**Assignment Activity Unit 2**

Hassan O. Olowofela

MSIT 5210: Databases

Instructor: Dr. Ada Ajunwa

Due: Thursday, 19 September 2024

Date of submission: Sunday, 15 September 2024

**Database Design for an E-Commerce Platform**

**Introduction**

A robust and scalable database is essential for effective operations and top-notch customer support in the quickly changing world of e-commerce. The Entity-Relationship (E-R) model is essential for organizing data to support business goals by outlining relationships between entities in a logical and structured manner.

This write-up attempts to define the necessary entities and their attributes for an e-commerce platform where customers can place multiple orders, each consisting of one or more products, each product falls under a specific category, and customers can maintain multiple shipping addresses. An Entity-Relationship (E-R) diagram, primary keys, relationships, and the identification of entities are all part of the design process. The conceptual and physical design models will then be compared. The report will also include two new needs to increase the platform's capability. With database management systems and the E-R model, a blueprint supporting the platform's operations may be created (Vidhya et al., 2016).

**List of Entities and Their Attributes**

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | **Entities** | **Attributes** |
| 1 | **Customer** |  |
|  |  | CustomerID (Primary Key) |
|  |  | FirstName |
|  |  | LastName |
|  |  | Email |
|  |  | PhoneNumber |
|  |  | Password |
|  |  | AccountCreationDate |
| 2 | **Order** |  |
|  |  | OrderID (Primary Key) |
|  |  | CustomerID (Foreign Key, references Customer) |
|  |  | OrderDate |
|  |  | OrderStatus |
|  |  | TotalAmount |
|  |  | ShippingAddressID (Foreign Key, references ShippingAddress) |
| 3 | **Product** |  |
|  |  | ProductID (Primary Key) |
|  |  | ProductName |
|  |  | Description |
|  |  | Price |
|  |  | StockQuantity |
|  |  | CategoryID (Foreign Key, references Category) |
| 4 | **Category** |  |
|  |  | CategoryID (Primary Key) |
|  |  | CategoryName |
|  |  | CategoryDescription |

|  |  |  |
| --- | --- | --- |
| 5 | **ShippingAddress** |  |
|  |  | ShippingAddressID (Primary Key) |
|  |  | CustomerID (Foreign Key, references Customer) |
|  |  | AddressLine1 |
|  |  | AddressLine2 (optional) |
|  |  | City |
|  |  | State |
|  |  | PostalCode |
|  |  | Country |

|  |  |  |
| --- | --- | --- |
|  | **OrderProduct (Associative Entity)** |  |
|  |  | OrderProductID (Primary Key) |
|  |  | OrderID (Foreign Key, references Order) |
|  |  | ProductID (Foreign Key, references Product) |
|  |  | Quantity |
|  |  | UnitPrice |

**Primary Keys and Justifications**

1. **CustomerID:** This guarantees that all client-related data may be referred to individually by acting as a unique identifier for every customer. Maintaining user-specific data across orders, delivery addresses, and other actions depends on this (Vidhya et al., 2016).
2. **OrderID**: Every order must be uniquely identified to record and manage specific transactions. OrderID guarantees that each order is unique, even if placed by the same buyer.
3. **ProductID**: The product's unique identification makes it possible to catalog and retrieve product data quickly. It also eliminates problems brought on by having identical product names.
4. **CategoryID**: This distinctly identifies and guarantees that each product category is identified in the database exclusively, aiding in the logical grouping of items.
5. **ShippingAddressID**: Allows multiple addresses to be stored for each customer, ensuring each order is directed to the correct location.
6. **OrderProductID**: A many-to-many associative entity that tracks which products are a part of which orders by connecting orders and products.

**Relationships and Mapping Cardinality**

1. **Customer ↔ Order**: **(1 to Many)**

**Justification**: Orders can be placed by a single client; however, each order is linked to that specific consumer. Doing this directly connects clients and their transactions (Hoffer et al., 2019).

1. **Order ↔ Product: (Many to Many)**

**Justification**: It is possible for a product to be a part of many orders and for an order to contain multiple products. This is managed by the associated entity OrderProduct.

1. **Product ↔ Category: (Many to 1)**

**Justification**: A product can be a part of more than one category, but each product only belongs to one category. This guarantees that items are arranged correctly so they may be browsed.

1. **Customer ↔ ShippingAddress: (1 to Many)**
   * **Justification**: This ensures that multiple shipping addresses are allowed for a single client, but each address is distinct.
2. **Order ↔ ShippingAddress: (Many to 1)**
   * **Justification**: While only one mailing address is used for each order, that address may eventually be used for several orders mailed at different times.

**Additional Requirements**

|  |  |  |
| --- | --- | --- |
| **Entities** | **Attribute** | **Relationships:** |
| **Customer Reviews** |  |  |
|  | ReviewID (Primary Key) | Customer ↔ Review  (1 to Many ) |
|  | CustomerID (Foreign Key, references Customer) | Product ↔ Review:  (1 to Many ) |
|  | ProductID (Foreign Key, references Product) |  |
|  | Rating |  |
|  | ReviewComment |  |
|  | ReviewDate |  |

**Justification**: Multiple reviews can be left by customers and by-products. However, each review is associated with a single consumer and product.

|  |  |  |
| --- | --- | --- |
| **Entities** | **Attribute** | **Relationships:** |
| **Promotions** |  |  |
|  | PromotionID (Primary Key) | Product ↔ Promotion:  (Many to Many) |
|  | PromotionName |  |
|  | DiscountPercentage |  |
|  | StartDate |  |
|  | EndDate |  |
| **ProductPromotion (Associative Entity):** |  |  |
|  | ProductPromotionID (Primary Key) |  |
|  | ProductID (Foreign Key) | References Product |
|  | PromotionID (Foreign Key) | References Promotion |

**Justification**: A product may be eligible for more than one promotion, and each promotion may be used on various items.

**E-R Diagram**

The relationships between the entities are shown in the E-R diagram for this database using Crow's Foot Notation. The graphic shows the cardinality, primary, and foreign keys among several entities, including the customer, order, product, shipping address, review, and promotion.

.A computer screen shot of a computer

Description automatically generated

**Differences Between Conceptual and Physical Design Models**

1. **Level of Abstraction**:

|  |  |  |
| --- | --- | --- |
|  | **Conceptual Design** | **Physical Design:** |
|  | Pays less attention to the data storage in favor of specifying high-level connections between entities. Vidhya et al. (2016) state that the conceptual model specifies relationships between customers and orders. | It focuses on specifics like data types, file formats, and indexing algorithms that will be used to store the data in the actual database (Connolly & Begg, 2014). |

1. **Performance and Storage Optimization**:

|  |  |  |
| --- | --- | --- |
|  | **Conceptual Design** | **Physical Design:** |
|  | They primarily focus on gathering data linkages and business needs rather than performance concerns (Hoffer et al., 2019). | It focuses on improving storage allocation, indexing, and query speed to make sure the system functions effectively at scale (Elmasri & Navathe, 2017). |

**Conclusion**

Finding important entities, specifying their characteristics, and building associations with the correct cardinality are all necessary when designing a database for an e-commerce platform. If physical design concentrates on the system's performance and storage, conceptual design makes comprehending the relationships between elements possible. The platform may supply a more comprehensive experience for users and administrators by adding extra features like customer reviews and promotions. As the platform grows in the future, its architecture supports it by ensuring scalability, performance, and flexibility.

**References**

Connolly, T., & Begg, C. (2014). *Database systems: A practical approach to design, implementation, and management* (6th ed.). Pearson.

Elmasri, R., & Navathe, S. B. (2017). *Fundamentals of database systems* (7th ed.). Pearson.

Hoffer, J. A., Venkataraman, R., & Topi, H. (2019). *Modern database management* (13th ed.). Pearson.

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). *Database system concepts* (7th ed.). McGraw-Hill.

Vidhya, V., Jeyaram, G., & Ishwarya, K. (2016). *Database management systems*. Alpha Science International.