Inventory Management System for a Retail Store

Part One: Database Design

Inventory Management System

Introduction

Inventory also known as stock is perhaps the most important asset held by retail stores across the world. According to Munyaka and Yadavali (2022), this fact is attributed to the fact that inventory represent half of the total capital investment in organizations, firms, and retail stores. As Wei et al. (2023) put it, inventory keeps on moving in and out of the retail stores as it represents the bulk of the activities carried out in the retail stores. This makes the management of inventory a crucial part in the supply chain of retail stores as it has an impact on how retail stores can effectively meet product demand and convert sales into profits. In order to generate profits, retail stores must strive for customer satisfaction. This means that the inventory must be restocked on time, delivery done in time, and orders must be processed at a reasonable rate. According to Alam et al. (2023), the only sure way retail stores can achieve customer satisfaction is by instituting an Inventory Management System (IMS).

According to Atnafu et al. (2023), an IMS is a framework that retail stores can use to control, manage, and organize its inventory by continuously updating stock levels and estimating future demand. Therefore, an IMS is responsible for ensuring a retail store has an adequate supply of goods at al times. Consequently, the IMS can aid retail stores avoid retaining excessive stock which has an impact the working capital, improve the flow of inventory, meet customer demand, and reduce the cost of warehousing by stocking products that will meet the demand. The IMS achieves this by allowing retailers to connect various aspects of their trades such as customers, orders, and inventory and update them more efficiently. For this to work, the IMS must be connected to a database that has various inbuild functions to serve a libraries objective.

Objectives of IMS

Wei et al. (2023) maintains that a normal IMS objective is to provide better control over inventory, orders, delivery, and excellent customer service management. To streamline our IMS to these objectives, we identified the following goals;

Efficient inventory control

Our IMS will eliminate the need for manually tacking and updating incoming and outgoing stock. In addition, the system's real-time tracking capabilities will reduce the likelihood of understocking, through automated restocking request to suppliers

Improved operational efficiency

By eradicating the need for manual updates, the employees can focus on other aspects such as packaging, customer service, and delivery.

Providing efficient customer service

The IMS will feature a customer service feature that will request for feedback and allow customers to provide feedback on the quality of products and the services rendered.

Facilitate Informed decision making

The system can provide variable information and insights on how inventory moves and consequently aid in optimizing inventory levels.

Error reduction

The system can reduce on human-related errors in inventory management as it minimizes the need for manual registers.

Scope and Limitations of the IMS

Our IMS system is optimized for all retail stores. This means that the size of the retail store does not matter. However, for retails stores with multiple branches, reconfiguration of the IMS is needed. As such, this system can be adapted by retail stores from diverse industries including liquor, grocery, clothes, and cosmetic stores and supermarkets. However, since this is an IMS, it is only restricted to

inventory management and can therefore not track items such as payment of salaries to employees. However, it can process and handle order payments and payments made to suppliers.

User Requirements

The system will meet the following requirements for retail stores;

1. ***Inventory tracking*** – The system will track the quantity of stock per product available in the store to avoid overstocking and understocking. In addition, it will track where each item is located in the shelf to ease retrieval of the item in case of online purchases.
2. ***User alerts*** – Since the system is tracking the quantity of items per product, the system will automatically send a warning when the quantity falls below a specific threshold. Also, online customers wishing to buy diminished stock will be notified before placing their orders.
3. ***Report on sales*** – The system can generate sale reports for the day, week, month, year, or any other time period on prompt. This will make it easy to draw insights on sales, and the revenues associated with individual product categories or items.
4. ***Security and privacy*** – As a database system, the system will feature security features that will prevent accessibility of information by unauthorized persons.
5. ***Support data analytics to grow sales*** – The system is designed in such a way that it can provide data and information vital to conducted customer segmentations with a view of improving sales.
6. ***Tracking returns*** – After an order is fulfilled and product removed from the inventory, it is necessary to track if the purchased product has been returned and why it has been returned to gain insight on whether it should be returned to the shelf or not.
7. ***Integration with Point-of Sales (PoS) gadgets capabilities*** – The system aims at minimizing the need for manual registers. Therefore, the system will allow integration of PoS instruments such as bar-code readers to allow seamless updating of records in the stores. This will minimize the INT of errors made thus improving operational efficiency.

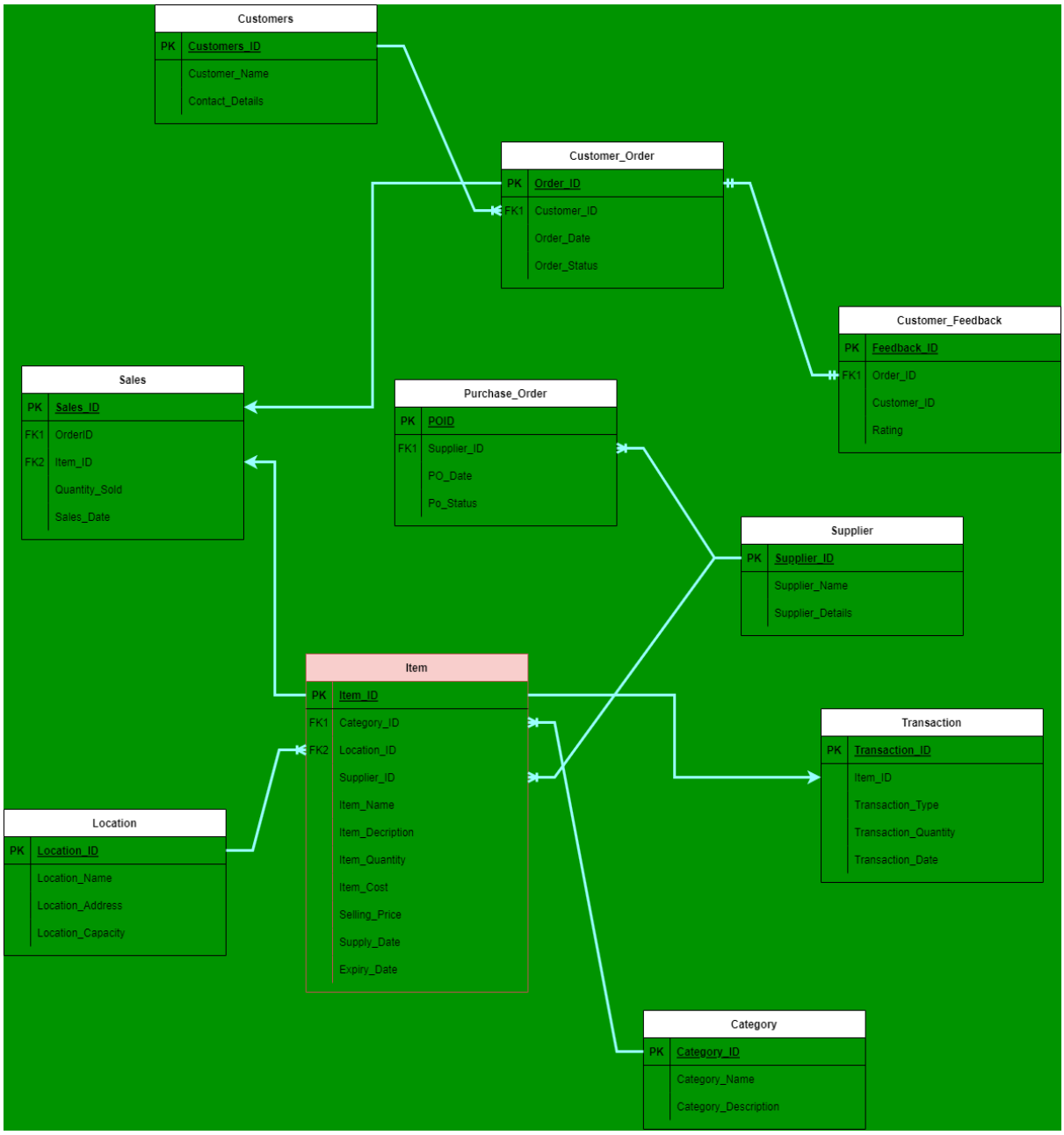
While the user requirements are unique for each store, the requirements above are configured in such a way that the following business rules are not violated.

Business Rules

Business rules refers to the set guidelines and constraints that govern how a business defines its business processes and manages its data. For the IMS, the business rules are defined as follows. The first business rule is all items must have a unique identifier, a name, description, quantity available, cost, and the selling price. Secondly, inventory must be supplied by a supplier, and therefore, they must be ordered. Consequently, supplier(s) must be identifiable by a unique identifier, a name, a contact address, and a location.

Thirdly, each transaction must be associated with a particular item or product. Fourthly, inventory can be stored in different locations or different warehouses. Finally, customers orders are only created when a customer requests items, therefore customer orders are then used to update the sales records. After defining the business rules, it is imperative to visualize how different entities of the IMS interact with each other. The best way to visualize this is to develop ERD diagrams.

ERD



Relationship

Item to other items

Many items can belong to one category. In addition, an item can be supplied by many suppliers, and a supply can supply many unique items. An item can have multiple transactions, but a transaction involves only one item. Finally, an item can be in different location and each location can have multiple items.

Purchases and Supplier

We can purchase an item many times from different suppliers. However, each unique purchase can only be associated with one supplier. This makes the relationship between purchases and suppliers a one-to-many kind of relationship.

Customer Orders and Sales

A customer can order an item (recorded as sales) many times. However, each item sold item will be recorded in a unique order. Therefore, customer orders and sales have a one-to-many kind of a relationship.

Data Dictionary

A data dictionary provides detailed description of entities, data types, relationship with other entities, and other constraints. The IMS utilizes the following data dictionary;

Entity	Column Name	Data Type	Description
Item	Item_ID	Auto-INT	Unique identifier of an item
	Item_Name	TEXT	Item Name
	Item_Description	TEXT	A description of the item
	Item_Quantity	INT	Items remaining on shelf
	Item_Cost	INT	Cost of buying 1 unit of the item
	Selling Price	INT	Cost of selling 1 unit of the item
	Supply_Date	Date	Date the item was supplied
	Expiry_Date	Date	Expiry date of promotions if applicable
	Category_ID	Auto-INT	Foreign key from category table
Supplier	Supplier_ID	Auto-INT	Unique identifier of the suppplier
	Supplier_Name	TEXT	Supplier name
	Contact_Details	TEXT	Supplier email
Purchase Order	POID	Auto-INT	Unique identifier of inventory purchases
	PO_ Date	Date	Date inventory is purchased
	PO_Status	TEXT	Check if inventory is delivered
	Supplier_ID	Auto-INT	Foreign key referencing supplier
Transaction	Transaction_ID	Auto-INT	Unique identify of transaction
	Transaction_Type	TEXT	
	Transaction_Quantity	INT	Quantity of transaction
	Transaction_Date	Date	Date of transaction
	Item_ID	Auto-INT	Foreign key referencing item
Location	Location_ID	Auto-INT	Unique identifier of shelf/ location
	Location_Name	TEXT	Name designated to the location
	Location_ Address	TEXT	Exact location address
	Location_Capacity	INT	Max quantity that can be stored
Category	Category_ID	Auto-INT	Unique identifier of item category
	Category_Name	TEXT	Assigned category name

	Category_Description	TEXT	A short description of the category
Customer Order	Order_ID	Auto-INT	Unique identifier of each order
	Order_Date	Date	Date the order was made
	Order_Status	TEXT	
	Customer_ID	Auto_INT	Foreign key referencing customer
Sales	Sales_ID	Auto-INT	Unique identifier of each sale
	Quantity_Sold	INT	Quantity sold
	Sales_Date	Date	Date the item was sold
	Total_Amount	INT	Price at which the item was sold
	Item_ID	Auto-INT	Foreign key referencing item
	Order_ID	Auto-INT	Foreign key referencing order

Part Two: Demonstrating Effectiveness of the database

Entity Generation

```
CREATE TABLE Supplier (
    Supplier_ID INT AUTO_INCREMENT PRIMARY KEY,
    Supplier_Name TEXT,
    Contact_Details TEXT
);

CREATE TABLE Purchase_Order (
    POID INT AUTO_INCREMENT PRIMARY KEY,
    PO_Date DATE,
    PO_Status TEXT,
    Supplier_ID INT,
    FOREIGN KEY (Supplier_ID) REFERENCES Supplier(Supplier_ID)
);

CREATE TABLE Category (
    Category_ID INT AUTO_INCREMENT PRIMARY KEY,
    Category_Name TEXT,
    Category_Description TEXT
);

CREATE TABLE Item (
    Item_ID INT AUTO_INCREMENT PRIMARY KEY,
    Item_Name TEXT,
    Item_Description TEXT,
    Item_Quantity INT,
    Item_Cost INT,
    Selling_Price INT,
    Supply_Date DATE,
    Expiry_Date DATE,
    Category_ID INT,
    Location_ID INT,
    FOREIGN KEY (Category_ID) REFERENCES Category(Category_ID)
    FOREIGN KEY (Location_ID) REFERENCES Location(Location_ID)
);

CREATE TABLE Transaction (
    Transaction_ID INT AUTO_INCREMENT PRIMARY KEY,
    Transaction_Type TEXT,
    Transaction_Quantity INT,
    Transaction_Date DATE,
    Item_ID INT,
    FOREIGN KEY (Item_ID) REFERENCES Item(Item_ID)
);

CREATE TABLE Location (
    Location_ID INT AUTO_INCREMENT PRIMARY KEY,
    Location_Name TEXT,
    Location_Address TEXT,
    Location_Capacity INT
);

CREATE TABLE Customer_Order (
    Order_ID INT AUTO_INCREMENT PRIMARY KEY,
    Order_Date DATE,
    Order_Status TEXT,
    Customer_ID INT,
```

```
    FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID)
);
CREATE TABLE Sales (
    Sales_ID INT AUTO_INCREMENT PRIMARY KEY,
    Quantity_Sold INT,
    Sales_Date DATE,
    Item_ID INT,
    Order_ID INT,
    FOREIGN KEY (Item_ID) REFERENCES Item(Item_ID),
    FOREIGN KEY (Order_ID) REFERENCES Customer_Order(Order_ID)
);

-- Create Customer table
CREATE TABLE Customer (
    Customer_ID INT AUTO_INCREMENT PRIMARY KEY,
    Customer_Name TEXT,
    Contact_Details TEXT
);
CREATE TABLE Customer_Feedback (
    Feedback_ID INT AUTO_INCREMENT PRIMARY KEY,
    Customer_ID INT,
    Order_ID INT,
    Rating INT,
    FOREIGN KEY (Order_ID) REFERENCES Customer_Order(Order_ID)
);
```

Entity Generation

Customer table

```
INSERT INTO Customer (Customer_Name, Contact_Details)
VALUES
    ('John D', 'johndk@mail.com'),
    ('Jane Shiny', 'janeshiny@mail.com'),
    ('Jame Johnson', 'jimmyjohnson@yahoo.com'),
    ('Bob Brown', 'bobbrown@mail.com'),
    ('Eva George', 'evageorge@gmail.com'),
    ('Michael Power', 'michaelp@mail.com'),
    ('Laureen Lee', 'lee@edu.com'),
    ('Zipporah M', 'zipporahs@oligarcg.com'),
    ('Olivia Kate', 'kateso@example.com'),
    ('Michael J', 'jacksonm@main.com'),
    ('Sarah Wairimu', 'wairishsarah@example.com'),
    ('Ryan Kamau', 'kamaryan@gmail.com'),
    ('Emma Wambui', 'wambu@mail.com'),
    ('Matthew Kate', 'mattkate@example.com'),
    ('Faith One', 'faith23@gmail.com'),
    ('James Kamau', 'kamaujames@example.com'),
    ('Chloe Decker', 'decker@nypd.com'),
    ('David Njogo', 'njogodavid@mail.com'),
    ('Sophie Pine', 'pineshopie@yahoo.com'),
    ('Samuel Sure', 'sure@microsoft.com'),
    ('Grace Wanjiru', 'wanjirugrace@example.com'),
    ('Logan Collins', 'logancollins@example.com'),
    ('Hannah Hill', 'hannahhill@example.com'),
    ('Ethan Ward', 'ethanward@example.com'),
    ('Zoe Johnson', 'zoejohn@main.com');
```

Customer_ID	Customer_Name	Contact_Details
1	John D	johndk@mail.com
2	Jane Shiny	janeshiny@mail.com
3	Jame Johnson	jimmyjohnson@yahoo.com
4	Bob Brown	bobbrown@mail.com
5	Eva George	evageorge@gmail.com
6	Michael Power	michaelp@mail.com
7	Laureen Lee	lee@edu.com
8	Zipporah M	zipporahs@oligarcg.com
9	Olivia Kate	kateso@example.com
10	Michael J	jacksonm@main.com
11	Sarah Wairimu	wairishsarah@example.com
12	Ryan Kamau	kamaryan@gmail.com
13	Emma Wambui	wambu@mail.com
14	Matthew Kate	mattkate@example.com
15	Faith One	faith23@gmail.com
16	James Kamau	kamaujames@example.com
17	Chloe Decker	decker@nypd.com
18	David Njogo	njogodavid@mail.com
19	Sophie Pine	pineshonie@yahoo.com

Figure 1: Customer table showing customer details including name and email addresses

```
-- Sample data for Supplier table
INSERT INTO Supplier (Supplier_Name, Contact_Details)
VALUES
    ('Tech Supplier Co.', 'techsupplier@example.com'),
    ('Mobile Tech Inc.', 'mobiletech@example.com'),
    ('SmartConnection Ltd.', 'smartphone@smartconnection.com'),
    ('Peripheral Devices Inc.', 'peripheraldevices@peri.com'),
    ('Top 10 Office Equipment Suppliers', 'officesupp@gmail.com'),
    ('Gadget World Ltd.', 'gadgetworld@mail.com');
```

Supplier_ID	Supplier_Name	Contact_Details
1	Tech Supplier Co.	techsupplier@example.com
2	Mobile Tech Inc.	mobiletech@example.com
3	SmartConnection Ltd.	smartphone@smartconnection.com
4	Peripheral Devices Inc.	peripheraldevices@peri.com
5	Top 10 Office Equipment Suppliers	officesupp@gmail.com
6	Gadget World Ltd.	gadgetworld@mail.com

Figure 2: A list of suppliers supplying gadgets to the retail store

Purchase_Order table

```
INSERT INTO Purchase_Order (PO_Date, PO_Status, Supplier_ID)
VALUES
    ('2023-01-15', 'Delivered', 1),
    ('2023-02-20', 'In Transit', 2),
    ('2023-03-25', 'Pending', 3),
    ('2023-04-10', 'Delivered', 4),
    ('2023-05-05', 'Delivered', 5),
    ('2023-06-30', 'In Transit', 6),
    ('2023-07-12', 'Pending', 2),
```

```
    ('2023-08-18', 'Delivered', 3),
    ('2023-09-22', 'In Transit', 4),
    ('2023-10-28', 'Pending', 5),
    ('2023-11-01', 'Delivered', 6),
    ('2023-11-02', 'In Transit', 1),
    ('2023-11-03', 'Pending', 3),
    ('2023-11-04', 'Delivered', 2),
    ('2023-11-05', 'Delivered', 6),
    ('2023-11-06', 'In Transit', 5),
    ('2023-11-07', 'Pending', 1),
    ('2023-11-08', 'Delivered', 2),
    ('2023-11-09', 'In Transit', 1),
    ('2023-11-10', 'Pending', 1),
    ('2023-11-11', 'Delivered', 2),
    ('2023-11-12', 'In Transit', 3),
    ('2023-11-13', 'Pending', 6),
    ('2023-11-14', 'Delivered', 5),
    ('2023-11-15', 'In Transit', 2);
```

POID	PO_Date	PO_Status	Supplier_ID
1	2023-01-15	Delivered	1
2	2023-02-20	In Transit	2
3	2023-03-25	Pending	3
4	2023-04-10	Delivered	4
5	2023-05-05	Delivered	5
6	2023-06-30	In Transit	6
7	2023-07-12	Pending	2
8	2023-08-18	Delivered	3
9	2023-09-22	In Transit	4
10	2023-10-28	Pending	5
11	2023-11-01	Delivered	6
12	2023-11-02	In Transit	1
13	2023-11-03	Pending	3
14	2023-11-04	Delivered	2
15	2023-11-05	Delivered	6
16	2023-11-06	In Transit	5
17	2023-11-07	Pending	1
18	2023-11-08	Delivered	2
19	2023-11-09	In Transit	1
20	2023-11-10	Pending	1
21	2023-11-11	Delivered	2

Figure 3: Purchases made from different suppliers over the last few months

Category table

```
INSERT INTO Category (Category_Name, Category_Description)
VALUES
    ('Electronics', 'Electronic devices especially computers, laptops,
sound systems, etc.'),
    ('Office Equipment', 'Furniture and office supplies'),
    ('Computer Parts', 'Parts of a computer such as internal harddisk, RAM,
BUS cables etc.'),
```

```
    ('Smartphones', 'Mobile phones and accessories'),
    ('Peripherals', 'Computer peripherals such as mouse,keyboard, webcam,

etc.');
```

Category_ID	Category_Name	Category_Description
1	Electronics	Electronic devices especially computers, laptops...
2	Office Equipment	Furniture and office supplies
3	Computer Parts	Parts of a computer such as internal harddisk, R...
4	Smartphones	Mobile phones and accessories
5	Peripherals	Computer peripherals such as mouse,keyboard,...

Figure 4: A list of the various categories available in the store and their short description

Item table

```
INSERT INTO Item (Item_Name, Item_Description, Item_Quantity, Item_Cost,
Selling_Price, Supply_Date, Expiry_Date, Category_ID, Location_ID)
VALUES
    ('Laptop', 'HP Pavilion', 50, 800, 1200, '2023-01-01', NULL, 1,1),
    ('Smartphone', 'Techno Spark', 100, 500, 800, '2023-02-01', NULL, 1,1),
    ('Printer', 'HP Color laser printer 35660S', 30, 300, 500, '2023-03-
01', NULL, 2,2),
    ('Desk', 'Wooden office desk', 20, 200, 350, '2023-04-01', NULL, 3,3),
    ('Chair', 'Ergonomic chair S32', 40, 150, 250, '2023-05-01', NULL, 3,
3),
    ('Headphones', 'Samsung Noise-cancelling headphones', 60, 100, 180,
'2023-06-01', NULL, 4, 4),
    ('Keyboard', 'LG Mechanical gaming keyboard', 25, 80, 120, '2023-07-
01', NULL, 4, 4),
    ('Mouse', 'Dell Wireless mouse', 35, 40, 70, '2023-08-01', NULL, 4),
    ('Monitor', 'HP 27-inch LED monitor', 15, 250, 400, '2023-09-01', NULL,
2, 2),
    ('Tablet', 'Blackberry Tablet device', 45, 300, 500, '2023-10-01',
NULL, 1, 1),
    ('Speaker', 'Bluetooth speaker', 25, 50, 90, '2023-11-01', NULL, 4, 4),
    ('External Hard Drive', '1TB HDD', 30, 70, 100, '2023-11-02', NULL, 5,
5),
    ('Wireless Router', 'High-speed router', 15, 80, 120, '2023-11-03',
NULL, 4, 4),
    ('Camera', 'DSLR camera', 20, 300, 500, '2023-11-04', NULL, 1, 1),
    ('Printer Ink', 'Black ink cartridge', 40, 20, 40, '2023-11-05', NULL,
2, 2),
    ('USB Flash Drive', '32GB USB stick', 50, 15, 25, '2023-11-06', NULL,
5, 5),
    ('Office Chair', 'Ergonomic office chair', 10, 120, 200, '2023-11-07',
NULL, 3, 3),
    ('Table Lamp', 'Desk lamp', 35, 25, 45, '2023-11-08', NULL, 4, 4),
    ('Power Bank', '10000mAh power bank', 30, 30, 50, '2023-11-09', NULL,
4, 4),
    ('HDMI Cable', 'High-speed HDMI cable', 60, 10, 20, '2023-11-10', NULL,
5, 5),
    ('Smartwatch', 'Fitness smartwatch', 25, 150, 250, '2023-11-11', NULL,
1, 1),
    ('Scanner', 'Flatbed scanner', 15, 200, 300, '2023-11-12', NULL, 2, 2),
```

```
    ('Desk Organizer', 'Office desk organizer', 20, 20, 35, '2023-11-13',
NULL, 3, 3),
    ('WiFi Extender', 'Signal booster', 10, 50, 80, '2023-11-14', NULL, 4,
4),
    ('Laptop Bag', '15-inch laptop bag', 40, 40, 60, '2023-11-15', NULL, 5,

5);
```

Item_ID	Item_Name	Item_Description	Item_Quantity	Item_Cost	Selling_Price	Supply_Date	Expiry_Date	Category_ID	Location_ID
26	Laptop	HP Pavilion	50	800	1200	2023-01-01	NULL	1	1
27	Smartphone	Techno Spark	100	500	800	2023-02-01	NULL	1	1
28	Printer	HP Color laser printer 35660S	30	300	500	2023-03-01	NULL	2	2
29	Desk	Wooden office desk	20	200	350	2023-04-01	NULL	3	3
30	Chair	Ergonomic chair S32	40	150	250	2023-05-01	NULL	3	3
31	Headphones	Samsung Noise-cancelling headphones	60	100	180	2023-06-01	NULL	4	4
32	Keyboard	LG Mechanical gaming keyboard	25	80	120	2023-07-01	NULL	4	4
33	Mouse	Dell Wireless mouse	35	40	70	2023-08-01	NULL	4	4
34	Monitor	HP 27-inch LED monitor	15	250	400	2023-09-01	NULL	2	2
35	Tablet	Blackberry Tablet device	45	300	500	2023-10-01	NULL	1	1
36	Speaker	Bluetooth speaker	25	50	90	2023-11-01	NULL	4	4
37	External Ha...	1TB HDD	30	70	100	2023-11-02	NULL	5	5
38	Wireless Ro...	High-speed router	15	80	120	2023-11-03	NULL	4	4
39	Camera	DSLR camera	20	300	500	2023-11-04	NULL	1	1
40	Printer Ink	Black ink cartridge	40	20	40	2023-11-05	NULL	2	2
41	USB Flash D...	32GB USB stick	50	15	25	2023-11-06	NULL	5	5
42	Office Chair	Ergonomic office chair	10	120	200	2023-11-07	NULL	3	3
43	Table Lamp	Desk lamp	35	25	45	2023-11-08	NULL	4	4
44	Power Bank	10000mAh power bank	30	30	50	2023-11-09	NULL	4	4
45	HDMI Cable	High-speed HDMI cable	60	10	20	2023-11-10	NULL	5	5
46	Smartwatch	Fitness smartwatch	25	150	250	2023-11-11	NULL	1	1

Figure 5: A table showing all the items available for sale, their quantity, prices, and expiry dates if applicable

Transaction table

```
INSERT INTO Transaction (Transaction_Type, Transaction_Quantity,
Transaction_Date, Item_ID)
VALUES
    ('Purchase', 100, '2023-01-20', 26),
    ('Sale', 50, '2023-02-25', 26),
    ('Purchase', 20, '2023-03-30', 28),
    ('Purchase', 10, '2023-04-15', 30),
    ('Purchase', 80, '2023-05-10', 31),
    ('returns', 5, '2023-01-20', 31),
    ('Purchase', 200, '2023-02-25', 32),
    ('Sale', 50, '2023-03-30', 32),
    ('Purchase', 40, '2023-04-15', 34),
    ('Sale', 80, '2023-05-10', 35),
    ('Purchase', 120, '2023-06-25', 36),
    ('Purchase', 30, '2023-07-05', 37),
    ('Return', 10, '2023-08-20', 38),
    ('Purchase', 20, '2023-09-25', 39),
    ('Purchase', 90, '2023-10-30', 40),
    ('Purchase', 20, '2023-11-05', 41),
    ('Sale', 5, '2023-11-07', 41),
    ('Purchase', 30, '2023-11-09', 28),
    ('Sale', 10, '2023-11-11', 28),
    ('Sale', 15, '2023-11-15', 37),
    ('Return', 8, '2023-11-17', 40),
    ('Purchase', 55, '2023-11-19', 45),
    ('Sale', 20, '2023-11-21', 45),
```

```
    ('Purchase', 70, '2023-11-25', 48),
    ('Sale', 25, '2023-11-27', 48),
    ('Sale', 30, '2023-11-29', 45),
    ('Return', 15, '2023-12-04', 49),
    ('Purchase', 90, '2023-12-03', 49);
```

Transaction_ID	Transaction_Type	Transaction_Quantity	Transaction_Date	Item_ID
1	Purchase	100	2023-01-20	26
2	Sale	50	2023-02-25	26
3	Purchase	20	2023-03-30	28
4	Purchase	10	2023-04-15	30
5	Purchase	80	2023-05-10	31
6	returns	5	2023-01-20	31
7	Purchase	200	2023-02-25	32
8	Sale	50	2023-03-30	32
9	Purchase	40	2023-04-15	34
10	Sale	80	2023-05-10	35
11	Purchase	120	2023-06-25	36
12	Purchase	30	2023-07-05	37
13	Return	10	2023-08-20	38
14	Purchase	20	2023-09-25	39
15	Purchase	90	2023-10-30	40
16	Purchase	20	2023-11-05	41
17	Sale	5	2023-11-07	41

Figure 6: Table showing the various transactions involving inventory movement

Location table

```
INSERT INTO Location (Location_Name, Location_Address, Location_Capacity)
VALUES
    ('Aisle A', 'Entire Row', 800),
    ('Aisle B', 'Shelves 1, 2, 4', 1200),
    ('Aisle C', 'Shelve 5, 6', 800),
    ('Aisle D', 'Shelf 2', 450),
    ('Aisle E', 'Shelf 4', 400);
```

Location_ID	Location_Name	Location_Address	Location_Capacity
1	Aile A	Entire Row	800
2	Aile B	Shelves 1, 2, 4	1200
3	Aile C	Shelve 5, 6	800
4	Aile D	Shelf 2	450
5	Aile E	Shelf 4	400

Figure 7: Shows the locations where a product can be placed

Customer_Order table

```
INSERT INTO Customer_Order (Order_Date, Order_Status, Customer_ID)
VALUES
    ('2023-01-05', 'Pending', 1),
    ('2023-02-10', 'Shipped', 2),
```

```
('2023-03-15', 'Delivered', 3),
('2023-04-20', 'Pending', 4),
('2023-05-25', 'Shipped', 5),
('2023-06-30', 'Delivered', 6),
('2023-07-05', 'Pending', 7),
('2023-08-10', 'Shipped', 6),
('2023-09-15', 'Delivered', 5),
('2023-10-20', 'Pending', 10),
('2023-11-05', 'Pending', 11),
('2023-11-10', 'Shipped', 12),
('2023-11-15', 'Delivered', 8),
('2023-11-20', 'Pending', 14),
('2023-11-25', 'Shipped', 9),
('2023-11-30', 'Delivered', 16),
('2023-12-05', 'Pending', 10),
('2023-12-9', 'Shipped', 18),
('2023-12-03', 'Delivered', 19),
('2023-12-08', 'Pending', 11),
('2023-01-12', 'Shipped', 21),
('2023-11-20', 'Delivered', 22),
('2023-11-25', 'Pending', 13),
('2023-10-13', 'Shipped', 24),
('2024-01-15', 'Delivered', 1);
```

Order_ID	Order_Date	Order_Status	Customer_ID
1	2023-01-05	Pending	1
2	2023-02-10	Shipped	2
3	2023-03-15	Delivered	3
4	2023-04-20	Pending	4
5	2023-05-25	Shipped	5
6	2023-06-30	Delivered	6
7	2023-07-05	Pending	7
8	2023-08-10	Shipped	6
9	2023-09-15	Delivered	5
10	2023-10-20	Pending	10
11	2023-11-05	Pending	11
12	2023-11-10	Shipped	12
13	2023-11-15	Delivered	8
14	2023-11-20	Pending	14
15	2023-11-25	Shipped	9
16	2023-11-30	Delivered	16
17	2023-12-05	Pending	10
18	2023-12-09	Shipped	18
19	2023-12-03	Delivered	19
20	2023-12-08	Pending	11
21	2023-01-12	Shipped	21

Figure 8: Shows the orders made by customers and whether the orders have been fulfilled.

Sales table

```
INSERT INTO Sales (Quantity_Sold, Sales_Date, Item_ID, Order_ID)
VALUES
```

```
    (5, '2023-01-10', 26, 1),
    (8, '2023-02-15', 26, 2),
    (12, '2023-03-20', 27, 3),
    (3, '2023-04-25', 29, 4),
    (10, '2023-05-30', 26, 5),
    (15, '2023-06-05', 30, 6),
    (7, '2023-07-10', 32, 7),
    (9, '2023-08-15', 35, 8),
    (6, '2023-09-20', 36, 9),
    (11, '2023-10-25', 32, 10),
    (4, '2023-11-06', 40, 11),
    (7, '2023-11-08', 42, 12),
    (9, '2023-11-10', 44, 13),
    (3, '2023-11-12', 43, 14),
    (10, '2023-11-14', 45, 15),
    (12, '2023-11-16', 28, 16),
    (6, '2023-11-18', 32, 17),
    (8, '2023-11-20', 33, 18),
    (11, '2023-11-22', 44, 19),
    (14, '2023-11-24', 45, 20),
    (5, '2023-11-26', 26, 21),
    (13, '2023-11-28', 27, 22),
    (7, '2023-11-30', 28, 23),
    (9, '2023-12-02', 29, 24),
    (16, '2023-12-04', 40, 25);
```

Sales_ID	Quantity_Sold	Sales_Date	Item_ID	Order_ID
178	5	2023-01-10	26	1
179	8	2023-02-15	26	2
180	12	2023-03-20	27	3
181	3	2023-04-25	29	4
182	10	2023-05-30	26	5
183	15	2023-06-05	30	6
184	7	2023-07-10	32	7
185	9	2023-08-15	35	8
186	6	2023-09-20	36	9
187	11	2023-10-25	32	10
188	4	2023-11-06	40	11
189	7	2023-11-08	42	12
190	9	2023-11-10	44	13
191	3	2023-11-12	43	14
192	10	2023-11-14	45	15
193	12	2023-11-16	28	16
194	6	2023-11-18	32	17

Figure 9: Highlights sales made for every order.

Customer-Feedback Table

```
INSERT INTO Customer_Feedback (Customer_ID, Order_ID, Rating)
VALUES
    (3, 3, 5),
    (6, 6, 4),
    (5, 13, 5),
    (16, 16, 4),
    (8, 19, 4),
    (19, 22, 3),
```

```
(22, 25, 5),
(1, 9, 4.5),
(1, 1, NULL),
(10, 17, NULL),
(24, 24, 1);
```

Feedback_ID	Customer_ID	Order_ID	Rating
16	3	3	5
17	6	6	4
18	5	13	5
19	16	16	4
20	8	19	4
21	19	22	3
22	22	25	5
23	1	9	5
24	1	1	NULL
25	10	17	NULL
26	24	24	1

Figure 10: Customer ratings on thei respective orders

Data Retrieval and Simple Reports

How many unique items are in the inventory per category and their quantities?

```
SELECT c.Category_Name, COUNT(DISTINCT i.Item_ID) AS
Unique_Items_Per_Category, SUM(i.Item_Quantity) AS
Total_Quantity_Per_Category
FROM Category c
LEFT JOIN Item i ON c.Category_ID = i.Category_ID
GROUP BY c.Category_Name;
```

Category_Name	Unique_Items_Per_Category	Total_Quantity_Per_Category
Computer Parts	4	90
Electronics	5	240
Office Equipment	4	100
Peripherals	4	180
Smartphones	8	235

Who are the top five buyers? What did they purchase? How much did they spend?

```
SELECT c.Customer_ID, c.Customer_Name, s.Item_ID, i.Item_Name,
SUM(s.Quantity_Sold * i.Selling_Price) AS Total_Spent
FROM Customer_Order co
JOIN Sales s ON co.Order_ID = s.Order_ID
JOIN Customer c ON co.Customer_ID = c.Customer_ID
JOIN Item i ON s.Item_ID = i.Item_ID
GROUP BY c.Customer_ID, c.Customer_Name, s.Item_ID, i.Item_Name
ORDER BY Total_Spent DESC
LIMIT 5;
```

Customer_Name	Item_ID	Item_Name	Total_Spent
Eva George	26	Laptop	12000
Logan Collins	27	Smartphone	10400
Jane Shiny	26	Laptop	9600
Jame Johnson	27	Smartphone	9600
John D	26	Laptop	6000

Which item is on demand? How many customers have bought the item? How many items have we sold so far? How much stock is remaining per item?

```
SELECT i.Item_ID, i.Item_Name,
       i.Item_Quantity AS Original_Stock,
       COALESCE(SUM(CASE WHEN t.Transaction_Type = 'Purchase' THEN
t.Transaction_Quantity ELSE 0 END), 0) AS Purchased_Stock,
       COUNT(DISTINCT s.Order_ID) AS Customers_Bought,
       COALESCE(SUM(s.Quantity_Sold), 0) AS Total_Items_Sold,
       (i.Item_Quantity + COALESCE(SUM(CASE WHEN t.Transaction_Type =
'Purchase' THEN t.Transaction_Quantity ELSE 0 END), 0)) -
COALESCE(SUM(s.Quantity_Sold), 0) AS Stock_Remaining
FROM Item i
LEFT JOIN Transaction t ON i.Item_ID = t.Item_ID
LEFT JOIN Sales s ON i.Item_ID = s.Item_ID
GROUP BY i.Item_ID, i.Item_Name, i.Item_Quantity
ORDER BY Total_Items_Sold DESC
limit 10;
```

Item_ID	Item_Name	Original_Stock	Purchased_Stock	Customers_Bought	Total_Items_Sold	Stock_Remaining
45	HDMI Cable	60	110	2	72	98
28	Printer	30	100	2	57	73
26	Laptop	50	400	4	56	394
32	Keyboard	25	600	3	48	577
40	Printer Ink	40	180	2	40	180
27	Smartphone	100	0	2	25	75
44	Power Bank	30	0	2	20	10
30	Chair	40	10	1	15	35
29	Desk	20	0	2	12	8
35	Tablet	45	0	1	9	36

How many orders have a status of pending or shipped (orders not yet delivered)?

```
SELECT Order_Status, COUNT(*) AS Order_Count
FROM Customer_Order
WHERE Order_Status IN ('Pending', 'Shipped')
GROUP BY Order_Status
WITH ROLLUP;
```

Order_Status	Order_Count
Pending	9
Shipped	8
	17

How many customers have rated our delivery services below three?

```
SELECT co.Customer_ID, co.Order_ID, i.Item_Name
FROM Customer_Order co
JOIN Sales s ON co.Order_ID = s.Order_ID
JOIN Customer_Feedbackz cf ON co.Customer_ID = cf.Customer_ID
JOIN Item i ON s.Item_ID = i.Item_ID
WHERE cf.Rating < 3;
```

Customer_ID	Order_ID	Item_Name
24	24	Desk