GROUP 5

22016968 Sarpong Jeffrey Somuah

22042860 Mensah Philemon

22043997 Aidoo-Taylor Kwamena

22231991 Henry Otwey Baidoo

22132121 Amoah Pearl Owusua

22020624 Manal Abdul - Kadi Mohammed

22071976 Eklo Christopher Yao

22049764 Krampah Jonathan

11365437 Captain Godsgift

**Database Design Report: E-Commerce Retail Shop System**

**Rationale Behind the Design**

The Entity-Relationship Diagram (ERD) represents the design of an e-commerce retail shop system, ensuring clear relationships among entities and efficient data handling. Below are the key design choices:

**Entities and Attributes:**

1. **Customers**:
   * Attributes: Id, Name, Number, Email.
   * Rationale: Essential to uniquely identify customers and store their contact information. Attributes are atomic to comply with first normal form (1NF).
2. **Orders**:
   * Attributes: Id, Date initiated, Date expected, Date delivered, Status, Quantity.
   * Rationale: Captures all relevant information about customer orders. The inclusion of Status helps track the order lifecycle.
3. **Products**:
   * Attributes: Id, Name, Stock, Price.
   * Rationale: Provides information about the product catalog. The Stock attribute ensures inventory tracking, while Price supports dynamic pricing.
4. **Shipping**:
   * Attributes: Id, Date initiated, Date expected, Date delivered, Status.
   * Rationale: Tracks shipping details for orders to ensure delivery timelines and statuses are updated.

**Relationships:**

1. **Places (Customers - Orders)**:
   * A customer places one or more orders. This relationship supports a one-to-many connection, ensuring data integrity.
2. **Contains (Orders - Products)**:
   * Each order can contain multiple products. This many-to-many relationship is implemented through a junction table for normalization.
3. **Has (Shipping - Orders)**:
   * Each shipping record corresponds to an order containing shipment. This one-to-one relationship ensures clarity in logistics tracking.

**Normalization Considerations**

The database design follows normalization principles to ensure data consistency and efficient representation:

1. **First Normal Form (1NF):**
   * All attributes are atomic (e.g., Email and Number are not lists).
   * Each table has a unique identifier (Id).
2. **Second Normal Form (2NF):**
   * All non-key attributes are fully dependent on the primary key. For instance, in Orders, attributes like Status and Quantity depend on Id.
3. **Third Normal Form (3NF):**
   * No transitive dependencies exist. For example, customer contact information is stored in the Customers table rather than in Orders.

**Efficient Representation in PostgreSQL**

The design ensures efficient representation of data in PostgreSQL:

1. **Indexing:**
   * Primary keys (Id) are indexed by default, enabling fast lookups.
2. **Relationships:**
   * Foreign key constraints maintain referential integrity.
3. **Data Types:**
   * Attributes use appropriate PostgreSQL data types, e.g., VARCHAR for Name, INTEGER for Quantity, and DATE for Date initiated.
4. **Partitioning:**
   * Large tables like Orders can be partitioned by date to improve query performance and maintainability.

**Attached Files and Insights**

1. **queries.sql:**
   * Contains SQL queries that interact with the database. Review to ensure compliance with best practices for efficient query execution.
2. **Retail\_shop.drawio.png:**
   * The diagram outlines the ERD structure, ensuring all relationships and attributes are clearly defined.
3. **schema.sql:**
   * Includes the SQL code to create the database schema. Review this file for consistency with the proposed design.

By adhering to these guidelines, the database will be well-optimized for real-world usage, ensuring scalability, maintainability, and efficiency.