E-Commerce Data Simulation & Schema Design

Student Id: 24081274

GitHub

1. Data Generation Methodology

1.1 Python Script Approach

A Python-based method became used to generate synthetic statistics with the assist of the Faker library. This allowed the advent of practical names, addresses, telephone numbers, and dates whilst ensuring no actual user statistics was used.

1.2 Realistic Distributions

Weighted random distributions were applied for fields like:

- `Order status` (e.g., Delivered = 60%, Cancelled = 10%)
- `Customer tier` (Bronze, Silver, Gold, Platinum)

These weights simulate real-world business patterns.

1.3 Data Quality Patterns

To simulate real-world data imperfections:

- **3–5% missing values** were intentionally added
- Duplicate rows were generated in a controlled manner
- Occasional invalid values were introduced for testing data validation

80	153	2024-01-07	Cancelled	397.96	321 Elm St	PayPal
81	49	2024-03-08	Delivered	88.3	123 Main St	Bank Transfer
82	136	2024-04-22	Confirmed	63.75		Credit Card
83	168	2024-05-10	Confirmed	378.31	654 Maple Dr	Credit Card

Figure 1: Missing values in data

1.4 Relationship Integrity

Referential integrity was strictly maintained across:

- Customers → Orders
- Products → Order Items

All foreign keys were consistent and tested.

```
FOREIGN KEY (order_id) REFERENCES orders(order_id) ON DELETE CASCADE,
FOREIGN KEY (product_id) REFERENCES products(product_id),
```

Figure 2: Foreign key query in Sql

2. Database Schema Justification

2.1 Multiple Tables Rationale

The database schema was normalized to reduce redundancy and improve efficiency.

- **Normalization:** Customer info is stored once, not repeated in orders.
- **Data Integrity:** Constraints enforce correct and meaningful values.
- **Query Performance:** Indexing benefits from separated tables.
- **Business Logic:** Clean separation between customers, products, orders, etc.

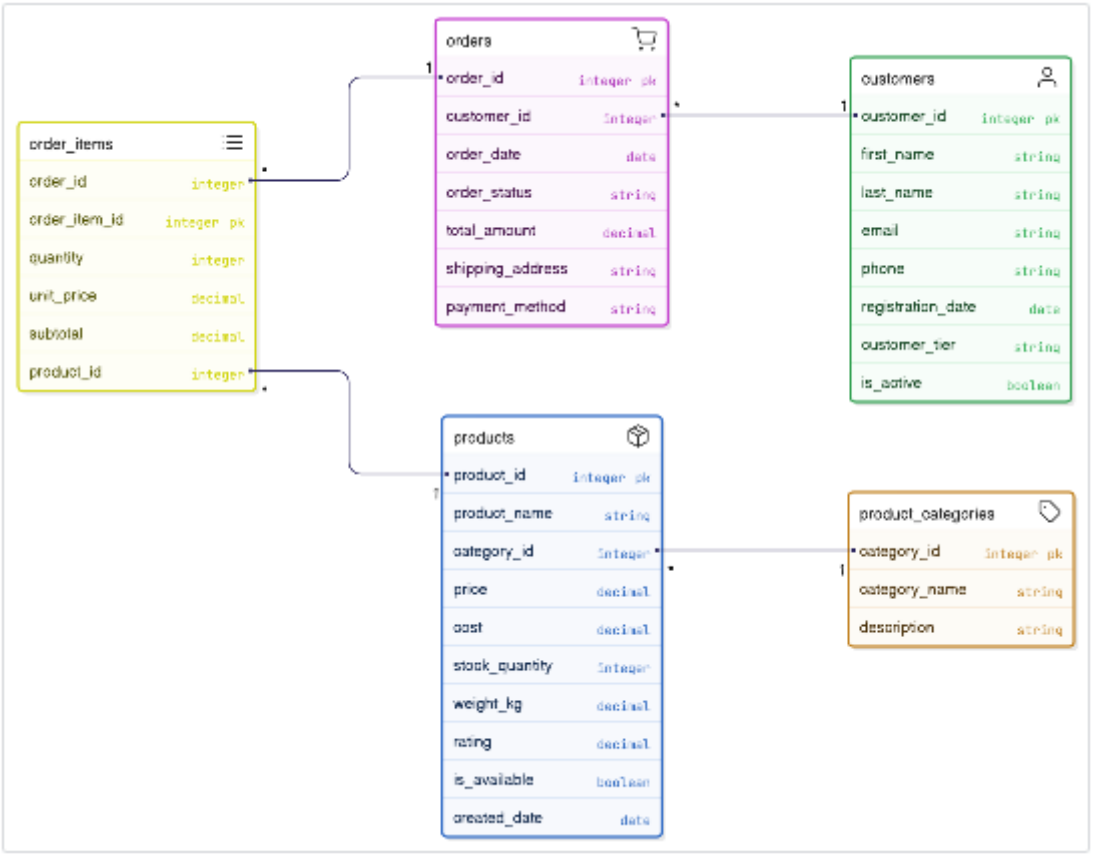

Figure 3: ER Diagram of e-commerce data

2.2 Constraints Implementation

To maintain high data quality:

- **CHECK constraints** ensure valid ranges (e.g., price ≥ 0)
- **UNIQUE constraints** enforce business keys (e.g., SKU)
- **FOREIGN KEY constraints** ensure relationship validity
- **Data type validation** prevents invalid entries

3. Data Types Representation

Different real-world data types were represented:

Nominal

- `customer_tier`
- `order_status`
- `category_name`

Ordinal

- `rating` (0–5 scale)
- `customer_tier` progression from Bronze → Platinum

Interval

- `order_date`
- `registration_date`

Ratio

- `price`, `cost`
- `quantity`
- `weight_kg`

4. Ethical and Data Privacy Considerations

4.1 Data Privacy Measures

- All data used is **synthetic**
- Emails are constructed from fake names
- Phone numbers and addresses generated via Faker
- No personally identifiable real data is included

4.3 Potential Real-world Concerns

If this were real data, issues could include:

- Purchase history privacy
- Payment information protection
- Delivery address confidentiality
- Potential misuse of behavioral analytics

5. Realism Features

The dataset included several realistic behaviors:

- Customer ordering frequency distribution
- Product pricing based on cost + markup
- Seasonal variations in order dates
- Inventory and stock-level patterns
- Missing data & occasional errors