FINAL PROJECT REPORT

Inventory Control Management

BUAN 6320

Group 5 (Team GenSEEK.Co)

Group Members

1. Aditya Deoke- aad230001
2. Haider Sultan- hxs230021
3. Sujay Singh Ravi-sxr230005
4. Amy Le-ahl170000
5. Prakruthi Pole-pxp220089
6. Pavan Kumar Ramisetty- pxr230004

Team Member’s Contribution –

- Database Designing & ERR Diagram – Haider Sultan & Aditya Deoke
- .SQL file creation – Haider Sultan
- Set 1 table creation and Data Entry - Haider Sultan, Aditya Deoke, Amy Le
- Set 2 table creation and Data Entry - Haider Sultan, Aditya Deoke, Pavan Ramisetty
- Set 3 table creation and Data Entry - Haider Sultan, Aditya Deoke, Prakruthi Pole
- 7 Complex Queries- Haider Sultan & Aditya Deoke
- 4 Complex Queries- Sujay Singh, Pavan Ramisetty, Prakruthi Pole, Amy Le
- Query creation for Stored Procedure - Sujay Singh
- Query creation for View – Aditya Deoke
- Functions-Haider Sultan
- Triggers-Aditya Deoke
- Video Script-Amy Le & Prakruthi Pole
- Project Write Up – Aditya Deoke, Pavan Ramisetty, Prakruthi Pole, Amy Le
- Report Consolidation-Aditya Deoke, Sujay Singh

Project Charter

Business Scenario and Project Plan: Inventory Control Management

Introduction:

In the fast-paced world of retail and e-commerce, efficient inventory management is crucial for businesses to thrive. Our Inventory Management System (IMS) is a comprehensive solution designed to streamline the processes of tracking, managing, and analyzing inventory data. The system is tailored to meet the needs of businesses dealing with diverse products, suppliers, customers, and warehouses.

Key Features:

1. **Product Management:**
    - Create and manage product categories.
    - Track detailed information about each product, including SKU, name, quantity, unit price, category, and supplier.
    - Monitor stock levels and receive alerts for low inventory.
2. **Supplier and Customer Management:**
    - Maintain a database of suppliers with contact details.
    - Manage customer information, including names, email, phone, and address.
    - Easily handle customer returns with detailed reasons for return.
3. **Order Management:**
    - Create and process customer orders seamlessly.
    - Generate reports on order details, including order date, items, quantity, and total amount.
    - Track order fulfillment through the integrated shipment tracking system.
4. **Warehouse Management:**
    - Maintain a list of warehouses with location details.
    - Monitor warehouse inventory levels and transfers between warehouses.
    - Assign employees to specific warehouses for efficient management.
5. **Employee Management:**
    - Keep track of employees with details such as name, email, phone, hire date, and job title.
    - Assign employees to specific warehouses for efficient inventory handling.
6. **Shipment Tracking:**
    - Track the status of shipments in real-time.
    - Generate reports on shipments, including tracking numbers and delivery status.
7. **Payment Methods:**
    - Manage and track various payment methods used for transactions.
    - Integrate with the order system to handle payments seamlessly.
8. **Sales Transactions:**
    - Record and track sales transactions, including total amounts and payment methods.
    - Generate reports on sales performance over time.
9. **Product Reviews:**
    - Allow customers to leave reviews for products.

- Display average ratings and customer feedback for each product.

10. **Product Discounts and Images:**
    - Apply discounts to products based on specified criteria.
    - Store and display product images for a better visual representation.

Reports and Queries:

1. **Inventory Reports:**
    - Stock levels for each product.
    - Inventory turnover reports.
    - Low stock and out-of-stock alerts.
2. **Order and Sales Reports:**
    - Order history for each customer.
    - Sales performance reports.
    - Returns and refunds analysis.
3. **Supplier and Customer Reports:**
    - Supplier performance reports.
    - Customer purchase history.
4. **Warehouse Reports:**
    - Warehouse inventory levels and transfers.
    - Employee assignments and productivity.

Potential UI Features:

1. **Dashboard:**
    - Overview of key metrics such as total sales, inventory status, and order fulfillment.
2. **Product Catalog:**
    - User-friendly display of products with images and details.
3. **Order Processing:**
    - Easy-to-use interface for creating and processing orders.
4. **Reports Section:**
    - Access to various reports for informed decision-making.
5. **User Management:**
    - Secure login and role-based access control.
6. **Notifications:**
    - Automated alerts for low inventory, order fulfillment, and shipment status.
7. **ChatBot Assistance:**
    - Inventory Chatbot Assistance: Including a chatbot that can respond to questions about inventory and help users manage their inventory efficiently.

PROJECT SCOPE:

1. **Scope Inclusions:**

- Inventory Adjustment Queries: Develop SQL queries to handle adjustments to inventory levels, taking into account factors like sales, returns, and new stock additions.
- Customer Database: Implement SQL tables and queries for storing and managing customer information, including order history and feedback.
- Feedback and Rating System: Create SQL structures to store customer feedback and ratings for each inventory item, along with queries to calculate and display average ratings.
- Dynamic Pricing: Integrate SQL logic to dynamically adjust product prices based on inventory levels, demand, and promotional events.
- Inventory Updates: Ensure SQL-based triggers and procedures for real-time updates to inventory levels, enabling timely notifications and accurate stock tracking.
- Order Management: Develop SQL functionalities to manage orders, track order status, and handle returns, ensuring synchronization with the inventory system.

2. **Scope Exclusions:**

- Financial Transactions: Financial transactions, such as payment processing, are excluded from the scope and will not be handled within the SQL-based inventory management system.
- Advanced Analytics: Advanced analytics beyond basic reporting are not included in the initial scope but may be considered for future enhancements.
- External System Integrations: Integration with external systems beyond the e-commerce platform is not within the project's current scope.

3. **Timeline and Resources:**

- Define a timeline for the development, testing, and deployment of SQL-based functionalities, considering the resources available, including developers and database administrators.
- Regular collaboration with stakeholders, including inventory managers and customer service representatives, for feedback and refinement of SQL queries.
- By focusing on these specific SQL-based objectives and scope, the inventory management system will effectively handle crucial aspects such as inventory tracking, customer management, feedback, pricing, and order processing.

DRAWBACKS:

- Inventory control management has its drawbacks, such as the risk of having too little stock, which can lead to increased storage expenses. Additionally, there is a possibility of products becoming obsolete making it challenging to handle and maintain inventory data. It also requires monitoring and adjustments adding to the complexities involved.

Relational Data Model

One-to-Many Relationships:
-CUSTOMERS_T to ORDERS_T: One customer can place multiple orders.
-PRODUCT_CATEGORIES_T to INVENTORY_ITEMS_T: One category can have multiple inventory items.
-SUPPLIERS_T to INVENTORY_ITEMS_T: One supplier can supply multiple items.
-SHIPMENT_TRACKING_T to INVENTORY_ITEMS_T: One tracking entry for each inventory item.
-RETURN_REASONS_T to ORDERS_T: One return reason for multiple orders.
-PRODUCT_REVIEWS_T to INVENTORY_ITEMS_T: One review for each inventory item.
-EMPLOYEES_T to EMPLOYEE_ASSIGNMENTS_T: One employee can have multiple assignments.
-PAYMENT_METHODS_LOOKUP to ORDERS_T: One payment method for multiple orders.
-WAREHOUSE to WAREHOUSE_INVENTORY_T: One warehouse can have multiple inventory items.
-ORDERS_T to ORDER_ITEMS_T: One order can have multiple order items.

Many-to-One Relationships:
-**ORDERS_T to CUSTOMERS_T:** Multiple orders can belong to one customer.
-INVENTORY_ITEMS_T to PRODUCT_CATEGORIES_T: Multiple items can belong to one category.
-INVENTORY_ITEMS_T to SUPPLIERS_T: Multiple items can be supplied by one supplier.
-SHIPMENT_TRACKING_T to INVENTORY_ITEMS_T: Multiple items can be associated with one tracking entry.

-ORDERS_T to RETURN_REASONS_T: Multiple orders can have the same return reason.
-INVENTORY_ITEMS_T to PRODUCT_REVIEWS_T: Multiple reviews for one inventory item.
-EMPLOYEE_ASSIGNMENTS_T to EMPLOYEES_T: Multiple assignments for one employee.
-ORDERS_T to PAYMENT_METHODS_LOOKUP: Multiple orders can use the same payment method.
-WAREHOUSE_INVENTORY_T to WAREHOUSE: Multiple items in a warehouse can belong to the same warehouse.
-ORDER_ITEMS_T to ORDERS_T: Multiple items can belong to the same order.

One-to-One Relationships:

-CUSTOMERS_T to US_STATES_LOOKUP: Each customer is associated with one state.

-EMPLOYEES_T to EMPLOYEE_ASSIGNMENTS_T: One assignment for each employee.

Assumptions/Notes About Data Entities and Relationships

1)Product Categories:
-Assumes that each product category is uniquely identified by a category_id_PK.
-Categories are defined by their names (category_name).
2)Suppliers:
-Each supplier is uniquely identified by a supplier_id_PK.
-Suppliers are characterized by their names (supplier_name), contact names, phone numbers, and email addresses.
3)US States Lookup:
-Assumes that US states are uniquely identified by a state_code_PK (2-character state code) and have corresponding state names.
4)Customers:
-Each customer is identified by a unique customer_id_PK.
-Customer information includes first name (F_name), last name (L_name), email, phone, address, state (referenced from US_STATES_LOOKUP), city, and pin code.
5)Inventory Items:
-Each inventory item has a unique item_id_PK.
-Items are identified by SKU (SKU_PK), name, quantity, unit price, category_id_FK (references PRODUCT_CATEGORIES_T), and supplier_id_FK (references SUPPLIERS_T).
6)Delivery Status Lookup:
-A lookup table for shipment delivery status with unique status_id and corresponding status names.
7)Shipment Tracking:
-Each shipment is tracked by a unique tracking_id_PK.
-Tracks items (item_id_FK) with shipment dates, tracking numbers, and delivery status (status_id_FK, references DELIVERY_STATUS_LOOKUP).
8)Warehouse:
-Each warehouse has a unique warehouse_id_PK.
-Warehouses are identified by names and locations.
9)Warehouse Inventory:
-Tracks the quantity of items in each warehouse (warehouse_inventory_id_PK).
-References the Warehouse table and the Inventory Items table.
10)Return Reasons:
-Provides reasons for returns with a unique reason_id_PK and a description.
11)Product Reviews:
-Reviews are identified by a unique review_id_PK.
-Associates with specific items (item_id_FK), customers (customer_id_FK), ratings, and review text.
12)Payment Methods Lookup:
-A lookup table for payment methods with unique method_id_PK and payment names.

13)Orders:
-Each order has a unique order_id_PK.
-Contains information about customers (customer_id_FK), items (item_id_FK), quantity, total amount, order date, payment method (payment_method_id_FK), shipment tracking (tracking_id_FK), return reasons (return_reason_id_FK), and reviews (review_id_FK).
14)Sales Transactions:
-Records sales transactions with a unique transaction_id_PK.
-Associates with specific orders, transaction dates, total amounts, and payment methods.
15)Employees:
-Each employee is uniquely identified by an employee_id_PK.
-Employee information includes name, email, phone, hire date, supervisor (reports_to), and job title.
16)Employee Assignments:
-Tracks employee assignments (assignment_id_PK) to specific warehouses (warehouse_id_FK) with start and end dates.
17)Product Discounts:
-Discounts are associated with specific items (item_id_FK) and have unique discount_id_PK, percentage, start date, and end date.
18)Product Images:
-Each product image has a unique image_id_PK and is associated with specific items (item_id_FK).
19)Order Items:
-Details of items included in each order, identified by order_item_id_PK.
-References specific orders (order_id_FK), items (item_id_FK), quantity, unit price, and total amount.

Scenarios covered for the database

1. **Inventory Addition:**

- Scenario: When new products are added to the inventory, the database captures the details, including product name, description, initial stock quantity, unit price, and supplier information.

2. **Inventory Sales:**

- Scenario: Upon a successful sale, the database updates the stock levels for the sold items, records the transaction details (order number, customer details), and triggers relevant notifications.

3. **Inventory Returns:**

- Scenario: In the case of product returns, the database adjusts the stock levels accordingly, updates the return status in the Orders entity, and maintains a history of return transactions

4. **Customer Feedback:**

- Scenario: When a customer provides feedback and a rating for a product, the database records this information, associating it with the specific inventory item and customer. The average rating for each product is calculated dynamically.

5. **Dynamic Pricing and Discounts:**

- Scenario: The database dynamically adjusts product prices based on factors like inventory levels, demand, and ongoing promotions. It stores information about regular prices, active discounts, and any promotional events.

6. **Order Processing:**

- Scenario: Upon receiving an order, the database updates the stock levels for the items in the order, records the transaction details, and calculates the total cost. It also updates the order status and triggers notifications for order processing.

7. **Low Stock Notification**:

- Scenario: When an inventory item reaches a predefined low stock level, the database generates a notification to alert relevant stakeholders about the need for restocking.

8. **Supplier Relationship Management:**

- Scenario: The database manages supplier information, including contact details and a list of supplied products. It facilitates communication with suppliers, tracks product supplies, and supports negotiations.

9. **User Authentication and Access Control:**

- Scenario: The database ensures secure user authentication and access control, allowing authorized users to perform specific actions based on their roles and permissions.

10. **Historical Data Recording:**

- Scenario: The database maintains historical records for inventory items, pricing, and sales transactions. This historical data supports trend analysis, demand forecasting, and strategic decision-making.

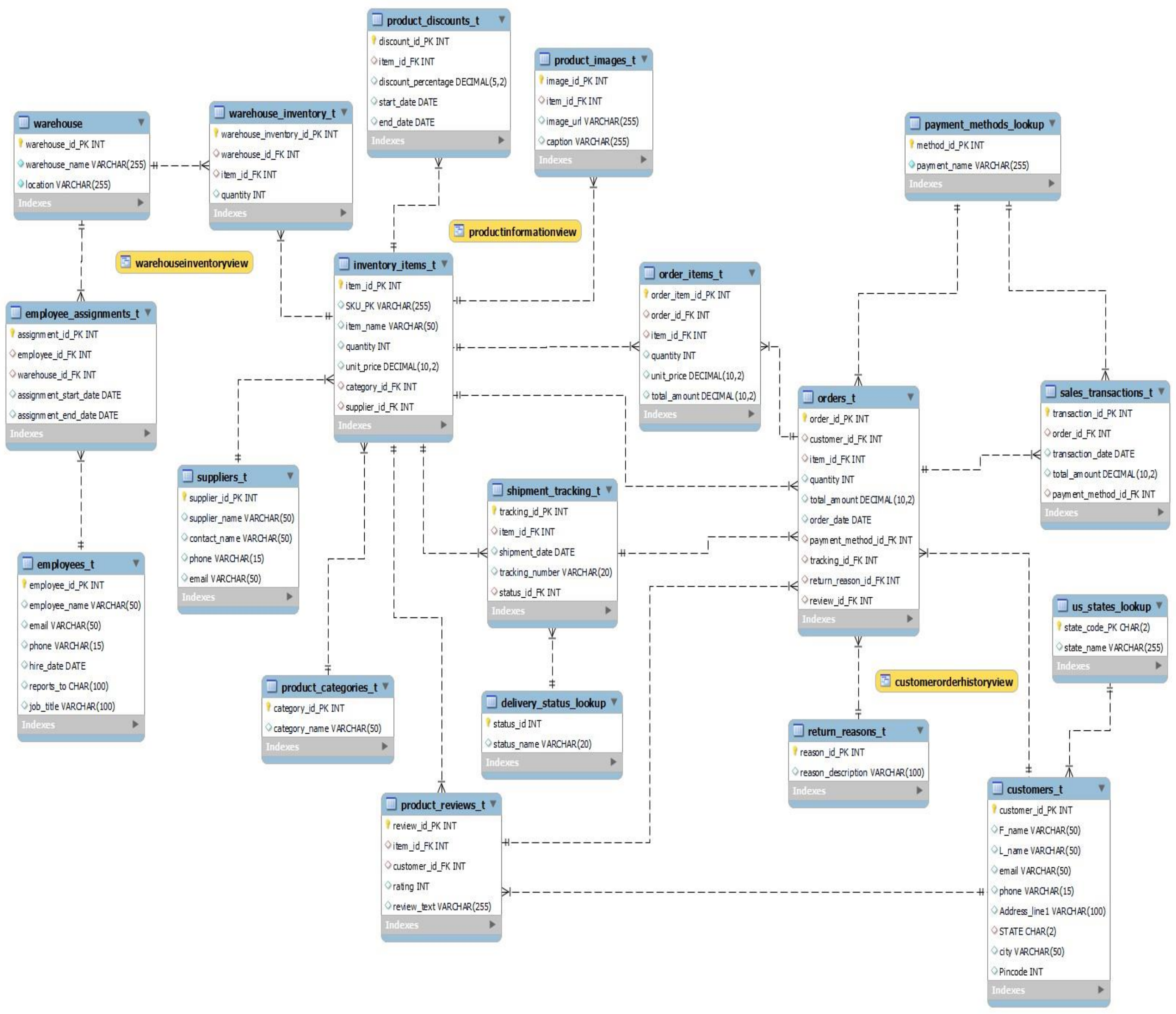
11. **Integration with E-commerce Platform:**

- Scenario: The database integrates seamlessly with the e-commerce platform, ensuring consistent and accurate data flow between systems. It supports real-time updates of product availability, order fulfillment, and customer transactions.

12. **Discount Application:**

- Scenario: The database applies discounts to specific products or customer segments based on predefined rules. It calculates the discounted prices dynamically and reflects them in the pricing information.

ERR Diagram


product_discounts_t
discount_id_PK INT
item_id_FK INT
discount_percentage DECIMAL(5,2)
start_date DATE
end_date DATE
Indexes
product_images_t
image_id_PK INT
item_id_FK INT
image_url VARCHAR(255)
caption VARCHAR(255)
Indexes
warehouse
warehouse_id_PK INT
warehouse_name VARCHAR(255)
location VARCHAR(255)
Indexes
warehouse_inventory_t
warehouse_inventory_id_PK INT
warehouse_id_FK INT
item_id_FK INT
quantity INT
Indexes
payment_methods_lookup
method_id_PK INT
payment_name VARCHAR(255)
Indexes
productinformationview
warehouseinventoryview
inventory_items_t
item_id_PK INT
SKU_PK VARCHAR(255)
item_name VARCHAR(50)
quantity INT
unit_price DECIMAL(10,2)
category_id_FK INT
supplier_id_FK INT
Indexes
order_items_t
order_item_id_PK INT
order_id_FK INT
item_id_FK INT
quantity INT
unit_price DECIMAL(10,2)
total_amount DECIMAL(10,2)
Indexes
employee_assignments_t
assignment_id_PK INT
employee_id_FK INT
warehouse_id_FK INT
assignment_start_date DATE
assignment_end_date DATE
Indexes
orders_t
order_id_PK INT
customer_id_FK INT
item_id_FK INT
quantity INT
total_amount DECIMAL(10,2)
order_date DATE
payment_method_id_FK INT
tracking_id_FK INT
return_reason_id_FK INT
review_id_FK INT
Indexes
sales_transactions_t
transaction_id_PK INT
order_id_FK INT
transaction_date DATE
total_amount DECIMAL(10,2)
payment_method_id_FK INT
Indexes
suppliers_t
supplier_id_PK INT
supplier_name VARCHAR(50)
contact_name VARCHAR(50)
phone VARCHAR(15)
email VARCHAR(50)
Indexes
shipment_tracking_t
tracking_id_PK INT
item_id_FK INT
shipment_date DATE
tracking_number VARCHAR(20)
status_id_FK INT
Indexes
employees_t
employee_id_PK INT
employee_name VARCHAR(50)
email VARCHAR(50)
phone VARCHAR(15)
hire_date DATE
reports_to CHAR(100)
job_title VARCHAR(100)
Indexes
us_states_lookup
state_code_PK CHAR(2)
state_name VARCHAR(255)
Indexes
customerorderhistoryview
product_categories_t
category_id_PK INT
category_name VARCHAR(50)
Indexes
delivery_status_lookup
status_id INT
status_name VARCHAR(20)
Indexes
return_reasons_t
reason_id_PK INT
reason_description VARCHAR(100)
Indexes
customers_t
customer_id_PK INT
F_name VARCHAR(50)
L_name VARCHAR(50)
email VARCHAR(50)
phone VARCHAR(15)
Address_line1 VARCHAR(100)
STATE CHAR(2)
city VARCHAR(50)
Pincode INT
Indexes
product_reviews_t
review_id_PK INT
item_id_FK INT
customer_id_FK INT
rating INT
review_text VARCHAR(255)
Indexes

Design of the Database

Sr no.	TABLE NAME	PRIMARY KEY	FOREIGN KEY	NON-KEY ATTRIBUTES	# OF ROWS IN TABLE
1	Categories Table	category_id_PK		category_name	20
2	SUPPLIERS_T	SUPPLIERS_T		supplier_name	20
				contact_name	
				phone	
				email	
3	US_STATES_LOOKUP	state_code_PK		state_name	20
4	CUSTOMERS_T	customer_id_PK	STATE	F_name	20
				L_name	
				email	
				phone	
				Address_line1	
				city	
				Pincode	
5	INVENTORY_ITEMS_T	item_id_PK	category_id_FK	SKU_PK	20
			supplier_id_FK	item_name	
				quantity	
				unit_price	
6	DELIVERY_STATUS_LOOKUP	status_id		status_name	20
7	SHIPMENT_TRACKING_T	tracking_id_PK	item_id_FK	shipment_date	20
			status_id_FK	tracking_number	
8	Warehouse	warehouse_id_PK		warehouse_name	8
				location	
9	WAREHOUSE_INVENTORY_T	warehouse_inventory_id_PK	warehouse_id_FK	quantity	20
			item_id_FK		
10	RETURN_REASONS_T	reason_id_PK		reason_description	20

11	PRODUCT_REVIEWS_T	review_id_PK	item_id_FK	rating	20
			customer_id_FK	review_text	
12	PAYMENT_METHODS_LOOKUP	method_id_PK		payment_name	4
13	ORDERS_T	order_id_PK	customer_id_FK	quantity	20
			item_id_FK	total_amount	
			payment_method_id_FK	order_date	
			tracking_id_FK		
			return_reason_id_FK		
			review_id_FK		
14	SALES_TRANSACTIONS_T	transaction_id_PK	order_id_FK	transaction_date	20
			payment_method_id_FK	total_amount	
15	EMPLOYEES_T	employee_id_PK		employee_name	20
				email	
				phone	
				hire_date	
				reports_to	
				job_title	
16	EMPLOYEE_ASSIGNMENTS_T	assignment_id_PK	employee_id_FK	assignment_start_date	20
			warehouse_id_FK	assignment_end_date	
17	PRODUCT_DISCOUNTS_T	discount_id_PK	item_id_FK	discount_percentage	20
				start_date	
				end_date	
18	PRODUCT_IMAGES_T	image_id_PK	item_id_FK	image_url	20
				caption	
19	ORDER_ITEMS_T	order_item_id_PK	order_id_FK	quantity	20
			item_id_FK	unit_price	
				total_amount	

3NF Database

-Each column in the tables stores single, indivisible values. Primary keys are appropriately defined for each table using the PRIMARY KEY constraint, ensuring unique identification of records and preventing duplicate entries.

-The relationships between tables seem to avoid transitive dependencies. Non-prime attributes do not depend on other non-prime attributes within the same table.

-Foreign keys are employed to establish relationships between tables. For example, in the INVENTORY_ITEMS_T table, foreign keys (category_id_FK and supplier_id_FK) reference the primary keys in other tables (PRODUCT_CATEGORIES_T and SUPPLIERS_T), maintaining referential integrity across tables.

-The structure of the tables suggests normalization, minimizing data redundancy and dependency by organizing data into related tables. For instance, the INVENTORY_ITEMS_T table has separate foreign keys for category and supplier, avoiding the need to repeat category and supplier information for each item.

-Lookup tables, such as DELIVERY_STATUS_LOOKUP, US_STATES_LOOKUP, and PAYMENT_METHODS_LOOKUP, are utilized to maintain consistent values across the database and prevent the need for duplicating information.

-Tables and columns follow a consistent naming convention, aiding in the readability and maintenance of the database structure. This consistency makes it easier for developers and administrators to understand the relationships and purpose of each element.

-In summary, the provided database structure exhibits key characteristics of a well-normalized and organized database design, adhering to best practices for data integrity and efficiency.

Complex Queries

1. Retrieve a list of product categories along with the average rating of products in each category and display the categories in descending order of the average rating.

2. Calculate the total purchase made by each customer, including their name and email, and display the results in descending order of total purchase amount.

3.Query to retrieve customer shipment status along with tracking number and customer details.

4.Calculate the total amount of sales made using each payment method, and display the payment method name and the total sales amount in descending order of sales.

5.Write a query to display the list of employees as an organization tree and show who reports to whom.(Self join)

6.Query to display real-time discounts on e-commerce products by fetching SKU, item name, and discount details for items with active promotions on the current date.

7. Obtain the total quantity of any item (Ex-'Black Paint Roller') across multiple warehouses to efficiently manage inventory and ensure accurate stock levels for the specified item.

8. Query to retrieve details of employee assignments, including assignment ID, employee name, warehouse name, location, and assignment dates, facilitating efficient tracking and management of workforce distribution across warehouses in an organization

9. Obtain a comprehensive view of customer reviews and product details, including customer names, product names, ratings, reviews, and associated return reasons, enhancing customer feedback analysis and product quality assessment in an e-commerce system.

10.Query to retrieve product names along with their average ratings, aiding quick assessment of overall customer satisfaction for each product in an inventory.
(Sub-query)
11.Query to retrieve customer names, order details, product names, and associated return reasons, facilitating efficient tracking and analysis of product returns in an e-commerce system for customer service and inventory management purposes.

Views & Stored Procedures created on Database

Stored Procedures

1) -- Retrieve a list of employees based on their job title.

```
DELIMITER $$
CREATE PROCEDURE get_employees_by_title (IN job_title VARCHAR(100))
BEGIN
  SELECT e.employee_id_pk, e.employee_name, e.job_title
  FROM employees_t e
  WHERE e.job_title = job_title;
END $$
DELIMITER ;
```

→CALL get_employees_by_title('CEO');

2) -- Generate a report showing the monthly sales trend over the past year.

```
DELIMITER $$
CREATE PROCEDURE GenerateMonthlySalesTrendReport()
BEGIN
   SELECT MONTH(order_date) AS month, YEAR(order_date) AS year, SUM(total_amount) AS
monthly_sales
   FROM ORDERS_T
   WHERE order_date >= DATE_SUB(NOW(), INTERVAL 1 YEAR)
   GROUP BY YEAR(order_date), MONTH(order_date)
   ORDER BY year, month;
END $$
DELIMITER ;
```

→CALL GenerateMonthlySalesTrendReport();

3) -- Identify products that have consistently low sales compared to their inventory quantity.

```
DELIMITER $$
CREATE PROCEDURE IdentifyLowPerformingProducts()
BEGIN
   SELECT i.item_id_PK, i.item_name, SUM(oi.quantity) AS total_sold, i.quantity AS current_inventory,
         (SUM(oi.quantity) / i.quantity) AS sales_inventory_ratio
   FROM INVENTORY_ITEMS_T i
   LEFT JOIN ORDER_ITEMS_T oi ON i.item_id_PK = oi.item_id_FK
   GROUP BY i.item_id_PK, i.item_name, i.quantity
   HAVING sales_inventory_ratio < 0.04; -- Identify products with less than 4% sales to inventory ratio
END $$
DELIMITER ;
```

→ CALL IdentifyLowPerformingProducts();

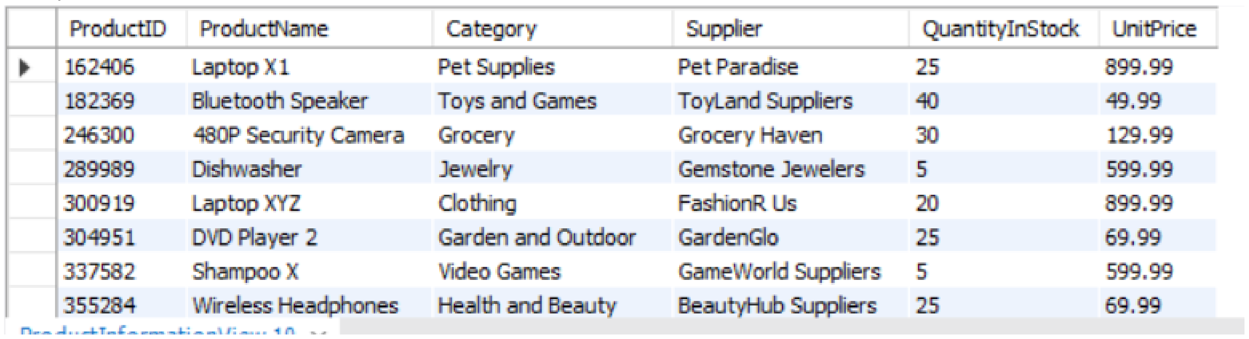
View

1) -- Displays information about products, including their name, category, supplier, current quantity in inventory, and unit price.

```
CREATE VIEW ProductInformationView AS
SELECT
   i.item_id_PK AS ProductID,
   i.item_name AS ProductName,
   pc.category_name AS Category,
   s.supplier_name AS Supplier,
   i.quantity AS QuantityInStock,
   i.unit_price AS UnitPrice
FROM
   INVENTORY_ITEMS_T i
   JOIN PRODUCT_CATEGORIES_T pc ON i.category_id_FK = pc.category_id_PK
   JOIN SUPPLIERS_T s ON i.supplier_id_FK = s.supplier_id_PK;
```

→SELECT * FROM ProductInformationView;

Sample Result-



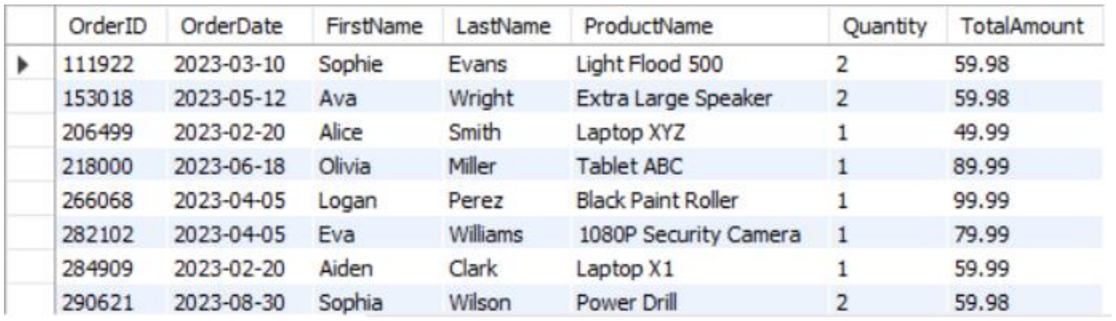
ProductID	ProductName	Category	Supplier	QuantityInStock	UnitPrice
162406	Laptop X1	Pet Supplies	Pet Paradise	25	899.99
182369	Bluetooth Speaker	Toys and Games	ToyLand Suppliers	40	49.99
246300	480P Security Camera	Grocery	Grocery Haven	30	129.99
289989	Dishwasher	Jewelry	Gemstone Jewelers	5	599.99
300919	Laptop XYZ	Clothing	FashionR Us	20	899.99
304951	DVD Player 2	Garden and Outdoor	GardenGlo	25	69.99
337582	Shampoo X	Video Games	GameWorld Suppliers	5	599.99
355284	Wireless Headphones	Health and Beauty	BeautyHub Suppliers	25	69.99

2) -- Shows the order history for each customer, including order date, product details, quantity, and total amount.

```
 CREATE VIEW CustomerOrderHistoryView AS
SELECT
    o.order_id_PK AS OrderID,
    o.order_date AS OrderDate,
    c.F_name AS FirstName,
    c.L_name AS LastName,
    i.item_name AS ProductName,
    oi.quantity AS Quantity,
    oi.total_amount AS TotalAmount
FROM
    ORDERS_T o
    JOIN CUSTOMERS_T c ON o.customer_id_FK = c.customer_id_PK
    JOIN ORDER_ITEMS_T oi ON o.order_id_PK = oi.order_id_FK
    JOIN INVENTORY_ITEMS_T i ON oi.item_id_FK = i.item_id_PK;
```

→SELECT * FROM CustomerOrderHistoryView;

Sample Result-



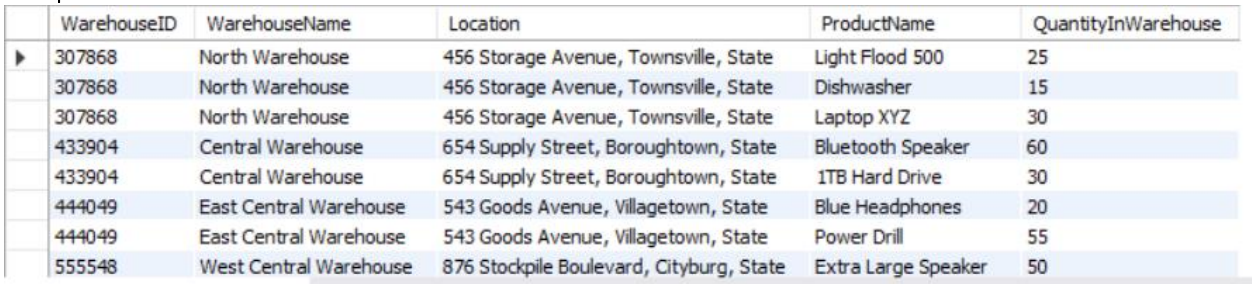
OrderID	OrderDate	FirstName	LastName	ProductName	Quantity	TotalAmount
111922	2023-03-10	Sophie	Evans	Light Flood 500	2	59.98
153018	2023-05-12	Ava	Wright	Extra Large Speaker	2	59.98
206499	2023-02-20	Alice	Smith	Laptop XYZ	1	49.99
218000	2023-06-18	Olivia	Miller	Tablet ABC	1	89.99
266068	2023-04-05	Logan	Perez	Black Paint Roller	1	99.99
282102	2023-04-05	Eva	Williams	1080P Security Camera	1	79.99
284909	2023-02-20	Aiden	Clark	Laptop X1	1	59.99
290621	2023-08-30	Sophia	Wilson	Power Drill	2	59.98

3) -- Provides information about the inventory in each warehouse, including the warehouse name, location, and the quantity of each product.

```
    CREATE VIEW WarehouseInventoryView AS
SELECT
    w.warehouse_id_PK AS WarehouseID,
    w.warehouse_name AS WarehouseName,
    w.location AS Location,
    i.item_name AS ProductName,
    wi.quantity AS QuantityInWarehouse
FROM
    WAREHOUSE_INVENTORY_T wi
    JOIN Warehouse w ON wi.warehouse_id_FK = w.warehouse_id_PK
    JOIN INVENTORY_ITEMS_T i ON wi.item_id_FK = i.item_id_PK;
```

→SELECT * FROM WarehouseInventoryView;

Sample Result-



WarehouseID	WarehouseName	Location	ProductName	QuantityInWarehouse
307868	North Warehouse	456 Storage Avenue, Townsville, State	Light Flood 500	25
307868	North Warehouse	456 Storage Avenue, Townsville, State	Dishwasher	15
307868	North Warehouse	456 Storage Avenue, Townsville, State	Laptop XYZ	30
433904	Central Warehouse	654 Supply Street, Boroughtown, State	Bluetooth Speaker	60
433904	Central Warehouse	654 Supply Street, Boroughtown, State	1TB Hard Drive	30
444049	East Central Warehouse	543 Goods Avenue, Villagetown, State	Blue Headphones	20
444049	East Central Warehouse	543 Goods Avenue, Villagetown, State	Power Drill	55
555548	West Central Warehouse	876 Stockpile Boulevard, Cityburg, State	Extra Large Speaker	50

FUNCTIONS

-- 1. Dynamic Pricing Strategy- This query enables real-time adjustment of product prices based on market demand, competitor prices, and trends, ensuring competitive pricing in the e-commerce landscape.

-- 2. Customer lifetime value- This Function offers valuable insights into the customer's long-term financial contribution to the business.

-- 3. Employee Assignment To Warehouse- This Function checks for any conflicting assignments and, if none exist, successfully assigns the employee to the warehouse for the specified period. The result indicates the status of the assignment, whether it was successful or if a conflicting assignment exists.

TRIGGERS

1)The trigger "AutoRestockTrigger" fires before updating the "INVENTORY_ITEMS_T" table. If the new quantity falls below a set threshold (10), it automatically increases the quantity by 20.

2)The trigger "before_delete_customers" prevents the deletion of a customer in the "CUSTOMERS_T" table if there are pending orders in the "ORDERS_T" table associated with that customer. It counts the orders for the customer being deleted and raises an error if any orders are found, stopping the deletion process.

3)The trigger "update_product_discount_expiry" is activated before inserting a new record into the "PRODUCT_DISCOUNTS_T" table. Its purpose is to automatically update the status of discounts by setting the "status" column to 'Expired' if the "end_date" is earlier than the current date.

MEETING LOG

S.No	Date	Time	Minutes of meeting	Absentees
1	09/18/2023	9:00 PM	Initial discussion(Project Scope)	Nil
2	09/21/2023	9:00 PM	Brainstorming tables and their attributes	Nil
3	09/25/2023	9:00 PM	Work split up for the initial write-up	Nil
4	10/02/2023	3:00 PM	Tables assigned to each member for creation	Nil
5	10/20/2023	2:00 PM	Check table creation and insertion of values into the tables	Nil
6	10/16/2023	9:00 PM	Discussion on complex queries and assign joins/ subqueries to be done.	Nil
7	10/27/2023	4:30 PM	Discussion on stored procedures, functions, and triggers and assigning work to each member	Nil
8	11/03/2023	3:30 PM	Brainstorming ideas for the video	Nil
9	11/15/2023	7:30 PM	Discussing the Video Script	
10	12/02/2023	3:30 PM	Shooting the video	Nil
11	12/03/2023	2:00 PM	Error checking in SQL code and discussion on presentation	Nil
12	12/09/2023	9:00 PM	Discussion on final report and work	Nil

Conclusion:

Our Inventory Management System stands as a formidable tool, poised to significantly elevate the efficiency and accuracy of inventory operations for businesses. With robust features and advanced reporting capabilities, this system is positioned to be an indispensable asset in the realm of inventory management. The incorporation of a user-friendly interface further amplifies its usability, establishing it as a comprehensive solution suitable for businesses of varying sizes.

Fuelled by a potent SQL-based database backend, accompanied by its powerful querying features and report generation capabilities, our system equips businesses with the essential tools for making informed decisions and ensuring a seamless inventory management experience. This initiative not only holds the potential to enhance operational efficiency but also to evolve into a commercially viable online retail service through the incorporation of a thoughtfully designed user interface.