**Design decisions and assumptions:**

* As mentioned in the specification we tried to maintain the number of entities around the number specified but to take make the project fulfill our business requirements we had to cross 20.
* We tried to limit the number of attributes where ever we can and in few entities, we had to cross 10 to keep it as close as much to the real-world project
* We tried to maintain the integrity of the tables and used id’s where ever possible to make the referencing easier
* We tried to connect link tables logically with the keys
* TableNames & Descriptions
  + **Branch**: Branch table stores the all the branch information of the e-nsure insurance company and it is given with the primary key called branch\_id to reference it easily with other tables and also given other attributes which stores other information of particular branch.
  + **Employee**: Employee table stores the data of the employees working for the company and are differentiated their branch with referencing to branch\_id as the foreign key form the branch table.
  + **Roles:** We are using the roles table as a subtable where it stores the different types of roles available in the company and they are given with the role\_id as primary key and linked with employee table.
  + **Policy:** The policy table will hold the data of a policy and is given a policy number,
    - policy\_id as primary key, linked with customer table cust\_id as foreign key constraint and policy table also holds the policy start date, end date
    - due\_date to know before when the payment has to be done
    - policy\_premium is wher the amount of the policy is stored, policy\_time is the period of policy taken for
  + **Policy\_status:** We are manly using policy status table as an edit log where it stores the various stages that a policy has undergone. It has a primary key and foreign key as policy\_id form policy table , employee\_id to know which employee has modified the status,log the modified time with the time stamp.

|  |  |  |
| --- | --- | --- |
| **policy\_status\_id** | **Policy\_id** | **Property\_subtype** |
| 1 | 45 | Underwriter-approved |
| 2 | 45 | Underwriter-rejected |
| 3 | 45 | With agent |
| 4 | 25 | With agent |

* **Customer:** A customer can be either an individual or a business entity where and and an individual each type is assigned with an id which is crucial and linked with many other entities.
* **Person:** The person table is where it stores the each and every person details like name ,email, phone ,address etc. The person table also has an attribute called type which stores if the person is primary or secondary. The person table also holds the details of persons for the group policy(corporate). Thus, the person table is very crucial in storing the customers data.
* **Corporate:** As we have divided the customer in to 2 categories, individual & corporate. The corporate table holds the details of the corporate customers/business entities
* **Policy\_items :** we can say policy item table is the bridge that ways the policy table and the insurance type. The policy item table holds policy\_item\_id which is primary key for the table and policy\_id form the policy table as the foreign key, policy\_item\_name to store the policy name
* **Markup\_table:** We understand not every person needs the same insurance and not every person owns same type of property or vehicle. So we are using markup table to different properties of each home or auto. Group entity divides if it is home or auot and markup\_name for example gives location , flood areas, customer not having covered parking etc.. and each is given with the percentage which can be added to the base policy price by the rater.
* **Markup\_policy\_coverage\_item:** This table is a many to many table for markup-table and policy\_coverage\_item to know which coverage uses how many markups.
* **Policy\_item\_coverage:** This table acts as the many to many table for policy\_items and coverage entities.But, the reason why we had to show it in the ER diagram is ,it does more than being a bridge table where markup table is linked as well to understand the price variations from policy to policy.
* **Coverage :**  coverage table holds the different types of coverages an insurance company holds, home and auto are differentiated by the group attribute. There is min and max amount for a coverage where underwriter will able to decide the coverage amount depending on different markups and discounts.
* **Discount:** Discount table is given with an discount\_id which is primary key , group to understand if it is home or auto , discount name and min and max percentage which can be checked by the rater/underwriter and discount it to the base policy amount.
* **policy\_discounts:** This table acts as a many to many table to understand which discount is applied for which policy
* **Property:** The property table holds the all necessary attributes to be collected for a home
* **Automobile:**  This table holds the information and features of a vehicle.
* **Bills:**  Bills table is related to policy which holds the bill id and the policy\_id, bill amount, where as policy details can be fetched form the policy table.
* **Claims:** Claims is one big segment of our project, a claim can be claimed by either customer or the claimant who is 3rd party. And if it is customer it is linked to the policy\_item\_coverage table to understand for which policy the claim has been done.
* **Reserve:** Reserve table stores claim\_id as foreign key ,amount which stores the reserve amount and also the adjustment amount.
* **Claimant:** Claimant table stores the 3rd party customers data who has applied for claim
* **Claim\_possesions:** This table contains the possessions that are maid against a particular claim and the salvage amount is stored
* **Lossparties:** it contains the data of the lossparties and also the claim\_id which they have investigated on
* **Payments:** Payments table is linked to reserve. It handles the payment related information inregarding to claims only.
* **Claimstatus:** A claim undergo many phases, even though once a claim is settled it can be re-opened for many reasons. Claim status table maintain the logs of that information
* **Property\_characterstics:** Different properties has different characterstics. This table maintains all that data given with an id as primary key
* **Auto\_characterstics:** This table is similar to the property\_characterstics which holds the different properties of a vehicle
* **Subrogate :** As subrogation is for a claim it is linked with the claim\_id as the foreign key and company details are collected
* **Accidents :**
* **Timestamp:**