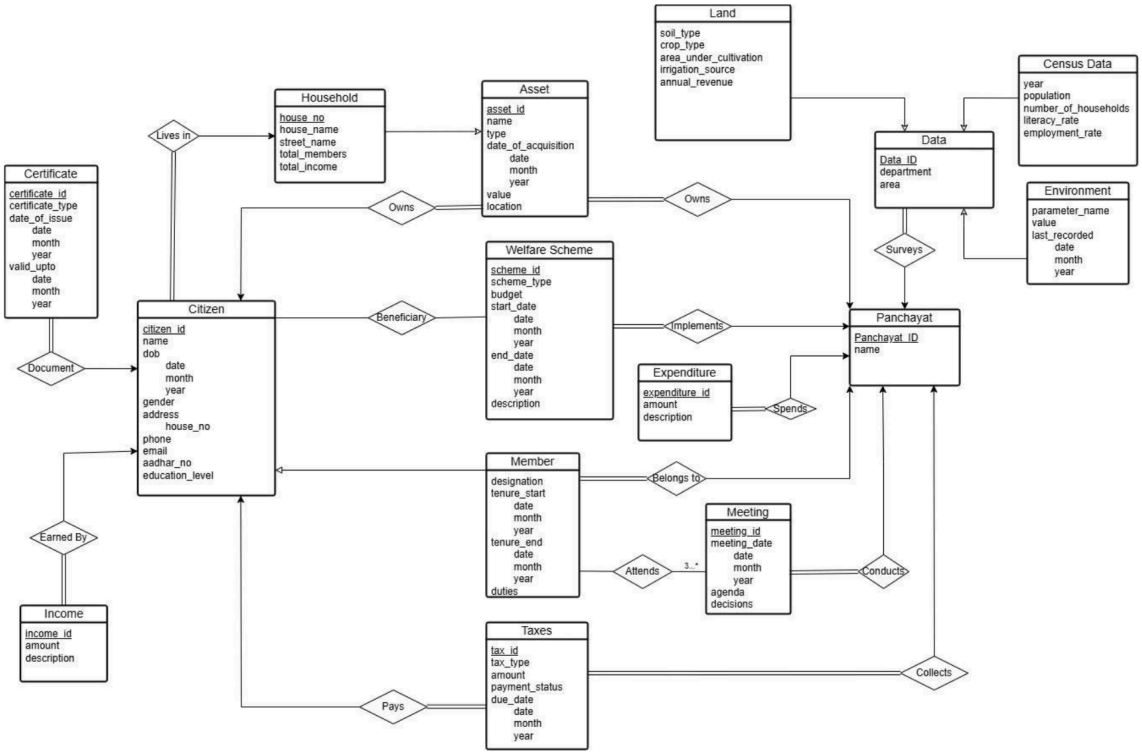
SQL Queries



The ER Diagram submitted in the Assignment 1, briefly encaptures the essence of a Panchayat, from its functioning to its data keeping (excluding a few details). It was designed keeping in mind the concepts of designing the ER Diagram, and ensuring that each of the entity, and its attributes with its relationships with other entities capture meaningful relations between the stored data to give better insights to the end viewer.

For the Assignment 2, as we were supposed to show few queries in the lab, the ER Diagram was trimmed down but cutting short a few entities, to reduce the complexity, but however keeping the integrity of the data the same.

Unlike in the ER Diagram, the relationships in the PostgreSQL were mostly captured using ***Foreign Key*** along with the ***Primary Key*** of the entities. Relationships with attributes were represented as separate table in the PostgreSQL.

Following relations were created :

```sql
-- table for citizens
CREATE TABLE citizens(
   citizen_id INT PRIMARY KEY,
   name VARCHAR(500),
   gender VARCHAR(10),
   dob DATE,
   house_no INT,
   phone CHARACTER VARYING(10) UNIQUE CHECK (phone ~ '^\d{10}$'),
   aadhar_no CHARACTER VARYING(12) UNIQUE CHECK (aadhar_no ~ '^\d{12}$'),
   edu_level VARCHAR(100)
);

-- table for households
CREATE TABLE households(
   house_no INT PRIMARY KEY,
   asset_id INT,
   address VARCHAR(500),
   total_members INT,
   total_income INT
);

-- table for lands
CREATE TABLE lands (
   asset_id INT PRIMARY KEY,
   soil_type VARCHAR(100),
   crop_type VARCHAR(100),
   area_in_acres INT,
   revenue INT
);

-- table for assets
CREATE TABLE assets(
   asset_id INT PRIMARY KEY,
   panchayat_owner_id INT,
   citizen_owner_id INT,
   type VARCHAR(100),
   date_of_acquisition DATE,
   value INT,
   location VARCHAR(500)
```

```sql
);

-- table for welfare_schemes
CREATE TABLE welfare_schemes (
   scheme_id INT PRIMARY KEY,
   scheme_type VARCHAR(100),
   budget INT,
   description VARCHAR(500)
);

-- table for beneficiaries
CREATE TABLE beneficiaries (
   citizen_id INT,
   scheme_id INT,
   enrollment_date DATE,
   PRIMARY KEY (citizen_id, scheme_id)
);

-- table for panchayat_members
CREATE TABLE panchayat_members(
   citizen_id INT,
   panchayat_id INT,
   designation VARCHAR(50),
   tenure_start DATE,
   tenure_end DATE,
   duties VARCHAR(500),
   PRIMARY KEY (citizen_id, panchayat_id)
);

-- table for panchayats
CREATE TABLE panchayats(
   panchayat_id INT PRIMARY KEY,
   name VARCHAR(100) UNIQUE
);

-- table for surveys
CREATE TABLE surveys(
   survey_id INT PRIMARY KEY,
   panchayat_id INT,
   department VARCHAR(100),
```

```
    area VARCHAR(100)
);

-- table for census
CREATE TABLE census(
    survey_id INT,
    population INT,
    no_of_households INT,
    literacy_rate NUMERIC(5, 2),
    employment_rate NUMERIC(5,2)
);

-- table for environment
CREATE TABLE environment(
    survey_id INT,
    parameter_name VARCHAR(100),
    value NUMERIC,
    last_recorded DATE,
    PRIMARY KEY (survey_id, parameter_name)
);
```

To capture the essence of relationship and also the specializations wherever necessary, modifications to the table were made accordingly:

```
---------------------- MAKE NECESSARY ALTERATIONS ---------------------
ALTER TABLE assets
ADD CONSTRAINT fk_assets_panchayat_owner_id FOREIGN KEY
(panchayat_owner_id) REFERENCES panchayats(panchayat_id),
ADD CONSTRAINT fk_assets_citizen_owner_id FOREIGN KEY (citizen_owner_id)
REFERENCES citizens(citizen_id);
ADD CONSTRAINT check_owner_constraint CHECK (
    (citizen_owner_id IS NOT NULL AND panchayat_owner_id IS NULL) OR
    (citizen_owner_id IS NULL AND panchayat_owner_id IS NOT NULL)
);

ALTER TABLE households
ADD CONSTRAINT fk_households_asset_id FOREIGN KEY (asset_id) REFERENCES
assets(asset_id);

ALTER TABLE lands
```

```
ADD CONSTRAINT fk_lands_asset_id FOREIGN KEY (asset_id) REFERENCES
assets(asset_id);

ALTER TABLE citizens
ADD CONSTRAINT fk_citizens_house_no FOREIGN KEY (house_no) REFERENCES
households(house_no);

ALTER TABLE beneficiaries
ADD CONSTRAINT fk_beneficiaries_citizen_id FOREIGN KEY (citizen_id)
REFERENCES citizens(citizen_id),
ADD CONSTRAINT fk_beneficiaries_scheme_id FOREIGN KEY (scheme_id)
REFERENCES welfare_schemes(scheme_id);

ALTER TABLE panchayat_members
ADD CONSTRAINT fk_panchayat_members_citizen_id FOREIGN KEY (citizen_id)
REFERENCES citizens(citizen_id),
ADD CONSTRAINT fk_panchayat_members_panchayat_id FOREIGN KEY
(panchayat_id) REFERENCES panchayats(panchayat_id);

ALTER TABLE surveys
ADD CONSTRAINT fk_surveys_panchayat_id FOREIGN KEY (panchayat_id)
REFERENCES panchayats(panchayat_id);

ALTER TABLE census
ADD CONSTRAINT fk_census_survey_id FOREIGN KEY (survey_id) REFERENCES
surveys(survey_id);

ALTER TABLE environment
ADD CONSTRAINT fk_environment_survey_id FOREIGN KEY (survey_id) REFERENCES
surveys(survey_id);
```

This ensured that all the PK, FK, relations and specializations were captured.
The database was then filled with dummy data (note that since data was dummy there are real life scenario inconsistencies in the data), and then the given queries were run to get the desired output:

```
-- A. Show names of all citizens who holds more than 1 acre of land
SELECT c.name
FROM citizens c
JOIN assets a ON c.citizen_id = a.citizen_owner_id
JOIN lands l ON a.asset_id = l.asset_id
WHERE l.area_in_acres > 1;
```

```sql
-- B. Show name of all girls who study in school with household income less than 1 Lakh per year
SELECT c.name
FROM citizens c
JOIN households h ON c.house_no = h.house_no
WHERE c.gender = 'Female'
 AND (c.edu_level = '10th' OR c.edu_level = '12th')
 AND h.total_income < 100000;

-- C. How many acres of land cultivate rice
SELECT SUM(l.area_in_acres) AS total_acres_rice
FROM lands l
WHERE l.crop_type = 'Rice';

-- D. Number of citizens who are born after 1.1.2000 and have educational qualification of 10th class
SELECT COUNT(*)
FROM citizens c
WHERE c.dob > '2000-01-01' AND c.edu_level = '10th';

-- E. Name of all employees of panchayat who also hold more than 1 acre land
SELECT c.name
FROM citizens c
JOIN panchayat_members pm ON c.citizen_id = pm.citizen_id
JOIN assets a ON c.citizen_id = a.citizen_owner_id
JOIN lands l ON a.asset_id = l.asset_id
WHERE l.area_in_acres > 1;

-- F. Name of the household members of Panchayat Pradhan
SELECT h.total_members
FROM households h
JOIN citizens c ON c.house_no = h.house_no
JOIN panchayat_members pm ON c.citizen_id = pm.citizen_id
WHERE pm.designation = 'Pradhan';

-- G. Total number of street light assets installed in a particular locality named Phulera that are installed in 2024
SELECT SUM(a.value) AS total_value
```

```sql
FROM assets a
WHERE a.type = 'Street Light'
 AND a.location = 'Phulera'
 AND EXTRACT(YEAR FROM a.date_of_acquisition) = 2024;

-- H. Number of vaccinations done in 2024 for the children whose educational qualification is class 10
SELECT COUNT(*)
FROM beneficiaries b
JOIN citizens c ON b.citizen_id = c.citizen_id
JOIN welfare_schemes ws ON b.scheme_id = ws.scheme_id
WHERE ws.scheme_type = 'Vaccination'
 AND EXTRACT(YEAR FROM b.enrollment_date) = 2024
 AND c.edu_level = '10th';

-- I. Total number of births of boy child in the year 2024
SELECT COUNT(*)
FROM citizens c
WHERE c.gender = 'Male' AND EXTRACT(YEAR FROM c.dob) = 2024;

-- J. Number of citizens who belong to the household of at least one panchayat employee.
SELECT SUM(h.total_members)
FROM households h
WHERE h.house_no IN (
   SELECT DISTINCT c.house_no
   FROM citizens c
   JOIN panchayat_members pm ON c.citizen_id = pm.citizen_id
);
```