1. **Introduction**

This report covers creating and populating a relational database for an E-commerce Shopping Cart System. The database models typical e-commerce by including Customers, Products, Orders, Shipping, and Customer Behavior. Data are programmatically generated using Python, Pandas, Numpy, and the Faker library to create realistic and randomized datasets to export as CSV files and import to an SQLite database.

2. Data Generation and Database Schema

The data for the E-commerce Shopping Cart System was generated programmatically using **Python**, **Pandas**, **Numpy**, and the **Faker** library. The following is a breakdown of how data was generated for each table and how missing data and duplicate entries were intentionally added to simulate real-world challenges.

Customers Table

- **`customer_id`** (Primary Key): A unique identifier for each customer generated using numpy.
- **`customer_name`**: Randomly generated using `fake.name()`.
- **`email`**: Email addresses were generated by combining the customer name (converted to lowercase) and a random domain.
- **`loyalty_points`** : A random number between 0 and 1000 was assigned.

Missing Data Example:

- **Missing Emails**: 50 emails were set to `NaN` to simulate incomplete customer registrations.

Duplicate Data Example:

- **Customer Duplicates**: 50 customer records were duplicated randomly to simulate data entry errors.

Products Table

- **`product_id`** (Primary Key): A unique identifier for each product.

- **`product_name`** : Product names such as `Laptop`, `Smartphone`, etc.
- **`product_brand`** : Brands like `Apple`, `Samsung`, etc.
- **`product_price`** : Prices were randomly generated between 50 and 2000.
- **`product_rating`** : Ratings between 1 and 5.

Missing Data Example:

- **Missing Ratings**: 30 product ratings were set to `NaN` to simulate missing customer reviews.

Orders Table

The **Orders table** links **Customers** to **Products**. A **composite key** of `customer_id` and `product_id` ensures uniqueness:

- **`customer_id`** (Foreign Key): Links to the **Customers** table.
- **`product_id`** (Foreign Key): Links to the **Products** table.
- **`quantity`** : Random quantity between 1 and 5.
- **`total_price`**: The total price is calculated from quantity and product price.
- **`order_status`**: Random order status selected from `Pending`, `Shipped`, `Delivered`, `Returned`.
- **`feedback_rating`**: Random feedback for the order.
- **`order_id`** (Primary Key): A unique identifier for each order

Missing Data Example:

- **Missing Order Status**: 40 orders had missing `order_status`.

Duplicate Data Example:

- **Duplicate Orders**: 30 orders were duplicated for the same product by the same customer

---

Shipping Table

The **Shipping table** links to **Orders** and contains shipping details:

- **`shipping_id`** (Primary Key): A unique identifier for shipping records.
- **`order_id`** (Foreign Key): Links to the **Orders** table.
- **`shipping_method`** : Random shipping method selected.
- **`shipping_cost`**: Shipping costs randomly generated between 5 and 50.
- **`discount_applied`**: A discount applied to shipping costs.

Customer Behavior Table

The **Customer Behavior table** tracks customer activities like cart abandonment:

- **`behavior_id`** (Primary Key): A unique behavior record identifier.
- **`customer_id`** (Foreign Key): Links to the **Customers** table.
- **`cart_abandonment`**: Random value of `True` or `False`.
- **`customer_segment`** : Random segmentation like `VIP`, `Regular`, or `New User`.

3. Justification for Separate Tables and Constraints

- Customers Table: The Customers table contains data related to individual customers. This table is essential for storing customer-specific information like names, emails, and loyalty points.
- Products Table: Products are stored in their own table to avoid redundancy. If we stored product data directly in the Orders table, we'd end up repeating the same product information for each order.
- Orders Table: The Orders table is the transactional table that links Customers and Products, capturing the specific details of each order. This table contains foreign keys that reference both Customers and Products to maintain referential integrity.
- Shipping Table: Shipping details are stored in a separate table and linked to the Orders table. Each order can have different shipping methods, which makes it appropriate to keep this information in a dedicated table.
- CustomerBehavior Table: This table tracks behaviors like cart abandonment and customer segmentation (e.g., VIP, Regular, New User), which are essential for understanding customer activity.

Foreign Keys and Constraints:

- Foreign Keys: Ensure referential integrity by linking data across tables. For example, each order is linked to a customer and a product, and each shipping record is linked to an order. Primary Keys: Uniquely identify each record in the table (e.g., `customer_id`,

`product_id`, `order_id`). Normalization: Data redundancy is minimized, and the database is normalized to ensure efficient storage and retrieval.

4. Ethics and Data Privacy

When creating and handling data, especially in a context like an **E-commerce Shopping Cart System**, **ethical considerations** and **data privacy** are of utmost importance. Even though this is **synthetic data** generated for testing purposes, it's essential to outline the responsible handling of real customer information in the context of data privacy laws and best practices.

Anonymization:

- **Synthetic Data**: All customer-related data, such as **names**, **emails**, and **addresses**, is randomly generated using the **Faker** , pandas and Numpy libraries. This data is entirely fictional and does not represent any real individual.
- **No Real Customer Data**: Since the dataset is entirely **randomized** and **anonymized**, there are no concerns about real customers' privacy or exposure of personal information.

Privacy: In real applications, **GDPR** and **CCPA** compliance is critical. Personal data should be encrypted, stored securely, and accessed only by authorized personnel.

- **Informed Consent**: Customers should consent to data collection and processing.
- **Data Minimization**: Only necessary data should be collected.
- **Data Security**: Real systems should use strong encryption and access controls.

Conclusion

This **E-commerce Shopping Cart System** database represents a realistic model of an e-commerce platform, with **composite keys**, **foreign keys**, and realistic issues like **missing data** and **duplicate entries**.

Data Generation Code:
https://github.com/sravanth-space/SQL_assignment/blob/main/ecommerce_db.ipynb

Database tables generation Code:
https://github.com/sravanth-space/SQL_assignment/blob/main/db_creation.sql

StudentId: 23001152

Student Name: Sravanth Baratam