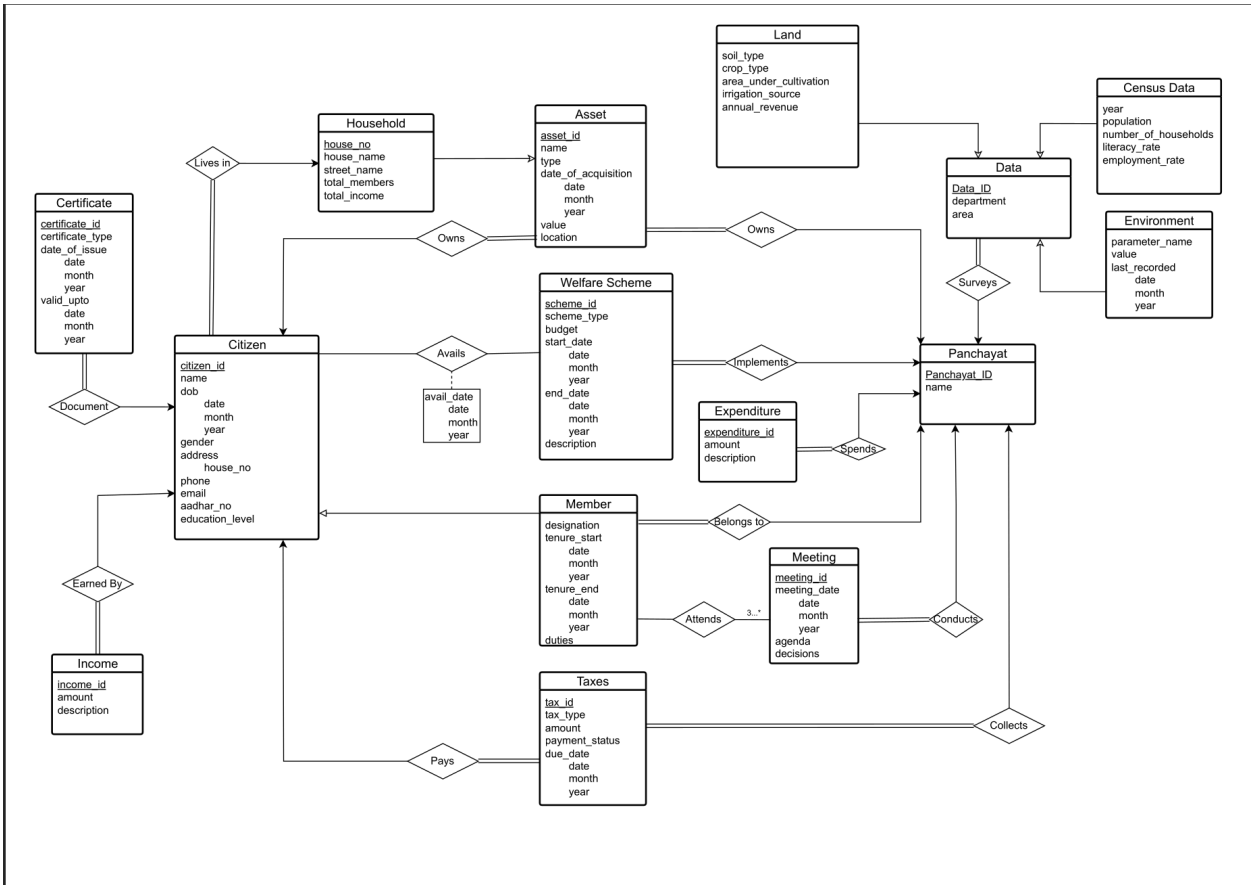
SQL Queries



The ER Diagram, briefly captures the essence of working of the Panchayat.
Let us first understand all the entities involved in the ER :

1. Citizen : To represent all the people living in the village
2. Household : To represent all the houses in the village (Specialization of Asset)
3. Certificate : To keep a track of all the government issued certificates, of the citizens
4. Income : To track all the income sources of the people, so that we can tax them based on the income
5. Asset : All the things that a citizen can own ranging from land, bike, car, etc.
6. Welfare Scheme : All the welfare schemes implemented by the local governing body in the village
7. Member : All the members of the Panchayat committee (Specialization of Citizen)
8. Taxes : To track all the taxations of let's say Tax Department
9. Panchayat : The governing body if the village (here only one will be present, on account of scalability this is a feasible entity, with few other changes to overall ER Diagram, will scale this to a multiple Panchayat system)

10. Expenditure : To track all the funds that are spent by the Panchayat over various kinds of things
11. Meeting : To track all the official meetings done by the Panchayat members for discussing official working things
12. Data : To represent all the data that the governing body has, can be of various types
13. Land : All the land data (Specialization of Data)
14. Census Data : All the census data (Specialization of Data)
15. Environment : All the environmental data (Specialization of Data)

Assume that we have created SQL Tables for all of these entities, and now we will modify the tables to introduce constraints and FK in them. So let us see how we will represent the relationships and the constraints on them in the SQL Table form:

1. Document : is a one to many relationship from citizen to certificate, and certificate entity has complete participation. This can be modeled by introducing a FK of citizen_id in the certificate entity and assuring that it never takes NULL value.
2. Earned By : is a one to many relationship from citizen to income, and income has complete participation. This can be modeled by introducing a FK of citizen_id in the income entity and then assuring it never takes NULL value.
3. Lives In : is a one to many relationship from household to citizens, and citizens have a complete participation. This can be modeled by making sure that house_no in citizens can never be NULL, and since it serves as the FK, unless that house exists in the household, the citizen cannot have that in its attributes.
4. Owns : is a one to many relationship from citizens, panchayat to assets, and the asset has complete participation. This can be modelled by introducing two FK as citizen_id and panchayat_id, as the asset can either be owned by a citizen or panchayat, and making sure that both of the foreign keys are not NULL and not NOT NULL at the same time.
5. **Beneficiary** : is a many to many relationship from welfare scheme to citizen, this can be modeled by keeping another **table for the beneficiaries** which will have 3 columns with 2 FK from citizen_id and scheme_id and additional relation attribute avail_date.
6. Pays : is a one to many relationship from citizen to taxes, with taxes having a complete participation. This can be modelled by introducing FK of citizen_id in the taxes table assuring that there are no NULL entries.
7. Collects : is a one to many relationship from panchayat to taxes, with taxes having a complete participation. This can be modelled by introducing FK of panchayat_id in the taxes table assuring that there are no NULL entries.
8. Conducts : is a one to many relationship from panchayat to meeting, with meeting having complete participation. This can be modelled by introducing a FK

of panchayat_id in the meeting schema, and making sure that it does not have any NULL value.

9. **Attends :** is a many to many relationship with a constraint that every meeting should have at least 3 members to participate, so this needs a **table for modelling** with two attributes namely meeting_id which will be a FK from the meeting, and attendees (which will be a multivalued attribute), which will contain a set of FKs with a constraint of no. >= 3.
10. Belongs to : is a one to many relationship from panchayat to member and is a complete participation from the member side. This can be modelled using the FK of panchayat_id in the member table and making sure that it never takes NULL values.
11. Spends : is a one to many relationship from panchayat to expenditure and is a complete participation from the expenditure side. This can be modelled using the FK of panchayat_id in the expenditure table and making sure that it never takes NULL values.
12. Implements : is a one many relationship from panchayat to welfare schemes, with complete participation from the welfare_scheme side. This can be modelled using the FK of panchayat_id in the welfare_scheme table and making sure that it never takes NULL values.
13. Surveys : is a one to many relationship from panchayat to data, with complete participation from the data side, this can be modelled using the FK of panchayat_id in the data and making sure that it does not contain any NULL values.

For all the specialization classes, as mentioned in the slides, we can just introduce the PK of the superclass within the specialised class, along with the additional attributes.

From the scalability side of the ER Diagram, the panchayat as an entity will play a major role, so it can be looked in such a way that all the entities can be aggregated with the panchayat entity so that we can scale this model for multi panchayat system, currently it is just handling a single panchayat system.