Deliverable 2

EER Schema to Relational Mapping

Team Number: 11
Section: CS-631 009

Team Members:
Shivrishvith Palthyavathu (SP2962)
Vinay Ram Gazula (VG472)
Ganesh sai Ravuru (GR359)

Goals of this Phase:

The main goal of this phase is to transform the conceptual schema or the extended Entity-Relation (EER) diagram into a Relational model. The goal of Logical Design is to create structured tables containing all the attributes that are required to reflect the company's business environment. These tables will be able to store all the data in a non-redundant manner and contain keys and foreign keys so that all relations among the entities are supported without losing any data. We also define any other constraints taken into consideration to ensure data integrity.

Revisions to the Previous Phase:

When compared to the previous version of our EER diagram there aren't any substantial changes to the current version. But there are some small changes like:

- All foreign keys are removed from the ER diagram.
- The entity "SHIPPING_ADDRESS" is changed into a Weak Entity Type.
- The relation "HAS" is changed into an Identifying Relation Type.
- In the entity "SHIPPING_ADDRESS" the multivalued composite attribute "Address" is replaced with an atomic attribute "Name" which is a weak primary key.
- In the entity "SHOPPING_BASKET" the multivalued composite attribute "Products" is removed.
- The Cardinality ratios of relations "STORES", "HAS", "IN" and "APPLIES" are changed.

Along with these there are some other small changes made to the EER diagram. The updated EER diagram which is used to generate the Relational model is included at the end of this document.

EER Diagram to Relational Model:

Using the EER to Relational Algorithm the following Relational Model is developed:

CUSTOMER

CustomerID (PK)	FName	SName	HAddress	PhoneNum	EmailID	MemStatus

Where, HAddress - Home Address; MemStatus - Membership Status.

SHIPPING_ADDRESS

CustomerID	SAName (Composite PK)	StNum	StName	City	State	Country	ZipCode

Where, SAName - Shipping Address Name; StNum - Street Number; StName - Street Name.

CREDIT_CARD

CardNum (PK)	CardType	SecurityNum	ExpDate	OwnerName	BAddress

Where, BAddress - Billing Address.

STORED_CARDS

CardNum (PK)	CustomerID (FK)	CName

Where, CName - Card Name.

SHOPPING_BASKET

BasketID (PK)	CustomerID (FK)	TotalAmount

TRANSACTION

TransactionID	BasketID (Composite PK)	CustomerID (FK)	SAName (FK)	CardNum (FK)	TStatus

Where, SAName - Shipping Address Name; TStatus - Transaction Status.

PRODUCTS

ProductID (PK)	PName	PType	PQuantity	PPrice	Description

IN_BASKET

BasketID	ProductID (Composite PK)	Quantity

OFFERS

OfferCode (PK)	BasketID (FK)

HAVE_OFFERS

OfferCode	ProductID (Composite PK)	OPrice

COMPUTER

ProductID (PK)	CPUType

LAPTOP

ProductID (PK)	CPUType	Weight	RunningTime

PRINTER

ProductID (PK)	PrinterType	Resolution

The schematic of the Relational Model is attached at the end of this document.

Constraints:

Referential Integrity Constraints:

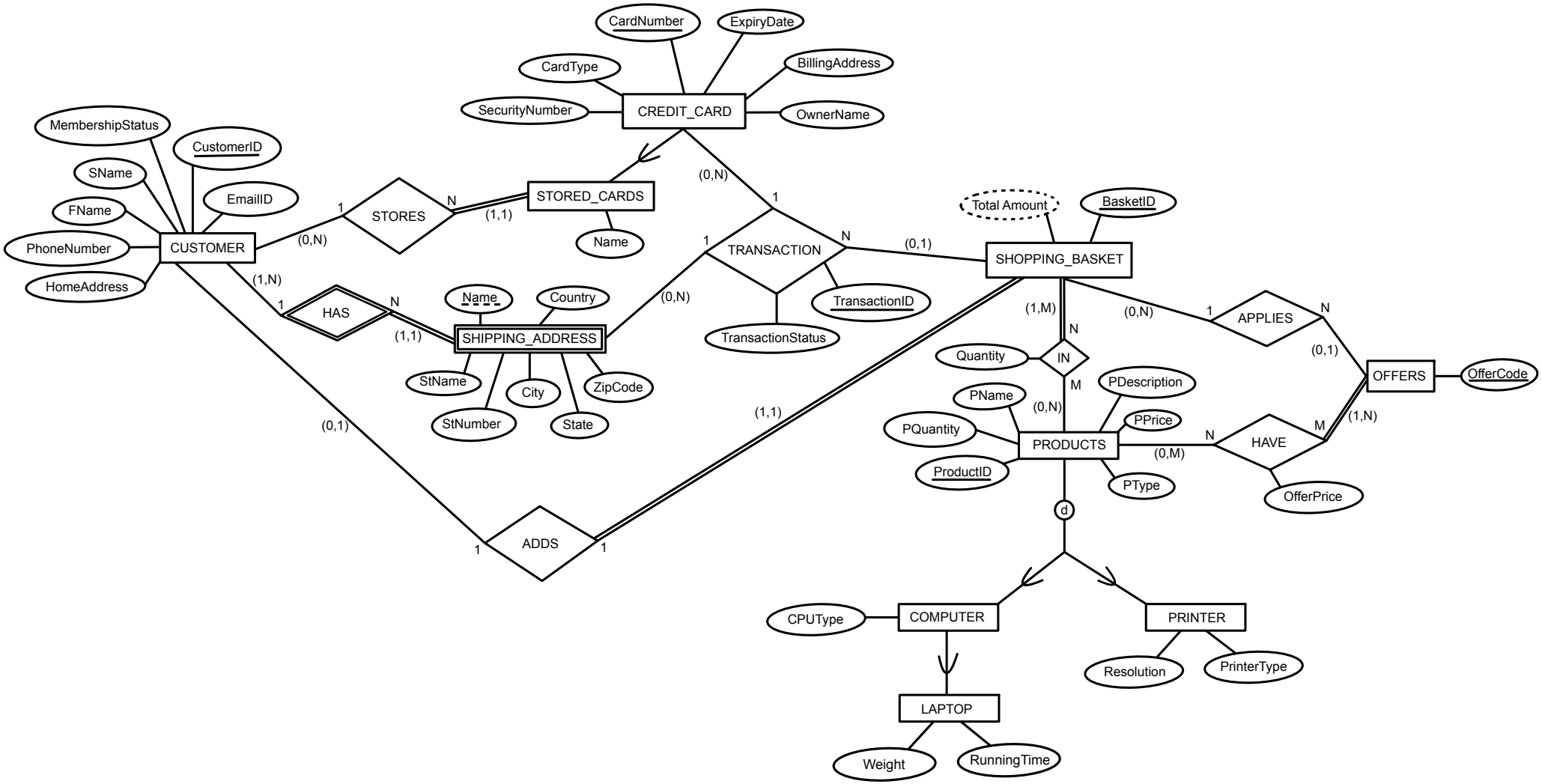
- `SHIPPING_ADDRESS.CustomerID` references `Customer.CustomerID`
- `STORED_CARDS.CustomerID` references `Customer.CustomerID`
- `SHOPPING_BASKET.CustomerID` references `Customer.CustomerID`
- `TRANSACTION.CustomerID,SAName`references `SHIPPING_ADDRESS.CustomerID,SAName`
- `STORED_CARDS.CardNum` references `CREDIT_CARD.CardNum`
- `TRANSACTION.CardNum` references `CREDIT_CARD.CardNum`
- `IN_BASKET.BasketID` references `SHOPPING_BASKET.BasketID`
- `OFFERS.BasketID` references `SHOPPING_BASKET.BasketID`
- `HAVE_OFFERS.OfferCode` references `OFFERS.OfferCode`
- `IN_BASKET.ProductID` references `PRODUCTS.ProductID`
- `HAVE_OFFERS.ProductID` references `PRODUCTS.ProductID`
- `COMPUTER.ProductID` references `PRODUCTS.ProductID`
- `PRINTER.ProductID` references `PRODUCTS.ProductID`
- `LAPTOP.ProductID` references `COMPUTER.ProductID`

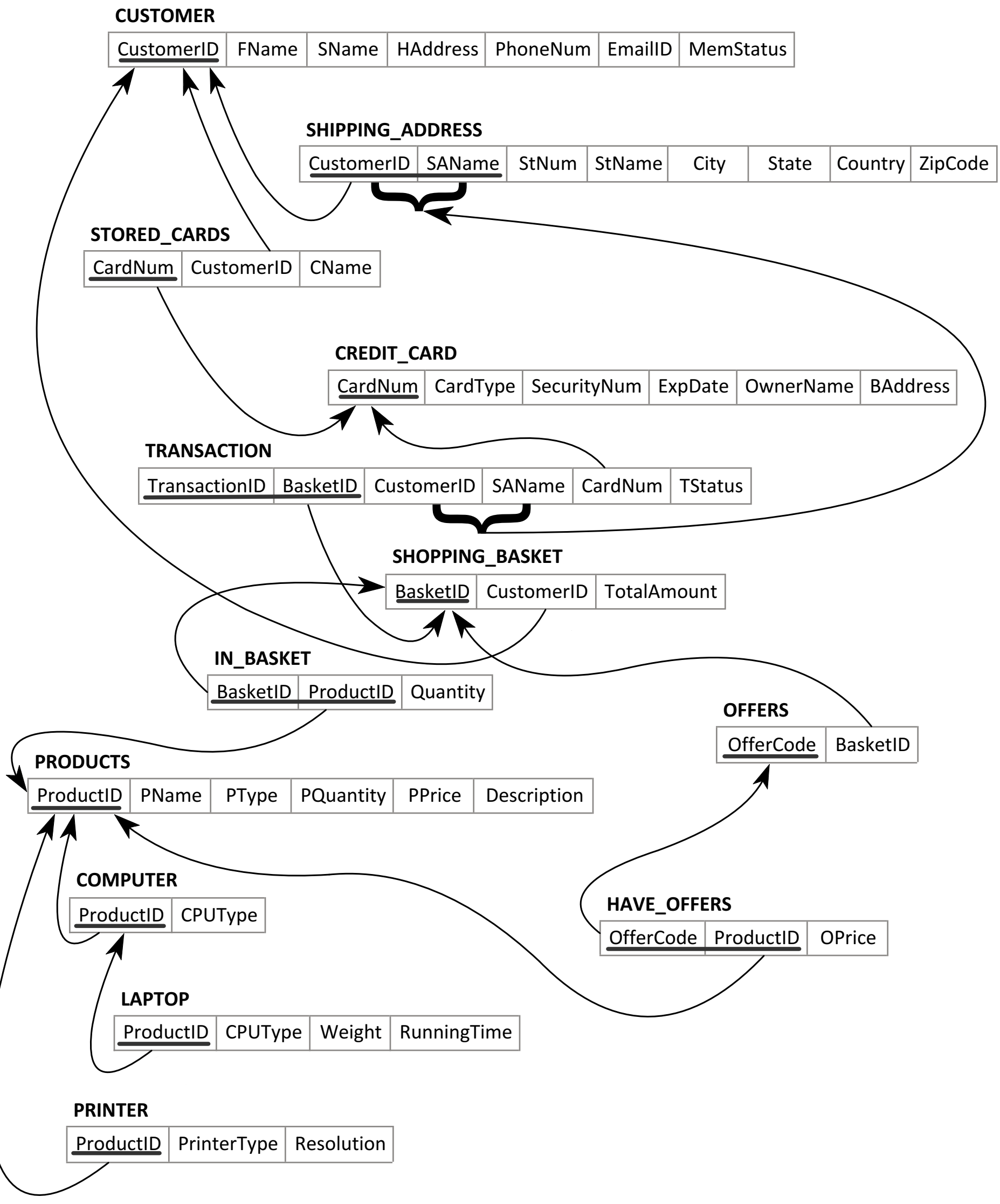
Other Constraints:

- Only gold and platinum customers can benefit from offers.
- A shopping basket is always non-empty.

Difficulties faced in this Phase:

- Deciding on the primary and foreign keys for each table, especially for those with multiple relationships.
- Ensuring that the schema is normalized to avoid data redundancy.
- Mapping specialized attributes for product types.


CardNumber
ExpiryDate
CardType
BillingAddress
SecurityNumber
CREDIT_CARD
OwnerName
MembershipStatus
CustomerID
SName
EmailID
FName
PhoneNumber
CUSTOMER
HomeAddress
1
STORES
N
(0,N)
(1,1)
STORED_CARDS
Name
(0,N)
1
TRANSACTION
N
(0,1)
Total Amount
BasketID
SHOPPING_BASKET
TransactionID
TransactionStatus
(1,N)
1
HAS
N
(1,1)
Name
Country
SHIPPING_ADDRESS
StName
City
ZipCode
StNumber
State
(0,1)
(1,1)
1
ADDS
1
(1,M)
N
Quantity
IN
M
(0,N)
1
APPLIES
N
(0,1)
OFFERS
OfferCode
PName
PDescription
PQuantity
PPrice
PRODUCTS
ProductID
PType
N
HAVE
M
(0,M)
(1,N)
OfferPrice
d
CPUType
COMPUTER
PRINTER
Resolution
PrinterType
LAPTOP
Weight
RunningTime


CUSTOMER
CustomerID
FName
SName
HAddress
PhoneNum
EmailID
MemStatus
SHIPPING_ADDRESS
CustomerID
SAName
StNum
StName
City
State
Country
ZipCode
STORED_CARDS
CardNum
CustomerID
CName
CREDIT_CARD
CardNum
CardType
SecurityNum
ExpDate
OwnerName
BAddress
TRANSACTION
TransactionID
BasketID
CustomerID
SAName
CardNum
TStatus
SHOPPING_BASKET
BasketID
CustomerID
TotalAmount
IN_BASKET
BasketID
ProductID
Quantity
OFFERS
OfferCode
BasketID
PRODUCTS
ProductID
PName
PType
PQuantity
PPrice
Description
COMPUTER
ProductID
CPUType
HAVE_OFFERS
OfferCode
ProductID
OPrice
LAPTOP
ProductID
CPUType
Weight
RunningTime
PRINTER
ProductID
PrinterType
Resolution