Group 3 Lab3

Group members:

Lee Yew Chuan Michael, U2021372J

Lau Chen Yi Wynne, U2020016B

Chua Wen Rong Jonathan, U2021875L

Li Ziyang, U1821317D

Customer(customerID, Email, Username, fullName, Phone#, Address, Password)

Keys: customerID, Email, Username
Primary Key: customerID
FDs:
customerID → Email, Username, fullName, Phone#, Address, Password
Email → customerID, Username, fullName, Phone#, Address, Password
Username → customerID, Email, fullName, Phone#, Address, Password

Assumption:

- Every customer can only register with one Phone number and one address on the platform.
- We assume that different customer accounts may have the same phone number.
- We assume that different customer accounts may have the same address.

The relation is in 3NF.
Justification:
Since all the non trivial FDs of the schema contains a key on the left-hand side. The relation is in BCNF. Since it is in BCNF, the relation is also in 3NF.

Product_Type(typeID, parentTypeID, typeDescription)
Keys: typeID
Primary Key: typeID
FDs:
typeID → parentTypeID, typeDescription

The relation is in 3NF.
Justification:
Since all the non trivial FDs of the schema contains a key on the left-hand side. The relation is in BCNF. Since it is in BCNF, the relation is also in 3NF.

Product(productID, typeID, shopID, Colour, productName, Description, productPrice, Size)
Keys: productID
Primary Key: productID
FDs:
productID → typeID, shopID, Colour, productName, Description, productPrice, Size

Assumption:
Every product can only be sold by one shop, but a shop can sell multiple products.

This relation is in 3NF.
Justification:
Since all the non trivial FDs of the schema contains a key on the left-hand side. The relation is in BCNF. Since it is in BCNF, the relation is also in 3NF.

Order_Item(Sequence#, orderID, shipmentID, productID, unitPrice, orderItemStatus, Qty)
Keys: (Sequence#, orderID)
Primary Key: (Sequence#, orderID)
FDs:
(Sequence#, orderID) → shipmentID, productID, unitPrice, orderItemStatus, Qty

Assumption:
- We assume the same product type can have different unit prices in different orders, as the customer may have gotten a discount or incentive in a particular order.
- Sequence# is only unique within an order, but not unique across different orders.

The relation is in 3NF.
Justification:
Since all the non trivial FDs of the schema contains a key on the left-hand side. The relation is in BCNF. Since it is in BCNF, the relation is also in 3NF.

Order(orderID, paymentStatus, orderDate, orderStatus, customerID)
Keys: orderID
Primary Key: orderID
FDs:
orderID → paymentStatus, orderDate, orderStatus, customerID

The relation is in 3NF.
Justification:
Since all the non trivial FDs of the schema contains a key on the left-hand side. The relation is in BCNF. Since it is in BCNF, the relation is also in 3NF.

Invoice(Invoice#, orderID, invoiceStatus, invoiceDate)
Keys: Invoice#, orderID
Primary Key:Invoice#
FDs:
Invoice# → invoiceStatus, invoiceDate, orderID
orderID → invoiceStatus, invoiceDate, Invoice#

The relation is in 3NF.
Justification:
Since all the non trivial FDs of the schema contains a key on the left-hand side. The relation is in BCNF. Since it is in BCNF, the relation is also in 3NF.

CreditCard(Card#, customerID, expiryDate)
Keys:Card#
Primary Key:Card#
FDs:
Card# → customerID, expiryDate

Assumption:
Assume each Customer can register multiple Credit Cards. But, each Credit Card can only be used by one Customer. (ie. A credit card cannot be shared among multiple customers)

The relation is in 3NF.
Justification:
Since all the non trivial FDs of the schema contains a key on the left-hand side. The relation is in BCNF. Since it is in BCNF, the relation is also in 3NF.

Shipment(shipmentID, Tracking#, shipmentDate)
Keys: shipmentID, Tracking#
Primary Key: shipmentID
FDs:
shipmentID → Tracking#, shipmentDate
Tracking# → shipmentID, shipmentDate

Assumption:
We assume that Tracking# is unique.

The relation is in 3NF.
Justification:
Since all the non trivial FDs of the schema contains a key on the left-hand side. The relation is in BCNF. Since it is in BCNF, the relation is also in 3NF.

Photo(photoID, productID)
Keys: photoID
Primary Key: photoID
FDs:
photoID → productID

The relation is in 3NF.
Justification:
Since all the non trivial FDs of the schema contains a key on the left-hand side. The relation is in BCNF. Since it is in BCNF, the relation is also in 3NF.

Shops(shopID, shopName)
Keys: shopID
Primary Key: shopID
FDs:
shopID → shopName

The relation is in 3NF.
Justification:
Since all the non trivial FDs of the schema contains a key on the left-hand side. The relation is in BCNF. Since it is in BCNF, the relation is also in 3NF.

Can_Sell(shopID, typeID)
Keys: (shopID, typeID)
Primary Key: (shopID, typeID)
FDs:
(shopID, typeID) → (shopID, typeID)

The relation is in 3NF.
Justification:
Since there are no non trivial FDs in the schema. The relation is in BCNF. Since it is in BCNF, the relation is also in 3NF.

ER approach taken for subclass

Payment(paymentID, paymentDate, amount)
Keys: paymentID
Primary Key: paymentID
FDs:
paymentID → paymentDate, amount

The relation is in 3NF.
Justification:
Since all the non trivial FDs of the schema contains a key on the left-hand side. The relation is in BCNF. Since it is in BCNF, the relation is also in 3NF.

Full_Payment(paymentID, Invoice#)
Keys: paymentID, Invoice#
Primary Key: paymentID
FDs:
paymentID → Invoice#
Invoice# → paymentID

The relation is in 3NF.
Justification:
Since all the non trivial FDs of the schema contains a key on the left-hand side. The relation is in BCNF. Since it is in BCNF, the relation is also in 3NF.

Partial_Payment(paymentID, Invoice#)
Keys: paymentID, Invoice#
Primary Key: paymentID
FDs:
paymentID → Invoice#
Invoice# → paymentID

The relation is in 3NF.
Justification:
Since all the non trivial FDs of the schema contains a key on the left-hand side. The relation is in BCNF. Since it is in BCNF, the relation is also in 3NF.