# 

**American International University Bangladesh**

**DataBase Project**

**“Apartment Management System”**

Submitted to

**Juena Ahmed Noshin**

[**juena@aiub.edu**](mailto:juena@aiub.edu)

Submitted by

|  |  |  |
| --- | --- | --- |
| NAME | ID | SECTION |
| 1.Nusrat, Jannatul | **19-40885-2** | **H** |
| 2.Sajib, Nahid Hasan | **19-40866-2** | **H** |
| 3.Sezan,Tanjin Rashid | **19-40887-2** | **H** |
| 4.Linkon, Ankon Sarkar | **19-40895-2** | **H** |
| 5.Rabby, H.M Jayed Chowdhury | **19-40893-2** | **H** |
| 6.Badhan, Walid Bin Wahid | **19-40845-2** | **H** |

**Course: Introduction to Database**

# **“Apartment Management System”**

# **CONTENTS:**

|  |  |
| --- | --- |
| Introduction | 03 |
| Scenario Description | 04 |
| ER Diagram | 05 |
| Normalization | 06-11 |
| Schema Diagram | 12 |
| Table Creation | 13-21 |
| Data Insertion | 22-29 |
| Query Writing | 30-35 |
| Relational Algebra | 36 |
| Conclusion | 37-38 |

# **Introduction:**

For our project, we have decided to make a model of “**Apartment Management System**”. The AMS requires a tool to be built for a local building management company wishing to automate many of the interactions between apartment members and apartment manager. The system needs to keep the record of every contact detail of the apartment manager and the apartment users. In addition to just handling rent money exchange, this system stores the details of apartment buildings, apartments, apartment facilities and so on.

The project proved to be a large undertaking as we spent a significant amount of time delving into the details of what the management system, an apartment building requires and all of the rent laws & other facilities.

We had team members who worked on storing contact details of both apartment manager & apartment member, apartment details as well as the building details, rent interactions. The following design document reflects all of those features and more.

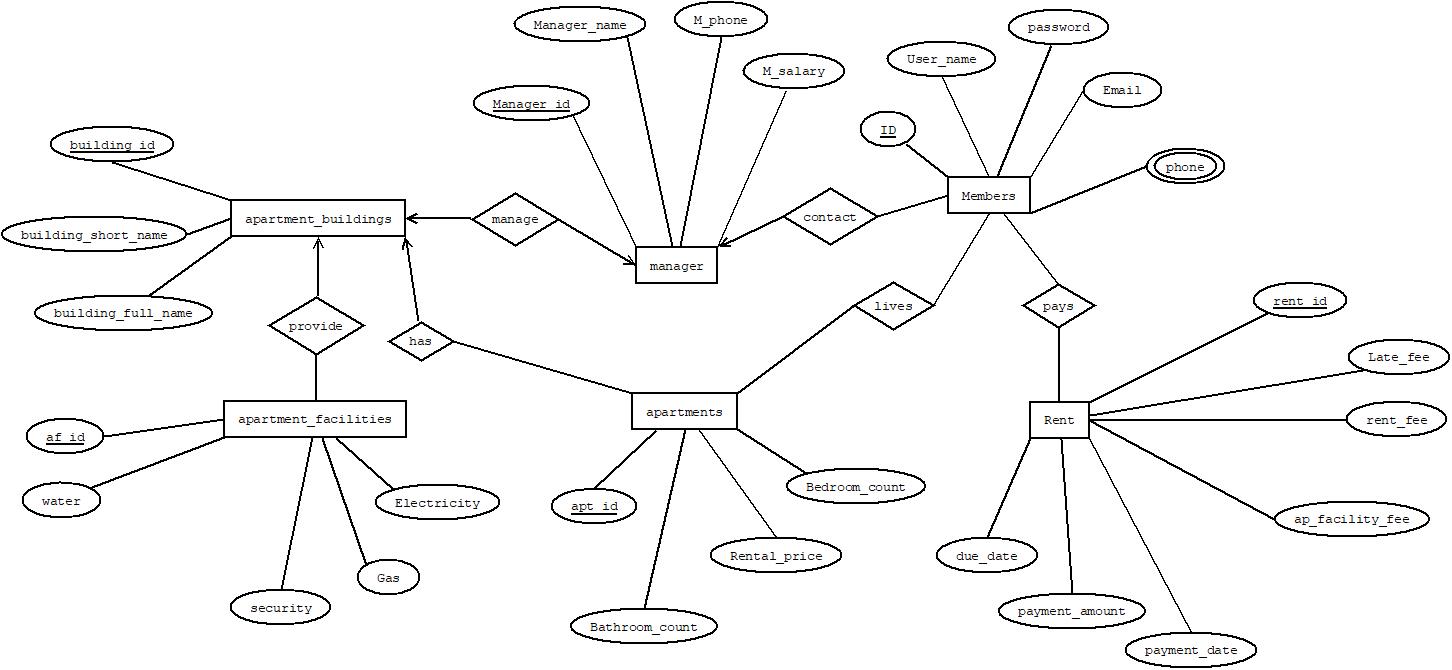
For the group members, this project proved to be quite the learning experience. We hope the following can accurately portray a sample of what such a software suite would require and how it could be coded to become a reality.

The overall goals of the system are to keep track of manager & members record, user management requests, document and rent collection, apartment facilities.

# **Scenario Description:**

In an “**Apartment Management System”** apartments are managed by a manager. The system stores manager id, name, phone and salary. A member can rent one or more than one apartment in the same building. Every member is identified by a user name. The system also stores member id, phone, password and email. In an apartment management system, apartment building has a unique id, building’s full name & also building’s short name. Apartment building has apartments. In an **AMS** details ofapartments’ such as id, rental price, bedroom count and bathroom count are also stored. The system also stores a list of apartment facilities (gas, electricity, water, security) with an unique id of apartment facilities. In addition to all these, this system also keeps a record of rent details (rent id, due date, rent fee, apartment facilities fee, late fee, payment date, payment amount) paid by the members.

**ER Diagram:**



**Normalization:**

**Provide**

**UNF**

Provide(building\_id,building\_short\_name,building\_full\_name,af\_id,water,security,gas,electricity)

**1NF**

1. There is no multi valued attribute.Relation already in1NF. building\_id,building\_short\_name,building\_full\_name,af\_id,water,security,gas,el ectricity.

**2NF**

1. building\_id,building\_short\_name,building\_full\_name
2. af\_id,water,security,gas,electricity

**3NF**

There is no transitive dependency

1. building\_id,building\_short\_name,building\_full\_name
2. af\_id,water,security,gas,electricity,building\_id.

**Has**

**UNF**

Has(building\_id,building\_short\_name,building\_full\_name,apt\_id,bathroom\_count,rental\_price,bedroom\_count)

**1NF**

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

1. building\_id,building\_short\_name,building\_full\_name,apt\_id,bathroom\_count,rental\_price,bedroom\_count.

**2NF**

1. building\_id,building\_short\_name,building\_full\_name.
2. apt\_id,bathroom\_count,rental\_price,bedroom\_count.

**3NF**

There is no transitive dependency.

1. building\_id,building\_short\_name,building\_full\_name.
2. apt\_id,bathroom\_count,rental\_price,bedroom\_count,building\_id.

**Manage**

**UNF**

Manage(building\_id,building\_short\_name,building\_full\_name,manager\_id,manager\_name,M\_phone,Msalary)

**1NF**

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

1. building\_id,building\_short\_name,building\_full\_name,manager\_id,manager\_name,M\_phone,M\_salary.

**2NF**

1. building\_id,building\_short\_name,building\_full\_name.
2. manager\_id,manager\_name,M\_phone,M\_salary.

**3NF**

There is no transitive dependency

1. building\_id,building\_short\_name,building\_full\_name.
2. manager\_id,manager\_name,M\_phone,M\_salary,building\_id.

**Contact**

**UNF**

Contact(manager\_id,manager\_name,M\_phone,M\_salary,ID,user\_name,password,email,phone)

**1NF**

Phone is a multi valued attribute.

1. manager\_id,manager\_name,M\_phone,M\_salary,ID,user\_name,password,email,phone.

**2NF**

1. manager\_id,manager\_name,M\_phone,M\_salary.
2. ID,user\_name,password,email,phone.

**3NF**

There is no transitive dependency.

1. manager\_id,manager\_name,M\_phone,M\_salary.
2. ID,user\_name,password,email,phone,manager\_id.

**Lives**

**UNF**

Lives(apt\_id,bathroom\_count,rental\_price,bedroom\_count, ID,user\_name,password,email,phone)

**1NF**

Phone is a multi valued attribute.

1. apt\_id,bathroom\_count,rental\_price,bedroom\_count, ID,user\_name,password,email,phone.

**2NF**

1. apt\_id,bathroom\_count,rental\_price,bedroom\_count.
2. ID,user\_name,password,email,phone.

**3NF**

There is no transitive dependency.

1. apt\_id,bathroom\_count,rental\_price,bedroom\_count.
2. ID,user\_name,password,email,phone.
3. apt\_id ,ID.

**Pays**

**UNF**

Pays(ID,user\_name,password,email,phone,rent\_id,late\_fee,rent\_fee,ap\_facility\_fee,payment\_date,payment\_amount,due\_date)

**1NF**

Phone is a multi valued attribute.

1. ID,user\_name,password,email,phone,rent\_id,late\_fee,rent\_fee,ap\_facility\_fee,payment\_date,payment\_amount,due\_date.

**2NF**

1. ID,user\_name,password,email,phone.
2. rent\_id,late\_fee,rent\_fee,ap\_facility\_fee,payment\_date,

payment\_amount,due\_date.

**3NF**

There is no transitive dependency.

1. ID,user\_name,password,email,phone.
2. rent\_id,late\_fee,rent\_fee,ap\_facility\_fee,payment\_date,

payment\_amount,due\_date.

1. ID,rent\_id.

**Temporary Tables:**

1. building\_id,building\_short\_name,building\_full\_name.
2. af\_id,water,security,gas,electricity,**building\_id**.
3. **~~building\_id~~**~~,building\_short\_name,building\_full\_name.~~
4. apt\_id,bathroom\_count,rental\_price,bedroom\_count,**building\_id**.
5. **~~building\_id~~**~~,building\_short\_name,building\_full\_name.~~
6. manager\_id,manager\_name,M\_phone,M\_salary,**building\_id**.
7. **~~manager\_id~~**~~,manager\_name,M\_phone,M\_salary.~~
8. ID,user\_name,password,email,phone,**manager\_id**.
9. ~~Apt\_id,bathroom\_count,rental\_price,bedroom\_count.~~
10. ~~ID,user\_name,password,email,phone.~~
11. **apt\_id ,ID**.
12. ~~ID,user\_name,password,email,phone.~~
13. rent\_id,late\_fee,rent\_fee,ap\_facility\_fee,payment\_date,

payment\_amount,due\_date.

1. **ID,rent\_id.**

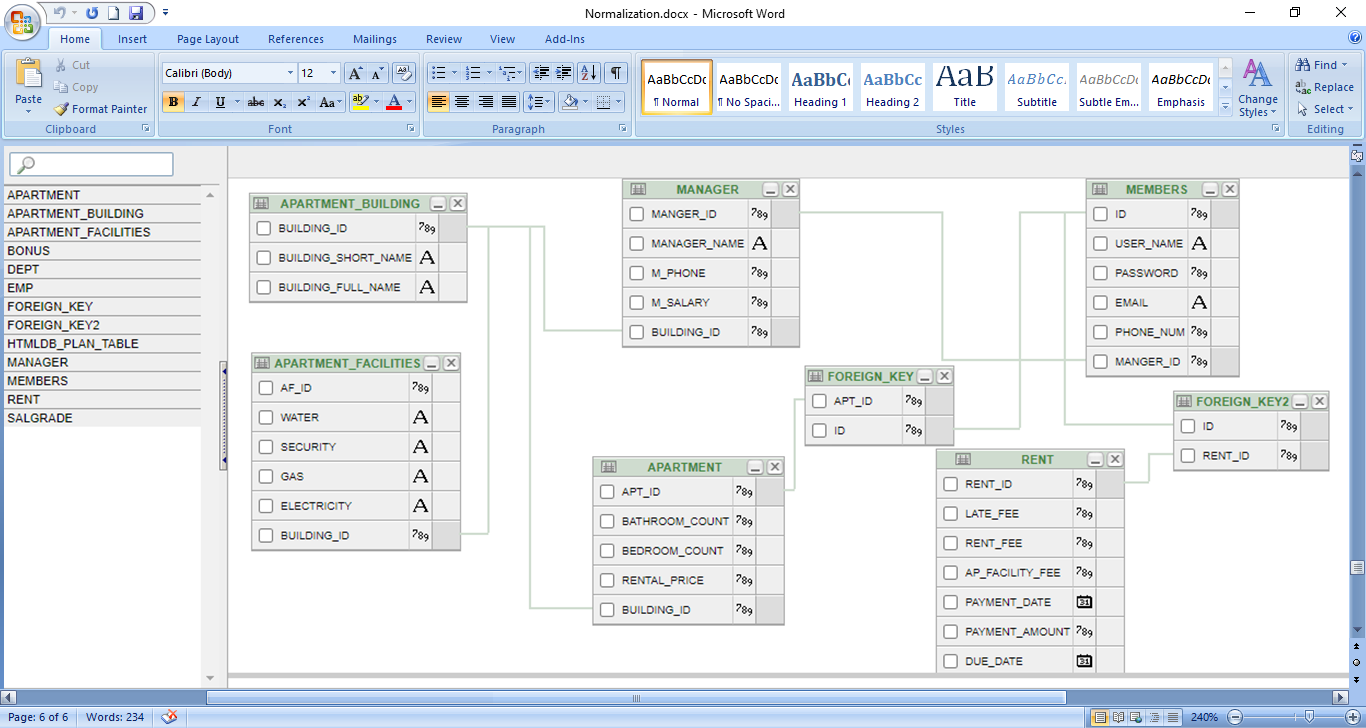
**Final Tables:**

1. **building\_id**,building\_short\_name,building\_full\_name.
2. af\_id,water,security,gas,electricity,**building\_id**.
3. apt\_id,bathroom\_count,rental\_price,bedroom\_count,**building\_id**.
4. manager\_id,manager\_name,M\_phone,M\_salary,**building\_id**.
5. ID,user\_name,password,email,phone,**manager\_id**.
6. **apt\_id** ,**ID**.
7. rent\_id,late\_fee,rent\_fee,ap\_facility\_fee,payment\_date,

payment\_amount,due\_date.

1. **ID**,**rent\_id**.

**Schema Diagram:**

****

**Table Creation:**

* **After normalization tables are (including constraints):**

1. APARTMENT\_BUILDING ( BUILDING\_ID, BUILDING\_SHORT\_NAME, BUILDING\_FULL\_NAME)

2.APARTMENT(APT\_ID,BATHROOM\_COUNT,BEDROOM\_COUNT,RENTAL\_PRICE,**BUILDING\_ID**)

3. APARTMENT\_FACILITIES (AF\_ID, WATER,SECURITY, GAS, ELECTRICITY, **BUILDING\_ID**)

4. MANAGER (MANAGER\_ID, MANAGER\_NAME, M\_PHONE, M\_SALARY, **BUILDING\_ID**)

5. MEMBERS(ID, USER\_NAME, PASSWORD, EMAIL, PHONE\_NUM, **MANAGER\_ID)**

6. FOREIGN\_KEY (**APT\_ID**, **ID** )

7. RENT (RENT\_ID, LATE\_FEE, RENT\_FEE, AP\_FACILITY\_FEE, PAYMENT\_DATE, PAYMENT\_AMOUNT, DUE\_DATE)

8. FOREIGN\_KEY2 (**ID, RENT\_ID** )

**Using DESC command:**

Table APARTMENT\_BUILDING:

CREATE TABLE APARTMENT\_BUILDING(BUILDING\_ID NUMBER(20)PRIMARY KEY,BUILDING\_SHORT\_NAME VARCHAR2(20),BUILDING\_FULL\_NAME VARCHAR2(20));

CREATE SEQUENCE BUILDING\_BUILDING\_ID INCREMENT BY 1 START WITH 105 MAXVALUE 500 NOCACHE NOCYCLE;

desc APARTMENT\_BUILDING;

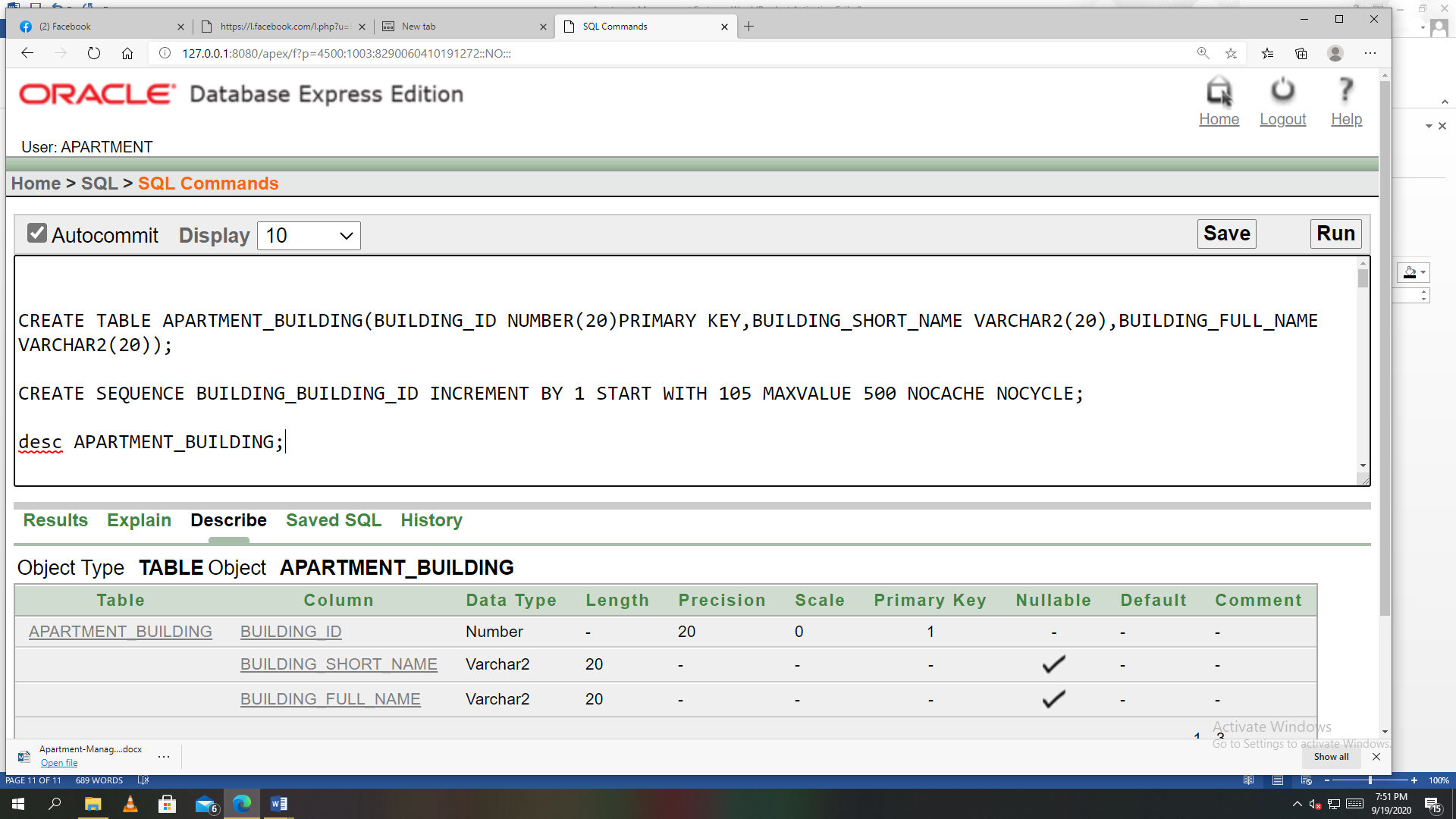


Table APARTMENT:

CREATE TABLE APARTMENT(APT\_ID NUMBER(20)PRIMARY KEY,BATHROOM\_COUNT NUMBER(20),BEDROOM\_COUNT NUMBER(20),RENTAL\_PRICE NUMBER(30),BUILDING\_ID NUMBER(20));

# ALTER TABLE APARTMENT ADD CONSTRAINT FK3 FOREIGN KEY(BUILDING\_ID) REFERENCES APARTMENT\_BUILDING(BUILDING\_ID);

# CREATE SEQUENCE APARTMENT\_APT\_ID INCREMENT BY 1 START WITH 1005 MAXVALUE 5000 NOCACHE NOCYCLE;

# desc APARTMENT;

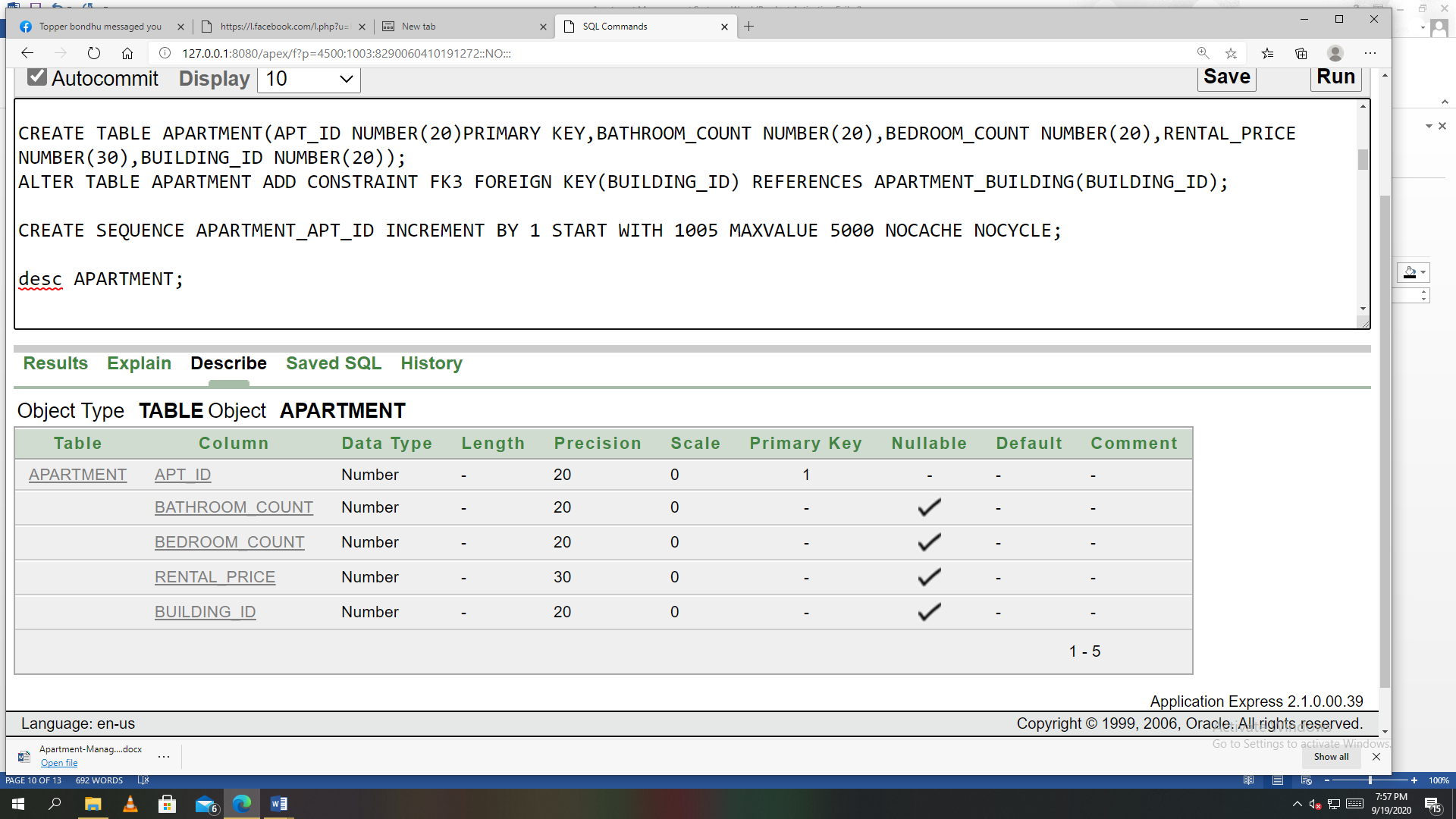


Table APARTMENT\_FACILITIES:

CREATE TABLE APARTMENT\_FACILITIES(AF\_ID NUMBER(20)PRIMARY KEY,WATER VARCHAR2(20),SECURITY VARCHAR2(20),GAS VARCHAR2(20),ELECTRICITY VARCHAR2(20),BUILDING\_ID NUMBER(20));

ALTER TABLE APARTMENT\_FACILITIES ADD CONSTRAINT FK4 FOREIGN KEY(BUILDING\_ID) REFERENCES APARTMENT\_BUILDING(BUILDING\_ID);

CREATE SEQUENCE APARTMENT\_FACILITIES\_AF\_ID INCREMENT BY 1 START WITH 5 MAXVALUE 100 NOCACHE NOCYCLE;

desc APARTMENT\_FACILITIES;

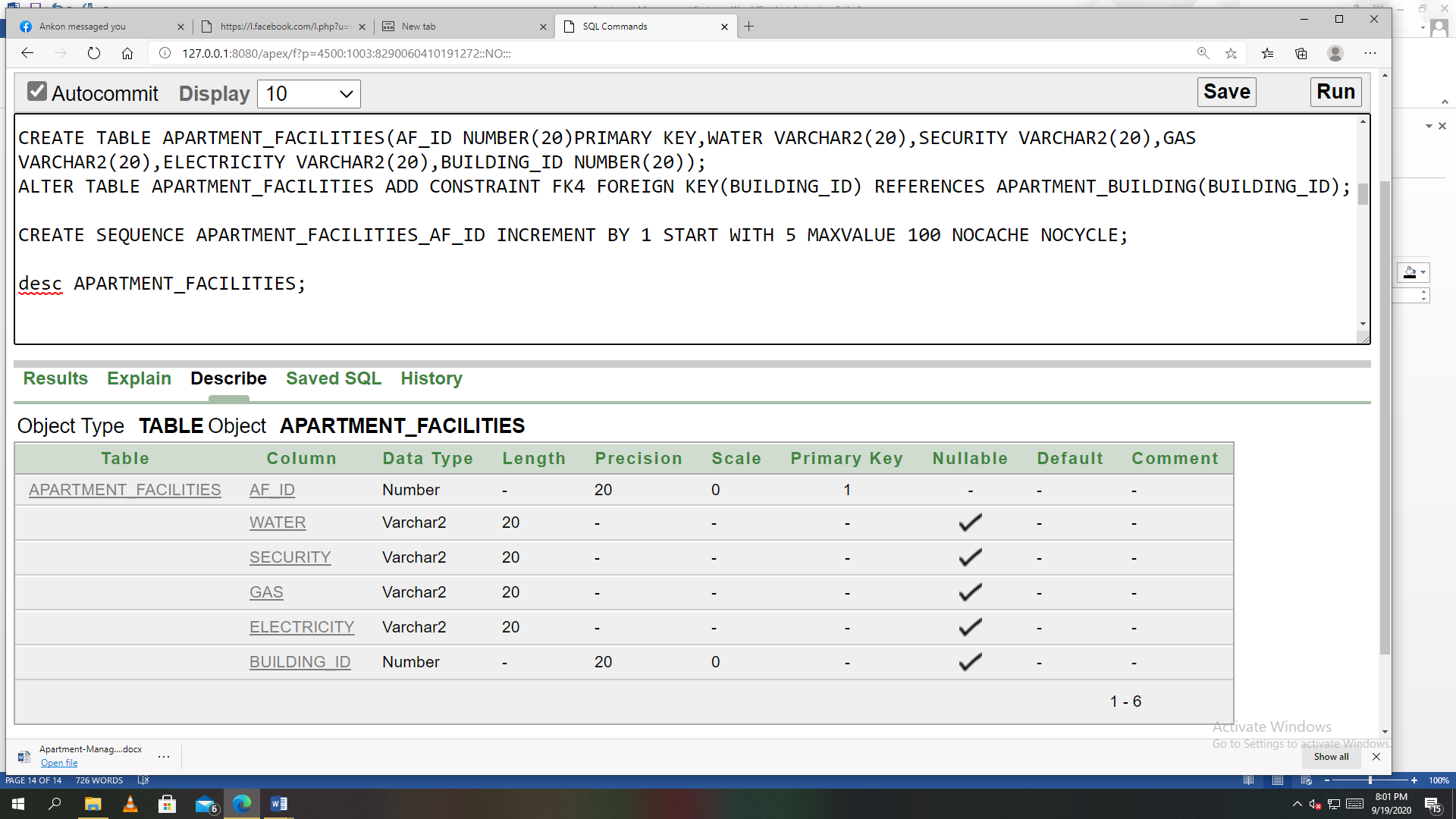


Table MANAGER:

CREATE TABLE MANAGER (MANAGER\_ID NUMBER(20)PRIMARY KEY,MANAGER\_NAME VARCHAR2(30),M\_PHONE NUMBER(20),M\_SALARY NUMBER(20),BUILDING\_ID NUMBER(20));

ALTER TABLE MANAGER ADD CONSTRAINT FK5 FOREIGN KEY(BUILDING\_ID) REFERENCES APARTMENT\_BUILDING(BUILDING\_ID);

CREATE SEQUENCE MANAGER\_MANAGER\_ID INCREMENT BY 1 START WITH 115 MAXVALUE 500 NOCACHE NOCYCLE;

desc MANAGER;

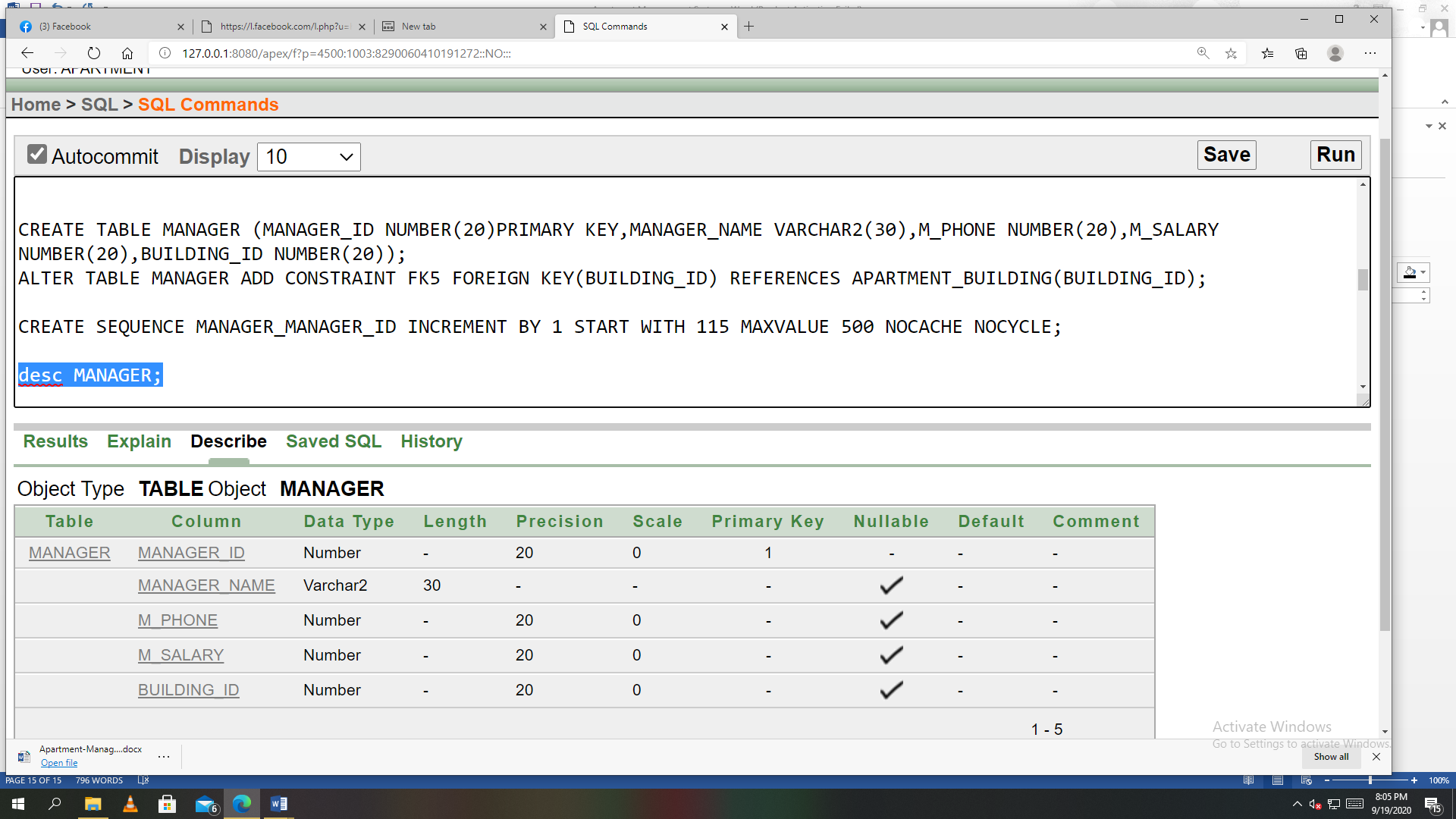


Table MEMBERS:

CREATE TABLE MEMBERS(ID NUMBER(20)PRIMARY KEY,USER\_NAME VARCHAR2(20),PASSWORD NUMBER(20),EMAIL VARCHAR2(20),PHONE\_NUM NUMBER(20),MANAGER\_ID NUMBER(20));

ALTER TABLE MEMBERS ADD CONSTRAINT FK6 FOREIGN KEY(MANAGER\_ID) REFERENCES MANAGER(MANAGER\_ID);

CREATE SEQUENCE MEMBERS\_ID INCREMENT BY 1 START WITH 4005 MAXVALUE 5000 NOCACHE NOCYCLE;

desc MEMBERS;

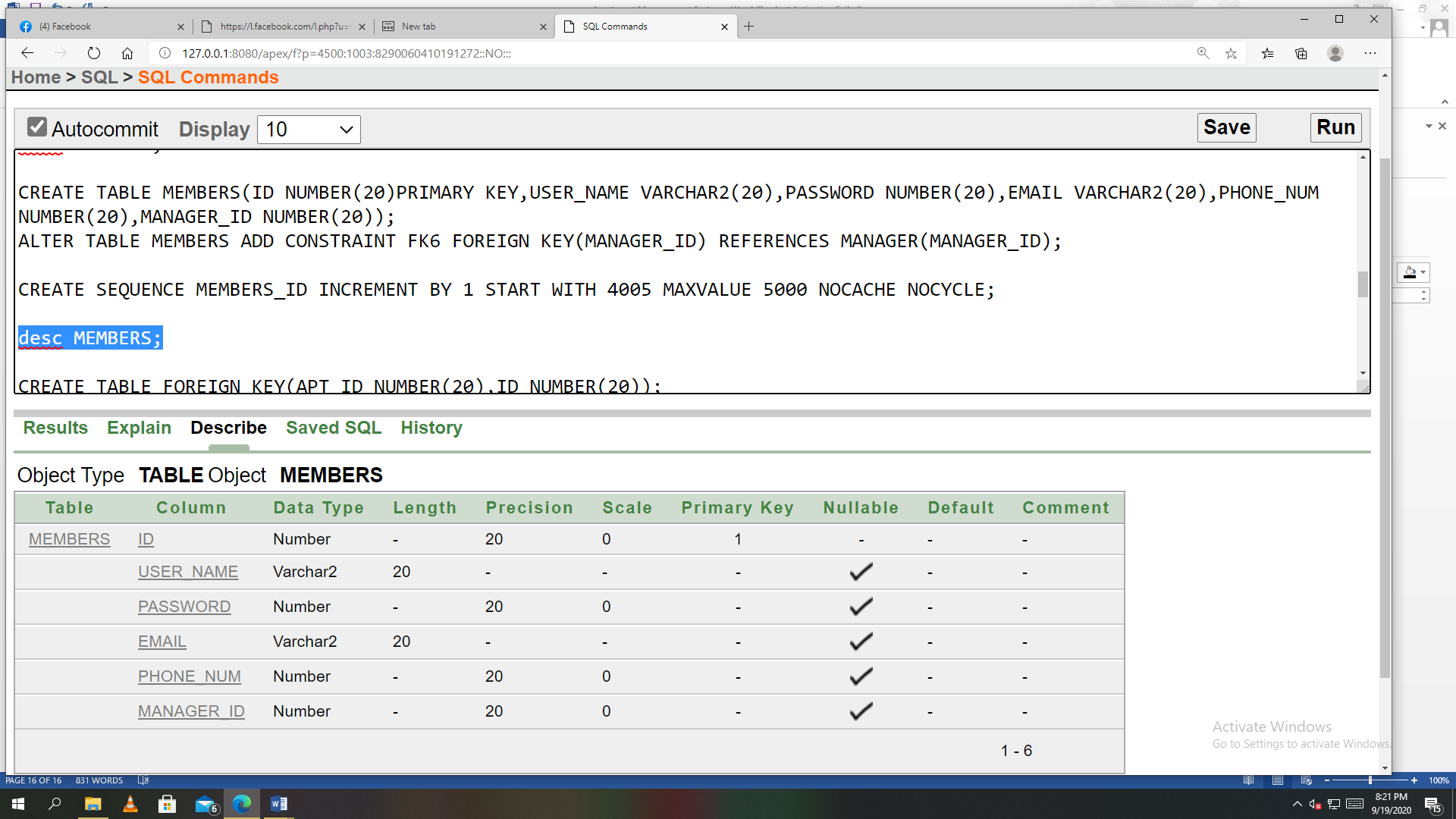


Table FOREIGN\_KEY:

CREATE TABLE FOREIGN\_KEY(APT\_ID NUMBER(20),ID NUMBER(20));

ALTER TABLE FOREIGN\_KEY ADD CONSTRAINT FK7 FOREIGN KEY(APT\_ID) REFERENCES APARTMENT(APT\_ID);

ALTER TABLE FOREIGN\_KEY ADD CONSTRAINT FK8 FOREIGN KEY(ID) REFERENCES MEMBERS(ID);

DESC FOREIGN\_KEY;

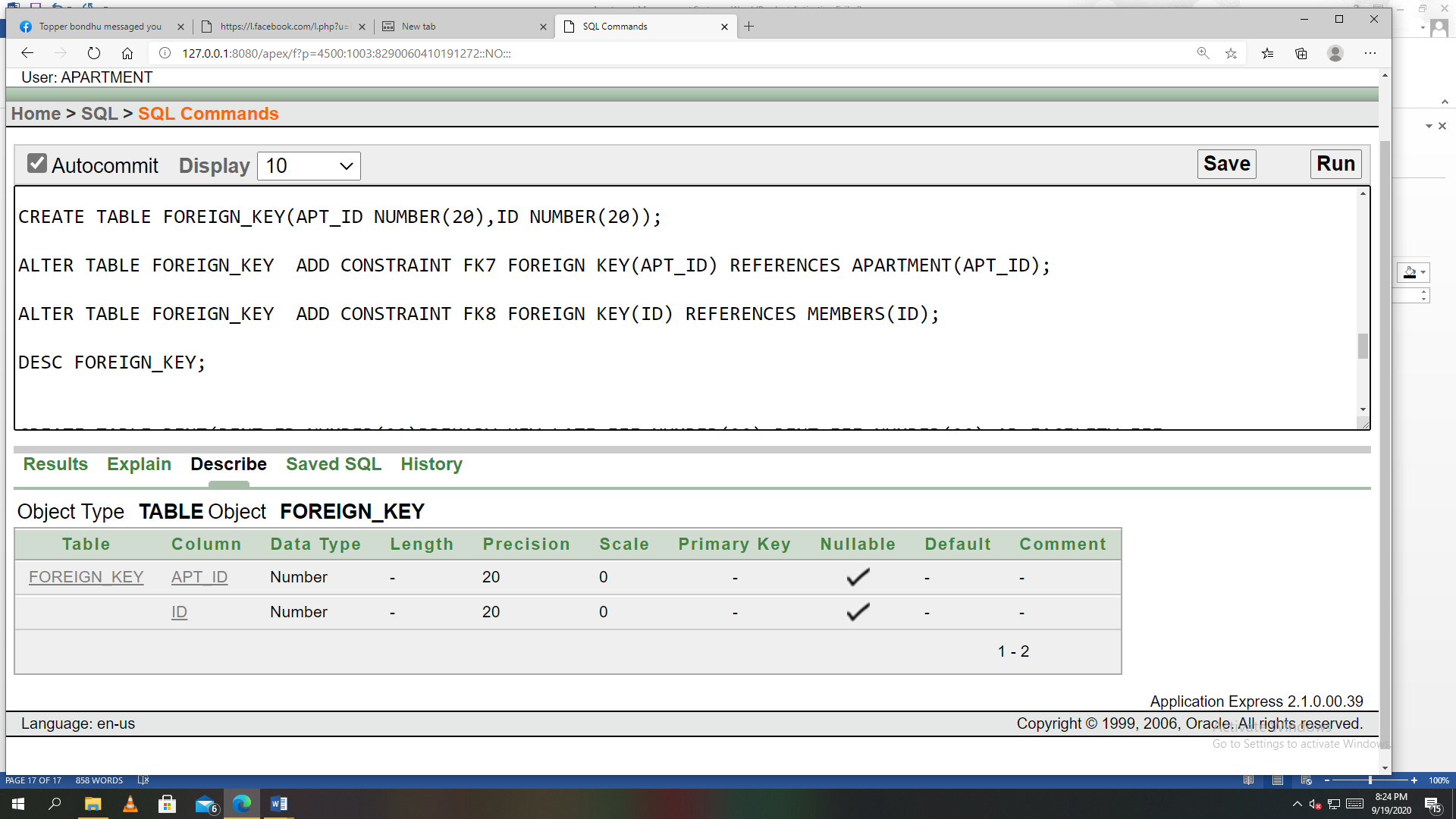


Table RENT:

CREATE TABLE RENT(RENT\_ID NUMBER(20)PRIMARY KEY,LATE\_FEE NUMBER(20),RENT\_FEE NUMBER(20),AP\_FACILITY\_FEE NUMBER(20),PAYMENT\_DATE DATE,PAYMENT\_AMOUNT NUMBER(20),DUE\_DATE DATE);

CREATE SEQUENCE RENT\_RENT\_ID INCREMENT BY 1 START WITH 5005 MAXVALUE 6000 NOCACHE NOCYCLE;

DESC RENT;

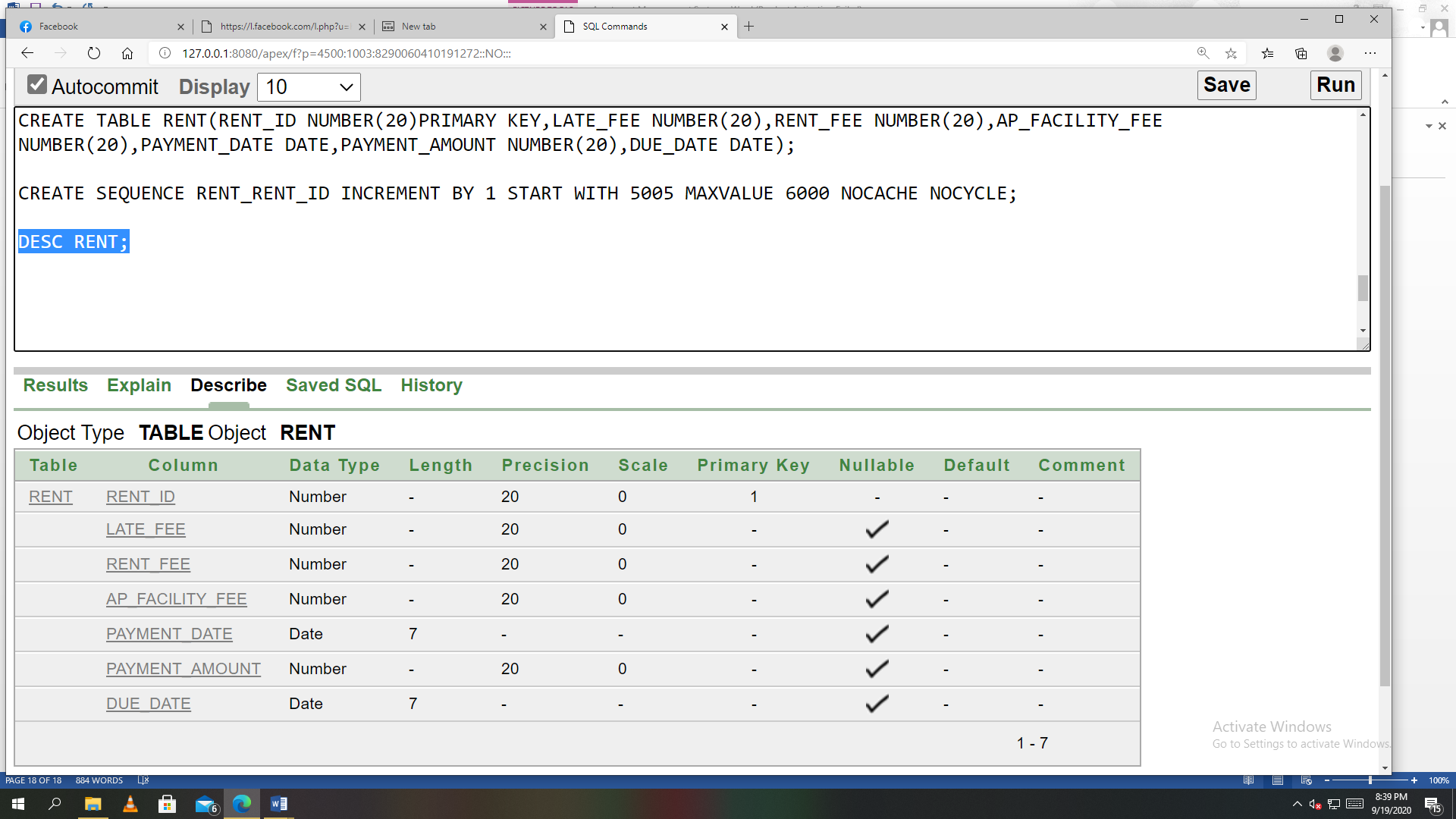


Table FOREIGN\_KEY2:

CREATE TABLE FOREIGN\_KEY2(ID NUMBER(20),RENT\_ID NUMBER(20));

ALTER TABLE FOREIGN\_KEY2 ADD CONSTRAINT FK9 FOREIGN KEY(ID) REFERENCES MEMBERS(ID);

ALTER TABLE FOREIGN\_KEY2 ADD CONSTRAINT FK10 FOREIGN KEY(RENT\_ID) REFERENCES RENT(RENT\_ID);

DESC FOREIGN\_KEY2;



**Data Insertion:**

Table APARTMENT\_BUILDING:

CREATE TABLE APARTMENT\_BUILDING(BUILDING\_ID NUMBER(20)PRIMARY KEY,BUILDING\_SHORT\_NAME VARCHAR2(20),BUILDING\_FULL\_NAME VARCHAR2(20));

INSERT INTO APARTMENT\_BUILDING VALUES (101,'CCH','CITY CENTER,DHAKA');

INSERT INTO APARTMENT\_BUILDING VALUES (102,'HD','HILTON DHAKA');

INSERT INTO APARTMENT\_BUILDING VALUES (103,'NTG','NAVANA TOWER GULSHAN');

INSERT INTO APARTMENT\_BUILDING VALUES (104,'ACI','AZIZ COURT IMPERIAL');

INSERT INTO APARTMENT\_BUILDING VALUES (105,'AV','ANKON VILLA');

CREATE SEQUENCE BUILDING\_BUILDING\_ID INCREMENT BY 1 START WITH 105 MAXVALUE 500 NOCACHE NOCYCLE;

SELECT \* FROM APARTMENT\_BUILDING;

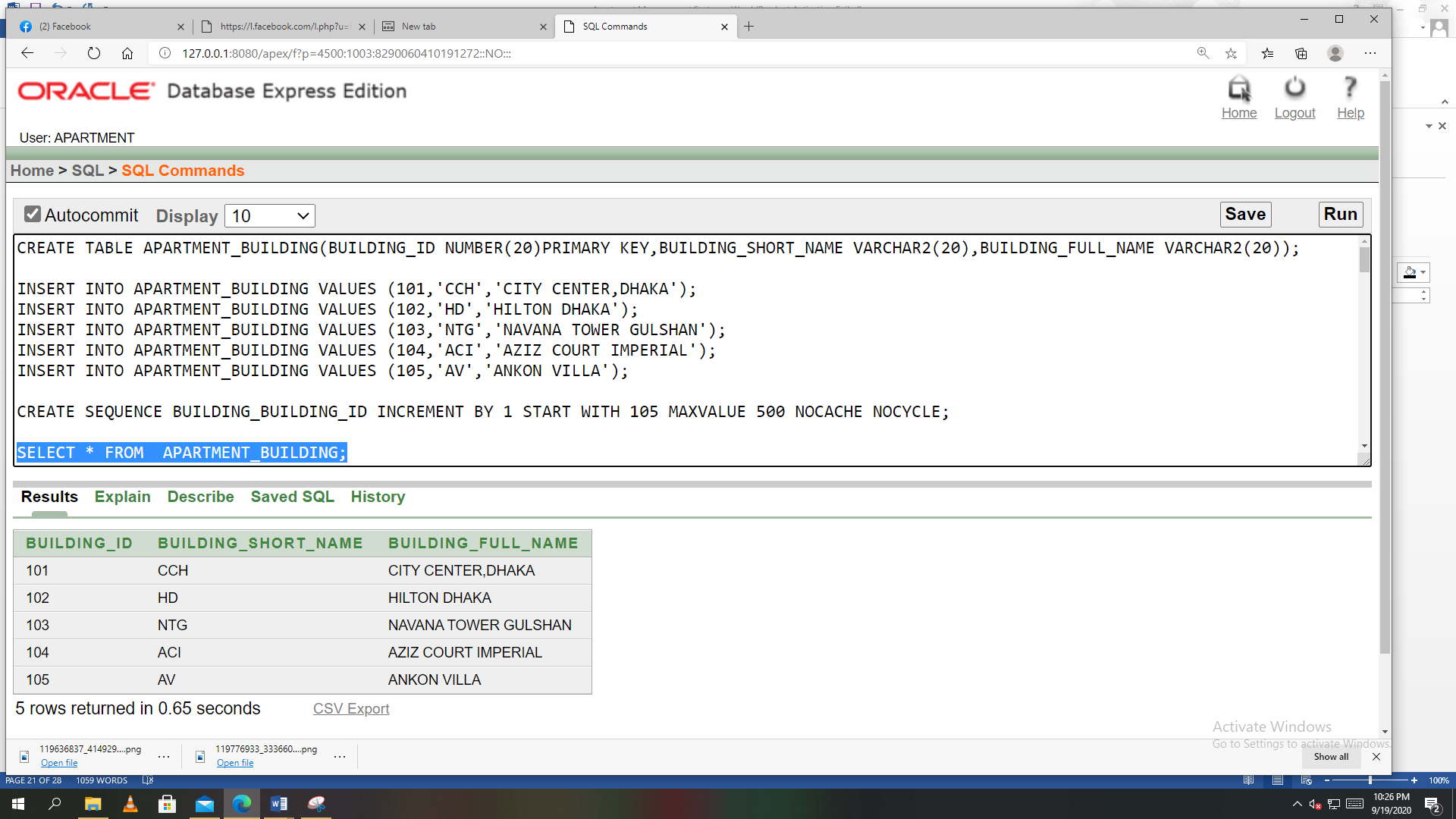


Table APARTMENT:

CREATE TABLE APARTMENT(APT\_ID NUMBER(20)PRIMARY KEY,BATHROOM\_COUNT NUMBER(20),BEDROOM\_COUNT NUMBER(20),RENTAL\_PRICE NUMBER(30),BUILDING\_ID NUMBER(20));

ALTER TABLE APARTMENT ADD CONSTRAINT FK3 FOREIGN KEY(BUILDING\_ID) REFERENCES APARTMENT\_BUILDING(BUILDING\_ID);

INSERT INTO APARTMENT VALUES (1001,3,4,40000,101);

INSERT INTO APARTMENT VALUES (1002,4,5,50000,102);

INSERT INTO APARTMENT VALUES (1003,2,3,35000,103);

INSERT INTO APARTMENT VALUES (1004,3,3,38000,104);

INSERT INTO APARTMENT VALUES (1005,4,6,55000,105);

CREATE SEQUENCE APARTMENT\_APT\_ID INCREMENT BY 1 START WITH 1005 MAXVALUE 5000 NOCACHE NOCYCLE;

SELECT \* FROM APARTMENT;

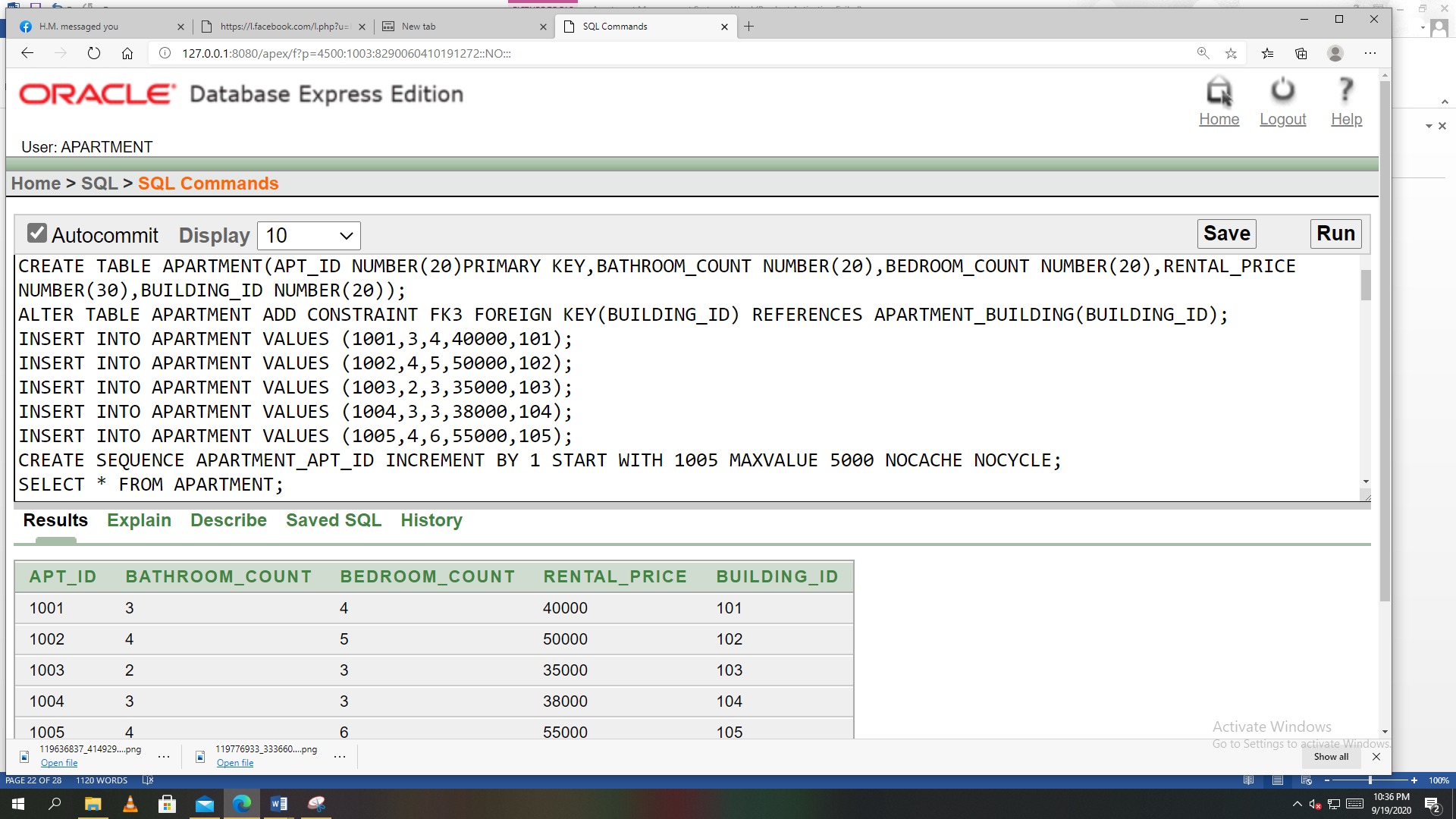


Table APARTMENT\_FACILITIES:

CREATE TABLE APARTMENT\_FACILITIES(AF\_ID NUMBER(20)PRIMARY KEY,WATER VARCHAR2(20),SECURITY VARCHAR2(20),GAS VARCHAR2(20),ELECTRICITY VARCHAR2(20),BUILDING\_ID NUMBER(20));

ALTER TABLE APARTMENT\_FACILITIES ADD CONSTRAINT FK4 FOREIGN KEY(BUILDING\_ID) REFERENCES APARTMENT\_BUILDING(BUILDING\_ID);

INSERT INTO APARTMENT\_FACILITIES VALUES (01,'AVAILABLE','AVAILABLE','AVAILABLE','AVAILABLE',101);

INSERT INTO APARTMENT\_FACILITIES VALUES (02,'AVAILABLE','AVAILABLE','AVAILABLE','AVAILABLE',102);

INSERT INTO APARTMENT\_FACILITIES VALUES (03,'AVAILABLE','AVAILABLE',' NOT AVAILABLE','AVAILABLE',103);

INSERT INTO APARTMENT\_FACILITIES VALUES (04,'AVAILABLE','AVAILABLE','AVAILABLE','AVAILABLE',104);

INSERT INTO APARTMENT\_FACILITIES VALUES (05,'AVAILABLE','AVAILABLE','AVAILABLE','AVAILABLE',105);

CREATE SEQUENCE APARTMENT\_FACILITIES\_AF\_ID INCREMENT BY 1 START WITH 5 MAXVALUE 100 NOCACHE NOCYCLE;

SELECT \* FROM APARTMENT\_FACILITIES;

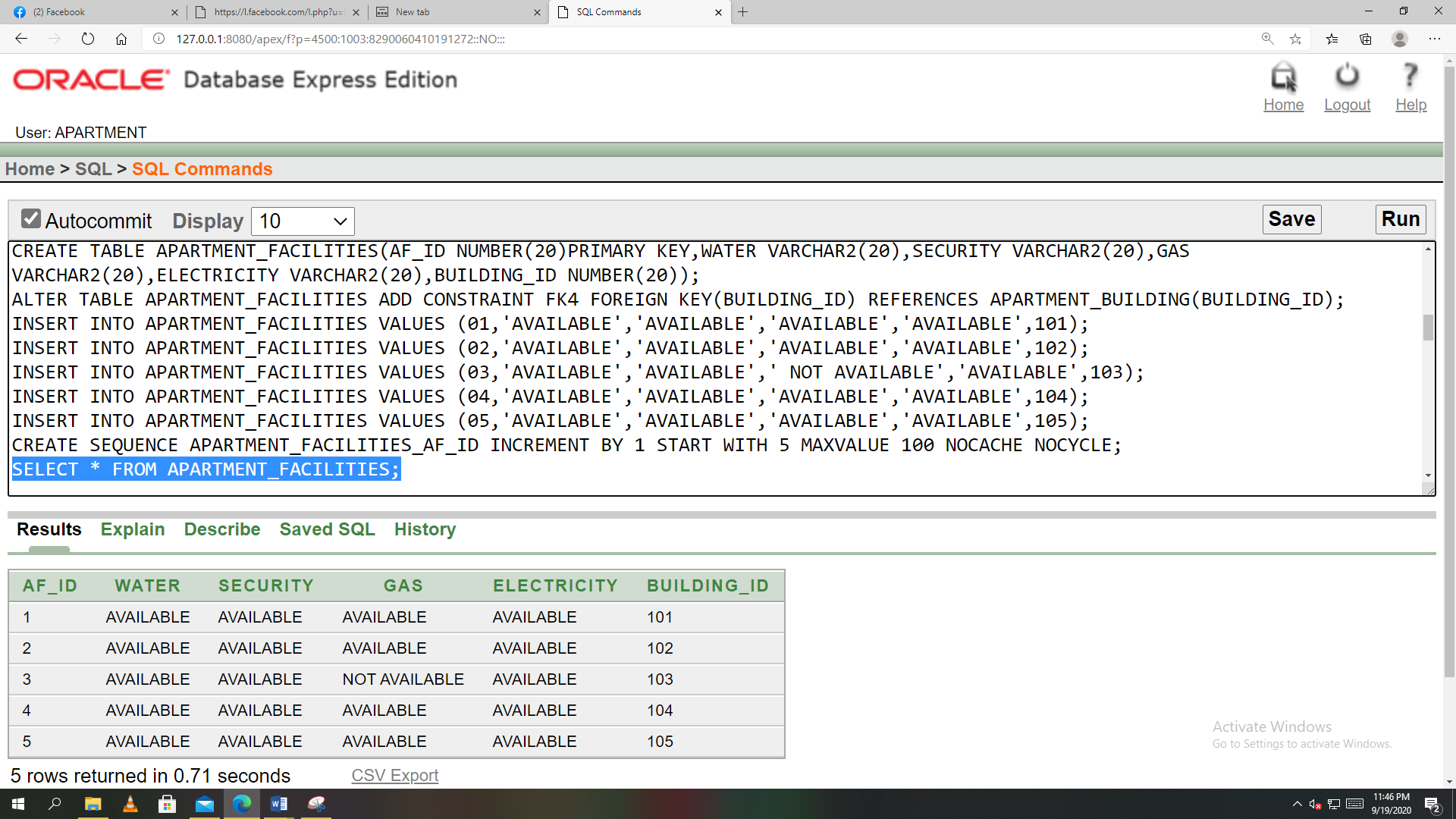


Table MANAGER:

CREATE TABLE MANAGER (MANGER\_ID NUMBER(20)PRIMARY KEY,MANAGER\_NAME VARCHAR2(30),M\_PHONE NUMBER(20),M\_SALARY NUMBER(20),BUILDING\_ID NUMBER(20));

ALTER TABLE MANAGER ADD CONSTRAINT FK5 FOREIGN KEY(BUILDING\_ID) REFERENCES APARTMENT\_BUILDING(BUILDING\_ID);

INSERT INTO MANAGER VALUES (111,'MD.RAHIM',01819809803,30000,101);

INSERT INTO MANAGER VALUES (112,'MD.KARIM',01997239740,35000,102);

INSERT INTO MANAGER VALUES (113,'MD.JASHIM',01831967980,20000,103);

INSERT INTO MANAGER VALUES (114,'MD.NILOY',01819834803,25000,104);

INSERT INTO MANAGER VALUES (115,'MD.RAFI',01991957687,15000,105);

CREATE SEQUENCE MANAGER\_MANAGER\_ID INCREMENT BY 1 START WITH 115 MAXVALUE 500 NOCACHE NOCYCLE;

SELECT \* FROM MANAGER;

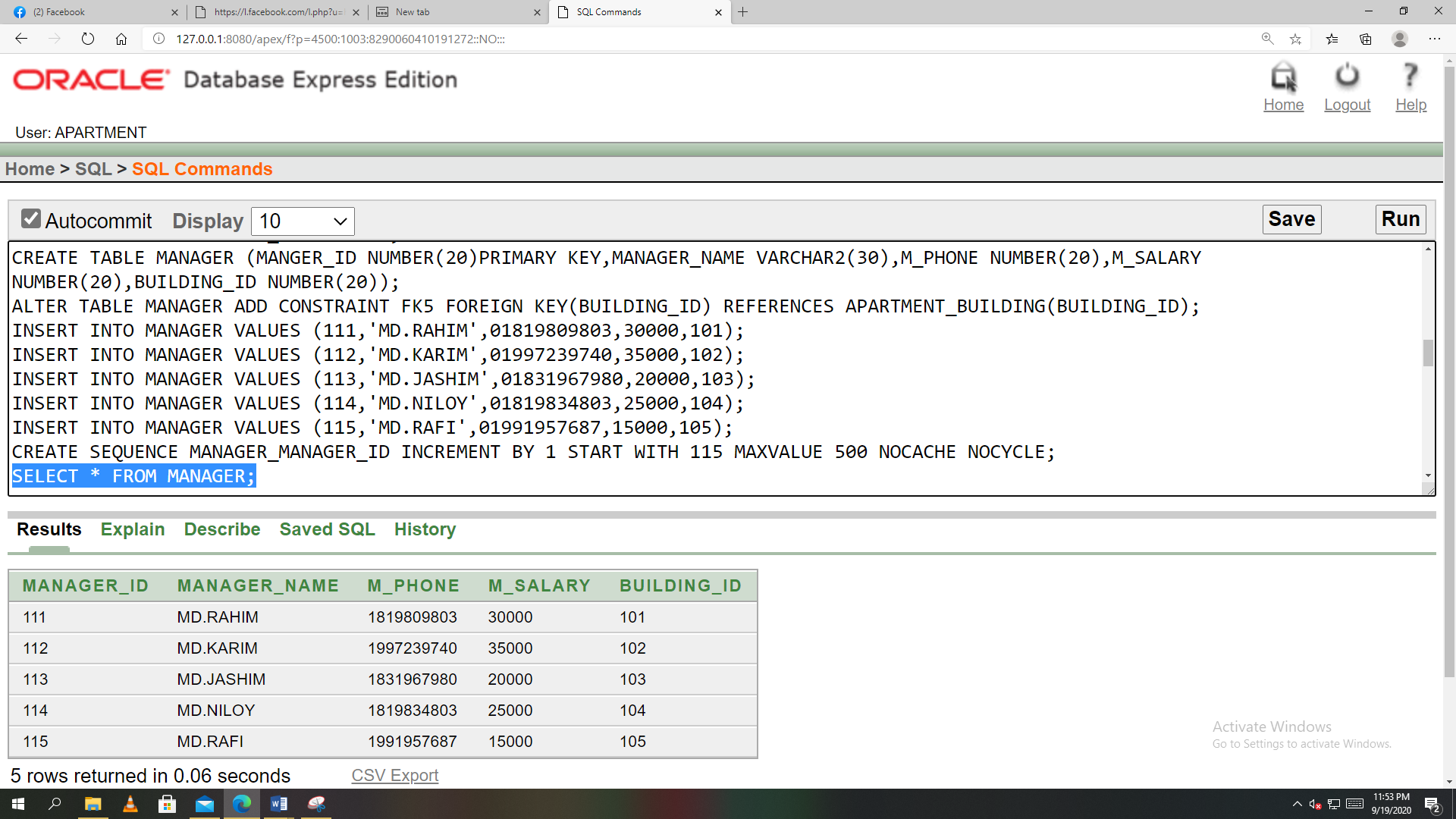


Table MEMBER:

CREATE TABLE MEMBERS(ID NUMBER(20)PRIMARY KEY,USER\_NAME VARCHAR2(20),PASSWORD NUMBER(20),EMAIL VARCHAR2(20),PHONE\_NUM NUMBER(20),MANGER\_ID NUMBER(20));

ALTER TABLE MEMBERS ADD CONSTRAINT FK6 FOREIGN KEY(MANGER\_ID) REFERENCES MANAGER(MANGER\_ID);

INSERT INTO MEMBERS VALUES(4001,'MD.ANKON',9876,'ankon@gmail.com',01987654321,111);

INSERT INTO MEMBERS VALUES(4002,'MD.NAHID',7834,'nahid@gmail.com',01812345678,112);

INSERT INTO MEMBERS VALUES(4003,'Ms.TURNA',8749,'turna@gmail.com',01709123498,113);

INSERT INTO MEMBERS VALUES(4004,'MD.RABBY',4792,'rabby@gmail.com',01934569876,114);

INSERT INTO MEMBERS VALUES(4005,'MD.RAFI',1234,'rafi@gmail.com',01712345678,115);

CREATE SEQUENCE MEMBERS\_ID INCREMENT BY 1 START WITH 4005 MAXVALUE 5000 NOCACHE NOCYCLE;

SELECT \* FROM MEMBERS;

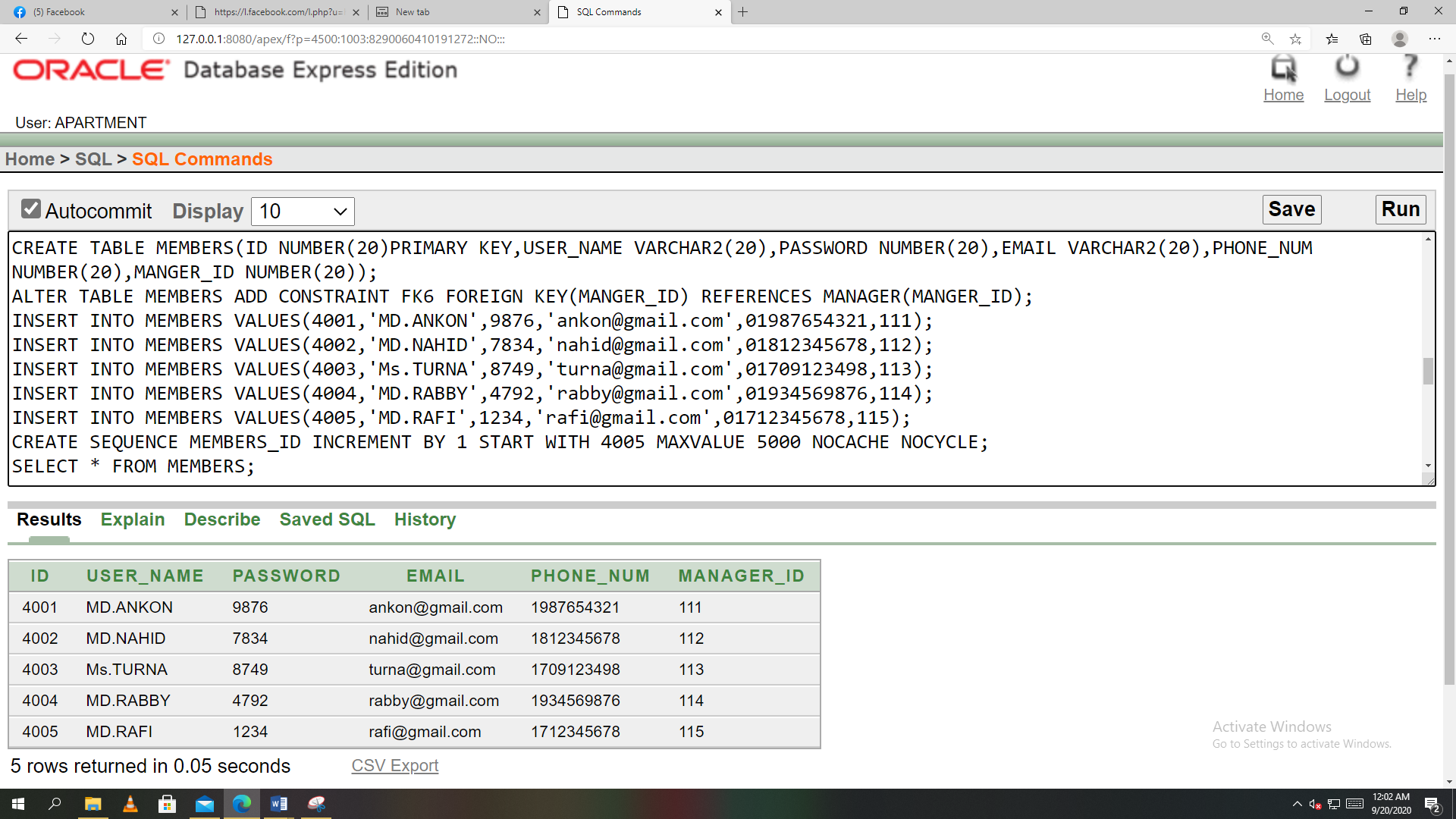


Table FOREIGN\_KEY:

CREATE TABLE FOREIGN\_KEY(APT\_ID NUMBER(20),ID NUMBER(20));

ALTER TABLE FOREIGN\_KEY ADD CONSTRAINT FK7 FOREIGN KEY(APT\_ID) REFERENCES APARTMENT(APT\_ID);

ALTER TABLE FOREIGN\_KEY ADD CONSTRAINT FK8 FOREIGN KEY(ID) REFERENCES MEMBERS(ID);

INSERT INTO FOREIGN\_KEY VALUES (1001,4001);

INSERT INTO FOREIGN\_KEY VALUES (1002,4002);

INSERT INTO FOREIGN\_KEY VALUES (1003,4003);

INSERT INTO FOREIGN\_KEY VALUES (1004,4004);

INSERT INTO FOREIGN\_KEY VALUES (1005,4005);

SELECT \* FROM FOREIGN\_KEY;

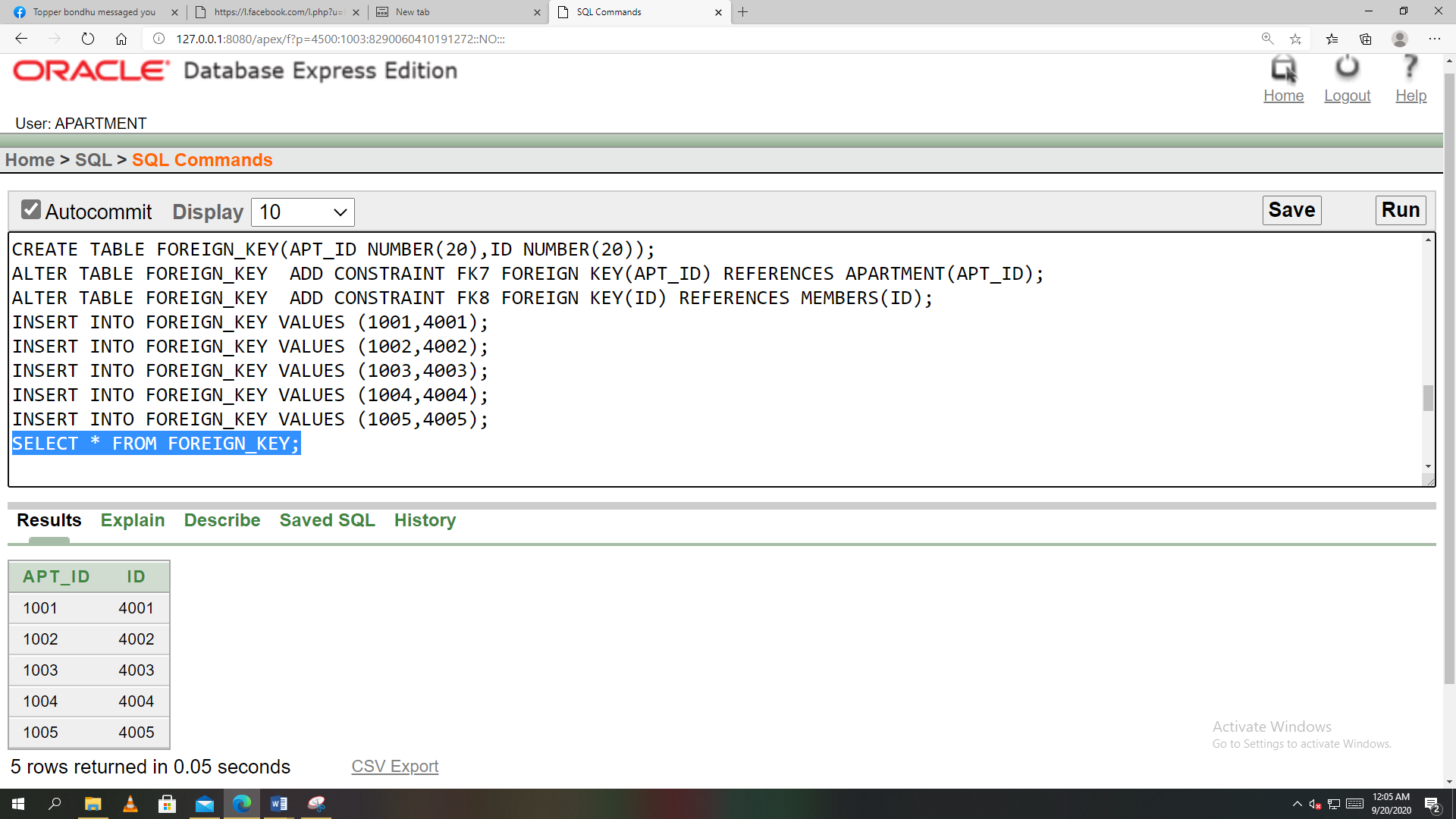


Table RENT:

CREATE TABLE RENT(RENT\_ID NUMBER(20)PRIMARY KEY,LATE\_FEE NUMBER(20),RENT\_FEE NUMBER(20),AP\_FACILITY\_FEE NUMBER(20),PAYMENT\_DATE DATE,PAYMENT\_AMOUNT NUMBER(20),DUE\_DATE DATE);

INSERT INTO RENT VALUES (5001,200,37000,3000,TO\_DATE('5-1-2020','dd-mm-yyyy'),40000,TO\_DATE('10-1-2020','dd-mm-yyyy'));

INSERT INTO RENT VALUES (5002,200,45000,5000,TO\_DATE('1-1-2020','DD-MM-YYYY'),50000,TO\_DATE('10-1-2020','DD-MM-YYYY'));

INSERT INTO RENT VALUES (5003,200,31000,4000,TO\_DATE('4-1-2020','DD-MM-YYYY'),35000,TO\_DATE('15-1-2020','DD-MM-YYYY'));

INSERT INTO RENT VALUES (5004,200,33000,5000,TO\_DATE('1-1-2020','DD-MM-YYYY'),38000,TO\_DATE('10-1-2020','DD-MM-YYYY'));

INSERT INTO RENT VALUES (5005,200,50000,5000,TO\_DATE('5-1-2020','DD-MM-YYYY'),55000,TO\_DATE('14-1-2020','DD-MM-YYYY'));

CREATE SEQUENCE RENT\_RENT\_ID INCREMENT BY 1 START WITH 5005 MAXVALUE 6000 NOCACHE NOCYCLE;

SELECT \* FROM RENT;

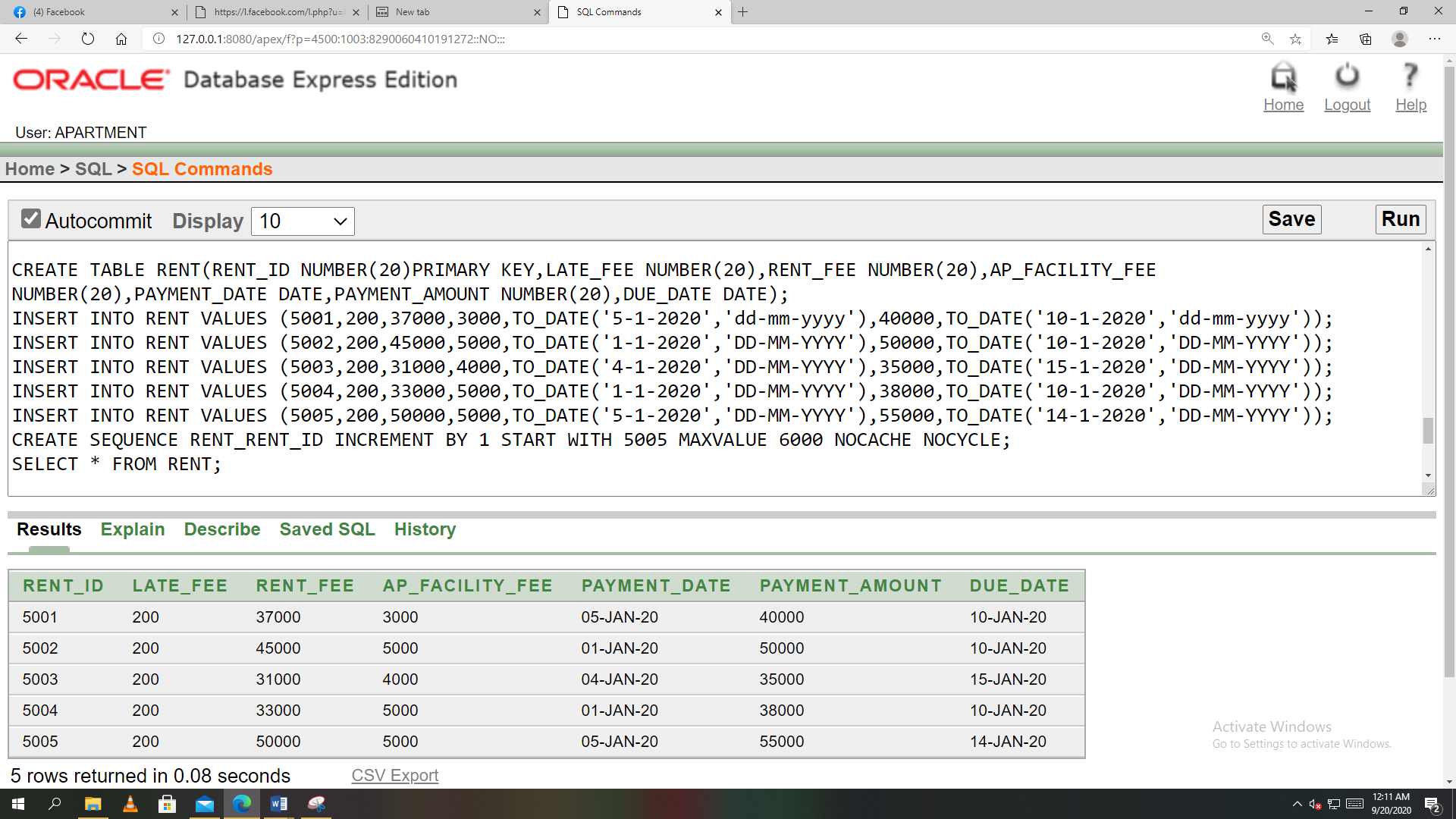


Table FOREIGN\_KEY2:

CREATE TABLE FOREIGN\_KEY2(ID NUMBER(20),RENT\_ID NUMBER(20));

ALTER TABLE FOREIGN\_KEY2 ADD CONSTRAINT FK9 FOREIGN KEY(ID) REFERENCES MEMBERS(ID);

ALTER TABLE FOREIGN\_KEY2 ADD CONSTRAINT FK10 FOREIGN KEY(RENT\_ID) REFERENCES RENT(RENT\_ID);

INSERT INTO FOREIGN\_KEY2 VALUES(4001,5001);

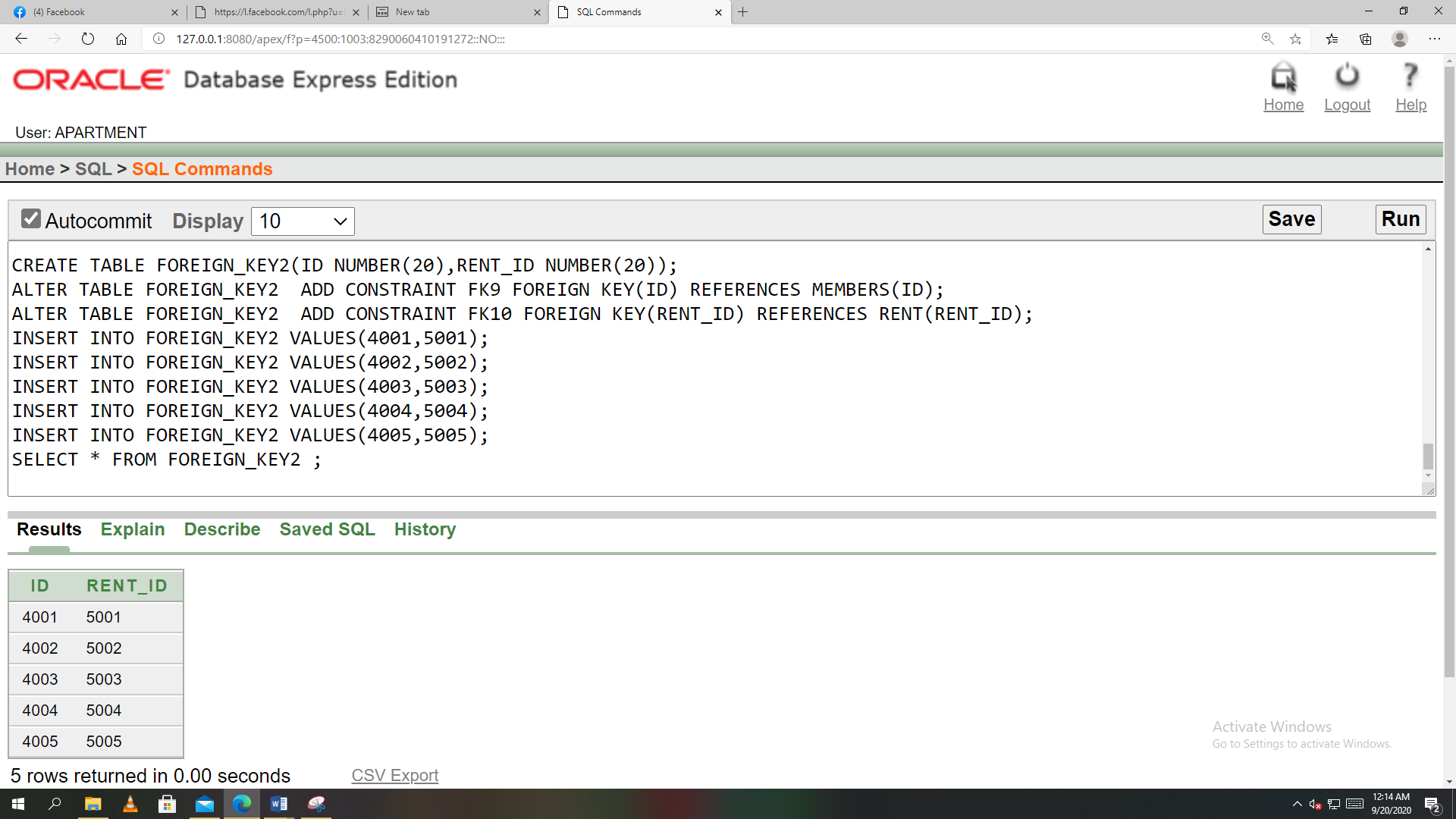
INSERT INTO FOREIGN\_KEY2 VALUES(4002,5002);

INSERT INTO FOREIGN\_KEY2 VALUES(4003,5003);

INSERT INTO FOREIGN\_KEY2 VALUES(4004,5004);

INSERT INTO FOREIGN\_KEY2 VALUES(4005,5005);

SELECT \* FROM FOREIGN\_KEY2 ;

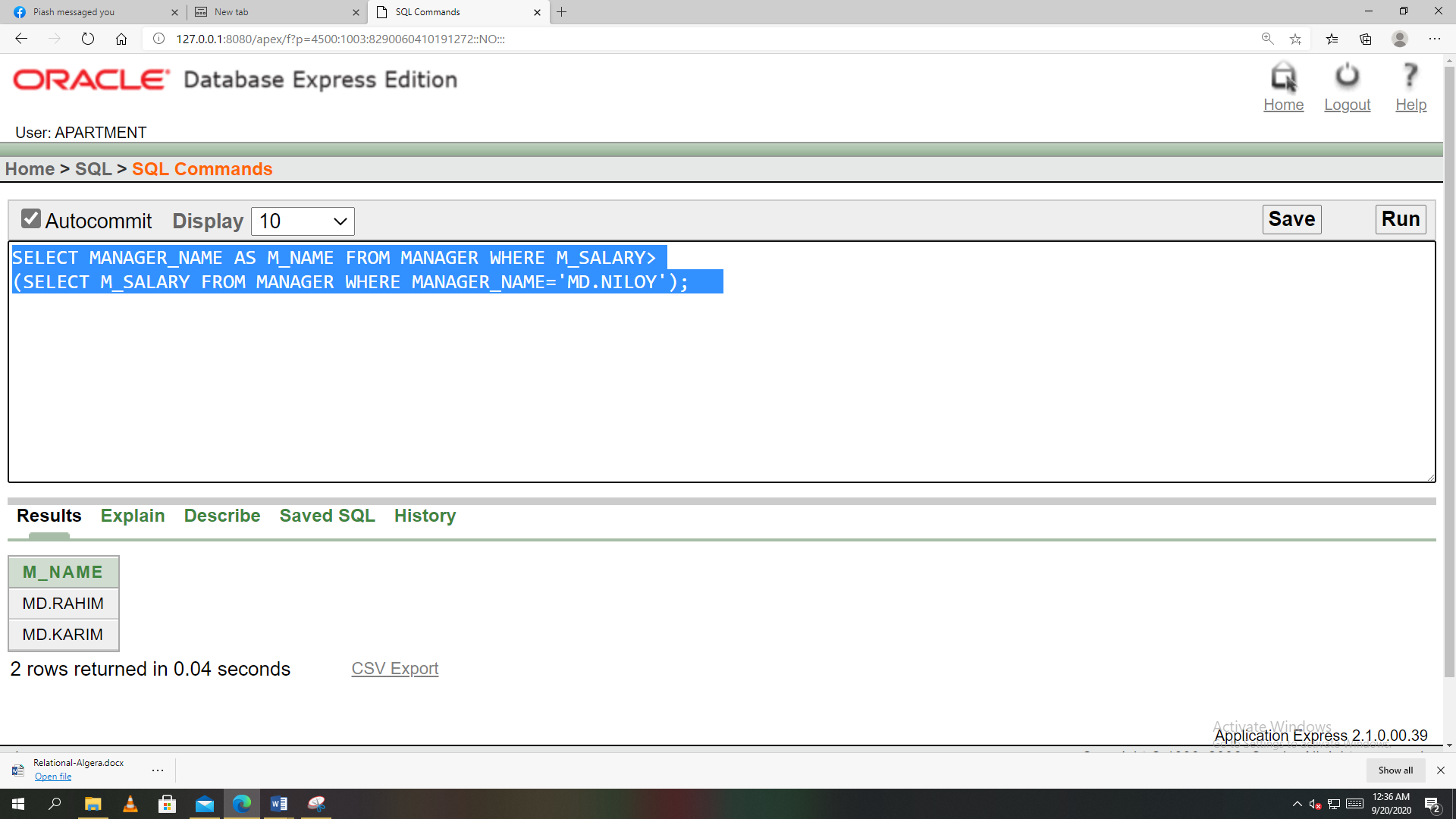


**Query Writing:**

**SUB-QUERY:**

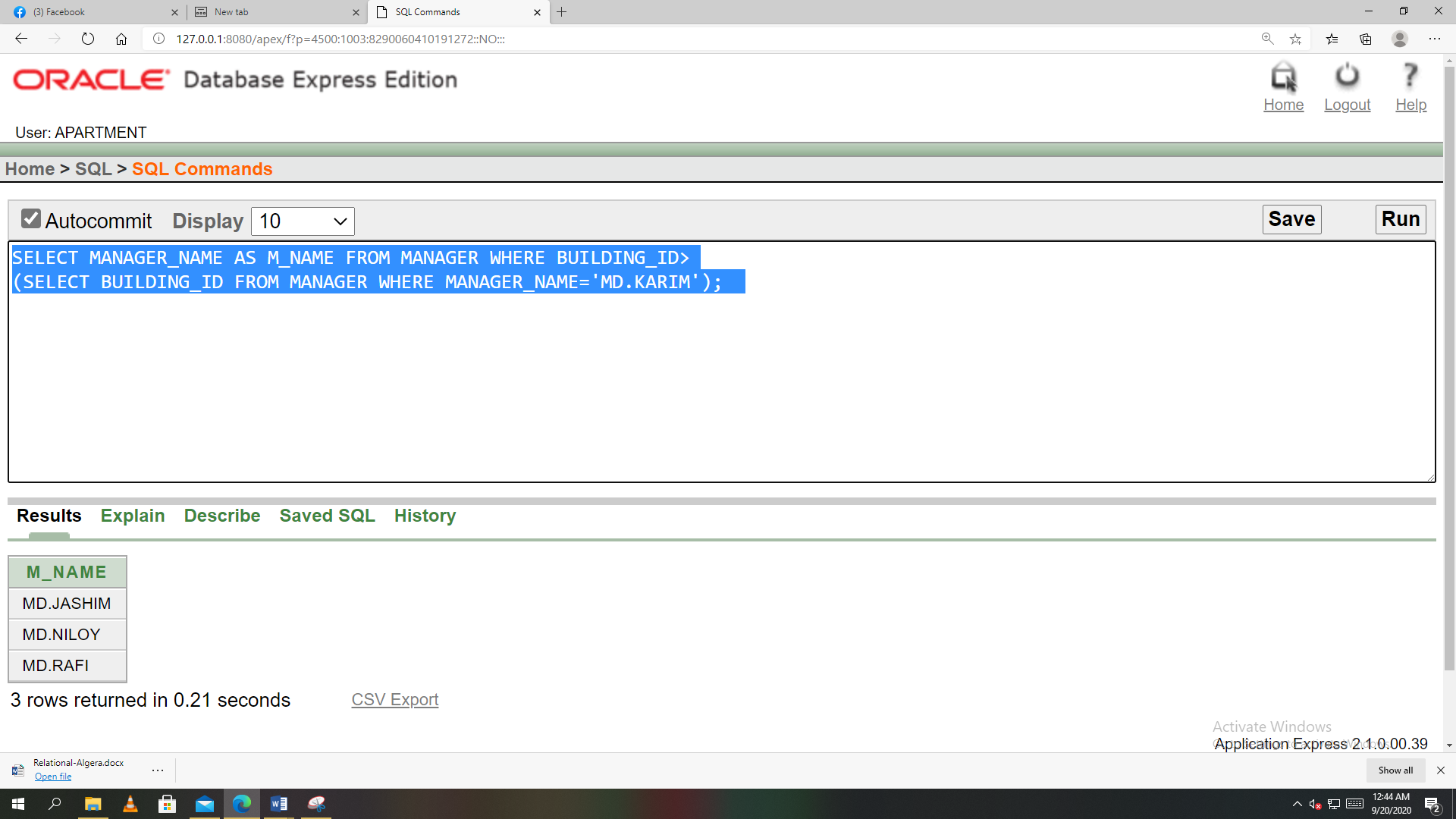
**1. Display the manager name who earns more than MD. Niloy.**

* SELECT MANAGER\_NAME AS M\_NAME FROM MANAGER WHERE M\_SALARY>(SELECT M\_SALARY FROM MANAGER WHERE MANAGER\_NAME='MD.NILOY');



**2. Display the manager name who joined after MD. Karim.**

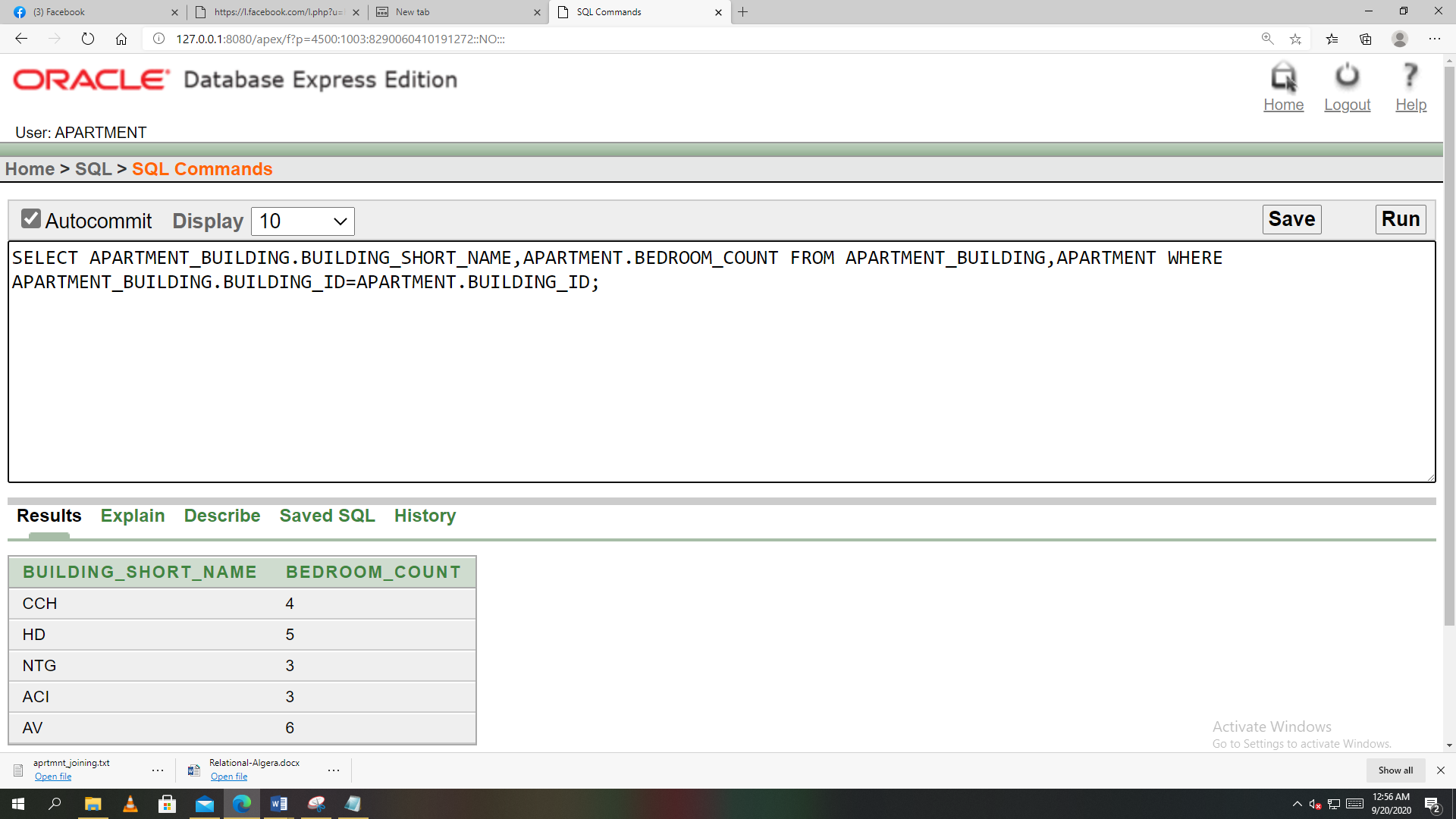
* SELECT MANAGER\_NAME AS M\_NAME FROM MANAGER WHERE BUILDING\_ID>(SELECT BUILDING\_ID FROM MANAGER WHERE MANAGER\_NAME='MD.KARIM');



**JOINING:**

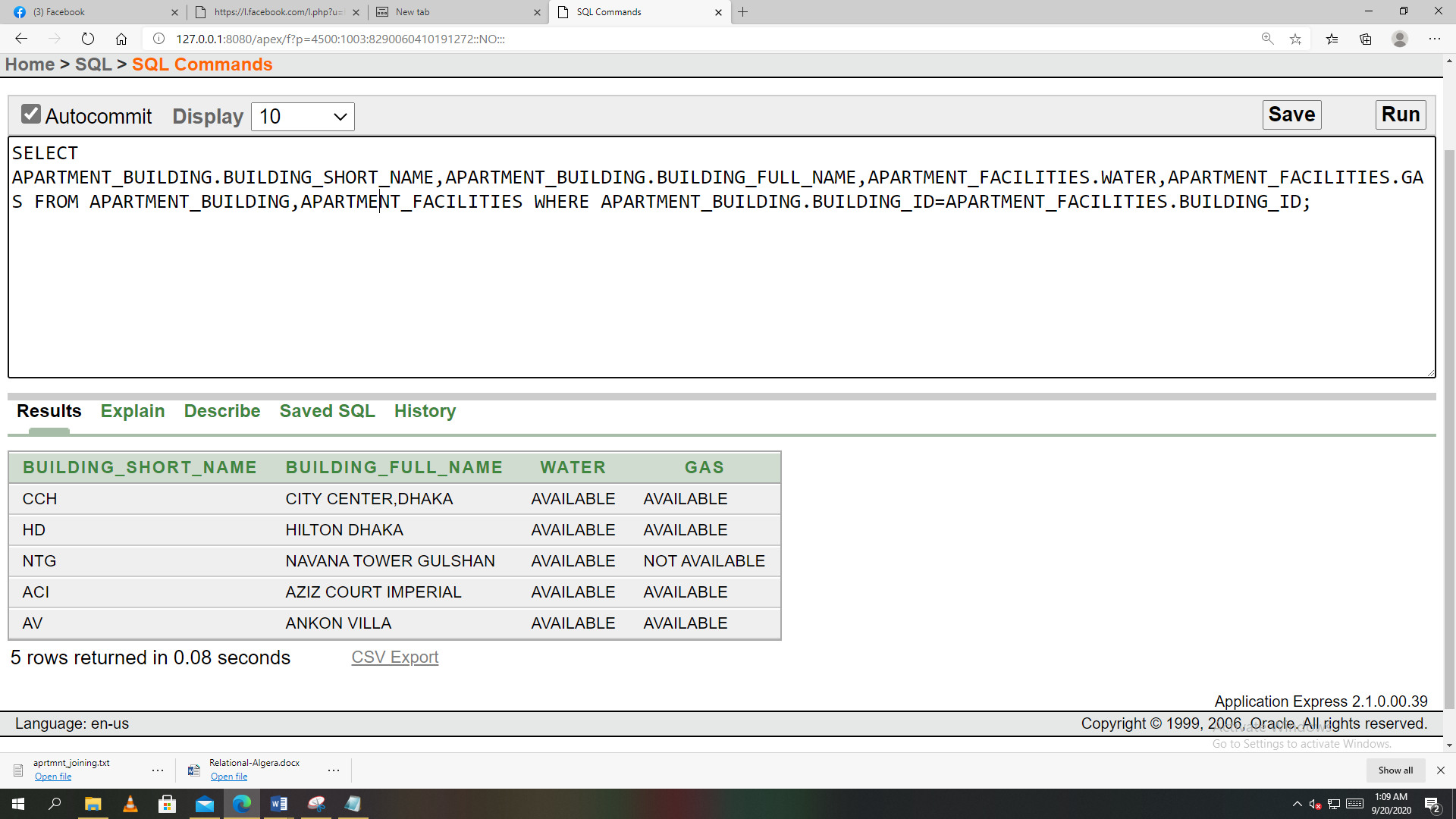
**1. Write a query to display building\_short\_name, building\_full\_name, water, gas from the table apartment\_building and apartment\_facilities?**

* SELECT APARTMENT\_BUILDING.BUILDING\_SHORT\_NAME,APARTMENT.BEDROOM\_COUNT FROM APARTMENT\_BUILDING,APARTMENT WHERE APARTMENT\_BUILDING.BUILDING\_ID=APARTMENT.BUILDING\_ID;



**2. Write a query to display building short name, building full name, water, gas from the table apartment building and apartment facilities ?**

* SELECT APARTMENT\_BUILDING.BUILDING\_SHORT\_NAME,APARTMENT\_BUILDING.BUILDING\_FULL\_NAME,APARTMENT\_FACILITIES.WATER,APARTMENT\_FACILITIES.GAS FROM APARTMENT\_BUILDING,APARTMENT\_FACILITIES WHERE APARTMENT\_BUILDING.BUILDING\_ID=APARTMENT\_FACILITIES.BUILDING\_ID;



**VIEW:**

**1.Create a view called RENTVIEW based on the RENT\_ID,RENT\_FEE and PAYMENT\_AMOUNT from the RENT table.**

* CREATE TABLE RENT(RENT\_ID NUMBER(20)PRIMARY KEY,LATE\_FEE NUMBER(20),RENT\_FEE NUMBER(20),AP\_FACILITY\_FEE NUMBER(20),PAYMENT\_DATE DATE,PAYMENT\_AMOUNT NUMBER(20),DUE\_DATE DATE);

INSERT INTO RENT VALUES (5001,200,37000,3000,TO\_DATE('5-1-2020','dd-mm-yyyy'),40000,TO\_DATE('10-1-2020','dd-mm-yyyy'));

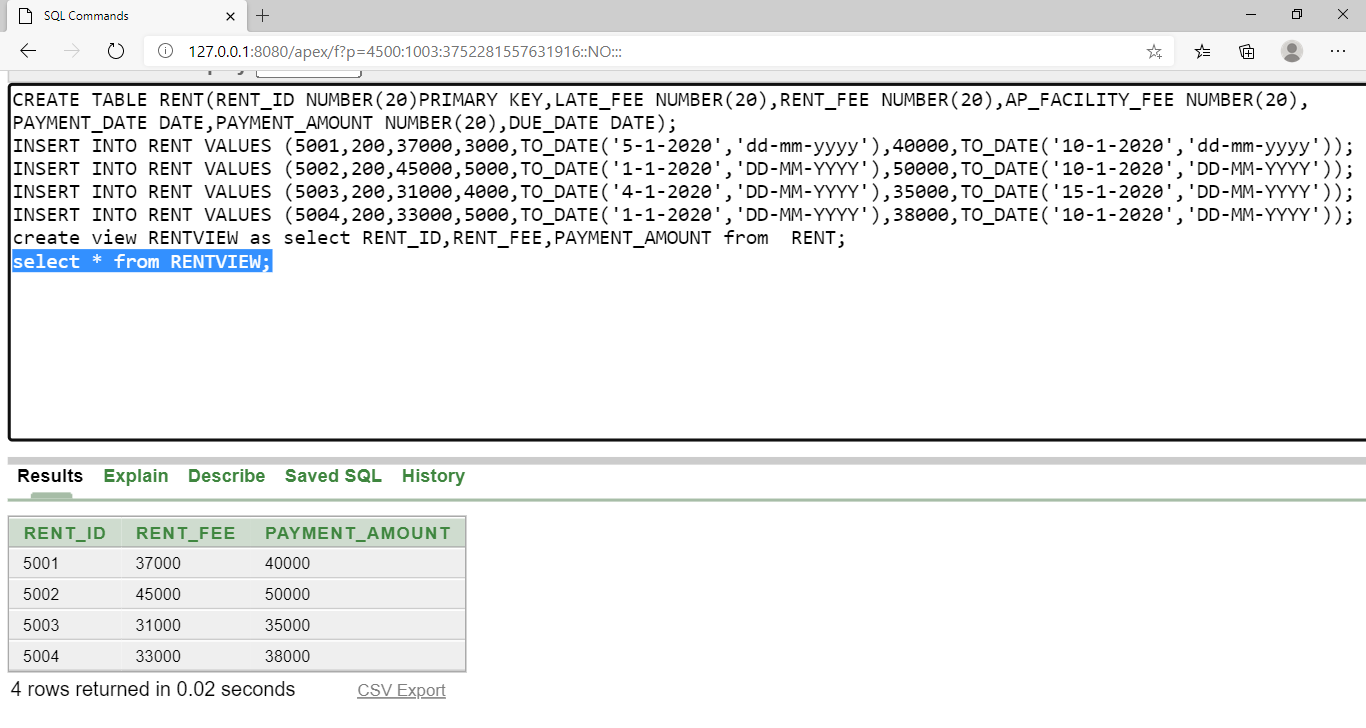
INSERT INTO RENT VALUES (5002,200,45000,5000,TO\_DATE('1-1-2020','DD-MM-YYYY'),50000,TO\_DATE('10-1-2020','DD-MM-YYYY'));

INSERT INTO RENT VALUES (5003,200,31000,4000,TO\_DATE('4-1-2020','DD-MM-YYYY'),35000,TO\_DATE('15-1-2020','DD-MM-YYYY'));

INSERT INTO RENT VALUES (5004,200,33000,5000,TO\_DATE('1-1-2020','DD-MM-YYYY'),38000,TO\_DATE('10-1-2020','DD-MM-YYYY'));

Create view RENTVIEW as select RENT\_ID,RENT\_FEE,PAYMENT\_AMOUNT from RENT;

select \* from RENTVIEW**;**

****

**2. Create a view called APARTMENTVIEW based on the APT\_ID,BATHROOM\_COUNT,BEDROOM\_COUNT and RENTAL\_PRICE from the APARTMENT table.**

* CREATE TABLE APARTMENT(APT\_ID NUMBER(20)PRIMARY KEY,BATHROOM\_COUNT NUMBER(20),BEDROOM\_COUNT NUMBER(20),RENTAL\_PRICE NUMBER(30));

INSERT INTO APARTMENT VALUES (1001,3,4,40000);

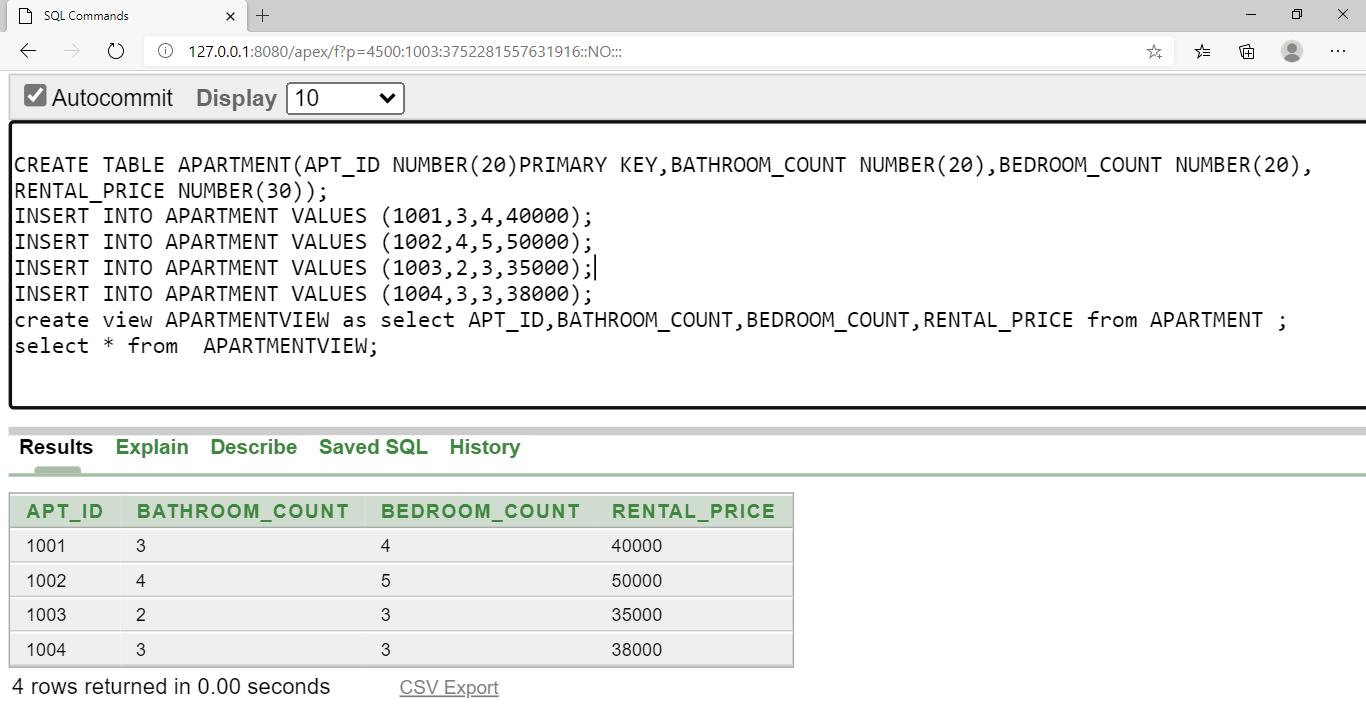
INSERT INTO APARTMENT VALUES (1002,4,5,50000);

INSERT INTO APARTMENT VALUES (1003,2,3,35000);

INSERT INTO APARTMENT VALUES (1004,3,3,38000);

Create view APARTMENTVIEW as select APT\_ID,BATHROOM\_COUNT,BEDROOM\_COUNT,RENTAL\_PRICE from APARTMENT ;

Select \* from APARTMENTVIEW;



**Relational Algebra:**

* + - 1. **Find the manager\_name where m\_salary is greater than 25000.**
* π manager\_name(σ m\_salary>25000(manager)) {Table: MANAGER}
  + - 1. **Find the user\_name and email where manager \_id is 112.**
* π user\_name,email(σ manager\_id =”112”(Member)) {Table: MEMBER}
  + - 1. **Find the apt\_id where rental\_price is greater than 38000.**
* π Apt\_id(σ rental\_price>38000(Apertment)) {Table: APARTMENT\_TABLE}

1. **Find the F\_id where building\_id is 103.**

* π F\_id(σ building\_id=’103’(Apartment\_facilities)) {Table: APARTMENT\_FACILITIES }

1. **Find the rent\_id where payment\_amount is 35000.**

* π rent\_id(σ payment\_amount=”35000”(Rent)) {Table: RENT}

**Conclusion:**

This project is relatively simple to understand and implement. It fulfills all the current requirements of local building management company. The system is very user-friendly; a person with basic computer skills can easily use this system.

Queries of this project **“Apartment Management System”** are run in 'Oracle 10g'.Here we made 6 relationship among all the entities with cardinality.

In our project, we have mentioned the queries as well as inserted screenshots of the tables we created using those queries. Normalization process makes this project simpler.

**This project overall covers the following fields:**

* + - 1. **Apartment details:** This AMS stores all the information, records and data related to apartment. These data include total number of flats and rooms, types of rooms.
      2. **Personal Information:** Personal information of every manager & member is also stored in this system.
      3. **Facilities Provided:** Whether an apartment provides facilities ( water, gas, electricity, security ) or not is also mentioned in this system.
      4. **Rent Collection:** This system also gives solution to rent collection issues.

**Difficulties & Problems**

* **Leak of personal information:** Personal information of manager and members may leak when someone search for an available apartment.

* **Less Security:**  Due to lack of proper security this system can be easily hacked.
* **Parking Slot Confusion:** This system doesn’t give any solution to mark parking slot for each member individually.

**Future Development:**

Adding more security to this system will solve the possibility of information leakage and hacking.

If parking slots are given unique id for per apartment, that will solve the parking slot confusion.

Including GPS tracking in this system will give the system a smart touch. People will be able to find out vacant apartment building through online searching.

Moreover, for one or more apartments in a building, the database has to collect a large amount of data, which makes the system very expensive to upgrade and store data. In future, we will try to make it more convenient to reduce the costing.

If these issues are addressed, it is expected that in Future this system will be more convenient and user friendly.

Lastly, new features can be added into the system as per user requirement. The project is very flexible in that aspect.