Assignment 3
Swapnil Darmora darmora2

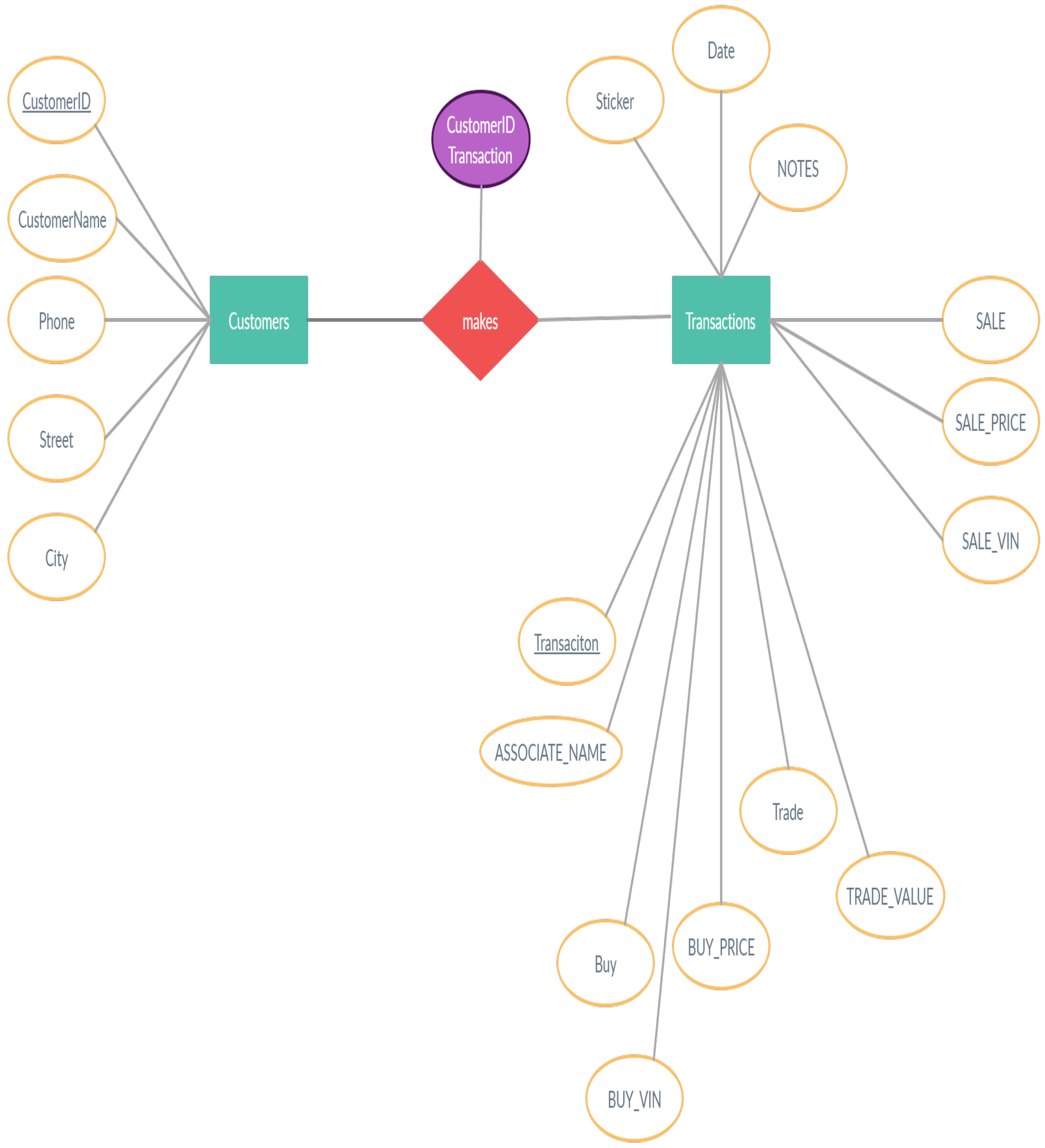
1) There are two tables provided in the file. The table customer has consistent data and the quality of data in the customer table is up to the mark. It has a transaction id which is associated with Transactions table id and it has been used as foreign key relation in the ER diagram for Customer entity as it is not exactly the attribute that belongs to the customer data.

   The data provided in the Transactions table is not up to the mark with some irregularity in the way data is represented in the table. The initial two columns Transaction and Associate Name are reasonable and the data for these two attributes is good enough. However moving on to the next column/attribute which is Buy, the value is Y,y which seems to be yes but for value no we have no,NO,n,-- and NULL values. The ideal situation would be to chose n if we are choosing y for yes. Similarly for the next column TRADE, although there are no irregularities, instead of NULL n can be used. For SALE we again have a combination of NULL,-- and no which can be avoided and instead we should be choosing n for values other than y.

   Additionally we could have one field that would depict the type of transaction if it's sale, buy or tradein.

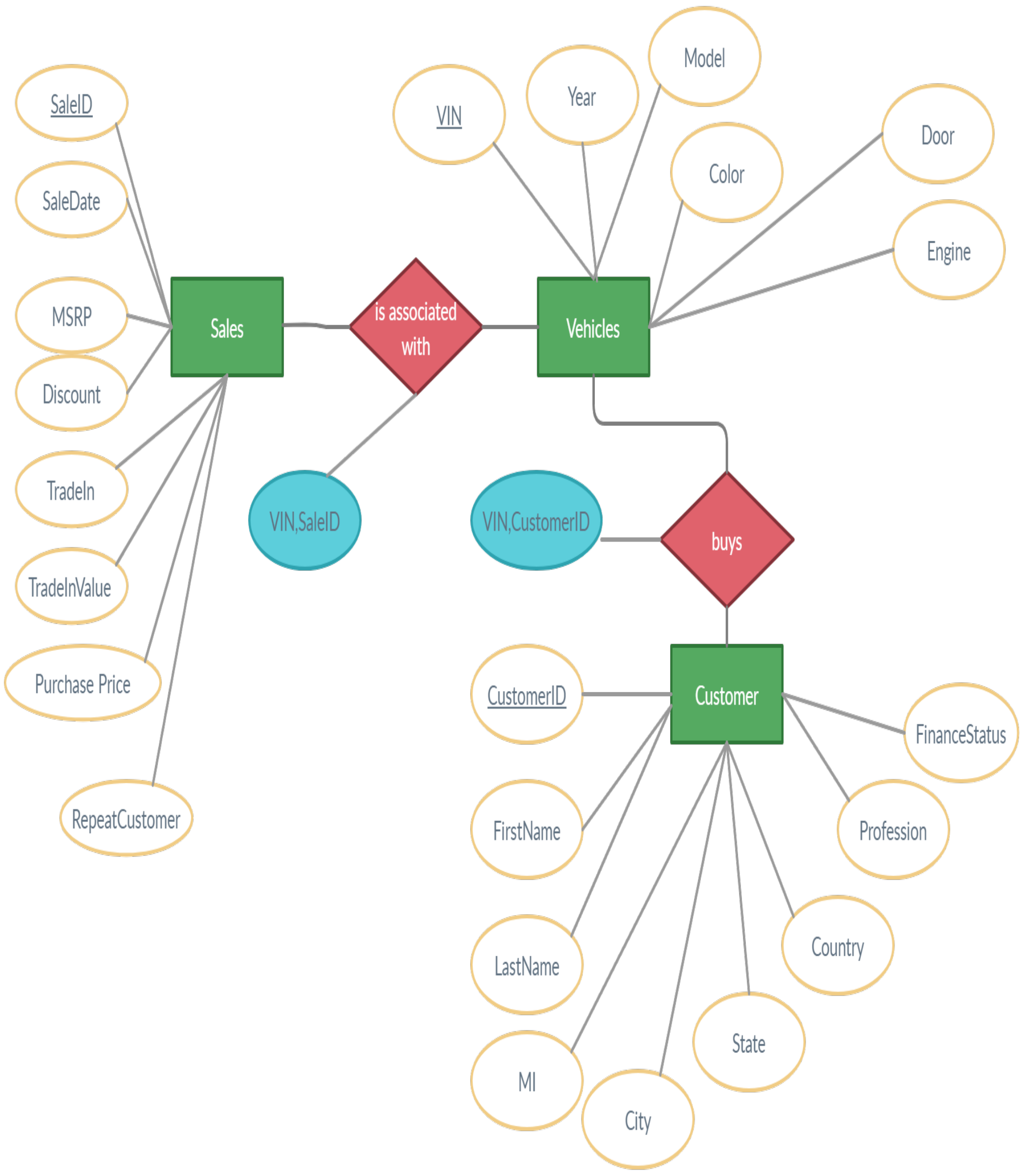
   Similarly for columns that contain price eg BUY_PRICE,TRADE_VALUE,STICKER,SALE_PRICE for values which cannot be populated there is a combination of N/A,NULL and blank value which could have been avoided. Date column is alright.

   However BUY_VIN and SALE_VIN again have NULL,-- and blank values for the values which are not applicable. Notes is repeated twice and should only be included once.



The table Customers and Transactions are translated into Entities and the columns as attributes in the above ER Diagram

2) The schema from assignment 1 with changes from instructor feedback. Initially four tables were created but the task could have been done using three tables only without any attribute changes. The updated schema has been included in the assignment folder as an excel file. The basic 3 tables are CustomerTable SalesTable and VehicleTable

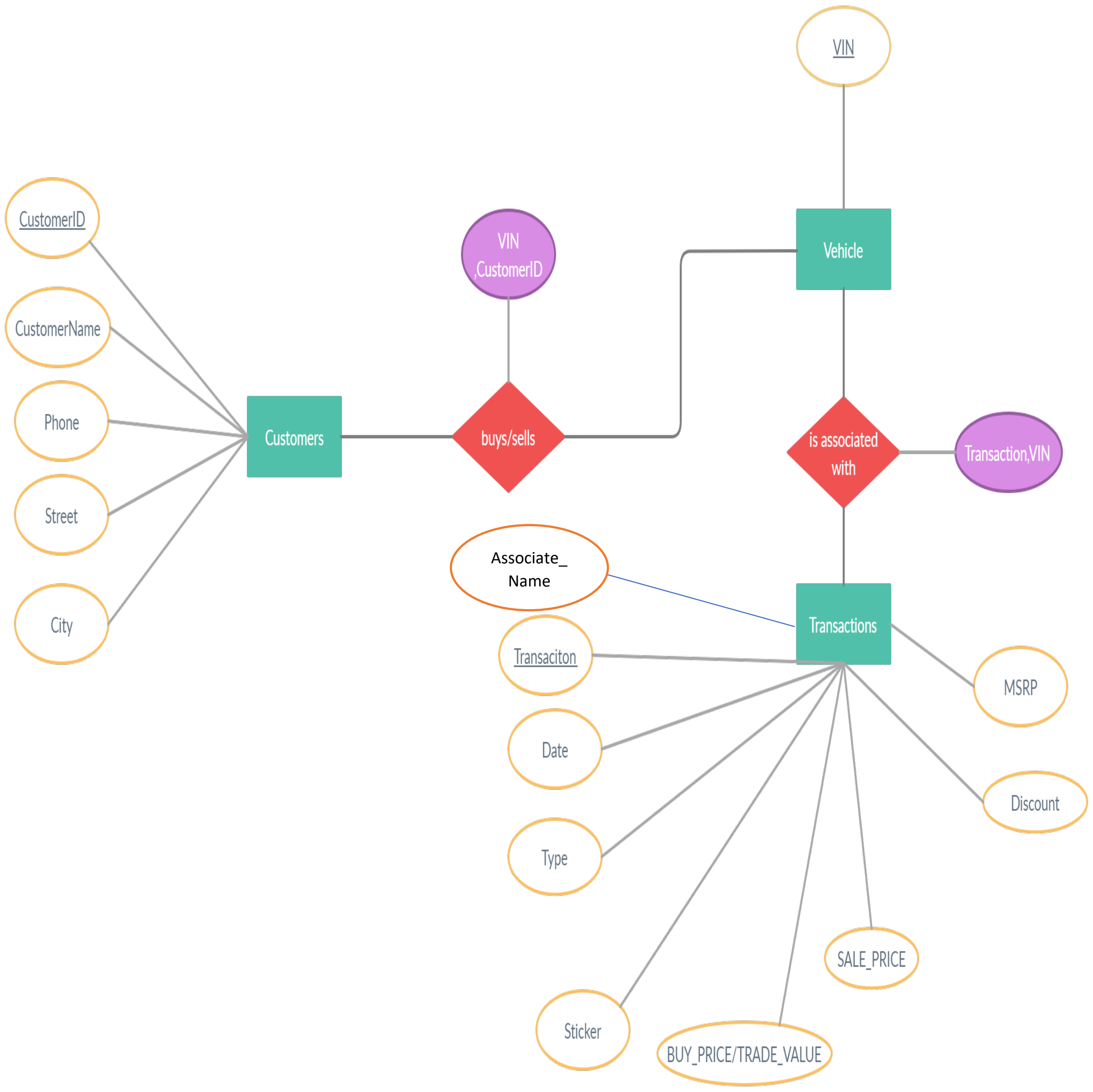


3) We have 3 entities in part 2 (Sales,Customer,Vehicle) and two entities(customer, Transaction) in part1

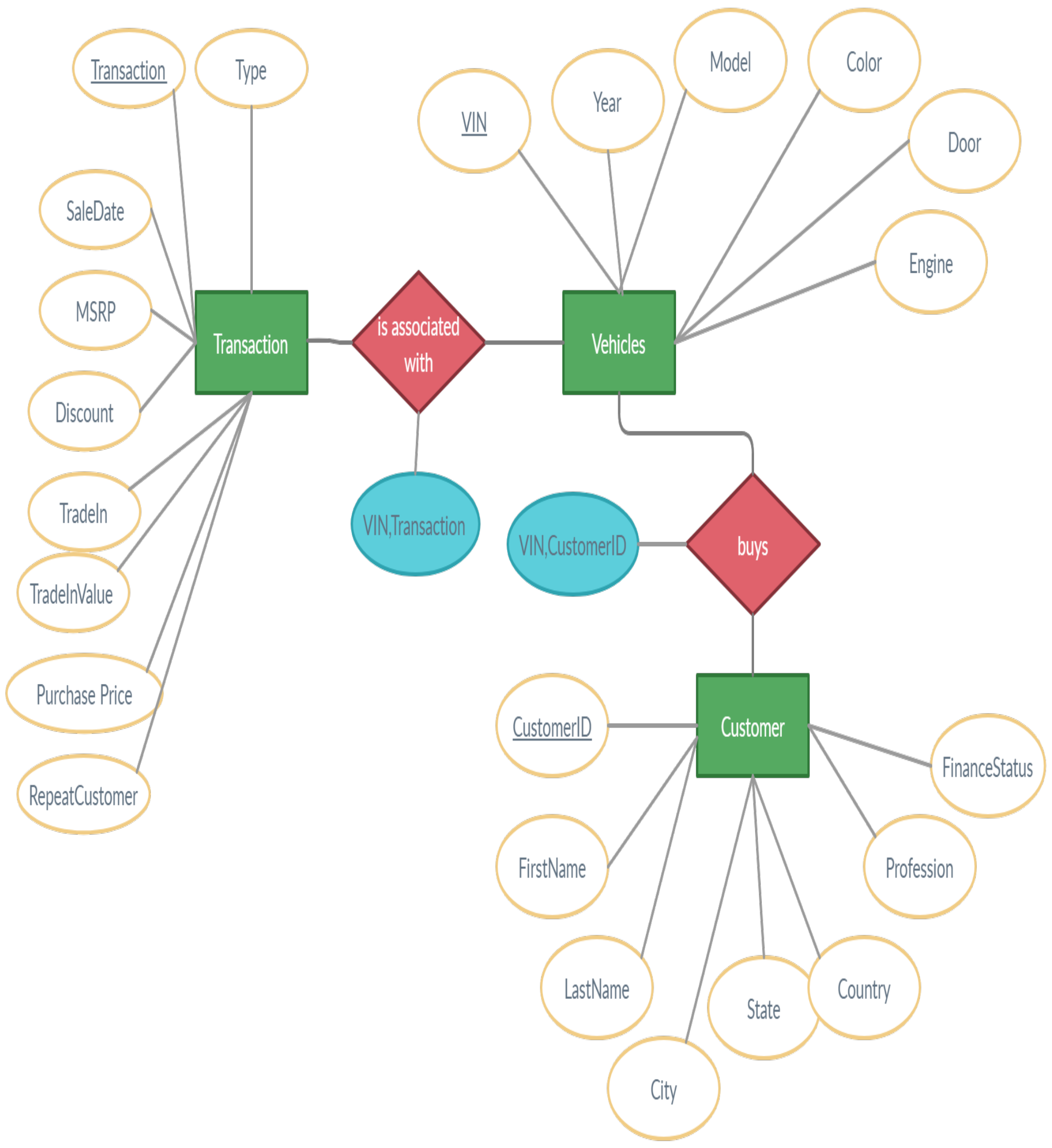
   (i)To integrate these schemas we firstly need to combine customer entity and merge attributes. In the first schema we have a relation between transaction and customer (we have transaction id column to establish this relationship) whereas in part2 we have a direct relation between VehicleVIN and CustomerID and an indirect relationship between sales and customer id. In order to merge these entities we will firstly make changes in Schema 1 which is described as following

   We will now create a new entity vehicle which contains VIN of the vehicle(it includes both BUYVIN and SELLVIN in TransactionsTable as the foreignkey for this would be transaction id. Also we will create a new foreignkey customer id in Vehicle to associate it with the customer. This will be obtained using TransactionID from both the tables and from there we will get a mapping from VIN to CustomerID

   Now the issue with the VIN to transaction mapping is that how will we resolve whether the VIN is buy VIN or sale VIN for that for transactions entity we will have a new attribute type(transaction type) which will replace (Buy,Sale and Trade attributes). Now the new attribute can have four values Buy,Sale,Buy Trade and Sale Trade. This is done so that we can identify whether a vehicle which has a Trade attribute is bought or sold. Similarly from data we can observe that BuyPrice and TradeValue are not populated simultaneously and hence they can be combined to form a single attribute. We will keep Sale Price as it is because it doesn't depend on other two attributes.

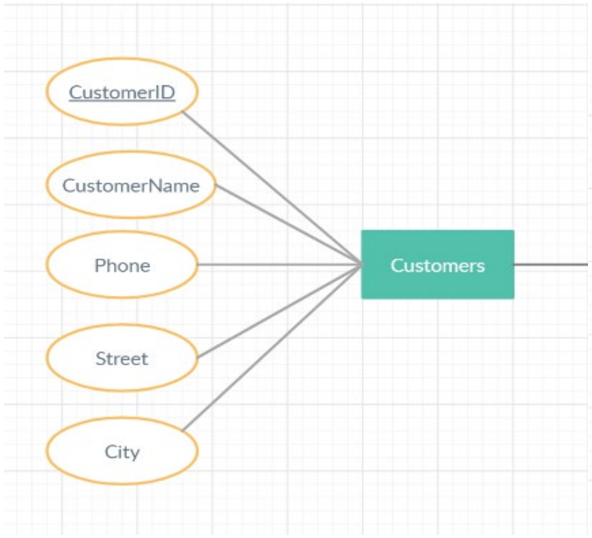


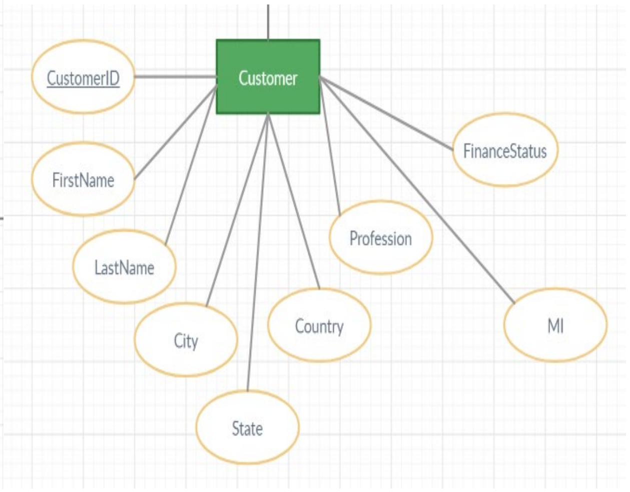
For schema2 we will rename Sales to Transaction and add type to it which will be Sale by default for the values present in Schema2. This is shown in the below picture

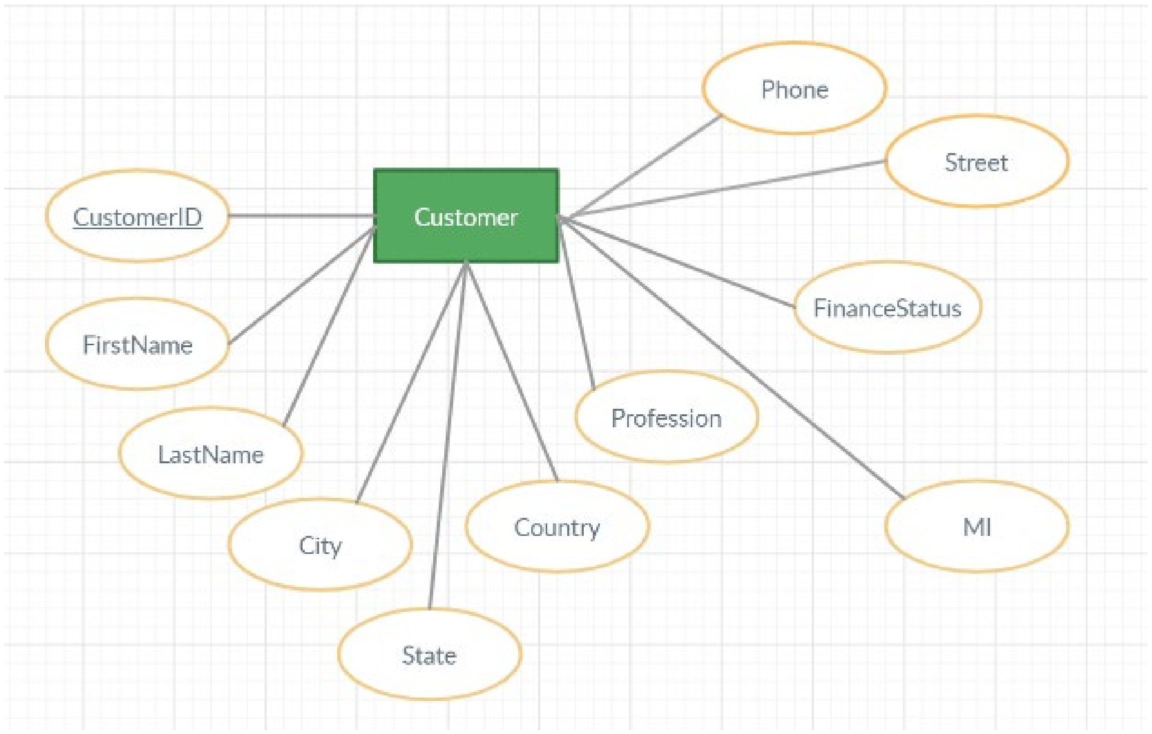


(ii)Now we have similar entities in both of the Schemas after changing/renaming them. Now we can merge attributes of similar Entities

Merging Customer Entity from both the schemas, we get following Customer Entity

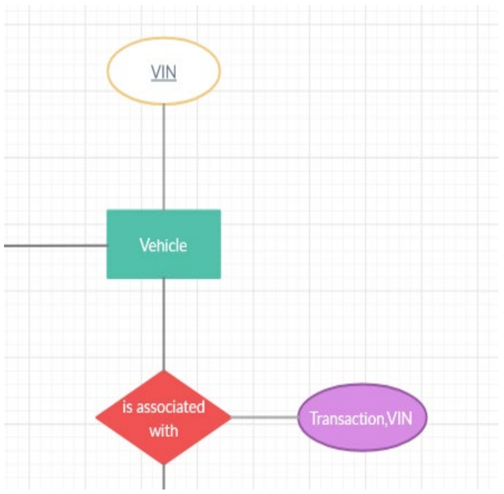


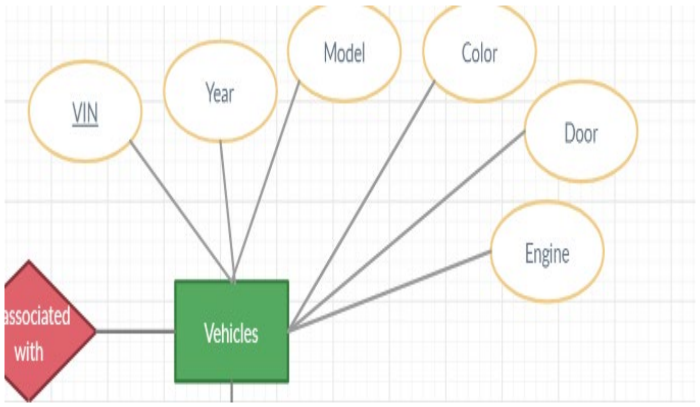


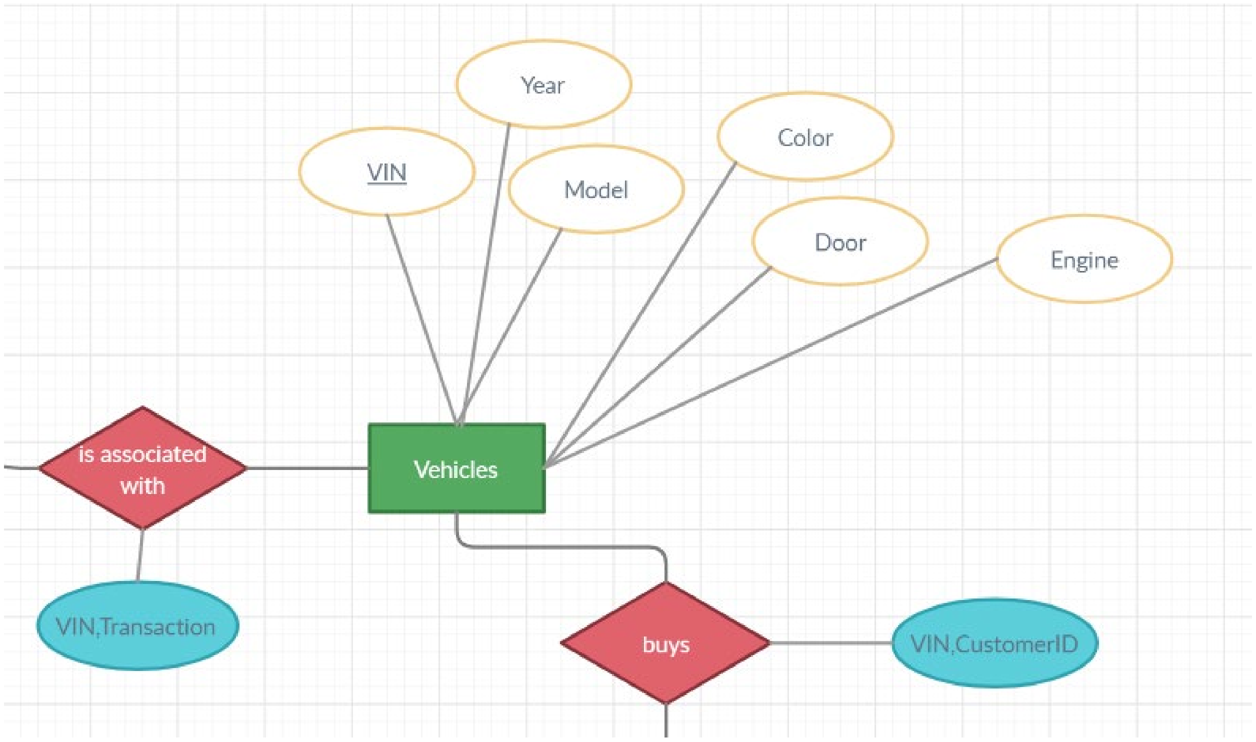


In the appended diagram, we have CustomerID taken from both the entities. Now customer name from Schema1 can be split into FirstName LastName and MiddleName just like Schema2 ensuring there is no data loss. Phone and Street is added from Schema1 whereas other fields of Customer from Schema2 are taken.

Merging vehicle entities

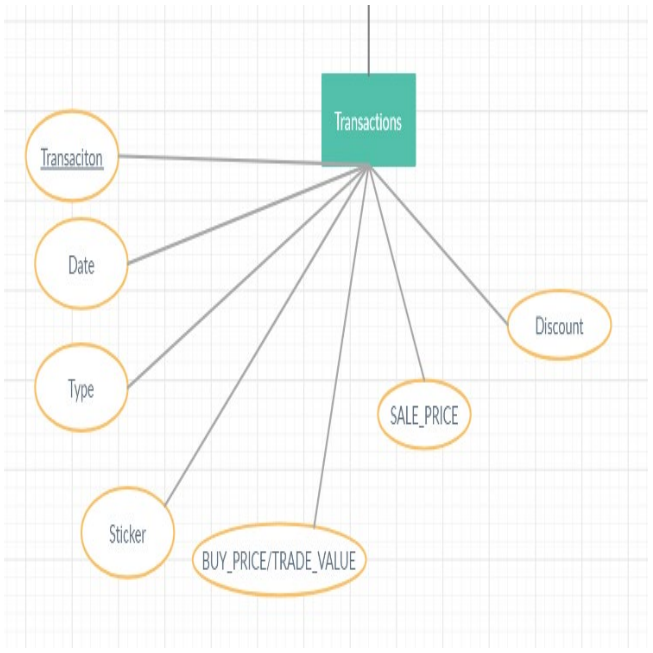


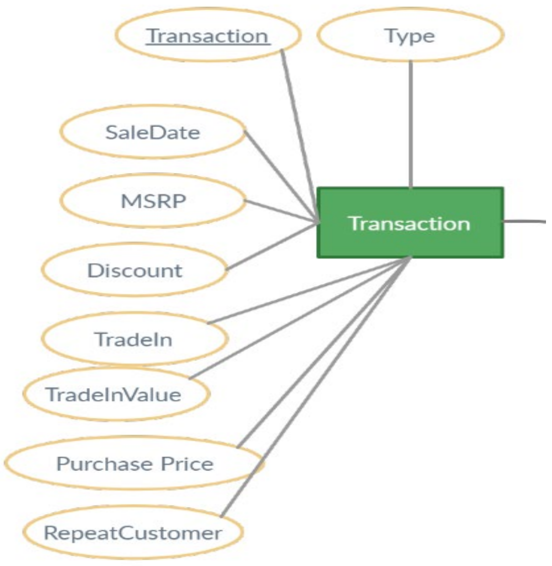




Merging Vehicle is very simple because we only had VIN in schema 1 and we also have VIN in schema2 Thus we can merge these values. If there are any duplicate values of VIN in Schema1 and Schema2 we will take the values from Schema2 for other attributes. If a VIN is only present in Schema1 we will

Now we will merge Transaction entities from both Schema

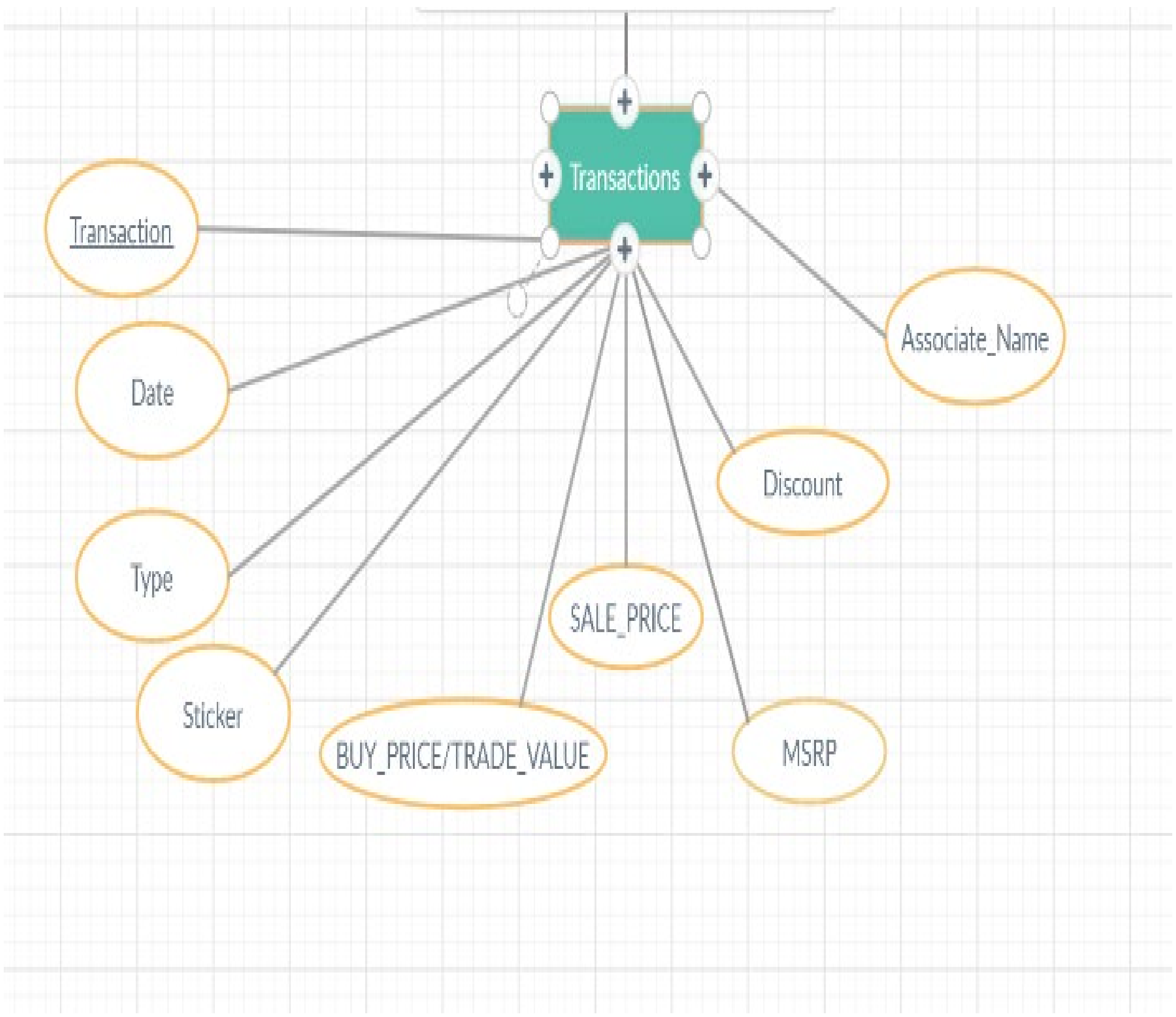




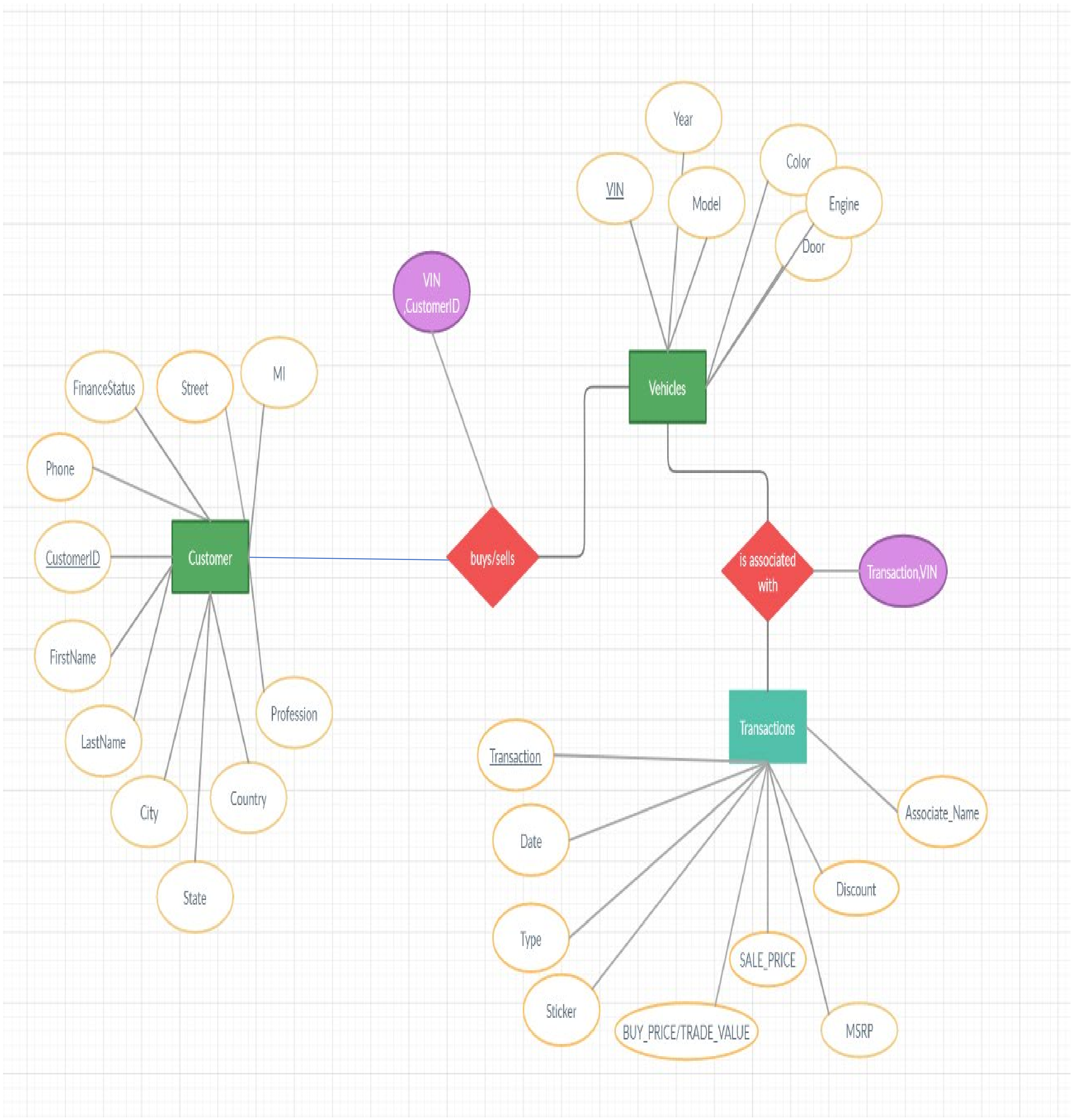
So Transacation attribute from both will be merged which is the primary key. Date and SaleDate will be merged. Type will also be merged Sticker will be taken from Schema1. Buy Price and TradeInValue fields will be merged and Type will be merged accordingly. We have previously discussed about Type attribute in part 1 of the integration process already. SalePrice and Purchase price seem to be similar fields so we will have one common field for these. The only possible issue could be different values of PurchasePrice and SalePrice for same transaction however we don't have any such data points in any of these schema. Discount will be merged. The other step is to remove repeat customer and notes because they are captured in discount type and finance in customer. The pro of this activity is that we will remove redundancy however to fetch the data we will need more queries on different entities as well. We will also remove TradeIn from the Schema2 as we will merge it into the Type of merged entity. So Sale and Tradein in Schema2 will be translated to "Sale Tradein" type and Sale only will be translated to "Sale" type. The other issue we have here is that for Schema2 we

don’t have two VINs for one transaction if type is TradeIn we only have VIN for the vehicle that customer bought so we have irregularity in data although there is no loss of data in this case.
The Associate_Name will be present from Schema1 which may not be available for Schema2. MSRP will be taken from Schema1

Following is merged diagram for entity Transactions



(3) Now we will combine all these entities to perform the final integration schema which as follows



PROs of integrated schema

1) The integrated schema provides a unified view of all the data from both the Schemas.
2) We have created a new attribute Type for Entity transaction and new attribute for TradeIn and Buy Price creating these new attributes helps in minimizing missing values in the schema and it is also more understandable/readable from user's perspective. Otherwise to get information about the transaction type one would be required to query three attributes and then arrive at the conclusion. This optimizes both space as well as time.
3) By splitting the entity Transactions into Vehicle and Transactions in Schema 2 we can now achieve the mapping of VIN to transaction id. Previously Transaction ID was mapped to VIN which is still the case but the reverse mapping was not present and hence we wanted to get the Transaction details for a Vehicle we would have to search for each Transaction and then get a corresponding Transaction matching either Sell_VIn or Buy_VIN. Now we can just fetch VIN using Vehicle Entity this would be fast because it's the primary attribute we can then fetch TransactionID for that record and then search the transaction in Transactions which will again be fast because TransactionID is primary attribute for Transactions.

4) We have also modified the relationship between customer and Transactions in Schema 2 . The customer essentially have stronger relationship with the Vehicle because Customer usually is not associated with the sale details and these details are more relevant to sales department. Therefore we have an indirect relationship with the Transaction for customer where Vehicle is the intermediate entity.

Cons:

1) The queries that we could run on Schema2 to fetch transaction details associated with a customer will be less efficient now as we don’t have a direct relationship with the Customer and Transactions entities. We will first need to fetch the Vehicle details and use that to fetch the data. It is essentially a Trade off
2) For Vehicles data from Schema2 we only have one VIN for any TradeIn Transaction whereas in our schema1 we had two VINs(buy and sell) for TradeIn Transactions. This will create some TradeIn values to have only one vehicle associated with it and while some will have two VINs associated to them in case of TradeIN type. Although it doesn’t create any data inconsistencies as type attribute resolves that issue but we still have some sort of unstructured data.
3) We can have missing values for customer data in case of certain customer ids. Also merging customerids is a tedious task because we may have to redefine them because two schemas can have same customer ids for two different customers.

Goals of data curation:

- Collection : The data has been collected from two different Schemas and each Schema having different source of data
- Organization : We are using ER diagram/relation model to organize the data
- Storage: This schema be stored either as a meta schema or we can create a single dataset for this schema. The attributes can be validated as well
- Preservation The attributes and elements will help in understanding the data. We have a relational schema and that will help our application in identifying data present inside relations using entity and attirbutes
- Discoverability: The data can be searched using the elements. There are various frameworks DBMS softwares with which we can access our data using keys in our application
- Access   The access can be taken care by the DBMS which implements this shchema and stores the data in records

- Workflow   Support the ability to systematize data workflows. This will also be taken care by publishing software. A utility can be written on the top of the apis to automate various workflows and that can be documented. Eg we can document how data will be populated from text document to relational schema. This is part of Standard ETL.
- Identification We have various constraints and attributes and keys which will help in authenticating and validating the data. We have primary keys and foreign keys for the entities
- Integration The data has been integrated from two different schemas and the source of data for these schemas have been text files, rtf files and excel files
- Security: This is also managed by DBMS where we will be implementing our schema but we need to ensure that the permissions are managed accordingly.
- Compliance This will be ensured by the legal team of the organization. Any data that we need to populate in the data should be first approved by the legal team.