

# **DATABASE MANAGEMENT SYSTEM PROJECT**

## **COMMUNITY MANAGEMENT SYSTEM DURING COVID PANDEMIC**



**Vipparthi Venkata Krishna Vamsi**

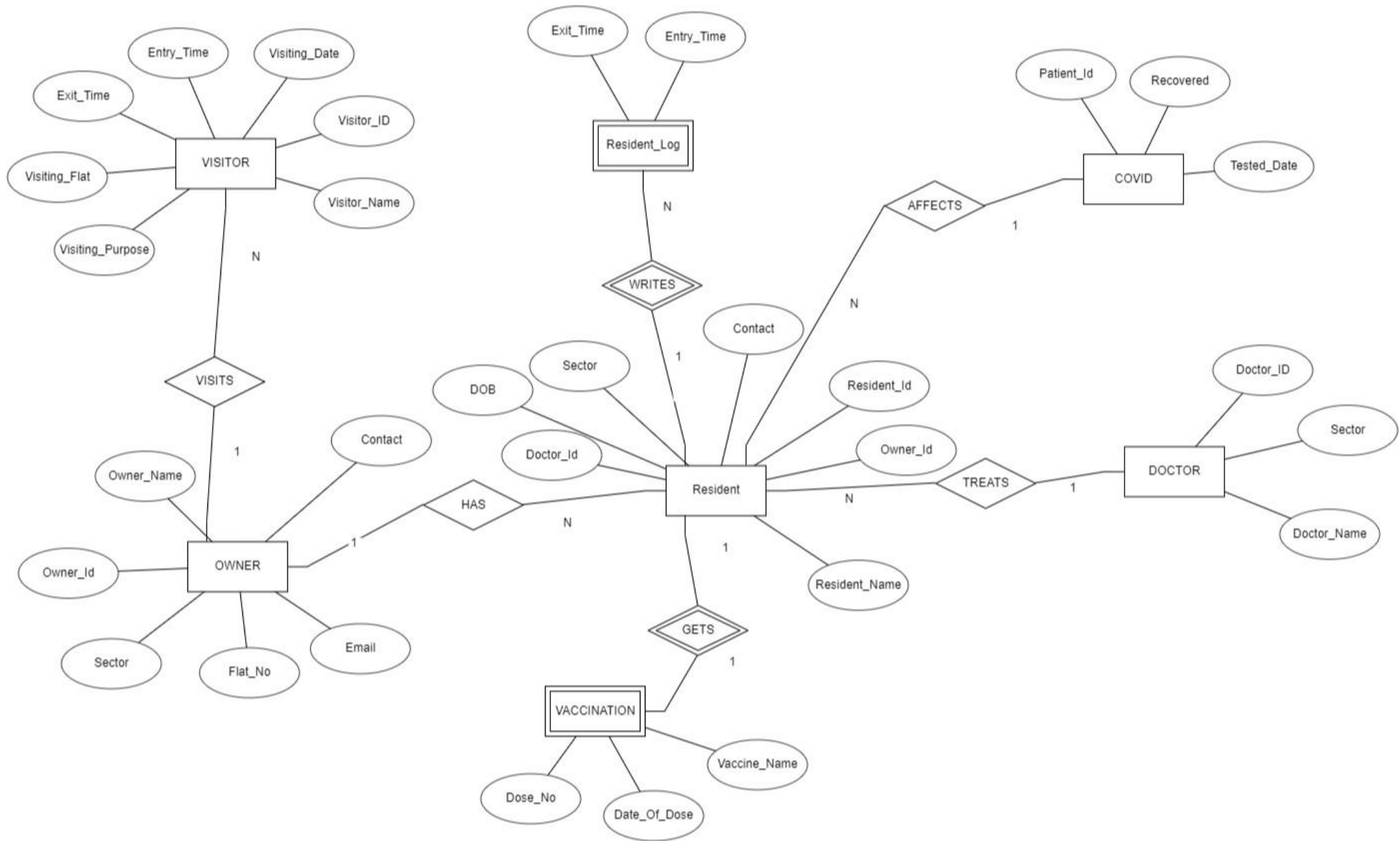
**NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL**

# 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.



# CREATION OF TABLES

## 1) OWNER:

```
1 • CREATE TABLE Owner(  
2     Owner_Id INT PRIMARY KEY,  
3     Owner_Name VARCHAR(50),  
4     Sector INT,  
5     Flat_No INT,  
6     Contact VARCHAR(20),  
7     Email VARCHAR(50)  
8 );  
9  
10
```

```
1 • INSERT INTO Owner VALUES (801, 'Aditya Verma', 2, 202, '9345728394', 'aditya@gmail.com'),  
2     (802, 'Rohan Sharma', 4, 402, '9562839237', 'rohan@gmail.com'),  
3     (803, 'Sai Viswanadh', 3, 303, '8725704689', 'sai@gmail.com'),  
4     (804, 'Adrika Dev', 2, 201, '9237534245', 'adrika@gmail.com'),  
5     (805, 'Rohit Singh', 1, 101, '9163790421', 'rohit@gmail.com'),  
6     (806, 'Tripti Patel', 1, 103, '8538019432', 'tripti@gmail.com'),  
7     (807, 'Kavya Sinha', 4, 403, '9327301999', 'kavya@gmail.com'),  
8     (808, 'Raj Singhania', 2, 203, '91000582249', 'raj@gmail.com'),  
9     (809, 'Varun Malhotra', 3, 302, '96231119056', 'varun@gmail.com'),  
10    (810, 'Vaibhav Malik', 3, 301, '8246678321', 'vaibhav@gmail.com'),  
11    (811, 'Mohammed Maaz', 1, 102, '9666777553', 'maaz@gmail.com'),  
12    (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 Singhanian	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




```

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

```

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

```
1 • CREATE TABLE Resident(
2     Resident_Id INT PRIMARY KEY,
3     Resident_Name VARCHAR(50),
4     DOB DATE,
5     Owner_Id INT,
6     Doctor_Id INT,
7     Flat_No INT,
8     Sector INT,
9     FOREIGN KEY(Owner_Id) REFERENCES Owner(Owner_Id),
10    FOREIGN KEY(Doctor_Id) REFERENCES Doctor(Doctor_Id)
11 );
```



```
Limit to 1000 rows
1 • INSERT INTO Resident VALUES (801, 'Aditya Verma', '1972-02-15',801,102, 202, 2),
2 (802, 'Rohan Sharma', '1992-11-12',802,104, 402, 4),
3 (803, 'Sai Viswanadh', '1975-07-25',803, 103, 303, 3),
4 (804, 'Adrika Dev', '1969-02-19',804, 102, 201, 2),
5 (805, 'Rohit Singh', '1982-12-14',805, 101, 101, 1),
6 (806, 'Tripti Patel', '1987-09-26',806, 101, 103, 3),
7 (807, 'Kavya Sinha', '1972-03-03',807, 104, 403, 3),
8 (808, 'Raj Singhanian', '1989-10-28',808, 102, 203, 2),
9 (809, 'Varun Malhotra', '1965-08-09',809, 103, 302, 3),
10 (810, 'Vaibhav Malik', '1986-06-19',810, 103, 301, 3),
11 (811, 'Mohammed Maaz', '1976-07-05',811, 101, 102, 1),
12 (812, 'Sophia Charles', '1982-12-17',812, 104, 401, 4),
13 (813, 'Diya Verma', '1973-09-05',801, 102, 202, 2),
14 (814, 'Sanket Verma', '2002-04-30',801,102, 202, 2),
15 (815, 'Rahul Sharma', '1973-12-13',802, 104, 402, 4),
16 (816, 'Abhay Dev', '1969-09-18',804, 102, 201, 2),
17 (817, 'Anchal Singh', '1981-05-30',805, 101, 101, 1),
18 (818, 'Raunak Singh', '2005-10-24',805, 101, 101, 1),
19 (819, 'Saina Malhotra', '1967-12-23',808, 102, 302, 3),
20 (820, 'Mohammad Razia', '1978-04-19',810, 103, 102, 1),
21 (821, 'Steve Charles', '1982-01-17',812,104, 401, 4),
22 (822, 'Shrey Sinha', '1971-08-11',807, 104, 403, 3),
23 (823, 'Somal Sinha', '1999-12-15',807,104, 403, 3),
24 (824, 'Siya Sinha', '2005-06-21',807, 104, 403, 3),
25 (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 Singhanian	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

```

1 • CREATE TABLE Visitor(
2     Visitor_Id INT,
3     Visitor_Name VARCHAR(50),
4     Visiting_Flat_No INT,
5     Visiting_Date DATE,
6     Visiting_Purpose VARCHAR(50),
7     Entry_Time VARCHAR(10),
8     Exit_Time VARCHAR(10),
9     FOREIGN KEY(Visiting_Flat_No) REFERENCES Owner(Flat_No),
10    PRIMARY KEY(Visitor_Id,Entry_Time)
11 );

```



```

1 • INSERT INTO Visitor VALUES (1201, 'Sheela', 301, '2021-01-23', 'House Keeping', '08:22', '2:35');
2 • INSERT INTO Visitor VALUES (1202, 'Ramu', 103, '2021-01-23', 'Food Delivery', '01:35', '1:52');
3 • INSERT INTO Visitor VALUES (1203, 'Kalyani', 202, '2021-01-24', 'House Keeping', '07:54', '12:34');
4 • INSERT INTO Visitor VALUES (1204, 'Ramesh', 401, '2021-01-25', 'Gardening', '08:22', '2:35');
5 • INSERT INTO Visitor VALUES (1205, 'Rupa', 303, '2021-01-25', 'Visiting', '12:36', '3:39');
6 • INSERT INTO Visitor VALUES (1206, 'Suresh', 402, '2021-01-25', 'Food Delivery', '20:20', '20:35');
7 • INSERT INTO Visitor VALUES (1207, 'Chintu', 101, '2021-01-26', 'House Keeping', '10:19', '1:48');
8 • INSERT INTO Visitor VALUES (1208, 'Rajni', 103, '2021-01-27', 'Cook', '12:12', '1:56');
9 • INSERT INTO Visitor VALUES (1201, 'Sheela', 202, '2021-01-23', 'House Keeping', '03:22', '4:35');
10 • INSERT INTO Visitor VALUES (1202, 'Ramu', 303, '2021-01-23', 'Food Delivery', '02:35', '3:52');
11 • 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


```

1 • CREATE TABLE Resident_Log(
2     Resident_Id INT NOT NULL,
3     Exit_Time VARCHAR(10),
4     Entry_Time VARCHAR(10),
5     FOREIGN KEY (Resident_Id) REFERENCES Resident(Resident_Id),
6     PRIMARY KEY(Resident_Id,Exit_Time)
7 );
8

```



Limit to 1000 rows



```
1 • INSERT INTO Resident_Log VALUES(824, '08:56', '10:45');
2 • INSERT INTO Resident_Log VALUES(817, '10:34', '12:12');
3 • INSERT INTO Resident_Log VALUES(803, '12:29', '2:34');
4 • INSERT INTO Resident_Log VALUES(823, '3:49', '6:50');
5 • INSERT INTO Resident_Log VALUES(803, '09:14', '10:39');
```

	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

```
1 • CREATE TABLE Vaccination(  
2     Resident_Id INT,  
3     Vaccine_Name VARCHAR(30),  
4     Dose_No INT,  
5     Date_Of_Dose DATE,  
6     FOREIGN KEY(Resident_Id) REFERENCES Resident(Resident_Id),  
7     PRIMARY KEY(Resident_Id,Dose_No)  
8 );
```

```
1 • INSERT INTO Vaccination VALUES(801, 'Covishield', 1, '2021-01-02');  
2 • INSERT INTO Vaccination VALUES(812, 'Covaxin', 1, '2021-01-13');  
3 • INSERT INTO Vaccination VALUES(810, 'Covaxin', 1, '2021-01-13');  
4 • INSERT INTO Vaccination VALUES(825, 'Covishield', 1, '2021-01-14');  
5 • INSERT INTO Vaccination VALUES(814, 'Covaxin', 1, '2021-01-25');  
6 • INSERT INTO Vaccination VALUES(813, 'Covishield', 1, '2021-01-25');  
7 • INSERT INTO Vaccination VALUES(806, 'Sputnik', 1, '2021-01-25');  
8 • INSERT INTO Vaccination VALUES(801, 'Covishield', 2, '2021-02-02');  
9 • INSERT INTO Vaccination VALUES(810, 'Covaxin', 2, '2021-02-13');  
10 • INSERT INTO Vaccination VALUES(812, 'Covaxin', 2, '2021-02-13');  
11 • INSERT INTO Vaccination VALUES(820, 'Covishield', 1, '2021-02-14');  
12 • INSERT INTO Vaccination VALUES(825, 'Covishield', 2, '2021-02-15');  
13 • INSERT INTO Vaccination VALUES(813, 'Covishield', 2, '2021-02-25');  
14 • 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  $\rightarrow$  Owner\_Id, Owner\_Name, Flat\_No, Sector, Email, Contact

Flat\_No  $\rightarrow$  Flat\_No, Owner\_Id, Owner\_Name, Sector, Email, Contact

### Closure OF Owner\_Id:

Owner\_Id<sup>+</sup> = { Owner\_Id, Owner\_Name, Flat\_No, Sector, Email, Contact }

### Closure OF Flat\_No:

Flat\_No<sup>+</sup> = { 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  $\rightarrow$  DoctorID, Doctor Name, Sector

Sector  $\rightarrow$  Sector, DoctorID, Doctor Name

### Closure of DoctorID:

DoctorID<sup>+</sup> = { DoctorID, Doctor Name, Sector }

### Closure of Sector:

Sector<sup>+</sup> = { 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  $\rightarrow$  ResidentName, OwnerID, DoctorID, DOB, Sector, Flat\_No

DoorNo  $\rightarrow$  Sector

Doctor\_Id  $\rightarrow$  Sector

#### Closure of ResidentID:

ResidentID<sup>+</sup> = {ResidentName, OwnerID, DoctorID, DOB, Sector, Flat\_No}

#### Closure of DoorNo:

DoorNo<sup>+</sup> = {Flat\_No, Sector}

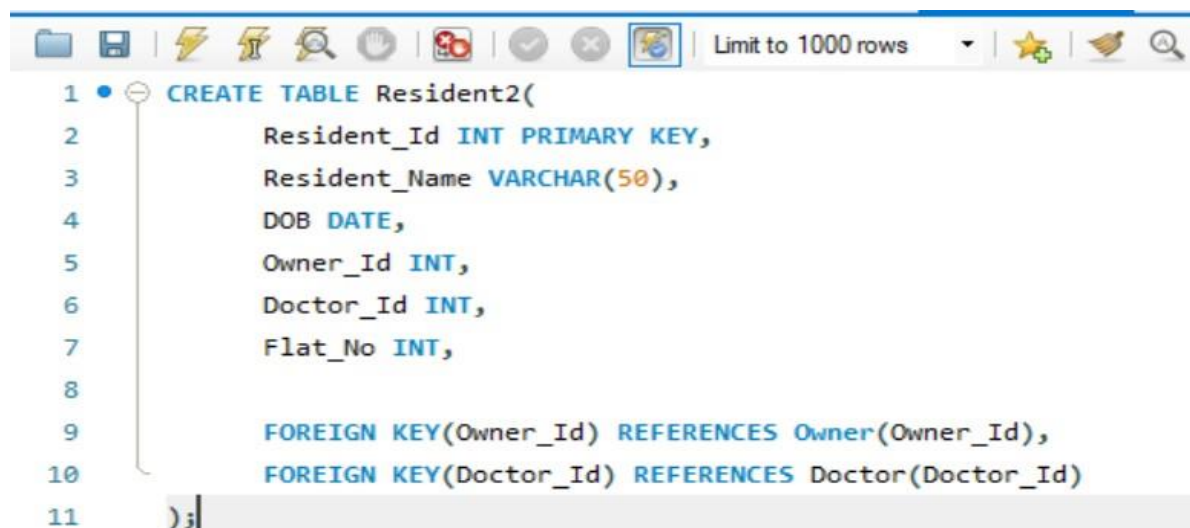
#### Closure of Doctor\_Id:

Doctor\_Id<sup>+</sup> = {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  $\rightarrow$  Sector and Doctor\_Id  $\rightarrow$  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  $\rightarrow$  Sector and Doctor\_Id  $\rightarrow$  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.



```
1 CREATE TABLE Resident2(  
2     Resident_Id INT PRIMARY KEY,  
3     Resident_Name VARCHAR(50),  
4     DOB DATE,  
5     Owner_Id INT,  
6     Doctor_Id INT,  
7     Flat_No INT,  
8  
9     FOREIGN KEY(Owner_Id) REFERENCES Owner(Owner_Id),  
10    FOREIGN KEY(Doctor_Id) REFERENCES Doctor(Doctor_Id)  
11 );
```

```

1 • INSERT INTO Resident2 VALUES (801, 'Aditya Verma', '1972-02-15',801,102, 202),
2                                (802, "Rohan Sharma", '1992-11-12', 802,104, 402),
3                                (803, 'Sai Viswanadh' , '1975-07-25',803, 103, 303),
4                                (804, 'Adrika Dev', '1969-02-19',804, 102, 201),
5                                (805, 'Rohit Singh', '1982-12-14',805, 101, 101),
6                                (806, 'Tripti Patel', '1987-09-26', 806, 101, 103),
7                                (807, 'Kavya Sinha', '1972-03-03',807, 104, 403),
8                                (808, 'Raj Singhanian', '1989-10-28',808, 102, 203),
9                                (809, 'Varun Malhotra', '1965-08-09',809, 103, 302),
10                               (810, 'Vaibhav Malik', '1986-06-19',810, 103, 301),
11                               (811, 'Mohammed Maaz', '1976-07-05',811, 101, 102),
12                               (812, 'Sophia Charles', '1982-12-17',812, 104, 401),
13                               (813, 'Diya Verma', '1973-09-05',801, 102, 202),
14                               (814, 'Sanket Verma', '2002-04-30', 801,102, 202),
15                               (815, 'Rahul Sharma', '1973-12-13',802, 104, 402),
16                               (816, 'Abhay Dev', '1969-09-18',804 ,102, 201),
17                               (817, 'Anchal Singh', '1981-05-30',805, 101, 101),
18                               (818, 'Raunak Singh', '2005-10-24',805, 101, 101),
19                               (819, 'Saina Malhotra', '1967-12-23',808, 102, 302),
20                               (820, 'Mohammad Razia', '1978-04-19',810, 103, 102),
21
22                               (821, 'Steve Charles', '1982-01-17',812,104, 401),
23                               (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 Singhanian	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<sup>+</sup> = {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

```

1 CREATE TABLE Area(
2     Flat_No INT PRIMARY KEY,
3     Sector INT
4 );

```

```

1 • INSERT INTO Area VALUES (101, 1);
2 • INSERT INTO Area VALUES (102, 1);
3 • INSERT INTO Area VALUES (103, 1);
4 • INSERT INTO Area VALUES (201, 2);
5 • INSERT INTO Area VALUES (202, 2);
6 • INSERT INTO Area VALUES (203, 2);
7 • INSERT INTO Area VALUES (301, 3);
8 • INSERT INTO Area VALUES (302, 3);
9 • INSERT INTO Area VALUES (303, 3);
10 • INSERT INTO Area VALUES (401, 4);
11 • INSERT INTO Area VALUES (402, 4);
12 • 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  $\rightarrow$  Sector

**Closure of DoorNo:** Flat\_No<sup>+</sup> = {Flat\_No, Sector}

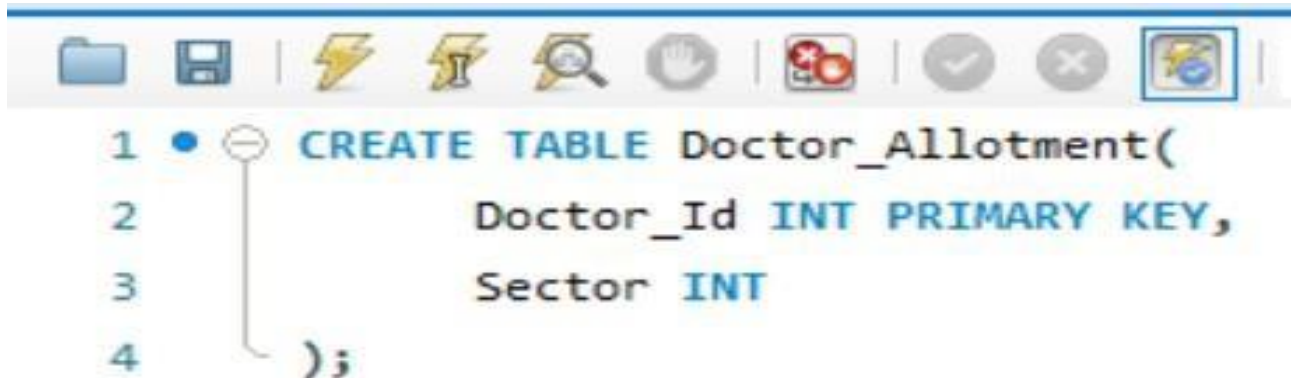
**Candidate Keys:** Flat\_No



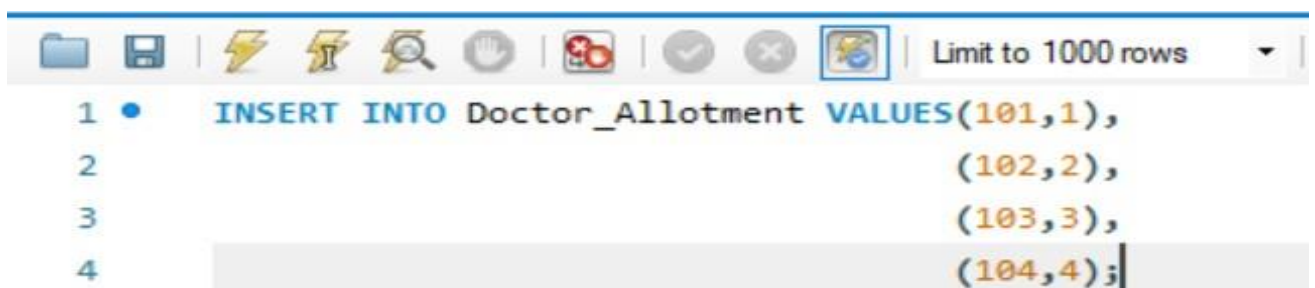
**Primary Key:** Flat\_No

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:**

$\text{Doctor\_Id} \rightarrow \text{Sector}$

$\text{Sector} \rightarrow \text{Doctor\_Id}$

**Closure of Doctor\_Id:**  $\text{Doctor\_Id}^+ = \{\text{Doctor\_Id}, \text{Sector}\}$

**Closure of Sector:**  $\text{Sector}^+ = \{\text{Sector}, \text{Doctor\_Id}\}$

**Candidate Keys:**  $\text{Doctor\_Id}, \text{Sector}$

**Primary Key:**  $\text{Doctor\_Id}$

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

To Ensure the Functional dependencies are preserved lets

$F1 = \text{Resident\_Id} \rightarrow \text{Resident\_Id}, \text{Resident\_Name}, \text{Owner\_Id}, \text{Doctor\_Id}, \text{DOB}, \text{Flat\_No}.$

$F2 = \text{Flat\_No} \rightarrow \text{Flat\_No}, \text{Sector}$

$F3 = \text{Doctor\_Id} \rightarrow \text{Doctor\_Id}, \text{Sector}$

$F1 \text{ intersection } F2 \text{ intersection } F3 \neq \text{NULL}$  and

$F1 \text{ intersection } F2 = \text{Flat\_No}$  which is Candidate key in  $F2$

$F1 \text{ intersection } F3 = \text{Doctor\_Id}$  which is candidate key in  $F3$

There fore no functional dependencies are lost

**Hence the decomposition is lossless**

## 6)VISITOR

**Functional Dependencies:**

$\{\text{Visitor\_Id}, \text{Entry\_Time}\} \rightarrow \text{Visitor\_Id}, \text{Visitor\_Name}, \text{Entry\_Time}, \text{Exit\_Time}, \text{Visiting\_Purpose}, \text{Visiting\_Date}, \text{Visiting\_Flat\_No}$

**Candidate Key:**  $\{\text{VisitorID}, \text{Entry\_Time}\}$

**Primary Key:**  $\{\text{VisitorID}, \text{Entry\_Time}\}$

The given relation is in its highest normal form i.e, BCNF, since the LHS of all the functional dependencies are superkeys ( $\text{VisitorID}, \text{Entry\_time}$ ) for the relation.

## 7. RESIDENTLOG

### Functional Dependencies:

$\{Resident\_Id, Exit\_Time\} \rightarrow \{ResidentID, Exit\_Time, Entry\_Time\}$

### 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:

$Patient\_Id \rightarrow Patient\_Id, Patient\_Name, Recovered, Tested\_Date$

**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:

$\{Resident\_Id, Dose\_No\} \rightarrow Resident\_Id, Dose\_No, Vaccine\_Name, Date\_Of\_Dose$

**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

