KATHMANDU UNIVERSITY

DHULIKHEL, NEPAL

Department of Computer Science & Engineering (DoCSE)

Lab Assignment

COMP-232

Submitted By:

Sambeg Shrestha (45)

Computer Engineering

2nd Year, 2nd Semester

Submitted to:

Mr. Santosh Khanal

24 September, 2020

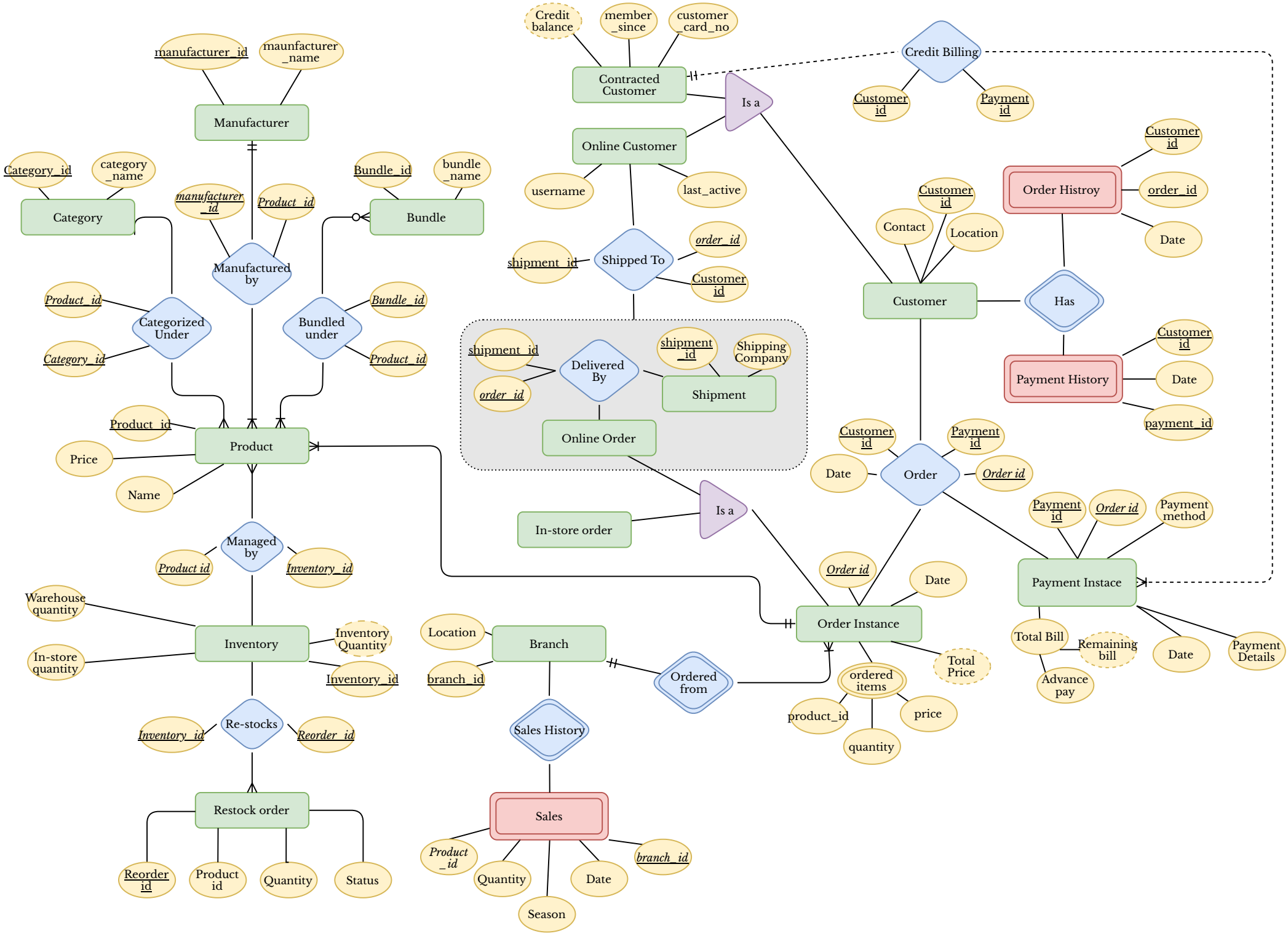
Case study

Study the following scenario and answer the following question below.

Application Description: *The application is an electronics vendor that operates both a Web site and a chain of many physical stores. Examples include Best Buy and Circuit City. To find out more about this application, think about any experiences you may have had making purchases both online and in-store, and browse their Web sites. In our hypothetical company, it has been decided to redesign a major part of the database that underlies company operations. Unfortunately, the manager assigned to solicit database design proposals is not very computer literate and is unable to provide a very detailed specification at the technical level. Fortunately, you are able to do that. Here are a few points to consider:*

- *There are many different products, categorized into different groups such as notepad, laptop, phone and camera etc. Each Products are identified by its unique product number its description and price tag.*
- *Some customers have a contract with the company and bill their purchases to an account number. They are billed monthly. Other customers pay with a credit or debit card or cash.*
- *Online sales must be sent to a shipper. The shipping company provides the tracking number for the product so it can respond to customer inquiries.*
- *Company staffs are responsible to maintain the inventory in store and to ship to online customers. When inventory is low, a reorder should be sent to the Supplier and listed in the database. When goods arrive, inventory should be updated and reorders marked as having been filled.*
- *Sales data are important for corporate planning. Marketers may want to look at sales data by time period, product, product grouping, season, region (for stores), etc.*

On the basis of the above scenario you are asked to design the complete ER Diagram. You have to state each entity with its all attributes and relationships. State the dependencies with its PK FK. Show the cardinality.


manufacturer_id
maunfacturer_name
Manufacturer
Category_id
category_name
Category
manufacturer_id
Product_id
Manufactured by
Bundle_id
bundle_name
Bundle
Product_id
Category_id
Categorized Under
Bundled under
Bundle_id
Product_id
Product_id
Price
Name
Product
Managed by
Product id
Inventory_id
Warehouse quantity
In-store quantity
Inventory
Inventory Quantity
Inventory_id
Inventory_id
Re-stocks
Reorder_id
Restock order
Reorder id
Product id
Quantity
Status
Credit balance
member_since
customer card_no
Contracted Customer
Is a
Online Customer
username
last_active
shipment_id
Shipped To
order_id
Customer id
shipment_id
Delivered By
order_id
shipment_id
Shipping Company
Shipment
Online Order
In-store order
Is a
Credit Billing
Customer id
Payment id
Customer id
Order Histroy
order_id
Date
Customer id
Contact
Location
Customer
Has
Customer id
Payment History
Date
payment_id
Customer id
Payment id
Date
Order
Order id
Payment id
Order id
Payment method
Payment Instace
Total Bill
Remaining bill
Date
Payment Details
Advance pay
Order id
Date
Order Instance
Total Price
ordered items
product_id
price
quantity
Location
Branch
branch_id
Ordered from
Sales History
Sales
Product_id
Quantity
Date
branch_id
Season

ENTITIES

1. **Manufacturer**: Stores information about different Manufacturers

- manufacturer_id: Primary key id
- manufacturer_name: Name of the Manufacturer

2. **Category**: Stores information about type of categorization

- category_id: Primary key id
- category_name: Name of Category

3. **Bundle**: Stores information about different groupings

- bundle_id: Primary key id
- bundle_name: Name of Bundle

4. **Product**: Stores information of different products

- product_id: Primary key id
- price: Price of item
- name: Name of item

5. **Inventory**: Stores information about quantity of products

- inventory_id: Primary key id

- Warehouse quantity: Quantity of particular item in the warehouse
- In-store quantity: Quantity of particular item in the store
- Inventory quantity: Sum of Warehouse and In-store quantity

6. **Restock order**: Stores information about products that needs to be restocked for ordering

- reorder_id: Primary key id
- product_id: Id of particular products
- quantity: Quantity of the product
- status: Status whether Restocked or pending

7. **Customer**: Stores information about customer

- customer_id: Primary key id
- Contact: Contact Number of the customer
- Location: Address of the Customer

8. **Online Customer**: Specialization of customer stores additional customer information who uses online services

- username: Account name for online user

last_active: Date and Time when user was last online

9. **Contracted Customer**: Specialization of customer stores additional customer information who uses in-store contract services

- member_since: Date when the customer entered into a contract
- customer_card_no: Unique membership card number
- Credit Balance: Pending Balance to be cleared

10. **Order History**: Stores information about orders made by a customer

- customer_id: Id of customer (FK)
- order_id: Id of order instance (FK)
- Date: Date and time when the order was placed

11. **Payment History**: Stores information about payments made by a customer

- customer_id: Id of customer (FK)
- payment_id: Id of payment instance (FK)
- Date: Date and time when the payment was made

12. **Order Instance**: Stores information about one particular ordering

- order_id: Primary key id
- Date: Date when order was placed

- Ordered items: Multi-valued attribute that stores product_id, Price, Quantity
- product_id: id of ordered product
- Price: Price of each item
- Quantity: Quantity of particular type of item
- Total Price: Derived attribute depending on Sum of product of quantity and prices of particular ordered items

13. **Online Order**: Specialization of Order Instance, it specializes order instance to online

14. I**n-store Order**: Specialization of Order Instance, it specializes order instance to in-store

15. **Payment Instance:** Stores information about the payment made by customer

- payment_id: Id of payment instance
- order_id: Id of order instance
- Payment method: The way payment occurred- Cash or Card
- Payment Details: Details regarding Payment
- Date: The Date and Time when payment occurred
- Total Bill: Total amount of required monetary payment
- Advance Pay: Total amount paid upfront

- Remaining Bill: Derived attribute, amount remainder from difference from Total Bill and Advance Pay

16. **Branch**: It stores information about store branch

- Location: Address of the Branch store
- branch_id: Primary Key id

17. **Sales**: It stores information about sales made by the branch

- product_id: Id of particular product
- quantity: Amount of particular product
- season: The Season when sale occurred
- date: The Date of the sale
- branch_id: Id of the branch

WEAK ENTITIES

1. Order History: Is dependent on customer and has discriminating composite primary key ids of customer and order instance
2. Payment History: Is dependent on customer and has discriminating composite primary key ids of customer and payment instance
3. Sales: Is dependent on Branch and has discriminating key id of branch

SPECIALIZATIONS

Customer

- Online Customer
- In-store customer

Order Instance

- -Online Order
- -instore Order

RELATIONSHIPS, Attributes, Cardinality and Dependency

1. **Manufactured by(Product, Manufacturer)**: Relationship between product and its manufacturer

- product_id
- manufacturer_id

Composite key of product_id and manufacturer_id.

One product can only have **one** manufacturer.

Product **MUST HAVE** a manufacturer.

2. Categorized Under(Product category): Relationship between Product and its category type

- product_id
- category_id

Composite key of product_id and category_id.

Many product can only have **one** category.

Product **MIGHT NOT HAVE** a category.

3**. Bundled Under(Product, Bundle)**: Relationship between Product in Bundle offerings.

- product_id
- bundle_id

Many products can have **many** bundle.

A bundle must **HAVE** a product, however a product **MIGHT NOT HAVE** a bundle.

4. Managed by(Product, Inventory): Relationship where the product in managed by inventory

- product_id
- inventory_id

One product can be managed by **one** inventory

A Product **MUST HAVE** an Inventory

***5.* Restocks(Inventory, Re-stockorder):** Relationship An inventory will make order for restocking particular products.

- inventory_id
- reorder_id

Composite key of inventory_id, reorder_id

Many reorders can be made by **many** inventories

An Inventory **MUST HAVE** a Restock order

6. Order(Order Instance, Product Instance, Customer): Relationship where a customer can make order instance and make payments for it

- Customer_id
- Payment_id
- Order id
- Date

Composite key of Customer_id, Payment_id, Order id

One order instance can be made by one customer with one payment instance

Order Instance MUST HAVE a customer and MIGHT NOT HAVE a payment instance

7. Ordered from(Order Instance, Branch): Relationship where an order instance is made from a branch

- branch_id
- order_id

Composite key of branch_id, order_id

One Branch can have **Many** order instance

An Order Instance **MUST HAVE** a Branch

8. Sales History(Branch, Sales): Relationship where a Branch can have a history of Sales

- branch_id

One Branch can have **Many** sales history

A Sales history **MUST HAVE** a Branch

9. Delivered By(Online Order, Shipment): Relationship where an online order is delivered by a shipment

- order_id
- shipment_id

One Online order can be delivered by **One** Shipment

An online order **MUST HAVE** a shipment

10. Shipped To(OnlineOrder-Shipment, Online Customer):
Relationship where an online order delivered by a shipment is Shipped to an online customer.

- order_id
- shipment_id
- customer_id

One OnlineOrder-Shipment can be shipped to **One** customer

An OnlineOrder-Shipment **MUST HAVE** a Customer

11. Has(Customer, Order History): Relationship where a customer has a history of orders

- customer_id

One Customer can have **Many** Order History

Order history **MUST HAVE** a customer

12. Has(Customer, Payment History): Relationship where a customer has a history of payments

- customer_id

One Customer can have **Many** Payment History

Payment history **MUST HAVE** a customer

13. Credit Billing(Contracted Customer, Payment Instant): Relationship where a contracted customer can make payments for their due credits.

- customer_id
- payment_id

Composite key of customer_id, payment_id

One Customer can make **Many** payment instances

Customer **MIGHT NOT HAVE** payment instances