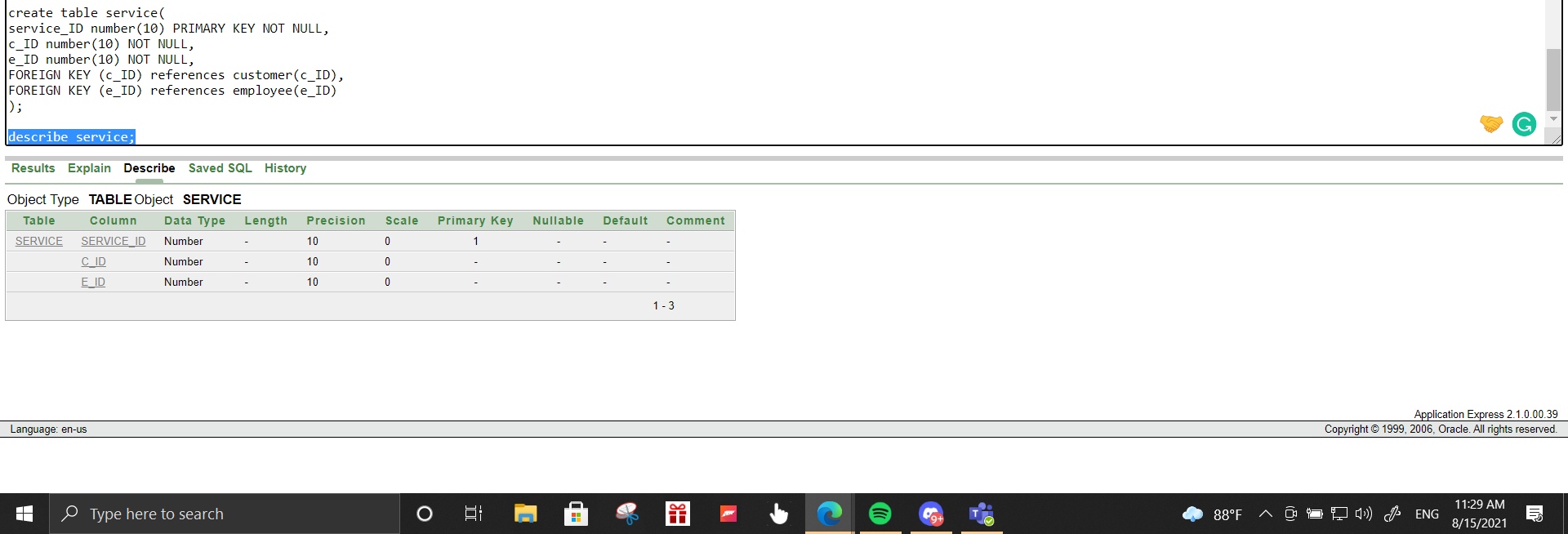
INSERT INTO SERVICE VALUES(1202, 1,101);

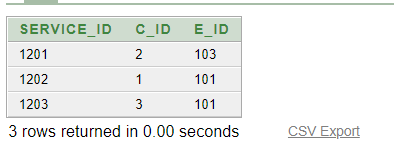
INSERT INTO SERVICE VALUES(1203, 3,101);

select \* from service;

describe service;

drop table service;





* create table invoice(

invoice\_ID number(10) PRIMARY KEY NOT NULL,

service\_ID number(10) NOT NULL,

invoice\_amount number(10) NOT NULL,

FOREIGN KEY (service\_ID) references service(service\_ID)

);

INSERT INTO INVOICE VALUES(1301, 1201, 1200);

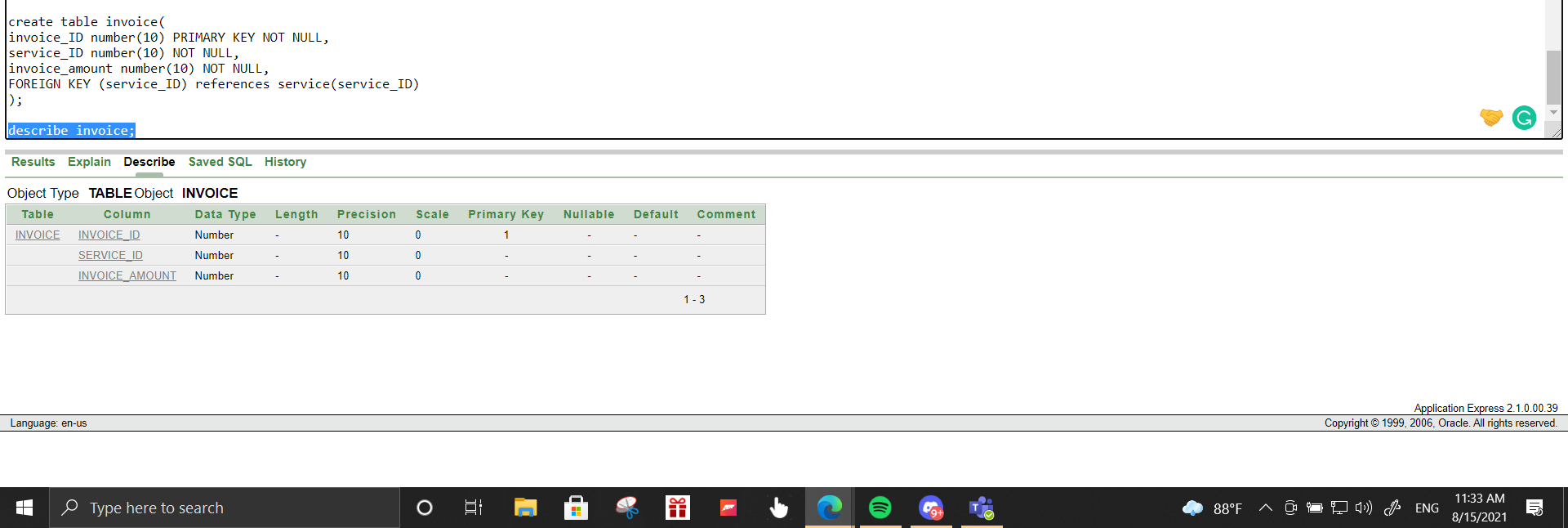
INSERT INTO INVOICE VALUES(1302, 1202, 1500);

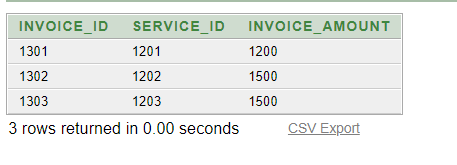
INSERT INTO INVOICE VALUES(1303, 1203, 1500);

select \* from invoice;

describe invoice;

drop table invoice;





* create table payment(

p\_ID number(10) PRIMARY KEY NOT NULL,

invoice\_ID number(10) NOT NULL,

c\_ID number(10) NOT NULL,

p\_type varchar2(10) NOT NULL,

p\_date date NOT NULL,

p\_amount number(10) NOT NULL,

FOREIGN KEY(c\_ID) references customer(c\_ID),

FOREIGN KEY(invoice\_ID) references invoice(invoice\_ID)

);

INSERT INTO PAYMENT VALUES(1401, 1301, 2, 'CASH', '7-AUG-2021', 1200);

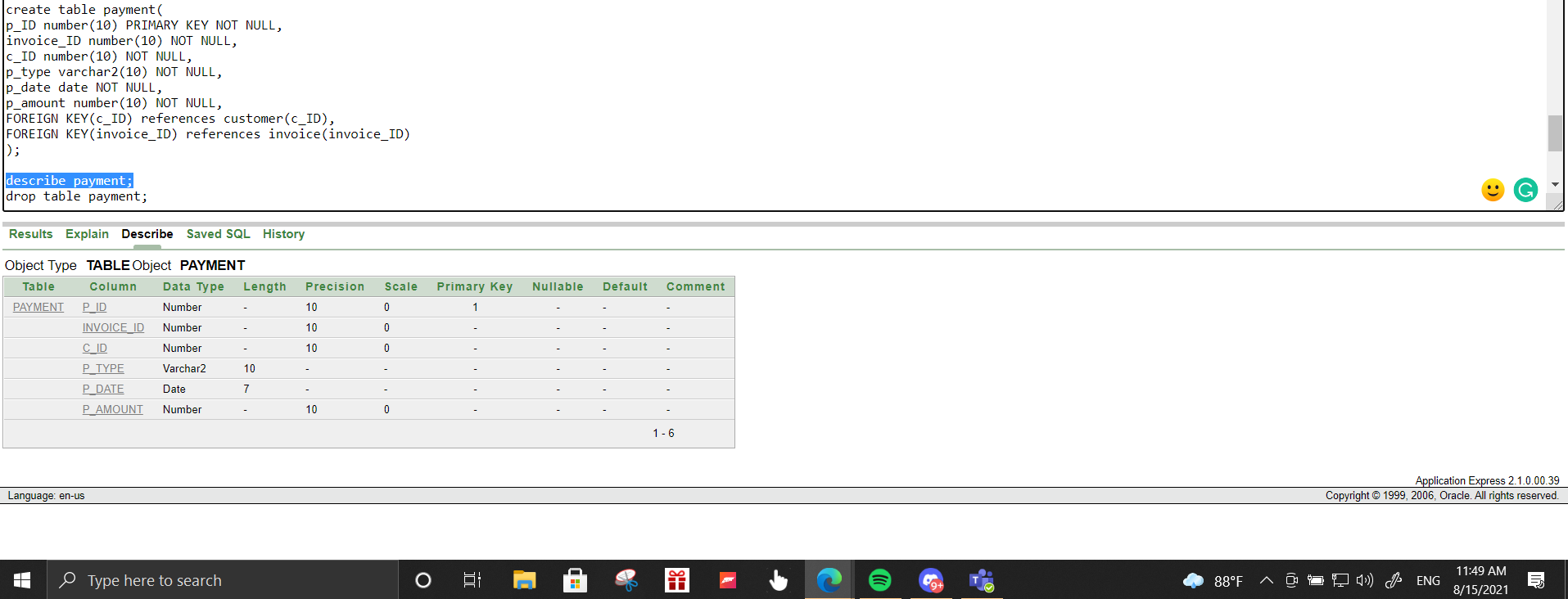
INSERT INTO PAYMENT VALUES(1402, 1302, 1, 'CARD', '7-AUG-2021', 1500);

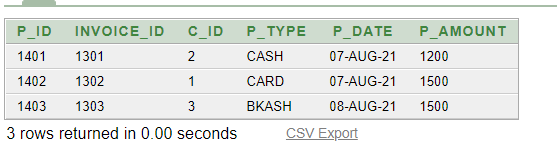
INSERT INTO PAYMENT VALUES(1403, 1303, 3, 'BKASH', '8-AUG-2021', 1500);

select \* from payment;

describe payment;

drop table payment;





* create table phone(

phone\_ID number(10) PRIMARY KEY NOT NULL,

c\_ID number(10),

e\_ID number(10),

phone\_No number(11) NOT NULL,

FOREIGN KEY (c\_ID) references customer(c\_ID),

FOREIGN KEY (e\_ID) references employee(e\_ID)

);

INSERT INTO PHONE(phone\_ID, e\_ID, phone\_No) VALUES(21, 101, 01798673314);

INSERT INTO PHONE(phone\_ID, e\_ID, phone\_No) VALUES(22, 102, 01798673231);

INSERT INTO PHONE(phone\_ID, e\_ID, phone\_No) VALUES(23, 103, 01798679832);

INSERT INTO PHONE(phone\_ID, c\_ID, phone\_No) VALUES(31, 1, 01598673314);

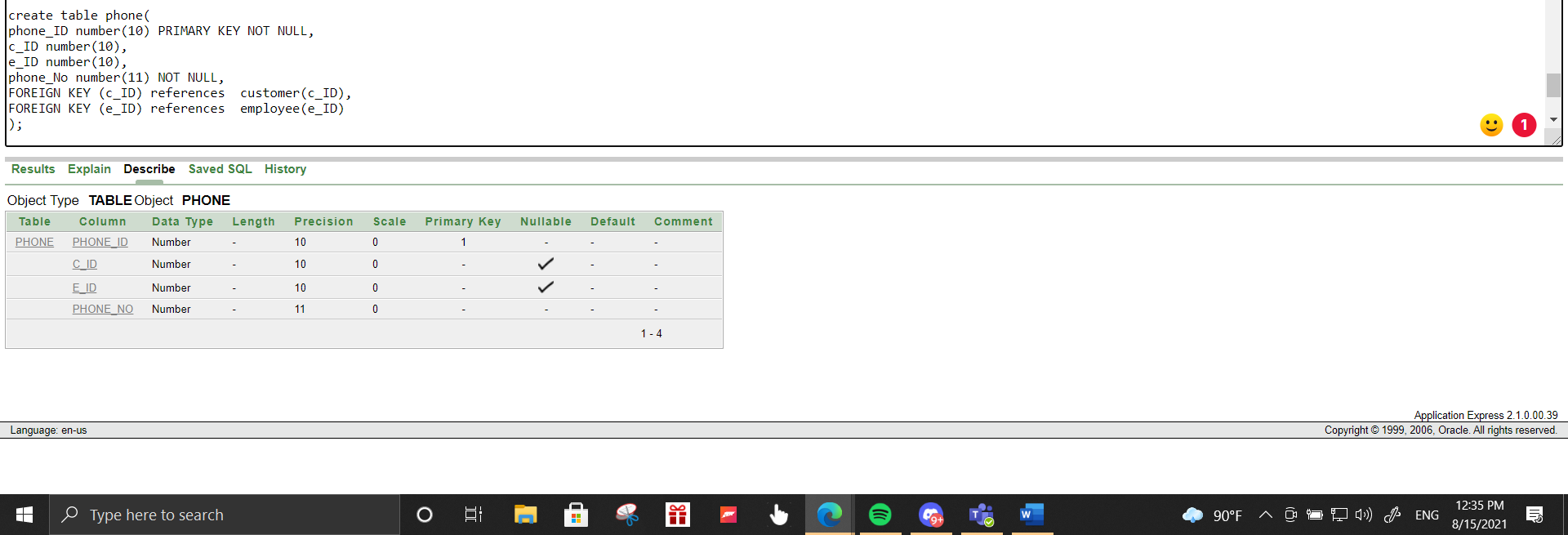
INSERT INTO PHONE(phone\_ID, c\_ID, phone\_No) VALUES(32, 2, 01769773314);

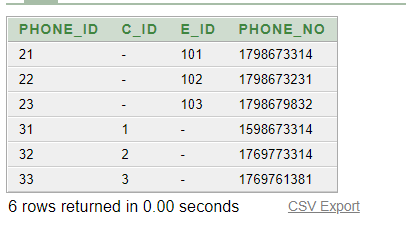
INSERT INTO PHONE(phone\_ID, c\_ID, phone\_No) VALUES(33, 3, 01769761381);

select \* from phone;

describe phone;

drop table phone;





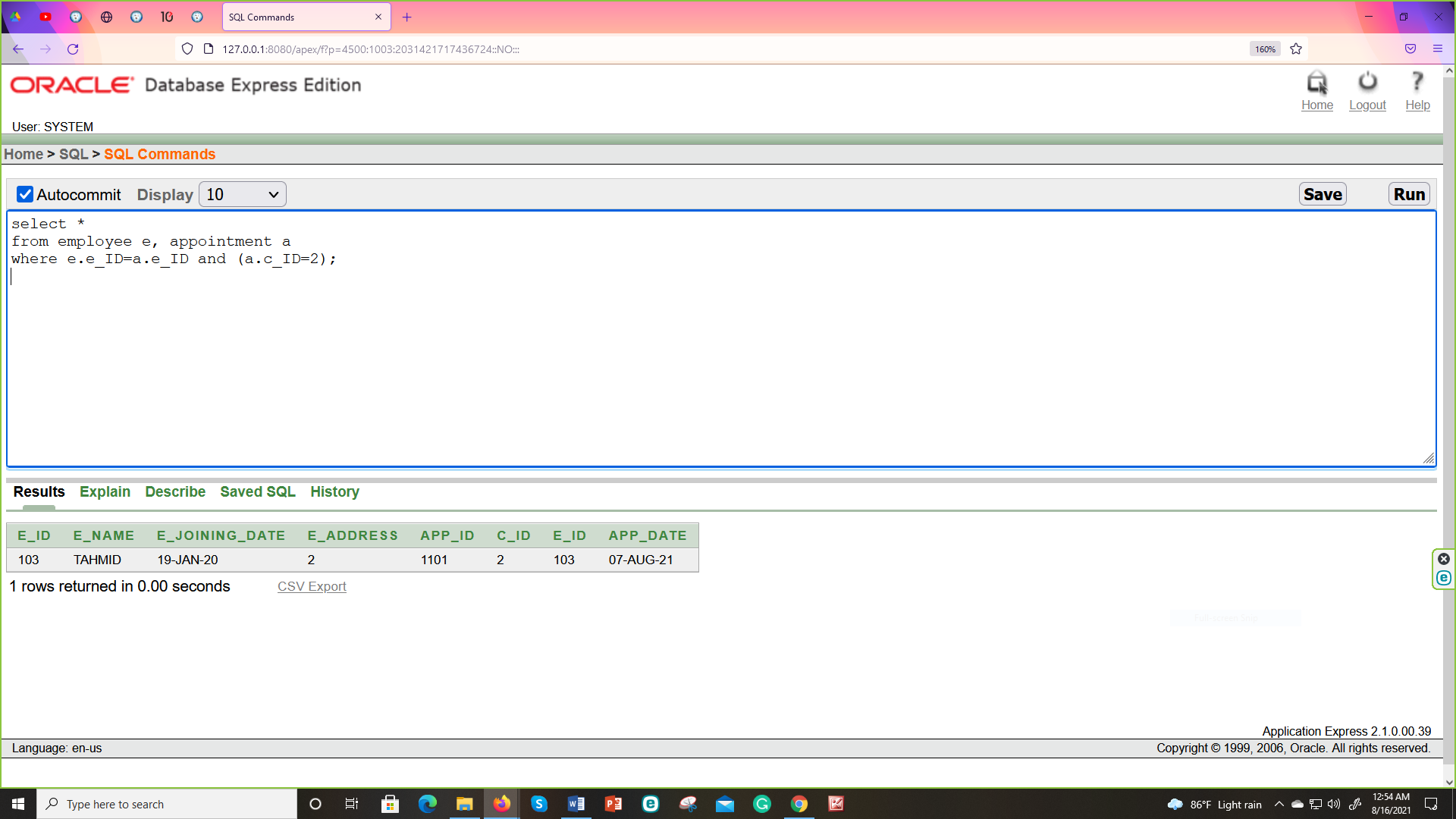
Joining:-

1. Display employee who is appointed to customer Mr.Z

* select \*

from employee e, appointment a

where e.e\_ID=a.e\_ID and (a.c\_ID=2);

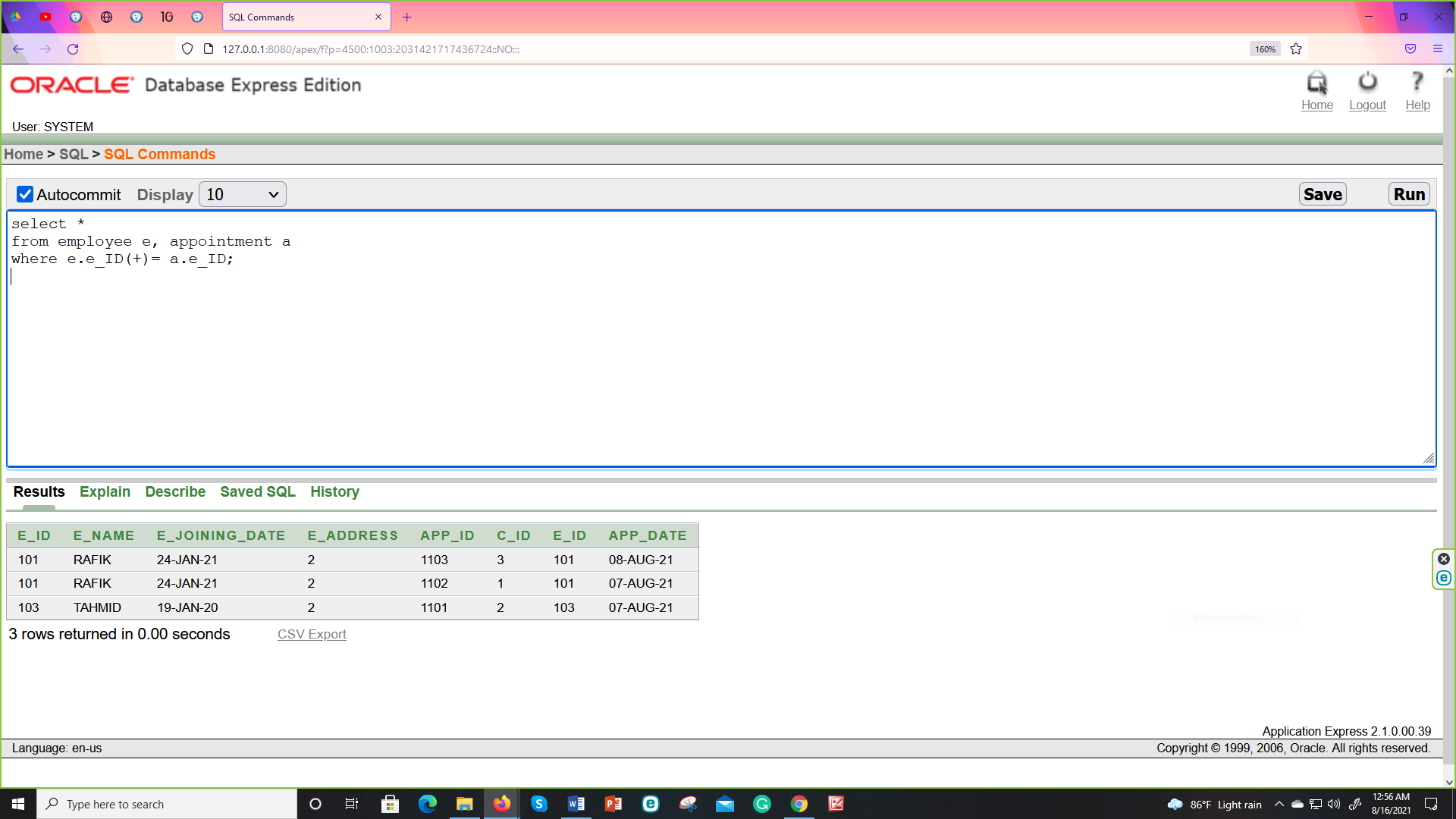


1. Get all the matching & non-matching records from employee and appointment.

* select \*

from employee e, appointment a

where e.e\_ID(+)= a.e\_ID;

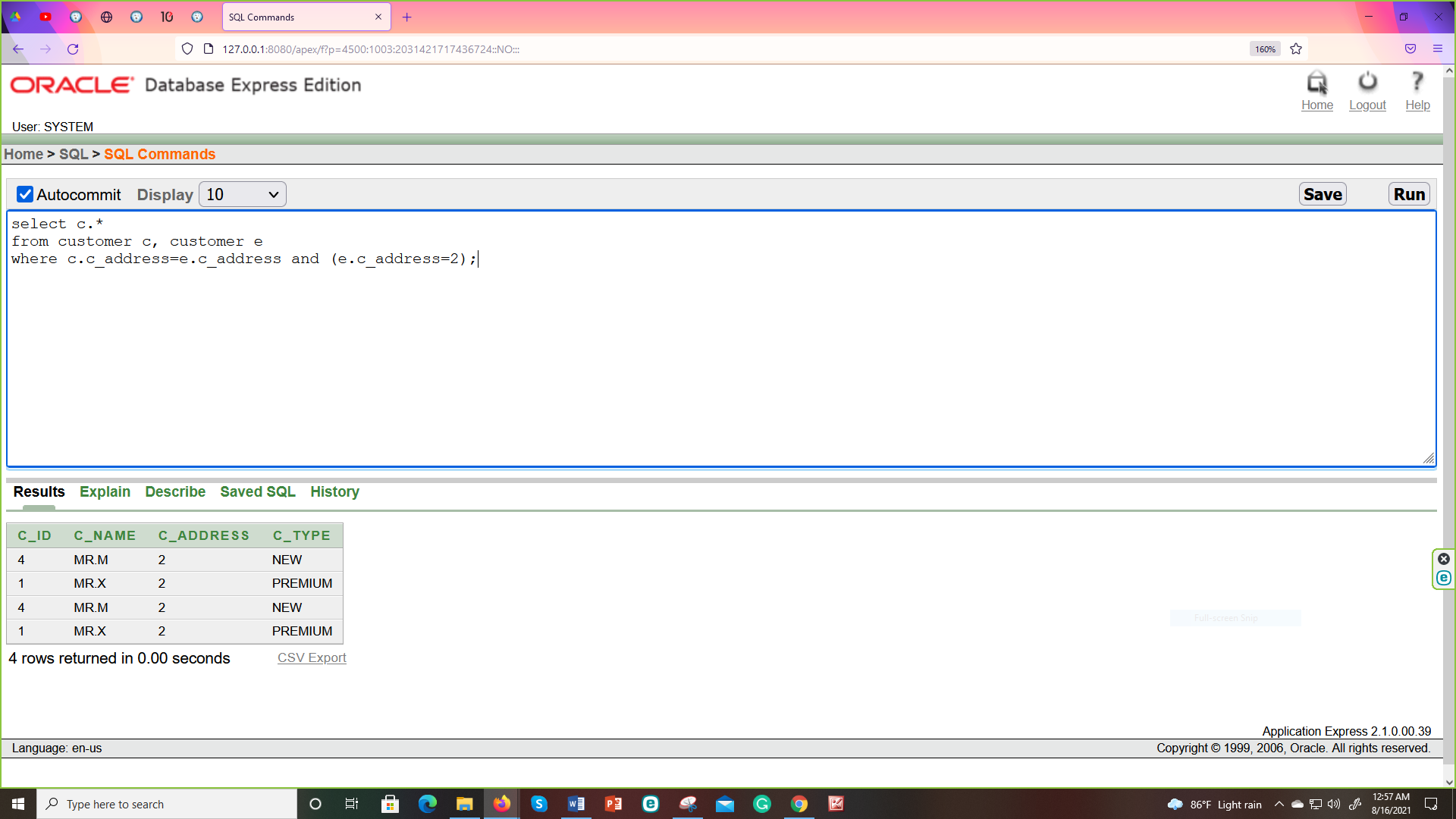


1. Get all the employee who have the same address as MR.X

* select c.\*

from customer c, customer e

where c.c\_address=e.c\_address and (e.c\_address=2);



Sub-Query:-

1. Show the name of the employee who lives in DHAKA?

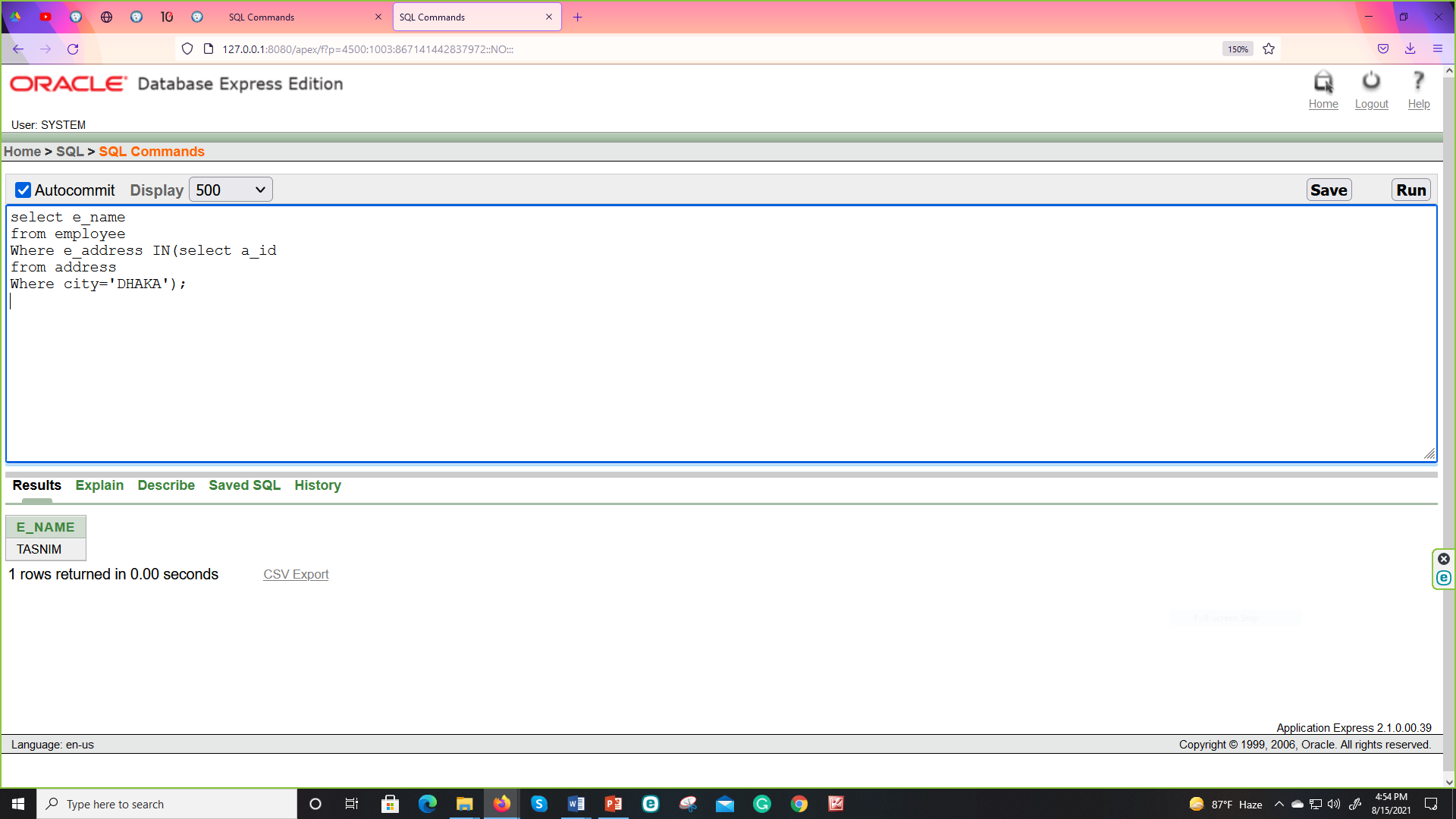
* select e\_name

from employee

Where e\_address IN(select a\_id

from address

Where city='DHAKA');



1. Show the name of the customer who’s paytype is CASH?

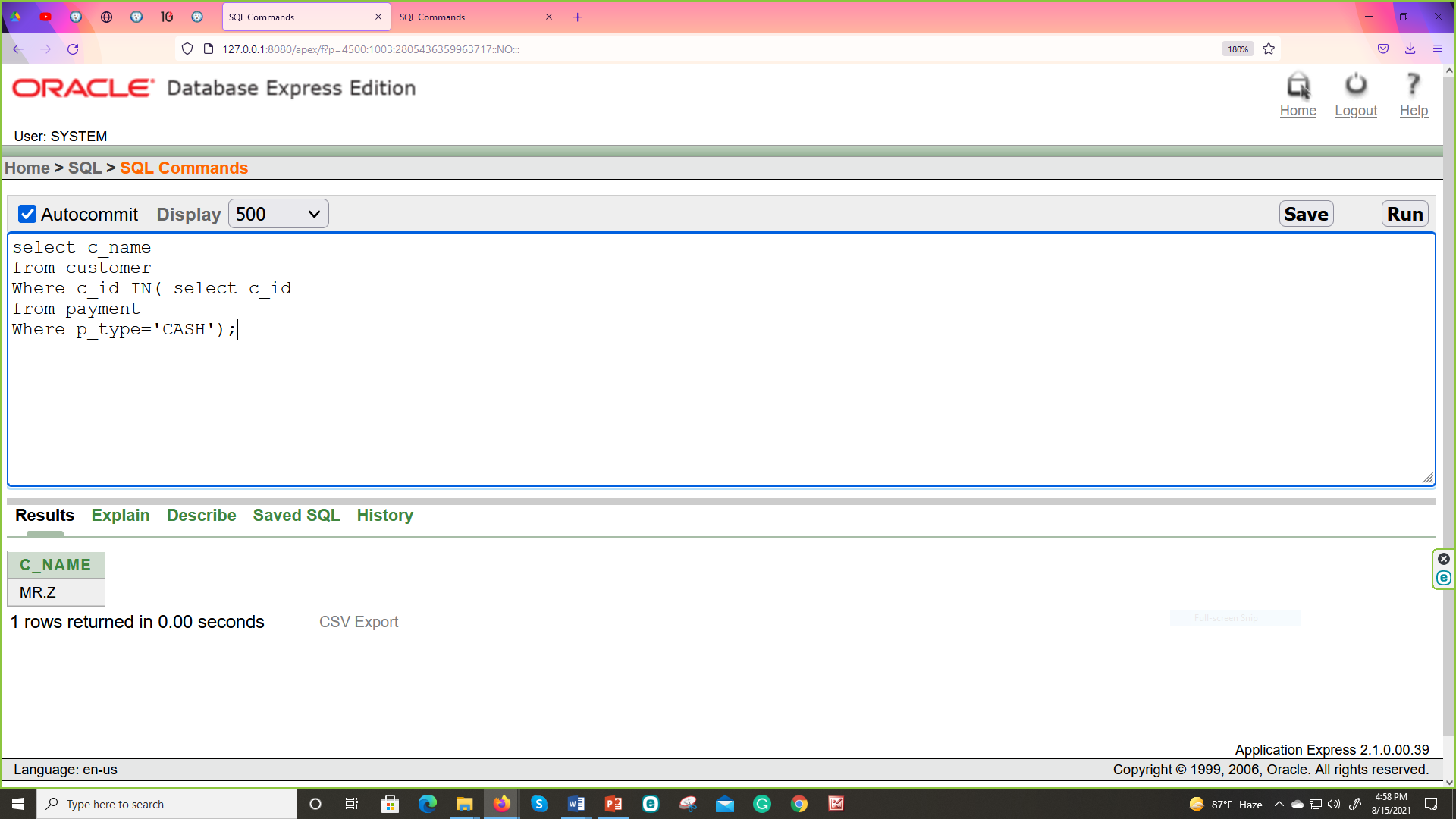
* select c\_name

from customer

Where c\_id IN( select c\_id

from payment

Where p\_type='CASH');



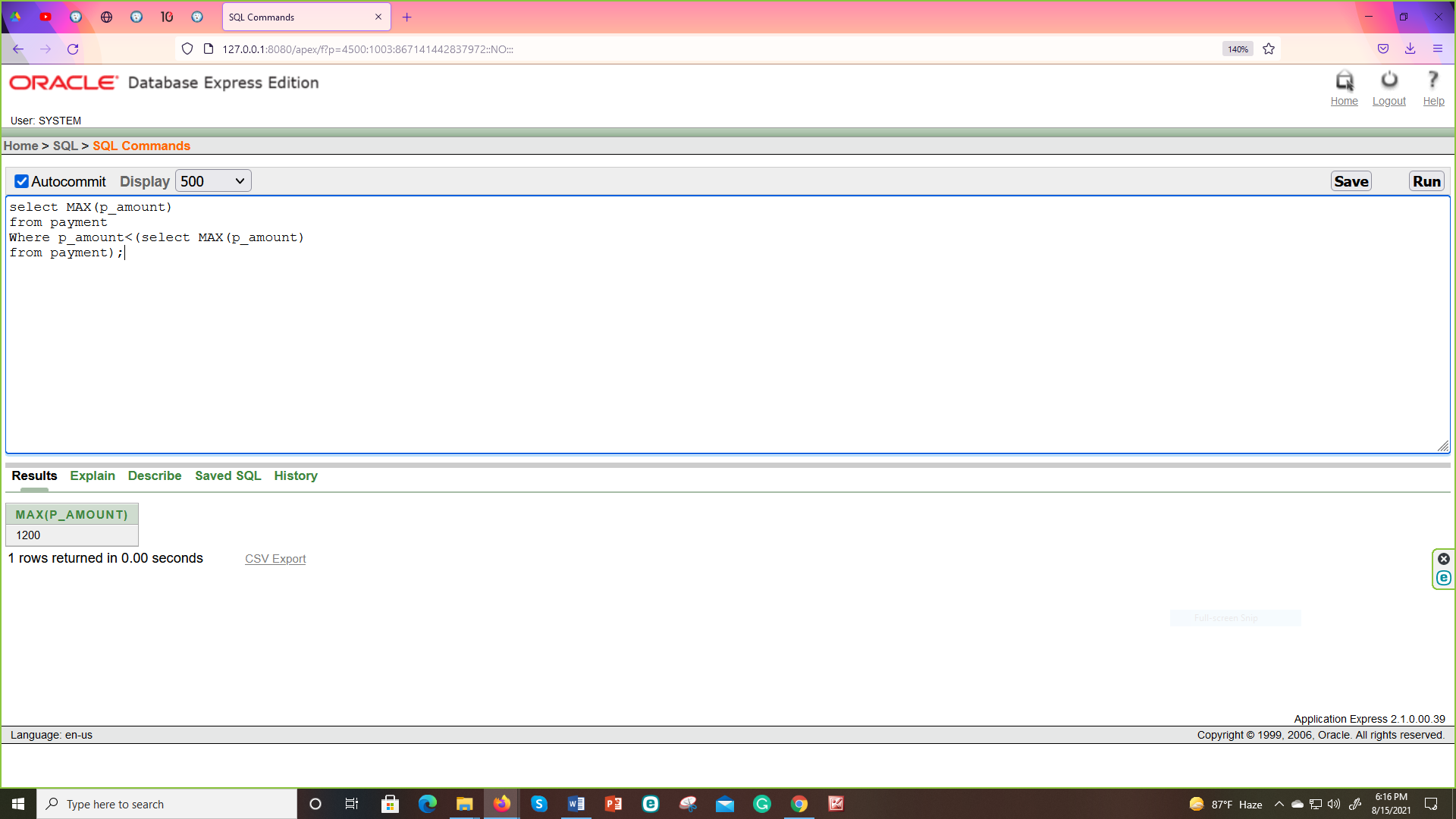
1. Display the second maximum paid amount?

* select MAX(p\_amount)

from payment

Where p\_amount<(select MAX(p\_amount)

from payment);



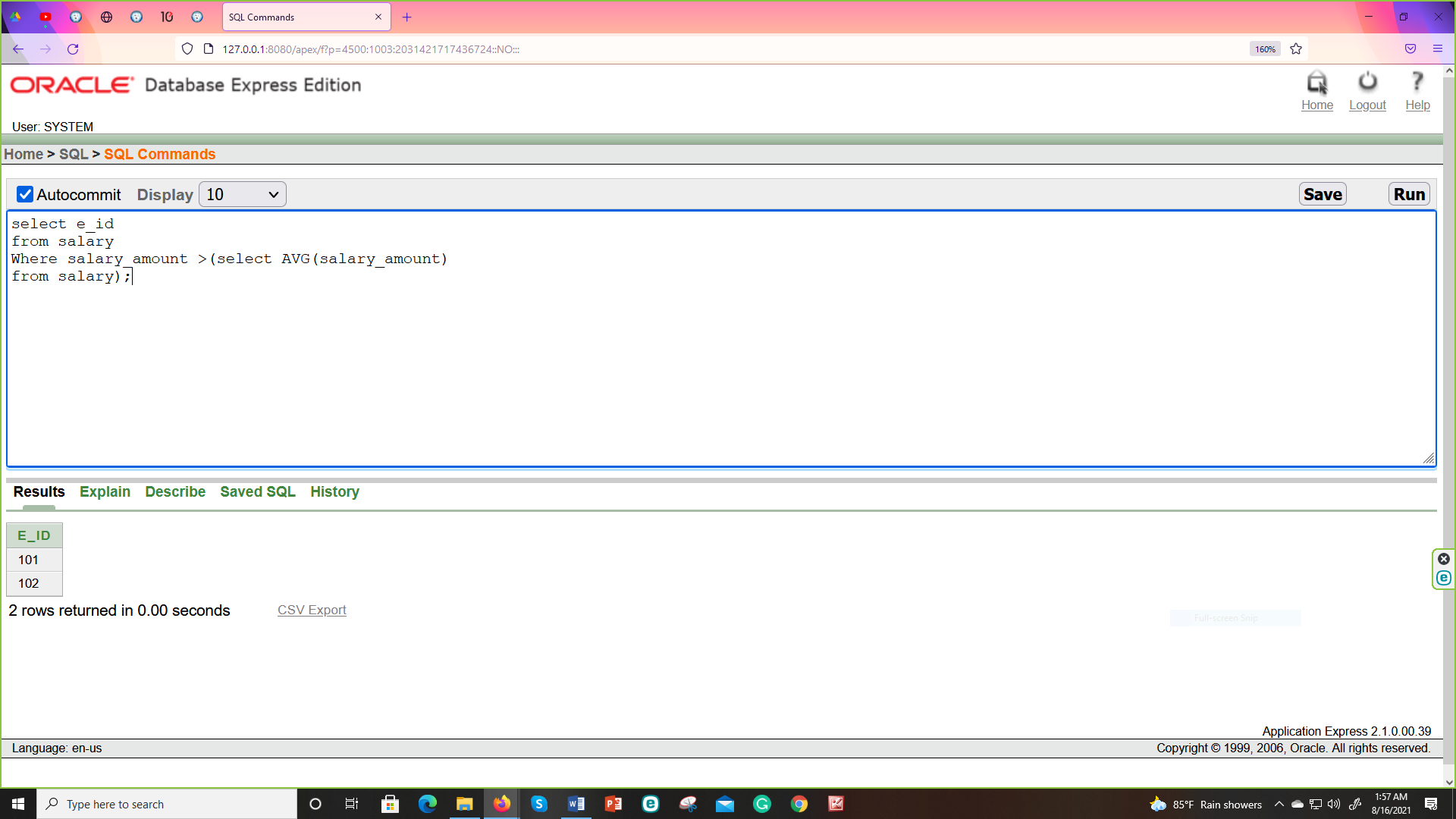
1. Display all the employee's IDs who are getting more than the average salaries of all the employees.

* select e\_id

from salary

Where salary\_amount >(select AVG(salary\_amount)

from salary);



View:-

* CREATE OR REPLACE VIEW paid as

select e.c\_ID, e.c\_name, i.p\_type, i.p\_amount

from customer e, payment i

where e.c\_ID=i.c\_ID and i.p\_amount=(select MIN(p\_amount) from payment );

select \* from paid;

drop view paid;

