SPRING 2025 DATA MANAGEMENT AND DATABASE DESIGN (DAMG-6210)

TEAM MEMBERS		
NAME	**NUID**	**NU EMAIL**
Dharana Kashyap	002055360	Kashyap.dh@northeastern.edu
Vidhi Patel	002389224	patel.vidhide@northeastern.edu
Udit Chaturvedi	002246428	chaturvedi.ud@northeastern.edu
Siddhanth Pai	002054575	pai.sid@northeastern.edu

Title: Predictive Inventory & Sales Analytics for E-commerce

Problem Statement

E-commerce businesses face significant challenges in inventory management, sales forecasting, and stock optimization.

- Overstocking leads to increased holding costs, while understocking results in lost sales and dissatisfied customers.
- Additionally, businesses struggle to identify top-selling products, seasonal demand patterns, return trends, and effective discount strategies in real time.

This project aims to develop a transactional database-driven **Predictive Inventory & Sales Analytics System** that allows businesses to:

- Track **real-time sales performance.**
- Predict **demand trends** using transactional queries.
- Optimize **stock levels** without relying on **AI/ML**
- Detect **seasonal demand fluctuations.**
- Improve **inventory management decisions.**

The system will be built using **Oracle SQL Developer** and hosted on **Oracle Cloud** following **OLTP (Online Transaction Processing) principles**.

Project Objective

The objective of this project is to develop a **real-time database-driven Inventory & Sales Analytics System** that enables businesses to:

1. **Identify Top-Selling and Slow-Moving Products**: Track which products sell the most and which ones are stagnant.
2. **Predict Demand for Future Sales**: Use transactional queries to estimate future demand based on past sales data.
3. **Manage Inventory Efficiently**: Prevent **stockouts** and **overstocking** by monitoring stock levels.
4. **Analyze Customer Return Trends**: Detect which products have **higher return rates** and the reasons behind them.
5. **Optimize Discount Strategies**: Identify which discount offers drive the highest **sales volume**.
6. **Detect Seasonal Trends in Sales**: Determine which products see spikes in demand during specific seasons.
7. **Track Supplier Lead Time for Stock Replenishment**: Ensure **timely restocking** based on supplier delivery patterns.
8. **Analyze Customer Purchase Frequency**: Identify **repeat customers** vs. **infrequent buyers** to improve marketing strategies.

Key Relationship Types

One-to-Many (1:M) Relationships:

- A Customer can place multiple Orders.
- A Customer can have many Addresses
- A Product appears in multiple Discounts.
- A Category can have many Products.
- A Warehouse can have multiple Products.
- A Product is stored in Inventory, with multiple stock updates.

Many-to-One (M:1) Relationships:

- Many Order Details refer to one Product.
- Many Returns refer to one Product.
- Many Customer Reviews relate to one Product.

Many to Many (N:M) Relationships:

- An Order can have multiple products, and one product can be in multiple orders.

One to One Relationship:

- One Order can have one Payment Information.
- One Order can be returned once.

- One Customer will have one Customer Behavior.
- One Supplier can have one Address.
- One Product will have only one Inventory and vice versa.

Entities and Attributes

The following entities and attributes will be used to structure the database:

1. Customers

- customer_id INTEGER (PK)f
- irst_name VARCHAR2(100)
- last_name VARCHAR2(100)
- email VARCHAR2(255)
- phone VARCHAR2(20)
- dob DATE
- gender CHAR(1)

2. Orders

- order_id INTEGER (PK)
- order_date DATE
- total_amount NUMBER(10,2)
- status VARCHAR2(20)
- customer_id INTEGER (FK)

3. Order_Items

- order_item_id INTEGER (PK)
- order_id INTEGER (FK)
- product_id INTEGER (FK)
- product_quantity INTEGER
- unit_price NUMBER(10,2)

4. Returns

- return_id INTEGER (PK)
- return_reason VARCHAR2(255)
- refund_amount NUMBER(10,2)
- status VARCHAR(20)
- order_id INTEGER (FK)

5. Payment_Information

- payment_id INTEGER (PK)
- payment_method VARCHAR2(50)
- masked_card_number VARCHAR2(16)
- mask_expiry_date DATE
- status VARCHAR(20)
- order_id INTEGER (FK)

6. Products

- product_id INTEGER (PK)
- product_name VARCHAR2(255)
- price NUMBER(10,2)
- category_id INTEGER (FK)
- warehouse_id INTEGER (FK)

7. Categories

- category_id INTEGER (PK)
- category_name VARCHAR2(100)

8. Warehouses

- warehouse_id INTEGER (PK)
- warehouse_name VARCHAR2(100)
- capacity NUMBER(10,2)
- location VARCHAR2(255)
- contact_person VARCHAR(100)
- contact_number VARCHAR(15)

9. Suppliers

- supplier_id INTEGER (PK)
- supplier_name VARCHAR2(100)
- address_id INTEGER (FK)

10. Address

- address_id INTEGER (PK)
- address_line VARCHAR2(255)
- city VARCHAR2(100)
- state VARCHAR2(100)
- zip_code VARCHAR2(20)
- country VARCHAR2(100)
- customer_id INTEGER (FK)

11. Inventory

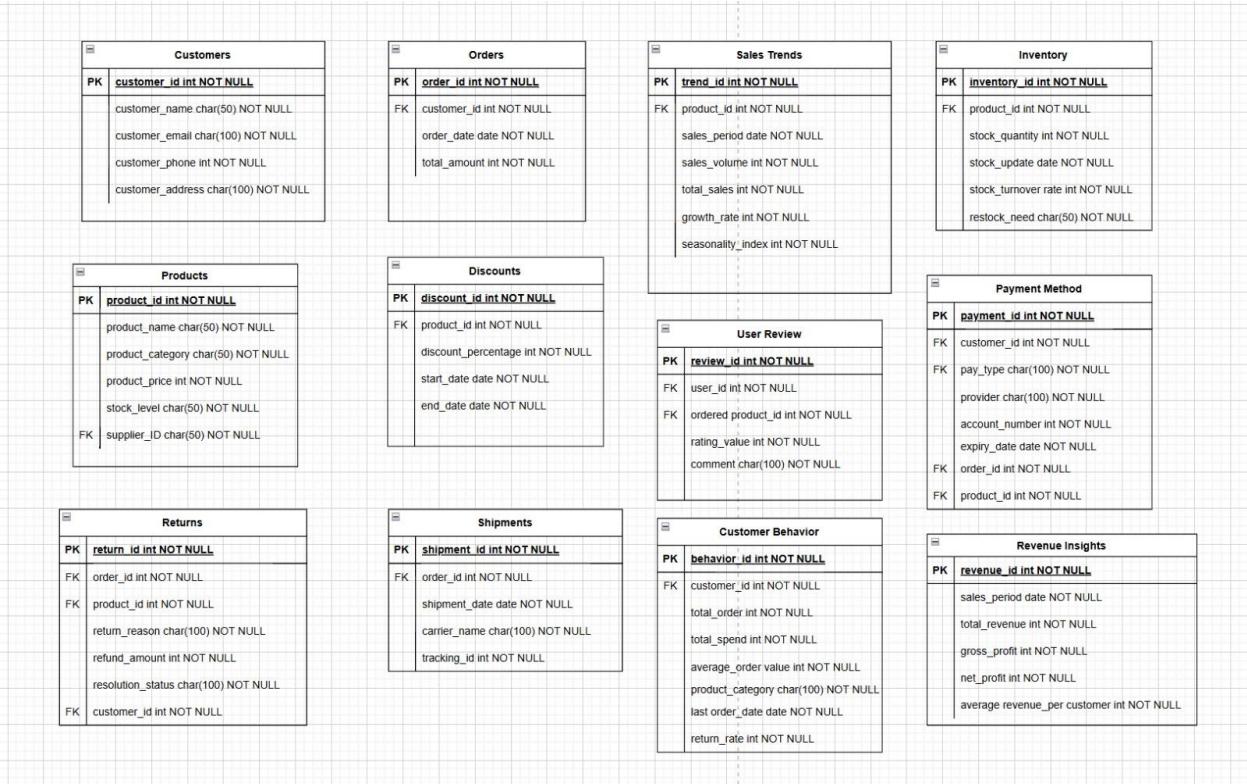
- inventory_id INTEGER (PK)
- product_id INTEGER (FK)
- quantity INTEGER
- last_restock_date DATE

12. Discounts

- discount_id INTEGER (PK)
- promo_code VARCHAR2(50)
- discount_percentage NUMBER(5,2)
- start_date DATE
- end_date DATE
- product_id INTEGER (FK)

13. Customer_Behavior

- behavior_id INTEGER (PK)
- total_orders INTEGER
- total_spend NUMBER(15,2)
- average_order_value NUMBER(15,2)
- return_rate NUMBER(5,2)
- customer_id INTEGER (FK)

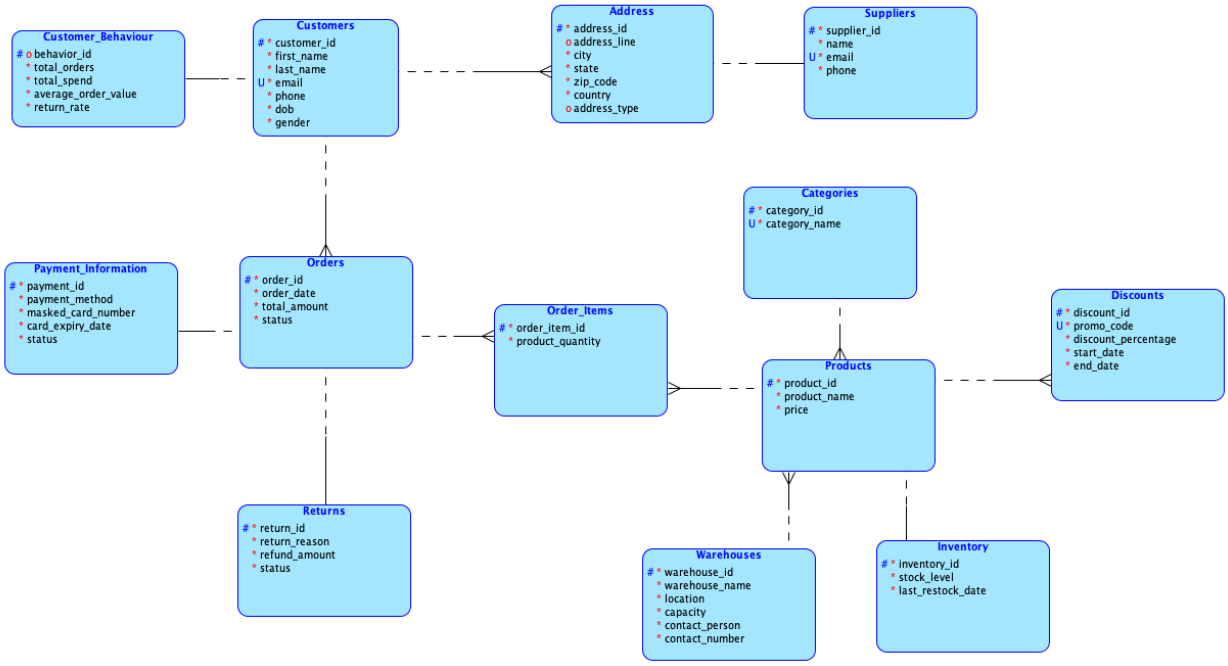


User Roles Table

Role	Responsibilities	Permissions
Admin Role	Full access to all database objects and operations.	- Create, alter, drop tables, views, functions, stored procedures, triggers, and indexes. - Manage users and roles. - Oversee backups and restore.

Data Analyst	Run reports on sales, inventory, customer behavior, and more.	- Read access to all relevant tables (e.g., Customers, Orders, Inventory). - Execute reporting functions/procedures.
Customer Support	Manage customer queries, process returns, and update order statuses.	- Read/write access to Orders, Returns, and Customer Reviews tables. - Update customer information in the Customers table.
Sales/Inventory Management	Manage inventory, process orders, and handle discounts/promotions.	- Read/write access to Products, Inventory, Sales Trends. - Manage prices, discounts, and products.
Customer Role	Interact with the system, place orders, view product information, and submit reviews.	- Read access to their own data (e.g., profile, orders). - Write access to reviews and Order table (for their own orders).
Read-Only Role	View data but cannot modify any information.	- Read access to all tables, views, and data. - No update or delete permissions.

LOGICAL MODEL



RELATIONAL MODEL