Introduction to Database

Finalterm Assignment

Name	ID
1. MD. Mohituzzaman Bhuiyan	20-42546-1
2. Mostaque Ahammed Naim	20-42140-1
3. Joysree Dey Sukanna	20-41946-1
4. Amit Podder	20-42273-1

5. Shakib MD. Shahadot Hossain	20-42385-1

Name of the Assignment

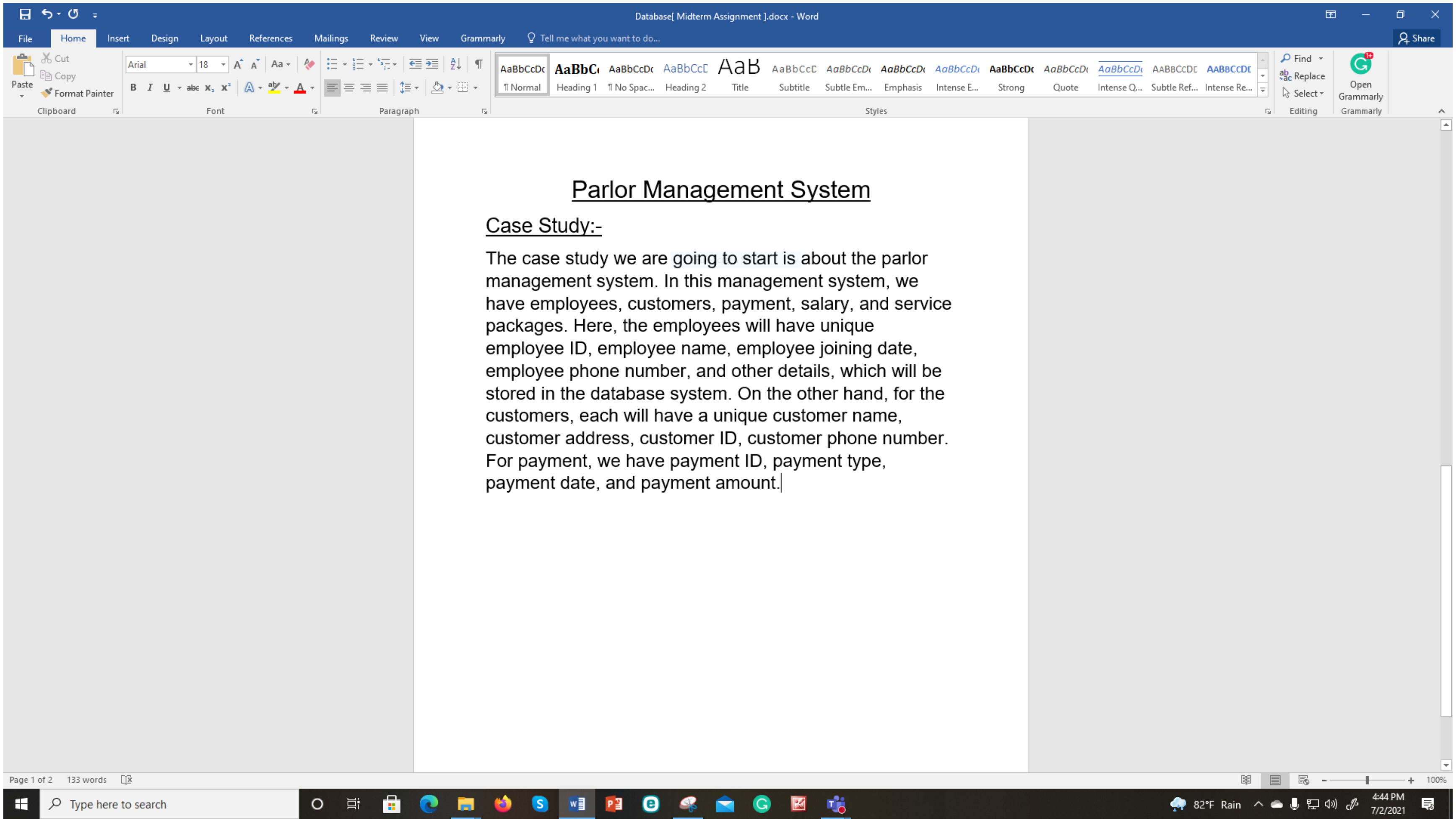
- Parlor Management System

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:-

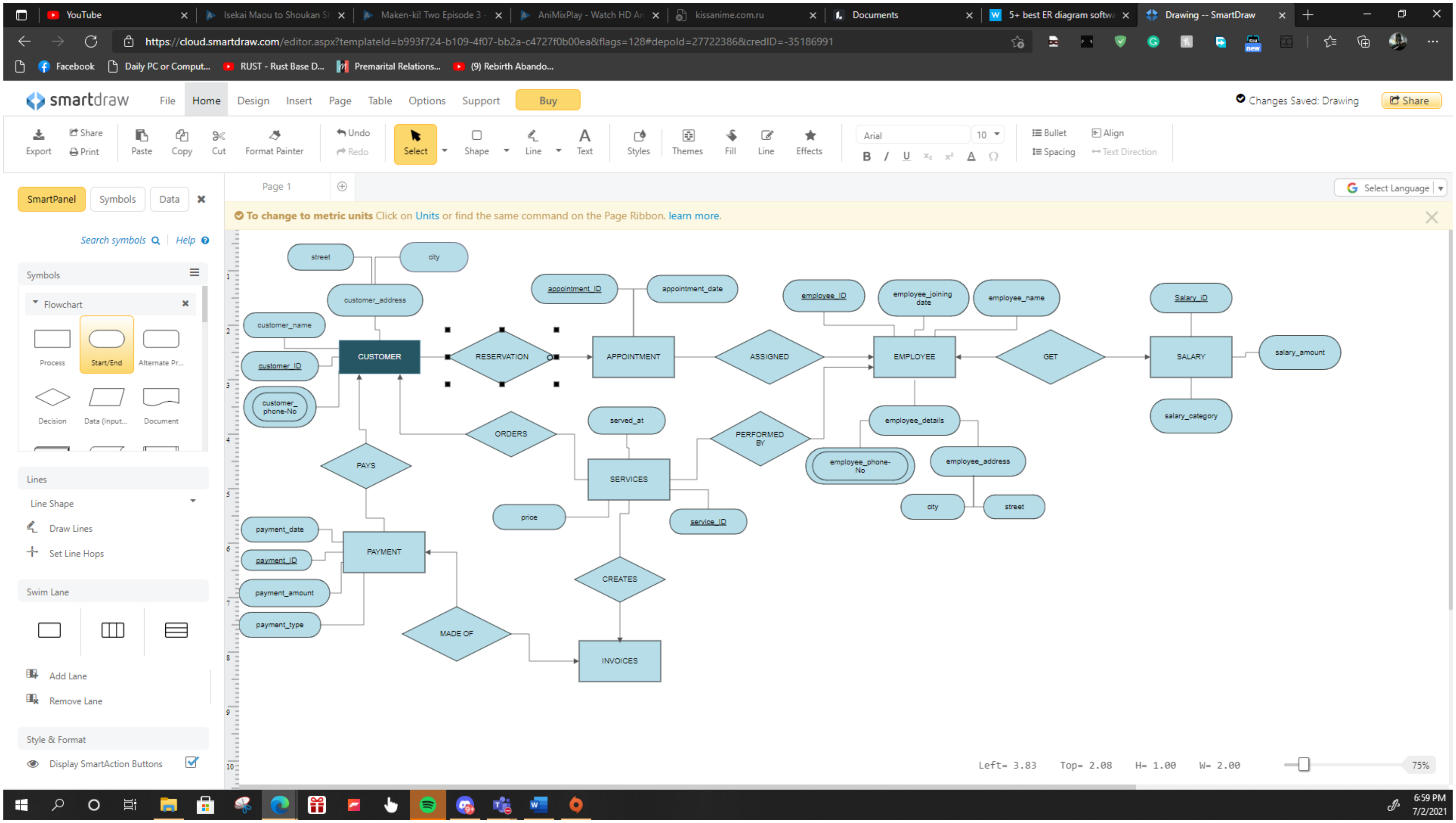


Parlor Management System

Case Study:-

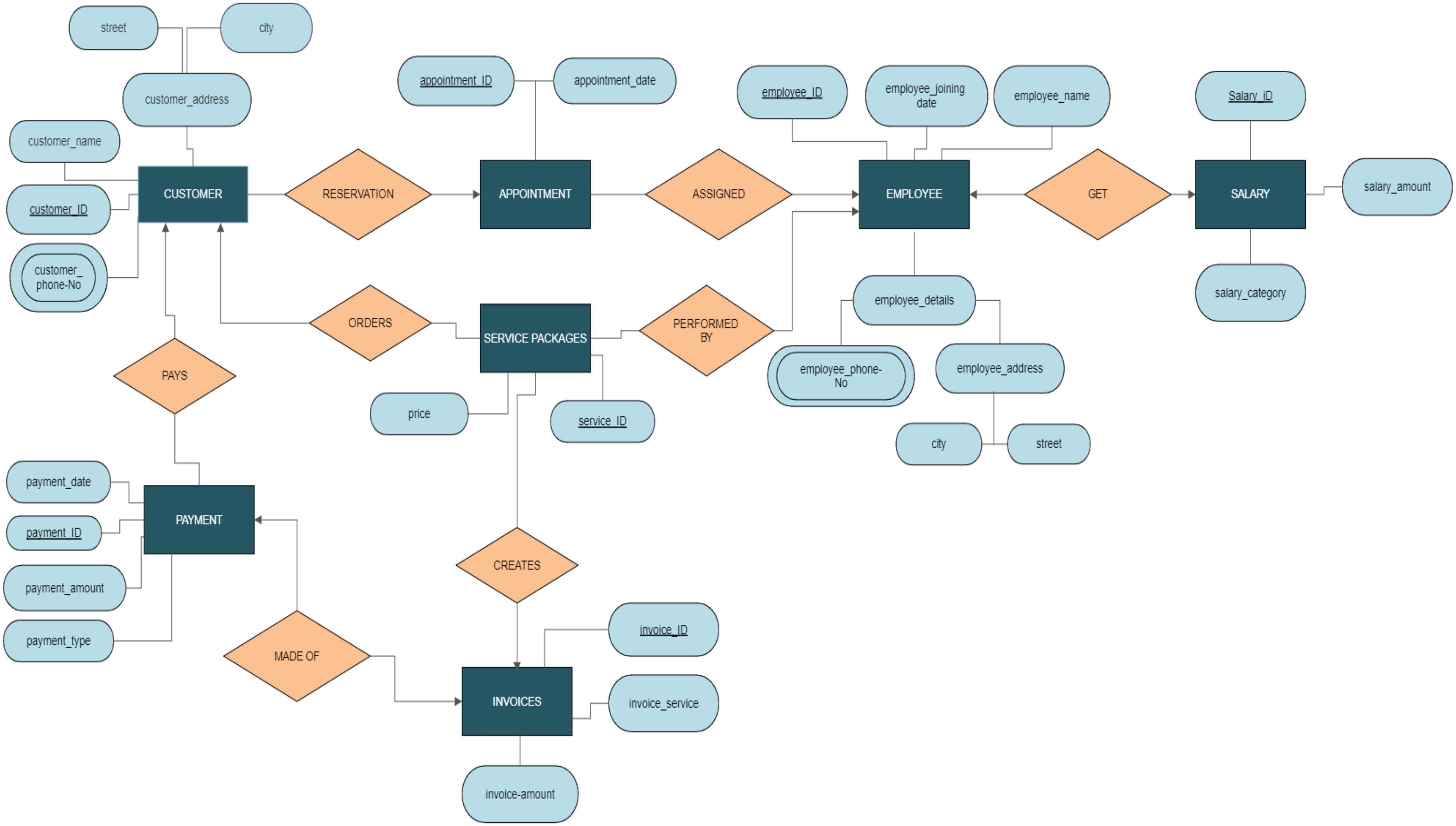
The case study we are going to start is about the parlor management system. In this management system, we have employees, customers, payment, salary, and service packages. Here, the employees will have unique employee ID, employee name, employee joining date, employee phone number, and other details, which will be stored in the database system. On the other hand, for the customers, each will have a unique customer name, customer address, customer ID, customer phone number. For payment, we have payment ID, payment type, payment date, and payment amount.

Before Finishing Screenshot:-



To change to metric units Click on Units or find the same command on the Page Ribbon. learn more.

ER Diagram:-


street
city
customer_address
customer_name
customer_ID
customer_ phone-No
CUSTOMER
RESERVATION
appointment_ID
appointment_date
APPOINTMENT
ASSIGNED
employee_ID
employee_joining date
employee_name
EMPLOYEE
GET
Salary_iD
SALARY
salary_amount
salary_category
ORDERS
SERVICE PACKAGES
PERFORMED BY
employee_details
employee_phone-No
employee_address
city
street
PAYS
price
service_ID
payment_date
payment_ID
payment_amount
payment_type
PAYMENT
CREATES
MADE OF
invoice_ID
invoice_service
INVOICES
invoice-amount

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:-

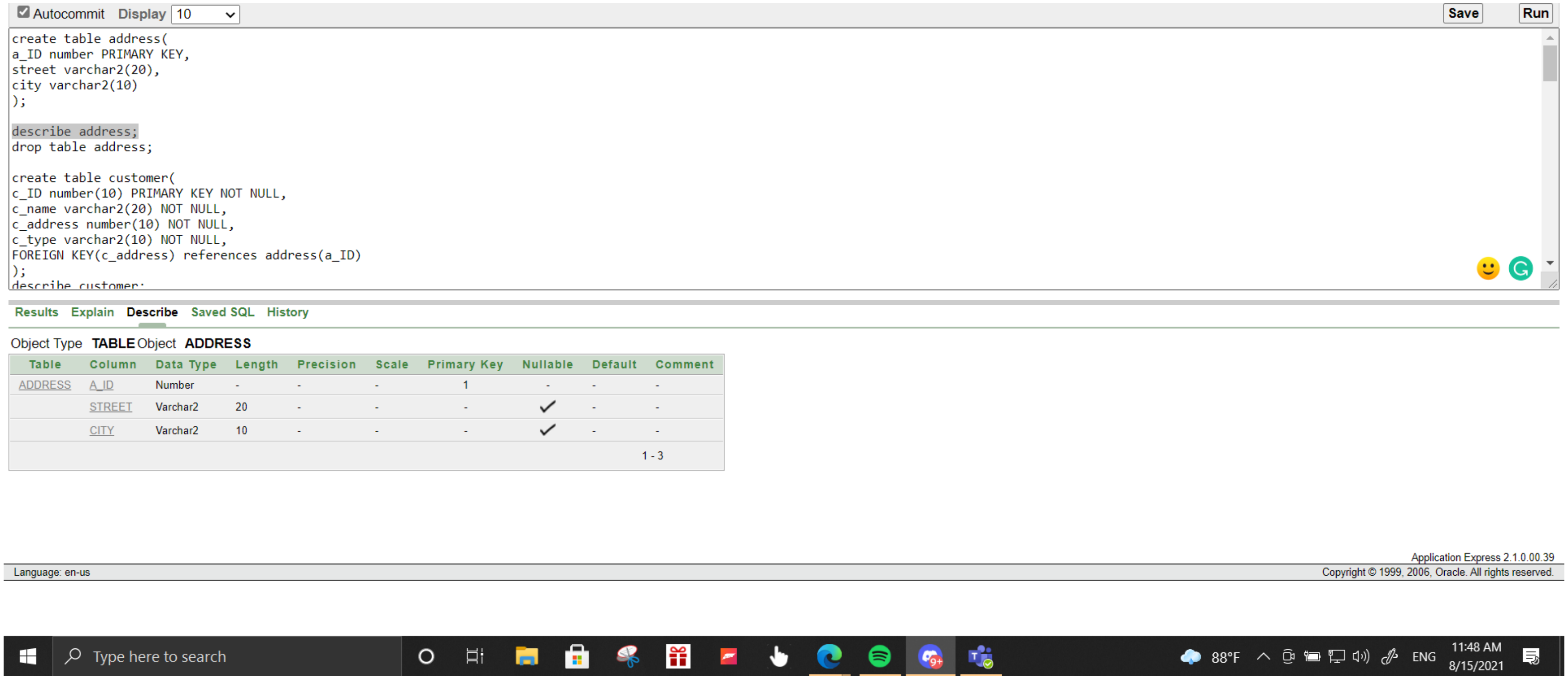
1. customer_id, customer_name, customer_address
2. a_ID, street, city
3. service_ID, employee_ID, employee_phone-no
4. employee_ID, employee_details, employee_joining-date
5. salary_ID, employee_ID, salary_type, salary_amount
6. payment_ID, payment_date, payment_amount, payment_type, customer_ID, invoice_ID
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');
```



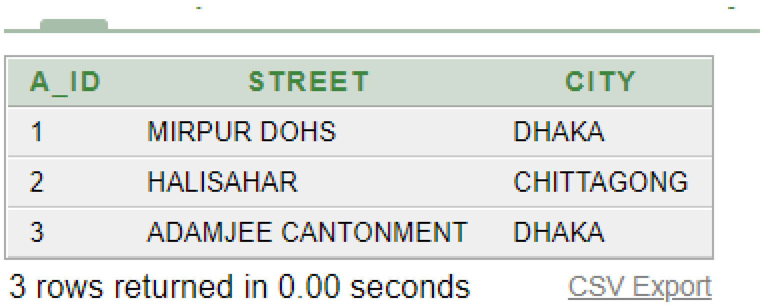
```
create table address(
a_ID number PRIMARY KEY,
street varchar2(20),
city varchar2(10)
);

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)
);
describe customer;
```

Object Type **TABLE** Object **ADDRESS**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ADDRESS	A_ID	Number	-	-	-	1	-	-	-
	STREET	Varchar2	20	-	-	-	✓	-	-
	CITY	Varchar2	10	-	-	-	✓	-	-

1 - 3

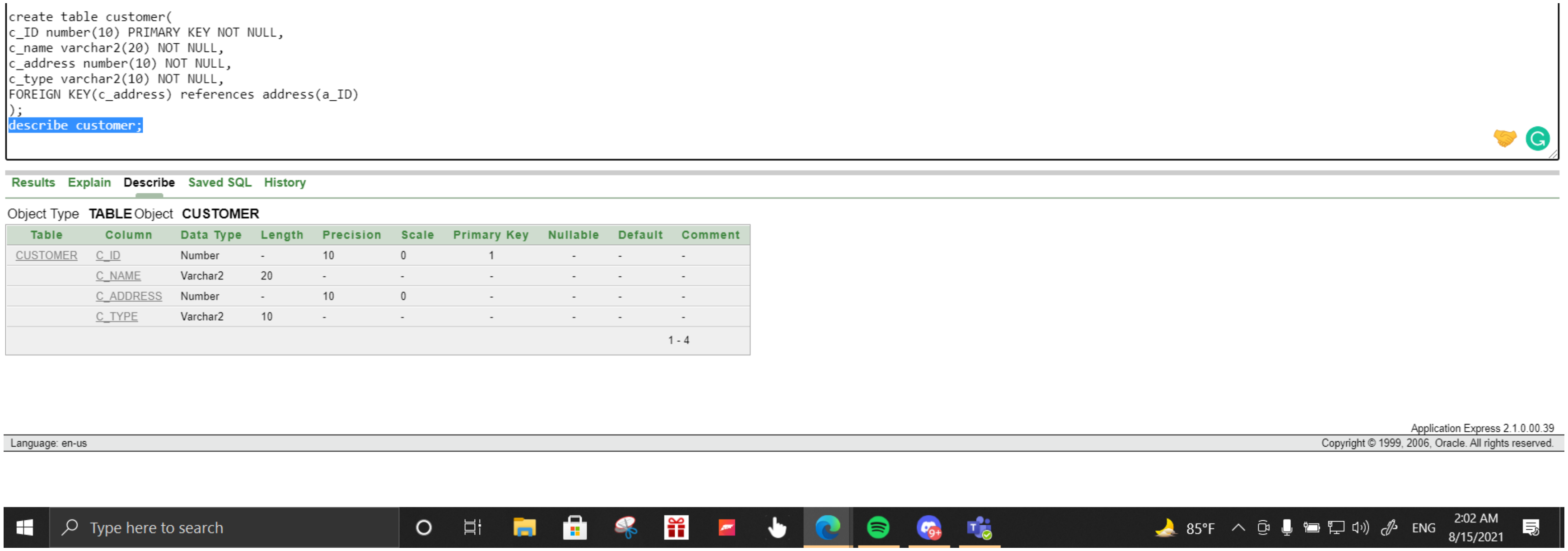
Language: en-us
Application Express 2.1.0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved.
Type here to search
88°F
ENG
11:48 AM
8/15/2021



A_ID	STREET	CITY
1	MIRPUR DOHS	DHAKA
2	HALISAHAR	CHITTAGONG
3	ADAMJEE CANTONMENT	DHAKA

3 rows returned in 0.00 seconds

```
➢ 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');
```



```
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)
);
describe customer;
```

Object Type **TABLE** Object **CUSTOMER**

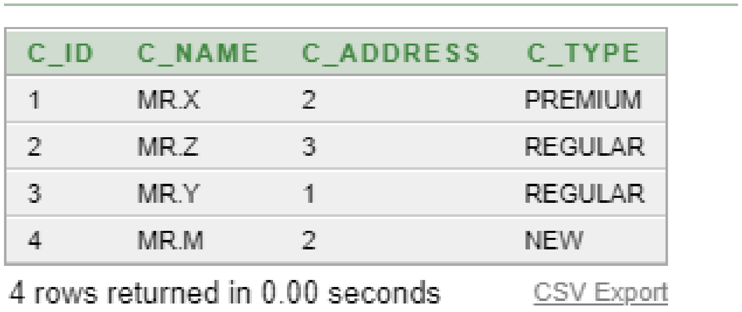
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER	C_ID	Number	-	10	0	1	-	-	-
	C_NAME	Varchar2	20	-	-	-	-	-	-
	C_ADDRESS	Number	-	10	0	-	-	-	-
	C_TYPE	Varchar2	10	-	-	-	-	-	-

1 - 4

Language: en-us

Application Express 2.1.0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved.

Type here to search
85°F
ENG
2:02 AM
8/15/2021



C_ID	C_NAME	C_ADDRESS	C_TYPE
1	MR.X	2	PREMIUM
2	MR.Z	3	REGULAR
3	MR.Y	1	REGULAR
4	MR.M	2	NEW

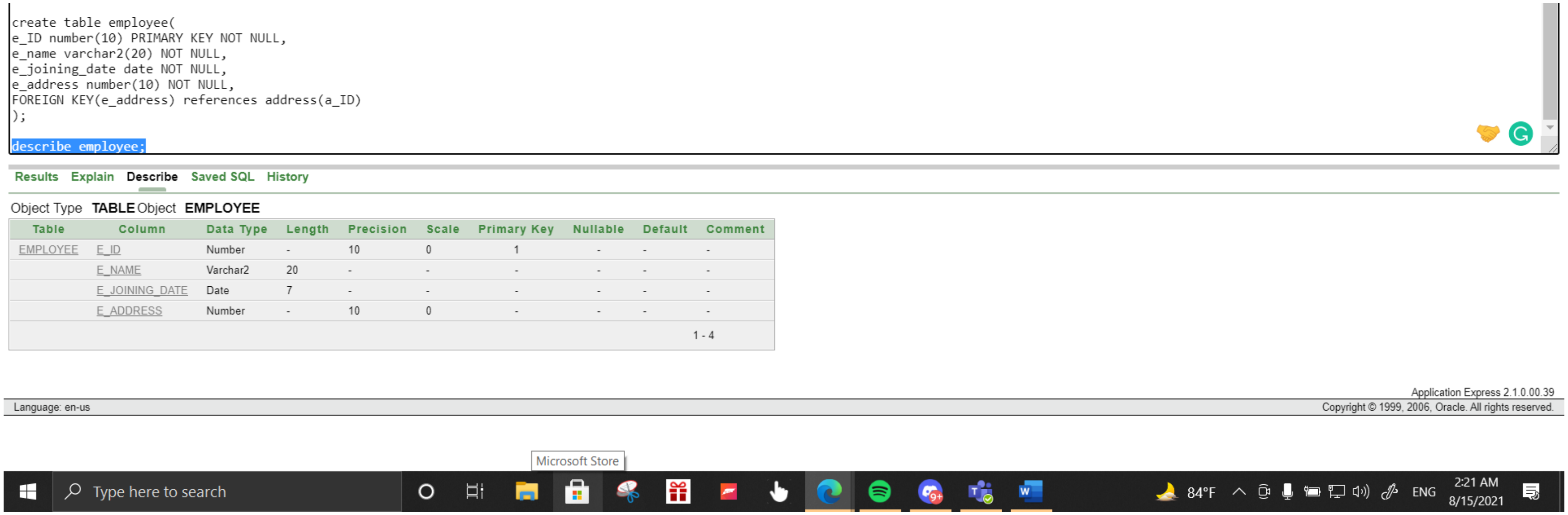
4 rows returned in 0.00 seconds

CSV Export

- 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);
```



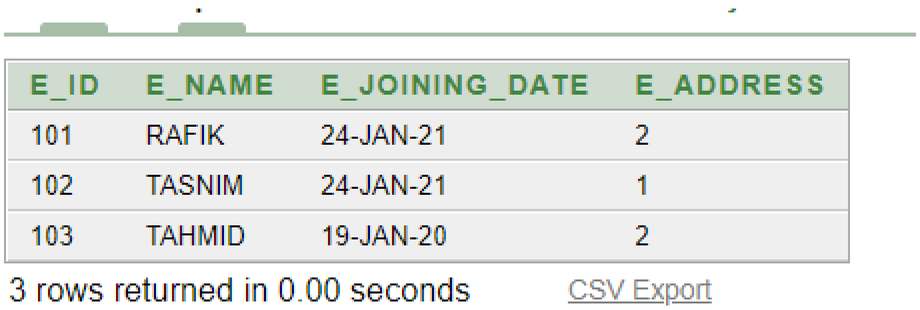
```
create table employee(
e_ID number(10) PRIMARY KEY NOT NULL,
e_name varchar2(20) NOT NULL,
e_joining_date date NOT NULL,
e_address number(10) NOT NULL,
FOREIGN KEY(e_address) references address(a_ID)
);

describe employee;
```

Object Type **TABLE** Object **EMPLOYEE**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMPLOYEE	E_ID	Number	-	10	0	1	-	-	-
	E_NAME	Varchar2	20	-	-	-	-	-	-
	E_JOINING_DATE	Date	7	-	-	-	-	-	-
	E_ADDRESS	Number	-	10	0	-	-	-	-

1 - 4

Language: en-us
Application Express 2.1.0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved.
Microsoft Store
Type here to search
84°F
ENG
2:21 AM
8/15/2021



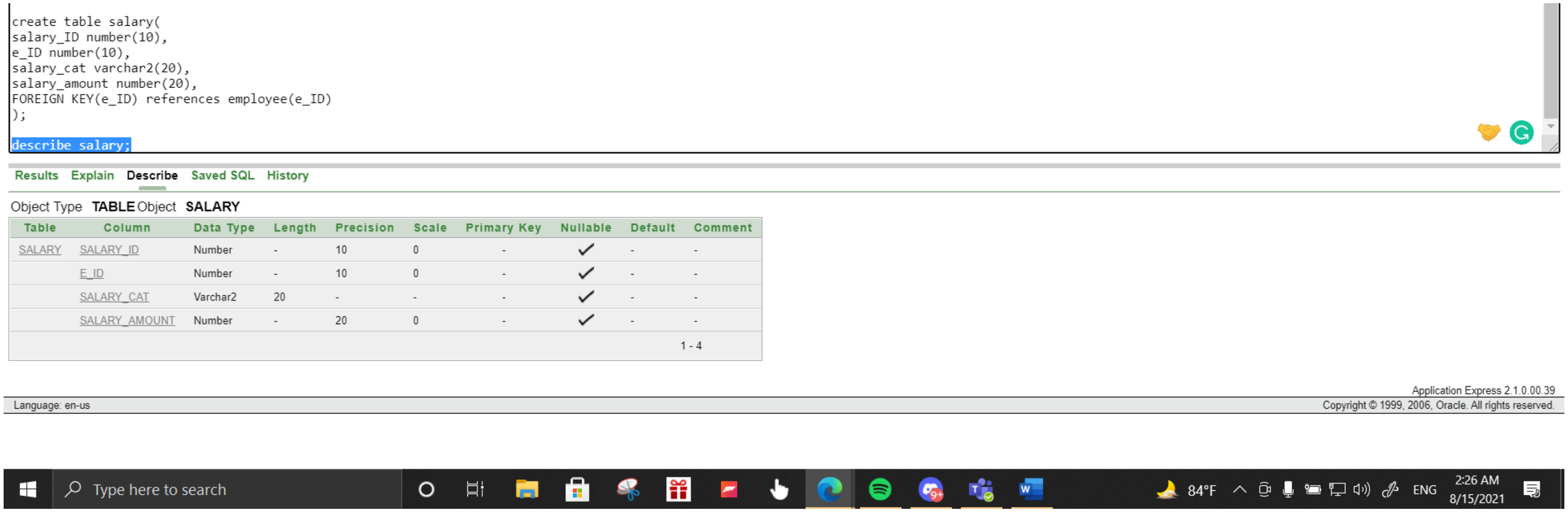
E_ID	E_NAME	E_JOINING_DATE	E_ADDRESS
101	RAFIK	24-JAN-21	2
102	TASNIM	24-JAN-21	1
103	TAHMID	19-JAN-20	2

3 rows returned in 0.00 seconds CSV Export

- 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);
```



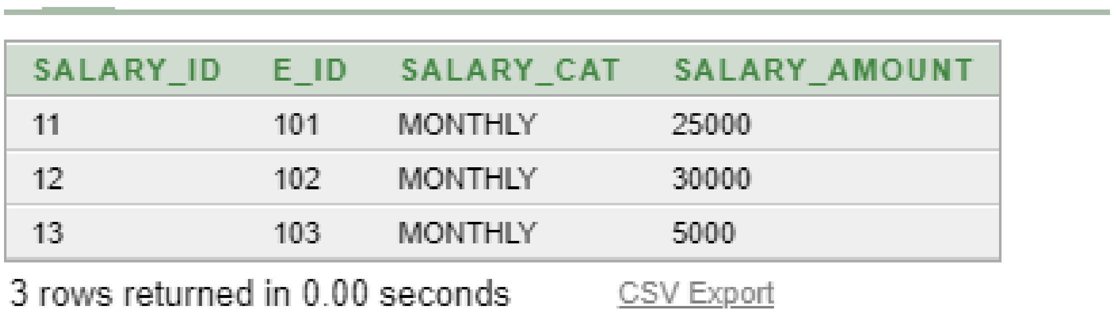
```
create table salary(
salary_ID number(10),
e_ID number(10),
salary_cat varchar2(20),
salary_amount number(20),
FOREIGN KEY(e_ID) references employee(e_ID)
);

describe salary;
```

Object Type **TABLE** Object **SALARY**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
SALARY	SALARY_ID	Number	-	10	0	-	✓	-	-
	E_ID	Number	-	10	0	-	✓	-	-
	SALARY_CAT	Varchar2	20	-	-	-	✓	-	-
	SALARY_AMOUNT	Number	-	20	0	-	✓	-	-
								1 - 4	

Language: en-us

Application Express 2.1.0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved.



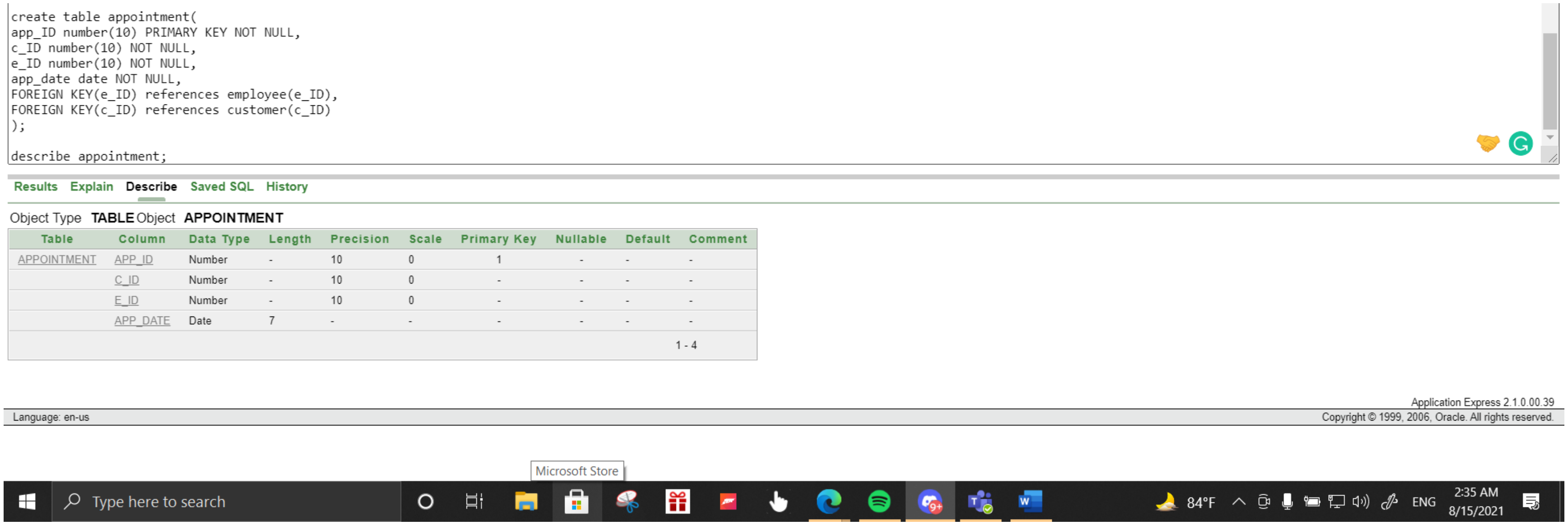
SALARY_ID	E_ID	SALARY_CAT	SALARY_AMOUNT
11	101	MONTHLY	25000
12	102	MONTHLY	30000
13	103	MONTHLY	5000

3 rows returned in 0.00 seconds

- 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');
```



```
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)
);

describe appointment;
```

Object Type **TABLE** Object **APPOINTMENT**

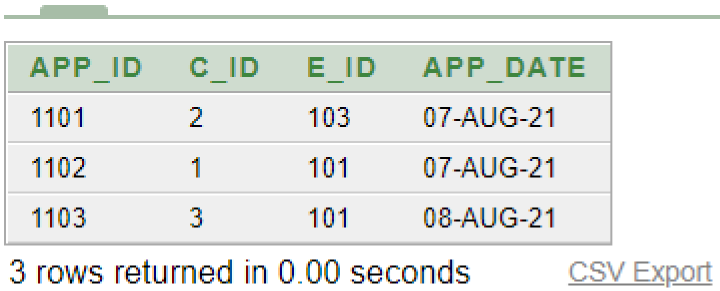
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
APPOINTMENT	APP_ID	Number	-	10	0	1	-	-	-
	C_ID	Number	-	10	0	-	-	-	-
	E_ID	Number	-	10	0	-	-	-	-
	APP_DATE	Date	7	-	-	-	-	-	-

1 - 4

Application Express 2.1.0.00.39

Language: en-us

Copyright © 1999, 2006, Oracle. All rights reserved.



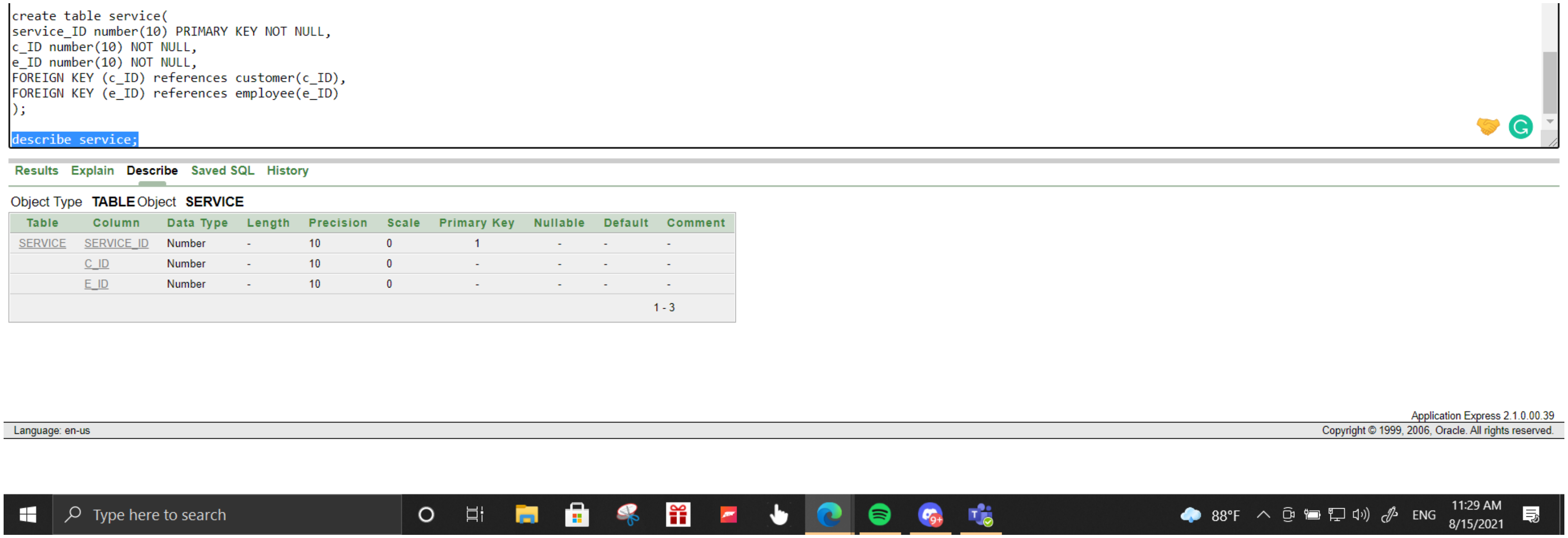
APP_ID	C_ID	E_ID	APP_DATE
1101	2	103	07-AUG-21
1102	1	101	07-AUG-21
1103	3	101	08-AUG-21

3 rows returned in 0.00 seconds

- 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);
```



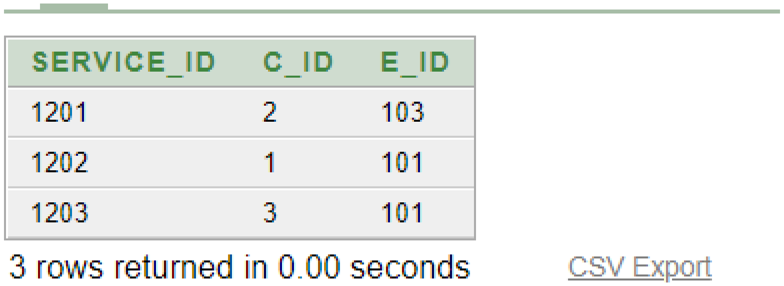
```
create table service(
service_ID number(10) PRIMARY KEY NOT NULL,
c_ID number(10) NOT NULL,
e_ID number(10) NOT NULL,
FOREIGN KEY (c_ID) references customer(c_ID),
FOREIGN KEY (e_ID) references employee(e_ID)
);

describe service;
```

Object Type TABLE Object SERVICE

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
SERVICE	SERVICE_ID	Number	-	10	0	1	-	-	-
	C_ID	Number	-	10	0	-	-	-	-
	E_ID	Number	-	10	0	-	-	-	-

1 - 3

Application Express 2.1.0.00.39
Language: en-us
Copyright © 1999, 2006, Oracle. All rights reserved.
Type here to search
88°F
ENG
11:29 AM
8/15/2021



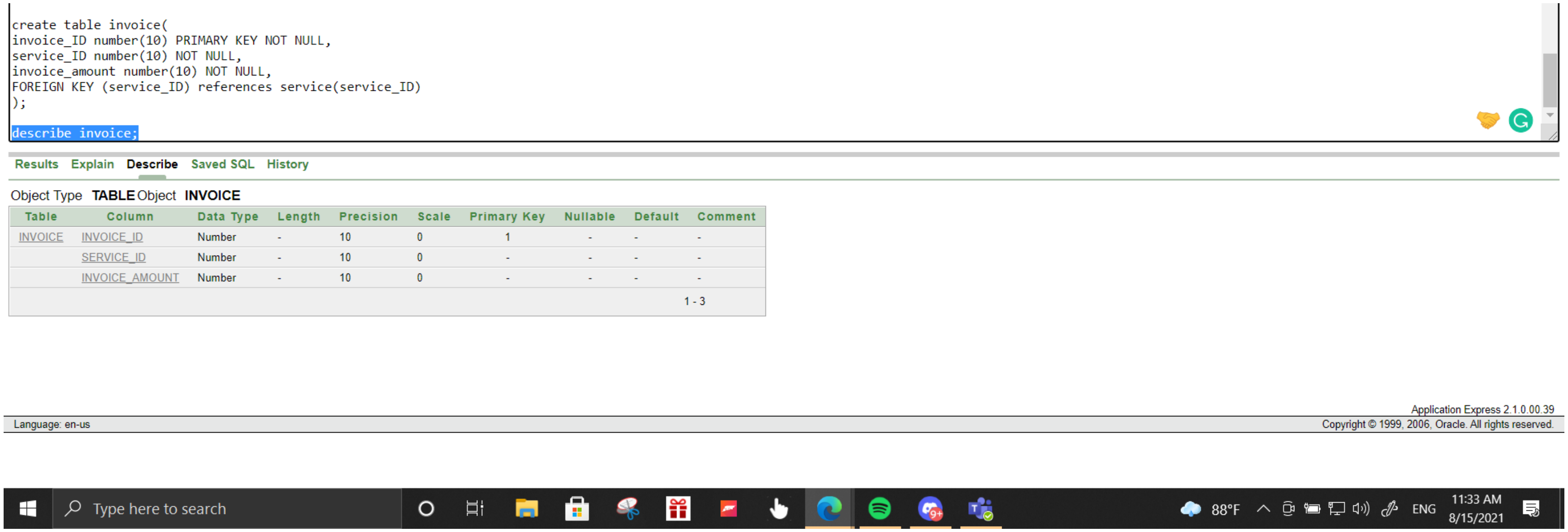
SERVICE_ID	C_ID	E_ID
1201	2	103
1202	1	101
1203	3	101

3 rows returned in 0.00 seconds

- 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);
```



```
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)
);

describe invoice;
```

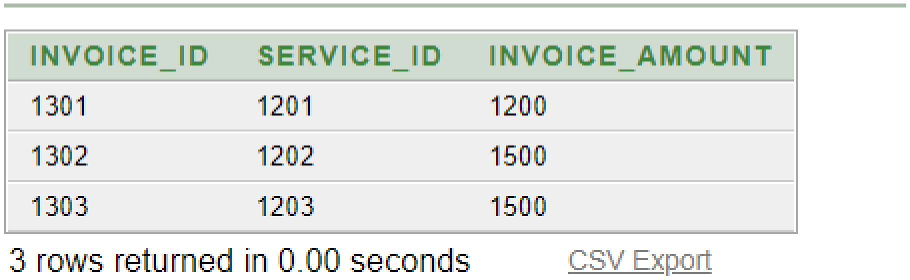
Object Type **TABLE** Object **INVOICE**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
INVOICE	INVOICE_ID	Number	-	10	0	1	-	-	-
	SERVICE_ID	Number	-	10	0	-	-	-	-
	INVOICE_AMOUNT	Number	-	10	0	-	-	-	-

1 - 3

Language: en-us

Application Express 2.1.0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved.



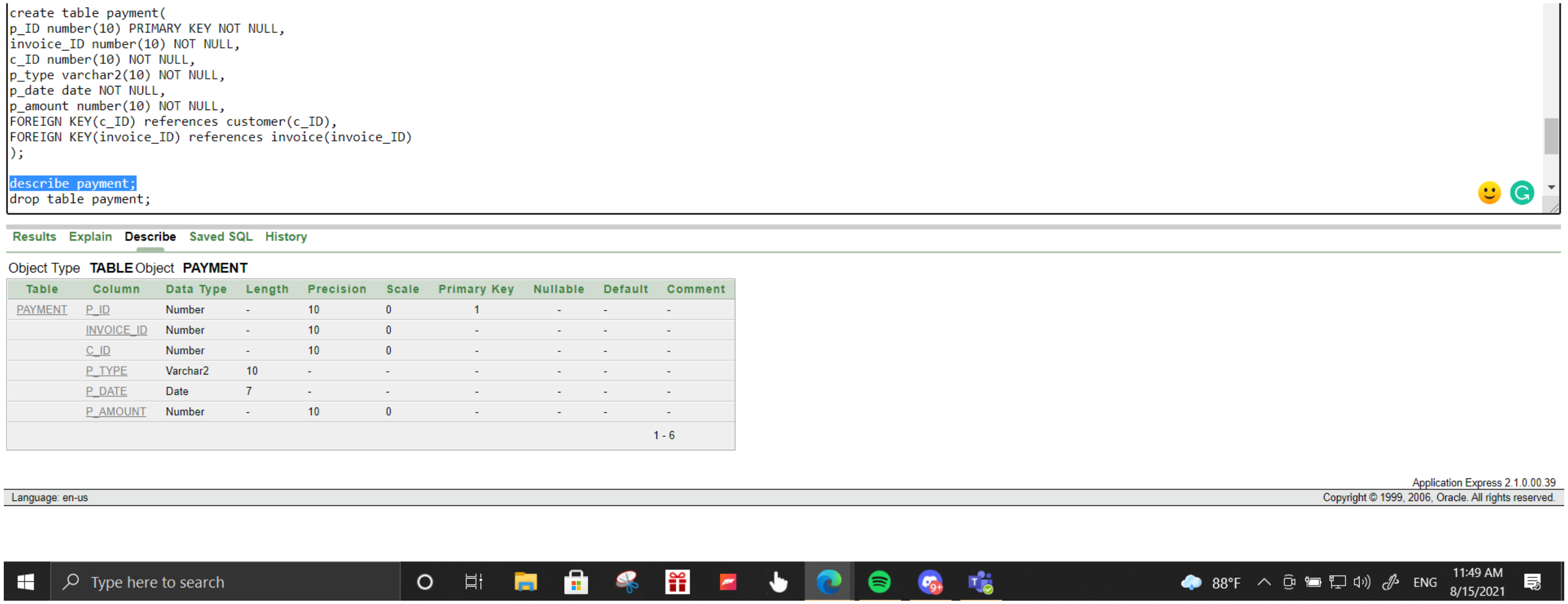
INVOICE_ID	SERVICE_ID	INVOICE_AMOUNT
1301	1201	1200
1302	1202	1500
1303	1203	1500

3 rows returned in 0.00 seconds CSV Export

- 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);
```



```
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)
);

describe payment;
drop table payment;
```

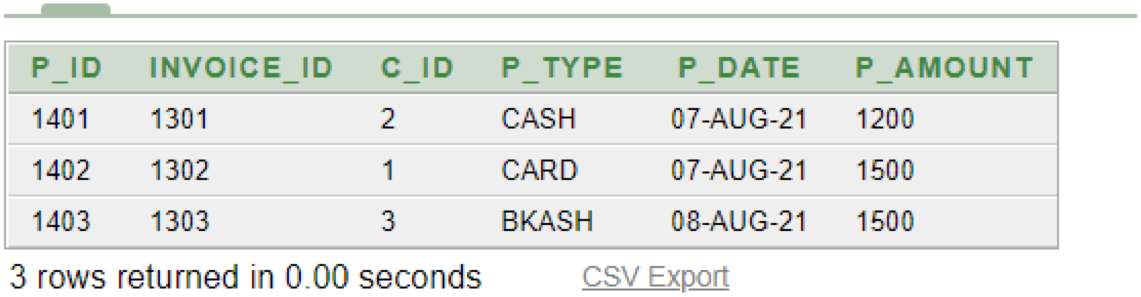
Object Type **TABLE** Object **PAYMENT**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PAYMENT	P_ID	Number	-	10	0	1	-	-	-
	INVOICE_ID	Number	-	10	0	-	-	-	-
	C_ID	Number	-	10	0	-	-	-	-
	P_TYPE	Varchar2	10	-	-	-	-	-	-
	P_DATE	Date	7	-	-	-	-	-	-
	P_AMOUNT	Number	-	10	0	-	-	-	-
								1 - 6	

Language: en-us

Application Express 2.1.0.00.39

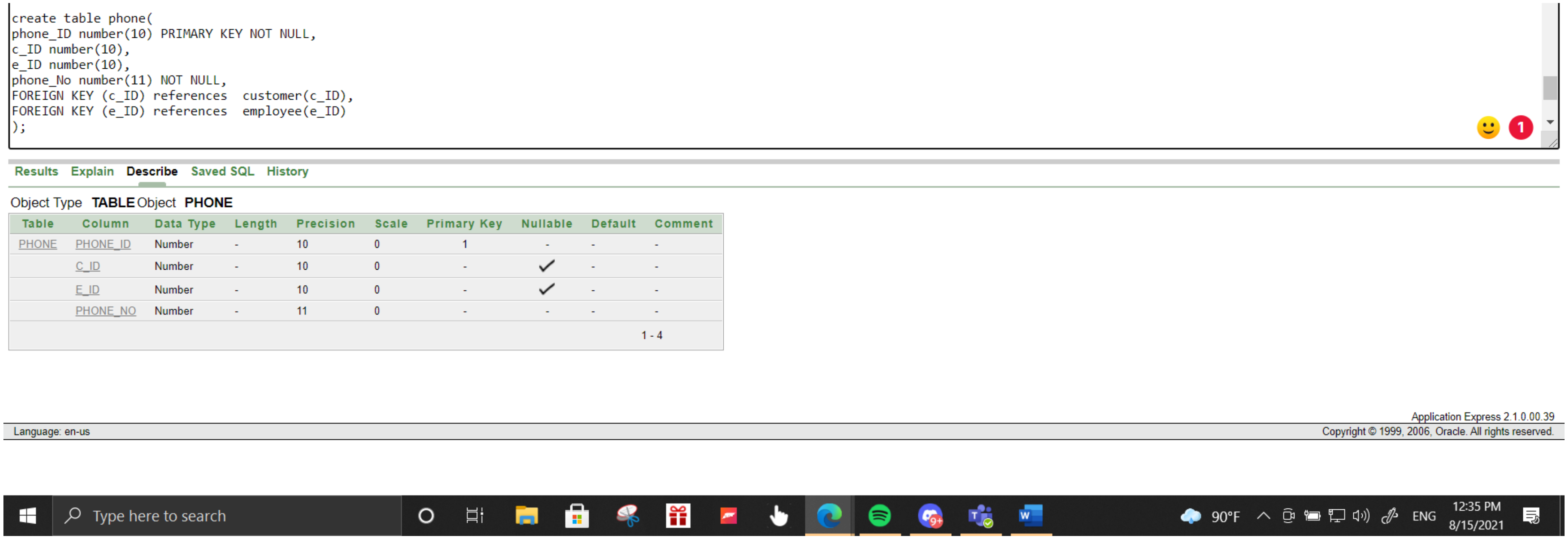
Copyright © 1999, 2006, Oracle. All rights reserved.



P_ID	INVOICE_ID	C_ID	P_TYPE	P_DATE	P_AMOUNT
1401	1301	2	CASH	07-AUG-21	1200
1402	1302	1	CARD	07-AUG-21	1500
1403	1303	3	BKASH	08-AUG-21	1500

3 rows returned in 0.00 seconds   CSV Export

```
➢ 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);
```



```
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)
);
```

Object Type **TABLE** Object **PHONE**

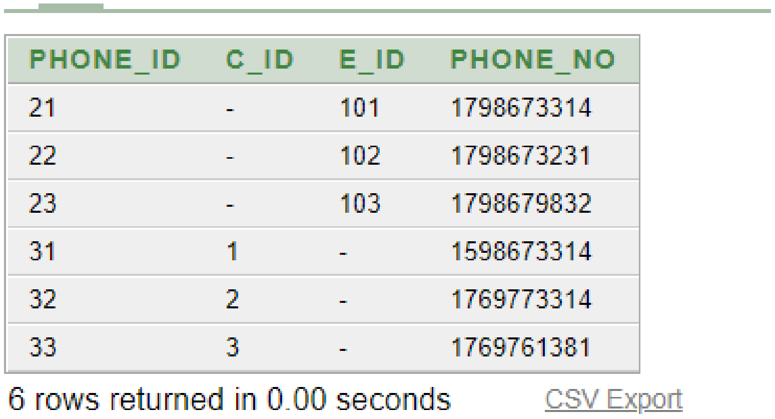
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PHONE	PHONE_ID	Number	-	10	0	1	-	-	-
	C_ID	Number	-	10	0	-	✓	-	-
	E_ID	Number	-	10	0	-	✓	-	-
	PHONE_NO	Number	-	11	0	-	-	-	-

1 - 4

Application Express 2.1.0.00.39

Language: en-us

Copyright © 1999, 2006, Oracle. All rights reserved.



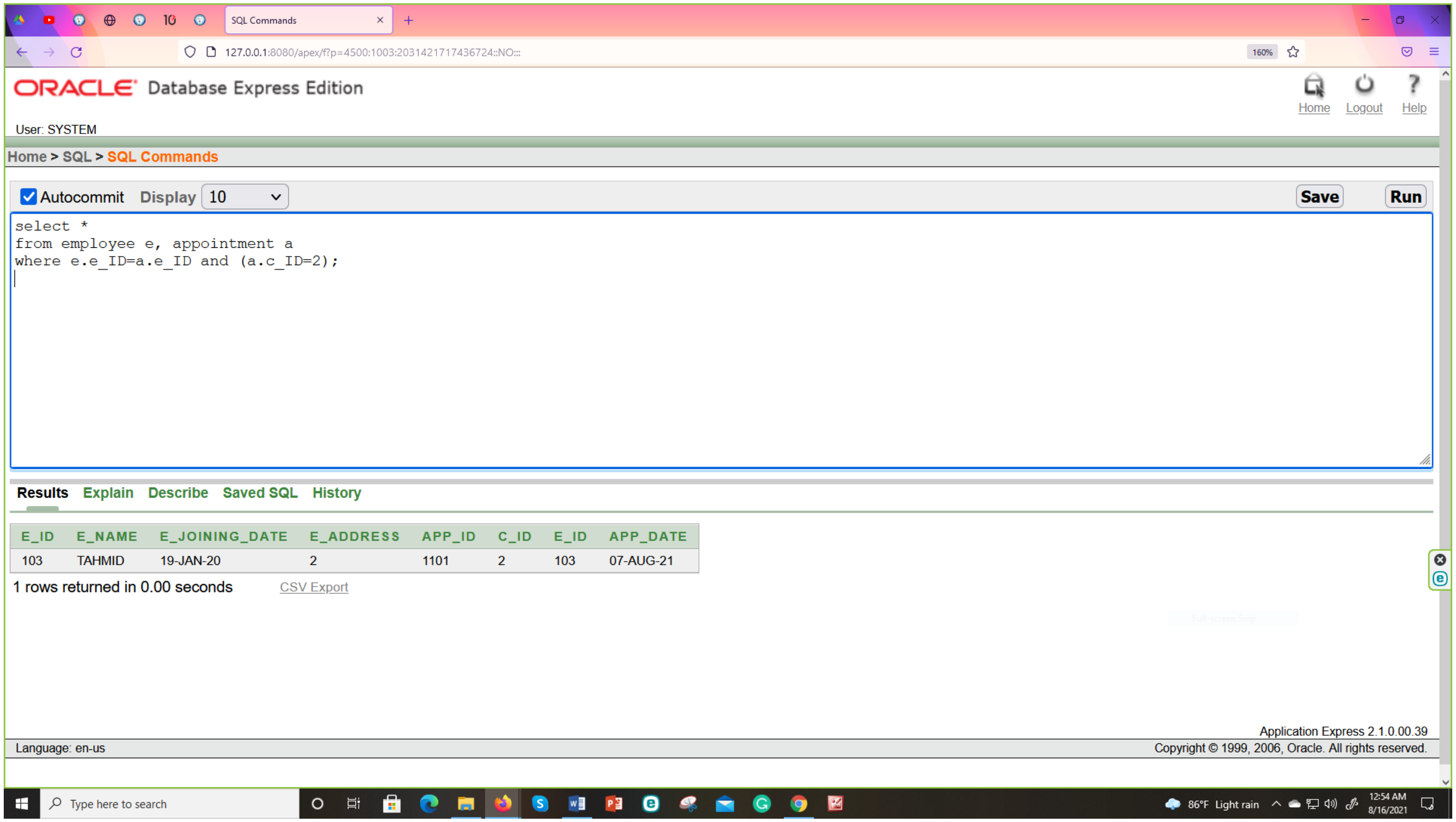
PHONE_ID	C_ID	E_ID	PHONE_NO
21	-	101	1798673314
22	-	102	1798673231
23	-	103	1798679832
31	1	-	1598673314
32	2	-	1769773314
33	3	-	1769761381

6 rows returned in 0.00 seconds

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);


SQL Commands
127.0.0.1:8080/apex/f?p=4500:1003:2031421717436724::NO:::
ORACLE Database Express Edition
Home
Logout
Help
User: SYSTEM
Home > SQL > SQL Commands
☑ Autocommit
Display 10
Save
Run
Results
Explain
Describe
Saved SQL
History
CSV Export
Language: en-us
Application Express 2.1.0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved.
Type here to search

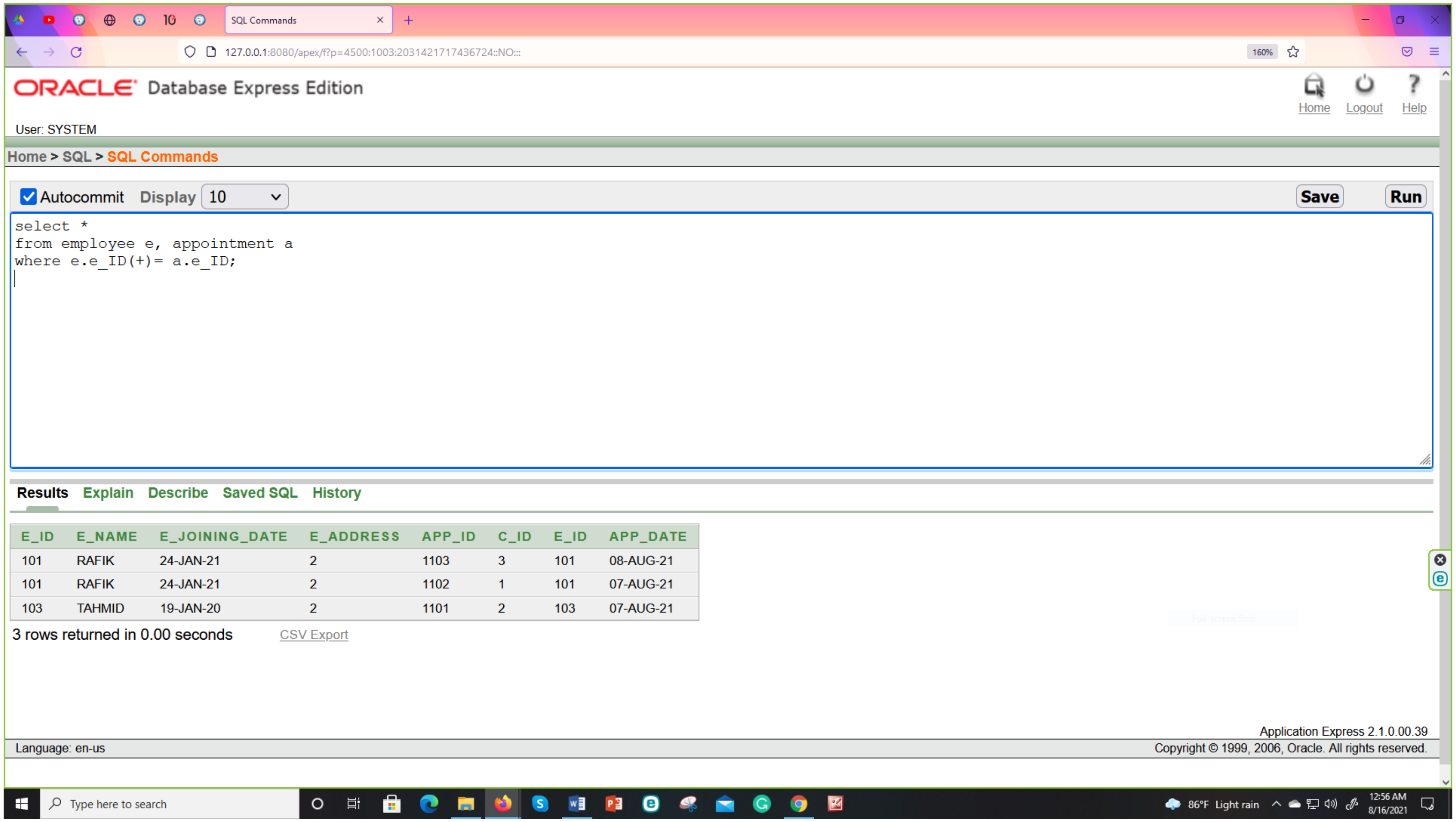
```
select *
from employee e, appointment a
where e.e_ID=a.e_ID and (a.c_ID=2);
```

E_ID	E_NAME	E_JOINING_DATE	E_ADDRESS	APP_ID	C_ID	E_ID	APP_DATE
103	TAHMID	19-JAN-20	2	1101	2	103	07-AUG-21

1 rows returned in 0.00 seconds

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

- ➢ select *
  from employee e, appointment a
  where e.e_ID(+)= a.e_ID;



SQL Commands
127.0.0.1:8080/apex/f?p=4500:1003:2031421717436724::NO:::
ORACLE Database Express Edition
Home
Logout
Help
User: SYSTEM
Home > SQL > SQL Commands
☑ Autocommit
Display 10
Save
Run
Results
Explain
Describe
Saved SQL
History
CSV Export
Language: en-us
Application Express 2.1.0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved.
Type here to search
86°F Light rain
12:56 AM
8/16/2021

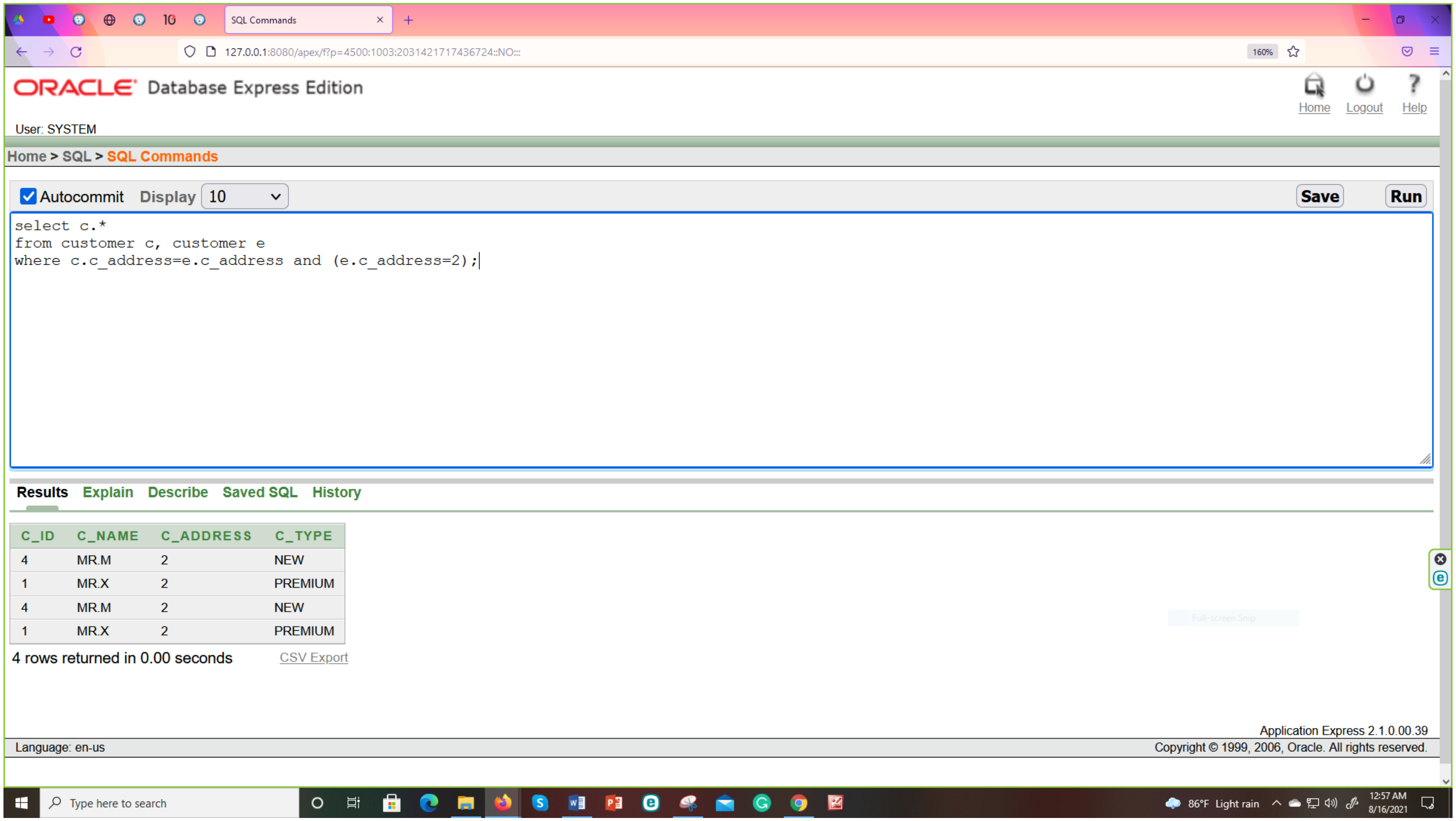
```
select *
from employee e, appointment a
where e.e_ID(+)= a.e_ID;
```

E_ID	E_NAME	E_JOINING_DATE	E_ADDRESS	APP_ID	C_ID	E_ID	APP_DATE
101	RAFIK	24-JAN-21	2	1103	3	101	08-AUG-21
101	RAFIK	24-JAN-21	2	1102	1	101	07-AUG-21
103	TAHMID	19-JAN-20	2	1101	2	103	07-AUG-21

3 rows returned in 0.00 seconds

3.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);



```
select c.*
from customer c, customer e
where c.c_address=e.c_address and (e.c_address=2);
```

C_ID	C_NAME	C_ADDRESS	C_TYPE
4	MR.M	2	NEW
1	MR.X	2	PREMIUM
4	MR.M	2	NEW
1	MR.X	2	PREMIUM

4 rows returned in 0.00 seconds

Application Express 2.1.0.00.39

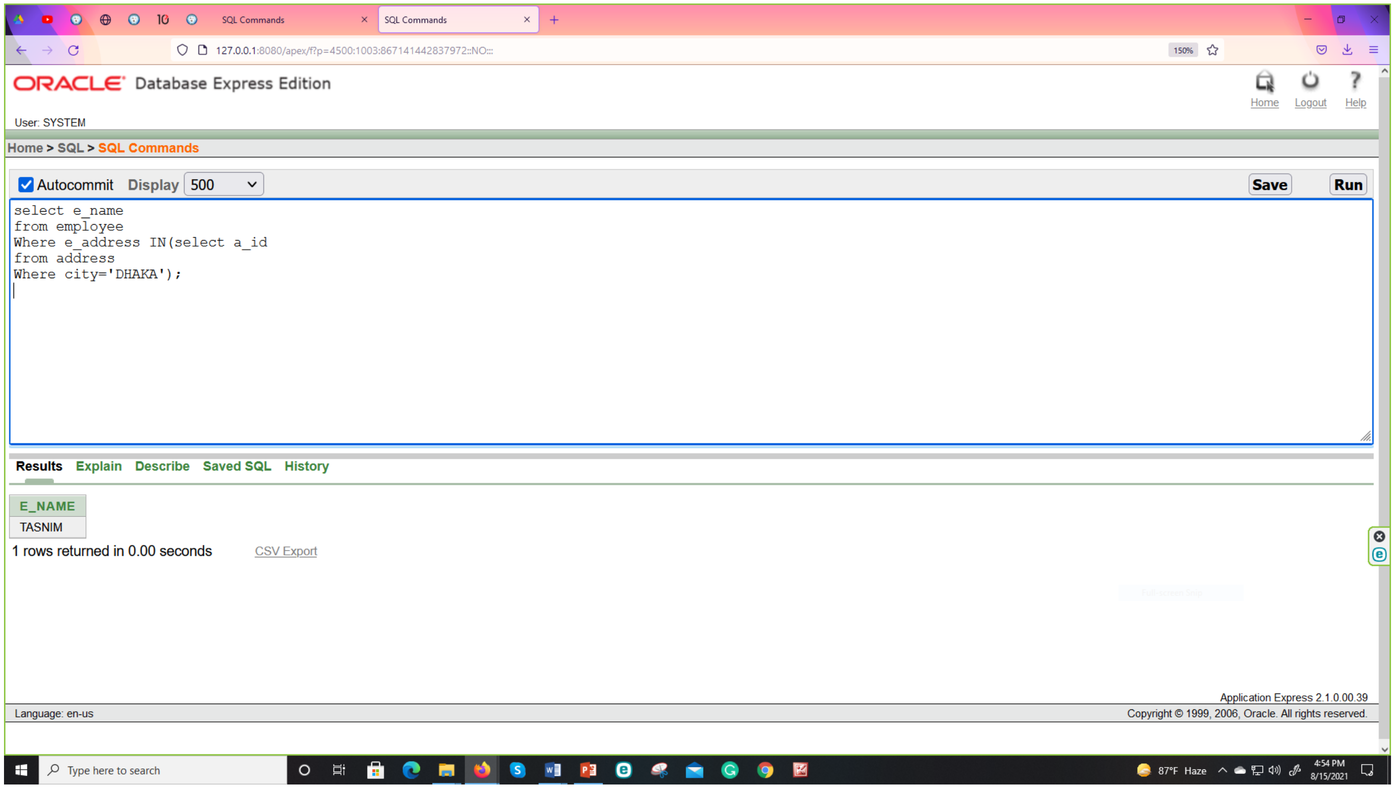
Language: en-us

Copyright © 1999, 2006, Oracle. All rights reserved.

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');



☑ Autocommit Display 500

```
select e_name
from employee
Where e_address IN(select a_id
from address
Where city='DHAKA');
```

E_NAME
TASNIM

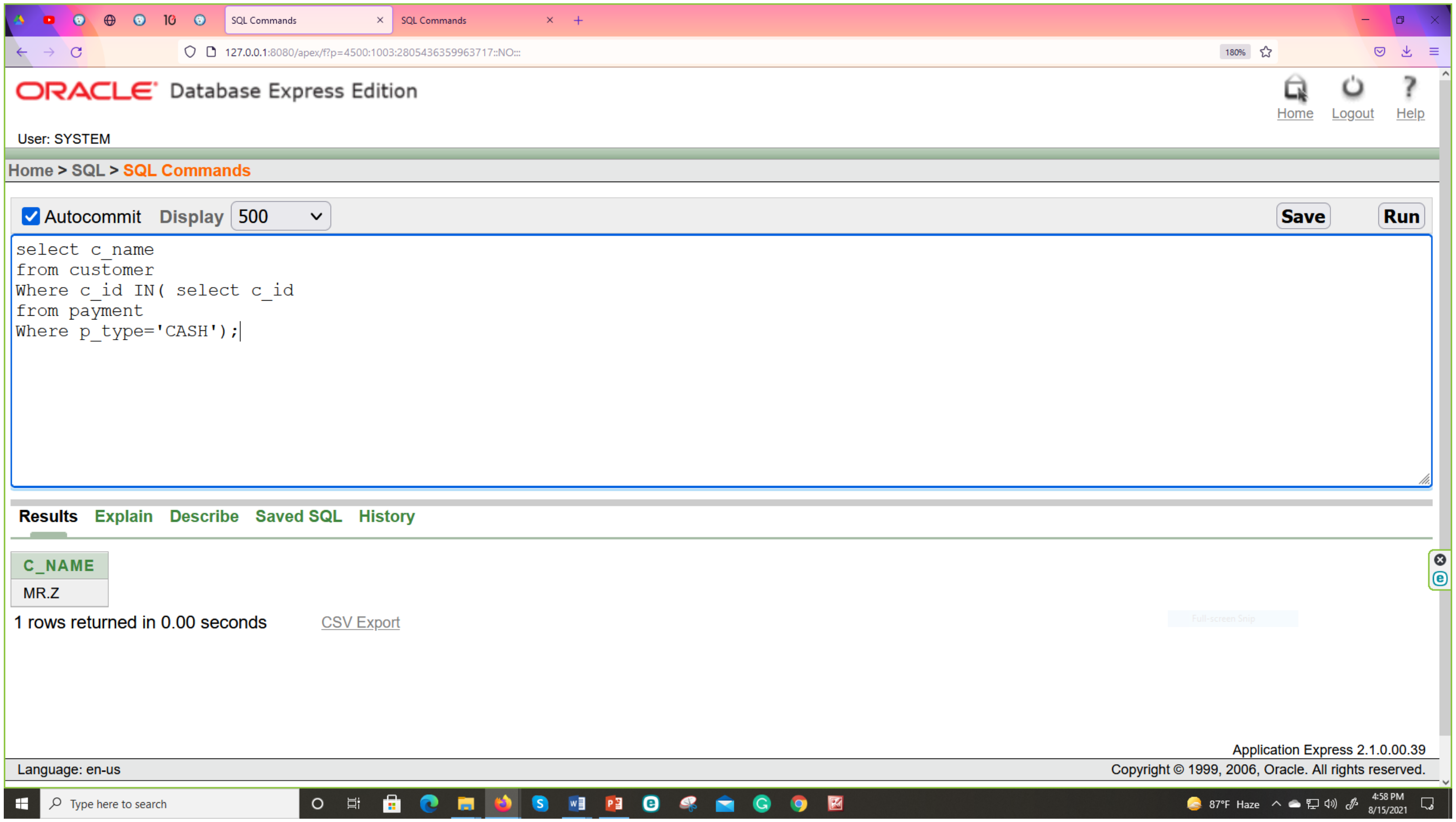
1 rows returned in 0.00 seconds

Language: en-us

Application Express 2.1.0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved.

2. 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');


SQL Commands
SQL Commands
127.0.0.1:8080/apex/f?p=4500:1003:2805436359963717::NO:::
180%
ORACLE Database Express Edition
Home
Logout
Help
User: SYSTEM
Home > SQL > SQL Commands
☑ Autocommit
Display 500
Save
Run
Results
Explain
Describe
Saved SQL
History
CSV Export
Language: en-us
Application Express 2.1.0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved.
Type here to search
87°F Haze
4:58 PM
8/15/2021

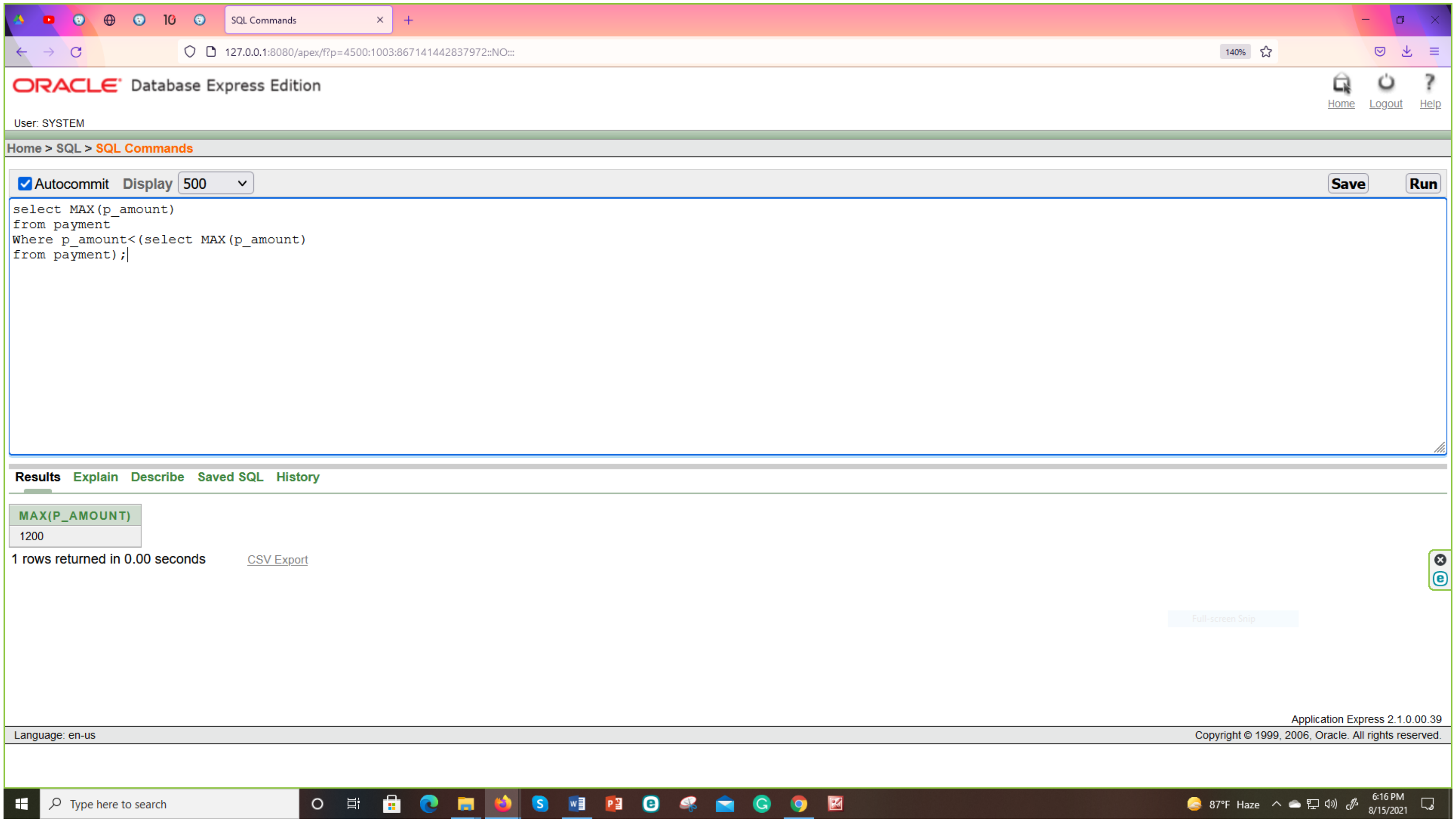
```
select c_name
from customer
Where c_id IN( select c_id
from payment
Where p_type='CASH');
```

C_NAME
MR.Z

1 rows returned in 0.00 seconds

3. Display the second maximum paid amount?

➢ select MAX(p_amount)
  from payment
  Where p_amount<(select MAX(p_amount)
  from payment);



```
select MAX(p_amount)
from payment
Where p_amount<(select MAX(p_amount)
from payment);
```

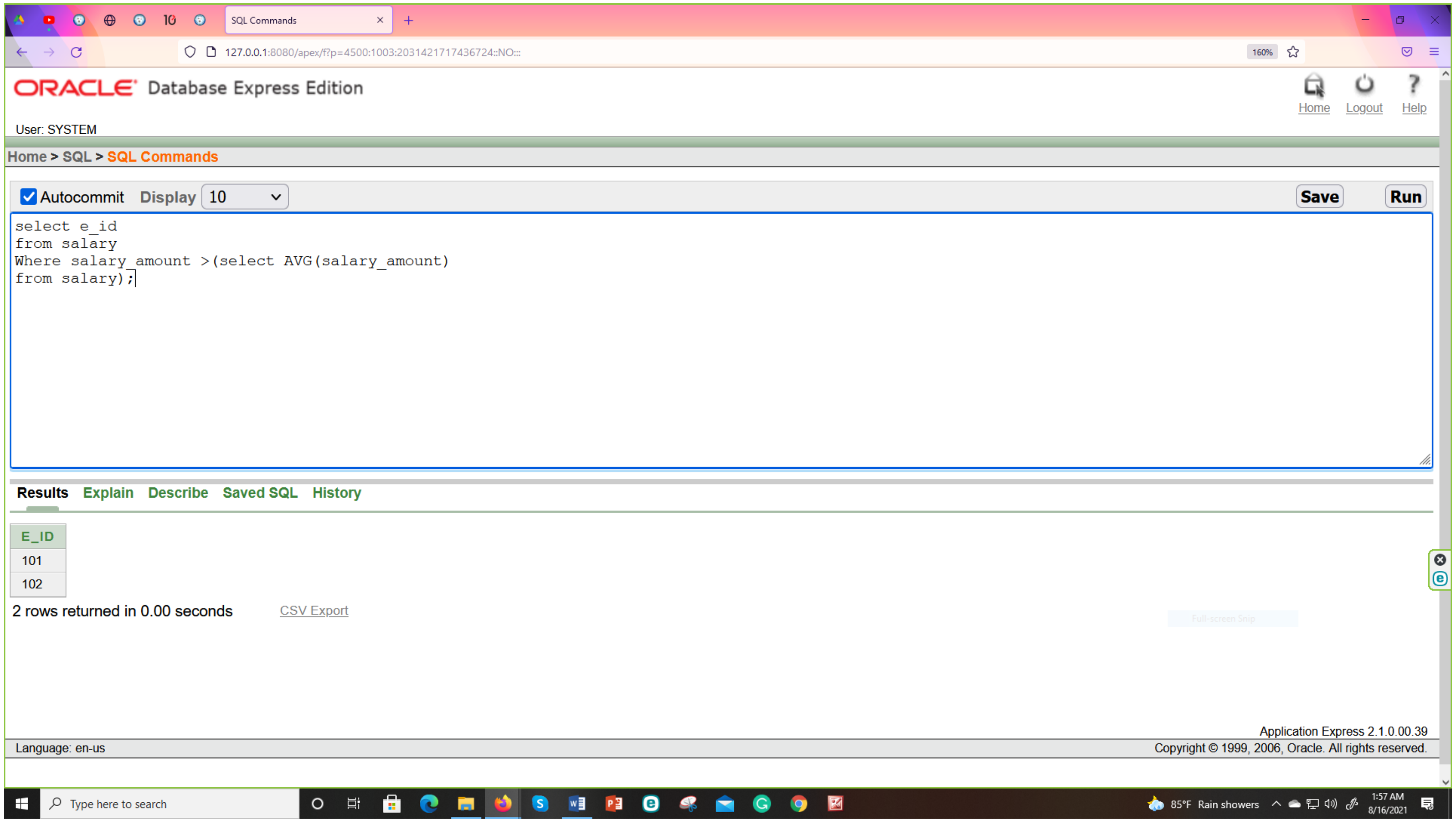
MAX(P_AMOUNT)
1200

1 rows returned in 0.00 seconds

SQL Commands
127.0.0.1:8080/apex/f?p=4500:1003:867141442837972::NO:::
ORACLE Database Express Edition
User: SYSTEM
Home
Logout
Help
Home > SQL > SQL Commands
☑ Autocommit
Display 500
Save
Run
Results
Explain
Describe
Saved SQL
History
CSV Export
Application Express 2.1.0.00.39
Language: en-us
Copyright © 1999, 2006, Oracle. All rights reserved.
Type here to search
87°F Haze
6:16 PM
8/15/2021

4. 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);


ORACLE Database Express Edition
Home
Logout
Help
User: SYSTEM
Home > SQL > SQL Commands
☑ Autocommit
Display 10
Save
Run
Results
Explain
Describe
Saved SQL
History
CSV Export
Language: en-us
Application Express 2.1.0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved.

```
select e_id
from salary
Where salary_amount >(select AVG(salary_amount)
from salary);
```

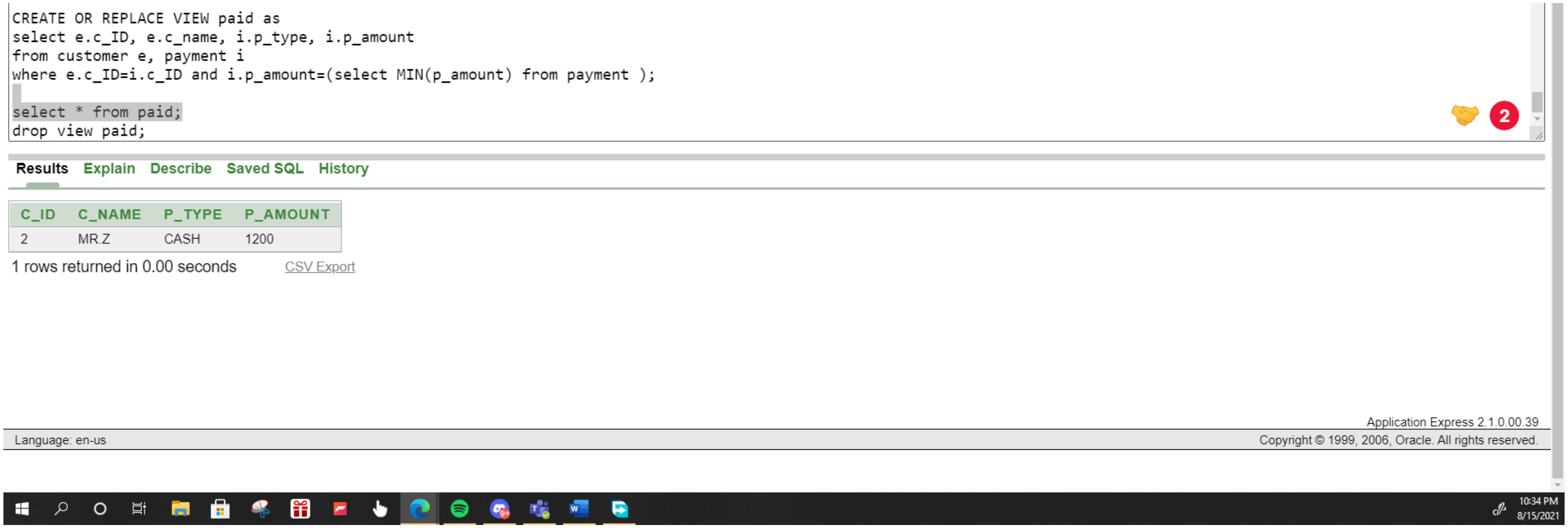
E_ID
101
102

2 rows returned in 0.00 seconds

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;



```
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;
```

Results
Explain
Describe
Saved SQL
History

C_ID	C_NAME	P_TYPE	P_AMOUNT
2	MR.Z	CASH	1200

1 rows returned in 0.00 seconds

CSV Export
Application Express 2.1.0.00.39
Language: en-us
Copyright © 1999, 2006, Oracle. All rights reserved.
10:34 PM
8/15/2021