Laboratory 3

Title of the Laboratory Exercise: data model to relational model

1. Introduction and Purpose of Experiment

The ER schema is to be converted into a relational schema as data cannot be stored in an ER schema. A relation schema gives the basic information describing a table or relation. It is the logical definition of a table. This includes a set of column names, and the data types associated with each column. By doing this lab, students will be able to map ER schema to relational schema.

2. Aim and Objectives

Aim

- To map data model to relational model

Objectives

At the end of this lab, the student will be able to

- Map ER schema to relational schema
- Insert tuples using SQL commands for the developed database schema

3. Experimental Procedure

i. Map all the components in the ER diagram to corresponding relation entities and instances
ii. Insert tuples using SQL commands
iii. Design SQL commands using aggregate functions in SQL
iv. Execute SQL commands
v. Test the executed commands
vi. Document the Results
vii. Analyse and discuss the outcomes of your experiment

4. Questions

a. Consider the ER diagram you have drawn in Laboratory 2. Convert the ER diagram to corresponding relational database schema.

b. Insert the tuples (minimum five) for the developed database schema using SQL commands. Perform aggregate functions in SQL based on the developed database schema.

5. Calculations/Computations/Algorithms

Step 1: Mapping of Regular Entity Types

Here the relations Client Manager, Clients, Accounts,Bank,Branch,Transaction and ATM in the relational schema corresponding to the regular entities in the ER diagram are represented

After Step 1

Client Manager

Manager_id	Client Manger Name

Clients

Client_id	Client_name	Client_Address	

Bank

B_Address	B_Name	Bank_ID

Accounts

Acc_ID	Acc_date	Acc_type

Transaction(Witdraw,Transfer,Deposit)

Transaction_id	Amount

Branch

Branch_Address	Branch_Name	Branch_ID

ATM

ATM_id	Location

Step 2: Mapping of Weak Entity Types

There are no Weak Entity Types in the ER Diagram

Step 3: Mapping of Binary 1:1 Relation Types

we map the 1:1 relationship type from the Figure by choosing the participating entity type Bank to serve in the role of *S* because its participation in the operates relationship type is total and Included the primary key of the Client Manager relation as foreign key in the Bank relation and renamed it Bank_Mang_ID.

Step 4: Mapping of Binary 1:N Relation Types

- For Operates we include the primary key Bank_ID and Bank_Name of the Bank relation as foreign key in the ATM relation and call it ATM_Bank_ID andATM_ Bank_Name.
- **For Holds we include the primary key Account_ID of the Accounts relation as foreign key in the Client relation and call it Account_Client_ID.**
- For dealswith we include the primary key Client _manger_Id of the Client Manger relation as foreign key in the Clients relation and call it Client_Mgr_ID.
- For Transacts we include the primary key Account_ID of the Account relation as foreign key in the Transaction relation and call it Transaction_AC_ID.
- For Held at we include the primary key Branch_ID of the Branch relation as foreign key in the Account relation and call it Account_Branch_ID.
- For belong to we include the primary key Bank_ID and Bank_Name of the Bank relation as foreign key in the Branch relation and call it Branch_Bank_ID and Branch_Bank_Name.

Step 5: Mapping of Binary M:N Relationship Types

- One Customer can have more than one Accounts and also one Account can be Managed by one or more clients(Joint Account). The primary keys of the Clients and Accounts relations are included as foreign keys in holds and renamed hClient_ID and hAccount_ID respectively .

Holds

hClient_ID	hACcount_ID

Step 6: Mapping of Multivalued attributes

- The attribute SClient_Phonenumber represents the multivalued attribute Client_Phonenumber of clients, while SClient_ID - as foreign key - represents the primary key of the Client relation

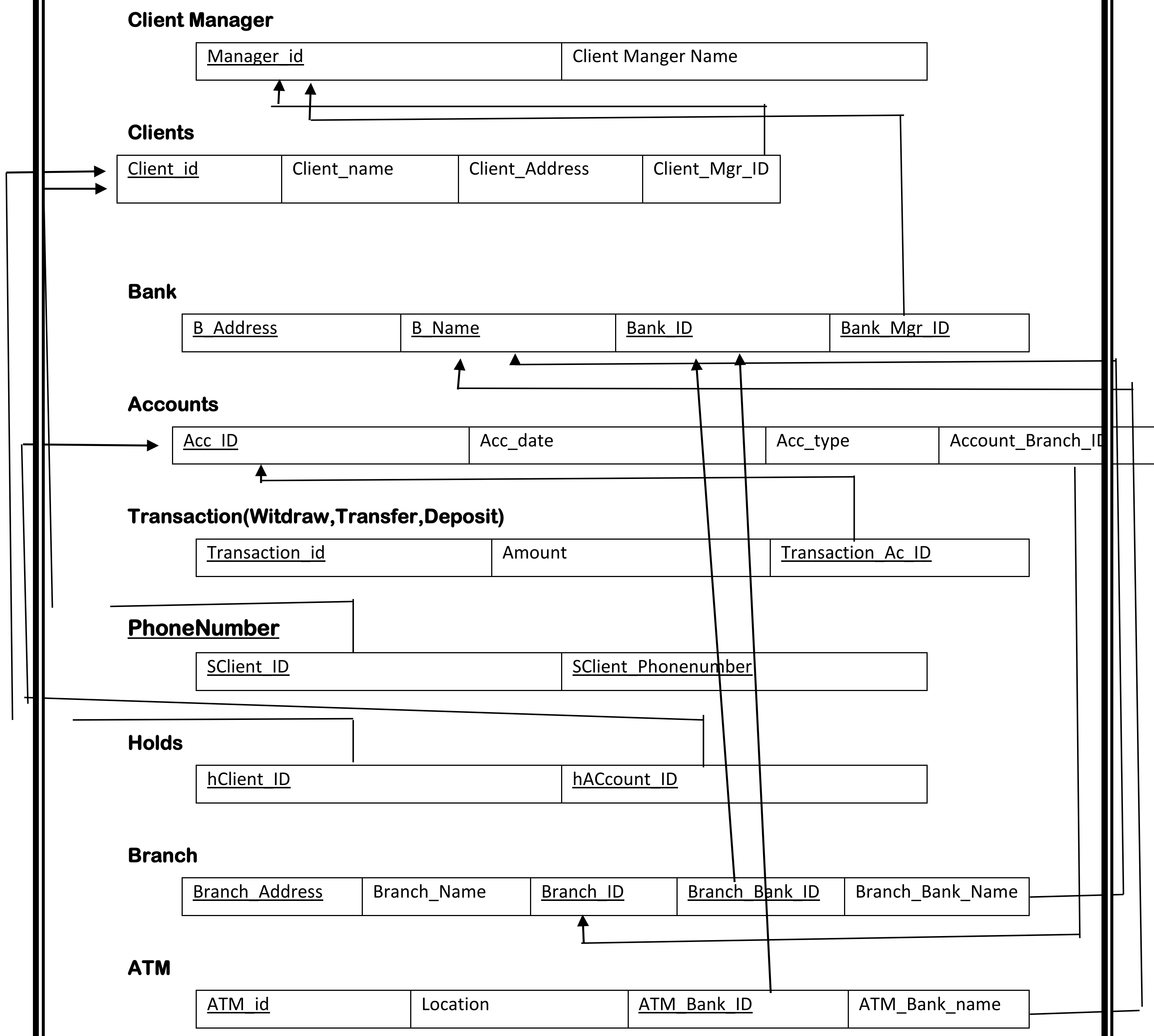
PhoneNumber

SClient_ID	SClient_Phonenumber

Step 7: Mapping of N-ary Relationship Types

There are no N-ary relationships in the ER Diagram

6. Presentation of Results



Client Manager

Manager_id	Client Manger Name

Clients

Client_id	Client_name	Client_Address	Client_Mgr_ID

Bank

B_Address	B_Name	Bank_ID	Bank_Mgr_ID

Accounts

Acc_ID	Acc_date	Acc_type	Account_Branch_ID

Transaction(Witdraw,Transfer,Deposit)

Transaction_id	Amount	Transaction_Ac_ID

PhoneNumber

SClient_ID	SClient_Phonenumber

Holds

hClient_ID	hACcount_ID

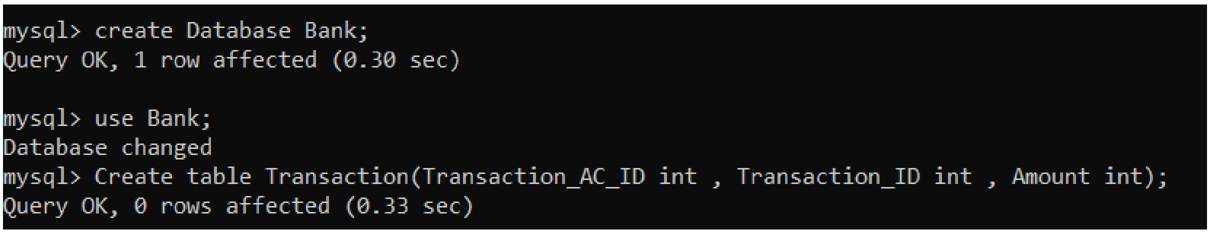
Branch

Branch_Address	Branch_Name	Branch_ID	Branch_Bank_ID	Branch_Bank_Name

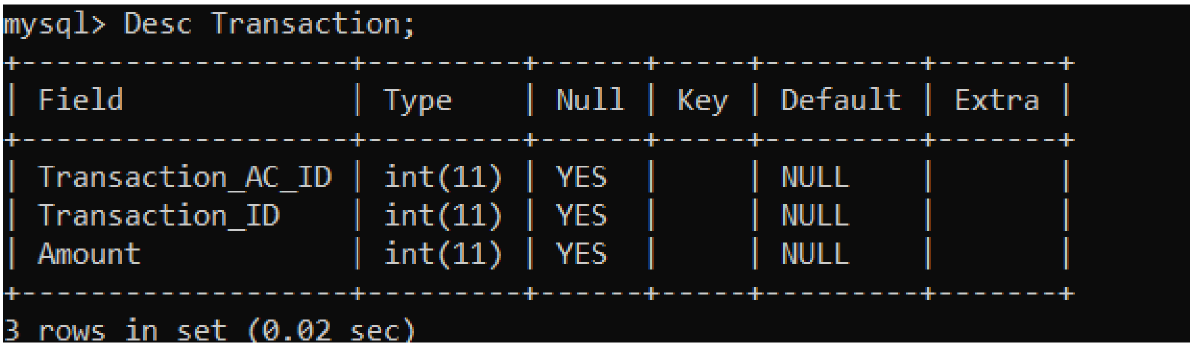
ATM

ATM_id	Location	ATM_Bank_ID	ATM_Bank_name

b) Aggregate functions in SQL based on the developed database schema.



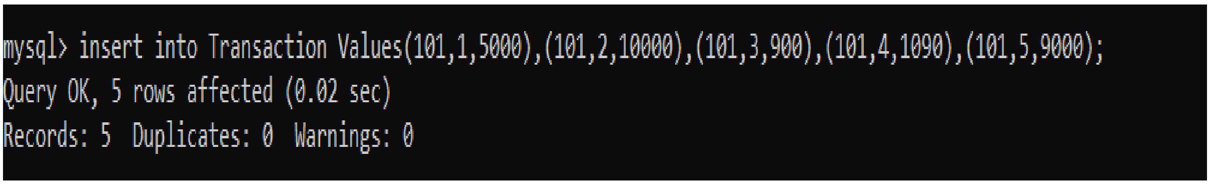
```
mysql> create Database Bank;
Query OK, 1 row affected (0.30 sec)

mysql> use Bank;
Database changed
mysql> Create table Transaction(Transaction_AC_ID int , Transaction_ID int , Amount int);
Query OK, 0 rows affected (0.33 sec)
```



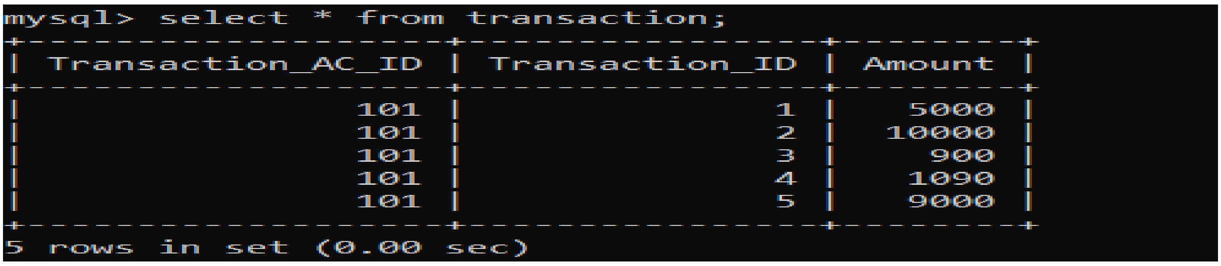
```
mysql> Desc Transaction;
+-------------------+---------+------+-----+---------+-------+
| Field             | Type    | Null | Key | Default | Extra |
+-------------------+---------+------+-----+---------+-------+
| Transaction_AC_ID | int(11) | YES  |     | NULL    |       |
| Transaction_ID    | int(11) | YES  |     | NULL    |       |
| Amount            | int(11) | YES  |     | NULL    |       |
+-------------------+---------+------+-----+---------+-------+
3 rows in set (0.02 sec)
```

Fig 3.1 Creation of Database Bank and Transaction Table

Here we have created transaction table which as attributes TransactionAC_ID ,Transaction_ID , Amount.

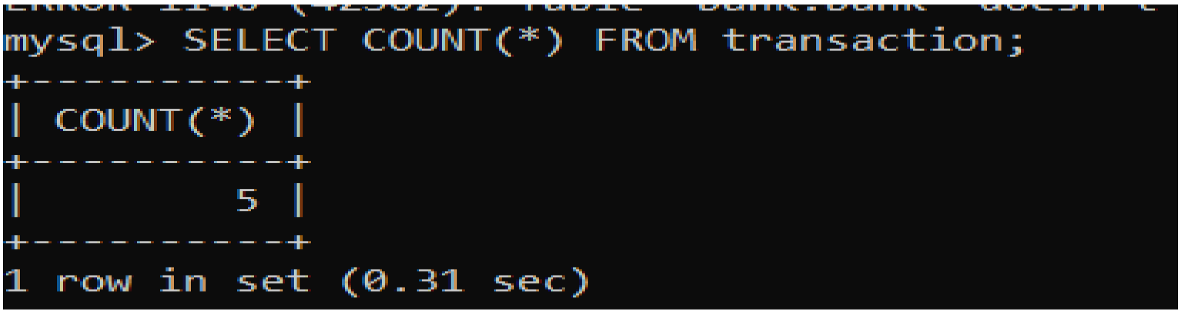


```
mysql> insert into Transaction Values(101,1,5000),(101,2,10000),(101,3,900),(101,4,1090),(101,5,9000);
Query OK, 5 rows affected (0.02 sec)
Records: 5  Duplicates: 0  Warnings: 0
```



```
mysql> select * from transaction;
+-------------------+----------------+--------+
| Transaction_AC_ID | Transaction_ID | Amount |
+-------------------+----------------+--------+
|               101 |              1 |   5000 |
|               101 |              2 |  10000 |
|               101 |              3 |    900 |
|               101 |              4 |   1090 |
|               101 |              5 |   9000 |
+-------------------+----------------+--------+
5 rows in set (0.00 sec)
```

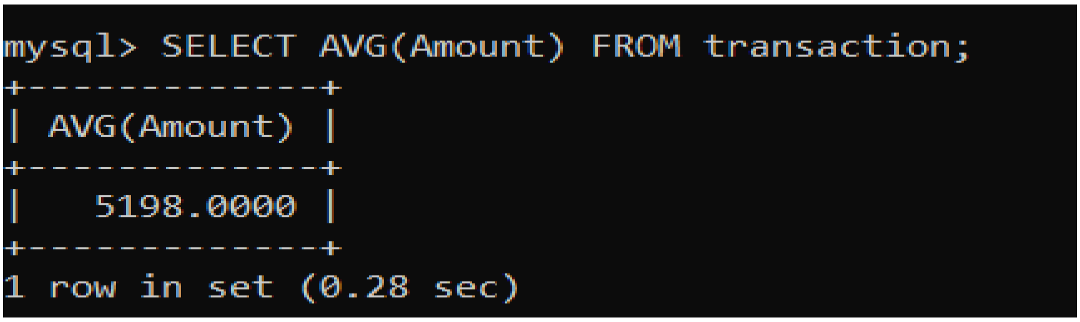
Fig 3.2 Inserting values into the transaction table



```
mysql> SELECT COUNT(*) FROM transaction;
+----------+
| COUNT(*) |
+----------+
|        5 |
+----------+
1 row in set (0.31 sec)
```

Fig 3.3 Use of COUNT(*) Function.

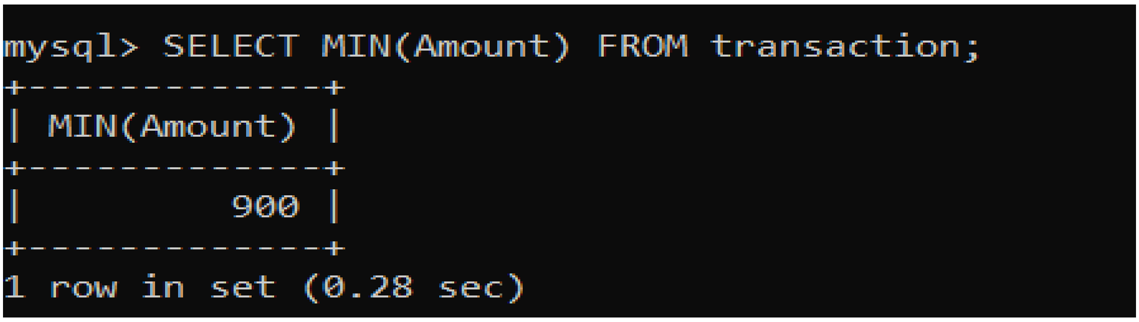
Here we have used COUNT(*) function Returns the number of rows in Transaction table



```
mysql> SELECT AVG(Amount) FROM transaction;
+-------------+
| AVG(Amount) |
+-------------+
|   5198.0000 |
+-------------+
1 row in set (0.28 sec)
```

Fig 3.4 Use of AVG() Function

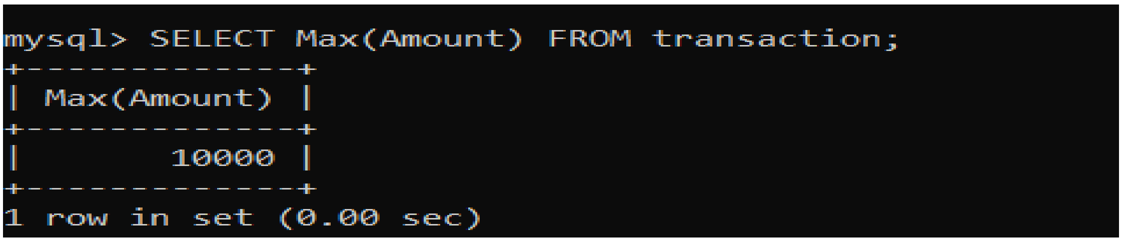
Here we have used Avg() function to calculate Average amount transacted



```
mysql> SELECT MIN(Amount) FROM transaction;
+-------------+
| MIN(Amount) |
+-------------+
|         900 |
+-------------+
1 row in set (0.28 sec)
```

Fig 3.5 Use of Min() Function

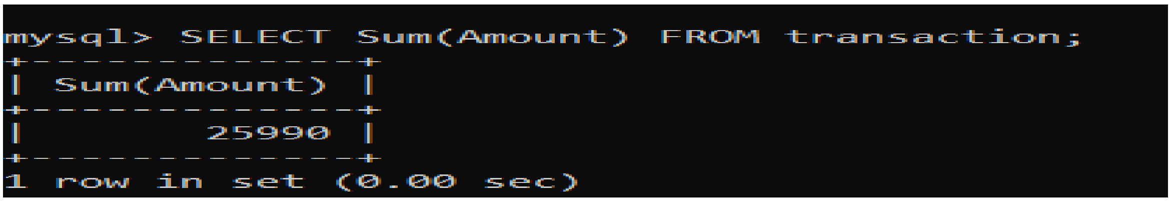
Here we have used Min() function to calculate Minimum amount transacted



```
mysql> SELECT Max(Amount) FROM transaction;
+-------------+
| Max(Amount) |
+-------------+
|       10000 |
+-------------+
1 row in set (0.00 sec)
```

Fig 3.6 Use of Max() Function

Here we have used Max() function to calculate Maximum amount transacted



```
mysql> SELECT Sum(Amount) FROM transaction;
+-------------+
| Sum(Amount) |
+-------------+
|       25990 |
+-------------+
1 row in set (0.00 sec)
```

Fig 3.7 Use of Sum() Function

Here we have used Sum() function to calculate Sum of amount transacted

7. Conclusions

The ER Model is intended as a description of real-world entities. Although it is constructed in such a way as to allow easy translation to the relational schema model, this is not an entirely trivial process. The ER diagram represents the conceptual level of database design meanwhile the relational schema is the logical level for the database design.

We can generate relational database schema using the ER diagram. Following are some key points to keep in mind while doing so:

1. Entity gets converted into Table, with all the attributes becoming fields(columns) in the table.
2. Relationship between entities is also converted into table with primary keys of the related entities also stored in it as foreign keys.
3. Primary Keys should be properly set.
4. For any relationship of Weak Entity, if primary key of any other entity is included in a table, foreign key constraint must be defined.

8. Comments

1. Limitations of Experiments

Not for unstructured data: Unless the data is cleanly delineated into different fields, rows or columns, ER diagrams are probably of limited use. The same is true of semi-structured data, because only some of the data will be useful.

Difficulty integrating with an existing database: Using ER Models to integrate with an existing database can be a challenge because of the different architectures.

2. Limitations of Results

In the results, we haven’t yet defined the primary keys and foreign keys, even though entity integrity and referential integrity is preserved, its not clear from the schema made from the ER diagram.

3. Learning happened

We learnt how to convert an ER diagram to a Relational Schema.

4. Recommendations

Beyond concerns over meeting the constraint requirements for primary keys, we must also assure adherence to the referential integrity constraints. We identify the referential integrity constraints by locating the corresponding attribute in each relation that is linked via a relationship. We then determine which of the relations contain the tuple that if the reference attribute were deleted or changed would jeopardize the integrity of the database.