Database Management and Database Design (DAMG6120) Project

P3 – Database Design, Final ERD

TOPIC: E-Commerce Database Management System
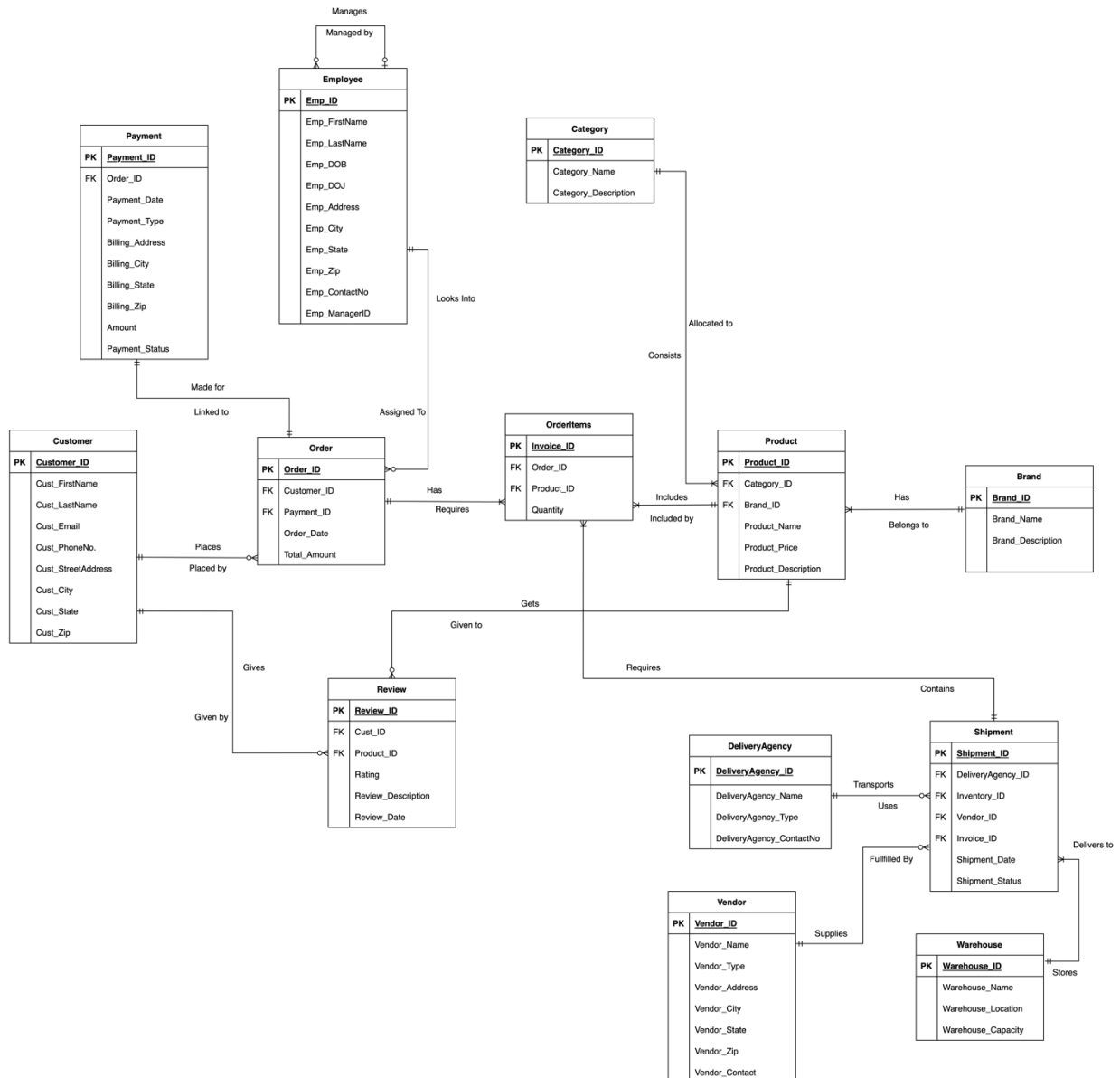
GitHub Link: https://github.com/swapnilbiradar/DAMG6210Group10_Final_Project

Group 10: Team Members

Name	NUID	Email ID
Hema Sree Bojanapally	002709710	bojanapally.h@northeastern.edu
Prakhya Dasari	002774723	dasari.pr@northeastern.edu
Swapnil Shankarrao Biradar	002775707	biradar.sw@northeastern.edu
Sahil Qureshi	002743939	qureshi.sa@northeastern.edu
Vennela Bandla	002776750	bandla.v@northeastern.edu

ER Diagram:

Group 10
ELECTRONIC E-COMMERCE DATABASE SYSTEM

Manages
Managed by

Employee

PK	Emp_ID
	Emp_FirstName
	Emp_LastName
	Emp_DOB
	Emp_DOJ
	Emp_Address
	Emp_City
	Emp_State
	Emp_Zip
	Emp_ContactNo
	Emp_ManagerID

Category

PK	Category_ID
	Category_Name
	Category_Description

Payment

PK	Payment_ID
FK	Order_ID
	Payment_Date
	Payment_Type
	Billing_Address
	Billing_City
	Billing_State
	Billing_Zip
	Amount
	Payment_Status

Looks Into

Allocated to

Consists

Made for

Linked to

Assigned To

OrderItems

PK	Invoice_ID
FK	Order_ID
FK	Product_ID
	Quantity

Product

PK	Product_ID
FK	Category_ID
FK	Brand_ID
	Product_Name
	Product_Price
	Product_Description

Brand

PK	Brand_ID
	Brand_Name
	Brand_Description

Customer

PK	Customer_ID
	Cust_FirstName
	Cust_LastName
	Cust_Email
	Cust_PhoneNo.
	Cust_StreetAddress
	Cust_City
	Cust_State
	Cust_Zip

Order

PK	Order_ID
FK	Customer_ID
FK	Payment_ID
	Order_Date
	Total_Amount

Has
Requires

Includes
Included by

Has
Belongs to

Places
Placed by

Gets
Given to

Gives
Given by

Requires
Contains

Review

PK	Review_ID
FK	Cust_ID
FK	Product_ID
	Rating
	Review_Description
	Review_Date

DeliveryAgency

PK	DeliveryAgency_ID
	DeliveryAgency_Name
	DeliveryAgency_Type
	DeliveryAgency_ContactNo

Shipment

PK	Shipment_ID
FK	DeliveryAgency_ID
FK	Inventory_ID
FK	Vendor_ID
FK	Invoice_ID
	Shipment_Date
	Shipment_Status

Transports
Uses

Fullfilled By

Delivers to

Vendor

PK	Vendor_ID
	Vendor_Name
	Vendor_Type
	Vendor_Address
	Vendor_City
	Vendor_State
	Vendor_Zip
	Vendor_Contact

Supplies

Warehouse

PK	Warehouse_ID
	Warehouse_Name
	Warehouse_Location
	Warehouse_Capacity

Stores

Changes Made:

1) Relationship between Brand and Product is reversed.

2) Changed the table name from Reviews to Review

3) Changed the name of Inventory table to Warehouse.

4) Added a new table Payment and its relationship with Order Table.

5) Cardinality of the Product to Category changed.

6) Followed standard naming convention practices.

Business Problem:

The business problem presented is that an electronic e-commerce company wants to develop a database system to manage its online store operations. The company needs to manage different aspects of the e-commerce store, including product inventory, customer orders, payments, and shipping. To accomplish this, the company requires a database that can store and organize all relevant data and allow easy access to it when needed.

One of the primary requirements of the database system is to track inventory levels and reviews. This means that the system must be able to record the available quantity of each product in the inventory and be able to track the reviews of each product written by customers. To achieve this, the database needs to maintain a connection between the "Product" and "Inventory" entities, so that the inventory levels can be tracked for each product.

The database system must also track customer orders, payments, and shipping. This requires a connection between the "Order", "Payment", and "Shipment" entities. The system should be able to store customer order details, including the ordered items and their respective quantities, delivery information, and payment information. This information should be updated and maintained in real-time as orders are placed, payments are received, and shipments are made.

The electronic e-commerce company also works with multiple vendors and delivery agencies. To keep track of these relationships, the database must maintain a connection between the "Vendor" and "Delivery Agency" entities. This allows the company to keep track of which vendor supplies which product and which delivery agency is responsible for delivering the order.

In summary, the electronic e-commerce company's database system must include the following entities: "Employee", "Customer", "Category", "Order", "Order Items", "Product", "Brand", "Shipment", "Delivery Agency", "Vendor", "Inventory", "Reviews", and "Payment". To create

relationships between these entities, the database system should implement a combination of one-to-one, one-to-many, and many-to-many relationships. The key design decisions for this system include selecting appropriate data types for each entity and relationship, determining the cardinality and modality of each relationship, and ensuring the database can handle high volumes of data and user traffic.

List of Entities:

Employee: This entity represents the company's employees, including information such as name, DOB, DOJ, address, contact number and managerID.

Customer: This entity represents the company's customers, including information such as name, email, customerID, phone number and address.

Category: This entity represents the product categories, including information such as categoryID, name and description.

Order: This entity represents the customer orders, including information such as orderID , order date, total amount and customer ID.

Order Items: This entity represents the individual items in each customer order, including information such as product ID, orderID, quantity.

Product: This entity represents the products for sale, including information such as product ID, categoryID, BrandID, name, description, and price.

Brand: This entity represents the product brands, including information such as brandID, name and description.

Shipment: This entity represents the shipment of an order, including information such as shipmentID, deliveryAgencyID, InventoryID, VendorID, InvoiceID,shipment date and shipment status.

Delivery Agency: This entity represents the delivery agency, including information such as agencyID, name and contact information.

Vendor: This entity represents the company's vendors, including information such as vendorID, name, type, address and contact information.

Inventory: This entity represents the product inventory levels, including information such as inventoryID, name, location and capacity

Reviews: This entity represents the product reviews, including information such as reviewID, customerID,description, date, rating, and product ID.

Payment: This entity represents the payment information for each customer order, including information such as paymentID, orderID, date, type, address information, payment amount and payment status.

Relationships among Entities:

Customer and Order: One customer can have multiple orders, but each order can only be associated with one customer. This relationship is represented as a one-to-many relationship.

Customer and Reviews: One customer can have multiple reviews, but each review can only be given by one customer. This relationship is represented as a one-to-many relationship.

Employee: Each employee can be managed by one employee. Unary relation between employee with itself.

Employee and Order: One employee can look into multiple orders, and each order can only be assigned to one employee. This relationship is represented as a one-to-many relationship.

Order and Order Items: One order can have multiple order items, and each order item can only be associated with one order. This relationship is represented as a one-to-many relationship.

Product and Category: One product can belong to multiple categories, and each category can have multiple products. This relationship is represented as a many-to-many relationship.

Product and Brand: One product can belong to one brand, and each brand can have multiple products. This relationship is represented as a one-to-many relationship.

Order Items and Shipment: One order item can have one shipment, and each shipment can be associated with many order items. This relationship is represented as a many-to-one relationship.

Product and Order Items: One product can include multiple order items, and each order item can be included by one product. This relationship is represented as a one-to-many relationship.

Shipment and Delivery Agency: One shipment can be associated with one delivery agency, and each delivery agency can have multiple shipments. This relationship is represented as a one-to-many relationship.

Product and Reviews: One product can have multiple reviews, and each review can only be associated with one product. This relationship is represented as a one-to-many relationship.

Shipment and Vendor: One vendor fulfills multiple shipments, and each shipment is supplied from one Vendor. This relationship is represented as a many-to-one relationship.

Inventory and Shipment: One inventory store multiple shipment, and each shipment delivers to one inventory. This relationship is represented as a one-to-many relationship.

Order and Payment: One order can have one payment, and each payment can only be associated with one order. This relationship is represented as a one-to-one relationship .

Key Design Decisions:

Employee

- Store employee information such as name, email, address, phone number and date of joining.
- Create a relationship with the order entity to track which employee is responsible for each order.

Customer

- Store customer information such as name, email, street address and phone number.
- Use a unique customer ID to provide reviews and track orders.
- Implement security measures to protect customer data and comply with data privacy regulations such as GDPR or CCPA.

Category

- Store electronic product categories such as Mobiles, Laptops, Tv's etc.
- Use a category ID to categorize each product in the database.
- Create a relationship with the product entity to associate each product with a category.

Order

- Store order information such as order ID, customer ID, order date, shipping address, and billing address.
- Use a unique order ID to track each order and its associated items.
- Create relationships with the customer, employee, and shipment entities to track relevant information.

Order Items

- Store information about the items included in each order such as product ID, quantity, and price.
- Create relationships with the order and product entities to track relevant information.

Product

- Store product information such as product name, description, price, and availability.
- Use a unique product ID to track each product and its associated information.
- Create relationships with the category, brand, inventory, and review entities to track relevant information.

Brand

- Store information about the brands associated with each product such as brand name and description.
- Use a unique brand ID to track each brand and its associated products.
- Create relationships with the product entity to track relevant information.

Shipment

- Store information about each shipment such as shipment ID, order ID, shipment date, and expected delivery date.
- Create relationships with the order and delivery agency entities to track relevant information.

Delivery Agency

- Store information about the delivery agency such as name, contact information, and delivery rates.
- Use a unique agency ID to track each delivery agency and its associated information.
- Create relationships with the shipment entity to track relevant information.

Vendor

- Store information about vendors such as name, contact information, and product inventory.
- Use a unique vendor ID to track each vendor and its associated products.
- Create relationships with the product and inventory entities to track relevant information.

Inventory

- Store information about product inventory such as product ID, vendor ID, quantity, and restocking dates.
- Create relationships with the product and vendor entities to track relevant information.

Review

- Store customer reviews and ratings for each product.
- Use a unique review ID to track each review and its associated product.
- Create a relationship with the product entity to track relevant information.

Payment

- Store payment information such as payment ID, customer ID, order ID, payment date, and payment amount.
- Use a unique payment ID to track each payment and its associated information.
- Implement security measures to protect sensitive payment information.