Online Computer Store Project

Deliverable 1

Course: CS631-002
Group Number: 17
Name: Advaithbarath Raghuraman Bhuvaneswari

1. Goals of this Phase

The main objective of this phase is to design a comprehensive Extended Entity-Relationship (EER) diagram for the Online Computer Store, capturing the essential components and workflows of the system. The design will focus on defining key entities such as customers, orders, products, payments, and suppliers, along with their attributes and relationships. It will also include business rules, constraints, and cardinality to ensure data integrity and accurate representation of the store's operations. The EER diagram will provide a structured framework for managing customer interactions, order processing, product management, payment handling, and customer feedback, forming the foundation for the store's database and supporting efficient and scalable operations.

2. Extended ER Diagram

Entity Types, Relationships Types, and Attributes

Note: The cardinality ratio are based on min-max notations.

1. **CUSTOMER**: *CustomerID*, FirstName, Surname, HomeAddress (StreetName, StreetNumber, City, State, ZipCode, Country), TelephoneNumber, EmailAddress, Status, CreditAmount.
    - A **CUSTOMER** can have many **CREDIT_CARDS** but not all of them have it.(0:N).
    - A **CUSTOMER** can have multiple **SHIPPING_ADDRESSES** (1:N).
    - A **CUSTOMER** can create multiple **SHOPPING_BASKETS** (1:N).
    - A **CUSTOMER** can place many **ORDERS** but not all of them have it(0:N).
    - A **CUSTOMER** does or dosent refer other **CUSTOMERS** ((0:N)(1,1), recursive relationship).
2. **CREDIT_CARD**: *CreditCardNumber*, SecurityNumber, OwnerName, BillingAddress (StreetName, StreetNumber, City, State, ZipCode, Country), Type, ExpiryDate.
    - A **CREDIT_CARD** belongs to exactly one **CUSTOMER** (1:1).
    - A **CREDIT_CARD** can purchase none or many **ORDER**(0:N).
3. **SHIPPING_ADDRESS**: *ShippingAddressname*, AddressName, StreetName, StreetNumber, City, State, ZipCode, Country.

- A **SHIPPING_ADDRESS** belongs to exactly one **CUSTOMER** (1:1).
- A **SHIPPING_ADDRESS** does have none or multiple **ORDER** (0:N).
- A **SHIPPING_ADDRESS** can or cannot replace another **SHIPPING_ADDRESS** (0:1, recursive relationship).

4. **SHOPPING_BASKET**: *BasketID*, CreatedDate, Quantity.
   - A **SHOPPING_BASKET** contains none or many **PRODUCT** (0:N).
   - A **SHOPPING_BASKET** belongs to exactly one **CUSTOMER** (1:1).
   - A **SHOPPING_BASKET** convert to exactly one **ORDER** (1:1).
5. **ORDER**: *OrderID*, OrderDate, Status, TotalAmount.
   - An **ORDER** can contain many **ORDER_PRODUCTS** (1:N).
   - An **ORDER** can have exactly one **SHIPPING_ADDRESS** (1:1).
   - An **ORDER** is placed by exactly one **CUSTOMER** (1:1).
   - An **ORDER** is paid using exactly one **CREDIT_CARD** (1:1).
   - An **ORDER** can or cannot replace another **ORDER** (0:1, recursive relationship).
   - An **ORDER** can contain exactly one **SHOPPING_BASKET** (1:1).
6. **ORDER_PRODUCT(***Weak entity)*: Quantity, Price.
   - An **ORDER_PRODUCT** can include none or many **PRODUCT** (0:N).
   - An **ORDER_PRODUCT** is part of exactly one **ORDER** (1:1).
7. **PRODUCT**: *ProductID*, Name, RecommendedPrice, Description, ProductType, QuantityInStock, DiscountedPrice.
   - A **PRODUCT** can be included in none or many **SHOPPING_BASKET** (0:N).
   - A **PRODUCT** can be included in none or many **ORDER_PRODUCTS** (0:N).
   - A **PRODUCT** can or cannot have an **OFFER** (0:1).
   - A **PRODUCT** is categorized by exactly one **PRODUCT_TYPE** (1:1).
   - A **PRODUCT** can specialize as a **DESKTOP**, **LAPTOP**, or **PRINTER** .
8. **PRODUCT_TYPE**: *ProductTypeID*, Category.
   - A **PRODUCT_TYPE** can categorize many **PRODUCTS** (1:N).
9. **OFFER**: *OfferID*, ReducedPrice, ValidForStatus.
   - An **OFFER** exactly belong to none or one **PRODUCT** (0:1).
10. **DESKTOP, LAPTOP, PRINTER**: CPUType (Desktop, Laptop), Weight (Laptop), BatteryRunningTime (Laptop), Resolution (Printer), PrinterType (Printer).

- A **DESKTOP**, **LAPTOP**, or **PRINTER** specializes in exactly one **PRODUCT**.

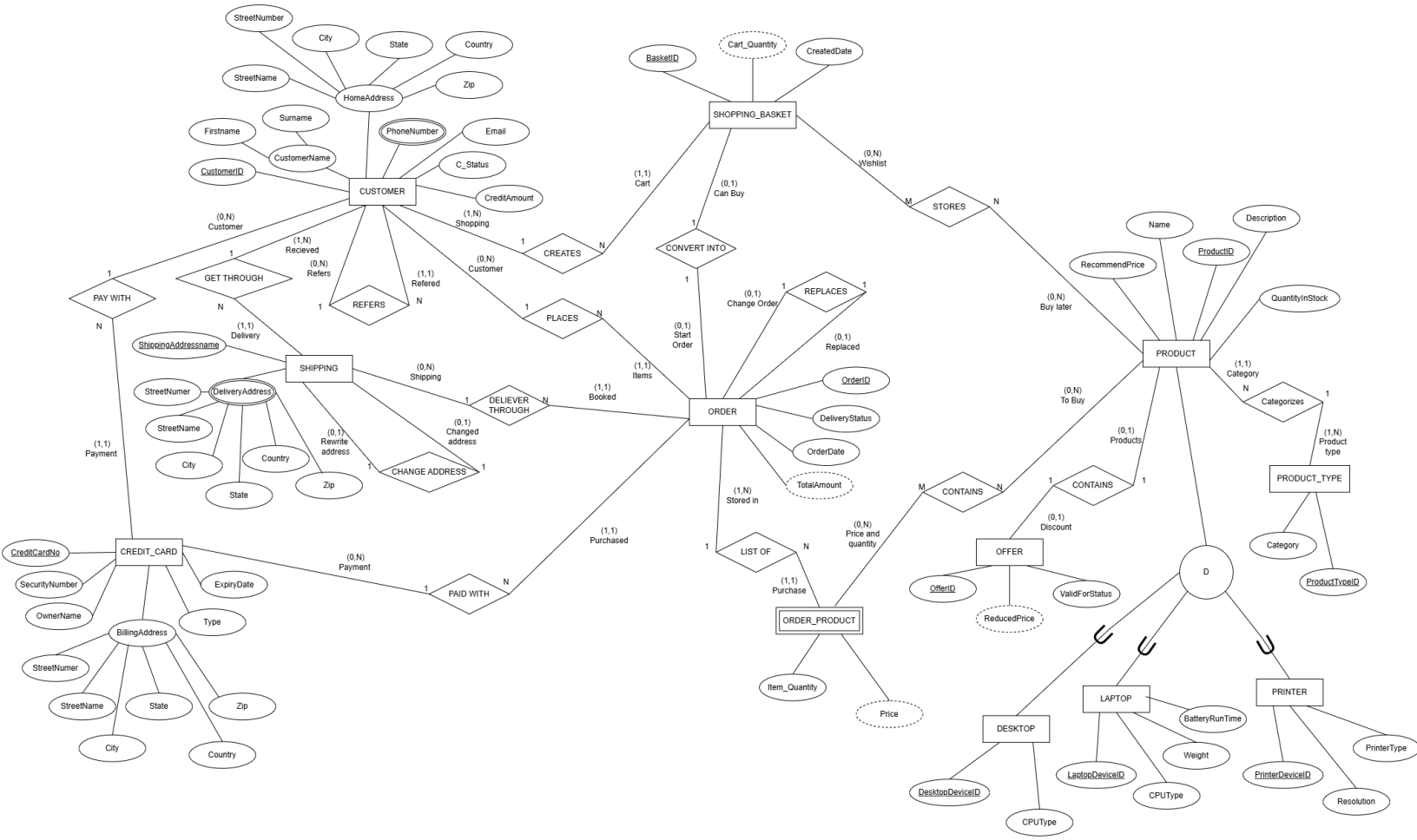
Key Attributes

- **Primary Keys** are italicized in the entity descriptions above (e.g., *CustomerID*, *OrderID*, *ProductID*, etc.).

Structural Constraints

- Cardinalities are specified next to each relationship in the format (Min, Max), indicating the minimum and maximum participation constraints.

3. Assumptions



- A customer can place multiple orders, but an order is always linked to a single customer (1:N)
- A customer can have multiple shipping addresses, but an order must be linked to exactly one shipping address (1:N).
- A shopping basket may replace another basket if the customer updates their cart before checkout (1:1).
- Each product belongs to exactly one category in Product_Type (1:N).
- A product must be one of Laptop, Desktop, or Printer (i.e., disjoint specialization).
- An order must contain at least one product, and a product may be associated with multiple orders (1:N).
- A credit card is linked to a single customer, but a customer can have multiple credit cards (1:N).4. Additional Constraints

4. Additional Constraints

Some business rules and constraints cannot be directly represented in the ER diagram but are critical to implementation:

1.Unique Constraints:
- Each credit card number must be unique for a customer.
- Each shipping address must have a unique name per customer.

2. Data Integrity Rules:
- Orders must be linked to a valid shipping address and payment method.
- Special discount offers are only applicable to Gold and Platinum customers.

3. Referential Integrity:
- If a customer is deleted, their associated shopping baskets and shipping addresses are also removed.
- Orders, once placed, cannot be altered but can be marked as failed or completed.

4. Transaction Constraints:
- An order cannot proceed unless the total payment amount is confirmed.
- A customer can place multiple orders, but each order is independent in terms of payment and delivery.

5. Challenges and Solutions

Several challenges were encountered while designing the conceptual schema. Below are the key challenges and solutions:

1. Defining Cardinality Constraints:
- Ensuring proper one-to-many relationships was crucial (e.g., a customer can have multiple credit cards, but each card belongs to one customer).
- Solution: Used cardinality ratios to enforce correct entity relationships.

2. Handling Product Specialization:
- The system needed to differentiate between desktops, laptops, printers, and accessories.
- Solution: Implemented specialization/generalization in the ER model to distinguish product attributes while maintaining a common structure.

3. Managing Shopping Baskets:
- Ensuring that shopping baskets are always valid and contain products before checkout.
- Solution: Introduced business rules preventing empty shopping baskets and enforcing order validation checks.

4. Stock and Availability Management:
- Customers should not be able to purchase out-of-stock products.
- Solution: Implemented a stock validation check before adding products to the shopping basket.