KATHMANDU UNIVERSITY

SCHOOL OF ENGINEERING

DHULIKHEL, KAVRE

GEOM 318: Spatial Database Management System

COVID Vaccine Distribution Database Management System

Mini Project Report

Submitted by:

Aroj KC (024371-18)

Nimesh Bhandari (024364-18)

Puja Pudasaini (024379-18)

Rohit Yadav (024390-18)

Subham Subedi (024384-18)

Submitted to:

Mr. Kushal Sharma

Lecturer

Department of Geomatics Engineering

June 2021

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ER	Entity - Relationship
GIS	Geographic Information System
SQL	Structured Query Language

1. INTRODUCTION

1.1 Background

The vaccination against COVID-19 is probably the biggest task in public health sector in the near history, in present context as well as in upcoming future. In view of the dramatic development of the COVID pandemic, time is pressing and with the development of vaccines, the vaccination distribution strategy must be implemented operationally. In addition, the high demands for vaccine among the skyrocketing population of the supply chain and addressing the comprehensive requirements like low temperature storage requirement for vaccines is a challenging task.

PostgreSQL is a powerful, open source object-relational database system that uses and extends the SQL language along with many others features and extensions for accurate and effective storage and easy retrieval of data and scale the most complicated data workloads (PostgreSQL, 2021). One of the such extensions is PostGIS. PostGIS is a spatial database extender for PostgreSQL object-relational database. It adds support for geographic objects allowing location queries to be run in SQL (PostGIS, n.d.). By utilizing this powerful system, an approach for developing a COVID Vaccine Distribution Database Management System is embraced.

1.2 Problem Statement

It's not vaccines that will stop the pandemic, it's proper and systematic vaccination. However, in our country Nepal, where all the vaccines available are based on the ofttimes donations from other countries, their systematic and effective distribution has been the great question mark. In absence of proper digital database, the needy people are not being prioritized or more precisely, they are not getting opportunities, but the people in the power and their flatters are given the first priority. There is no transparency in the distribution process.

To address this issue, a prototype of vaccination database system has been prepared here that can be helpful for the proper management and distribution of the vaccine to each citizen. It integrates all parties involved in the vaccination process in a holistic approach

- from the vaccine information to the vaccination center and to the health personal to the citizens to be vaccinated. This ensures the smooth handling of the entire vaccination process as well as a high level of information transparency. This computer-based system aids in in making the whole functioning paperless along with facilitating in lowering risks and managing records effectively.

1.3 Importance of COVID Vaccine Distribution Database Management System

The key importance of the COVID Vaccine Distribution Database Management System are highlighted below:

- The records are the collection of organized information that can easily be accessed, managed and updated.
- This management system helps in providing a framework to facilitate data quality initiatives. In turn, higher quality information helps in making better and faster decisions about what can be done further.
- It provides a paperless management.

2. METHODOLOGY OUTLINE

2.1 Conceptual Data Model

An entity relationship model is a high-level conceptual data model that allows us to describe the data involved in a real-world enterprise in terms of objects and their relationships. It is the ER model that helps us to describe and analyze the requirements of data in more detailed and precise form, systematically, prior to production a well-designed database (Ramakrisnan & Gehrke, n.d.). A conceptual model was developed prior to the development of COVID Vaccination Distribution Database consisting of different entities with the relationship among them and the associated attributes as shown in Figure 1.

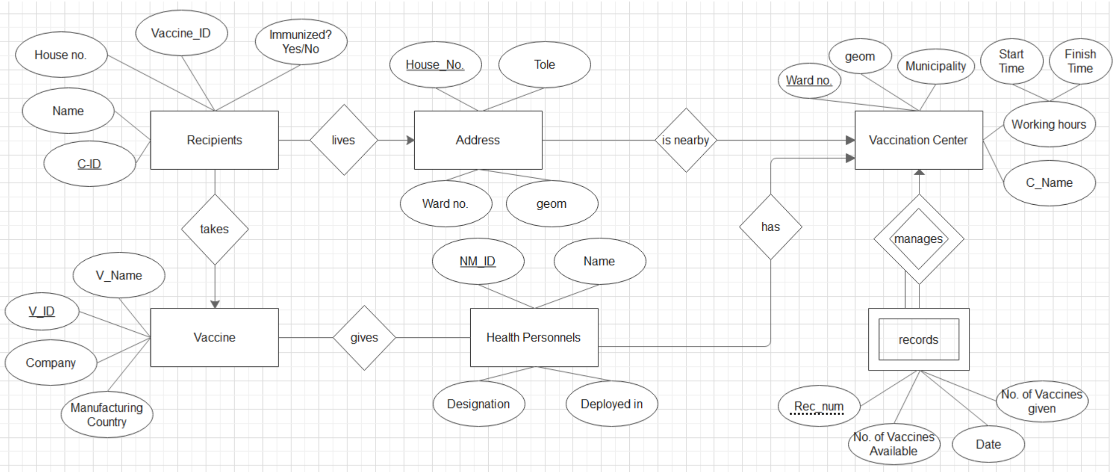


Figure 1. ER Diagram Showing Entities and their Relationship for COVID Vaccination Distribution System

The model consists of six entity sets, one of which is weak entity set that doesn't have a primary key. In each entity set, there is a primary key represented by underlined attribute whereas the discriminator or partial key (attribute that distinguishes among all the entities of a set) in a weak entity is represented by the dotted underline.

Regarding the relationship among the entities, the entities are mapped with the help of arrows. The schematic representations of different cardinalities are shown below:

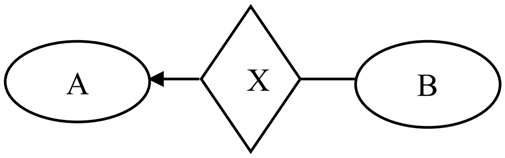


Figure 2. One to Many Relationship

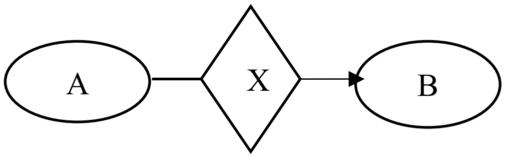


Figure 3. Many to One Relationship

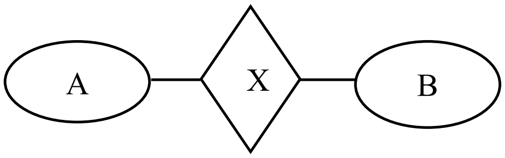


Figure 5. Many to Many Relationship

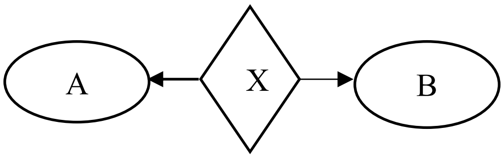


Figure 4. One to One Relationship

2.2 Relational Data Model

The relational data model is the most widely used data model which is a collection of one or more relations where each relation is a table with rows and columns. The main construct for representing data in the relational model is a relation (Ramakrisnan & Gehrke, n.d.).

The relation for each of the entities are created by using the Structured Query Language (SQL). Simultaneously, sample data were inserted into the created table as follows:

A) Creating a Database

SQL for Creating database:

```
CREATE DATABASE Vaccination;
```

B) Creating Tables and Inserting Sample Values

i) Recipient

■ SQL for Creating Table Recipient:

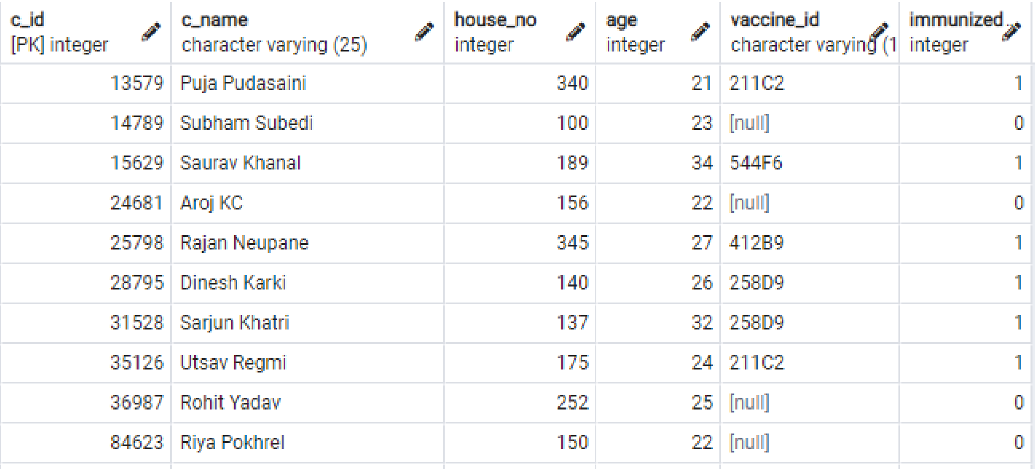
```
CREATE TABLE Recipient
(
        C_ID INTEGER PRIMARY KEY NOT NULL,
        C_Name VARCHAR(25),
        House_no INTEGER,
        Age INTEGER,
        Vaccine_ID VARCHAR(10),
        Immunized INTEGER DEFAULT 0
)
```

→ In Immunized field, 0 = Not Immunized | 1 = Immunized

■ SQL for Inserting Records in Table Recipient:

```
INSERT INTO Recipient VALUES(13579, 'Puja Pudasaini', 340,21,'211C2',1);

INSERT INTO Recipient VALUES(24681, 'Aroj KC', 156,22);

INSERT INTO Recipient VALUES(14789, 'Subham Subedi',100,23);

INSERT INTO Recipient VALUES(36987, 'Rohit Yadav',252,25);

INSERT INTO Recipient VALUES(25798, 'Rajan Neupane',345,27,'412B9',1);

INSERT INTO Recipient VALUES(84623, 'Riya Pokhrel',150,22);

INSERT INTO Recipient VALUES(35126, 'Utsav Regmi',175,24,'211C2',1);

INSERT INTO Recipient VALUES(28795, 'Dinesh Karki',140,26,'258D9',1);

INSERT INTO Recipient VALUES(31528, 'Sarjun Khatri',137,32,'258D9',1);

INSERT INTO Recipient VALUES(15629, 'Saurav Khanal',189,34,'544F6',1);
```

■ Resulting Table:



c_id [PK] integer	c_name character varying (25)	house_no integer	age integer	vaccine_id character varying (1	immunized integer
13579	Puja Pudasaini	340	21	211C2	1
14789	Subham Subedi	100	23	[null]	0
15629	Saurav Khanal	189	34	544F6	1
24681	Aroj KC	156	22	[null]	0
25798	Rajan Neupane	345	27	412B9	1
28795	Dinesh Karki	140	26	258D9	1
31528	Sarjun Khatri	137	32	258D9	1
35126	Utsav Regmi	175	24	211C2	1
36987	Rohit Yadav	252	25	[null]	0
84623	Riya Pokhrel	150	22	[null]	0

Figure 6. Recipient Table

ii) Vaccine

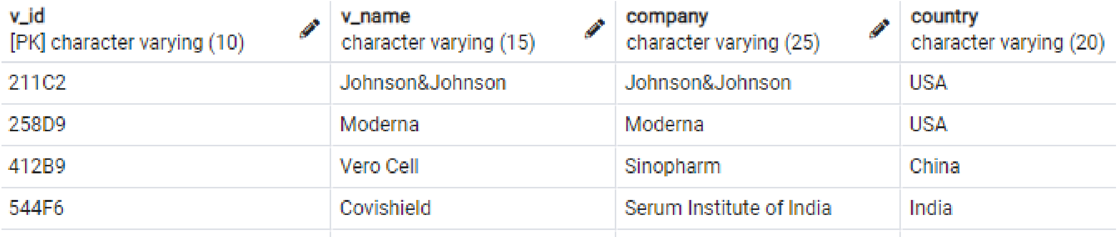
■ SQL for Creating Table Vaccine:

```
CREATE TABLE Vaccine

(

        V_ID VARCHAR(10) PRIMARY KEY NOT NULL,

        V_Name VARCHAR(15),

        Company VARCHAR (25),

        Country VARCHAR (20)

)
```

■ SQL for Inserting Records in Table Vaccine:

```
INSERT INTO Vaccine VALUES('211C2', 'Johnson&Johnson', 'Johnson&Johnson','USA');
INSERT INTO Vaccine VALUES('412B9', 'Vero Cell', 'Sinopharm','China');
INSERT INTO Vaccine VALUES('258D9', 'Moderna', 'Moderna','USA');
INSERT INTO Vaccine VALUES('544F6', 'Covishield', 'Serum Institute of India','India');
```

■ Resulting Table



v_id [PK] character varying (10)	v_name character varying (15)	company character varying (25)	country character varying (20)
211C2	Johnson&Johnson	Johnson&Johnson	USA
258D9	Moderna	Moderna	USA
412B9	Vero Cell	Sinopharm	China
544F6	Covishield	Serum Institute of India	India

Figure 7. Vaccine Table

iii) <u>Health Personnels</u>

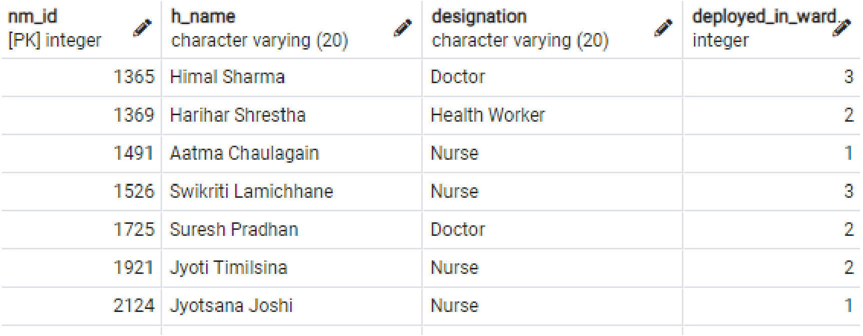
■ SQL for Creating Table Health Personnel:

```
CREATE TABLE Health_Personnels
(
        NM_ID INTEGER PRIMARY KEY NOT NULL,
        H_Name VARCHAR(20),
        Designation VARCHAR (20),
        Deployed_in_Ward INTEGER
)
```

■ SQL for Inserting Records in Table Health Personnel:

```
INSERT INTO Health_Personnels VALUES (1365, 'Himal Sharma', 'Doctor', 3);
INSERT INTO Health_Personnels VALUES (2124, 'Jyotsana Joshi', 'Nurse', 1);
INSERT INTO Health_Personnels VALUES (1921, 'Jyoti Timilsina', 'Nurse', 2);
INSERT INTO Health_Personnels VALUES (1369, 'Harihar Shrestha', 'Health Worker', 2);
INSERT INTO Health_Personnels VALUES (1491, 'Aatma Chaulagain', 'Nurse', 1);
INSERT INTO Health_Personnels VALUES (1526, 'Swikriti Lamichhane', 'Nurse', 3);
INSERT INTO Health_Personnels VALUES (1725, 'Suresh Pradhan', 'Doctor', 2);
```

- Resulting Table



nm_id [PK] integer	h_name character varying (20)	designation character varying (20)	deployed_in_ward integer
1365	Himal Sharma	Doctor	3
1369	Harihar Shrestha	Health Worker	2
1491	Aatma Chaulagain	Nurse	1
1526	Swikriti Lamichhane	Nurse	3
1725	Suresh Pradhan	Doctor	2
1921	Jyoti Timilsina	Nurse	2
2124	Jyotsana Joshi	Nurse	1

Figure 8. Health Personnel Table

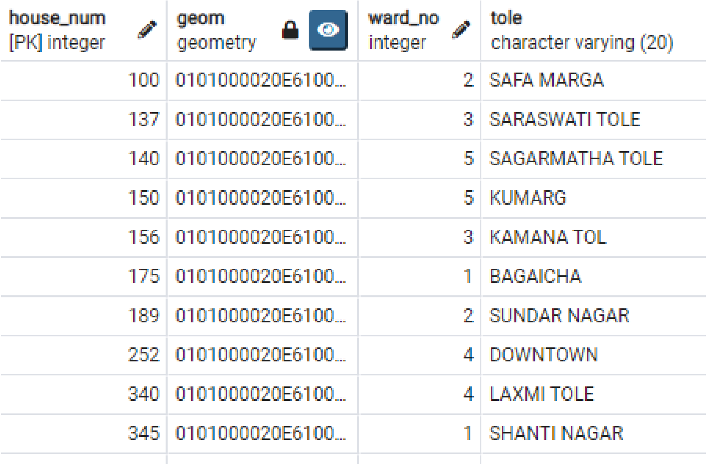
iv) Address

As a relation consisting of spatial field, first the spatial feature for address was created/developed on a mapping software ArcGIS. This means, the sample point features for Address were created on a ArcMap so that the real world geographical spatial position of points automatically embeds along with the feature and stored on a Geometry field of the table. Later on, a connection was established between ArcMap and PostgreSQL using a localhost server and thus developed spatial feature was imported in PostgreSQL and other fields were created by altering table.

- SQL for Creating More Fields in Table Address:

```
ALTER TABLE address
ADD COLUMN ward_no INTEGER,
ADD COLUMN tole VARCHAR(25);
```

And the sample records were inserted in respective fields.



house_num [PK] integer	geom geometry	ward_no integer	tole character varying (20)
100	0101000020E6100...	2	SAFA MARGA
137	0101000020E6100...	3	SARASWATI TOLE
140	0101000020E6100...	5	SAGARMATHA TOLE
150	0101000020E6100...	5	KUMARG
156	0101000020E6100...	3	KAMANA TOL
175	0101000020E6100...	1	BAGAICHA
189	0101000020E6100...	2	SUNDAR NAGAR
252	0101000020E6100...	4	DOWNTOWN
340	0101000020E6100...	4	LAXMI TOLE
345	0101000020E6100...	1	SHANTI NAGAR

Figure 10. Address Table

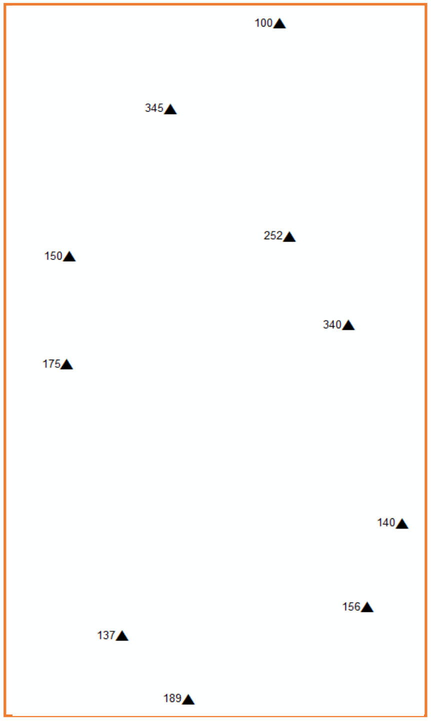


Figure 9. Geometric Output (Location) of Households

v) <u>Vaccination Center</u>

Similar to the spatial feature created in ArcMap for address, similar features were created for vaccination center as well because we needed a spatial field in this relation. In th same manner, the connection was established between ArcMap and PostgreSQL using localhost server and other fields were created in PostgreSQL.

■ SQL for Creating More Fields in Table Vaccination Center:

```
ALTER TABLE V_center
ADD COLUMN c_name VARCHAR(30),
ADD COLUMN municipality VARCHAR(20),
ADD COLUMN start_time TIME,
ADD COLUMN finish_time TIME;
```

The sample records were imported in the respective field of the table.

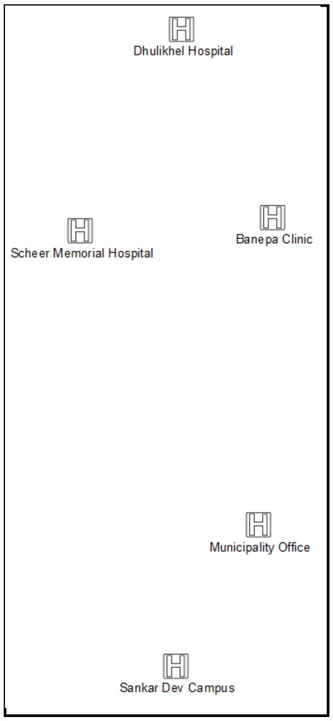
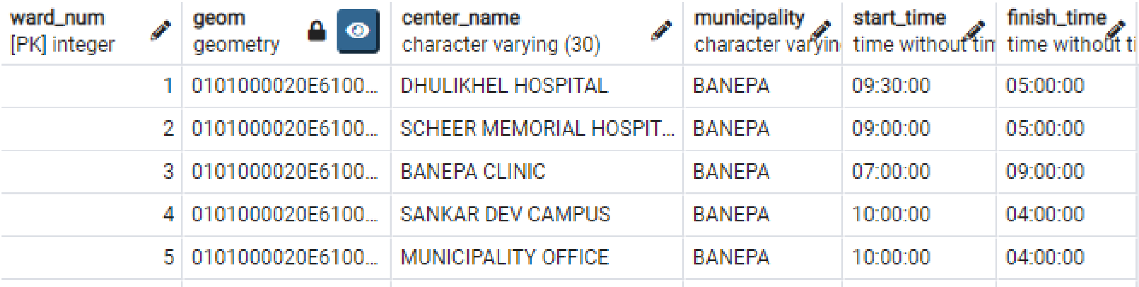


Figure 11. Geometric View of Vaccination Center

■ Resulting Table:



ward_num [PK] integer	geom geometry	center_name character varying (30)	municipality character varyin	start_time time without tin	finish_time time without ti
1	0101000020E6100...	DHULIKHEL HOSPITAL	BANEPA	09:30:00	05:00:00
2	0101000020E6100...	SCHEER MEMORIAL HOSPIT...	BANEPA	09:00:00	05:00:00
3	0101000020E6100...	BANEPA CLINIC	BANEPA	07:00:00	09:00:00
4	0101000020E6100...	SANKAR DEV CAMPUS	BANEPA	10:00:00	04:00:00
5	0101000020E6100...	MUNICIPALITY OFFICE	BANEPA	10:00:00	04:00:00

Figure 12. Vaccination Center Table

vi) <u>Records</u>

Records, being a weak entity set, it entirely depends on the existence of an identifying entity set (Vaccination Center, here). Thus, the table is created in reference to the table Vaccination Center.

One thing to carefully notice here is, the weak entity set ‘Records’ doesn’t have its own primary key, so the primary key ‘ward_num’ of owning entity ‘Vaccination Center’ is used as an attribute in the table ‘Records’ (*Weak Entities to Tables*, n.d.).

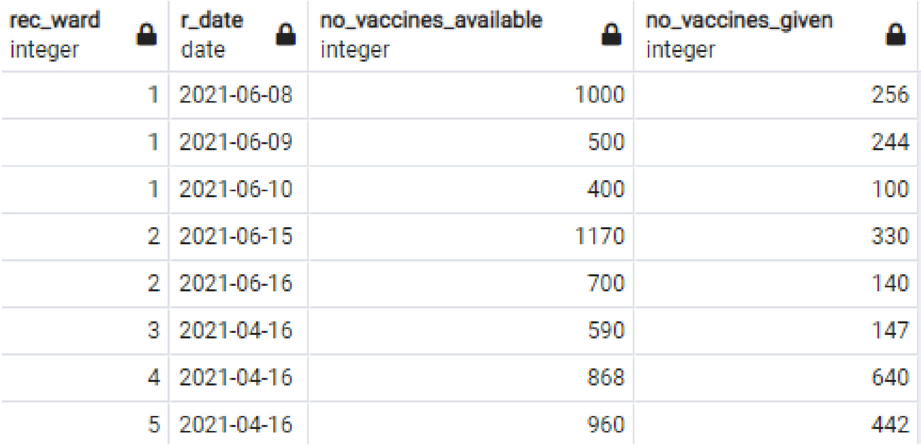
■ SQL for Creating Table Record:

```
CREATE TABLE records (
        rec_ward INTEGER REFERENCES v_center (ward_num) ON DELETE CASCADE,
        r_date DATE,
        no_vaccines_available INTEGER,
        no_vaccines_given INTEGER
);
```

■ SQL for Inserting Records in Table ‘Record’:

```
INSERT INTO records VALUES(1,'2021-06-08', 1000, 256);
INSERT INTO records VALUES(1,'2021-06-09', 500, 244);
INSERT INTO records VALUES(1,'2021-06-10', 400, 100);
INSERT INTO records VALUES(2,'2021-06-15', 1170, 330);
INSERT INTO records VALUES(2,'2021-06-16', 700, 140);
INSERT INTO records VALUES(3,'2021-04-16', 590, 147);
INSERT INTO records VALUES(4,'2021-04-16', 868, 640);
INSERT INTO records VALUES(5,'2021-04-16', 960, 442);
```

■ Resulting Table:



rec_ward integer	r_date date	no_vaccines_available integer	no_vaccines_given integer
1	2021-06-08	1000	256
1	2021-06-09	500	244
1	2021-06-10	400	100
2	2021-06-15	1170	330
2	2021-06-16	700	140
3	2021-04-16	590	147
4	2021-04-16	868	640
5	2021-04-16	960	442

Figure 13. Record Table

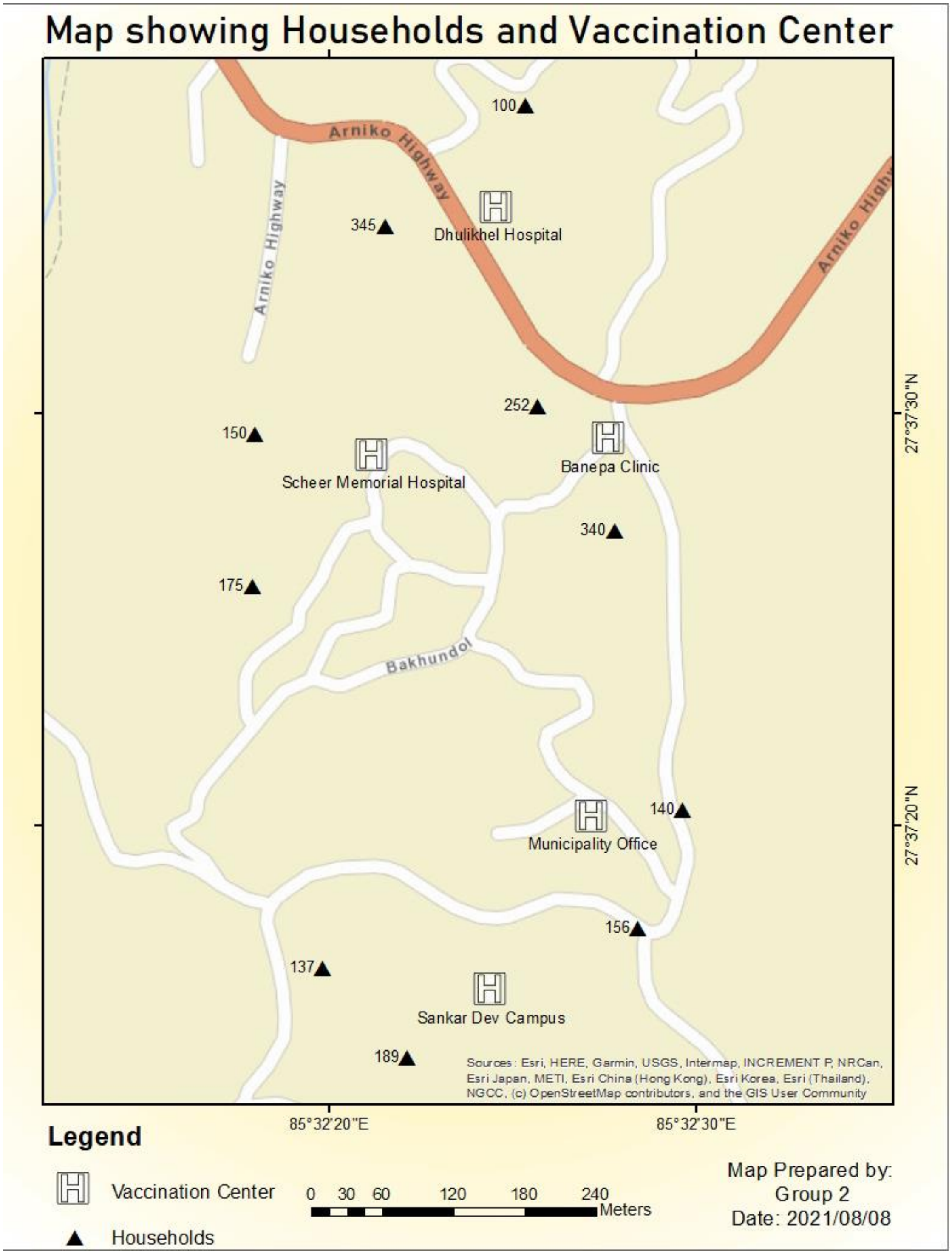


Figure 14. Map Showing Vaccination Centers and Households

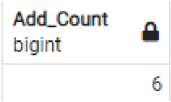
(Map Prepared on ArcMAP based on Database created on PostgreSQL)

2.3 Queries based on COVID Vaccine Distribution Database

(A) Count the number of households located within 100 m distance from Vaccination Center.

■ SQL:

```
SELECT COUNT(*) AS "Add_Count" FROM Address a,v_center v
WHERE ST_DWithin(a.geom, v.geom, 100);
```



Add_Count bigint
6

Figure 15. Query Output (a)

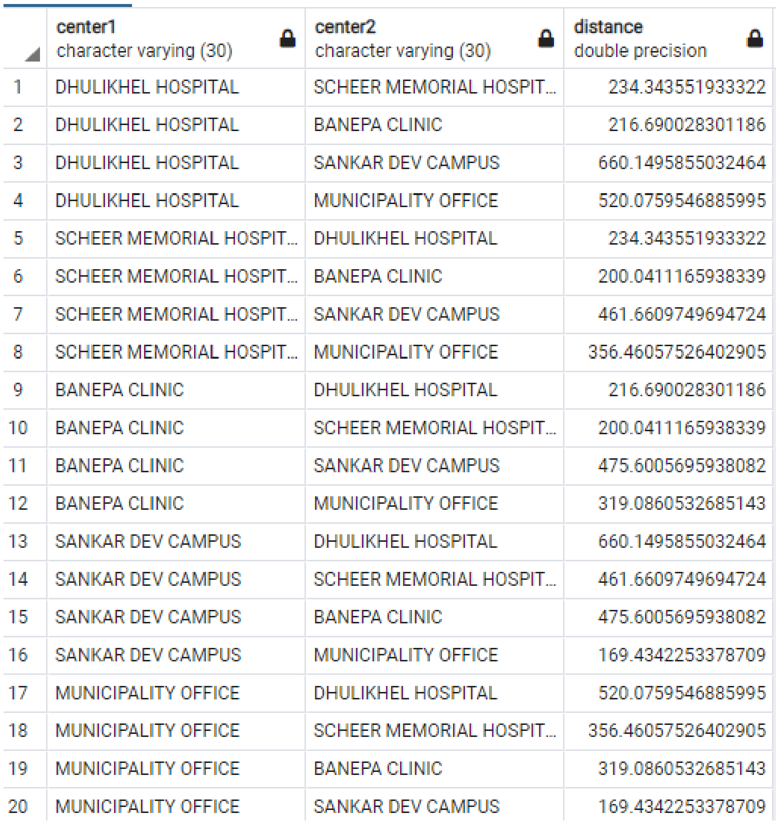
■ Result:

→ Six households are within a distance of 100 m from Vaccination Centers.

(B) Distance of each Vaccination Center from other.

■ SQL:

```
SELECT c1.center_name AS Center1, c2.center_name AS Center2,
ST_DISTANCE(ST_CENTROID(c1.geom), ST_CENTROID(c2.geom)) AS Distance
FROM v_center c1, v_center c2
WHERE c1.ward_num != c2.ward_num;
```



	center1 character varying (30)	center2 character varying (30)	distance double precision
1	DHULIKHEL HOSPITAL	SCHEER MEMORIAL HOSPIT...	234.343551933322
2	DHULIKHEL HOSPITAL	BANEPA CLINIC	216.690028301186
3	DHULIKHEL HOSPITAL	SANKAR DEV CAMPUS	660.1495855032464
4	DHULIKHEL HOSPITAL	MUNICIPALITY OFFICE	520.0759546885995
5	SCHEER MEMORIAL HOSPIT...	DHULIKHEL HOSPITAL	234.343551933322
6	SCHEER MEMORIAL HOSPIT...	BANEPA CLINIC	200.0411165938339
7	SCHEER MEMORIAL HOSPIT...	SANKAR DEV CAMPUS	461.6609749694724
8	SCHEER MEMORIAL HOSPIT...	MUNICIPALITY OFFICE	356.46057526402905
9	BANEPA CLINIC	DHULIKHEL HOSPITAL	216.690028301186
10	BANEPA CLINIC	SCHEER MEMORIAL HOSPIT...	200.0411165938339
11	BANEPA CLINIC	SANKAR DEV CAMPUS	475.6005695938082
12	BANEPA CLINIC	MUNICIPALITY OFFICE	319.0860532685143
13	SANKAR DEV CAMPUS	DHULIKHEL HOSPITAL	660.1495855032464
14	SANKAR DEV CAMPUS	SCHEER MEMORIAL HOSPIT...	461.6609749694724
15	SANKAR DEV CAMPUS	BANEPA CLINIC	475.6005695938082
16	SANKAR DEV CAMPUS	MUNICIPALITY OFFICE	169.4342253378709
17	MUNICIPALITY OFFICE	DHULIKHEL HOSPITAL	520.0759546885995
18	MUNICIPALITY OFFICE	SCHEER MEMORIAL HOSPIT...	356.46057526402905
19	MUNICIPALITY OFFICE	BANEPA CLINIC	319.0860532685143
20	MUNICIPALITY OFFICE	SANKAR DEV CAMPUS	169.4342253378709

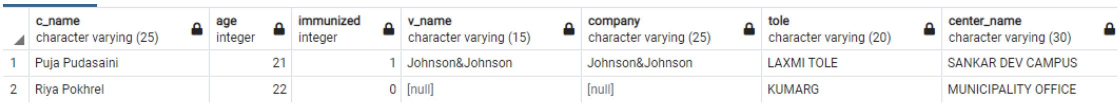
Figure 16. Query Output (b) Showing Distance of each Vaccination Center from Other

(C) Find all the name, vaccine used, vaccination center, address of citizens having Citizinship ID number = 84623 and Citizinship ID number = 13579.

■ SQL:

```
SELECT c_name, age, immunized, v_name, company, tole, center_name FROM
v_center JOIN (address JOIN  (recipient LEFT JOIN vaccine ON recipient.vaccine_id =
vaccine.v_id) ON recipient.house_no = address.house_num) ON address.ward_no =
v_center.ward_num
WHERE c_id = 84623 or c_id = 13579;
```

■ Result:



	c_name character varying (25)	age integer	immunized integer	v_name character varying (15)	company character varying (25)	tole character varying (20)	center_name character varying (30)
1	Puja Pudasaini	21	1	Johnson&Johnson	Johnson&Johnson	LAXMI TOLE	SANKAR DEV CAMPUS
2	Riya Pokhrel	22	0	[null]	[null]	KUMARG	MUNICIPALITY OFFICE

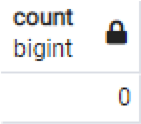
Figure 17. Query Output (c)

(D) Find the number of citizens from Ward number 1 who are not immunized.

■ SQL:

```
SELECT COUNT(*) FROM RECIPIENT JOIN (ADDRESS JOIN V_CENTER ON
V_CENTER.WARD_NUM = ADDRESS.WARD_NO) ON ADDRESS.HOUSE_NUM =
RECIPIENT.HOUSE_NO WHERE V_CENTER.WARD_NUM = 1 AND
RECIPIENT.IMMUNIZED = 0;
```

■ Result:



count bigint
0

Figure 18. Query Output (d)

→ Everyone on Ward number 1 is vaccinated.

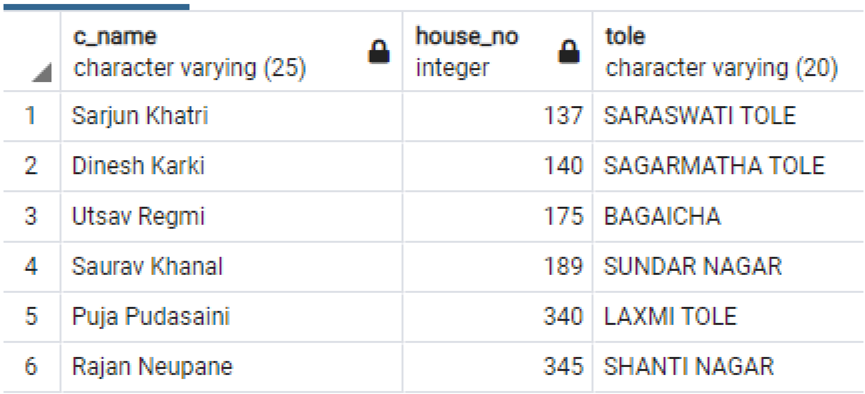
(E) Find out the name and house number of all the citizens in Banepa Municipality who are vaccinated in ascending order of house number.

→ Since, this is the database of Banepa Municipality, following SQL can be adopted.

■ SQL:

```
SELECT recipient.c_name, recipient.house_no, address.tole FROM recipient JOIN
(address JOIN v_center ON v_center.ward_num = address.ward_no) ON
```

```
address.house_num=recipient.house_no WHERE recipient.immunized = 1 ORDER BY
recipient.house_no ASC;
```



	c_name character varying (25)	house_no integer	tole character varying (20)
1	Sarjun Khatri	137	SARASWATI TOLE
2	Dinesh Karki	140	SAGARMATHA TOLE
3	Utsav Regmi	175	BAGAICHA
4	Saurav Khanal	189	SUNDAR NAGAR
5	Puja Pudasaini	340	LAXMI TOLE
6	Rajan Neupane	345	SHANTI NAGAR

Figure 19. Query Output (e)

3. CONCLUSION AND RECOMMENDATIONS

To sum up, this is a simple paradigm of the greater possibilities and endeavors. This database of COVID Vaccine Distribution dealt with creating the sample database of the information implanted with the spatial relation in the project which helps us for the digital representation of the spatial data as well. Expanding it on a large scale, it’s existence is vital in nationwide sector in present context of Nepal. Recording the data in digital format using database platforms and relating them spatially by using more platforms like ArcGIS, etc. can help in the systematic management of the whole nation data and also help in the further decision-making processes.

Nevertheless, with the completion of this project we have learned about designing the database, the real-world application of database management and its greater possibilities in diversified sectors. Along with this, it aided in harnessing our skills and guides for further curiosity and explorations.

4. REFERENCES

PostGIS, T. (n.d.). *PostGIS — Spatial and Geographic Objects for PostgreSQL*. Retrieved August 9, 2021, from https://postgis.net/

PostgreSQL, P. G. D. (2021, August 9). *PostgreSQL*. PostgreSQL. https://www.postgresql.org/

Ramakrisnan, R., & Gehrke, J. (n.d.). *Database Management Systems* (Second). McGraw Hill Education Pvt Ltd. www.learnengineering.in

Weak entities to Tables. (n.d.). Retrieved August 10, 2021, from https://www.brainbell.com/tutors/php/php_mysql/Step_2_Convert_weak_entities_to_tables.html