

**Course Name** : Introduction to Database

**Project Name** : Apartment Management System

|  |  |  |  |
| --- | --- | --- | --- |
| SL No: | Name | ID | Section |
| 01 | **Islam,Mohammed Aftahi** | **17-34454-2** | **E1** |
| 02 | **Hossain,Farhana Anwar** | **17-34666-2** | **E1** |
| 03 | **Ahmed,Farjana** | **17-34634-2** | **E1** |
| 04 | **Haiat,Md.Asif** | **17-34664-2** | **E2** |

**INTRODUCTION :** A database management system (DBMS) is a computer software application that interacts with ends-user, other application and the database itself to capture an analyze data. A general purpose DBMS allows the definition, creation, querying, update and administration of database.

In our project (Apartment management system) was created by the concept of DBMS. So the main purpose of our project is to know about the basic concept of database management system.

**SCENARIO DESCRIPTION**

In an apartment management system a real estate company is maintained by an owner to build up an apartment. Following this, many real estate company have many owners. Many real estate company builds many apartments. Company is identified by phone(multivalued), address, name, email, founder, license. In this apartment management system manager is also appointed by one owner. Owner is identified by owner id. And an owner have also name, email, address, phone(multivalued). Apartment is identified by apartment no, address and floor number which is also stored. There are many apartment are owned by only one owner. In this system, there are many security employees who are works under manager for the safety of these apartments. Security employees are identified by ID, name, email, address, salary, and phone(multivalued). And a manager is identified by ID, name, email, address, salary, phone(multivalued). Sometimes owner contract a lot of customer to sell apartments. Owners also recruits manager who manages and maintains various apartments. Customer is identified by Id, name, email, address and phone(multivalued). Many renters are rented many apartments. In this case, each renter has ID, name, email, address, occupation, income and phone(multivalued). Sometimes buyers buy apartment. Every buyer have also an ID, name, email, address and phone no(multivalued). There are two types of apartments are builded by real estate companys. Such as residential unit and commercial unit. Residential has bedrooms, bathrooms and fireplace. And commercial type apartments has display capacity and storage capacity. There are many visitors visits these apartments. When a visitor visit in these apartments , security employees checks the visitors check in and check out.

**COMPANY---\*---BUILD---\*---APARTMENT**

**UNF :**

Build (license, founder, name, address, email, phone, ap\_no, floor\_number, address)

**1NF :**

Phone is a multivalued attribute.

1. (license, founder, name, address, email, phone, ap\_no, floor\_number, address)

**2NF :**

1. (license, founder, name, address, email, phone)
2. (ap\_no, floor\_number, address)

**3NF :**

(Already have 2NF and same as that)

1. (license, founder, name, address, email, phone)
2. (ap\_no, floor\_number, address)

**TABLE :**

1. (license, founder, name, address, email, phone)
2. (ap\_no, floor\_number, address)
3. (**license**, **ap\_no**)

**COMPANY---\*---MAINTAIN---\*---OWNER**

**UNF :**

Maintain (license, founder, name, address, email, phone, ow\_id, name, email, address, phone)

**1NF :**

Phone is a multivalued attribute.

1. (license, founder, name, address, email, phone, ow\_id, name, email, address, phone)

**2NF :**

1. (license, founder, name, address, email, phone)
2. (ow\_id, name, email, address, phone)

**3NF :**

(Already have 2NF and same as that)

1. (license, founder, name, address, email, phone)
2. (ow\_id, name, email, address, phone)

**TABLE :**

1. (license, founder, name, address, email, phone)
2. (ow\_id, name, email, address, phone)
3. (**license**, **ow\_id**)

**APARTMENT---\*---OWNED---1---OWNER**

**UNF :**

Owned (Ap\_no, floor\_number, address, Ow\_id, name, email, address, phone)

**1NF :**

Phone is a multivalued attribute.

1. (Ap\_no, floor\_number, address, Ow\_id, name, email, address, phone)

**2NF :**

1. (Ap\_no, floor\_number, address)
2. (Ow\_id, name, email, address, phone)

**3NF :**

(Already have 2NF and same as that)

1. (Ap\_no, floor\_number, address)
2. (Ow\_id, name, email, address, phone)

**TABLE :**

1. (Ap\_no, floor\_number, address, **Ow\_id**)
2. (Ow\_id, name, email, address, phone)

**MANAGER---\*---APPOINTED---1---OWNER**

**UNF :**

Appointed (Man\_id, name, email, address, salary, phone, Ow\_id, name, email, address, phone)

**1NF :**

Phone is a multivalued attribute.

1. (Man\_id, name, email, address, salary, phone, Ow\_id, name, email, address, phone)

**2NF :**

1. (Man\_id, name, email, address, salary, phone)
2. (Ow\_id, name, email, address, phone)

**3NF :**

(Already have 2NF and same as that)

1. (Man\_id, name, email, address, salary, phone)
2. (Ow\_id, name, email, address, phone)

**TABLE :**

1. (Man\_id, name, email, address, salary, phone, **Ow\_id**)
2. (Ow\_id, name, email, address, phone)

**APARTMENT---\*---RENT---\*---RENTER**

**UNF :**

Rent (Ap\_no, floor\_number, address, Rent\_id, name, email, address, income, occupation, phone)

**1NF :**

Phone is a multivalued attribute.

1. (Ap\_no, floor\_number, address, Rent\_id, name, email, address, income, occupation, phone)

**2NF :**

1. (Ap\_no, floor\_number, address)
2. (Rent\_id, name, email, address, income, occupation, phone)

**3NF :**

(Already have 2NF and same as that)

1. (Ap\_no, floor\_number, address)
2. (Rent\_id, name, email, address, income, occupation, phone)

**TABLE :**

1. (Ap\_no, floor\_number, address)
2. (Rent\_id, name, email, address, income, occupation, phone)
3. (**Ap\_no**, **Rent\_id**)

**MANAGER---\*---WORKS FOR---\*---SECURITY EMPLOYEE**

**UNF :**

Works for (Man\_id, name, email, address, salary, phone, Sec\_id, name, email, address, salary, phone)

**1NF :**

Phone is a multivalued attribute.

1. (Man\_id, name, email, address, salary, phone, Sec\_id, name, email, address, salary, phone)

**2NF :**

1. (Man\_id, name, email, address, salary, phone)
2. (Sec\_id, name, email, address, salary, phone)

**3NF :**

(Already have 2NF and same as that)

1. (Man\_id, name, email, address, salary, phone)
2. (Sec\_id, name, email, address, salary, phone)

**TABLE :**

1. (Man\_id, name, email, address, salary, phone)
2. (Sec\_id, name, email, address, salary, phone)
3. (**Man\_id**, **Sec\_id**)

**VISITOR---\*---VISITING---\*---SECURITY EMPLOYEE**

**UNF :**

Visiting (Check\_in, check\_out, Sec\_id, name, email, address, salary, phone)

**1NF :**

Phone is a multivalued attribute.

1. (Check\_in, check\_out, Sec\_id, name, email, address, salary, phone)

**2NF :**

1. (Check\_in, check\_out)
2. (Sec\_id, name, email, address, salary, phone)

**3NF :**

(Already have 2NF and same as that)

1. (Check\_in, check\_out)
2. (Sec\_id, name, email, address, salary, phone)

**TABLE :**

1. (c\_id, Check\_in, check\_out)
2. (Sec\_id, name, email, address, salary, phone)
3. (**c\_id**, **Sec\_id**)

**APARTMENT---\*---HAS---\*---UNIT**

**UNF :**

Has (Ap\_no, floor\_number, address, no\_bedrooms, no\_bathrooms, fireplace, display\_capacity, storage\_capacity)

**1NF :**

1. (Ap\_no, floor\_number, address, no\_bedrooms, no\_bathrooms, fireplace, display\_capacity, storage\_capacity)

**2NF :**

1. (Ap\_no, floor\_number, address)
2. (no\_bedrooms, no\_bathrooms, fireplace, display\_capacity, storage\_capacity)

**3NF :**

1. (Ap\_no, floor\_number, address)
2. (no\_bedrooms, no\_bathrooms, fireplace)
3. (display\_capacity, storage\_capacity)

**TABLE :**

1. (Ap\_no, floor\_number, address)
2. (r\_id, no\_bedrooms, no\_bathrooms, fireplace, **co\_id**)
3. (co\_id, display\_capacity, storage\_capacity)
4. (**Ap\_no**, **r\_id**)

**APARTMENT---\*---BUY---\*---BUYER**

**UNF :**

Buy (Ap\_no, floor\_number, address, Buyer\_id, name, email, address, phone)

**1NF :**

Phone is a multivalued attribute.

1. (Ap\_no, floor\_number, address, Buyer\_id, name, email, address, phone)

**2NF :**

1. (Ap\_no, floor\_number, address)
2. (Buyer\_id, name, email, address, phone)

**3NF :**

(Already have 2NF and same as that)

1. (Ap\_no, floor\_number, address)
2. (Buyer\_id, name, email, address, phone)

**TABLE :**

1. (Ap\_no, floor\_number, address)
2. (Buyer\_id, name, email, address, phone)
3. (**Ap\_no**, **Buyer\_id**)

**OWNER---\*---CONTRACT---\*---CUSTOMER**

**UNF :**

Contract (Ow\_id, name, email, address, phone, Cus\_id, name, email, address, phone)

**1NF :**

Phone is a multivalued attribute.

1. (Ow\_id, name, email, address, phone, Cus\_id, name, email, address, phone)

**2NF :**

1. (Ow\_id, name, email, address, phone)
2. (Cus\_id, name, email, address, phone)

**3NF :**

(Already have 2NF and same as that)

1. (Ow\_id, name, email, address, phone)
2. (Cus\_id, name, email, address, phone)

**TABLE :**

1. (Ow\_id, name, email, address, phone)
2. (Cus\_id, name, email, address, phone)
3. (**Ow\_id**, **Cus\_id**)

**TABLE CREATION :**

1. (license, founder, name, address, email, phone)
2. (**license**, **ap\_no**)
3. (Ow\_id, name, email, address, phone)
4. (**license**, **Ow\_id**)
5. (ap\_no, floor\_number, address, **Ow\_id**)
6. (Man\_id, name, email, address, salary, phone, **Ow\_id**)
7. (Rent\_id, name, email, address, income, occupation, phone)
8. (**ap\_no**, **Rent\_id**)
9. (Sec\_id, name, email, address, salary, phone)

10. (**Man\_id**, **Sec\_id**)

11. (c\_id, check\_in, check\_out)

12. (**c\_id**, **Sec\_id**)

13. (r\_id, no\_bedrooms, no\_bathrooms, fireplace, **co\_id**)

14. (co\_id, display\_capacity, storage\_capacity)

15. (**ap\_no**, **r\_id**)

16. (Buyer\_id, name, email, address, phone)

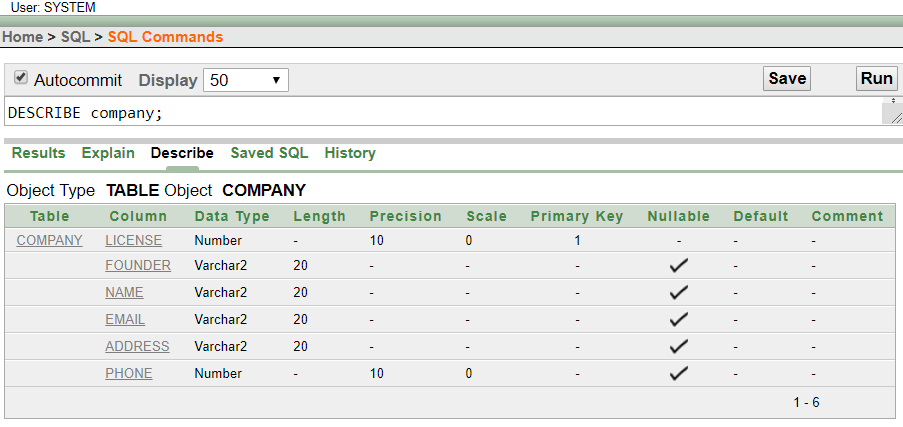
17. (**ap\_no**, **Buyer\_id**)

18. (Cus\_id, name, email, address, phone)

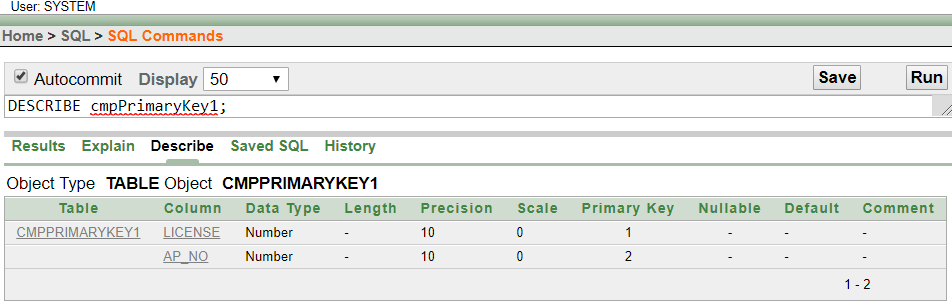
19. (**Ow\_id**, **Cus\_id**)

* **TABLE DESCRIPTION :**

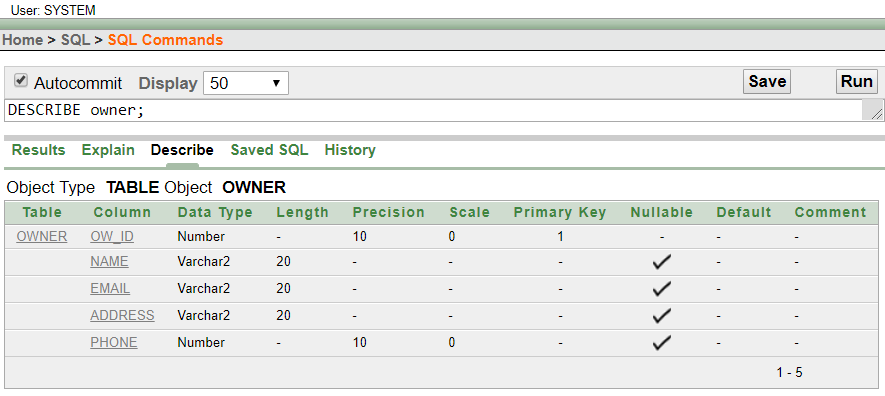
**1.** COMPANY :



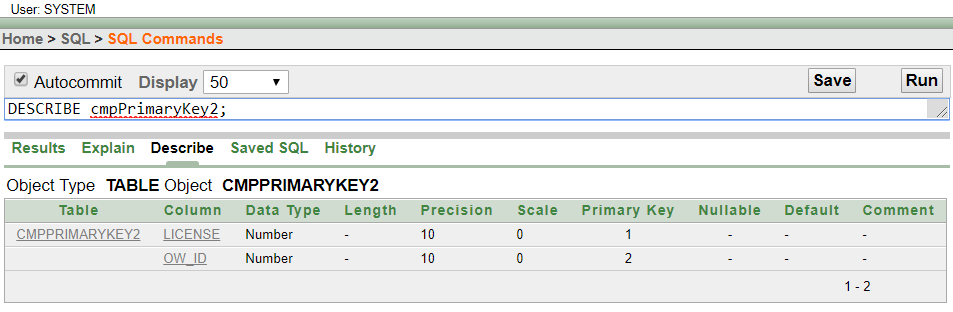
**2.** CMPPRIMARYKEY1:



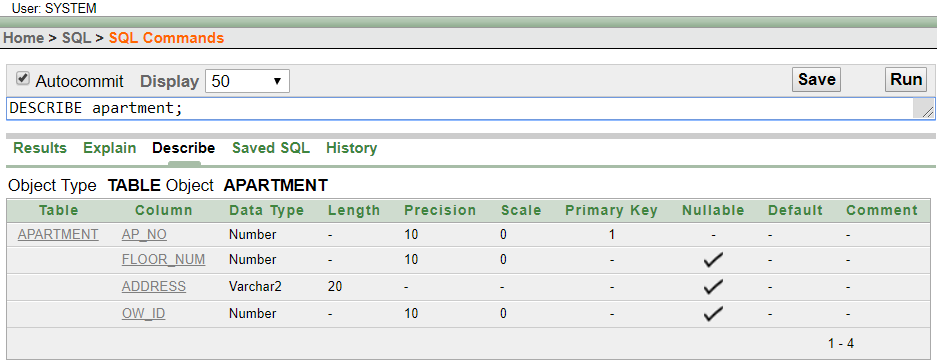
**3.** OWNER :



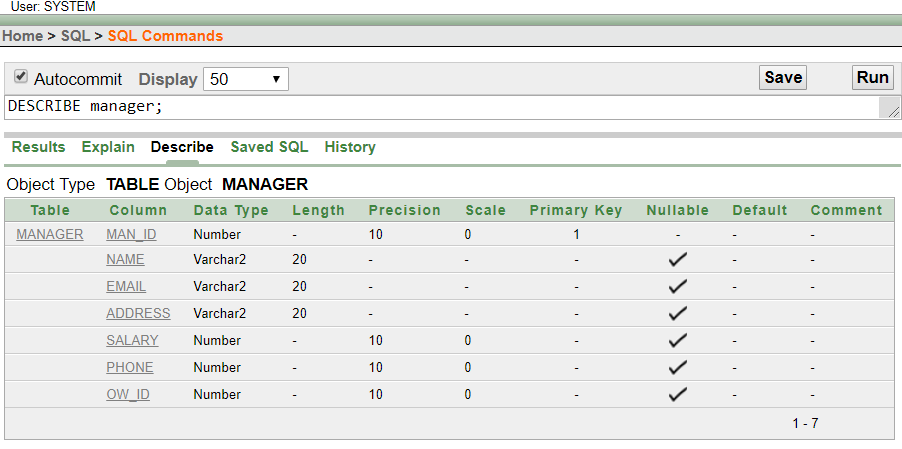
**4.** CMPPRIMARYKEY2 :



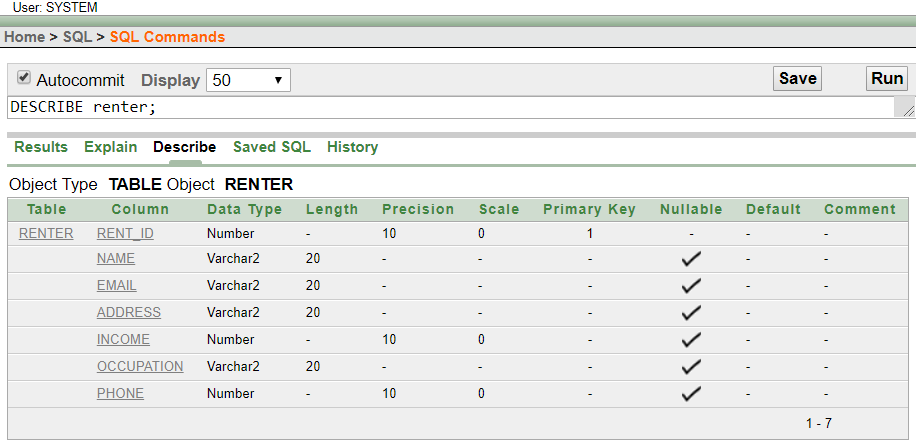
**5.** APARTMENT :



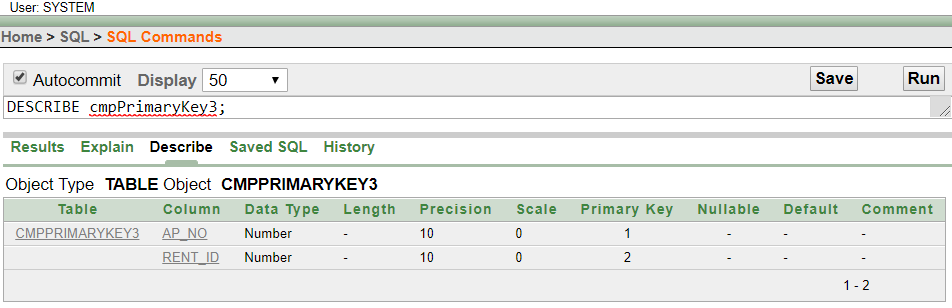
**6.** MANAGER :



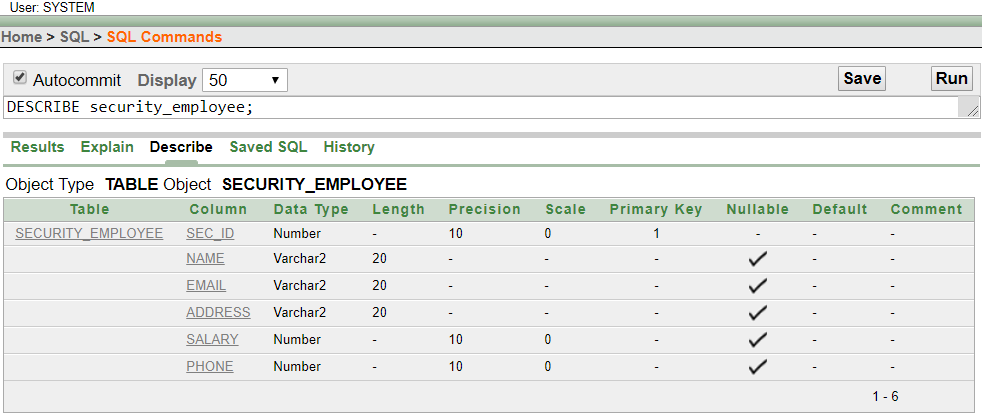
**7.** RENTER :



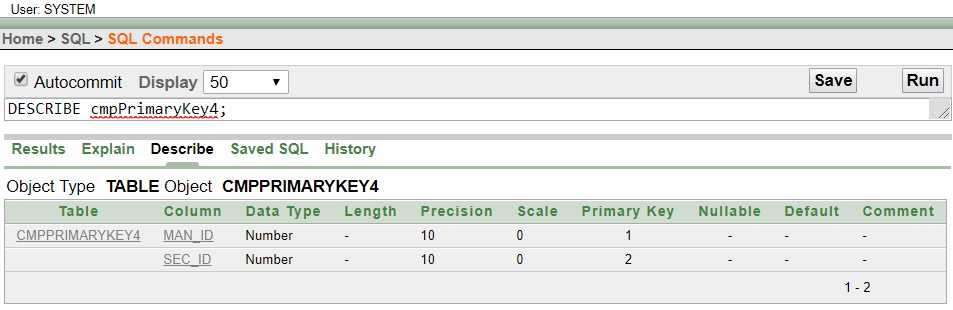
**8.** CMPPRIMARYKEY3 :



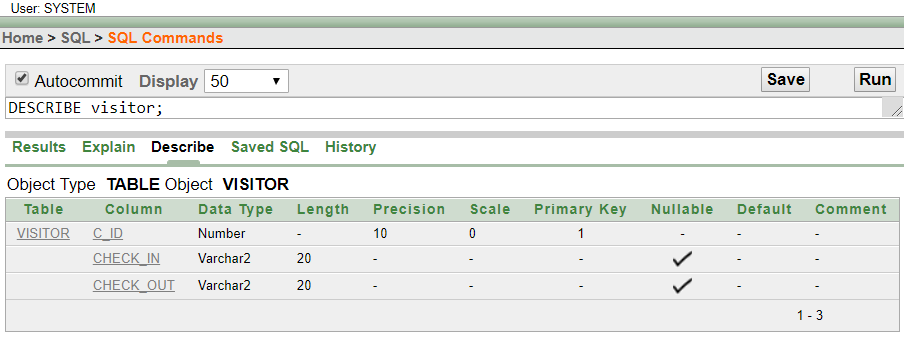
**9.** SECURITY\_EMPLOYEE :



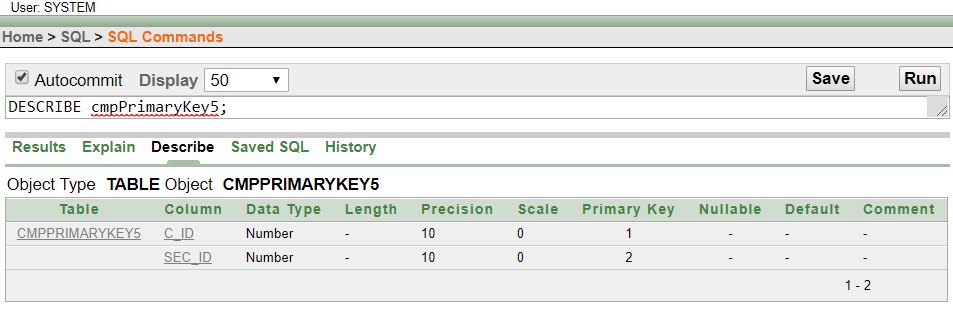
**10.** CMPPRIMARYKEY4 :



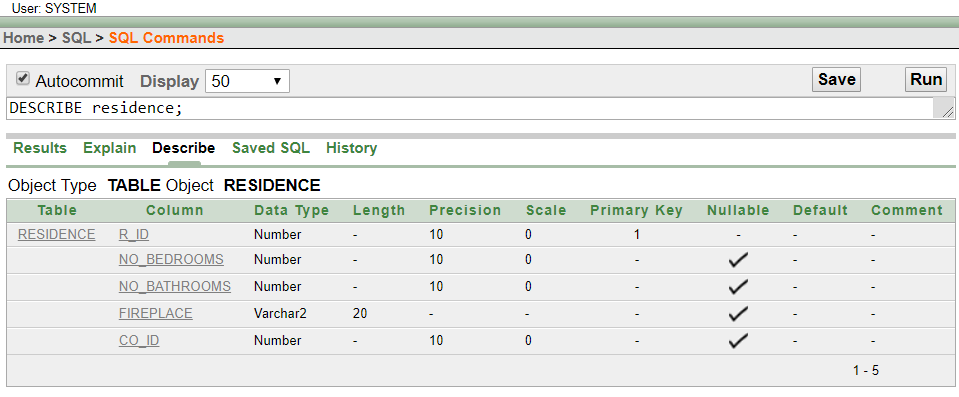
**11.** VISITOR :



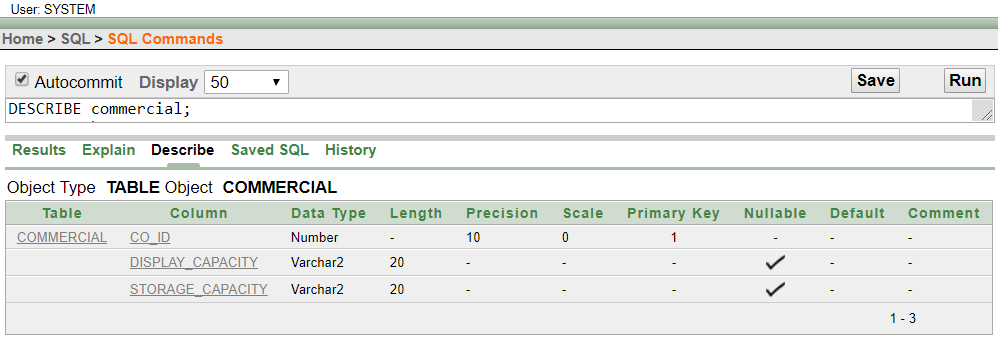
**12.** CMPPRIMARYKEY5 :



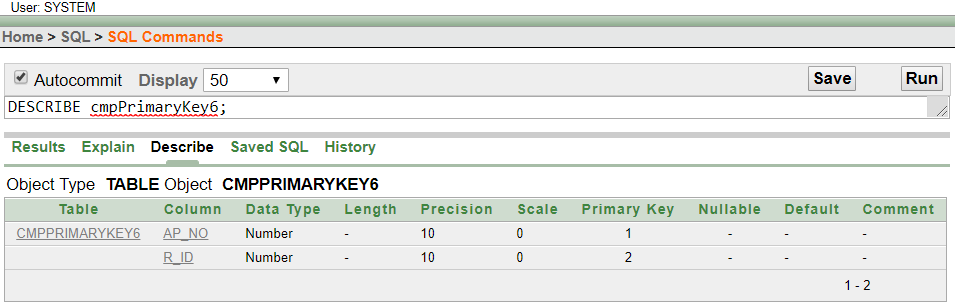
**13.** RESIDENCE :



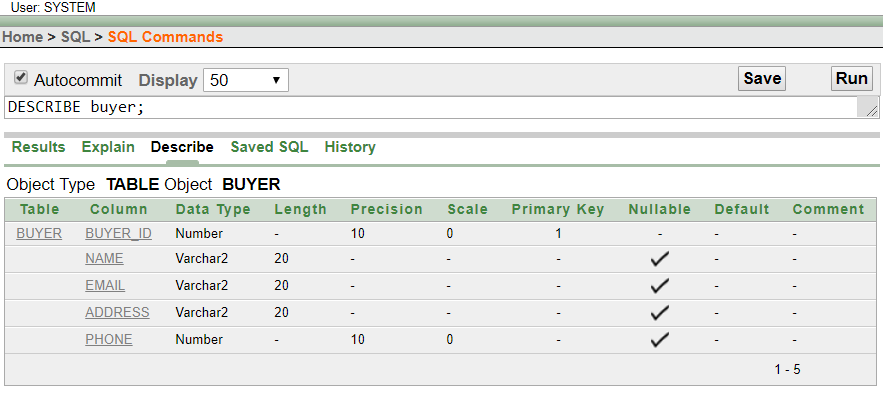
**14.** COMMERCIAL :



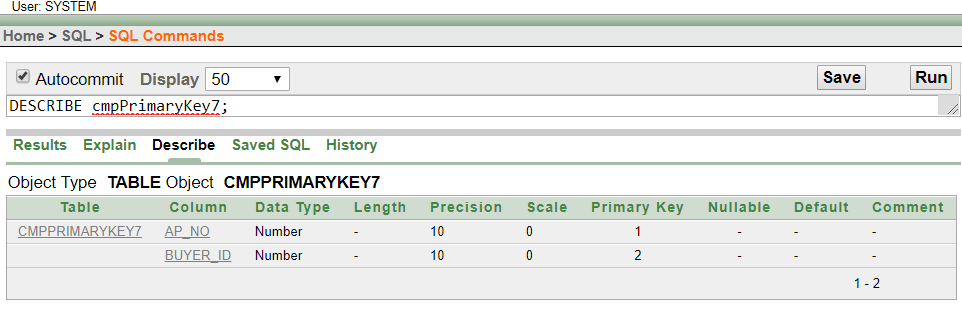
**15.** CMPPRIMARYKEY6 :



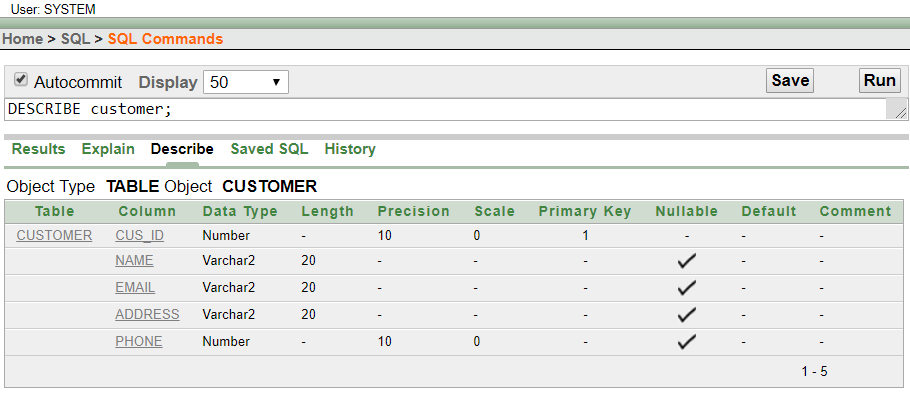
**16.** BUYER :



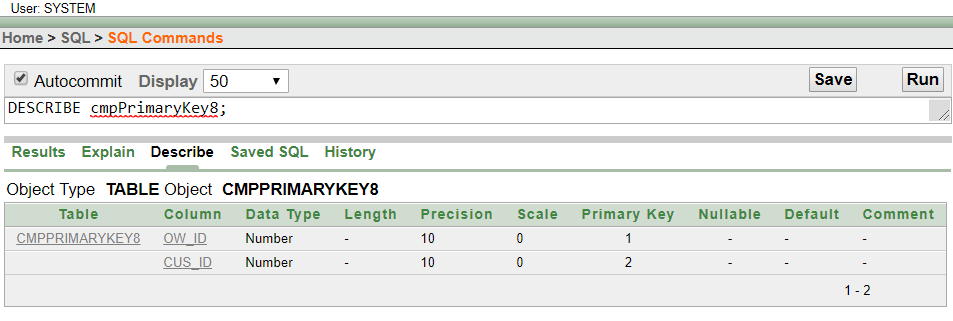
**17.** CMPPRIMARYKEY7 :



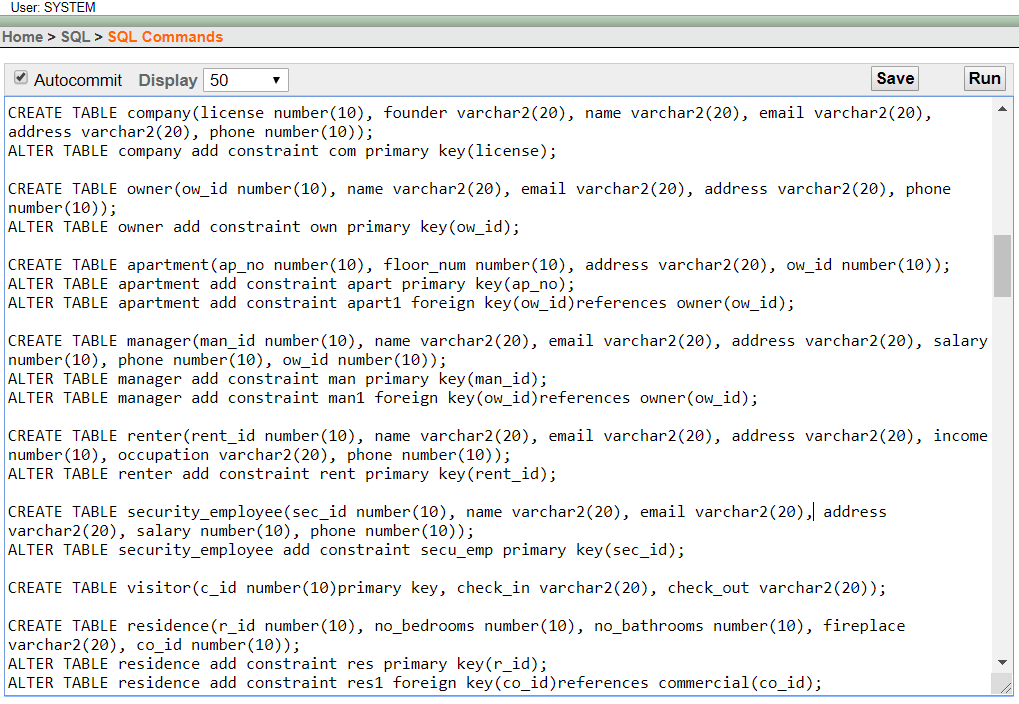
**18.** CUSTOMER :

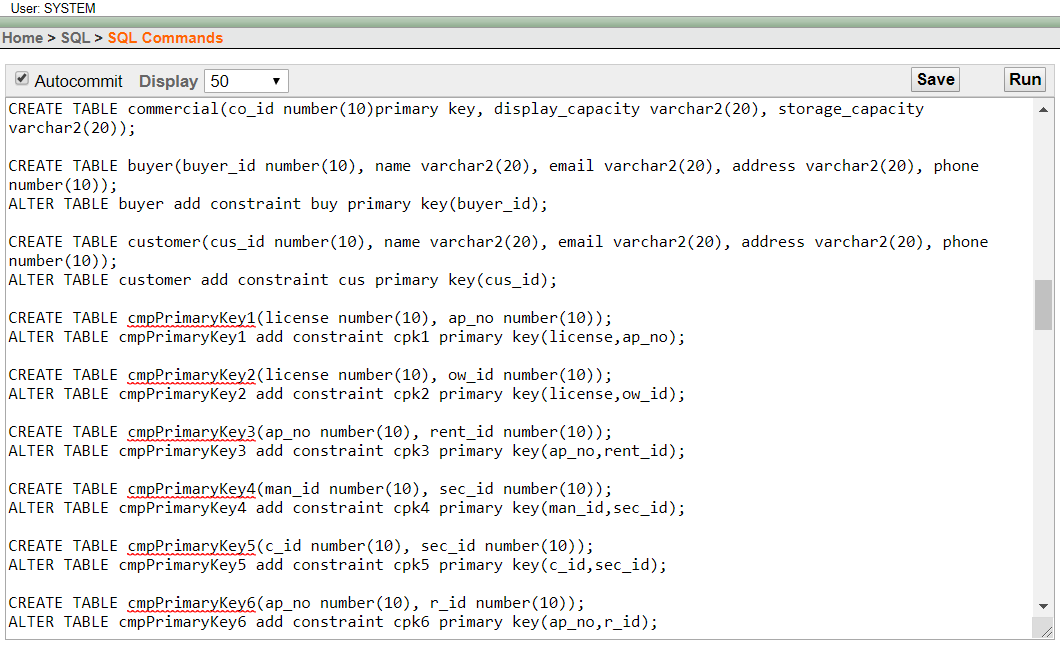


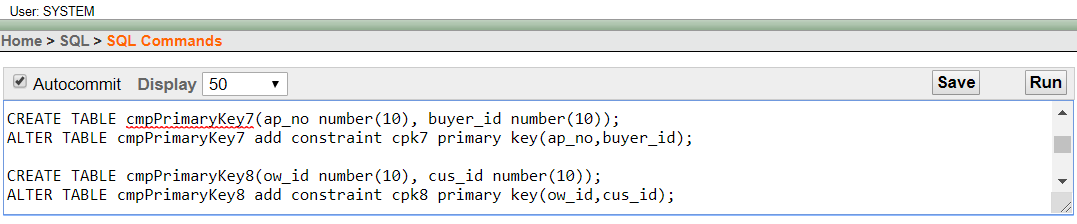
**19.** CMPPRIMARYKEY8 :



* **TABLE CREATION (SQL CODE)** :







* **DATA CONSTRAINTS** :

**1.** ALTER TABLE company add constraint com primary key(license);

**2.** ALTER TABLE owner add constraint own primary key(ow\_id);

**3.** ALTER TABLE apartment add constraint apart primary key(ap\_no);

**4.** ALTER TABLE apartment add constraint apart1 foreign key(ow\_id)references owner(ow\_id);

**5.** ALTER TABLE manager add constraint man primary key(man\_id);

**6.** ALTER TABLE manager add constraint man1 foreign key(ow\_id)references owner(ow\_id);

**7.** ALTER TABLE renter add constraint rent primary key(rent\_id);

**8.** ALTER TABLE security\_employee add constraint secu\_emp primary key(sec\_id);

**9.** ALTER TABLE residence add constraint res primary key(r\_id);

**10.** ALTER TABLE residence add constraint res1 foreign key(co\_id)references commercial(co\_id);

**11.** ALTER TABLE buyer add constraint buy primary key(buyer\_id);

1**2.** ALTER TABLE customer add constraint cus primary key(cus\_id);

**13.** ALTER TABLE cmpPrimaryKey1 add constraint cpk1 primary key(license,ap\_no);

**14.** ALTER TABLE cmpPrimaryKey2 add constraint cpk2 primary key(license,ow\_id);

**15.** ALTER TABLE cmpPrimaryKey3 add constraint cpk3 primary key(ap\_no,rent\_id);

**16.** ALTER TABLE cmpPrimaryKey4 add constraint cpk4 primary key(man\_id,sec\_id);

**17.** ALTER TABLE cmpPrimaryKey5 add constraint cpk5 primary key(c\_id,sec\_id);

**18.** ALTER TABLE cmpPrimaryKey6 add constraint cpk6 primary key(ap\_no,r\_id);

**19.** ALTER TABLE cmpPrimaryKey7 add constraint cpk7 primary key(ap\_no,buyer\_id);

**20.** ALTER TABLE cmpPrimaryKey8 add constraint cpk8 primary key(ow\_id,cus\_id);

* **DATA INSERTION** :

(1) COMPANY :

1. INSERT INTO company VALUES('123','A','DOM INNO','YAHOO','DHAKA','123654');
2. INSERT INTO company VALUES('234','B','NOGOR HOMES','GMAIL','CHITTAGONG','56987');
3. INSERT INTO company VALUES('345','C','APONGHOR','OUTLOOK','SYLHET','12369');
4. INSERT INTO company VALUES('456','D','BOSHUDHA','HOTMAIL','RAJSHAHI','789654');
5. INSERT INTO company VALUES('567','E','SAIMUM','ICLOUD','COMILLA','56321');

(2) OWNER :

1. INSERT INTO owner VALUES('54347','ABC','OUTLOOK','SHAHEENBAGH','5469789');
2. INSERT INTO owner VALUES('31234','BCD','GMAIL','WEST NAKHALPARA','125698');
3. INSERT INTO owner VALUES('53789','CDE','HOTMAIL','EAST NAKHALPARA','2314545');
4. INSERT INTO owner VALUES('34454','DEF','YAHOO','EAST TEJKUNIPARA','785369');
5. INSERT INTO owner VALUES('34666','EFG','ICLUOD','WEST TEJKUNIPARA','132132');

(3) APARTMENT :

1. INSERT INTO apartment VALUES('555','10','GULSHAN-1','54347');
2. INSERT INTO apartment VALUES('666','20','GULSHAN-2','31234');
3. INSERT INTO apartment VALUES('777','30','GULSHAN AVENUE','53789');
4. INSERT INTO apartment VALUES('888','40','BANANI','34454');
5. INSERT INTO apartment VALUES('999','50','DHANMONDI-1','34666');

(4) MANAGER :

1. INSERT INTO manager VALUES('934','AA','GMAIL','DHANMONDI-2','2000','123654','54347');
2. INSERT INTO manager VALUES('631','BB','ICLOUD','DHANMONDI-27','3000','101010','31234');
3. INSERT INTO manager VALUES('243','CC','OUTLOOK','DHANMONDI-9','4000','015236','53789');
4. INSERT INTO manager VALUES('389','DD','HOTMAIL','DHANMONDI-8','5000','01789','34454');
5. INSERT INTO manager VALUES('957','EE','YAHOO','DHANMONDI-7','6000','023698','34666');

(5) RENTER :

1. INSERT INTO renter VALUES('500','FF','HOTMAIL','SHONIR AKHRA','7000','BUSINESSMAN','1036025');
2. INSERT INTO renter VALUES('600','GG','ICLOUD','JATTRABARI','8000','EMPLOYEE','10230201');
3. INSERT INTO renter VALUES('700','HH','OUTLOOK','KACHPUR','9000','POLICE','23020230');
4. INSERT INTO renter VALUES('800','II','GMAIL','OLD DHAKA','10000','RAB','101020230');
5. INSERT INTO renter VALUES('100','JJ','YAHOO','NOTUN BAZAR','11000','BGB','809630');

(6) VISITOR :

1. INSERT INTO visitor VALUES('12','YES','YES');
2. INSERT INTO visitor VALUES('13','YES','NO');
3. INSERT INTO visitor VALUES('14','NO','YES');
4. INSERT INTO visitor VALUES('15','NO','NO');
5. INSERT INTO visitor VALUES('16','YES','YES');

(7) SECURITY\_EMPLOYEE :

1. INSERT INTO security\_employee VALUES('54','KK','ICLOUD','DHAKA CANTONMENT','12000','7080808');
2. INSERT INTO security\_employee VALUES('56','LL','OUTLOOK','MIRPUR CANTONMENT','13000','1569741');
3. INSERT INTO security\_employee VALUES('58','MM','HOTMAIL','BOGURA CANTONMENT','14000','1036690');
4. INSERT INTO security\_employee VALUES('48','NN','GMAIL','COMILLA CANTONMENT','15000','2013031');
5. INSERT INTO security\_employee VALUES('38','OO','YAHOO','RANGPUR CANTONMENT','16000','1036952');

(8) COMMERCIAL :

1. INSERT INTO commercial VALUES('31','NO','NO');
2. INSERT INTO commercial VALUES('32','NO','YES');
3. INSERT INTO commercial VALUES('33','YES','NO');
4. INSERT INTO commercial VALUES('34','YES','YES');
5. INSERT INTO commercial VALUES('35','NO','NO');

(9) RESIDENCE :

1. INSERT INTO residence VALUES('1000','2','3','YES','31');
2. INSERT INTO residence VALUES('1001','4','2','YES','32');
3. INSERT INTO residence VALUES('1010','3','2','NO','33');
4. INSERT INTO residence VALUES('1011','2','3','YES','34');
5. INSERT INTO residence VALUES('1100','2','1','NO','35');

(10) BUYER :

1. INSERT INTO buyer VALUES('510','PP','HOTMAIL','EAST KAFRUL','169874521');
2. INSERT INTO buyer VALUES('520','QQ','OUTLOOK','WEST KAFRUL','5632449');
3. INSERT INTO buyer VALUES('530','RR','GMAIL','NORTH KAFRUL','10232365');
4. INSERT INTO buyer VALUES('540','SS','YAHOO','SOUTH KAFRUL','596320');
5. INSERT INTO buyer VALUES('550','TT','ICLOUD','KAFRUL','02036985');

(11) CUSTOMER :

1. INSERT INTO customer VALUES('610','WW','HOTMAIL','JAMALPUR','741147');
2. INSERT INTO customer VALUES('620','XX','ICLOUD','NOAHKHALI','563256');
3. INSERT INTO customer VALUES('630','YY','OUTLOOK','COMILLA','225533');
4. INSERT INTO customer VALUES('640','ZZ','GMAIL','SHERPUR','896986');
5. INSERT INTO customer VALUES('650','DD','YAHOO','THAKURGAON','3696369');

(12) CMPPRIMARYKEY1 :

1. INSERT INTO cmpPrimaryKey1 VALUES('123','555');
2. INSERT INTO cmpPrimaryKey1 VALUES('234','666');
3. INSERT INTO cmpPrimaryKey1 VALUES('345','777');
4. INSERT INTO cmpPrimaryKey1 VALUES('456','888');
5. INSERT INTO cmpPrimaryKey1 VALUES('567','999');

(13) CMPPRIMARYKEY2 :

1. INSERT INTO cmpPrimaryKey2 VALUES('123','54347');
2. INSERT INTO cmpPrimaryKey2 VALUES('234','31234');
3. INSERT INTO cmpPrimaryKey2 VALUES('345','53789');
4. INSERT INTO cmpPrimaryKey2 VALUES('456','34454');
5. INSERT INTO cmpPrimaryKey2 VALUES('567','34666');

(14) CMPPRIMARYKEY3 :

1. INSERT INTO cmpPrimaryKey3 VALUES('555','500');
2. INSERT INTO cmpPrimaryKey3 VALUES('666','600');
3. INSERT INTO cmpPrimaryKey3 VALUES('777','700');
4. INSERT INTO cmpPrimaryKey3 VALUES('888','800');
5. INSERT INTO cmpPrimaryKey3 VALUES('999','100');

(15) CMPPRIMARYKEY4 :

1. INSERT INTO cmpPrimaryKey4 VALUES('934','54');
2. INSERT INTO cmpPrimaryKey4 VALUES('631','56');
3. INSERT INTO cmpPrimaryKey4 VALUES('243','58');
4. INSERT INTO cmpPrimaryKey4 VALUES('389','48');
5. INSERT INTO cmpPrimaryKey4 VALUES('957','38');

(16) CMPPRIMARYKEY5 :

1. INSERT INTO cmpPrimaryKey5 VALUES('12','54');
2. INSERT INTO cmpPrimaryKey5 VALUES('13','56');
3. INSERT INTO cmpPrimaryKey5 VALUES('14','58');
4. INSERT INTO cmpPrimaryKey5 VALUES('15','48');
5. INSERT INTO cmpPrimaryKey5 VALUES('16','38');

(17) CMPPRIMARYKEY6 :

1. INSERT INTO cmpPrimaryKey6 VALUES('555','1000');
2. INSERT INTO cmpPrimaryKey6 VALUES('666','1001');
3. INSERT INTO cmpPrimaryKey6 VALUES('777','1010');
4. INSERT INTO cmpPrimaryKey6 VALUES('888','1011');
5. INSERT INTO cmpPrimaryKey6 VALUES('999','1100');

(18) CMPPRIMARYKEY7 :

1. INSERT INTO cmpPrimaryKey7 VALUES('555','510');
2. INSERT INTO cmpPrimaryKey7 VALUES('666','520');
3. INSERT INTO cmpPrimaryKey7 VALUES('777','530');
4. INSERT INTO cmpPrimaryKey7 VALUES('888','540');
5. INSERT INTO cmpPrimaryKey7 VALUES('999','550');

(19) CMPPRIMARYKEY8 :

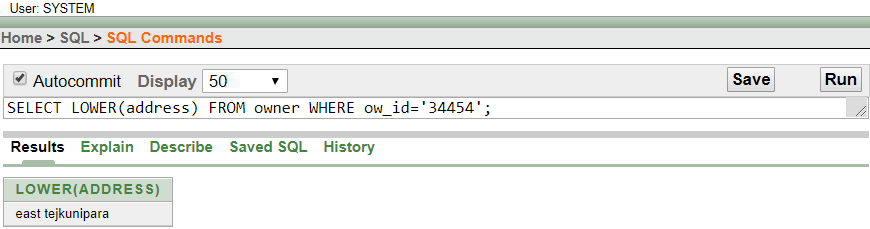
1. INSERT INTO cmpPrimaryKey8 VALUES('54347','610');
2. INSERT INTO cmpPrimaryKey8 VALUES('31234','620');
3. INSERT INTO cmpPrimaryKey8 VALUES('53789','630');
4. INSERT INTO cmpPrimaryKey8 VALUES('34454','640');
5. INSERT INTO cmpPrimaryKey8 VALUES('34666','650');

* **QUERY WRITING :**

**SINGLE ROW FUNCTIONS :**

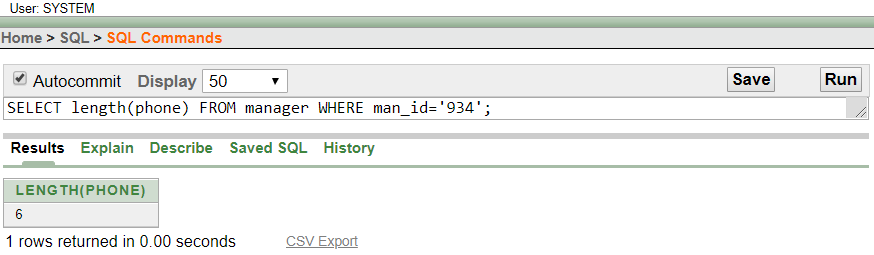
**QUES :** Display the address from owner table in lowercase letter where owner’s id is 34454.

**ANS :** SELECT LOWER(address) FROM owner WHERE ow\_id='34454';



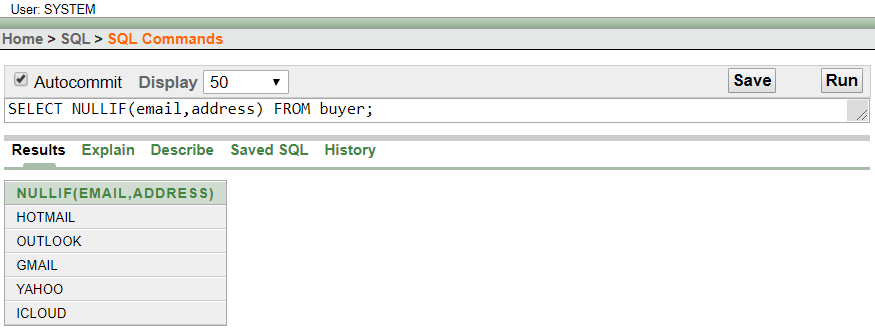
**QUES :** Calculate the length of phone no from manager table where Manager’s id is 934.

**ANS :** SELECT length(phone) FROM manager WHERE man\_id='934';



**QUES :** Check the values of email and address from buyer table. Using NULLIF function.

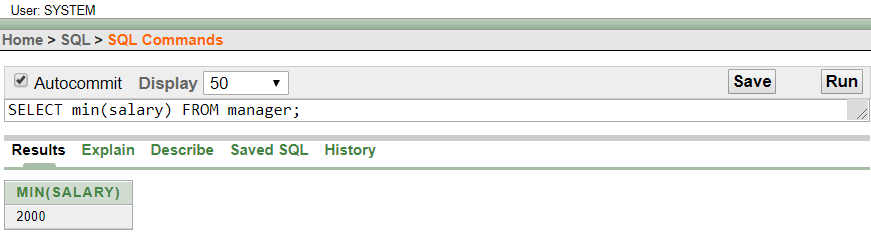
**ANS :** SELECT NULLIF(email,address) FROM buyer;



**GROUP FUNCTIONS :**

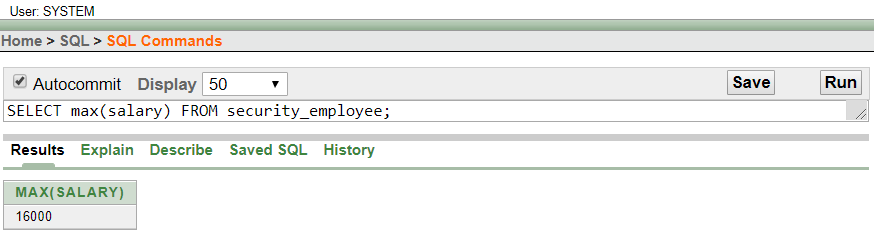
**QUES :** Display the minimum salary of manager from manager table.

**ANS :** SELECT min(salary) FROM manager;



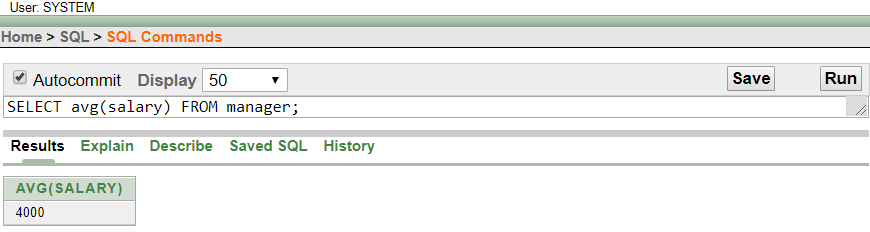
**QUES :** Display the maximum salary of Security Employee from security\_employee table.

**ANS :** SELECT max(salary) FROM security\_employee;



**QUES :** Display the average salary of manager from manager table.

**ANS :** SELECT avg(salary) FROM manager;



* **SUBQUERY :**

**QUES :** Whose manager’s salary is lower where manager ID is 957 ?

**ANS :** SELECT name

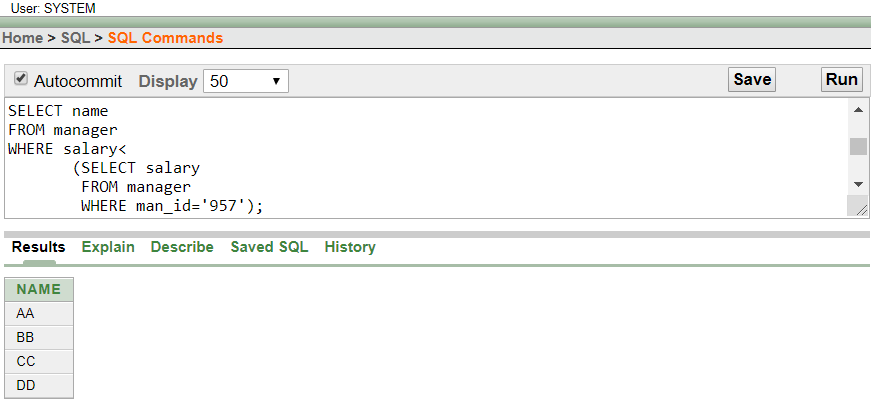
FROM manager

WHERE salary<

(SELECT salary

FROM manager

WHERE man\_id='957');



**QUES :** Who has a salary greater than LL ?

**ANS :** SELECT sec\_id,name

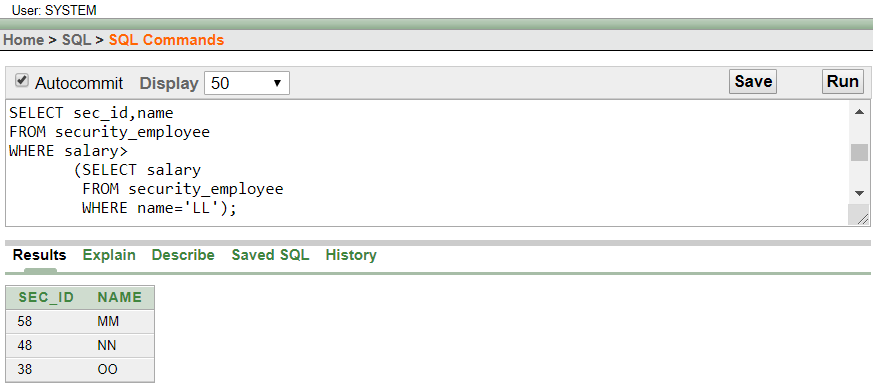
FROM security\_employee

WHERE salary>

(SELECT salary

FROM security\_employee

WHERE name='LL');



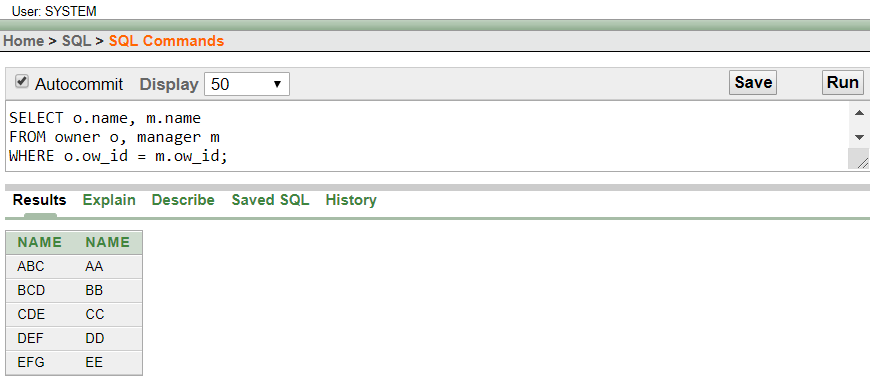
* **JOINING TABLES :**

**QUES :** Show the name of the both owner and manager tables.

**ANS :** SELECT o.name, m.name

FROM owner o, manager m

WHERE o.ow\_id = m.ow\_id;

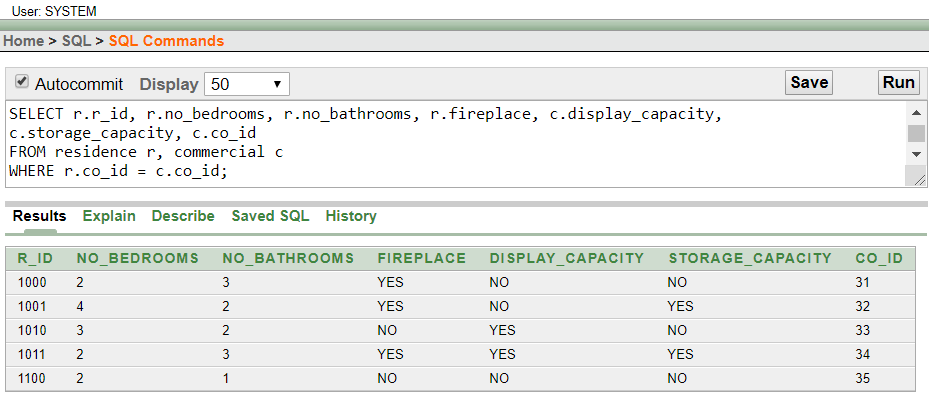


**QUES :** Show all the information of both residence and commercial tables.

**ANS :** SELECT r.r\_id, r.no\_bedrooms, r.no\_bathrooms, r.fireplace, c.display\_capacity, c.storage\_capacity, c.co\_id

FROM residence r, commercial c

WHERE r.co\_id = c.co\_id;



**CONCLUSION :**

After finished this project we have learned many things like how we can make database, making scenario, normalization, query and lot of this. We all have enjoyed.