DATABASE MANAGEMENT SYSTEM PROJECT

COMMUNITY MANAGEMENT SYSTEM DURING COVID PANDEMIC

NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL

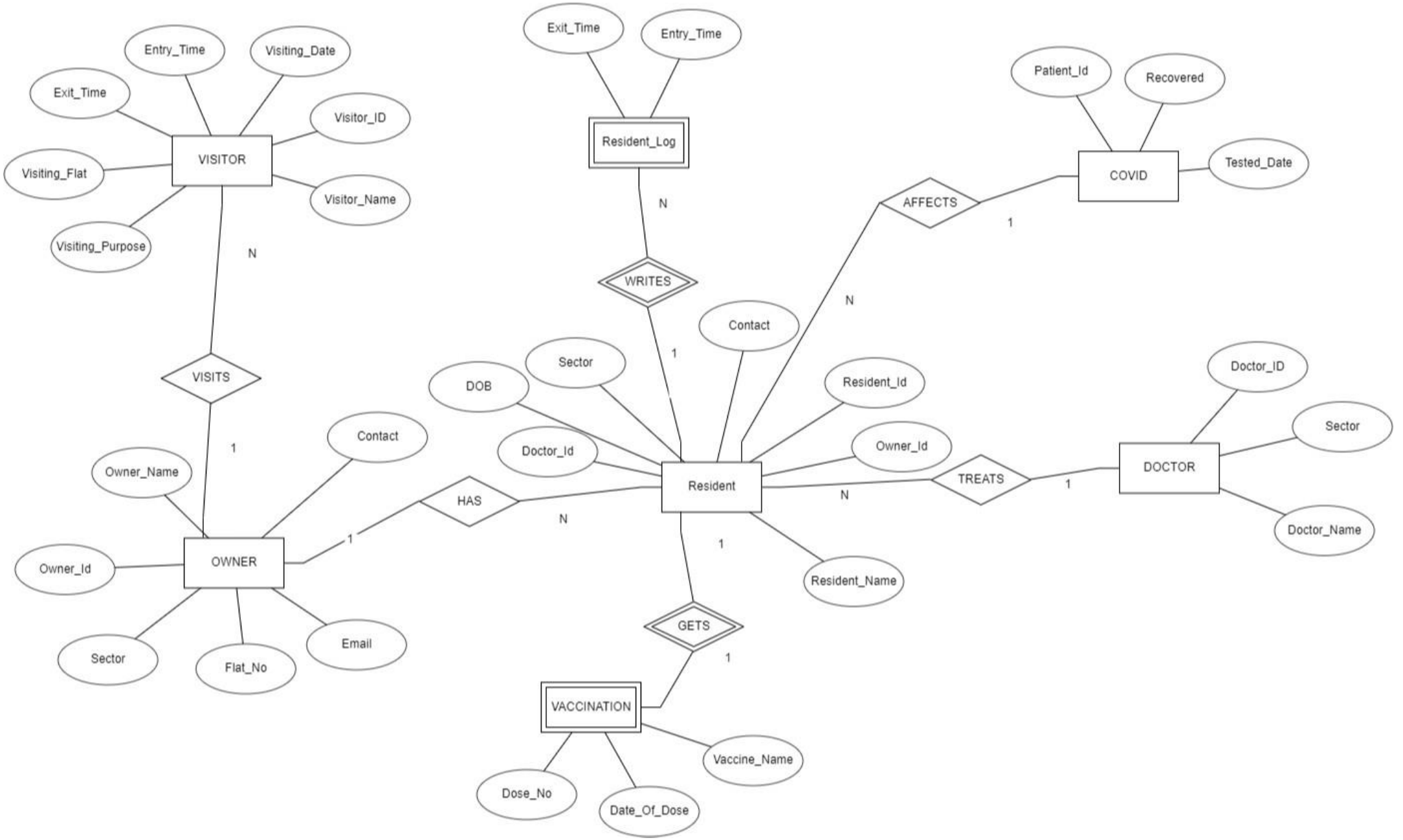
JULY '22

INTRODUCTION

In this project, I have designed a Database Management System to organize and store information about a community during the pandemic. This database contains data about residents, doctors assigned to them, residents who have tested positive for Covid-19 and whether they have recovered, and details about resident's vaccinations. It also stores information about the visitors and whom they are visiting. Through this project, we can efficiently store and retrieve crucial data that can avoid community transmission of Covid-19 by swiftly tracking down the source and isolating it.

ER MODEL ASSUMPTIONS

- An Owner can have multiple Residents living in his/her house.
- A Resident can have only one Owner. Each Owner's house has a Flat Number.
- An Owner can have multiple Visitors.
- A Visitor can go to any Flat and any Sector.
- Each Resident can have multiple logs in ResidentLog, one for each time they leave the community area.
- A Resident can be tested positive or negative for Covid.
- Each Vaccination can be given to one Resident. A Resident can take more than one dose of vaccination.
- Each Resident is allotted a Doctor. A Doctor can be allotted to multiple Residents but only in one Sector.

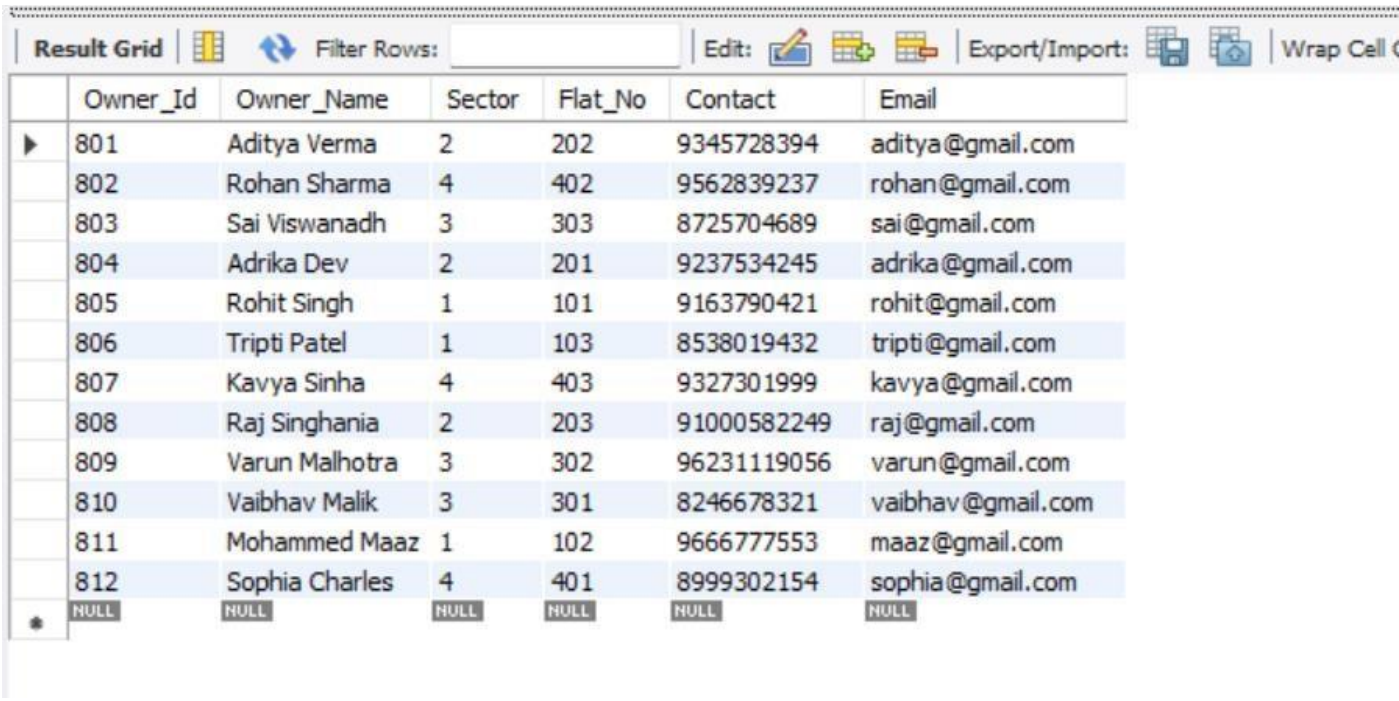

Entry_Time
Visiting_Date
Exit_Time
Visitor_ID
VISITOR
Visiting_Flat
Visitor_Name
Visiting_Purpose
N
VISITS
1
Owner_Name
Contact
Owner_Id
OWNER
Sector
Flat_No
Email
1
HAS
N
Exit_Time
Entry_Time
Resident_Log
N
WRITES
1
Contact
Sector
DOB
Resident_Id
Doctor_Id
Owner_Id
Resident
Resident_Name
1
GETS
1
VACCINATION
Vaccine_Name
Dose_No
Date_Of_Dose
Patient_Id
Recovered
COVID
Tested_Date
AFFECTS
1
N
Doctor_ID
Sector
DOCTOR
Doctor_Name
TREATS
1
N

CREATION OF TABLES

1) OWNER:

```sql
CREATE TABLE Owner(
    Owner_Id INT PRIMARY KEY,
    Owner_Name VARCHAR(50),
    Sector INT,
    Flat_No INT,
    Contact VARCHAR(20),
    Email VARCHAR(50)
);
```

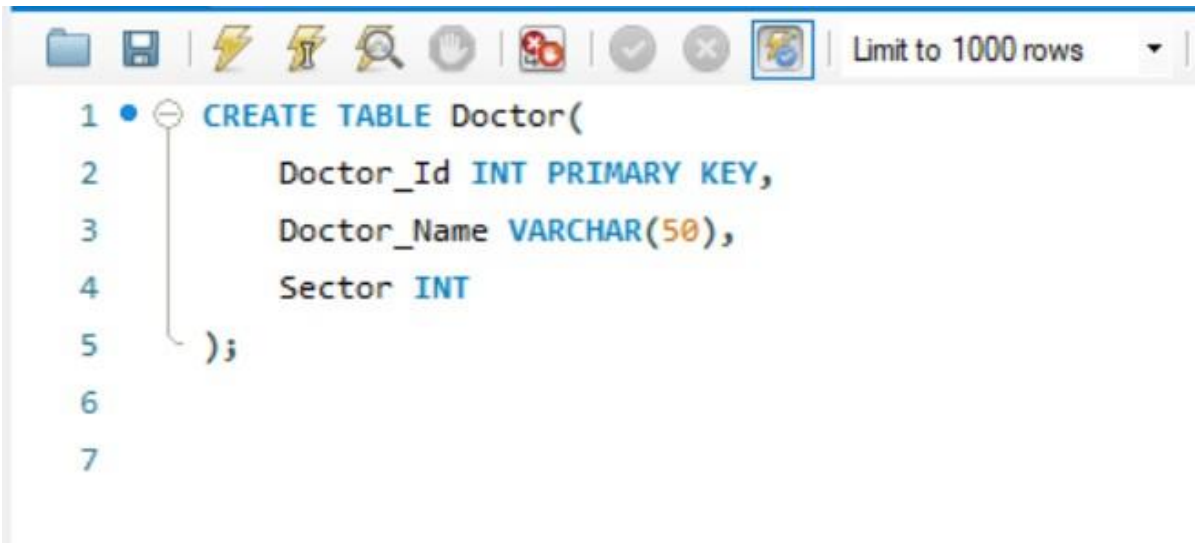
```sql
INSERT INTO Owner VALUES (801, 'Aditya Verma', 2, 202, '9345728394','aditya@gmail.com'),
                         (802, 'Rohan Sharma', 4, 402, '9562839237','rohan@gmail.com'),
                         (803, 'Sai Viswanadh', 3, 303, '8725704689','sai@gmail.com'),
                         (804, 'Adrika Dev', 2, 201, '9237534245','adrika@gmail.com'),
                         (805, 'Rohit Singh', 1, 101, '9163790421','rohit@gmail.com'),
                         (806, 'Tripti Patel', 1, 103, '8538019432','tripti@gmail.com'),
                         (807, 'Kavya Sinha', 4, 403, '9327301999','kavya@gmail.com'),
                         (808, 'Raj Singhania', 2, 203, '91000582249','raj@gmail.com'),
                         (809, 'Varun Malhotra', 3, 302, '96231119056','varun@gmail.com'),
                         (810, 'Vaibhav Malik', 3, 301, '8246678321','vaibhav@gmail.com'),
                         (811, 'Mohammed Maaz', 1, 102, '9666777553','maaz@gmail.com'),
                         (812, 'Sophia Charles', 4, 401, '8999302154','sophia@gmail.com');
```



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell

Owner_Id	Owner_Name	Sector	Flat_No	Contact	Email
801	Aditya Verma	2	202	9345728394	aditya@gmail.com
802	Rohan Sharma	4	402	9562839237	rohan@gmail.com
803	Sai Viswanadh	3	303	8725704689	sai@gmail.com
804	Adrika Dev	2	201	9237534245	adrika@gmail.com
805	Rohit Singh	1	101	9163790421	rohit@gmail.com
806	Tripti Patel	1	103	8538019432	tripti@gmail.com
807	Kavya Sinha	4	403	9327301999	kavya@gmail.com
808	Raj Singhania	2	203	91000582249	raj@gmail.com
809	Varun Malhotra	3	302	96231119056	varun@gmail.com
810	Vaibhav Malik	3	301	8246678321	vaibhav@gmail.com
811	Mohammed Maaz	1	102	9666777553	maaz@gmail.com
812	Sophia Charles	4	401	8999302154	sophia@gmail.com
NULL	NULL	NULL	NULL	NULL	NULL

2)DOCTOR



Limit to 1000 rows

```
1  CREATE TABLE Doctor(
2      Doctor_Id INT PRIMARY KEY,
3      Doctor_Name VARCHAR(50),
4      Sector INT
5  );
6
7
```

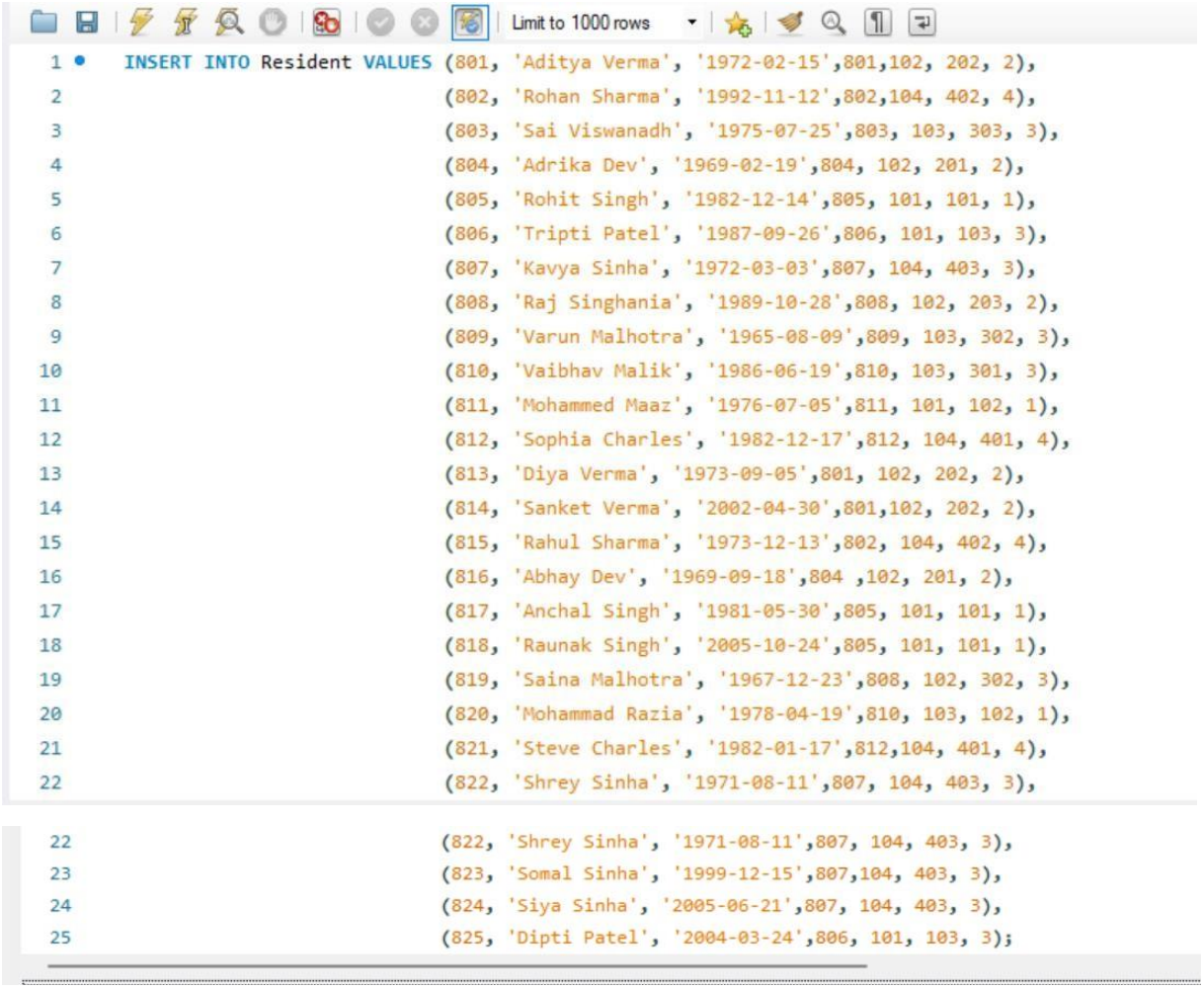
```
INSERT INTO Doctor VALUES(101, 'Shekhar Raj', 1),
                         (102, 'Tina Dubey', 2),
                         (103, 'Srinivas Reddy', 3),
                         (104, 'Devang Mukherjee', 4);
```

Result Grid | Filter Rows: | Edit:

Doctor_Id	Doctor_Name	Sector
101	Shekhar Raj	1
102	Tina Dubey	2
103	Srinivas Reddy	3
104	Devang Mukherjee	4
NULL	NULL	NULL

3)RESIDENT

```
CREATE TABLE Resident(
        Resident_Id INT PRIMARY KEY,
        Resident_Name VARCHAR(50),
        DOB DATE,
        Owner_Id INT,
        Doctor_Id INT,
        Flat_No INT,
        Sector INT,
        FOREIGN KEY(Owner_Id) REFERENCES Owner(Owner_Id),
        FOREIGN KEY(Doctor_Id) REFERENCES Doctor(Doctor_Id)
    );
```

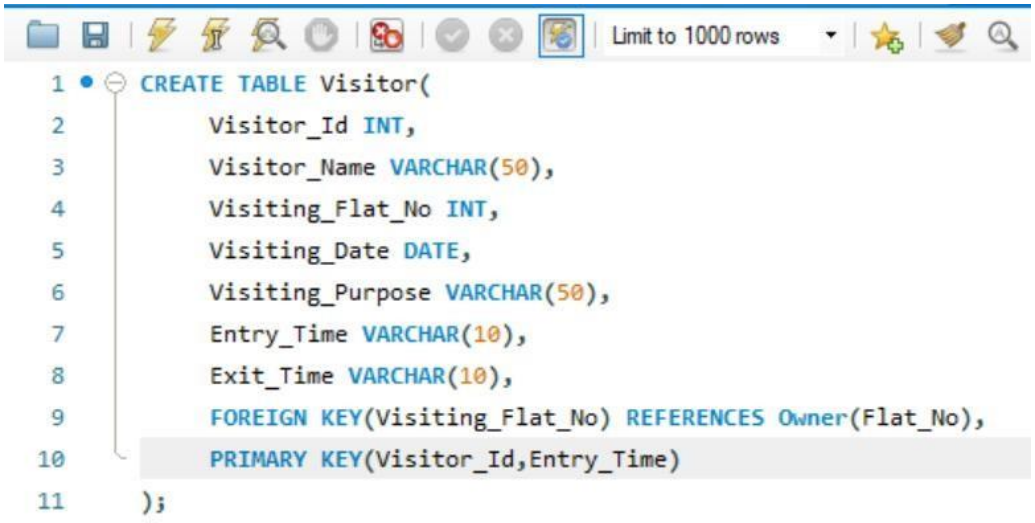


```
INSERT INTO Resident VALUES (801, 'Aditya Verma', '1972-02-15',801,102, 202, 2),
                            (802, 'Rohan Sharma', '1992-11-12',802,104, 402, 4),
                            (803, 'Sai Viswanadh', '1975-07-25',803, 103, 303, 3),
                            (804, 'Adrika Dev', '1969-02-19',804, 102, 201, 2),
                            (805, 'Rohit Singh', '1982-12-14',805, 101, 101, 1),
                            (806, 'Tripti Patel', '1987-09-26',806, 101, 103, 3),
                            (807, 'Kavya Sinha', '1972-03-03',807, 104, 403, 3),
                            (808, 'Raj Singhania', '1989-10-28',808, 102, 203, 2),
                            (809, 'Varun Malhotra', '1965-08-09',809, 103, 302, 3),
                            (810, 'Vaibhav Malik', '1986-06-19',810, 103, 301, 3),
                            (811, 'Mohammed Maaz', '1976-07-05',811, 101, 102, 1),
                            (812, 'Sophia Charles', '1982-12-17',812, 104, 401, 4),
                            (813, 'Diya Verma', '1973-09-05',801, 102, 202, 2),
                            (814, 'Sanket Verma', '2002-04-30',801,102, 202, 2),
                            (815, 'Rahul Sharma', '1973-12-13',802, 104, 402, 4),
                            (816, 'Abhay Dev', '1969-09-18',804 ,102, 201, 2),
                            (817, 'Anchal Singh', '1981-05-30',805, 101, 101, 1),
                            (818, 'Raunak Singh', '2005-10-24',805, 101, 101, 1),
                            (819, 'Saina Malhotra', '1967-12-23',808, 102, 302, 3),
                            (820, 'Mohammad Razia', '1978-04-19',810, 103, 102, 1),
                            (821, 'Steve Charles', '1982-01-17',812,104, 401, 4),
                            (822, 'Shrey Sinha', '1971-08-11',807, 104, 403, 3),
                            (822, 'Shrey Sinha', '1971-08-11',807, 104, 403, 3),
                            (823, 'Somal Sinha', '1999-12-15',807,104, 403, 3),
                            (824, 'Siya Sinha', '2005-06-21',807, 104, 403, 3),
                            (825, 'Dipti Patel', '2004-03-24',806, 101, 103, 3);
```

Resident_Id	Resident_Name	DOB	Owner_Id	Doctor_Id	Flat_No	Sector
801	Aditya Verma	1972-02-15	801	102	202	2
802	Rohan Sharma	1992-11-12	802	104	402	4
803	Sai Viswanadh	1975-07-25	803	103	303	3
804	Adrika Dev	1969-02-19	804	102	201	2
805	Rohit Singh	1982-12-14	805	101	101	1
806	Tripti Patel	1987-09-26	806	101	103	1
807	Kavya Sinha	1972-03-03	807	104	403	4
808	Raj Singhania	1989-10-28	808	102	203	2
809	Varun Malhotra	1965-08-09	809	103	302	3
810	Vaibhav Malik	1986-06-19	810	103	301	3
811	Mohammed Maaz	1976-07-05	811	101	102	1
812	Sophia Charles	1982-12-17	812	104	401	4
813	Diya Verma	1973-09-05	801	102	202	2
814	Sanket Verma	2002-04-30	801	102	202	2
815	Rahul Sharma	1973-12-13	802	104	402	4
816	Abhay Dev	1969-09-18	804	102	201	2

	817	Anchal Singh	1981-05-30	805	101	101	1
	818	Raunak Singh	2005-10-24	805	101	101	1
	819	Saina Malhotra	1967-12-23	808	102	203	2
	820	Mohammad Razia	1978-04-19	810	103	301	3
	821	Steve Charles	1982-01-17	812	104	401	4
	822	Shrey Sinha	1971-08-11	807	104	403	4
	823	Somal Sinha	1999-12-15	807	104	403	4
	824	Siya Sinha	2005-06-21	807	104	403	4
	825	Dipti Patel	2004-03-24	806	101	103	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

4)VISITORS

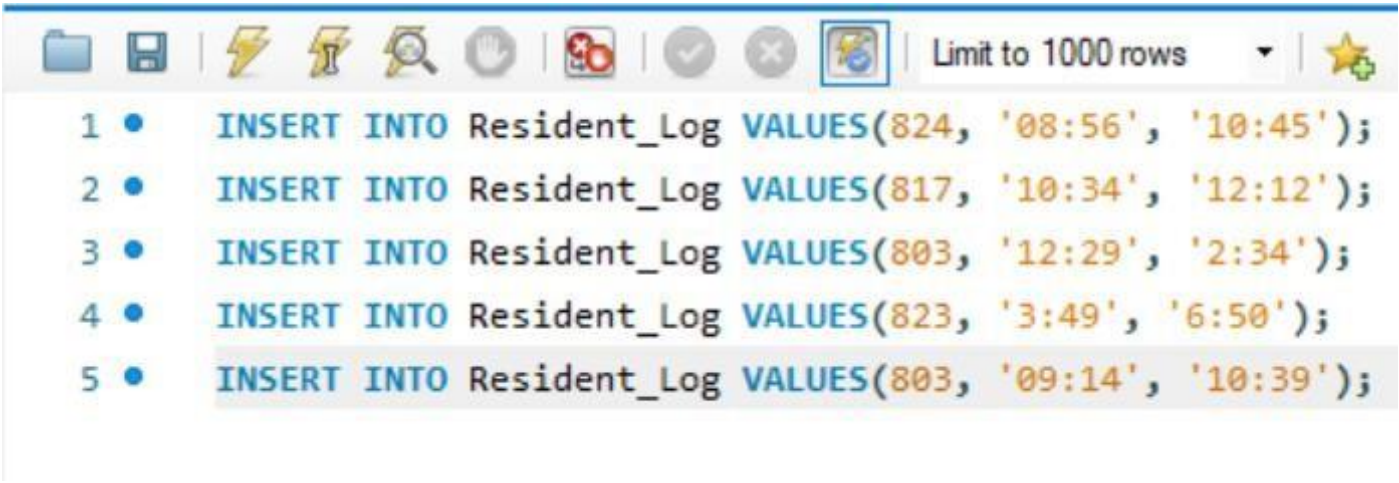


```
INSERT INTO Visitor VALUES (1201, 'Sheela', 301, '2021-01-23', 'House Keeping','08:22', '2:35');
INSERT INTO Visitor VALUES (1202, 'Ramu', 103, '2021-01-23', 'Food Delivery','01:35','1:52');
INSERT INTO Visitor VALUES (1203, 'Kalyani', 202, '2021-01-24', 'House Keeping','07:54', '12:34');
INSERT INTO Visitor VALUES (1204, 'Ramesh', 401, '2021-01-25', 'Gardening', '08:22','2:35');
INSERT INTO Visitor VALUES (1205, 'Rupa', 303, '2021-01-25', 'Visiting', '12:36', '3:39');
INSERT INTO Visitor VALUES (1206, 'Suresh', 402, '2021-01-25', 'Food Delivery','20:20', '20:35');
INSERT INTO Visitor VALUES (1207, 'Chintu', 101, '2021-01-26', 'House Keeping','10:19', '1:48');
INSERT INTO Visitor VALUES (1208, 'Rajni', 103, '2021-01-27', 'Cook', '12:12', '1:56');
INSERT INTO Visitor VALUES (1201, 'Sheela', 202, '2021-01-23', 'House Keeping','03:22', '4:35');
INSERT INTO Visitor VALUES (1202, 'Ramu', 303, '2021-01-23', 'Food Delivery','02:35','3:52');
INSERT INTO Visitor VALUES (1204, 'Ramesh', 401, '2021-01-25', 'Gardening', '05:22','6:35');
```

Visitor_Id	Visitor_Name	Visiting_Flat_No	Visiting_Date	Visiting_Purpose	Entry_Time	Exit_Time
1201	Sheela	202	2021-01-23	House Keeping	03:22	4:35
1201	Sheela	301	2021-01-23	House Keeping	08:22	2:35
1202	Ramu	103	2021-01-23	Food Delivery	01:35	1:52
1202	Ramu	303	2021-01-23	Food Delivery	02:35	3:52
1203	Kalyani	202	2021-01-24	House Keeping	07:54	12:34
1204	Ramesh	401	2021-01-25	Gardening	05:22	6:35
1204	Ramesh	401	2021-01-25	Gardening	08:22	2:35
1205	Rupa	303	2021-01-25	Visiting	12:36	3:39
1206	Suresh	402	2021-01-25	Food Delivery	20:20	20:35
1207	Chintu	101	2021-01-26	House Keeping	10:19	1:48
1208	Rajni	103	2021-01-27	Cook	12:12	1:56
NULL	NULL	NULL	NULL	NULL	NULL	NULL

5)RESIDENT LOG

```
CREATE TABLE Resident_Log(
        Resident_Id INT NOT NULL,
        Exit_Time VARCHAR(10),
        Entry_Time VARCHAR(10),
        FOREIGN KEY (Resident_Id) REFERENCES Resident(Resident_Id),
        PRIMARY KEY(Resident_Id,Exit_Time)
);
```



	Resident_Id	Exit_Time	Entry_Time
▶	803	09:14	10:39
	803	12:29	2:34
	817	10:34	12:12
	823	3:49	6:50
	824	08:56	10:45
*	NULL	NULL	NULL

6)COVID

```
1  CREATE TABLE Covid(
2      Patient_Id INT PRIMARY KEY,
3      Tested_Date DATE,
4      Recovered CHAR,
5      FOREIGN KEY(Patient_Id) REFERENCES Resident(Resident_Id)
6  );
```

```
1  INSERT INTO Covid VALUES(816,'2021-01-25','N');
2  INSERT INTO Covid VALUES(809,'2021-01-26','Y');
3  INSERT INTO Covid VALUES(822,'2021-01-26','Y');
4  INSERT INTO Covid VALUES(804,'2021-01-27','N');
```

	Patient_Id	Tested_Date	Recovered
▶	804	2021-01-27	N
	809	2021-01-26	Y
	816	2021-01-25	N
	822	2021-01-26	Y
*	NULL	NULL	NULL

7)VACCINATION

```sql
CREATE TABLE Vaccination(
    Resident_Id INT,
    Vaccine_Name VARCHAR(30),
    Dose_No INT,
    Date_Of_Dose DATE,
    FOREIGN KEY(Resident_Id) REFERENCES Resident(Resident_Id),
    PRIMARY KEY(Resident_Id,Dose_No)
);
```

```sql
INSERT INTO Vaccination VALUES(801, 'Covishield', 1, '2021-01-02');
INSERT INTO Vaccination VALUES(812, 'Covaxin', 1, '2021-01-13');
INSERT INTO Vaccination VALUES(810, 'Covaxin', 1, '2021-01-13');
INSERT INTO Vaccination VALUES(825, 'Covishield', 1, '2021-01-14');
INSERT INTO Vaccination VALUES(814, 'Covaxin', 1, '2021-01-25');
INSERT INTO Vaccination VALUES(813, 'Covishield', 1, '2021-01-25');
INSERT INTO Vaccination VALUES(806, 'Sputnik', 1, '2021-01-25');
INSERT INTO Vaccination VALUES(801, 'Covishield', 2, '2021-02-02');
INSERT INTO Vaccination VALUES(810, 'Covaxin', 2, '2021-02-13');
INSERT INTO Vaccination VALUES(812, 'Covaxin', 2, '2021-02-13');
INSERT INTO Vaccination VALUES(820, 'Covishield', 1, '2021-02-14');
INSERT INTO Vaccination VALUES(825, 'Covishield', 2, '2021-02-15');
INSERT INTO Vaccination VALUES(813, 'Covishield', 2, '2021-02-25');
INSERT INTO Vaccination VALUES(814, 'Covaxin', 2, '2021-02-25');
```

	Resident_Id	Vaccine_Name	Dose_No	Date_Of_Dose
▶	801	Covishield	1	2021-01-02
	801	Covishield	2	2021-02-02
	806	Sputnik	1	2021-01-25
	810	Covaxin	1	2021-01-13
	810	Covaxin	2	2021-02-13
	812	Covaxin	1	2021-01-13
	812	Covaxin	2	2021-02-13
	813	Covishield	1	2021-01-25
	813	Covishield	2	2021-02-25
	814	Covaxin	1	2021-01-25
	814	Covaxin	2	2021-02-25
	820	Covishield	1	2021-02-14
	825	Covishield	1	2021-01-14
	825	Covishield	2	2021-02-15
*	NULL	NULL	NULL	NULL

NORMALISATION

1)OWNER

Functional Dependencies:

Owner_Id → Owner_Id,Owner_Name,Flat_No,Sector,Email,Contact

Flat_No→ Flat_No,Owner_Id,Owner_Name, Sector,Email,Contact

Closure OF Owner_Id:

$Owner\_Id^+$ = { Owner_Id,Owner_Name,Flat_No,Sector,Email,Contact}

Closure OF Flat_No:

$Flat\_No^+$ = { Flat_No,Owner_Id,Owner_Name, Sector,Email,Contact}

Candidate Keys: Owner_Id,Flat_No

Primary keys: Owner_Id

The given relation is in its highest normal form i.e, BCNF, since the LHS of all the functional dependencies are superkeys (OwnerID, DoorNo) for the relation

2)DOCTOR

Functional Dependencies:

DoctorID → DoctorID, Doctor Name, Sector

Sector → Sector, DoctorID, Doctor Name

Closure of DoctorID:

$DoctorID^+$ = {DoctorID, Doctor Name, Sector}

Closure of Sector:

$Sector^+$ = {Sector, DoctorID, Doctor Name}

Candidate Keys: DoctorID, Sector

Primary Key: DoctorID

The given relation is in its highest normal form i.e, BCNF, since the LHS of all the functional dependencies are superkeys (DoctorID, Sector) for the relation

3. RESIDENT

Functional Dependencies:

ResidentID → ResidentName, OwnerID, DoctorID, DOB, Sector, Flat_No

DoorNo → Sector

Doctor_Id→Sector

Closure of ResidentID:

ResidentID$^+$ = {ResidentName, OwnerID, DoctorID, DOB, Sector, Flat_No}

Closure of DoorNo:

DoorNo$^+$ = {Flat_No, Sector}

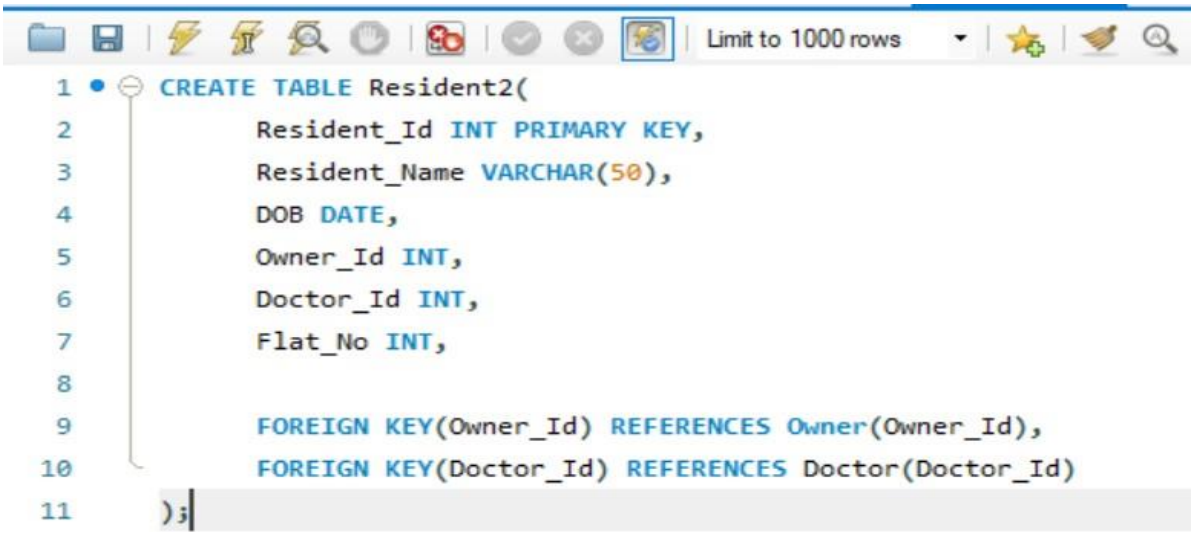
Closure of Doctor_Id:

Doctor_Id$^+$ = {Doctor_Id,Sector}

Candidate Keys: ResidentID

Primary Key: ResidentID

The given relation is not in BCNF because the LHS of the functional dependencies Flat_No → Sector and Doctor_Id→Sector i.e Flat_No and Doctor_Id is not a super key. The given relation is not in 3NF because the transitive functional dependencies exists. In the functional dependencies (Flat_No → Sector and Doctor_Id→Sector) both the LHS and RHS are non - prime attributes and therefore the relation is not in 3NF. The given relation is in 2NF because there are no partial dependencies, i.e., the proper subset of any candidate key doesn't determine a non prime attribute. To convert the given relation to a higher normal form, we decompose it into the following relations Resident2, Area , Doctor_Allotment.



```
INSERT INTO Resident2 VALUES (801, 'Aditya Verma', '1972-02-15',801,102, 202),
                             (802, "Rohan Sharma", '1992-11-12', 802,104, 402),
                             (803, 'Sai Viswanadh' , '1975-07-25',803, 103, 303),
                             (804, 'Adrika Dev', '1969-02-19',804, 102, 201),
                             (805, 'Rohit Singh', '1982-12-14',805, 101, 101),
                             (806, 'Tripti Patel', '1987-09-26', 806, 101, 103),
                             (807, 'Kavya Sinha', '1972-03-03',807, 104, 403),
                             (808, 'Raj Singhania', '1989-10-28',808, 102, 203),
                             (809, 'Varun Malhotra', '1965-08-09',809, 103, 302),
                             (810, 'Vaibhav Malik', '1986-06-19',810, 103, 301),
                             (811, 'Mohammed Maaz', '1976-07-05',811, 101, 102),
                             (812, 'Sophia Charles', '1982-12-17',812, 104, 401),
                             (813, 'Diya Verma', '1973-09-05',801, 102, 202),
                             (814, 'Sanket Verma', '2002-04-30', 801,102, 202),
                             (815, 'Rahul Sharma', '1973-12-13',802, 104, 402),
                             (816, 'Abhay Dev', '1969-09-18',804 ,102, 201),
                             (817, 'Anchal Singh', '1981-05-30',805, 101, 101),
                             (818, 'Raunak Singh', '2005-10-24',805, 101, 101),
                             (819, 'Saina Malhotra', '1967-12-23',808, 102, 302),
                             (820, 'Mohammad Razia', '1978-04-19',810, 103, 102),
                             (821, 'Steve Charles', '1982-01-17',812,104, 401),
                             (822, 'Shrey Sinha', '1971-08-11',807, 104, 403);
```

	Resident_Id	Resident_Name	DOB	Owner_Id	Doctor_Id	Flat_No
	801	Aditya Verma	1972-02-15	801	102	202
	802	Rohan Sharma	1992-11-12	802	104	402
	803	Sai Viswanadh	1975-07-25	803	103	303
	804	Adrika Dev	1969-02-19	804	102	201
	805	Rohit Singh	1982-12-14	805	101	101
	806	Tripti Patel	1987-09-26	806	101	103
	807	Kavya Sinha	1972-03-03	807	104	403
	808	Raj Singhania	1989-10-28	808	102	203
	809	Varun Malhotra	1965-08-09	809	103	302
	810	Vaibhav Malik	1986-06-19	810	103	301
	811	Mohammed Maaz	1976-07-05	811	101	102
	812	Sophia Charles	1982-12-17	812	104	401
	813	Diya Verma	1973-09-05	801	102	202
	814	Sanket Verma	2002-04-30	801	102	202
	815	Rahul Sharma	1973-12-13	802	104	402

	816	Abhay Dev	1969-09-18	804	102	201
	817	Anchal Singh	1981-05-30	805	101	101
	818	Raunak Singh	2005-10-24	805	101	101
	819	Saina Malhotra	1967-12-23	808	102	302
	820	Mohammad Razia	1978-04-19	810	103	102
▶	821	Steve Charles	1982-01-17	812	104	401
	822	Shrey Sinha	1971-08-11	807	104	403
*	NULL	NULL	NULL	NULL	NULL	NULL

Functional Dependencies:

Resident_Id → Resident_Id, Resident_Name, Owner_Id, Doctor_Id, DOB, Flat_No

Closure of ResidentID:

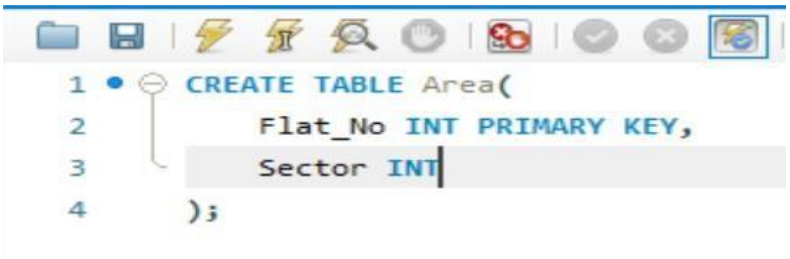
ResidentID$^+$ = {Resident_Id, Resident_Name, Owner_Id, Doctor_Id, DOB, Flat_No}

Candidate Keys: Resident_Id

Primary Key: Resident_Id

The given relation is in its highest normal form i.e, BCNF, since the LHS of all the functional dependencies are superkeys (Resident_Id) for the relation.

4)AREA



```sql
INSERT INTO Area VALUES (101, 1);
INSERT INTO Area VALUES (102, 1);
INSERT INTO Area VALUES (103, 1);
INSERT INTO Area VALUES (201, 2);
INSERT INTO Area VALUES (202, 2);
INSERT INTO Area VALUES (203, 2);
INSERT INTO Area VALUES (301, 3);
INSERT INTO Area VALUES (302, 3);
INSERT INTO Area VALUES (303, 3);
INSERT INTO Area VALUES (401, 4);
INSERT INTO Area VALUES (402, 4);
INSERT INTO Area VALUES (403, 4);
```

Flat_No	Sector
101	1
102	1
103	1
201	2
202	2
203	2
301	3
302	3
303	3
401	4
402	4
403	4
NULL	NULL

Functional Dependencies:

Flat_No → Sector

Closure of DoorNo: $Flat\_No^{+}$ = {Flat_No, Sector}

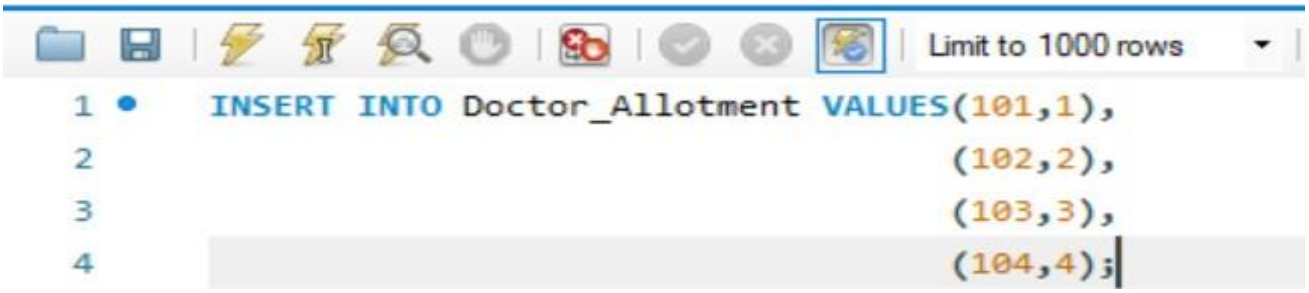
Candidate Keys: Flat_No

Primary Key: Flat_No

The given relation is in its highest normal form i.e, BCNF, since the LHS of all the functional dependencies are superkeys (Flat_No) for the relation.

5)DOCTOR_ALLOTMENT

```
1  CREATE TABLE Doctor_Allotment(
2          Doctor_Id INT PRIMARY KEY,
3          Sector INT
4  );
```



```
1  INSERT INTO Doctor_Allotment VALUES(101,1),
2                                     (102,2),
3                                     (103,3),
4                                     (104,4);
```

	Doctor_Id	Sector
▶	101	1
	102	2
	103	3
	104	4
*	NULL	NULL

Functional Dependencies:

Doctor_Id → Sector

Sector→Doctor_Id

Closure of Doctor_Id: Doctor_Id$^+$ = {Doctor_Id, Sector}

Closure of Sector: Sector+ = {Sector,Doctor_Id}

Candidate Keys: Doctor_Id,Sector

Primary Key: Doctor_Id

The given relation is in its highest normal form i.e, BCNF, since the LHS of all the functional dependencies are superkeys (Doctor_Id) for the relation.

To Ensure the Functional dependencies are preserved lets

F1 = Resident_Id→ Resident_Id,Resident_Name,Owner_Id,Doctor_Id,DOB,Flat_No.

F2 = Flat_No→Flat_No,Sector

F3 = Doctor_Id→Doctor_Id,Sector

F1 intersection F2 intersection F3 != NULL and

F1 intersection F2 = Flat_No which is Candidate key in F2

F1 intersection F3 = Doctor_Id which is candidate key in F3

There fore no functional dependencies are lost

Hence the decomposition is lossless

6)VISITOR

Functional Dependencies:

{Visitor_Id,Entry_Time} → Visitor_Id, Visitor_Name, Entry_Time, Exit_Time, Visiting_Purpose, Visiting_Date, Visiting_Flat_No

Candidate_Key: {VisitorID,Entry_Time}

Primary Key: {VisitorID,Entry_Time}

The given relation is in its highest normal form i.e, BCNF, since the LHS of all the functional dependencies are superkeys (VisitorID, Entry_time) for the relation.

7. RESIDENTLOG

Functional Dependencies:

{Resident_Id, Exit_Time} → {ResidentID, Exit_Time, Entry_Time}

Closure of {ResidentID, TimeOfDep}:

{ResidentID, TimeOfDep}$^{+}$ = {ResidentID, Exit_Time Entry_Time }

Candidate Keys: {ResidentID, Exit_Time }

Primary Key: {Resident_Id, Exit_Time}

The given relation is in its highest normal form i.e, BCNF, since the LHS of all the functional dependencies are superkeys ({Resident_Id, Exit_Time }) for the relation.

8. COVID

Functional Dependencies:

Patient_Id → Patient_Id, Patient_Name, Recovered, Tested_Date

Closure of Patient_Id: Patient_Id^{+} = {Patient_Id, Patient_Name, Recovered, TestDate}

Candidate Keys: PatientID

Primary Key: PatientID

The given relation is in its highest normal form i.e, BCNF, since the LHS of all the functional dependencies are superkeys (PatientID) for the relation.

9. VACCINATION

Functional Dependencies:

{Resident_Id, Dose_No} → Resident_Id, Dose_No, Vaccine_Name, Date_Of_Dose

Closure of {Resident_Id, Dose_No}: {Resident_Id, Dose_No}$^{+}$ = {Resident_Id, Dose_No, Vaccine_Name, Date_Of_Dose}

Candidate Keys:

{Resident_Id, Dose_No}

Primary Key:

{Resident_Id, Dose_No}

The given relation is in its highest normal form i.e, BCNF, since the LHS of all the functional dependencies are superkeys ({Resident_Id, Dose_No}) for the relation.

RELATIONAL SCHEMA WITH NORMALISED TABLES