****

**Apartment Management System**

|  |  |  |
| --- | --- | --- |
| Group-1 | | |
| **Name** | **ID** | **Contribution** |
| Sanuar Hossen | 19-39559-1 | Table Creation, Query Writing |
| Nafees Fuad Rahman | 19-39600-1 | Cover Page, ER Diagram, Relational Algebra |
| Md. Towhid Ahmed | 19-39601-1 | Data Insertion, Conclusion |
| Rabbi Hasan Himel | 20-42204-1 | Contents, Scenario Description, Normalization |
| Saifa Bintay Shahjahan | 20-42300-1 | Introduction, Schema Diagram |

**Introduction to Database**

**Section**-**E**

**Contents**

|  |  |
| --- | --- |
| **Report Contents** | **Page Number** |
| 1. Introduction | 02 |
| 1. Scenario Description | 03 |
| 1. ER Diagram | 04 |
| 1. Normalization | 05-08 |
| 1. Schema Diagram | 09 |
| 1. Table Creation | 10-12 |
| 1. Data Insertion | 13-14 |
| 1. Query Writing | 15 |
| 1. Relational Algebra | 16 |
| 1. Conclusion | 16 |

**Introduction**

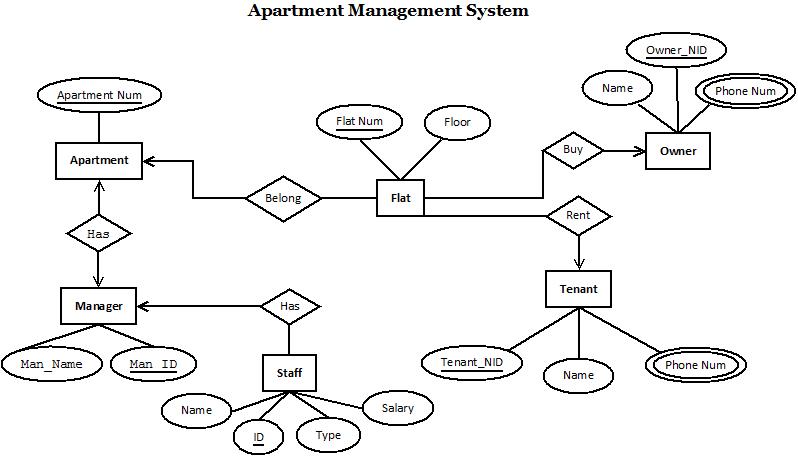
The project entitled "APARTMENT MANAGEMENT SYSTEM" is developed for complete management for the apartment related details. This project helps to keep track of daily transaction in electronic manner which saves a lot of recourses like energy, time and money. This ER diagram of Apartment Management System shows all the relations between each of the two. It uses structure data to define the relationships between data groups and system functionalities. The main entities of the system are apartment, flat, staff, tenant, owner and garage. In the system, the details of the individual records of the residents are entered separately and the expenses that has to pay by residents are entered a calculated automatically with fraction of second. The monthly expenses are calculated automatically for each resident which can save time and less employment.

Once the project is developed and implemented we will be able to keep track of the increasing resident's database. This database will help them to find out the regarding details of the available entities.

**Scenario Description of Apartment Management System**

In an apartment management system, every apartment has an apartment number. An apartment has a manager. A manager belongs to an apartment. The name and manager ID are stored in the system. An apartment has many flats. A flat always belongs to an apartment. The owner can buy one or many flats. A flat is bought by exactly one owner. The flat details are stored like a flat number & floor number. The owner is identified by National Identity Number. Besides, the system also stores the owner’s name and phone number. A tenant can rent one or more flats. But a flat is rented by exactly one client. The tenant’s details, such as NID number, phone number & Name are stored there. A manager has staff under his supervision. A staff works for a manager. To identify the staff, the system stores their Name, ID, Type & Salary.

**ER Diagram of Apartment Management System**



**Normalization**

**RENT:**

UNF: RENT (flat\_num, floor, Tenant\_NID, Name, Phone\_Num)

1NF: There is a multi-valued attribute which is Phone\_Num.

1. Flat\_num, floor, Tenant\_NID, Name, Phone\_Num

2NF:

1. Flat\_num, floor

2. Tenant\_NID, Name, Phone\_Num

3NF:

1. Flat\_num, floor

2. Tenant\_NID, Name, Phone\_Num

Table Creation:

1. Flat\_num, floor

2. Tenant\_NID, Name, Phone\_Num, **Flat\_num**

**Buy:**

UNF: Buy (Name, Owner\_NID, Phone\_Num, flat\_num, floor)

1NF: There is a multi-valued attribute which is Phone\_Num.

1. Flat\_num, floor, Owner\_NID, Name, Phone\_Num

2NF:

1. Flat\_num, floor,

2. Owner\_NID, Name, Phone\_Num

3NF:

1. Flat\_num, floor

2. Owner\_NID, Name, Phone\_Num

Table Creation:

1. Flat\_num, floor

2. Owner\_NID, Name, Phone\_Num, **Flat\_num**

**Belong:**

UNF: Belong (flat\_num, floor, A\_Num)

1NF:

1. Flat\_num, floor, A\_Num

2NF:

1. A\_Num

2. Flat\_num, floor

3NF:

1. Flat\_num, floor

2. A\_Num

Table Creation:

1. Flat\_num, floor, **A\_Num**

2. A\_Num

**Has: (Down)**

UNF: Has (Name, ID, Type, Salary, Man\_Name, Man\_ID)

1NF:

1. Name, ID, Type, Salary, Man\_Name, Man\_ID

2NF:

1. Man\_Name, Man\_ID

2. Name, ID, Type, Salary

3NF:

1. Man\_Name, Man\_ID

2. Name, ID, Type, Salary

Table Creation:

1. Man\_Name, Man\_ID

2. Name, ID, Type, Salary, **Man\_ID**

**Has: (Up)**

UNF: Has (A\_Num, Man\_Name, Man\_ID)

1NF:

1. A\_Num, Man\_Name, Man\_ID

2NF:

1. Man\_Name, Man\_ID

2. A\_Num

3NF:

1. Man\_Name, Man\_ID

2. A\_Num

Table Creation:

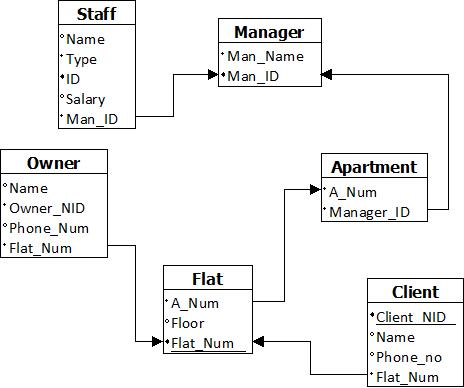
1. Man\_Name, Man\_ID

2. A\_Num, **Man\_ID**

**Final Tables:**

1. Tenant\_NID, Name, Phone\_Num, **Flat\_num**
2. Owner\_NID, Name, Phone\_Num, **Flat\_num**
3. Flat\_num, floor, **A\_Num**
4. Man\_Name, Man\_ID
5. Name, ID, Type, Salary, **Man\_ID**
6. A\_Num, **Man\_ID**

**Schema Diagram**



**Table Creation**

Create table T1 (Tenant\_NID number (18) primary key, Name Varchar2 (25), phone\_num number (11), flat\_Num Number (6));

Create table T2 (Owner\_NID number (18) primary key, Name Varchar2 (25), phone\_num number (11), flat\_Num Number (6));

Create table T3 (flat\_Num Number (6) primary key, floor number (2), A\_num number (5));

Create table T4 (Man\_Name Varchar2 (25), Man\_ID number (5) primary key);

Create table T5 (Name Varchar2 (25), ID number (5) primary key, Type varchar2 (15), Salary Number (5), Man\_ID number (5));

Create table T6 (A\_Num number (5) primary key, Man\_ID number (5));

**ADDING CONSTRAINT:**

Alter table T1 add constraint Ct1 foreign key (Flat\_num) references T3 (flat\_Num);

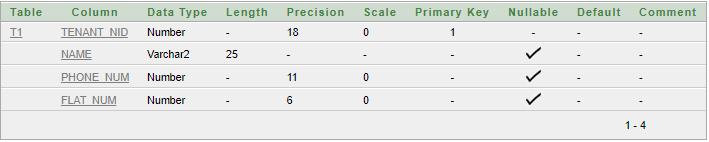
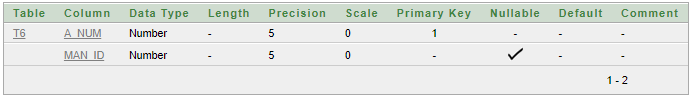
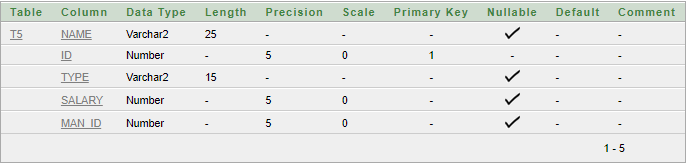
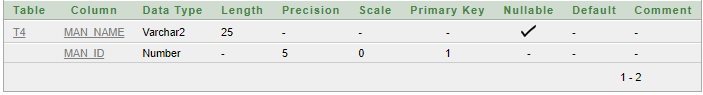
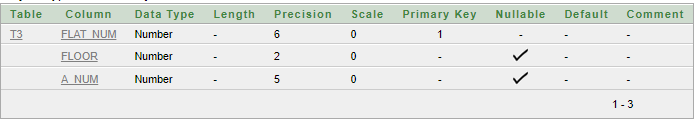
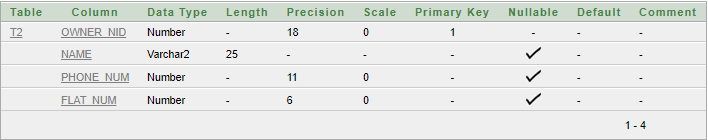
Alter table T2 add constraint Ct2 foreign key (Flat\_num) references T3 (flat\_Num);

Alter table T3 add constraint Ct3 foreign key (A\_Num) references T6 (A\_Num);

Alter table T5 add constraint Ct4 foreign key (Man\_ID) references T4 (Man\_ID);

Alter table T6 add constraint Ct5 foreign key (Man\_ID) references T4 (Man\_ID);

**SCREENSHOTS:**

****

**SEQUENCE:**

CREATE SEQUENCE T4\_Man\_ID INCREMENT BY 1 START WITH 10001 NOCACHE NOCYCLE;

**CREATING USER:**

Create user Manager\_Jabed identified by kite9046

GRANT UNLIMITED TABLESPACE TO MANAGER\_JABED

CREATE ROLE Manager;

Grant create table, create view, create sequence to Manager;

GRANT Manager to Manager\_Jabed;

**Data Insertion**

Insert into T1 values(98638272, 'SHUVO', 01711542556, 601);

Insert into T1 values(98656279, 'TUHIN', 01837542556, 303);

Insert into T1 values(98837921, 'HIMEL', 01414657556, 509);

Insert into T1 values(98632738, 'SAIFA', 01917482959, 202);

Insert into T1 values(98163084, 'NAFEES', 01683272924, 405);

Insert into T2 values(98632750, 'HOSSEN', 01482955568, 304);

Insert into T2 values(98651458, 'AHMED', 01937142556, 502);

Insert into T2 values(98830149, 'RABBI', 01502537590, 606);

Insert into T2 values(98630834, 'SHAHJAHAN', 01617488361, 207);

Insert into T2 values(98161272, 'FUAD', 01789344829, 401);

Insert into T3 values(303, 3, 1);

Insert into T3 values(509, 5, 2);

Insert into T3 values(601, 6, 3);

Insert into T3 values(202, 2, 4);

Insert into T3 values(405, 4, 5);

Insert into T3 values(304, 3, 1);

Insert into T3 values(502, 5, 2);

Insert into T3 values(606, 6, 3);

Insert into T3 values(207, 2, 4);

Insert into T3 values(401, 4, 5);

Insert into T4 (Man\_Name, Man\_ID) values ('ABUL', T4\_Man\_ID.NEXTVAL);

Insert into T4 (Man\_Name, Man\_ID) values ('KALAM', T4\_Man\_ID.NEXTVAL);

Insert into T4 (Man\_Name, Man\_ID) values ('AZAD', T4\_Man\_ID.NEXTVAL);

Insert into T4 (Man\_Name, Man\_ID) values ('MUSTAFA', T4\_Man\_ID.NEXTVAL);

Insert into T4 (Man\_Name, Man\_ID) values ('HUDA', T4\_Man\_ID.NEXTVAL);

Insert into T5 values('LITON', 77771, 'GUARD', 6500, 10001);

Insert into T5 values('MITHUN', 77772, 'CLEANER', 6000, 10002);

Insert into T5 values('SHANTO', 77773, 'GUARD', 6700, 10003);

Insert into T5 values('SUJON', 77774, 'CLEANER', 5700, 10004);

Insert into T5 values('MOSADDEK', 77775, 'GUARD', 6200, 10005);

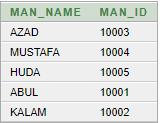
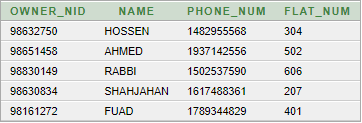
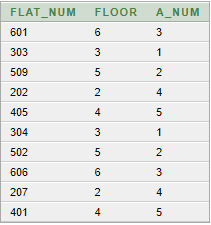
Insert into T6 values(1, 10001);

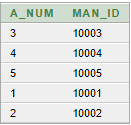
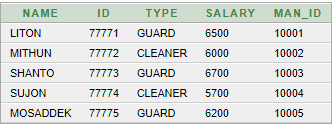
Insert into T6 values(2, 10002);

Insert into T6 values(3, 10003);

Insert into T6 values(4, 10004);

Insert into T6 values(5, 10005);





**Query Writing**

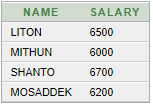
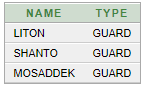
***Sub query:***

- Display the employee names who earn more than employee SUJON

select Name, Salary from t5 where salary>(select salary from t5 where name='SUJON');

- Display the employee names who have the same job as employee JONES

Select name, type from t5 where type=(select type from t5 where name= 'SHANTO');

***JOINING:***

- Display name & flat number where Tenant name is “HIMEL”

select t1.name, t1.flat\_num from t1 where t1.name='HIMEL'

- Display the staffs names and types of only those staff whose salary is greater than 4000.

SELECT t5.name, t5.type FROM t5 WHERE t5.salary>4000

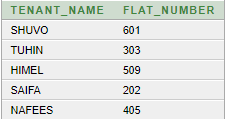
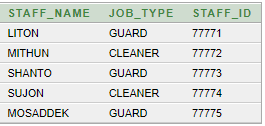
***View:***

- Create a view called Tenant\_List based on the Name and Flat Number

CREATE VIEW TENANT\_LIST AS SELECT Name TENANT\_NAME , Flat\_num FLAT\_NUMBER FROM t1 ;

- Create a view called STAFF\_List based on the Name, TYPE, ID

CREATE VIEW STAFF\_LIST AS SELECT Name STAFF\_NAME , Type JOB\_TYPE, ID STAFF\_ID FROM t5 ;

**Relational Algebra**

1. Find the name of the staff whose salary is greater than 6000.
2. Find the name of the owner of flat number 509.
3. Find the manager ID from the apartment number 4.
4. Find the ID of manager Huda.
5. Find the floor number of flat numbers 207.

Solutions:

1. ∏*Name* (σ*salary* > 6000 (*T5*))

2. ∏*Name* (σ*flat\_num=509* (*T1*))

3. ∏*Man\_ID* (σ*A\_Num=4* (*T6*))

4. ∏*Man\_ID* (σ*Name=’HUDA’* (*T5*))

5. ∏*Floor* (σ*flat\_num=207* (*T5*))

**Conclusion**

The project's future purpose is to assist us in developing a high-level database system. This project can serve as an example of our abilities. This project can be used as a sample by the client, and if they like it, they can offer us a new project. The Apartment Management Database System will make it easier for the client to manage a large apartment. This system enables them to change and delete information for the entire apartment building in a matter of seconds and is very simple to use. Our future goal for this project is to upgrade it to a higher level. This system will aid us in incorporating this database system into the software. Our ultimate goal is to sell it in the market as a commercial product, which will play an important role in our company's success.