University of Science and Technology Bannu.

Whole Sell Management

Project of Data Base by Sir Sadaqat

Niaz khan

Sufyan

Burhan

Saifullah

AbdulSamad

Zia-ur-Rahman

2023

WHOLE SALE MANAGEMENT

CONTENTS

INTRODUCTION

IN MOST DEVELOPING COUNTRIES SUCH AS OURS, WHOLESALE BUSINESSES ARE STILL MANAGED USING A PEN AND PAPER METHODOLOGY EVEN WITH THE ADVENT OF THE AGE OF TECHNOLOGY AND INTERNET. THIS CAN SOMETIMES LEADS TO INEFFICIENT HANDLING OF DAY TO-DAY BUSINESS MATTERS AND CAN ALSO LEAD TO SECURITY THREATS TO THE BUSINESS DUE TO THE PRESENCE OF PHYSICAL RECORDS.

THIS PROJECT AIMS TO DEVELOP SALES MANAGEMENT WEB APPLICATION FOR A WHOLESALE BUSINESS TO IMPROVE THE EFFICIENCY AND SECURITY OF THE BUSINESS. THE LIGHTWEIGHT NATURE OF THE WEB APPLICATION MAKES IT IDEAL FOR SMALL BUSINESSES WHICH CANNOT AFFORD THE EXPENSIVE ALTERNATES PRESENT IN THE MARKET.
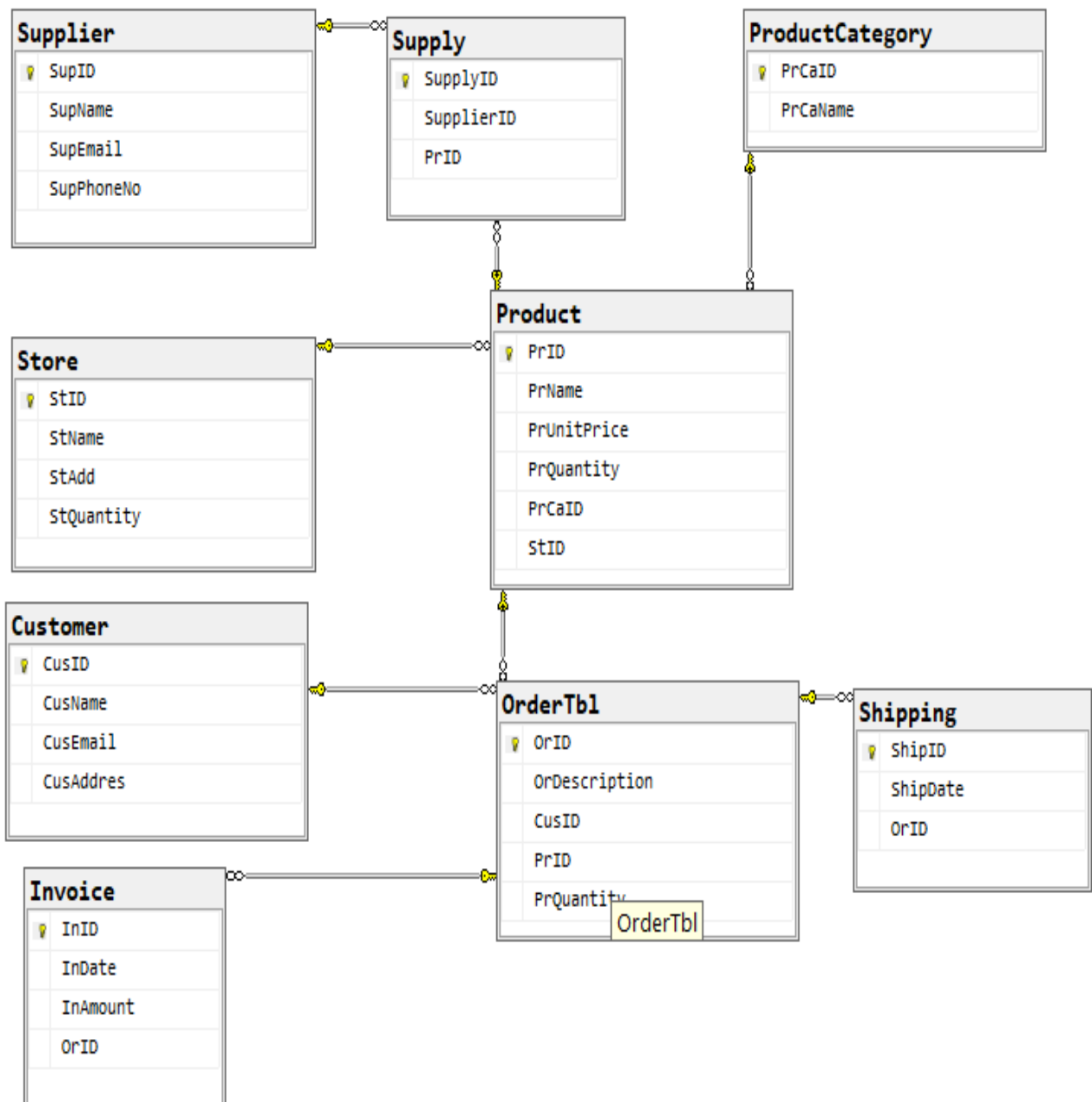
THE WHOLESALE MANAGEMENT SYSTEM (WMS) HAS AN ADMINISTRATOR LOGIN FOR THE WHOLESALER WITH BUILT-IN FEATURES TO KEEP A CHECK ON THE SALES AND THE INVENTORY SO AS TO EFFICIENTLY RUN THE SALES DIVISION. THE CUSTOMERS AND SUPPLIERS OF THE WHOLESALE BUSINESS CAN ALSO BE REGISTERED SO AS TO ENSURE THE SMOOTH FUNCTIONING OF SALES. THERE IS ALSO A CUSTOMER LOGIN OPTION WHICH HELPS THE CUSTOMER IN MONITORING HIS PURCHASES FROM THE WHOLESALER. BY HAVING THE ABOVE FEATURES, THE WMS GREATLY SIMPLIFIES THE DAILY FUNCTIONING OF THE BUSINESS.

SINCE, MANY DEVELOPED COUNTRIES ALREADY USE SUCH SOFTWARE TO SPEED UP DAILY BUSINESS ACTIVITIES; A LOT OF INFORMATION IS AVAILABLE IN THE PUBLIC DOMAIN REGARDING THE DESIRABLE STRUCTURE OF SUCH SOFTWARE. WE HAVE USED THIS INFORMATION TO DESIGN AND MODEL OUR PROJECT.

SOFTWARE NEEDED

BACKEND (DATABASE): MYSQL 2014 OR HIGHER

ENTITY RELATIONAL DIAGRAM

Supplier
- SupID
- SupName
- SupEmail
- SupPhoneNo

Supply
- SupplyID
- SupplierID
- PrID

ProductCategory
- PrCaID
- PrCaName

Store
- StID
- StName
- StAdd
- StQuantity

Product
- PrID
- PrName
- PrUnitPrice
- PrQuantity
- PrCaID
- StID

Customer
- CusID
- CusName
- CusEmail
- CusAddres

OrderTbl
- OrID
- OrDescription
- CusID
- PrID
- PrQuantity

OrderTbl

Shipping
- ShipID
- ShipDate
- OrID

Invoice
- InID
- InDate
- InAmount
- OrID

Short Description to Entities involved ERD

Product:

These are the items that the wholesale business sells, Such as electronics, clothes, furniture or food products.

Customer:

These are the buyers who purchase the products the wholesale business. Customer information may include Contact details, billing and shipping addresses, and order histories.

Supplier:

These are the entities that provide product to the wholesale business to sell. Suppliers may tracked in terms of contact information, product catalogs, and pricing information.

OrderTbl:

These are the sales transaction that TAKES place between wholesale business and customer. Order information may include product quantities, shipping, and billing information, and delivery dates.

Invoices:

These are the bills sent to customers to request payment for product ordered. invoices may be generated automatically by wholesale management system and include itemized lists of products ordered.

Store:

This refers to the stock of products that the wholesale business has on hand. Store levels may need to be tracked in real time to ensure that the products are not oversold or undersold.

SHIPPING:

ONCE AN ORDER IS PLACED, THE WHOLESALE BUSINESS WILL NEED TO ARRANGE FOR SHIPPING AND DELIVERY OF THE PRODUCTS. SHIPPING INFORMATION MAY INCLUDE TRACKING NUMBERS, CARRIER INFORMATION, AND ESTIMATED DELIVERY DATES.

SQL COMMANDS FOR THE ABOVE ERD

--> This command will create Whole sell management database <--

Create database WholeSellManagment   Use WholeSellManagment

--> This command will create Product table   <--

```
Create table Product(PrID int not null,PrName varchar(30),PrUnitPrice
              money, PrQuantity int,
              PrCaID int foreign key references
              ProductCategory (PrCaID) on delete cascade on
              Update cascade)
Alter table Product Add Primary key (PrID)
Alter table Product Add StID int foreign key references Store (StID)
        On delete cascade on update cascade
Insert into Product
        (PrID,PrName,PrUnitPrice,PrQuantity,PrCaID,StID)
values    (1,'Chai',1800,50,4,1),(2,'Juice',400,550,4,1),
          (3,'TV',5000,150,1,1),(4,'Head phone',250,50,1,1)
select * from Product
```

--> This command will create ProductCategory table   <--

create table ProductCategory
          (PrCaID int not null, PrCaName varchar(30))

Alter table ProductCategory Add Primary key (PrCaID)

Insert into ProductCategory

values   (1,'Electronics'),(2,'Clothes'),(3,'Furniture'),(4,'Edable')

select * from ProductCategory

-->   *Join table of Product and category*   **<--**

```sql
Select  Product.PrName,Product.PrUnitPrice,
        ProductCategory.PrCaName
From    Product,ProductCategory
Where   Product.PrCaID = ProductCategory.PrCaID
```

-->   *Creating a Table of Customer which store the data*   **<--**

```sql
Create table Customer
            (CusID int not null primary key, CusName varchar(30),
            CusEmail varchar(30),CusAddres varchar(40))
Insert into Customer
values      (1,'Niaz','momandniazkhan@gmail.com','Nurrar'),
            (2,'Khan','millikahn72@gmail.com','Bannu'),
            (3,'Amjad','amjadkhan34@gmail.com','Mamushkhel'),
            (4,'Salman','salkhanman23@gmail.com','Nurrar'),
            (5,'Hameed','hameedullah45@gmail.com','Kakki')
Select * from Customer
```

-->   *Creating a Table of Supplier whice store the data of*   **<--**

```sql
Create table Supplier
                (SupID int not null primary key,SupName varchar(30),
                SupEmail varchar(40),SupPhoneNo bigint)
Insert into Supplier
values      (2,'Ali','aliShahzor@gmail.com',03161054952),
            (3,'Zohib','zohib234@gmail.com',03161054952),
            (4,'Sher','Shirahzor@gmail.com',03161054952),
            (5,'Fayaz','fayazShahzor@gmail.com',03161054952),
            (6,'Gurbaz','GurbazShahzor@gmail.com',03161054952)
Select * from Supplier
```

-->   *Creating a Table of Supplier whice store the data of*   **<--**

```sql
Create table Supply
            (SupplyID int not null Primary key,SupplierID int not null,
            PrID int not null)
Alter table Supply Add       Foreign key (PrID) references Product(PrID)on delete
```

```
            cascade on update cascade
Alter table Supply
Add         Foreign key (SupplierID) references Supplier(SupID)on delete
            cascade on update cascade
```

-->   *Creating a Table of Order whice store the data of*   <--

```
Create table OrderTbl
            (OrID int not null Primary key, OrDescription varchar(50),
            CusID int not null,PrID int not null,PrQuantity int)
Alter table OrderTbl
Add          Foreign key (CusID) references Customer(CusID) on delete
            cascade on update cascade
Alter table OrderTbl
Add          Foreign key (PrID) references Product(PrID) on delete
            cascade on update cascade
```

-->   *Creating a Table of Order whice store the data of*   <--

```
Create table Store
            (StID int not null Primary key,StName varchar(30),
            StAdd varchar(30),StQuantity int)
Insert into Store
            (StID,StName,StQuantity,StAdd)
Values      (1,'godam',50000,'Nurrar')
```

-->   *A Command which can Delete Whole data base*   <--

```
Create table Invoice
            (InID int not null primary key,InDate date,InAmount
            money,OrID int)
Alter table Invoice
Add         Foreign key (OrID) references OrderTbl(OrID) on delete
            cascade on update cascade
```

-->   *A Commond which can Delete Whole data base*   <--

```
create table Shipping
            (ShipID int  not null primary key, ShipDate date, OrID int
             not null)
```

Alter table Shipping

Add **Foreign key** (**OrID**) **references** OrderTbl(**OrID**) **on delete**

cascade on update cascade

--> *A Commond which can Delete Whole data base* <--

Drop database WholeSellManagment

APPLICATIONS

THERE ARE TWO TYPES OF ACCOUNTS: ADMINISTRATOR AND CUSTOMER.

FEATURES FOR THE ADMINISTRATOR:

· ADD/UPDATE PRODUCT DETAILS BY ACCESSING THEM CATEGORY-WISE

· ADD/UPDATE SUPPLIER DETAILS · ADD/UPDATE CUSTOMER DETAILS

· STOCK MAINTENANCE BY HAVING A LOOK AT THE DEPLETED STOCKS AND SO FORTH.

· VIEW ALL THE TRANSACTIONS TAKEN PLACE IN A SPECIFIED TIME PERIOD.

· ADD A NEW TRANSACTION TO THE SYSTEM AS IT HAPPENS OFFLINE AND STORE THE PAYMENT DETAILS ACCORDINGLY.

· GENERATE A BILL FOR ANY PAST TRANSACTION USING ITS UNIQUE TRANSACTION ID IF REQUIRED.

FEATURES FOR THE CUSTOMER:

· VIEW ALL THE TRANSACTION DONE BY THE LOGGED-IN CUSTOMER FOR A SPECIFIED TIME PERIOD.

· SEARCH THROUGH THE AVAILABLE PRODUCTS CATEGORY-WISE

· GENERATE BILLS FOR ANY PREVIOUS TRANSACTION DONE BY THE CUSTOMER.

CONCLUSION: LIMITATIONS AND DISCUSSION

THIS IS A SMALL PROTOTYPE OF A SALES MANAGEMENT APPLICATION FOR A WHOLESALE BUSINESS. THE LIMITATION OF THE APPLICATION IS THAT IT LACKS ENOUGH FEATURES TO BE IMPLEMENTED IN A REAL LIFE SITUATION.

SUCH AN APPLICATION, IF BUILT WITH PROFESSIONAL EXPERTISE, CAN BE HIGHLY USEFUL COST EFFECTIVE WAY FOR SMALL BUSINESSES TO MANAGE THEMSELVES EFFICIENTLY. A WHOLESALE BUSINESS WORKS COMPLETELY ON THE CONCEPT OF OFFLINE TRANSACTIONS THUS MAKING IT EASY TO MANAGE SUCH AN APPLICATION SINCE IT DOESN'T HAVE TO MANAGE ONLINE TRANSACTIONS, YET AT THE SAME TIME MAKING THE BUSINESS RECORDS MORE SECURE. A LEGAL IMPLEMENTATION OF THIS APPLICATION CAN BE IF THE GOVERNMENT DECIDES TO PROVIDE A WMS APPLICATION TO SMALL BUSINESS OWNERS SO THAT THEIR SALES ARE ON THE OFFICIAL RECORD, THUS MAKING IT DIFFICULT FOR THEM TO EVADE TAXES.

REFERENCES

· DATABASE SYSTEMS, 6TH EDITION, BY RAMEZ ELMASRI AND SHAMKANT BNAVATHE.

· 'WHOLESALE MANAGEMENT SYSTEM' BY LEE YEE FHONG, NATIONAL TECHNICAL UNIVERSITY COLLEGE OF MALAYSIA, 2004

· 'DESIGNING A LOGICAL DATA MODEL FOR A SALES AND INVENTORY MANAGEMENT SYSTEM' BY HARI KRISHNA MAHAT.