Final Project Report

MSIS 618

Database Management

Spring 2022

Project
on
Inventory Management System

By
Sushobhit Nandkishor Lutade

Student ID - 02009732

College of Management

Table of Content

1. Executive Summary

Inventory Management system is the combination of innovation (Hardware and software) and forms and methods that direct the checking and support of supplied items, whether those items are company resources, crude materials and supplies, or wrapped up items prepared to be sent to sellers or consumers. This framework can broadly be utilized by ordinary shops, departmental stores or MNCs for keeping an appropriate track of the stock. It moreover comprises data like customer details, stock details etc.

The manual stock gives a tough time to the stock keeper, who needs to accommodate each receipt and the physical stock.The Inventory systems need to have security measures to keep the inventory from unauthorized people.The System should moreover assess how each item is faring in terms of deals.

This Inventory management project was undertaken to help improve the stock management at retail stores where this project will save time by converting manual inventory systems into electronic systems. The other scope of the project is that the inventory system adopted by the store will have accuracy such that it can reconcile the actual physical stock with the electronic records.This may include introducing bar code scanners at the point of deal scanners to stamp up each thing sold.The System will moreover assess how each item is faring in terms of deals.

Objectives of Inventory Management

1. ***Arranged approach towards working : -*** *The working within the organization will be well arranged and organized. The information will be Saved properly in data stores, which is able to help in recovery of data as well as its storage.*
2. ***Accuracy and reliability : -*** *The level of precision within the proposed framework will be higher. All operations would be done accurately, and it guarantees that any information coming from the center is accurate and reliable.*
3. ***No Redundancy : -*** *Within the proposed framework most care would be that no information is repeated anywhere, in storage or otherwise. This would guarantee financial utilization of storage space and consistency within the information stored.*
4. ***Quick recovery of data : -*** *The most objective of the proposed framework is to supply for a speedy and productive recovery of information.*
5. ***Simple to Function : -*** *The framework ought to be simple to function and should be such that it can be created inside a brief period of time and fit within the restricted budget of the client.*

2. Project description

The Project targets the retail outlet and medium to large size stores where manual inventory management is a tedious task and where there are budget constraints to run a profitable business. With the assistance of this system, we can set a threshold for any stock, below which we'll re-order that stock. This will help in great sales outcomes and will never run out of stock for any product.

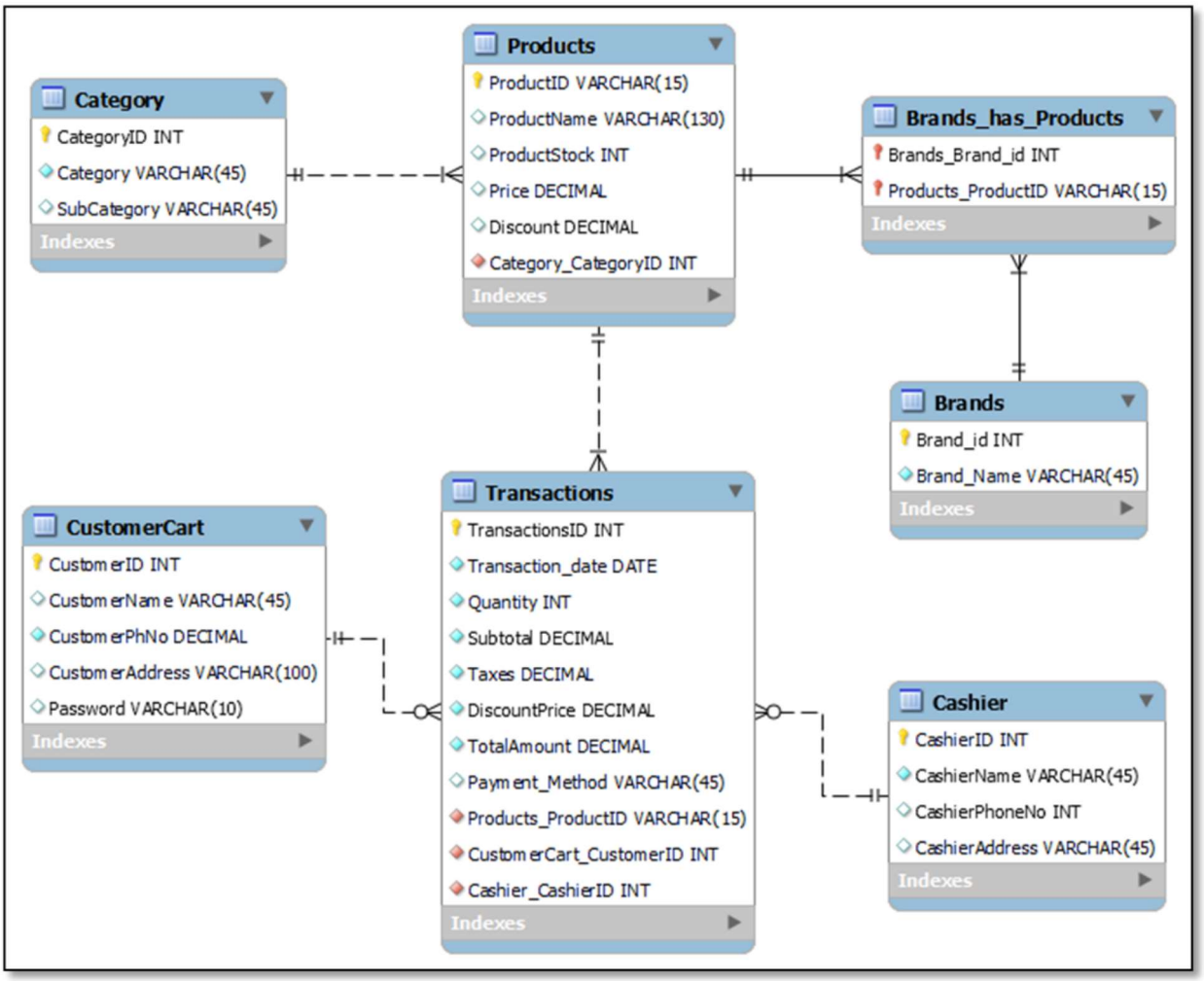
This application will have UI altered to diverse clients' needs. The cashier will only get to adjust the quantity of any item i.e., he/she can either create a receipt for any sold item. The supervisor will have the access to modify the rates if there exists any dynamic price inventory. The proprietor of the firm will have access to produce the final report which can be composed of sales done on any specific day.

The main challenge in store is stock management which manually takes a lot of time in reconciliation and also requires manpower. To address this we use an inventory management system to keep the stock of the product in a computerized database system where a lot of factors are stored which help in stock management and also help in increasing the sales or help in a strategic way to stock up popular products, seasonal goods, etc.

Reports generated from this system can help Store owners in identifying the most loyal customer and offering them premium services and deals to increase net promoter which will help in making brand image and improve revenue through these customers. Can track the productivity of their employees and help improve their performance. Report will help in identifying the most popular products and most popular categories, so that they never run out of these products and lose revenue.

Finally, the report will provide the financial record like revenue, taxes, discounts of the store over the year and identify the trend of customers and be able to predict and manage the store efficiently.

3. Database Design: ER Diagram



4. On SQL we created 7 tables named Below

Brand table hold the data of brand names available at the store.

Category table holds the data of product category and their sub category.

CustomerCart table holds the data of customer details like Names,Contact info, Address, Password.

Transactions table contains details of transaction date, quantity, payment methods,value of goods.

Cashier table holds the cashier name,contact number and address.

Products table has product details like name ,stock, pricing, discount.

Table Name	*Columns*
Brands	Brand_id, Brand_Name
Category	CategoryID, Category, SubCategory
CustomerCart	CustomerID, CustomerName, CustomerPhNo, CustomerAddress, Password
Transactions	TransactionsID, Transaction_date, Quantity, Subtotal, Taxes, DiscountPrice, TotalAmount, Payment_Method, Products_ProductID, CustomerCart_CustomerID, Cashier_CashierID
Cashier	CashierID, CashierName, CashierPhoneNo, CashierAddress
Products	ProductID, ProductName, ProductStock, Price, Discount, Category_CategoryID
Brands_has_products	Brands_Brand_id, Products_ProductID

5. Normalization

Database normalization is a planned procedure for organizing information inside a database.

Here are three reasons why you must utilize database normalization inside your retail or eCommerce brand:

- It Decreases Information Redundancy Database
- Normalization Kills Undesirable Characteristics Such As Cancellation Anomalies Plus,
- It Guarantees Information Is Put away Consistently For Effective Utilize

Below mentioned Table have been normalized

1. *Brands (Brand_id, Brand_Name)*
2. *Category (CategoryID, Category, SubCategory)*
3. *Cashier (CashierID, CashierName, CashierPhoneNo, CashierAddress)*
4. *CustomerCart (CustomerID, CustomerName, CustomerPhNo, CustomerAddress, Password)*
5. *Transactions (TransactionsID, Transaction_date, Quantity, Subtotal, Taxes, DiscountPrice, TotalAmount, Payment_Method, Products_ProductID, CustomerCart_CustomerID, Cashier_CashierID)*
    - Where Transactions.Cashier_CashierID must exist in Cashier.CashierID
    - Where Transactions.CustomerCart_CustomerID must exist in CustomerCart.CustomerID
    - Where Transactions.Products_ProductID must exist in Products.ProductID
6. *Products (ProductID, ProductName, ProductStock, Price, Discount, Category_CategoryID)*
    - Where must Products.Category_CategoryID must exist in Category.CategoryID
7. *Brands_has_products (Brands_Brand_id, Products_ProductID)*
    - Where must Brands_has_products. Brands_Brand_id must exist in Brands.Brand_id
    - Where must Brands_has_products. Products_ProductID Must exist in Products.Products_ProductID

As above relations are in BCNF now, the normalization is completed.

6. Database Implementation

The DDL statement for table creation was done by using the forward engineering option in MySQL software. And for populating data,Insert statement were created from CSV file by using the online website which convert the CSV file in to statements

A. Data Insert Sample

```
CREATE SCHEMA IF NOT EXISTS `InventoryManagement` DEFAULT CHARACTER SET utf8 ;
USE `InventoryManagement` ;
 -- -----------------------------------------------------
-- Table `InventoryManagement`.`Brands`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `InventoryManagement`.`Brands` (
  `Brand_id` INT NOT NULL AUTO_INCREMENT,
  `Brand_Name` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`Brand_id`))
;
 -- -----------------------------------------------------
-- Table `InventoryManagement`.`Category`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `InventoryManagement`.`Category` (
  `CategoryID` INT NOT NULL AUTO_INCREMENT,
  `Category` VARCHAR(45) NOT NULL,
  `SubCategory` VARCHAR(45) NULL,
  PRIMARY KEY (`CategoryID`))
;
 -- -----------------------------------------------------
-- Table `InventoryManagement`.`CustomerCart`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `InventoryManagement`.`CustomerCart` (
  `CustomerID` INT NOT NULL AUTO_INCREMENT,
  `CustomerName` VARCHAR(45) NULL DEFAULT NULL,
  `CustomerPhNo` DECIMAL NOT NULL,
  `CustomerAddress` VARCHAR(100) NULL DEFAULT NULL,
  `Password` VARCHAR(10) NULL DEFAULT NULL,
  PRIMARY KEY (`CustomerID`))
;
 -- -----------------------------------------------------
-- Table `InventoryManagement`.`Products`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `InventoryManagement`.`Products` (
  `ProductID` VARCHAR(15) NOT NULL,
  `ProductName` VARCHAR(130) NULL DEFAULT NULL,
  `ProductStock` INT NULL DEFAULT NULL,
  `Price` NUMERIC(9,4) NULL DEFAULT NULL,
  `Discount` DECIMAL NULL DEFAULT NULL,
  `Category_CategoryID` INT NOT NULL,
  PRIMARY KEY (`ProductID`),
  INDEX `fk_Products_Category1_idx` (`Category_CategoryID` ASC) VISIBLE,
  CONSTRAINT `fk_Products_Category1`
        FOREIGN KEY (`Category_CategoryID`)
        REFERENCES `InventoryManagement`.`Category` (`CategoryID`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
```

```
;
 -- -----------------------------------------------------
-- Table `InventoryManagement`.`Cashier`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `InventoryManagement`.`Cashier` (
  `CashierID` INT NOT NULL AUTO_INCREMENT,
  `CashierName` VARCHAR(45) NOT NULL,
  `CashierPhoneNo` NUMERIC NULL,
  `CashierAddress` VARCHAR(45) NULL,
  PRIMARY KEY (`CashierID`))
;
-- -----------------------------------------------------
-- Table `InventoryManagement`.`Transactions`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `InventoryManagement`.`Transactions` (
  `TransactionsID` INT NOT NULL AUTO_INCREMENT,
  `Transaction_date` DATE NOT NULL,
  `Quantity` INT NOT NULL,
  `Subtotal` NUMERIC(12,4) NOT NULL,
  `Taxes` NUMERIC(12,4) NOT NULL,
  `DiscountPrice` NUMERIC(12,4) NOT NULL,
  `TotalAmount` NUMERIC(12,4) NOT NULL,
  `Payment_Method` VARCHAR(45) NULL DEFAULT NULL,
  `Products_ProductID` VARCHAR(15) NOT NULL,
  `CustomerCart_CustomerID` INT NOT NULL,
  `Cashier_CashierID` INT NOT NULL,
  PRIMARY KEY (`TransactionsID`),
  INDEX `fk_Transactions_Products1_idx` (`Products_ProductID` ASC) VISIBLE,
  INDEX `fk_Transactions_CustomerCart1_idx` (`CustomerCart_CustomerID` ASC) VISIBLE,
  INDEX `fk_Transactions_Cashier1_idx` (`Cashier_CashierID` ASC) VISIBLE,
  CONSTRAINT `fk_Transactions_Products1`
        FOREIGN KEY (`Products_ProductID`)
        REFERENCES `InventoryManagement`.`Products` (`ProductID`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
  CONSTRAINT `fk_Transactions_CustomerCart1`
        FOREIGN KEY (`CustomerCart_CustomerID`)
        REFERENCES `InventoryManagement`.`CustomerCart` (`CustomerID`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
  CONSTRAINT `fk_Transactions_Cashier1`
        FOREIGN KEY (`Cashier_CashierID`)
        REFERENCES `InventoryManagement`.`Cashier` (`CashierID`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION);
-- -----------------------------------------------------
-- Table `InventoryManagement`.`Brands_has_Products`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `InventoryManagement`.`Brands_has_Products` (
  `Brands_Brand_id` INT NOT NULL,
  `Products_ProductID` VARCHAR(15) NOT NULL,
  PRIMARY KEY (`Brands_Brand_id`, `Products_ProductID`),
  INDEX `fk_Brands_has_Products_Products1_idx` (`Products_ProductID` ASC) VISIBLE,
  INDEX `fk_Brands_has_Products_Brands1_idx` (`Brands_Brand_id` ASC) VISIBLE,
  CONSTRAINT `fk_Brands_has_Products_Brands1`
        FOREIGN KEY (`Brands_Brand_id`)
```

```
      REFERENCES `InventoryManagement`.`Brands` (`Brand_id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_Brands_has_Products_Products1`
      FOREIGN KEY (`Products_ProductID`)
      REFERENCES `InventoryManagement`.`Products` (`ProductID`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION);
```

B. Data Insert Sample

```
-- -----------------------------------------------------
-- Insert into `InventoryManagement`.`Brands`
-- -----------------------------------------------------
INSERT INTO Brands(Brand_Name) VALUES ('Superstore');
-- -----------------------------------------------------
-- Insert into `InventoryManagement`.`Category`
-- -----------------------------------------------------
INSERT INTO Category(Category,SubCategory) VALUES ('Furniture','Bookcases');
INSERT INTO Category(Category,SubCategory) VALUES ('Furniture','Chairs');
INSERT INTO Category(Category,SubCategory) VALUES ('Office Supplies','Labels');
INSERT INTO Category(Category,SubCategory) VALUES ('Furniture','Tables');
-- -----------------------------------------------------
-- Insert into `InventoryManagement`.`CustomerCart`
-- -----------------------------------------------------
INSERT INTO CustomerCart(CustomerName,CustomerPhNo,CustomerAddress,Password) VALUES
('Claire Gute',9167035808,'Henderson,Kentucky,United States','NZ76S29');
INSERT INTO CustomerCart(CustomerName,CustomerPhNo,CustomerAddress,Password) VALUES
('Darrin Van Huff',7495777225,'Los Angeles,California,United States','ND90W12');
INSERT INTO CustomerCart(CustomerName,CustomerPhNo,CustomerAddress,Password) VALUES
('Sean O''Donnell',3795373234,'Fort Lauderdale,Florida,United States','NX38M64');
-- -----------------------------------------------------
-- Insert into `InventoryManagement`.`Cashier `
-- -----------------------------------------------------
INSERT INTO Cashier(CashierID,CashierName,CashierPhoneNo,CashierAddress) VALUES
(1,'Sushobhit',4174174178,'Umass boston');
INSERT INTO Cashier(CashierID,CashierName,CashierPhoneNo,CashierAddress) VALUES
(2,'Chaitanya',5175175178,'Boston comon');
INSERT INTO Cashier(CashierID,CashierName,CashierPhoneNo,CashierAddress) VALUES
(3,'Sharvari',6176176178,'Downtown boston');
-- -----------------------------------------------------
-- Insert into `InventoryManagement`.`Products`
-- -----------------------------------------------------
INSERT INTO
Products(ProductID,ProductName,ProductStock,Price,Discount,Category_CategoryID) VALUES
('FUR-FU-10001185','Advantus Employee of the Month Certificate Frame, 11 x
13-1/2','37',30.93,10,6);
INSERT INTO
Products(ProductID,ProductName,ProductStock,Price,Discount,Category_CategoryID) VALUES
('FUR-FU-10001196','DAX Cubicle Frames - 8x10','34',11.54,10,6);
INSERT INTO
Products(ProductID,ProductName,ProductStock,Price,Discount,Category_CategoryID) VALUES
('FUR-FU-10001215','Howard Miller 11-1/2" Diameter Brentwood Wall
Clock','22',34.504,10,6);
```

```
-- -----------------------------------------------------
-- Insert into `InventoryManagement`.`Transactions`
-- -----------------------------------------------------
INSERT INTO
transactions(Transaction_date,Quantity,Subtotal,Taxes,DiscountPrice,TotalAmount,Payment_
Method,Products_ProductID,CustomerCart_CustomerID,Cashier_CashierID) VALUES
('2014-11-24',2,46.4,2.61,4.64,49.01,'netbanking','TEC-AC-10002345',85,2);
INSERT INTO
transactions(Transaction_date,Quantity,Subtotal,Taxes,DiscountPrice,TotalAmount,Payment_
Method,Products_ProductID,CustomerCart_CustomerID,Cashier_CashierID) VALUES
('2016-12-02',2,5.73,0.36,0,6.09,'netbanking','OFF-BI-10001890',86,1);
INSERT INTO
transactions(Transaction_date,Quantity,Subtotal,Taxes,DiscountPrice,TotalAmount,Payment_
Method,Products_ProductID,CustomerCart_CustomerID,Cashier_CashierID) VALUES
('2014-08-30',8,282.88,15.91,28.29,298.79,'Debitcard','TEC-AC-10002323',87,1);
-- -----------------------------------------------------
-- Insert into `InventoryManagement`.`Brands_has_Products`
-- -----------------------------------------------------
INSERT INTO Brands_has_products(Brands_Brand_id,Products_ProductID) VALUES
(1,'TEC-AC-10002018');
INSERT INTO Brands_has_products(Brands_Brand_id,Products_ProductID) VALUES
(1,'TEC-AC-10002049');
INSERT INTO Brands_has_products(Brands_Brand_id,Products_ProductID) VALUES
(1,'TEC-AC-10002076');
```

C. Stored Procedure

Stored Procedure is used to find the out of stock product, so that we cannot run out of stock for any product. This stored procedure can be executed at the end of the day to prepare the purchase order for out of stock products.

```
-- -----------------------------------------------------
-- Procedure:
-- -----------------------------------------------------
Drop Procedure if exists Status;
DELIMITER //
Create procedure inventorymanagement.Status(In X Integer)
begin
select ProductName,
Totalstock,
case when Totalstock >= X then 'Stock Available'
when Totalstock < X then 'Out of Stock'
else Null
end as StockStatus from (
Select ProductName,Sum(Productstock) as Totalstock from Products
Group by Productname
order by Sum(Productstock)) as temp;
end //
DELIMITER ;
Call Status(10);
```

D. Trigger

Trigger here is adding an entry to the new_Products table whenever a new product is added to the products Table. This can be used to check which are new product and what is the stock quantity added

```
create table new_products(ProductName varchar(255) ,ProductStock numeric)
DELIMITER $$
CREATE TRIGGER after_product_insert
AFTER INSERT
ON Products FOR EACH ROW
BEGIN
        INSERT INTO new_products
        VALUES(new.ProductName,new.ProductStock);
END$$
DELIMITER ;

-- When we insert the new product in Products Table
INSERT INTO
Products(ProductID,ProductName,ProductStock,Price,Discount,Category_CategoryID) VALUES
('NEW-BO-1000015','Water Bottle','50',15,5,1);
```

	ProductName	ProductStock
▶	Water Bottle	50

7. Analysis Queries on Database

Q1). *Top 5 customers who spent the most cash in the store?*

```
Select customercart.customerId, customercart.customername, sum(TotalAmount) as
cash_Spent from customercart
              Join transactions
    on customercart.CustomerID=transactions.CustomerCart_CustomerID
group by CustomerCart_CustomerID
order by sum(TotalAmount) desc
Limit 5;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Co

customerId	customername	cash_Spent
599	Sean Miller	149159.7200
569	Tamara Chand	46589.6900
667	Christopher Conant	42625.5800
639	Tom Ashbrook	40967.1000
741	Bill Shonely	40200.9200

Q2). *What is the annual value and quantity of products sold also taxes Paid?*

```
select year(transaction_date), sum(Quantity) as No_of_Items_Sold, sum(TotalAmount) as
Value_of_Products_Sold, sum(Taxes) as TaxesPaid
from transactions
group by year(transaction_date)
order by Value_of_Products_Sold desc;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

year(transaction_date)	No_of_Items_Sold	Value_of_Products_Sold	TaxesPaid
2017	12425	1278951.8200	64624.9200
2016	9823	1173369.1100	60157.6700
2014	7365	949078.2000	48702.0800
2015	8109	837711.6300	42127.8700
2018	151	7567.7100	362.7400

Q3). What are the top ten most popular items in terms of price and quantity?(Product Performance Report)

A) By Price

```
Select transactions.Products_ProductID, products.ProductName,
Sum(transactions.TotalAmount) as Value from transactions
 join products
 on products.ProductID=transactions.Products_ProductID
Group by transactions.Products_ProductID
order by value desc
Limit 10;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

Products_ProductID	ProductName	Value
TEC-CO-10004722	Canon imageCLASS 2200 Advanced Copier	177974.4900
TEC-MA-10002412	Cisco TelePresence System EX90 Videoconferen...	143895.8400
TEC-MA-10001047	3D Systems Cube Printer, 2nd Generation, Mag...	60595.7800
TEC-MA-10000822	Lexmark MX611dhe Monochrome Laser Printer	48625.0200
OFF-SU-10000151	High Speed Automatic Electric Letter Opener	45797.6200
FUR-CH-10002024	HON 5400 Series Task Chairs for Big and Tall	40306.7900
TEC-CO-10001449	Hewlett Packard LaserJet 3310 Copier	38417.3800
FUR-BO-10004834	Riverside Palais Royal Lawyers Bookcase, Royal...	38078.1600
TEC-MA-10004125	Cubify CubeX 3D Printer Triple Head Print	33899.9200
TEC-MA-10001127	HP Designjet T520 Inkjet Large Format Printer -...	33370.1200

B) By Quantity

```
Select transactions.Products_ProductID, products.ProductName, Sum(transactions.Quantity)
as TotalQuantity from transactions
 join products
 on products.ProductID=transactions.Products_ProductID
Group by transactions.Products_ProductID
order by TotalQuantity desc
Limit 10;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

Products_ProductID	ProductName	TotalQuantity
TEC-AC-10003832	Imation 16GB Mini TravelDrive USB 2.0 Flash Drive	75
OFF-PA-10001970	Xerox 1881	70
OFF-BI-10001524	GBC Premium Transparent Covers with Diagonal...	67
OFF-BI-10002026	Ibico Recycled Linen-Style Covers	64
FUR-CH-10002647	Situations Contoured Folding Chairs, 4/Set	64
FUR-TA-10001095	Chromcraft Round Conference Tables	61
TEC-AC-10002049	Logitech G19 Programmable Gaming Keyboard	60
FUR-CH-10003774	Global Wood Trimmed Manager's Task Chair, Khaki	59
OFF-BI-10004728	Wilson Jones Turn Tabs Binder Tool for Ring Bin...	59
FUR-FU-10001473	Eldon Executive Woodline II Desk Accessories, ...	57

Q4). Which cashier has attended the most number of customers.

```
Select  Cashier.CashierName as Cashier, Count(Distinct
transactions.CustomerCart_CustomerID) as CustomerAttended from cashier
Join transactions
On cashier.CashierID=transactions.Cashier_CashierID
Group By Cashier_CashierID
Order by CustomerAttended desc;
```

	Cashier	CustomerAttended
▶	Sushobhit	286
	Chaitanya	255
	Sharvari	252

Q5). How many products are there in each category and What is value of stock inHand?

```
Select Category, SubCategory, Count(ProductName)as ProductCount,sum(ProductStock*Price)
as Value
from category join products
ON category.CategoryID=products.Category_CategoryID
Group by Category,SubCategory
Order By Category, ProductCount desc;
```

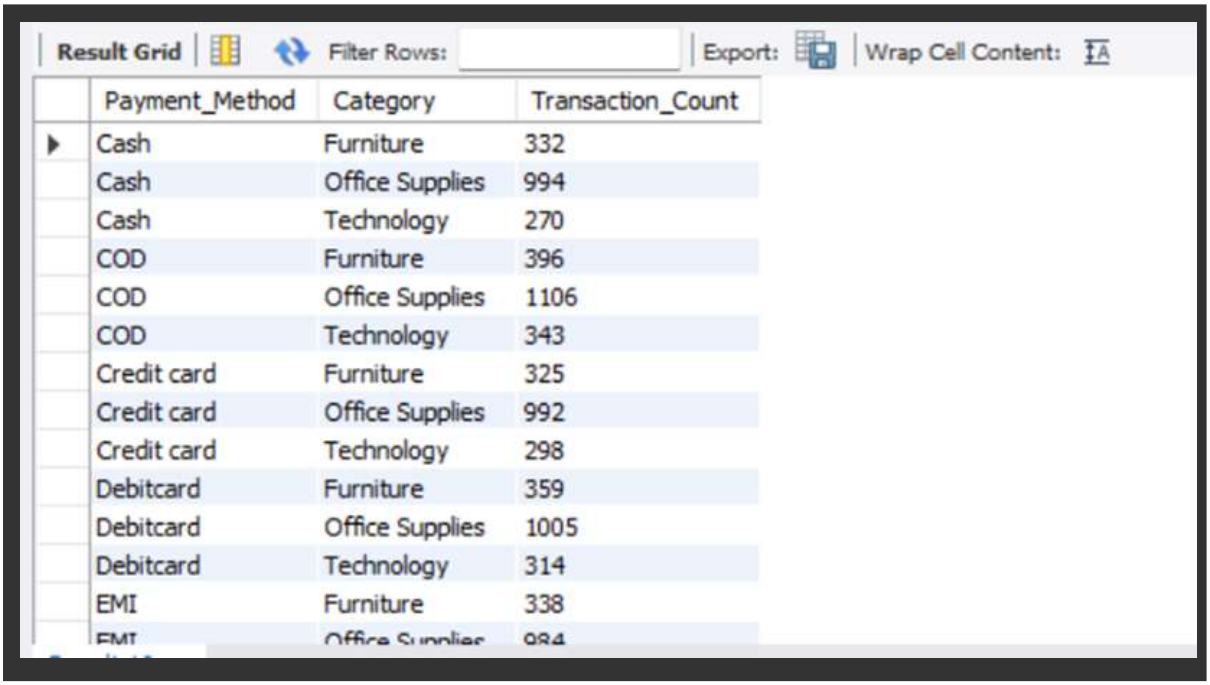
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	Category	SubCategory	ProductCount	Value
▶	Furniture	Furnishings	182	194165.6640
	Furniture	Chairs	87	645058.9840
	Furniture	Tables	57	394298.0315
	Furniture	Bookcases	49	268845.5964
	Office Supplies	Paper	276	164355.6740
	Office Supplies	Binders	210	326235.4430
	Office Supplies	Art	163	73213.0580
	Office Supplies	Storage	131	446156.5480
	Office Supplies	Appliances	98	220624.7660
	Office Supplies	Labels	70	29954.7240
	Office Supplies	Envelopes	54	36522.1180
	Office Supplies	Fasteners	43	8139.7460
	Office Supplies	Supplies	38	97580.7520
	Technology	Phones	184	675399.4840

Result 20

Q6). Which Payment method is used to pay according to Product category?

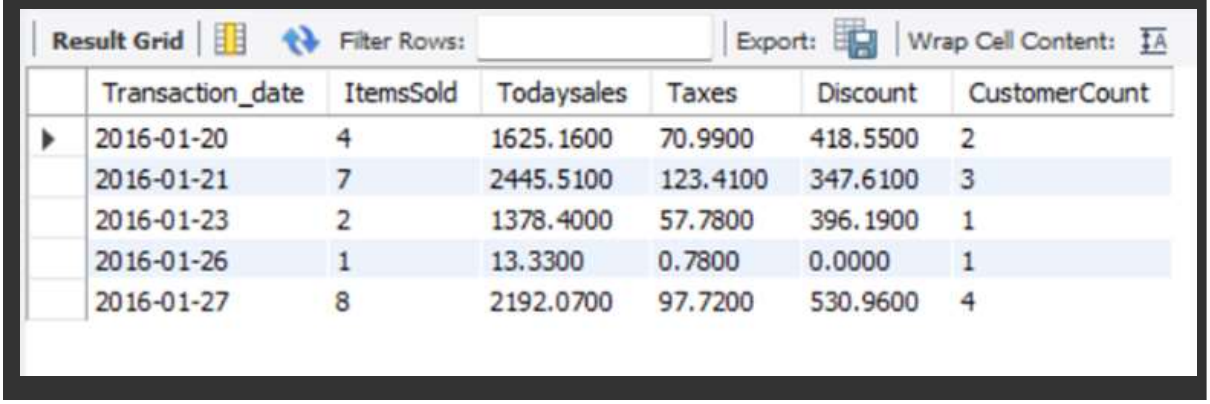
```
Select Payment_Method,Category ,  Count(Payment_Method) as Transaction_Count
 from transactions Join (Products Join category
              On category.CategoryID=products.Category_CategoryID)
 on transactions.Products_ProductID=products.ProductID
Group By category,Payment_Method
order by Payment_Method;
```



Payment_Method	Category	Transaction_Count
Cash	Furniture	332
Cash	Office Supplies	994
Cash	Technology	270
COD	Furniture	396
COD	Office Supplies	1106
COD	Technology	343
Credit card	Furniture	325
Credit card	Office Supplies	992
Credit card	Technology	298
Debitcard	Furniture	359
Debitcard	Office Supplies	1005
Debitcard	Technology	314
EMI	Furniture	338
EMI	Office Supplies	984

Q7). sales report for past 7 days

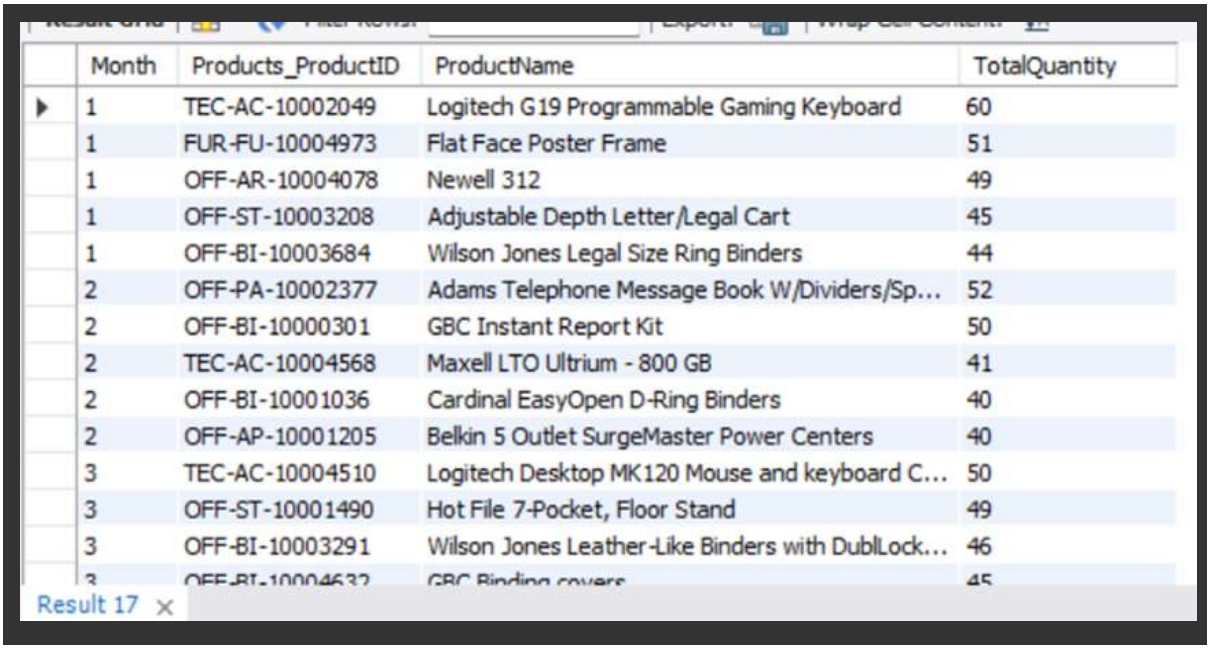
```
Select Transaction_date,count(quantity) as ItemsSold,Sum(TotalAmount) as
Todaysales,sum(Taxes) as Taxes,Sum(DiscountPrice) as Discount, Count(Distinct
CustomerCart_CustomerID) As CustomerCount
FROM transactions
Where Transaction_date  between  '2016-01-20' and '2016-01-27'
Group by Transaction_date;
```



Transaction_date	ItemsSold	Todaysales	Taxes	Discount	CustomerCount
2016-01-20	4	1625.1600	70.9900	418.5500	2
2016-01-21	7	2445.5100	123.4100	347.6100	3
2016-01-23	2	1378.4000	57.7800	396.1900	1
2016-01-26	1	13.3300	0.7800	0.0000	1
2016-01-27	8	2192.0700	97.7200	530.9600	4

Q8). Five Most selling products by month?

```
Select Month, Products_ProductID, ProductName, TotalQuantity from (
Select month(transaction_date) As Month, transactions.Products_ProductID,
products.ProductName, Sum(transactions.Quantity) as TotalQuantity,
rank() over (partition by month(transaction_date) order by Sum(transactions.Quantity)
desc) as Max_Quant
              from transactions
              join products
              on products.ProductID=transactions.Products_ProductID
              Group by transactions.Products_ProductID
              order by month , TotalQuantity desc) as tmp
    where max_quant<=5;
```



Month	Products_ProductID	ProductName	TotalQuantity
1	TEC-AC-10002049	Logitech G19 Programmable Gaming Keyboard	60
1	FUR-FU-10004973	Flat Face Poster Frame	51
1	OFF-AR-10004078	Newell 312	49
1	OFF-ST-10003208	Adjustable Depth Letter/Legal Cart	45
1	OFF-BI-10003684	Wilson Jones Legal Size Ring Binders	44
2	OFF-PA-10002377	Adams Telephone Message Book W/Dividers/Sp...	52
2	OFF-BI-10000301	GBC Instant Report Kit	50
2	TEC-AC-10004568	Maxell LTO Ultrium - 800 GB	41
2	OFF-BI-10001036	Cardinal EasyOpen D-Ring Binders	40
2	OFF-AP-10001205	Belkin 5 Outlet SurgeMaster Power Centers	40
3	TEC-AC-10004510	Logitech Desktop MK120 Mouse and keyboard C...	50
3	OFF-ST-10001490	Hot File 7-Pocket, Floor Stand	49
3	OFF-BI-10003291	Wilson Jones Leather-Like Binders with DublLock...	46
3	OFF-BI-10004632	GBC Binding covers	45

Result 17

8. Conclusion

In this project we created a total backend computer program in which we are able to upgrade the stock, alter stock, we are able figure the stock, create invoice. From this application we are able to get an overhaul that in case a specific stock or stock is less than a few pre-fixed amounts at that point it'll be simple for the manager/owner to reorder the item from the provider to overcome the "Out of Stock" stage. In expansion to this it can moreover offer assistance to oversee the inventory, including inventory which can be demonstrated as an exceptionally valuable feature From this project able to keep a track of exchanges performed by distinctive customers/clients. Ready to get a thought on how much finance we get from diverse installment methodologies.