1. **Requirements**

* A Customer has cId(unique), Name and phoneNo.
* Buyer is a Customer with attribute promoNo.
* Seller is a Customer with attribute types.
* Buyer can buy one or more the products with proId(Unique), Address,brandName.
* Seller can add one or more Items in the Products to sell.
* Products have their Category.
* The buyer can put the bought products in the Cart Items.
* Buyer make the payment for the bought items in the cart.
* The attributes of Payment pId(Unique), paymentType, date, amount.
* Buyer get the receipt while paying and receiptNo(unique).

1. **EER Diagram**

A picture containing text, map

Description automatically generated

1. **EER model to Relational Model**

Buyer(cId,Name,phoneNo,promoNo)

Products(proId,brandName,Address)

Buy(cId,proId,buying\_date)

Foreign key (cId) references Buyer(cId)

Foreign key (proId) references Product (proId)

Seller(cId,Name,phoneNo,types)

Adds(cId,proId,Date)

Foreign key (cId) references Seller(cId)

Foreign key (proId) references Products (proId)

Payment(pId,pDate,pType,amount,cId)

Foreign Key(cId) references Buyer(cId)

Category(name,proId)

Foreign Key(proId) references Products(proId)

CartItems(itemNumber,proId)

Foreign Key(proId) references Products(proId)

Recipt(reciptNo,pid)

Foreign Key(pId) references Products(proId)

**Answer for 3C**

Option 2 is used for mapping ISA hierarchy because of the followings reasons

* A customer who is not a buyer or seller(When ISA relationship is partial). But here (sub classes buyer and seller) cover the super class customer.
* A customer who is a buyer but not a seller(i.e. Disjoint)
* A customer who is both buyer and seller(overlapping classes)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Overlapping | Disjoint | Total | Partial |
| Option 2 |  |  |  | X |

1. **Database Table**

create table Buyer(

cId varchar(30) not null primary key,

phoneNo int CHECK ([phoneNo] LIKE REPLICATE('[0-9]', 10)),

promoN0 int,

)

create table Products(

proId varchar(30) not null primary key,

brandName varchar(30) not null,

Address varchar(100) not null

)

create table Buy(

cId varchar(30) not null foreign key references Buyer,

proId varchar(30) not null foreign key references Products,

buying\_date date DEFAULT GETDATE(),

primary key(cId,proId)

)

create table seller(

cId varchar (30) not null primary key,

name varchar(30),

phoneNo int CHECK ([phoneNo] LIKE REPLICATE('[0-9]', 10)),

types varchar(30),

)

create table Adds(

cId varchar(30) not null foreign key references Seller,

proId varchar(30) not null foreign key references Products,

Date Date default GETDATE(),

primary key(cID, proId)

)

create table Payment(

pId varchar(30) not null primary key,

amount int not null check (amount>0),

pDate date default GETDATE(),

pType varchar(30) check (pType='card' or pType = 'Cash' or pType= 'Paytm'),

cId varchar(30) not null foreign key references Buyer,

)

Create table Category(

name varchar(30) not null,

proId varchar(30) not null foreign key references Products

primary key(name)

)

create table CartItems(

itemNumber int not null,

proId varchar(30) not null foreign key references Products,

primary key(itemNumber)

)

create table Recipt(

receiptNo int not null,

pId varchar(30) not null foreign key references Payment,

primary key(receiptNo)

)