DATA MANAGEMENT AND SYSTEM DESIGN

CS 631-009 Group 10

Online Computer Store Project

Deliverable 1

Piyusha Sayal (ps332)
Yashwanth Renukanta (yr255)
Pradyun Reddy Bollepally (pb573)

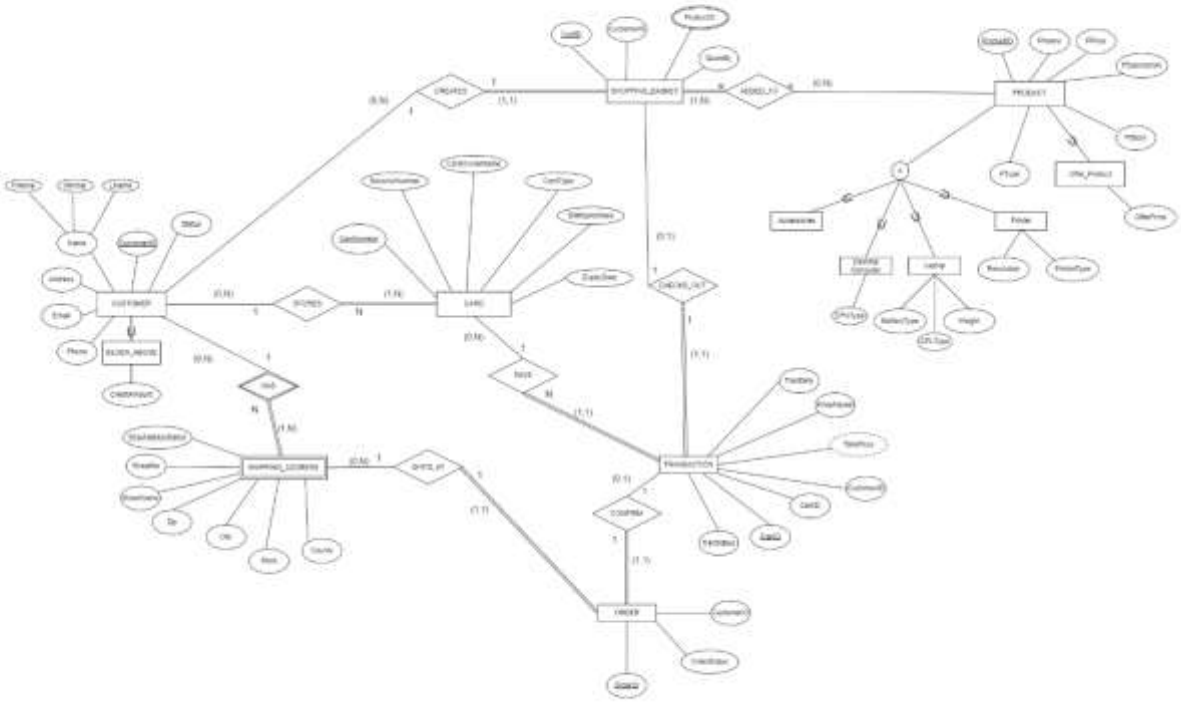
TABLE OF CONTENT

GOALS FOR THE PROJECT

1. **Understanding the Problem and Data Organization:** The primary goal of this deliverable is to thoroughly understand the problem statement offered for the Online Computer Store and to develop an effective technique for organizing the data using a Database Management System (DBMS). It is necessary to comprehend the business requirements and translate them into an organized database design.
2. **Assumptions for Efficient Database Operation:** To identify and describe any relevant assumptions for the database system's efficient operation. These assumptions include data consistency standards, how to handle unusual circumstances, and other system-critical features.
3. **Entity Relationship Analysis:** To determine the relationships between entities, analyze the entire application, including customer, product, and sales data. An Extended Entity-Relationship (ER) diagram is made to visually represent the entity kinds, relationship types, and features. The diagram follows standard notation and includes important features and structural constraints.
4. **Key Attribute Identification:** To clearly state the primary key attributes for each entity type in the ER diagram. This step is crucial for maintaining the uniqueness and integrity of the database's data.
5. **Constraint Specification:** Define and specify structural restrictions for relationship types, such as cardinality ratios and participation requirements. Both traditional notations and the (min, max) notation to depict these constraints on the ER diagram.
6. **Assumptions and Constraints:** Any assumptions made during the conceptual design phase that go beyond the information supplied in the project description are documented. It also includes any additional keys required for entity types for data integrity.
7. **Difficulties and obstructions:** A detailed account of any difficulties or impediments found throughout the conceptual design effort is provided. This includes unspecified criteria, complex relationships, and limits.
8. **Project Phase Alignment:** To recognize that the conceptual design phase serves as the foundation for the project's subsequent phases. Assure that the goals and outcomes of this phase are consistent with the overall project objectives and contribute to the effective implementation of the database system.
9. **Clarity:** To maintain clarity and precision in all areas of the product, from the ER diagram through assumptions and explanations.
10. **Flexibility:** Recognize that the conceptual design may change as the project advances and this design as needed in later phases can accommodate changing requirements or new insights.

ER DIAGRAM

https://tinyurl.com/53ufy5fp



DATABASE TABLES

Entity 1: CUSTOMER

Name(FName, MInitial, LName), CustomerID, Status, Address, Email, Phone

- SILVER_ABOVE is a Customer (Specialization)
- CustomerID is the primary key

Entity 2: SHIPPING_ADDRESS

ShipAddressName, StreetNo, StreetName, Zip, City, State, Country, CustomerID

- Weak entity, with Customer as its owner entity
- ShipAddressName is a partial key

Entity 3: CARD

CardNumber, SecurityNumber, CardOwnerName, CardType, BillingAddress, ExpiryDate

- CardNumber is the key

Entity 4: SHOPPING_BASKET

CartID, CustomerID, {ProductID},Quantity

- CartID is the primary key
- Product ID is the multivalued attribute
- CustomerID and ProductID is Foreign Key

Entity 5: PRODUCT

ProductID, PPrice,PName, PDescription, PStock, PType

- ProductID is the primary key
- Offer_Product is a Product(Specialization)
- Disjoint specialization: (Desktop Computer, Laptop, Printer, Accessories)
- Desktop Computer: CPUType
- Laptop: CPUType, BatteryType, Weight
- Printer: Resolution, PrinterType

Entity 6: TRANSACTION

TranID, TranDate, PricePayed, TotalPrice, CustomerID, CartID,TranStatus

- TranID is primary key
- TotalPrice is derived attribute
- If PricePayed matches the TotalPrice then an order is created
- CartID and CustomerID is foreign key

Entity 7: Order

OrderID, CustomerID, OrderStatus

- OrderID is primary key
- CustomerID is foreign key

RELATIONS

1. **CREATES (Customer: Shopping_Basket)**
   The customer creates one Shopping Basket, 1: 1 relationship.

2. **STORES (Customer: Card)**
   Customers can store multiple credit cards, 1: N relationship

3. **SHIPS_AT (Order: Shipping_Address)**
   Order is shipped to one shipping address, 1: 1 relationship

4. **CHECKS_OUT (Shopping_Basket: Transaction)**
   Shopping Basket checks out to one Transaction, 1: 1 relationship

5. **PAYS (Card: Transaction)**
   A card can pay for multiple transactions, 1: N relationship

6. **CONFIRM (Transaction: Order)**
   A transaction once verified gets confirmed as an order, 1: 1 relationship

7. **ADDED_TO (Product: Shopping_Basket)**
   Multiple products can be added to multiple shopping baskets, M: N relationship.

ASSUMPTIONS

1. Assume that the CustomerID is unique for each customer and it is considered as a primary key.
2. The Shipping address is assumed to be the same as the billing address
3. The Transaction is converted to an order only if the PricePayed is equivalent to the TotalPrice (Derived Attribute)
4. Assume that a new cart is created for a customer automatically when they add their first item to it.
5. Assume that a transaction is paid by only one credit card.
6. Assuming unique product attributes, we introduced specialized entities like "DesktopComputer," "Laptop," and "Printer" to accommodate these distinctions.
7. Disjoint Constraint is assumed in product specialization taking Accessories as the products which do not belong to the three main categories provided in product description.

CONSTRAINTS

Participation Constraints

- **CREATES (Customer: Shopping_Basket)**
  Customer-Creates Partial Participation
  Creates-Shopping_Basket: Total Participation

- **STORES (Customer: Card)**
  Customer-Stores: Partial Participation
  Stores-Card: Total Participation

- **SHIPS_AT (Order: Shipping_Address)**
  Order-Ships_At: Partial Participation
  Ships_At-Shipping_Address: Partial Participation

- **CHECKS_OUT (Shopping_Basket: Transaction)**
  Shopping_Basket-Checks_Out: Partial Participation
  Checks_Out-Transaction: Total Participation

- **PAYS (Card: Transaction)**
  Card-Pays: Partial Participation
  Pays-Transaction: Total Participation

- **CONFIRM (Transaction: Order)**
  Transaction-Confirm: Partial Participation
  Confirm-Order: Total Participation

- **ADDED_TO (Product: Shopping_Basket)**
  Shopping_Basket-Added_To: Total Participation
  Added_To-Product: Partial Participation

Cardinality Constraints

- One CUSTOMER can store multiple CARDS
  1: N relationship

- One CUSTOMER can have many SHIPPING_ADDRESS
  1: N relationship

- One CUSTOMER can make one SHOPPING_BASKET
  1:1 relationship

- One SHOPPING_BASKET can check out to one TRANSACTION
  1:1 relationship

- One CARD can be used to pay for multiple TRANSACTION
  1: N relationship

- One TRANSACTION can lead to one ORDER
  1:1 relationship

- One ORDER can be shipped to one SHIPPING_ADDRESS
  1:1 relationship

- Many PRODUCTS can be added to many SHOPPING_BASKET
  M: N relationship

Min Max Constraints

- CUSTOMER can STORE minimum 0 and maximum N CARD:
  CUSTOMER-STORE (0, N)

- CARD can belong to minimum 1 and maximum N CUSTOMER:
  STORE-CARD (1, N)

- CUSTOMER can have minimum 0 and maximum N SHIPPING_ADDRESS:
  CUSTOMER-HAS (0, N)

- SHIPPING_ADDRESS can belong to minimum 1 and maximum N CUSTOMER:
  HAS-SHIPPING_ADDRESS (1, N)

- CUSTOMER can create minimum 0 and maximum N SHOPPING_BASKET:
  CUSTOMER-CREATES (0, N)

- SHOPPING_BASKET can belong to minimum 1 and maximum 1 CUSTOMER:
  CREATES-CUSTOMER (1,1)

- ORDER can ship to minimum 1 and maximum 1 SHIPPING_ADDRESS:
  ORDER-SHIPS_AT (1,1)

- SHIPPING_ADDRESS can have minimum 0 and maximum N orders:
  SHIPPING_ADDRESS-SHIPS_AT (0, N)

- CARD can pay for minimum 0 and maximum N TRANSACTION:
  CARD-PAYS (0, N)

- TRANSACTION can be paid by minimum 1 and a maximum of 1 CARD:
  TRANSACTION-PAYS (1,1)

- TRANSACTION can CONFIRM minimum 0 and maximum 1 ORDER:
  TRANSACTION-CONFIRM (0,1)

- ORDER can be related to minimum 1 and maximum 1 TRANSACTION:
  ORDER-CONFIRM (1, 1)

- SHOPPING_BASKET can have a minimum 1 and maximum N PRODUCT:
  SHOPPING_BASKET-ADDED_TO (1, N)

- PRODUCT can be ADDED_TO minimum 0 and maximum N SHOPPING_BASKET: **PRODUCT-ADDED_TO (0, N)**

- SHOPPING_BASKET can CHECKS_OUT to minimum 0 and maximum 1 TRANSACTION: **SHOPPING_BASKET-CHECK_OUT (0, 1)**

- TRANSACTION can be CHECKS_OUT from minimum 1 and maximum 1 SHOPPING_BASKET: **TRANSACTION-CHECKS_OUTS (1, 1)**

DIFFICULTIES ENCOUNTERED

- During the conceptual design phase, choosing the right relationships between entities was one of the biggest challenges.

- In some cases, the initial design phase does not have access to all the necessary information. This may result in ERDs that are insufficient or erroneous. It is crucial to acquire as much data as you can.

- Choosing the appropriate cardinality, such as one-to-one, one-to-many, or many-to-many.

- Changing Business Requirements: Over the course of the project, business requirements may evolve, leading to necessary adjustments in the conceptual design. Adapting the entity relationships and cardinalities to accommodate these changes while maintaining data integrity can be a recurring challenge.