



Inventory Management System

Database Management System

APRIL 2022

BY

GROUP 17,

MOREM JAYANTH KUMAR 111901033

HARSH DUBEY 111901053

SRINIVAS 111901037

Overview :

Main facilities available in this project are:

1. We can forecast the sales by analysing the previous sales statistics.
2. We can get an idea of when we need to order new inventory.
3. We can reduce the chances of any kind of frauds done by the staff members in the inventory.
4. Customer details can be added.
5. Invoice generation.
6. We can keep a track of transactions received through different payment methods.

The project maintains three levels of users:

1. Online shopping store Level
2. Manager Level
3. Owner Level

Introduction :

An inventory management system is the combination of technology (hardware and software) and processes and procedures that oversee the monitoring and maintenance of stocked products, whether those products are company assets, raw materials and supplies, or finished products ready to be sent to vendors or end consumers. This system can widely be used by normal shops, departmental stores or MNCs for keeping a proper track of the stock. It also consists of information like manager details, customer details etc. With the help of this system we can fix a minimum quantity of any inventory below which we need to place an order for that inventory. This will help us in good sales results and never the out of stock stage for any inventory.



Scope:

1. This will help us in maintaining the exact count of any product.
2. Can help us to set a minimum quantity of any product below which we can order the product from the manufacturer.
3. Can reduce duplicate entries



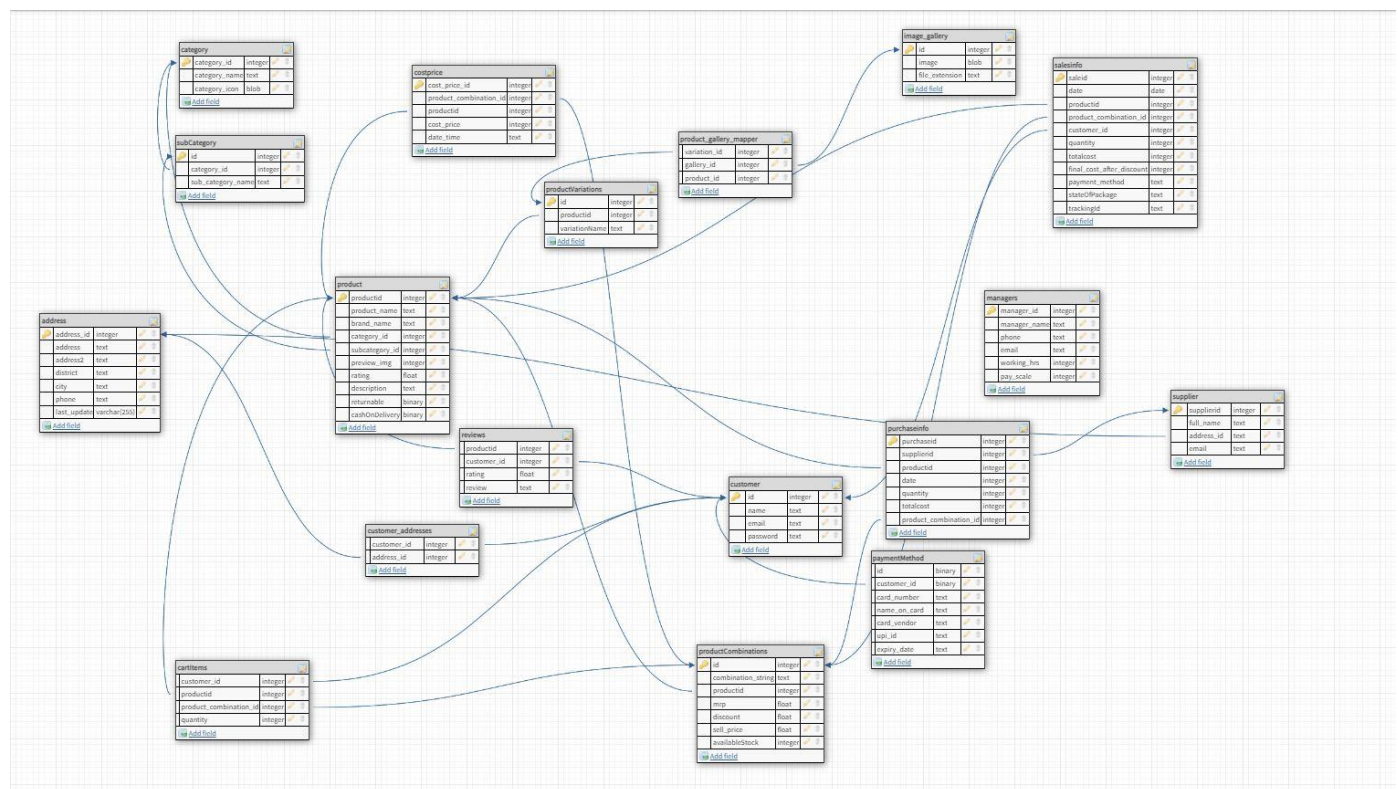
Goals of proposed system:

1. Planned approach towards working: - The working in the organisation will be well planned and organised. The data will be stored properly in data stores, which will help in retrieval of information as well as its storage.
2. Accuracy: - The level of accuracy in the proposed system will be higher. All operations would be done correctly and it ensures that whatever information is coming from the centre is accurate.
3. Reliability: - The reliability of the proposed system will be high due to the above stated reasons. The reason for the increased reliability of the system is that now there would be proper storage of information.
4. No Redundancy: - In the proposed system utmost care would be that no information is repeated anywhere, in storage or otherwise. This would assure economic use of storage space and consistency in the data stored.
5. Immediate retrieval of information: - The main objective of the proposed system is to provide for a quick and efficient retrieval of information.
6. Immediate storage of information: - In manual systems there are many problems with storing the largest amount of information.
7. Easy to Operate: - The system should be easy to operate and should be such that it can be developed within a short period of time and fit in the limited budget of the user

Background:

This application is nowadays a basic use of any company, firm, shop or departmental store because stock maintenance, stock forecasting are some things which are very essential these days for earning great profits. In ancient times we needed to maintain the complete inventory in paper pen method. The ancient method is quite uneasy, uncomfortable and sometimes inaccurate. To overcome this problem we came up with an inventory management system. From this system we can generate invoices for each and every purchase.

ER Diagram:





Relational Schema:

Address			
Attribute	Datatype	Primary or Foreign Key	Constraints
address_id	bigserial	PRIMARY key	
address	text		not null
address2	text		
district	text		not null
city	text		not null
phone	text		not null
last_update	text		default TIMEOFDAY()

supplier			
ATTRIBUTE	DATATYPE	PRIMARY OR FOREIGN KEY	CONSTRAINTS
supplied	BIGSERIAL	PRIMARY KEY	UNIQUE
full_name	VARCHAR(100)		NOT NULL
address_id	bigserial		NOT NULL
email	VARCHAR(100)		NOT NULL UNIQUE

CATEGORY			
ATTRIBUTE	DATATYPE	PRIMARY OR FOREIGN KEY	CONSTRAINTS
category_id	BIGSERIAL	PRIMARY KEY	UNIQUE
category_name	VARCHAR(50)		NOT NULL, UNIQUE

subCategory			
ATTRIBUTE	DATATYPE	PRIMARY OR FOREIGN KEY	CONSTRAINTS
id	bigserial	PRIMARY KEY	
category_id	int		not null
sub_category_name	text		not null

product			
ATTRIBUTE	DATATYPE	PRIMARY OR FOREIGN KEY	CONSTRAINTS
productid	BIGSERIAL	PRIMARY KEY	
product_name	VARCHAR(50)		NOT NULL
brand_name	VARCHAR(50)		
category_id	INT		NOT NULL
subcategory_id	INT		
preview_img_id	int		
rating	numeric(2,1)		

description	int		default 1
returnable	int		default 1
cashonDelivery	int		default 1

image_gallery			
ATTRIBUTE	DATATYPE	PRIMARY OR FOREIGN KEY	CONSTRAINTS
id	bigserial	PRIMARY KEY	
image	BYTEA		
file_extension	text		

product_gallery_mapper			
ATTRIBUTE	DATATYPE	PRIMARY OR FOREIGN KEY	CONSTRAINTS
variation_id	bigserial		
gallery_id	bigserial		
product_id	bigserial		

productVariations			
ATTRIBUTE	DATATYPE	PRIMARY OR FOREIGN KEY	CONSTRAINTS
id	bigserial		
productid	bigserial		not null
variationName	text		not null

--	--	--	--

productCombinations			
ATTRIBUTE	DATATYPE	PRIMARY OR FOREIGN KEY	CONSTRAINTS
id	bigserial		
combination_string	text		not null
productid	bigserial		not null
mrp	DECIMAL(6,2)		NOT NULL
discount	int		not null default 0
sell_price	DECIMAL(6,2)		
cost_price	DECIMAL(6,2)		NOT NULL
availableStock	int		not null

offers			
ATTRIBUTE	DATATYPE	PRIMARY OR FOREIGN KEY	CONSTRAINTS
offer_id	BIGSERIAL	PRIMARY KEY	UNIQUE
offer_decription	text		NOT NULL
offer_discount	int		NOT NULL, CHECK(offer_discount>=0 &offer_discount<100)
type	text		NOT NULL
min_purchase	int		NOT NULL

CUSTOMER			
ATTRIBUTE	DATATYPE	PRIMARY OR FOREIGN KEY	CONSTRAINTS
id	BIGSERIAL	PRIMARY KEY	UNIQUE
name	text		NOT NULL
email	text		NOT NULL, CHECK (email ~ '^[A-Za-z0-9._%~ @[A-Za-z0-9.-]+[.][A -Za-z]+\$')
password	text		NOT NULL

catItems			
ATTRIBUTE	DATATYPE	PRIMARY OR FOREIGN KEY	CONSTRAINTS
customer_id	bigserial		
productid	bigserial		
product_combination_id	bigserial		
quantity	int		

reviews

ATTRIBUTE	DATATYPE	PRIMARY OR FOREIGN KEY	CONSTRAINTS
productid	BIGSERIAL		
customer_id	BIGSERIAL		
rating	int		NOT NULL
review	text		

Customer_addresses

customer_addresses			
ATTRIBUTE	DATATYPE	PRIMARY OR FOREIGN KEY	CONSTRAINTS
customer_id	BIGSERIAL	FOREIGN KEY	not null
address_id	BIGSERIAL	FOREIGN KEY	not null

paymentMethods			
ATTRIBUTE	DATATYPE	PRIMARY OR FOREIGN KEY	CONSTRAINTS
id	bigserial		UNIQUE(customer_id,id)
customer_id	bigserial	FOREIGN KEY	UNIQUE(customer_id,id)
card_number	text		
name_on_card	text		
card_vendor	text		
upi_id	text		
expiry_date	date		

PURCHASEINFO			
ATTRIBUTE	DATATYPE	PRIMARY OR FOREIGN KEY	CONSTRAINTS
purchaseid	BIGSERIAL	PRIMARY KEY	UNIQUE
supplierid	INT	FOREIGN KEY	NOT NULL
productid	INT	FOREIGN KEY	NOT NULL
date	varchar(45) default CURRENT_TIMESTAMP		not null
quantity	int		not null, (quantity>0)
totalcost	int		not null, (totalcost>0)
product_combination_id	BIGSERIAL		not null

SALESINFO			
ATTRIBUTE	DATATYPE	PRIMARY OR FOREIGN KEY	CONSTRAINTS
saleid	BIGSERIAL	PRIMARY KEY	UNIQUE
date	varchar(45) default CURRENT_TIMESTAMP		NOT NULL
productid	INT	FOREIGN KEY	NOT NULL
product_combination_id	bigserial	FOREIGN KEY	not null
customer_id	BIGSERIAL	FOREIGN KEY	UNIQUE
quantity	int		not null, (quantity>0)

totalcost	int		not null, (totalcost>0)
final_cost_after_discount	int		not null
payment_method	varchar(45) default 'credit card'		not null
stateOfPackage	text		not null
trackingId	text		not null

managers			
ATTRIBUTE	DATATYPE	PRIMARY OR FOREIGN KEY	CONSTRAINTS
manager_id	bigserial	PRIMARY KEY	
manager_name	text		not null
phone	text		not null
email	text		CHECK (email ~ '^[A-Za-z0-9._%~]+ @[A-Za-z0-9.-]+[.][A- -Za-z]+\$')
working_hrs	int		
pay_scale	int		



Views:

productCombinationsCustomerView:

1. In the productCombinations table, we are storing the cost_price which is the price at which we have bought the product from the supplier.

2. We should not disclose this price to our customers, so we are creating a view which doesn't contain the cost_price and granting select permission on that view to the customers.

passwordsHiddenCustomersTableView:

1. In the customers table, we are storing the encrypted format of the user passwords.
2. Although the passwords are encrypted, it's a safer measure to hide the encrypted passwords from the public to be safe from cyber attacks.
3. So, I created a view on the customer table hiding the passwords attribute and granted select permissions to the manager and the customer roles.

customerAddresses:

1. For storing the addresses of both the suppliers and the customers, we are using the same table `address`.
2. In order to hide the addresses of the suppliers from the customers, we are creating a view on the address table where each tuple belongs to the customers only. Finally, I granted select permissions to the customer role.



Roles:

Dbadmin:

1. Db-admin role suits for the owner of the inventory store who has the authority of data in the entire database, supplier purchases, customers information, sales, purchases, managers who work under him/her.

Customer:

1. Customer role is assigned for the customers who shop in the inventory. This role has only a select permission to view all the product details, current offers, reviews of the products, and can see only the customer's saved addresses, his/her's payment methods and purchase information.

Manager:

1. Manager role is for all the managers working on maintaining the inventory store.
2. A manager has all the privileges on purchasing products from the suppliers and can modify the details of existing products in the store, add new offers, can look through the customers data except their passwords and can manage the sales information.



Files Attached :

Link to the files :

<https://drive.google.com/drive/folders/19owbqqjKgu51l11P4WhrmbGWbTUs8iDZ?usp=sharing>

1. init.sql -- contains all the commands to create the database with all the tables.
2. data.sql -- contains all the data
3. procedures.sql -- contains all the procedures and functions.
4. roles.sql -- contains all the commands to create the roles.
5. Image.py -- contains operations to store images in the database
6. Indices.sql -- contains all the indices in the database
7. Triggers.sql -- contains all the triggers in the database
8. Views.sql -- contains all the views created in the database

To import the database into your windows local machine, run installdb.bat accordingly.



THANK YOU