

**AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)**

**FACULTY OF SCIENCE & TECHNOLOGY**

**DEPARTMENT OF COMPUTER SCIENCE**

**INTRODUCTION TO DATABASE**

**Fall 2022-2023**

**Section: A**

**PROJECT ON**

Apartment Management System

**Supervised By**

*Juena Ahmed Noshin*

|  |  |  |  |
| --- | --- | --- | --- |
| Name | | ID | Contribution |
| Iffat Shamia Shairy | 22-46703-1 | | Normalization, Schema Diagram, Conclusion |
| Ajmy Alaly | 22-46733-1 | | Introduction, Scenario Description, ER Diagram |
| Faysal Ahmmed | 22-47069-1 | | Data Insert, Query Writing |
| Md. Asifur Rahman Sayem | 22-47092-1 | | Table Creation, Relational Algebra |

**Submitted By**

Date of Submission: **December 26, 2022**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **TOPICS** | | ***Page no*.** |
| 1. **Title Page** | | 1 |
| 1. **Table of Content** | | 2 |
| 1. **Introduction** | | 3 |
| 1. **Scenario Description** | | 3 |
| 1. **ER-Diagram** | | 4 |
| 1. **Normalization** | | 5 |
| 1. **Schema Diagram** | | 12 |
| 1. **Table Creation** | | 13 |
| 1. **Data Insert** | | 16 |
| 1. **Query Writing** | | 20 |
| 1. **Relational Algebra** | | 22 |
| 1. **Conclusion** | | 23 |
|  |

**Introduction:**

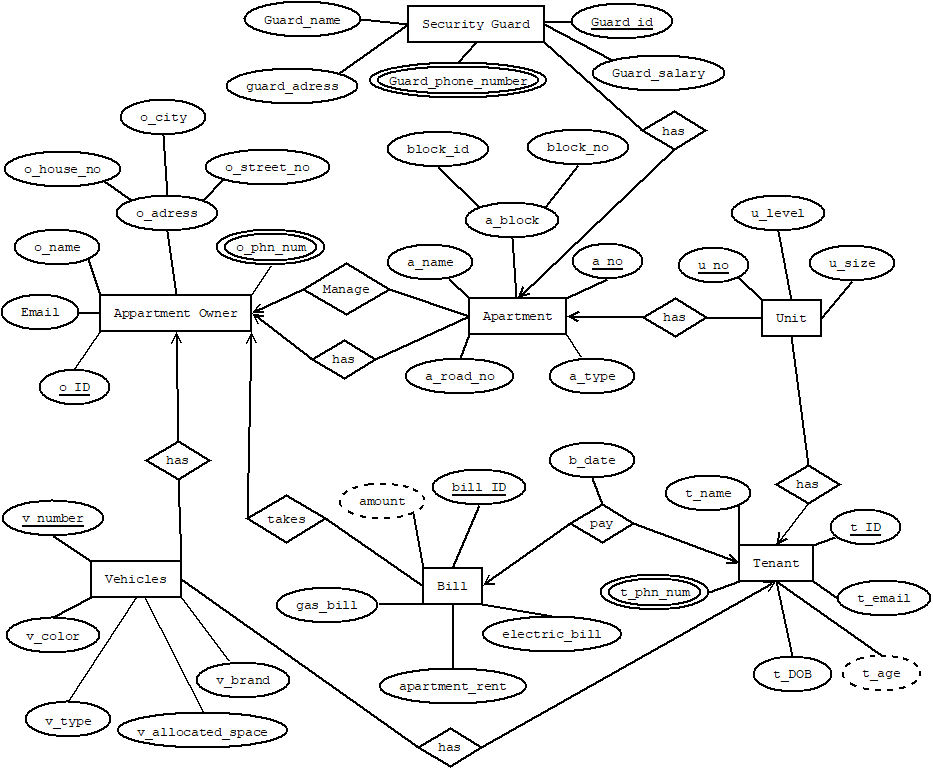
To manage the tasks and information of apartment or housing societies we require a software application which is known as an apartment management system. These systems collect all the data so that the tenants and the owner of a property can enjoy smooth processes for renting collection, maintenance, security, etc.

Our apartment management system also works in the same way. It collects data on every tenant who is living in the apartment. It collects information about the tenant’s maintenance bill, their vehicle details, their unit number in the apartment, etc. All these things can help apartment owners manage and maintain their apartments accurately and efficiently.

**Scenario Description:**

In an apartment management system, an apartment owner may have many apartments. But an apartment has only one apartment owner. An apartment is identified by its apartment number. The system also stores apartment name, apartment type, road number, and block. A block is composed of a block id and a block number. An apartment owner is identified by the apartment owner ID. Owner address, name, email, and phone number are also stored. An owner's address is composed of the house number, street name, and city. There may be multiple phone numbers for the owner. An apartment may have at least one unit, but a unit can only have one apartment. To identify the unit, the unit number is stored. Also, unit size and unit level are stored. A unit has a tenant. To uniquely identify tenants, their ID is stored. Tenants' name, phone number, email, date of birth, and age are stored. There may be multiple phone numbers for the tenant. Tenants' ages can be determined from their dates of birth. A tenant has to pay bills. A tenant can pay bills once a month, and a bill can be paid by only one tenant. To distinguish bills, bill id is stored. A bill consists of an electric bill, a gas bill, apartment rent, and a total amount. The total amount can be determined by adding all the bills. When a tenant pays the bill, the date is stored. The apartment owner takes the bill. An apartment owner can take many bills, but a bill can be taken only once by an apartment owner. An apartment owner may have vehicles, but a vehicle can only have one apartment owner. To identify vehicles, their number is stores. Also, vehicle brand, color, and type are stored along with the vehicle's allocation space to give the vehicle proper space in apartment parking. A tenant may also have vehicles. But a vehicle can have only one tenant. An apartment owner can manage many apartments, but an apartment can be managed by only one apartment owner. An apartment also has security guards to protect the apartment and its tenants. To identify guards differently, their ID is stored in the system. Also stored are their name, address, phone number, and salary. A security guard may have multiple phone numbers. An apartment can have many security guards, but a security guard can only give his or her service to one apartment at a time.

**ER Diagram:**



**Normalization:**

UNF:

Manage (o\_id, email, o\_name, o\_house\_no, o\_city, o\_street\_no, o\_phone\_number, a\_no, a\_name, block\_id, block\_no, a\_type, a\_road\_no)

1NF:

O\_phone\_number is a multivalued attribute.

1. o\_id, email, o\_name, o\_house\_no, o\_city, o\_street\_no, o\_phone\_number, a\_no, a\_name, block\_id, block\_no, a\_type, a\_road\_no

2NF:

1. o\_id, email, o\_name, o\_house\_no, o\_city, o\_street\_no, o\_phone\_number

2. a\_no, a\_name, block\_id, block\_no, a\_type, a\_road\_no

3NF:

There is no transitive dependency. Relation already in 3NF.

1. o\_id, email, o\_name, o\_house\_no, o\_city, o\_street\_no, o\_phone\_number

2. a\_no, a\_name, block\_id, block\_no, a\_type, a\_road\_no

Table Creation:

1. o\_id, email, o\_name, o\_house\_no, o\_city, o\_street\_no, o\_phone\_number

2. a\_no, a\_name, block\_id, block\_no, a\_type, a\_road\_no, **o\_id**

UNF:

Has (o\_id, email, o\_name, o\_house\_no, o\_city, o\_street\_no, o\_phone\_number, a\_no, a\_name, block\_id, block\_no, a\_type, a\_road\_no)

1NF:

O\_phone\_number is a multivalued attribute.

1. o\_id, email, o\_name, o\_house\_no, o\_city, o\_street\_no, o\_phone\_number, a\_no, a\_name, block\_id, block\_no, a\_type, a\_road\_no

2NF:

1. o\_id, email, o\_name, o\_house\_no, o\_city, o\_street\_no, o\_phone\_number

2. a\_no, a\_name, block\_id, block\_no, a\_type, a\_road\_no

3NF:

There is no transitive dependency. Relation already in 3NF.

1. o\_id, email, o\_name, o\_house\_no, o\_city, o\_street\_no, o\_phone\_number

2. a\_no, a\_name, block\_id, block\_no, a\_type, a\_road\_no

Table Creation:

1. o\_id, email, o\_name, o\_house\_no, o\_city, o\_street\_no, o\_phone\_number

2. a\_no, a\_name, block\_id, block\_no, a\_type, a\_road\_no, **o\_id**

UNF:

Has (a\_no, a\_name, block\_id, block\_no, a\_type, a\_road\_no, u\_no, u\_size, u\_level)

1NF:

There is no multivalued attribute. Relation already in 1NF.

1. a\_no, a\_name, block\_id, block\_no, a\_type, a\_road\_no, u\_no, u\_size, u\_level

2NF:

1. a\_no, a\_name, block\_id, block\_no, a\_type, a\_road\_no

2. u\_no, u\_size, u\_level

3NF:

There is no transitive dependency. Relation already in 3NF.

1. a\_no, a\_name, block\_id, block\_no, a\_type, a\_road\_no

2. u\_no, u\_size, u\_level

Table Creation:

1. a\_no, a\_name, block\_id, block\_no, a\_type, a\_road\_no

2. u\_no, u\_size, u\_level, **a\_no**

UNF:

Has (a\_no, a\_name, block\_id, block\_no, a\_type, a\_road\_no, guard\_id, guard\_salary, guard\_phone\_number, guard\_adress, guard\_name)

1NF:

Guard\_phone\_number is a multivalued attribute.

1. a\_no, a\_name, block\_id, block\_no, a\_type, a\_road\_no, guard\_id, guard\_salary, guard\_phone\_number, guard\_adress, guard\_name

2NF:

1. a\_no, a\_name, block\_id, block\_no, a\_type, a\_road\_no

2. guard\_id, guard\_salary, guard\_phone\_number, guard\_adress, guard\_name

3NF:

There is no transitive dependency. Relation already in 3NF.

1. a\_no, a\_name, block\_id, block\_no, a\_type, a\_road\_no

2. guard\_id, guard\_salary, guard\_phone\_number, guard\_adress, guard\_name

Table Creation:

1. a\_no, a\_name, block\_id, block\_no, a\_type, a\_road\_no

2. guard\_id, guard\_salary, guard\_phone\_number, guard\_adress, guard\_name, **a\_no**

UNF

Has(u\_no, u\_size, u\_level, t\_id, t\_email, t\_DOB, t\_name, t\_phone\_number)

1NF:

T\_phone\_number is a multivalued attribute.

1. u\_no, u\_size, u\_level, t\_id, t\_email, t\_DOB, t\_name, t\_phone\_number

2NF:

1. u\_no, u\_size, u\_level

2. t\_id, t\_email, t\_DOB, t\_name, t\_phone\_number

3NF:

There is no transitive dependency. Relation already in 3NF.

1. u\_no, u\_size, u\_level

2. t\_id, t\_email, t\_DOB, t\_name, t\_phone\_number

Table Creation:

1. u\_no, u\_size, u\_level, **t\_id**

2. t\_id, t\_email, t\_DOB, t\_name, t\_phone\_number

UNF:

Pay(t\_id, t\_email, t\_DOB, t\_name, t\_phone\_number, bill\_id, electric\_bill, apartment\_rent, gas\_bill, b\_date)

1NF:

T\_phone\_number is a multivalued attribute.

1. t\_id, t\_email, t\_DOB, t\_name, t\_phone\_number, bill\_id, electric\_bill, apartment\_rent, gas\_bill, b\_date

2NF:

1. t\_id, t\_email, t\_DOB, t\_name, t\_phone\_number

2. bill\_id, electric\_bill, apartment\_rent, gas\_bill, b\_date

3NF:

There is no transitive dependency. Relation already in 3NF.

1. t\_id, t\_email, t\_DOB, t\_name, t\_phone\_number

2. bill\_id, electric\_bill, apartment\_rent, gas\_bill, b\_date

Table Creation:

1. t\_id, t\_email, t\_DOB, t\_name, t\_phone\_number

2. bill\_id, electric\_bill, apartment\_rent, gas\_bill, b\_date, **t\_id**

UNF:

Takes(. bill\_id, electric\_bill, apartment\_rent, gas\_bill, b\_date, o\_id, email, o\_name, o\_house\_no, o\_city, o\_street\_no, o\_phone\_number)

1NF:

O\_phone\_number is a multivalued attribute.

1. bill\_id, electric\_bill, apartment\_rent, gas\_bill, b\_date, o\_id, email, o\_name, o\_house\_no, o\_city, o\_street\_no, o\_phone\_number

2NF:

1. bill\_id, electric\_bill, apartment\_rent, gas\_bill, b\_date

2. o\_id, email, o\_name, o\_house\_no, o\_city, o\_street\_no, o\_phone\_number

3NF:

There is no transitive dependency. Relation already in 3NF.

1. bill\_id, electric\_bill, apartment\_rent, gas\_bill, b\_date

2. o\_id, email, o\_name, o\_house\_no, o\_city, o\_street\_no, o\_phone\_number

Table Creation:

1. bill\_id, electric\_bill, apartment\_rent, gas\_bill, b\_date, **o\_id**

2. o\_id, email, o\_name, o\_house\_no, o\_city, o\_street\_no, o\_phone\_number

UNF:

Has(v\_number, v\_color, v\_brand, v\_type, v\_allocated\_space, o\_id, email, o\_name, o\_house\_no, o\_city, o\_street\_no, o\_phone\_number)

1NF:

O\_phone\_number is a multivalued attribute.

1. v\_number, v\_color, v\_brand, v\_type, v\_allocated\_space, o\_id, email, o\_name, o\_house\_no, o\_city, o\_street\_no, o\_phone\_number

2NF:

1. v\_number, v\_color, v\_brand, v\_type, v\_allocated\_space

2. o\_id, email, o\_name, o\_house\_no, o\_city, o\_street\_no, o\_phone\_number

3NF:

1. o\_id, email, o\_name, o\_house\_no, o\_city, o\_street\_no, o\_phone\_number

2. v\_number, v\_color, v\_brand

3. v\_id, v\_type, v\_allocated\_space

Table Creation:

1. o\_id, email, o\_name, o\_house\_no, o\_city, o\_street\_no, o\_phone\_number

2. v\_number, v\_color, v\_brand, **v\_id, o\_id**

3. v\_id, v\_type, v\_allocated\_space

UNF:

Has(v\_number, v\_color, v\_brand, v\_type, v\_allocated\_space, t\_id, t\_email, t\_DOB, t\_name, t\_phone\_number)

1NF:

T\_phone\_number is a multivalued attribute.

1. v\_number, v\_color, v\_brand, v\_type, v\_allocated\_space, t\_id, t\_email, t\_DOB, t\_name, t\_phone\_number

2NF:

1. v\_number, v\_color, v\_brand, v\_type, v\_allocated\_space

2. t\_id, t\_email, t\_DOB, t\_name, t\_phone\_number

3NF:

1. v\_number, v\_color, v\_brand

2. v\_id, v\_type, v\_allocated\_space

3. t\_id, t\_email, t\_DOB, t\_name, t\_phone\_number

Table Creation:

1. v\_number, v\_color, v\_brand, **v\_id, t\_id**

2. v\_id, v\_type, v\_allocated\_space

3. t\_id, t\_email, t\_DOB, t\_name, t\_phone\_number

**Temporary Tables:**

1. o\_id, email, o\_name, o\_house\_no, o\_city, o\_street\_no, o\_phone\_number

2. a\_no, a\_name, block\_id, block\_no, a\_type, a\_road\_no, **o\_id**

3. ~~o\_id, email, o\_name, o\_house\_no, o\_city, o\_street\_no, o\_phone\_number~~

4. ~~a\_no, a\_name, block\_id, block\_no, a\_type, a\_road\_no,~~ **~~o\_id~~**

5. ~~a\_no, a\_name, block\_id, block\_no, a\_type, a\_road\_no~~

6. u\_no, u\_size, u\_level, **a\_no**

7. ~~a\_no, a\_name, block\_id, block\_no, a\_type, a\_road\_no~~

8. guard\_id, guard\_salary, guard\_phone\_number, guard\_adress, guard\_name, **a\_no**

9. u\_no, u\_size, u\_level, **t\_id**

10. t\_id, t\_email, t\_DOB, t\_name, t\_phone\_number

11. ~~t\_id, t\_email, t\_DOB, t\_name, t\_phone\_number~~

12. bill\_id, electric\_bill, apartment\_rent, gas\_bill, b\_date, **t\_id**

13. bill\_id, electric\_bill, apartment\_rent, gas\_bill, b\_date, **o\_id**

14. ~~o\_id, email, o\_name, o\_house\_no, o\_city, o\_street\_no, o\_phone\_number~~

15. ~~o\_id, email, o\_name, o\_house\_no, o\_city, o\_street\_no, o\_phone\_number~~

16. v\_number, v\_color, v\_brand, **v\_id, o\_id**

17. v\_id, v\_type, v\_allocated\_space

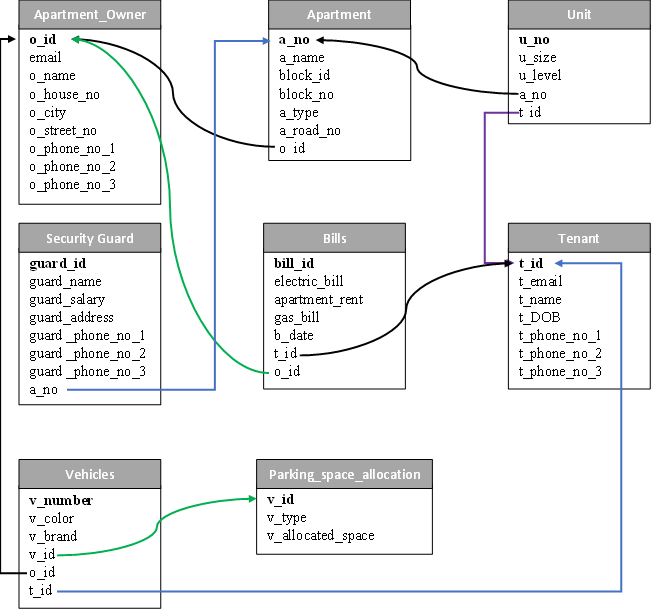
18. v\_number, v\_color, v\_brand, **v\_id, t\_id**

19. ~~v\_id, v\_type, v\_allocated\_space~~

20. ~~t\_id, t\_email, t\_DOB, t\_name, t\_phone\_number~~

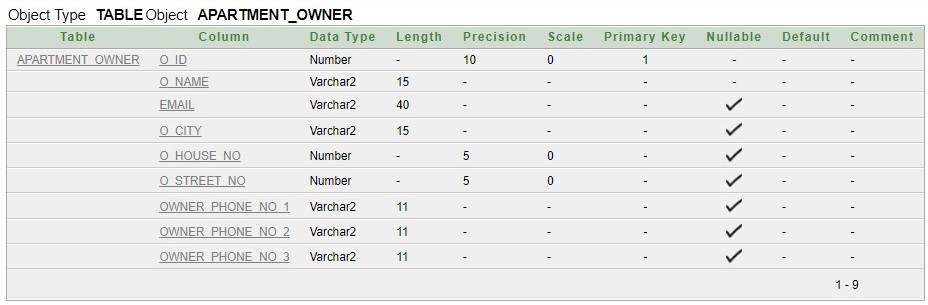
**Final Tables:**

1. o\_id, email, o\_name, o\_house\_no, o\_city, o\_street\_no, o\_phone\_number\_1, o\_phone\_number\_2, o\_phone\_number\_3
2. a\_no, a\_name, block\_id, block\_no, a\_type, a\_road\_no, **o\_id**
3. u\_no, u\_size, u\_level, **a\_no, t\_id**
4. guard\_id, guard\_salary, guard\_adress, guard\_name, guard\_phone\_number\_1, guard\_phone\_number\_2, guard\_phone\_number\_3, **a\_no**
5. t\_id, t\_email, t\_DOB, t\_name, t\_phone\_number\_1, t\_phone\_number\_2, t\_phone\_number\_3
6. bill\_id, electric\_bill, apartment\_rent, gas\_bill, b\_date, **t\_id, o\_id**
7. v\_number, v\_color, v\_brand, **v\_id, o\_id, t\_id**
8. v\_id, v\_type, v\_allocated\_space

**Schema Diagram:**

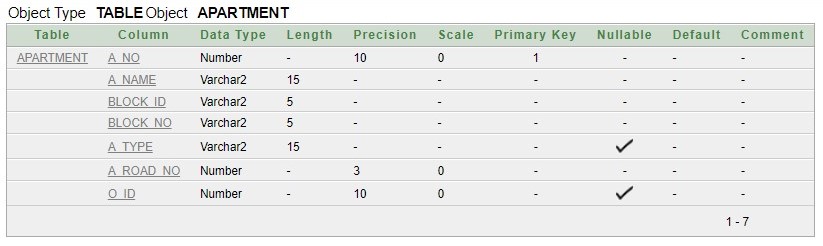
**Table Creation:**

**1. Apartment\_owner table:**

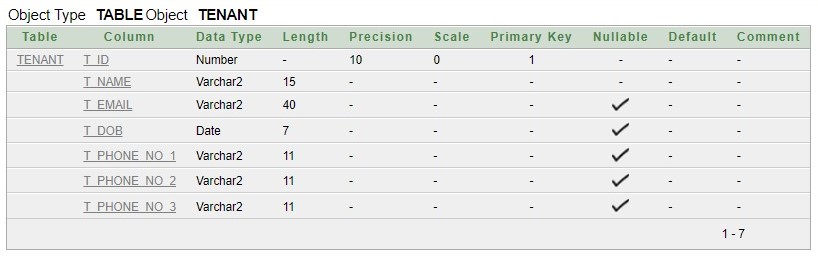
1. create table apartment\_owner (o\_id number(10)primary key, o\_name varchar2(15), email varchar2(40), o\_city varchar2(15), o\_house\_no number(5), o\_street\_no number(5), owner\_phone\_no\_1 varchar2(11), owner\_phone\_no\_2 varchar2(11), owner\_phone\_no\_3 varchar2(11));
2. ALTER TABLE apartment MODIFY a\_name NOT NULL;
3. ALTER TABLE apartment MODIFY block\_id NOT NULL;
4. ALTER TABLE apartment MODIFY block\_no NOT NULL;
5. ALTER TABLE apartment MODIFY a\_road\_no NOT NULL;
6. ALTER TABLE apartment\_owner ADD CONSTRAINT ou UNIQUE (owner\_phone\_no\_1);

**2. Apartment table:**

1. create table apartment(a\_no number(10)primary key, a\_name varchar2(15), block\_id varchar2(5), block\_no varchar2(5), a\_type varchar2(15), a\_road\_no number(3), o\_id number(10), FOREIGN KEY(o\_id) REFERENCES apartment\_owner(o\_id));
2. ALTER TABLE apartment\_owner MODIFY o\_name NOT NULL;



**3. Tenant table:**

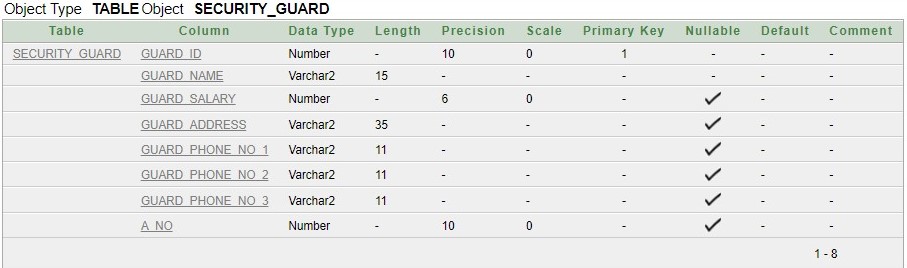
1. create table tenant(t\_id number(10)primary key, t\_name varchar2(15), t\_email varchar2(40), t\_DOB date, t\_phone\_no\_1 varchar2(11), t\_phone\_no\_2 varchar2(11), t\_phone\_no\_3 varchar2(11));
2. ALTER TABLE tenant MODIFY t\_name NOT NULL;
3. ALTER TABLE tenant ADD CONSTRAINT tu UNIQUE (t\_phone\_no\_1);

**4. Unit Table:**

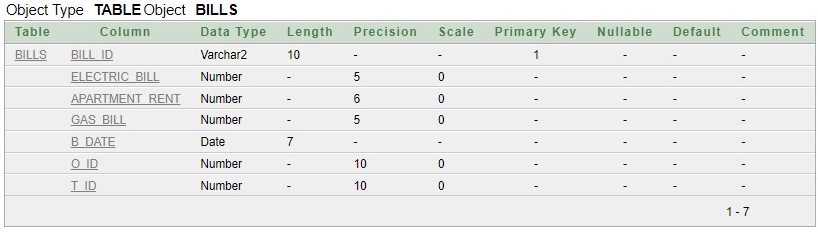
1. Table

   Description automatically generatedcreate table unit(u\_no varchar2(10)primary key, u\_size number(5), u\_level number(2), a\_no number(10), t\_id number(10), FOREIGN KEY(a\_no) REFERENCES apartment(a\_no), FOREIGN KEY(t\_id) REFERENCES tenant(t\_id));

**5. Security\_guard table:**

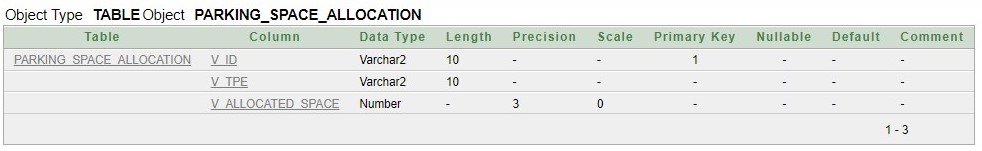
1. create table security\_guard(guard\_id number(10)primary key, guard\_name varchar2(15), guard\_salary number(6), guard\_address varchar2(35), guard\_phone\_no\_1 varchar2(11), guard\_phone\_no\_2 varchar2(11), guard\_phone\_no\_3 varchar2(11), a\_no number(10), FOREIGN KEY(a\_no) REFERENCES apartment(a\_no));
2. ALTER TABLE security\_guard MODIFY guard\_name NOT NULL;
3. ALTER TABLE security\_guard ADD CONSTRAINT su UNIQUE (guard\_phone\_no\_1);

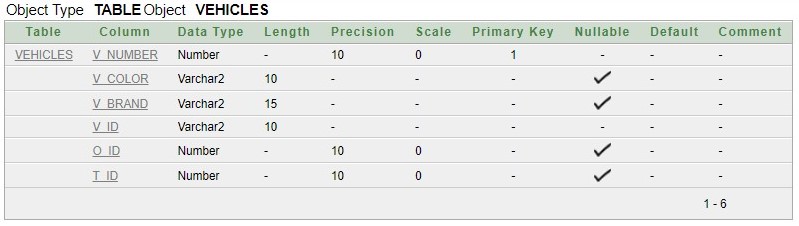
**6. Bills table:**

1. create table bills(bill\_id varchar2(10)primary key, electric\_bill number(5),apartment\_rent number(6), gas\_bill number(5), b\_date date, o\_id number(10), t\_id number(10), FOREIGN KEY(o\_id) REFERENCES apartment\_owner(o\_id), FOREIGN KEY(t\_id) REFERENCES tenant(t\_id));
2. ALTER TABLE bills MODIFY electric\_bill NOT NULL;
3. ALTER TABLE bills MODIFY apartment\_rent NOT NULL;
4. ALTER TABLE bills MODIFY gas\_bill NOT NULL;
5. ALTER TABLE bills MODIFY b\_date NOT NULL;
6. ALTER TABLE bills MODIFY t\_id NOT NULL;
7. ALTER TABLE bills MODIFY o\_id NOT NULL;

**7. Parking\_Space\_Allocation table:**

1. create table parking\_space\_allocation (v\_id varchar2(10) primary key, v\_tpe varchar2(10), v\_allocated\_space number(3));
2. ALTER TABLE parking\_space\_allocation MODIFY v\_tpe NOT NULL;
3. ALTER TABLE parking\_space\_allocation MODIFY v\_allocated\_space NOT NULL;

**8. Vehicles table:**

1. create table vehicles(v\_number number(10)primary key, v\_color varchar2(10), v\_brand varchar2(15), v\_id varchar2(10), o\_id number(10), t\_id number(10), FOREIGN KEY(v\_id) REFERENCES parking\_space\_allocation(v\_id), FOREIGN KEY(o\_id) REFERENCES apartment\_owner(o\_id), FOREIGN KEY(t\_id) REFERENCES tenant(t\_id));
2. ALTER TABLE vehicles MODIFY v\_id NOT NULL;

**Sequences:**

1. CREATE SEQUENCE apartment\_owner\_id INCREMENT BY 10 START WITH 10;
2. CREATE SEQUENCE apartment\_no INCREMENT BY 1 START WITH 101;
3. CREATE SEQUENCE guard\_id INCREMENT BY 1 START WITH 2001;
4. CREATE SEQUENCE t\_id INCREMENT BY 1 START WITH 1001;

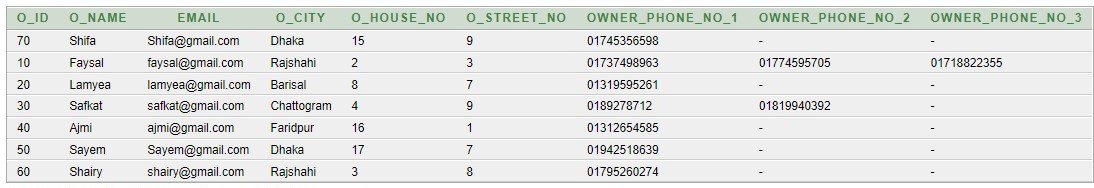
**User Creation:**

1. create user owners identified by apartment;
2. grant connect, resource, unlimited tablespace to owners;
3. Grant create view to owners;
4. ALTER USER owners DEFAULT TABLESPACE USERS;
5. ALTER USER owners TEMPORARY TABLESPACE TEMP;

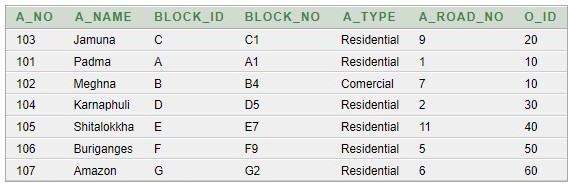
**Data Insert:**

**1. Appartment\_Owner table:**

1. insert into apartment\_owner values (APARTMENT\_OWNER\_ID.nextval, 'Faysal', 'faysal@gmail.com', 'Rajshahi', 02, 03,'01737498963', '01774595705', '01718822355');
2. insert into apartment\_owner values (APARTMENT\_OWNER\_ID.nextval, 'Lamyea', 'lamyea@gmail.com', 'Barisal', 08, 07,'01319595261', null, null);
3. insert into apartment\_owner values (APARTMENT\_OWNER\_ID.nextval, 'Safkat', 'safkat@gmail.com', 'Chattogram', 04, 09,'0189278712', '01819940392', null);
4. insert into apartment\_owner values (APARTMENT\_OWNER\_ID.nextval, 'Ajmi', 'ajmi@gmail.com', 'Faridpur', 16, 1,'01312654585', null, null);
5. insert into apartment\_owner values (APARTMENT\_OWNER\_ID.nextval, 'Sayem', 'Sayem@gmail.com', 'Dhaka', 17, 7,'01942518639', null, null);
6. insert into apartment\_owner values (APARTMENT\_OWNER\_ID.nextval, 'Shairy', 'shairy@gmail.com', 'Rajshahi', 3, 8,'01795260274', null, null);
7. insert into apartment\_owner values (APARTMENT\_OWNER\_ID.nextval, 'Shifa', 'Shifa@gmail.com', 'Dhaka', 15, 9,'01745356598', 'null', 'null');



**2. Apartment Table:**

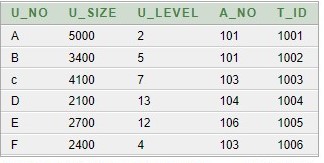
1. insert into apartment values (APARTMENT\_NO.nextval, 'Padma', 'A', 'A1', 'Residential', 1, 10);
2. insert into apartment values (APARTMENT\_NO.nextval, 'Meghna', 'B', 'B4', 'Comercial', 7, 10);
3. insert into apartment values (APARTMENT\_NO.nextval, 'Jamuna', 'C', 'C1', 'Residential', 9, 20);
4. insert into apartment values (APARTMENT\_NO.nextval, 'Karnaphuli', 'D', 'D5', 'Residential', 2, 30);
5. insert into apartment values (APARTMENT\_NO.nextval, 'Shitalokkha', 'E', 'E7', 'Residential', 11, 40);
6. insert into apartment values (APARTMENT\_NO.nextval, 'Buriganges', 'F', 'F9', 'Residential', 5, 50);
7. insert into apartment values (APARTMENT\_NO.nextval, 'Amazon', 'G', 'G2', 'Residential', 6, 60);

**3. Tenant Table:**

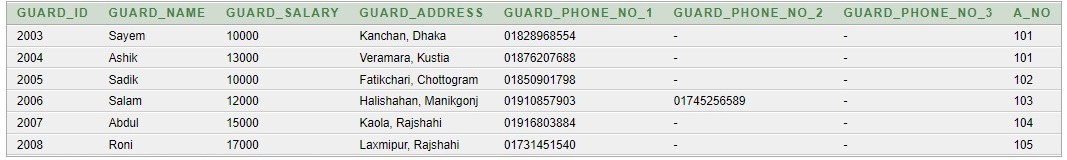
1. insert into tenant values (t\_id.nextval, 'Monalisha', 'monalisha@yahoo.com', to\_date('21-04-1997','dd-mm-yyyy'), '01732100203', null, null);
2. insert into tenant values (t\_id.nextval, 'Ayesha', 'ayesha@yahoo.com', to\_date('01-01-1988','dd-mm-yyyy'), '01766692501', '01819734712', null);
3. insert into tenant values (t\_id.nextval, 'Upanta', 'upanta@yahoo.com', to\_date('09-11-1973','dd-mm-yyyy'), '01812383971', null, null);
4. insert into tenant values (t\_id.nextval, 'Samiya', 'samiya@yahoo.com', to\_date('05-07-2001','dd-mm-yyyy'), '01612572333', null, null);
5. insert into tenant values (t\_id.nextval, 'Sadia', 'sadia@yahoo.com', to\_date('20-09-1966','dd-mm-yyyy'), '01317571923', null, null);
6. insert into tenant values (t\_id.nextval, 'Sanjana', 'samjana@yahoo.com', to\_date('12-08-1988','dd-mm-yyyy'), '01612733257', null, null);

**4. Unit Table:**

1. insert into unit values ('A', 5000, 2, 101, 1001);
2. insert into unit values ('B', 3400, 5, 101, 1002);
3. insert into unit values ('C', 4100, 7, 103, 1003);
4. insert into unit values ('D', 2100, 13, 104, 1004);
5. insert into unit values ('E', 2700, 12, 106, 1005);
6. insert into unit values ('F', 2400, 4, 103, 1006);

**5. Security\_Guard table:**

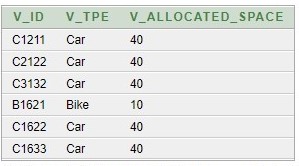
1. insert into security\_guard values (GUARD\_ID.nextval, 'Sayem', 10000, 'Kanchan, Dhaka', '01828968554', null, null, 101);
2. insert into security\_guard values (GUARD\_ID.nextval, 'Ashik', 13000, 'Veramara, Kustia', '01876207688', null, null, 101);
3. insert into security\_guard values (GUARD\_ID.nextval, 'Sadik', 10000, 'Fatikchari, Chottogram', '01850901798', null, null, 102);
4. insert into security\_guard values (GUARD\_ID.nextval, 'Salam', 12000, 'Halishahan, Manikgonj', '01910857903', '01745256589', null, 103);
5. insert into security\_guard values (GUARD\_ID.nextval, 'Abdul', 15000, 'Kaola, Rajshahi', '01916803884', null, null, 104);
6. insert into security\_guard values (GUARD\_ID.nextval, 'Roni', 17000, 'Laxmipur, Rajshahi', '01731451540', null, null, 105);



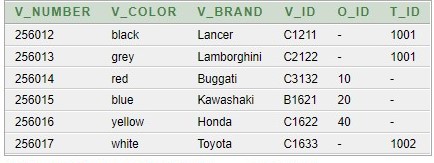
**6. Bills Table**:

1. insert into bills values('Jan2207', 2000, 27000, 1500, to\_date('07-02-2022','dd-mm-yyyy'), 10, 1001);
2. insert into bills values('Mar2203', 2500, 32000, 1000, to\_date('09-04-2022','dd-mm-yyyy'), 10, 1002);
3. insert into bills values('Sept2107', 2800, 42000, 2000, to\_date('01-10-2021','dd-mm-yyyy'), 20, 1003);
4. insert into bills values('Dec2113', 1800, 3700, 1500, to\_date('13-01-2022','dd-mm-yyyy'), 30, 1004);
5. insert into bills values('Oct2207', 2000, 40000, 1000, to\_date('10-11-2022','dd-mm-yyyy'), 50, 1005);
6. insert into bills values('Feb2202', 2100, 35000, 1500, to\_date('07-03-2022','dd-mm-yyyy'), 20, 1006);

**7. Parking\_Space\_Allocation Table:**

1. insert into parking\_space\_allocation values ('C1211', 'Car', 40);
2. insert into parking\_space\_allocation values ('C2122', 'Car', 40);
3. insert into parking\_space\_allocation values ('C3132', 'Car', 40);
4. insert into parking\_space\_allocation values ('B1621', 'Bike', 10);
5. insert into parking\_space\_allocation values ('C1622', 'Car', 40);
6. insert into parking\_space\_allocation values ('C1633', 'Car', 40);

**8. Vehicles Table:**

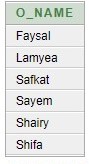
1. insert into vehicles values (256012, 'black', 'Lancer', 'C1211', null, 1001);
2. insert into vehicles values (256013, 'grey', 'Lamborghini', 'C2122', null, 1001);
3. insert into vehicles values (256014, 'red', 'Buggati', 'C3132', 10, null);
4. insert into vehicles values (256015, 'blue', 'Kawashaki', 'B1621', 20, null);
5. insert into vehicles values (256016, 'yellow', 'Honda', 'C1622', 40, null);
6. insert into vehicles values (256017, 'white', 'Toyota', 'C1633', null, 1002);

**Queries Questions and Answers:**

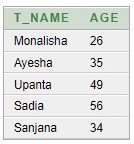
**Single row function:**

1. Display those apartment owner names whose name consists of more than 4 letters.

Ans: **select o\_name from apartment\_owner where length(o\_name)>4;**

****

2. Display those tenants name and age who are more than 25 years old.

Ans**: select t\_name, round((sysdate-t\_dob)/365) age from tenant where (sysdate-t\_dob)/365>25;**

**Group function:**

1. What is the maximum apartment rent?

Ans. **select max(apartment\_rent) as Maximum\_Apartment\_Rent from bills;**

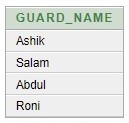
****

2. Who is the oldest tenant?

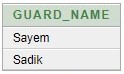
****Ans: **select t\_name from tenant where (sysdate-t\_dob)>= (select max(sysdate-t\_dob) from tenant);**

**Subquery:**

1. Which security guard has more salary than Sayem?

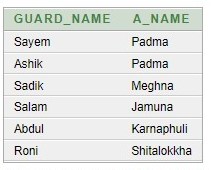
****Ans: **select guard\_name from security\_guard where guard\_salary> (select guard\_salary from security\_guard where guard\_name='Sayem');**

2. Which security guard/guards take minimum salary?

Ans: **select guard\_name from security\_guard where guard\_salary<= (select min(guard\_salary) from security\_guard);**

**Joining:**

1. Display guard name and the apartment name where they works.

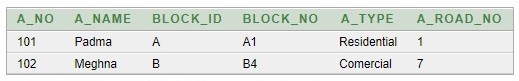
****Ans: **select g.guard\_name, a.a\_name from security\_guard g, apartment a where g.a\_no=(a.a\_no);**

2. Display which owner have which apartment with apartment types also show them who currently doesn’t hold any apartment ownership.

Ans: **select o.o\_name, a.a\_name, a.a\_type from apartment\_owner o, apartment a where o.o\_id=a.o\_id(+);**

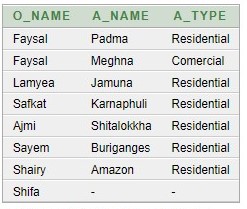
**View:**

1. Create a view which contains the apartment details of Faysal’s apartment.

****Ans: **create view Faysals\_Apartment as select a\_no, a\_name, block\_id, block\_no, a\_type, a\_road\_no from apartment where o\_id=(select o\_id from apartment\_owner where lower(o\_name)='faysal');**

2. Create a view to show total bills that tenants have to paid.

****Ans: **create view Total\_Bills as select bill\_id, electric\_bill+apartment\_rent+gas\_bill as amount, b\_date, t\_id, o\_id from bills;**



**Relational Algebra:**

1.Display only those rows of data from apartment where owner id is 10.

Ans: σo\_id= 10 (*Apartment\_Owner*)

2. Project guard name and salary from security guard table.

Ans: ∏guard\_name, guard\_salary  (Security\_Guard)

3. Find the tenant whose id is 1004.

Ans: ∏t\_name(σt\_id= 1004 (Tenant)

4. Find the apartment owner name whose city is Rajshahi.

Ans: ∏o\_name(σo\_city= “Rajshahi” (Apartment\_owner))

5. Find the guard id whose salary is higher than 15000.

Ans: ∏guard\_id(σguard\_salary> 15000 (Security\_guard))

**Conclusion:**

We have successfully created a database management system for apartments. Our apartment management system is like a joint apartment management system where a group of owners can control their apartments and store all the valuable information about tenants, parking, guards, vehicles, and so on. From our database management system, owners can easily find a tenant’s information. They can also check the bills for their apartments. In the future, we can improve our database by updating our parking system. We can add some other details, such as when a car enters or leaves parking or for how much time a vehicle is parked. Also, in the future, we can separate owners and make a database for each owner to control their apartments even more efficiently.