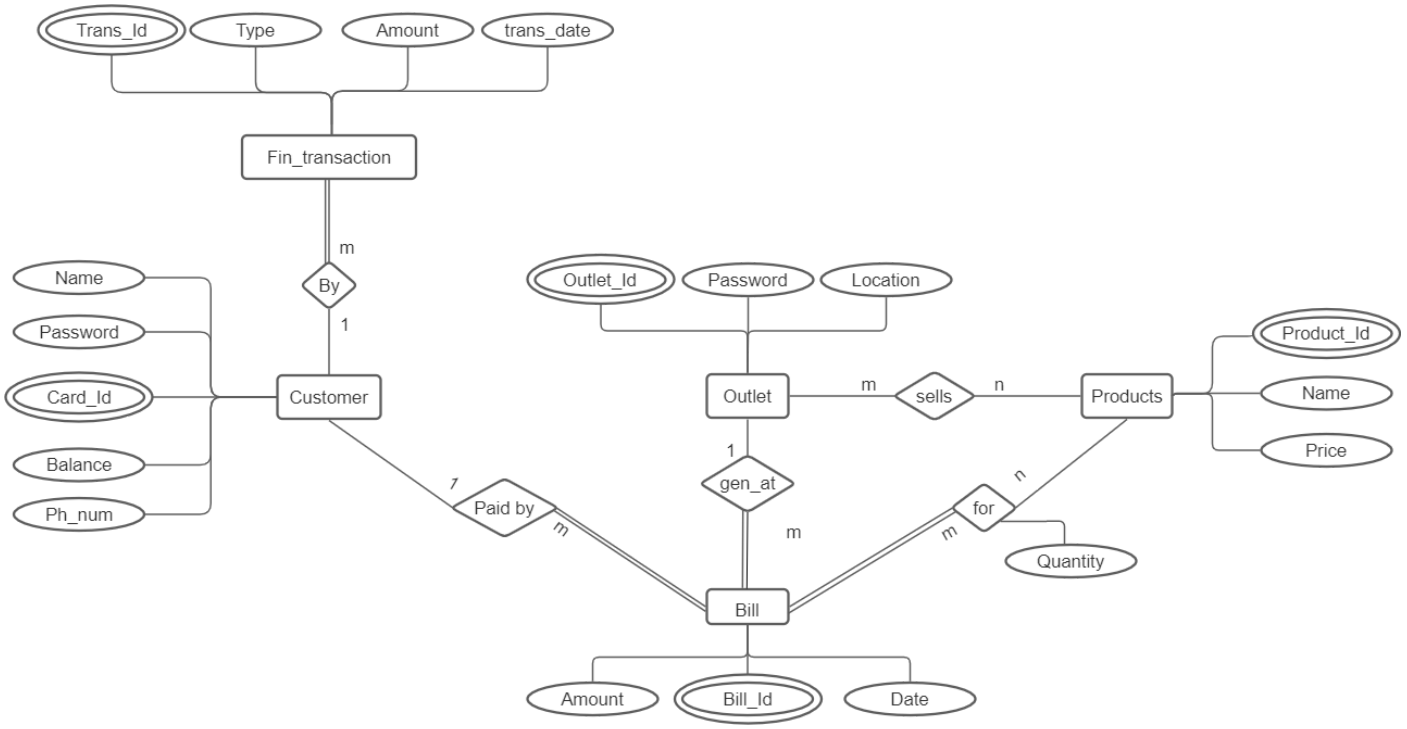
Starbucks Sales and Card Management System

K. Jayakar Reddy (18CS01004)
K. Kishorereddy (18CS01007)

Entities:

1. Customer
2. Outlet
3. Product
4. Bill (for purchases made)
5. fin_transaction

ER Diagram:



Conversion to Relational database tables:

Step 1: Mapping Regular Entity types

One table for each entity type E is created with simple attributes as columns.

Database so far:

- Bills(Bill_Id, Amount, Date);
- Products(Product_Id, Name, Price);
- Outlets(Outlet_Id, password, Location);
- Customers(Card_Id, Password, Name, Ph_num, Balance);
- Fin_transactions(Trans_Id, Type, Amount, trans_date);

Step 2: Mapping Weak Entities

Zero weak entities are present in the ER Model hence no changes to the Database.

Step 3: Mapping Binary 1:1 Relation types

No 1:1 relationship types are present in the ER Model hence no changes to the Database.

Step 4: Mapping 1:n Relation types

1. ‘gen_at’ relation between Bills and Outlets. Outlet_id (primary key of Outlets table) added as a foreign key to Bills table
2. 'by' relation between Customers and Fin_transactions table. Card_id (primary key of Customers table) added as a foreign key to Fin_transactions table.
3. 'paid_by' relation between Customers and Bills. Card_id(primary key of Customers) added as a foreign key to Bills table.

Database so far:

- Bills(Bill_Id, Card_Id, Outlet_Id Amount, Date);
- Products(Product_Id, Name, Price);
- Outlets(Outlet_Id, Password, Location);
- Customers(Card_Id, Password, Name, Ph_num, Balance);
- Fin_transactions(Trans_Id, Card_Id, Type, Amount, trans_date);

Step 5: Mapping m:n Relation types

1. ‘for’ relation between Bills and Products. Bill_Id (primary key of Bills table), Product_Id (primary key of Products table) are added as foreign keys to the newly created Purchases table. And the relational attribute Quantity is added as a column in this table.
2. ‘sells' relation between Outlets and Products. Outlet_Id (primary key of Outlets able), Product_Id (primary key of Products table) are added as foreign keys to the newly created Outlet_menus table.

Database so far:

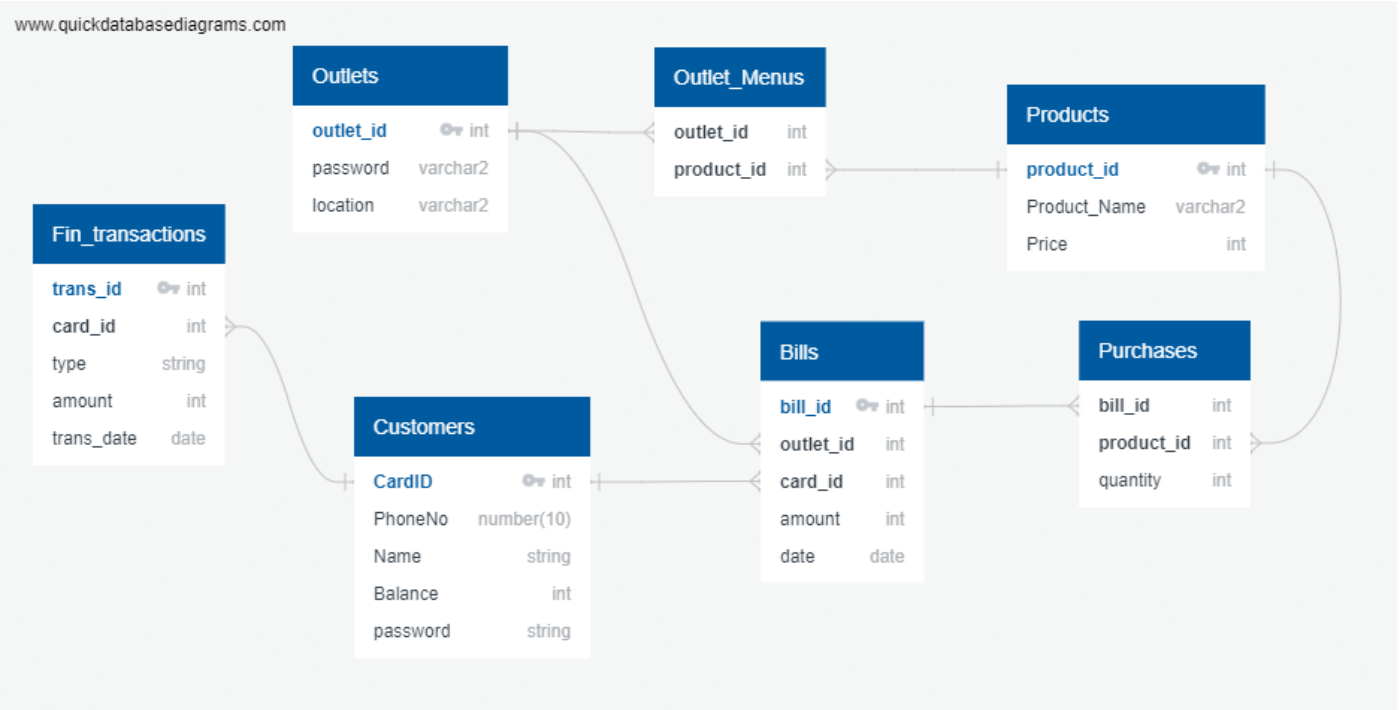
- Bills(Bill_Id, Card_Id, Outlet_Id Amount, Date);
- Products(Product_Id, Name, Price);
- Outlets(Outlet_Id, password, Location);
- Customers(Card_Id, Password, Name, Ph_num, Balance);
- Fin_transactions(Trans_Id, Card_Id, Type, Amount, trans_date);
- Purchases(Bill_Id, Product_Id, Quantity);
- Outlet_menus(Outlet_Id, Product_Id);

Steps 6,7,8 are not applicable to the E.R diagram.
Hence the final Relational Database is as follows.

Relational Database Tables:

- Bills(Bill_Id, Card_Id, Outlet_Id Amount, Date);
- Products(Product_Id, Name, Price);
- Outlets(Outlet_Id, password, Location);
- Customers(Card_Id, Password, Name, Ph_num, Balance);
- Fin_transactions(Trans_Id, Card_Id, Type, Amount, trans_date);
- Purchases(Bill_Id, Product_Id, Quantity);
- Outlet_menus(Outlet_Id, Product_Id);

From the above relational tables, it is easily observable that the database is in BCNF.
Primary Key constraints and Foreign key constraints are mentioned in the following graphic.



Functionalities:

- An Admin / HQ can add and alter outlets, credentials, and products
- Admin / HQ can see the sales stats
- Outlets can manage product menu
- An outlet can do cash transactions
- Outlets can issue customer cards with phone_num as username and temp password
- An outlet can do transactions based on cards
- An outlet can see its own sales stats
- Customers can change their password
- Customers can do credit or debit transactions on their cards
- Customers will be rewarded based on orders details will be in cred_debt trans as type = reward.
- Customers can see their own card transactions history

Assumptions:

- Phone number shall be verified on issuing a card
- During a purchase, card_num will be entered directly to imply card swiping
- Phone number is also entered directly but practically number will be verified via OTP before using the card linked with that number for the transaction.