DATABASE II ASSIGNMENT

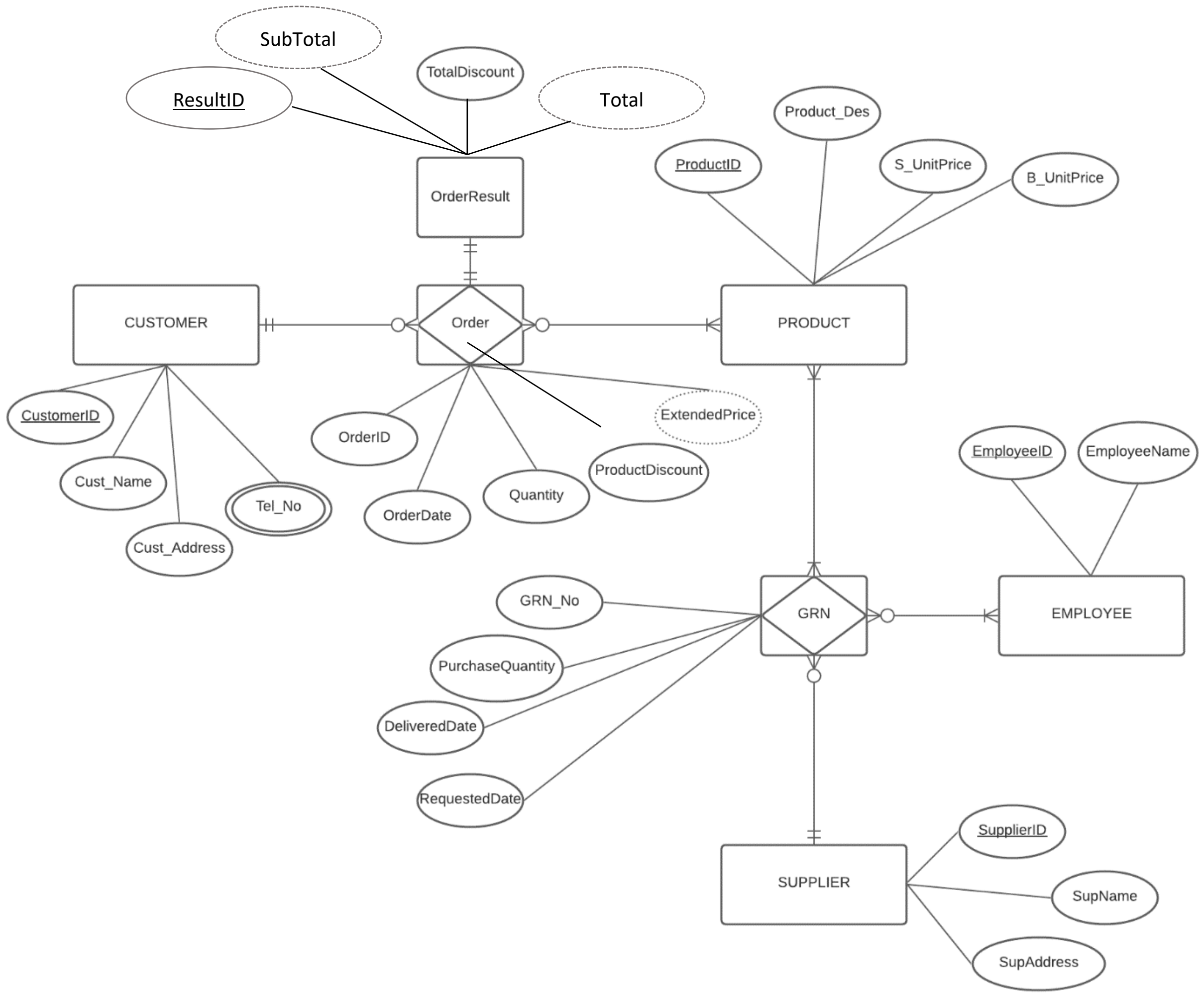
TithiraWithanaarachchi(Batch 4)

Lecturer-Mahesha Thejani

Contents

1)Task1

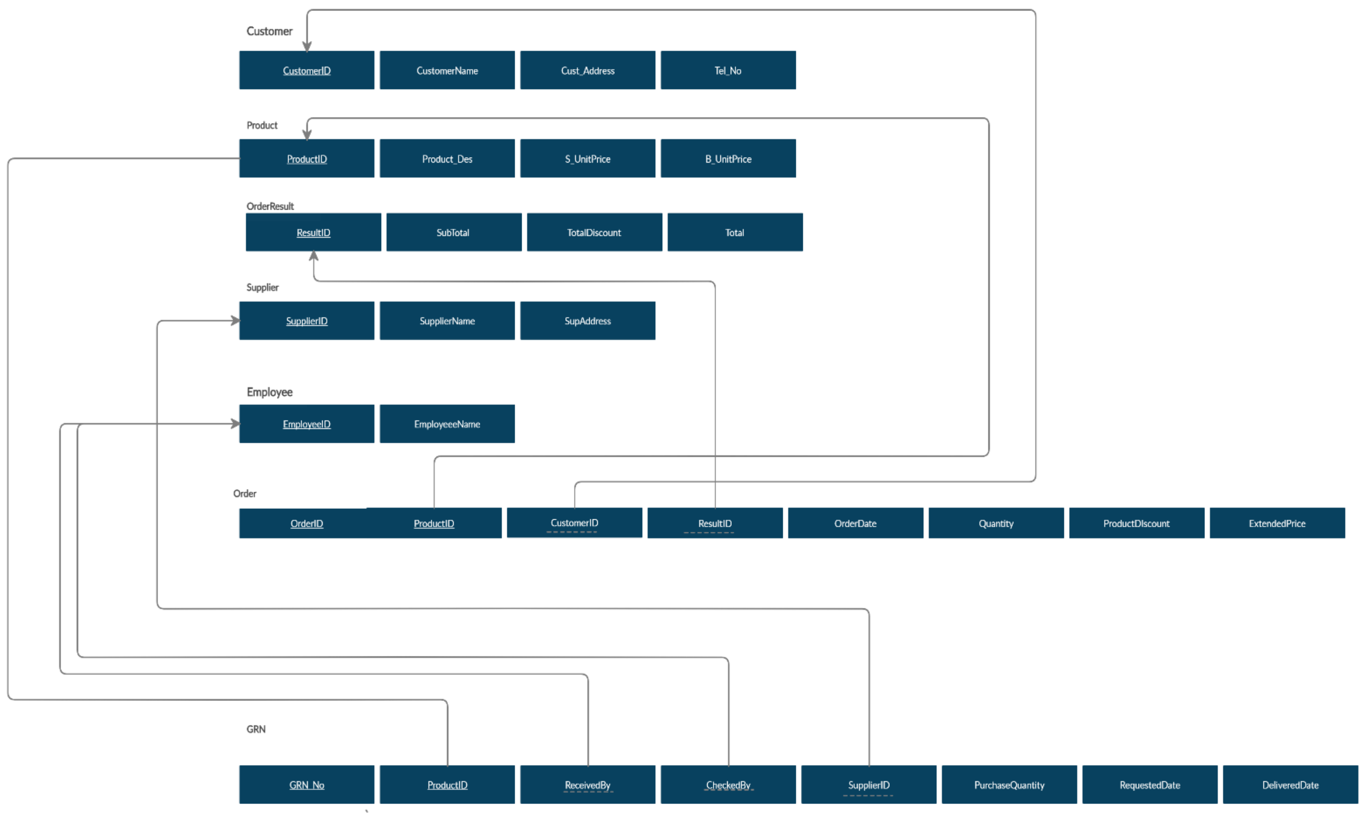
1.1-ER Diagram



<u>Assumptions</u>

1. "Tel_No" has been considered as an attribute of the Customer table which is a multivalued attribute to record the telephone number of the customer.
2. The Company is not buying the same product from different suppliers.
3. Employee has been considered as a separate entity which involves in receiving and checking the Orders purchased by the company.
4. Order Result has been considered as a separate Entity which records the Final Output of the order with the discounts as well.

1.2-Relational Schema



1.3-Sql Query

```
CREATE DATABASE abx_furniture;

USE abx_furniture;

CREATE TABLE Product
(
productID INT NOT NULL,
Prod_Des VARCHAR(50),
S_UnitPrice INT,
B_UnitPrice INT,
CONSTRAINT pk_pid PRIMARY KEY(productID)
);
DROP TABLE Product;

CREATE TABLE Customer
(
CustomerID VARCHAR(10) NOT NULL,
CustomerName VARCHAR(40),
Cust_Address VARCHAR(40),
Tel_No VARCHAR(10),
CONSTRAINT pk_cid PRIMARY KEY(CustomerID)
);

DROP TABLE Customer;




CREATE TABLE Employee
(
EmployeeID VARCHAR(10) NOT NULL,
EmployeeName VARCHAR(30),
CONSTRAINT pk_eid PRIMARY KEY(EmployeeID)
);

DROP TABLE Employee;


CREATE TABLE Supplier
(
SupplierID VARCHAR(10) NOT NULL,
Sup_Name VARCHAR(30),
```

```sql
Sup_Address VARCHAR(50)
CONSTRAINT pk_sid PRIMARY KEY(SupplierID)
);

DROP TABLE Supplier;




CREATE TABLE Order_details
(
OrderID VARCHAR(10) NOT NULL,
ProductID INT NOT NULL,
CustomerID VARCHAR(10) NOT NULL,
OrderDate DATE,
Quantity INT,
Prod_Discount INT,
CONSTRAINT pk_oid_prod PRIMARY KEY(OrderID, ProductID),
CONSTRAINT fk_pid FOREIGN KEY(productID) REFERENCES
Product(productID),
CONSTRAINT fk_cid FOREIGN KEY(customerID) REFERENCES
Customer(customerID),
);

DROP TABLE Order_details;





CREATE TABLE GRN_details
(
GRN_No VARCHAR(10) NOT NULL,
productID INT NOT NULL,
ReceivedBy VARCHAR(10) NOT NULL,
CheckedBy VARCHAR(10) NOT NULL,
SupplierID VARCHAR(10) NOT NULL,
purchase_quty INT,
Requested_Date DATETIME,
Delivered_Date DATETIME,
CONSTRAINT pk_grnNO_pid PRIMARY KEY(GRN_No, productID),
CONSTRAINT fk_pid1 FOREIGN KEY(productID) REFERENCES
Product(productID),
CONSTRAINT fk_r1 FOREIGN KEY(ReceivedBy) REFERENCES
Employee(EmployeeID),
CONSTRAINT fk_c1 FOREIGN KEY(CheckedBy) REFERENCES
Employee(EmployeeID),
CONSTRAINT fk_sid FOREIGN KEY(SupplierID) REFERENCES
Supplier(SupplierID)
);

DROP TABLE GRN_details;

CREATE TABLE Admin
(
adminid INT NOT NULL,
adminName VARCHAR(30),
CONSTRAINT pk_aid PRIMARY KEY(adminid)
);

DROP TABLE Admin;


CREATE TABLE Login
(
password VARCHAR (20)NOT NULL,
username VARCHAR(20) NOT NULL,
adid VARCHAR(20),
CONSTRAINT pk_usrname PRIMARY KEY (username)
);
DROP TABLE Login;
```

1.4-Using IDE

An IDE is a software which provides functionality of multiple programming processes within a single source.

Common features of an IDE,

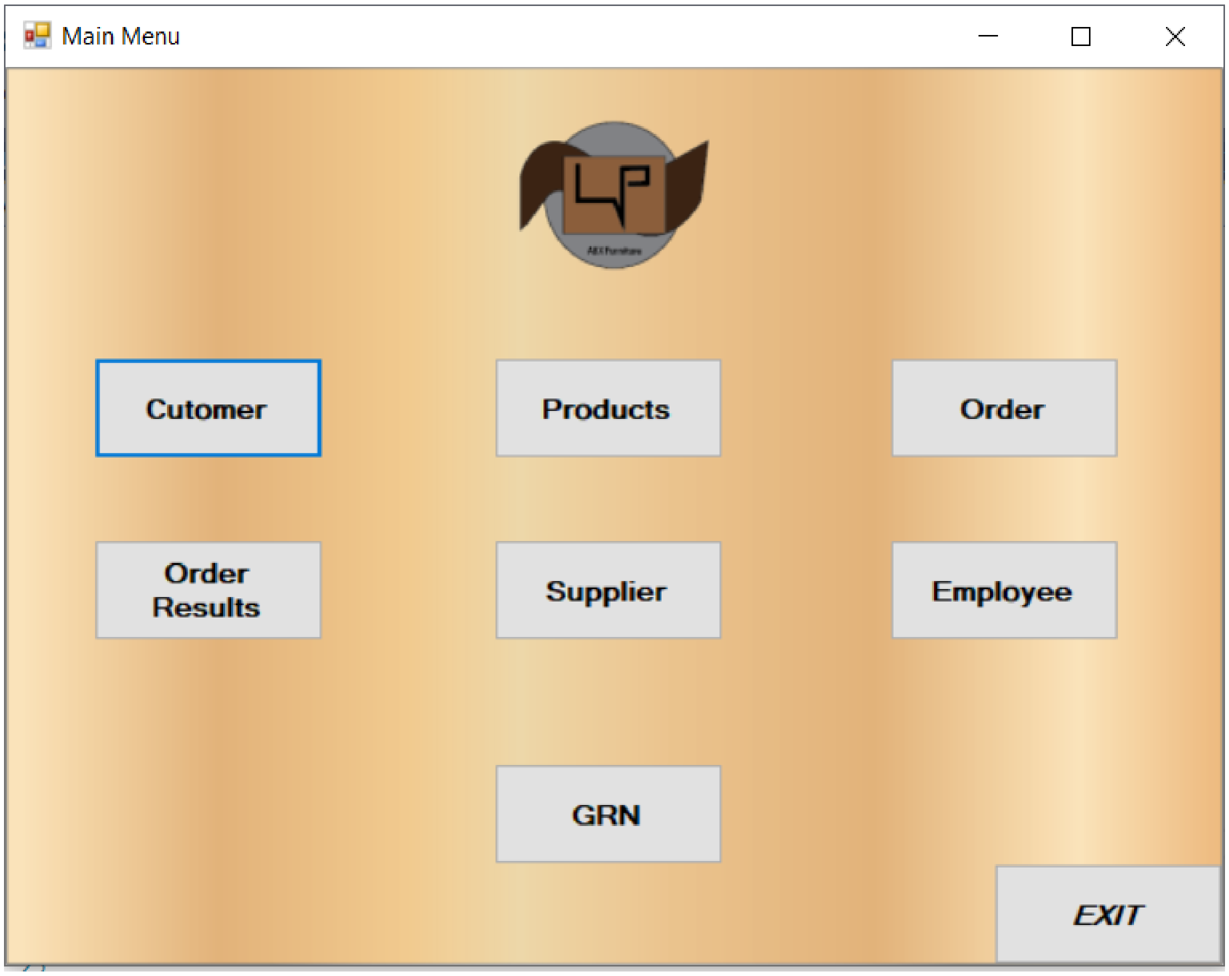
- Text Editor,

- Debugger,
- Compiler,
- Code Completion,
- Programming language support,
- Integrations.

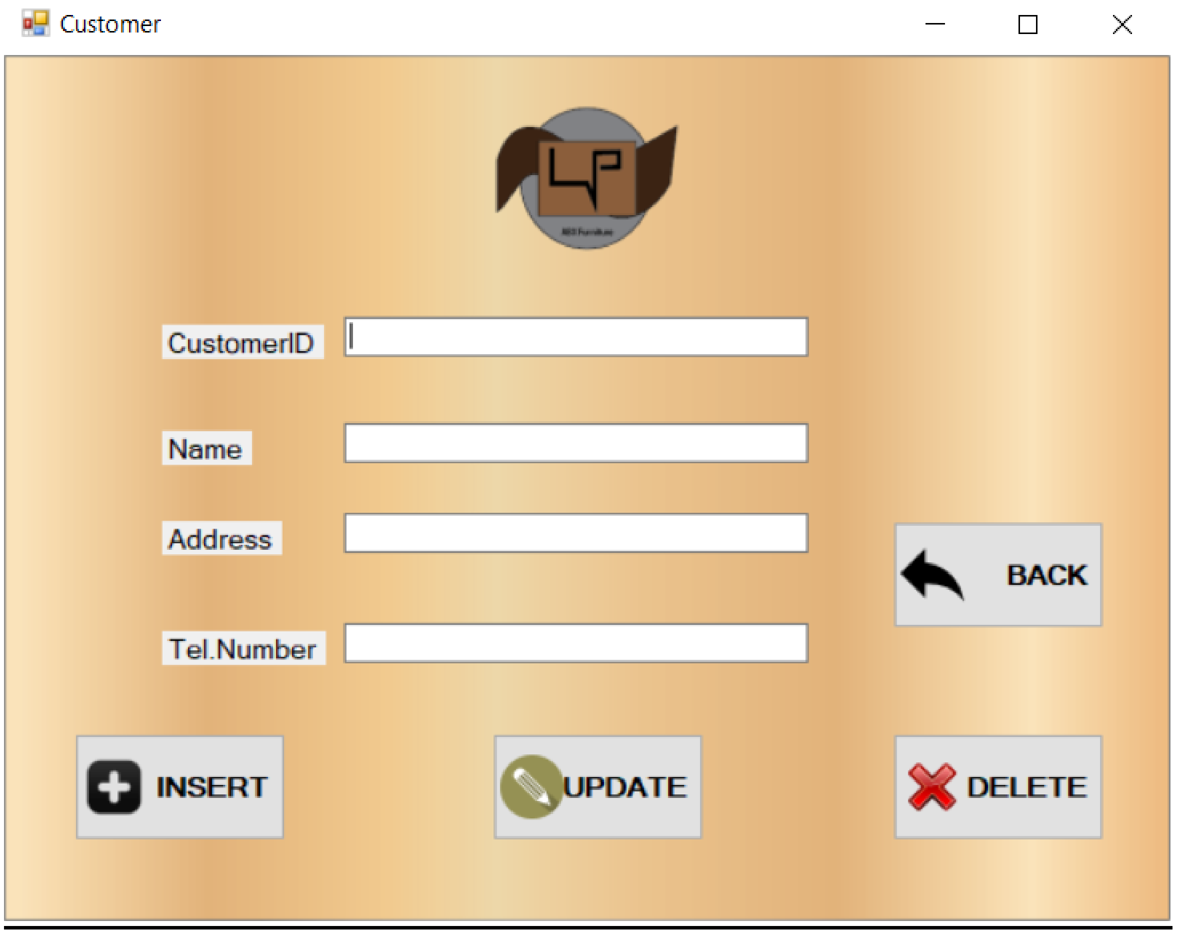
Main Benefits of an IDE,

- Works as a single environment for almost all the functions that are needed for a developer to build a program.
- Code completion functionality improves the smoothness of the workflow.
- Check errors automatically.
- User satisfaction [1].

IDE-Visual Studio



This is the Main Menu of the System which will guide the user to the respective interface of the entity which the user expect to manipulate data in, by clicking the relevant button.



This is the interface for the customer entity which will allow the user to,

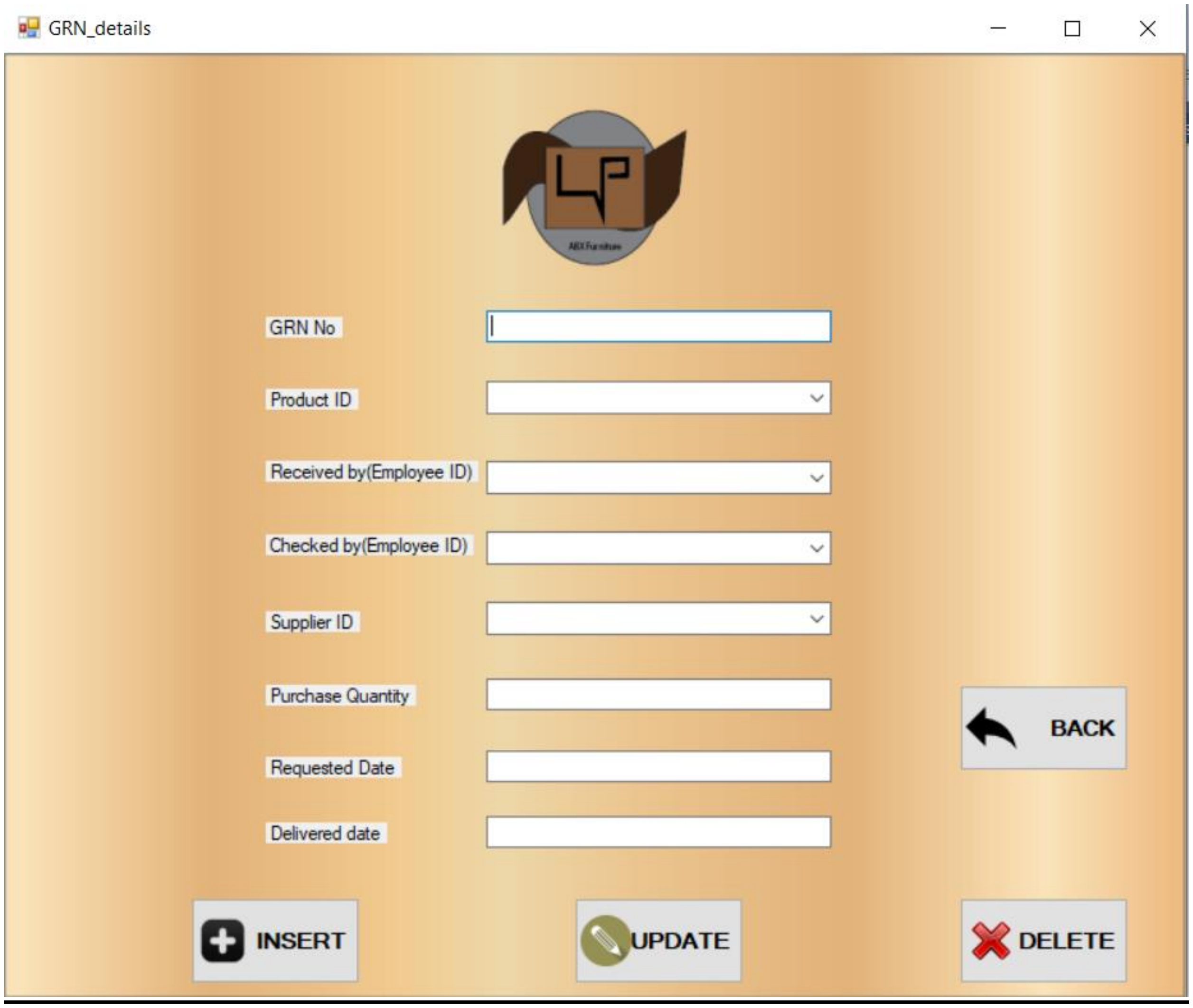
- Insert,
- Update,
- Delete

data in that Customer entity.

Other interfaces of entities such as,

- Employee
- Product
- Supplier
- Order Result

will have the same structure and the functionalities as the interface of customer entity.



GRN No
Product ID
Received by(Employee ID)
Checked by(Employee ID)
Supplier ID
Purchase Quantity
Requested Date
Delivered date

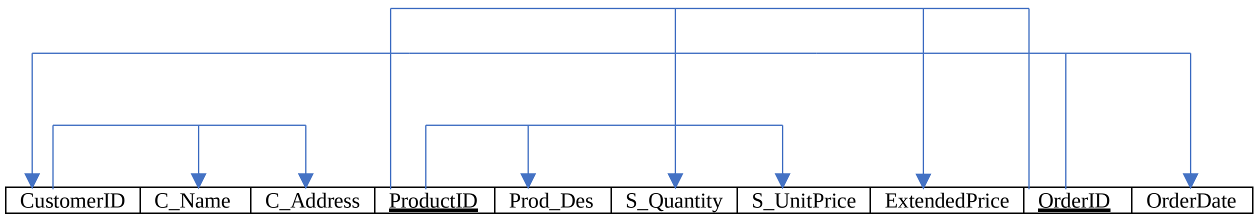
This is the interface of GRN details (Goods Received Note details) which will allow the same functionalities as in the customer interface and it also it provides the option of selecting the ID's of the relevant parties which is related to the entity.

Interface of Order details is having a similar structure and the same functionalities as of the interface for GRN details.

1.5-Normalization

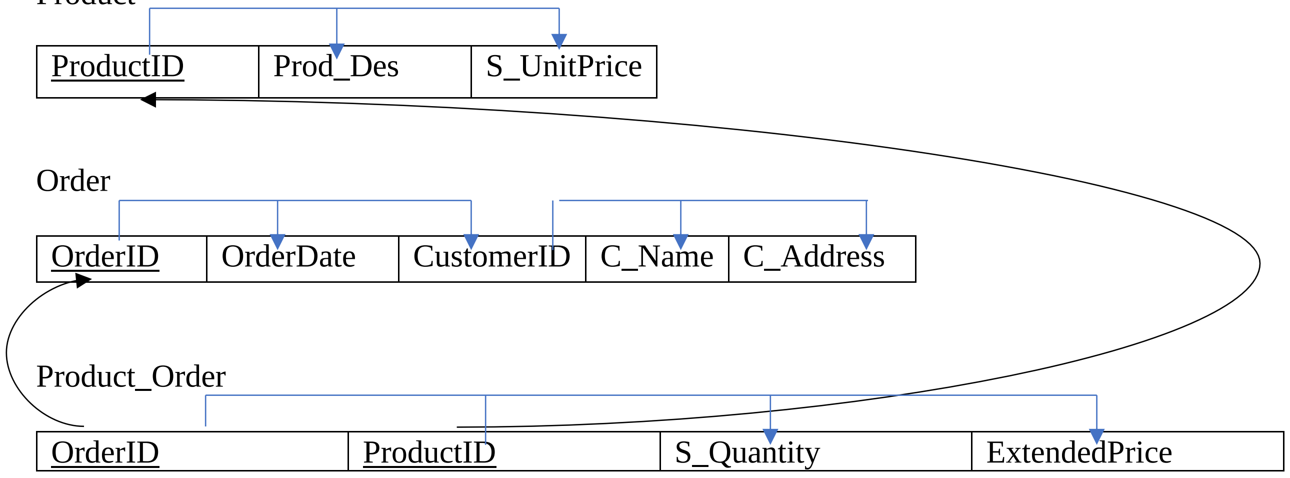
Customer Invoice

1st Normal Form -

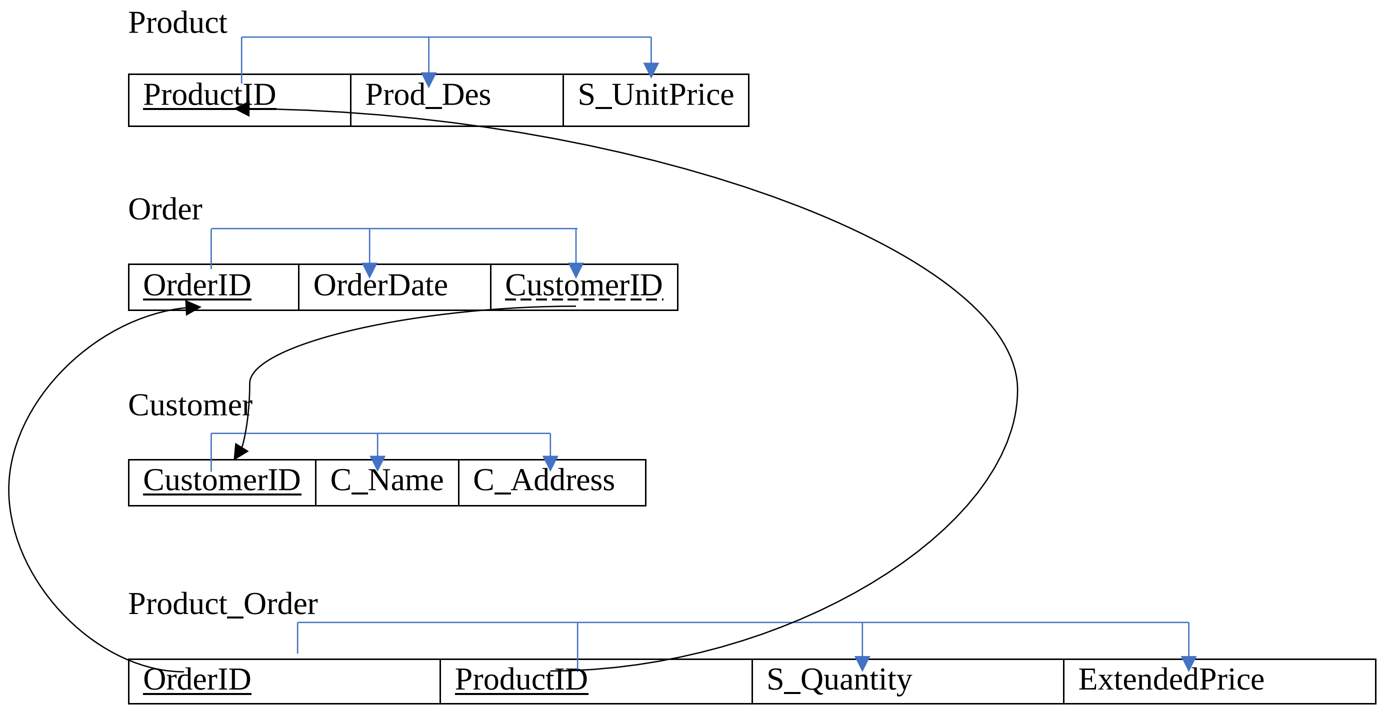


2nd Normal Form-

Product



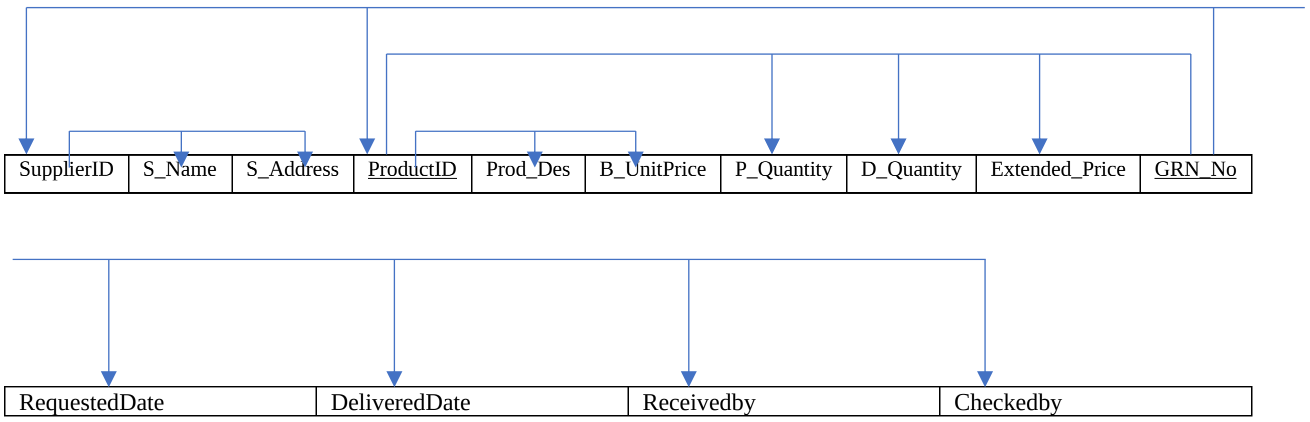
3rd Normal Form-



Since there are no prime attributes which depends on a non-prime attribute, above set of tables satisfies the Boyce Codd Normal form as well.

Goods Received Note

1st Normal Form

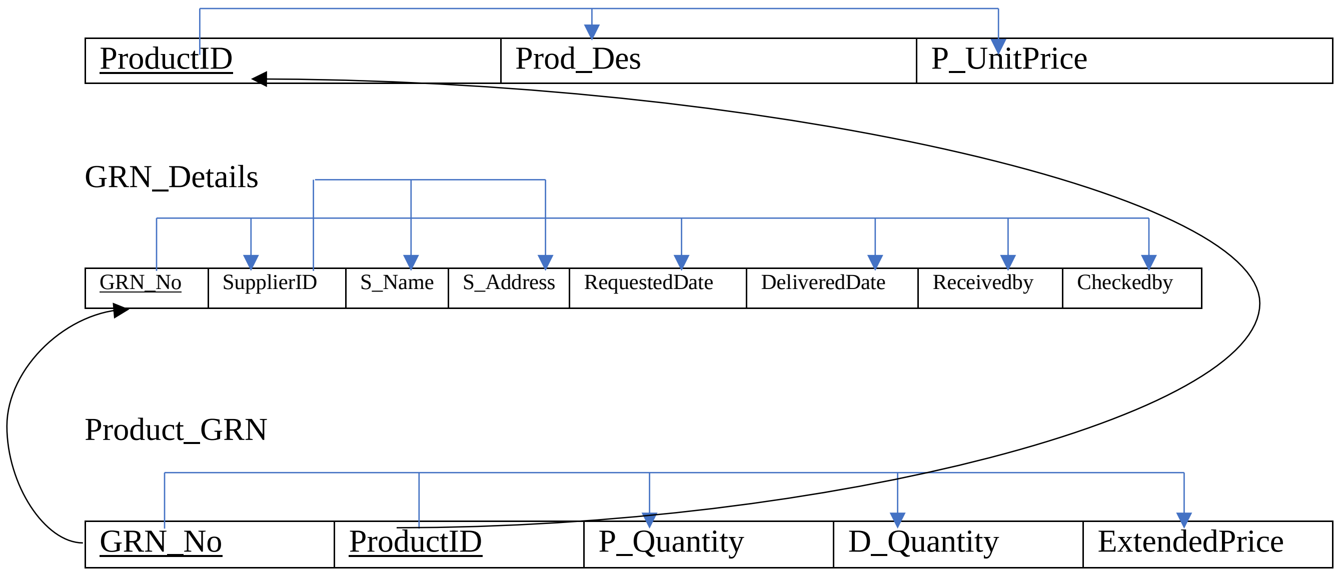


2nd Normal Form-

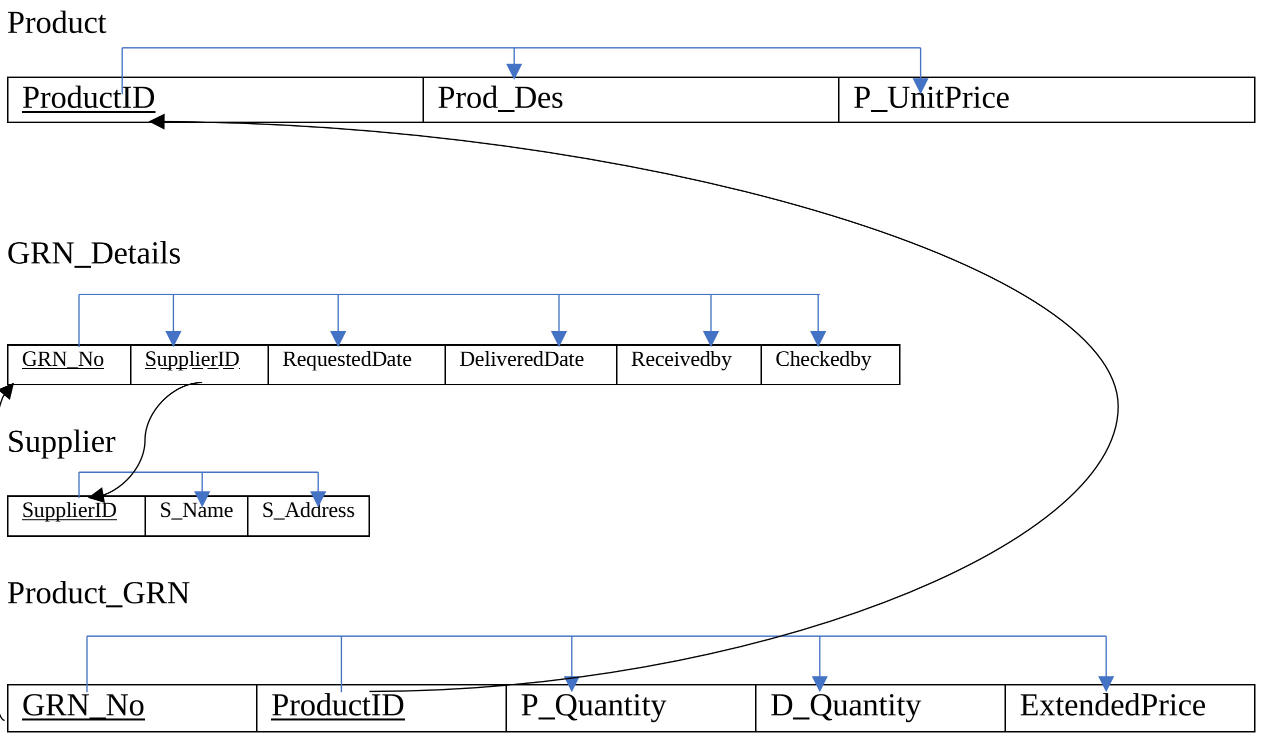
Product

GRN_Details

Product_GRN



3rd Normal Form-



Since there are no prime attributes which depends on a non-prime attribute, above set of tables satisfies the Boyce Codd Normal form as well.

2) Manipulation Queries

2.1 Relational algebra

2.1.1-

$\sigma[(\ \pi_{p.productID}\ (\ \sigma_{p.prod\_Des='Maple\ Wood\ Writers\ Table'}(product,p)))\ \bowtie_{p.pruductID=o.productID}\ (\rho(Order,o)\ )\bowtie_{o.customerID\ =c.customerID}\ (\rho(Customer,\ c)\ )]$
[2]

OrderID	ProductID	OrderDate	Quantity	Prod_Discount	CustomerID	CustomerName	CustomerAddress	Tel_NO
O1	123	2020-12-12	3	122	C12	Mr.Dasun Perera		
O99	123	2021-2-2	1	234	C12	Mr.Dasun Perera		

2.1.2-

σ [$\pi_{p.productID}$ ($\sigma_{p.prod\_Des='Red\ Wood\ Chairs'}$ ρ(product,p)) $\bowtie_{p.pruductID = g.productID}$ (σ $_{g.DeliveredDate>30/09/2020 \text{ AND } g.DeliveredDate<01/11/2020}$ ρ(GRN_Details,g)) $\bowtie_{g..SupplierID = s.SupplierID}$(ρ(Supplier, s))] [2]

GRN_No	ProductID	ReceivedBy	CheckedBy	Purchase_qty	Requested_Date	DeliveredDate	SupplierID	Sup_Name	Sup_Address

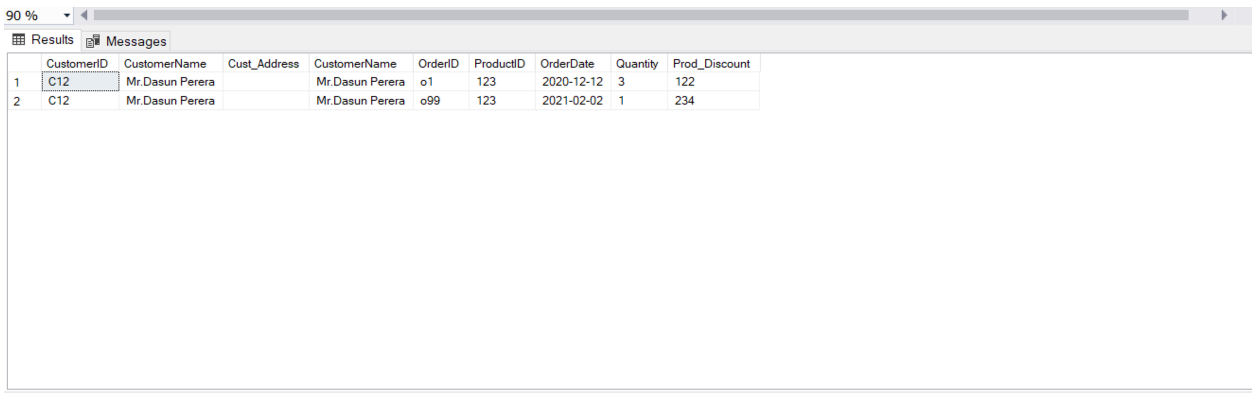
Non of data satisfies the condition.

2.2 Sql Queries

2.2.1-

```
CREATE PROCEDURE searchbyprodName @prodName VARCHAR(50)
AS
SELECT c.CustomerID, c.CustomerName,c.Cust_Address, c.CustomerName,
o.OrderID,o.ProductID, o.OrderDate, o.Quantity, o.Prod_Discount
FROM Customer c, Order_details o, Product p
WHERE c.CustomerID=o.CustomerID AND o.ProductID=p.productID AND
p.Prod_Des=@prodName;

EXEC searchbyprodName @prodName='Maple Wood Writer Tables';
```

Output-

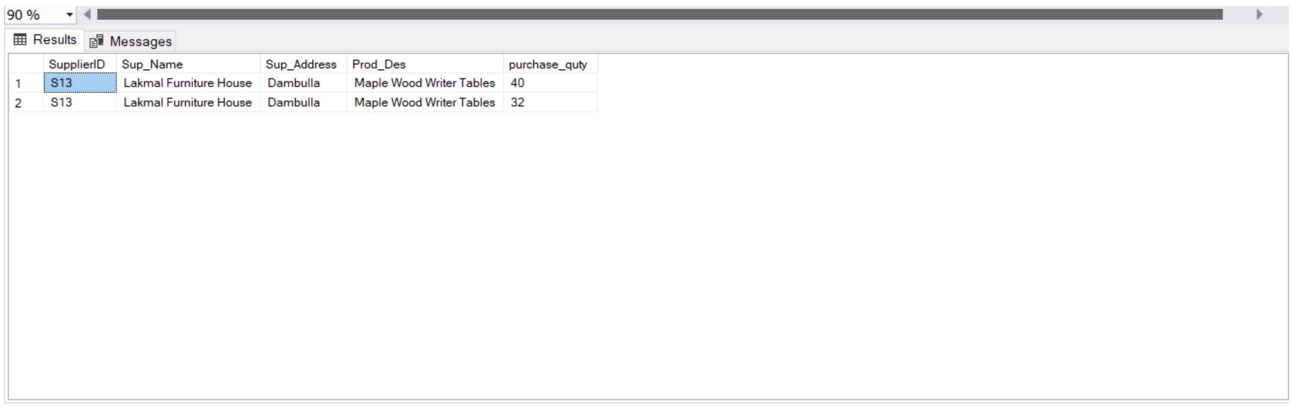


	CustomerID	CustomerName	Cust_Address	CustomerName	OrderID	ProductID	OrderDate	Quantity	Prod_Discount
1	C12	Mr.Dasun Perera		Mr.Dasun Perera	o1	123	2020-12-12	3	122
2	C12	Mr.Dasun Perera		Mr.Dasun Perera	o99	123	2021-02-02	1	234

2.2.2-

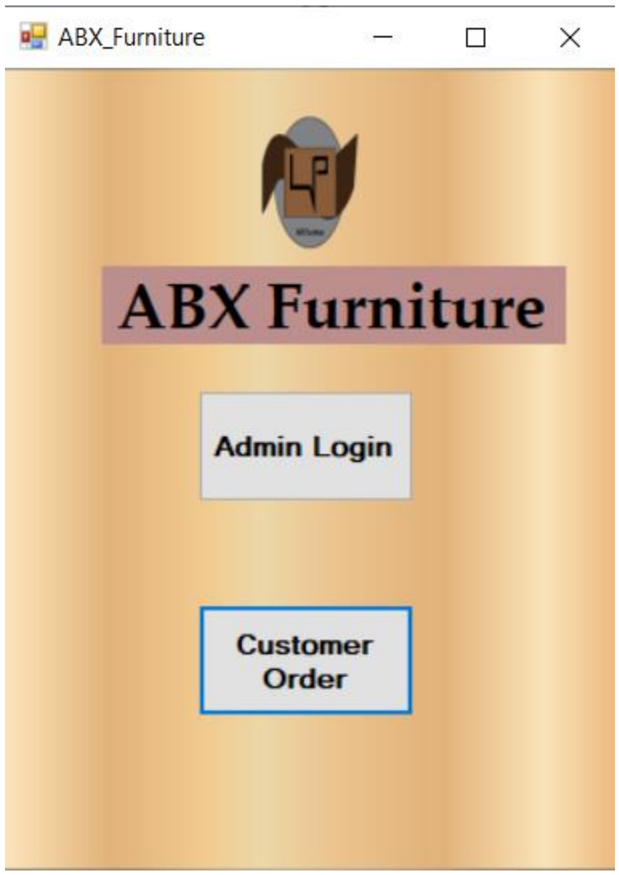
```
CREATE VIEW Supplying_details
AS
SELECT TOP (100) PERCENT s.SupplierID,
s.Sup_Name,s.Sup_Address,p.Prod_Des,g.purchase_quty
FROM Supplier s
INNER JOIN GRN_details g ON s.SupplierID=g.SupplierID
INNER JOIN Product p ON g.productID=p.productID
WHERE g.Delivered_Date > '2020-09-30' AND Delivered_Date < '2020-11-
01'
ORDER BY g.Delivered_Date;


SELECT * FROM Supplying_details;
```

Output-



	SupplierID	Sup_Name	Sup_Address	Prod_Des	purchase_quty
1	S13	Lakmal Furniture House	Dambulla	Maple Wood Writer Tables	40
2	S13	Lakmal Furniture House	Dambulla	Maple Wood Writer Tables	32

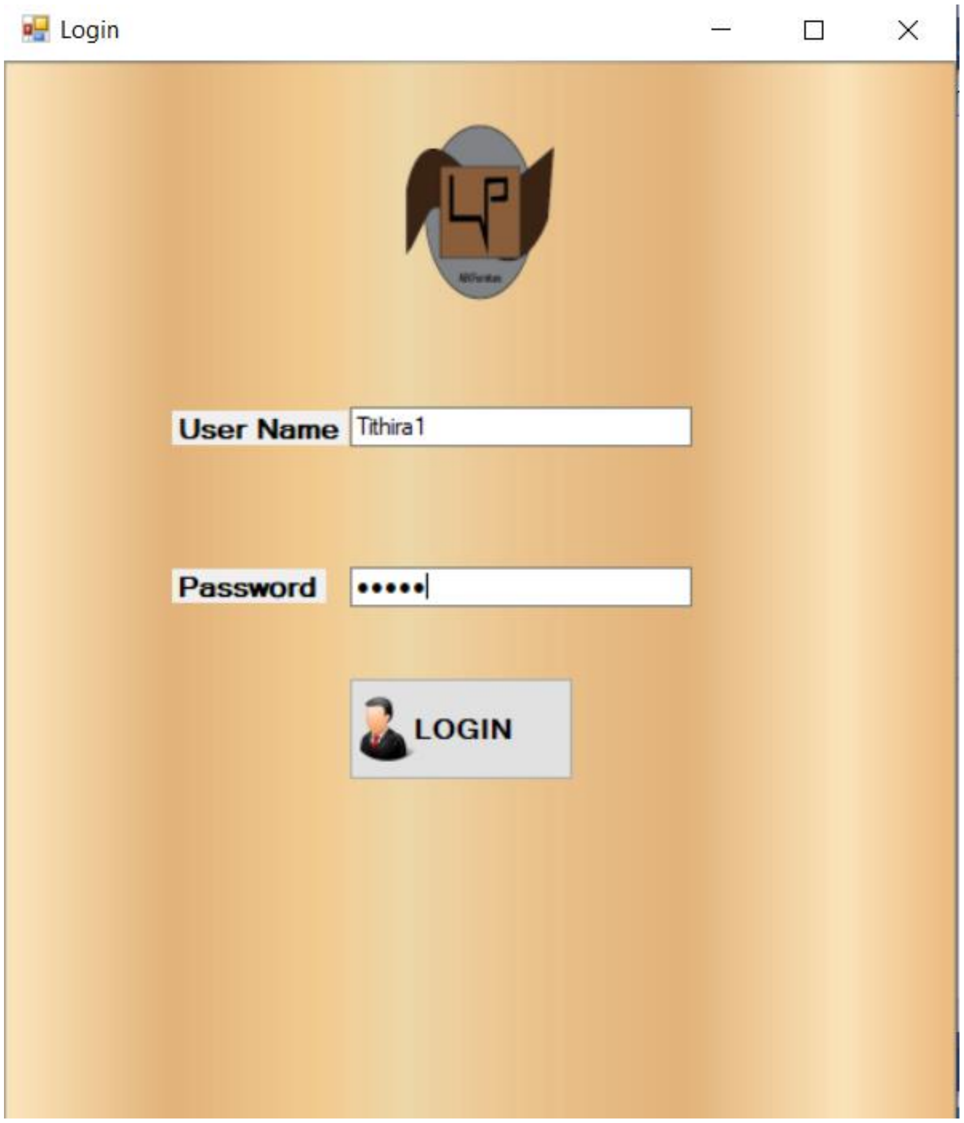
3) Implementing System



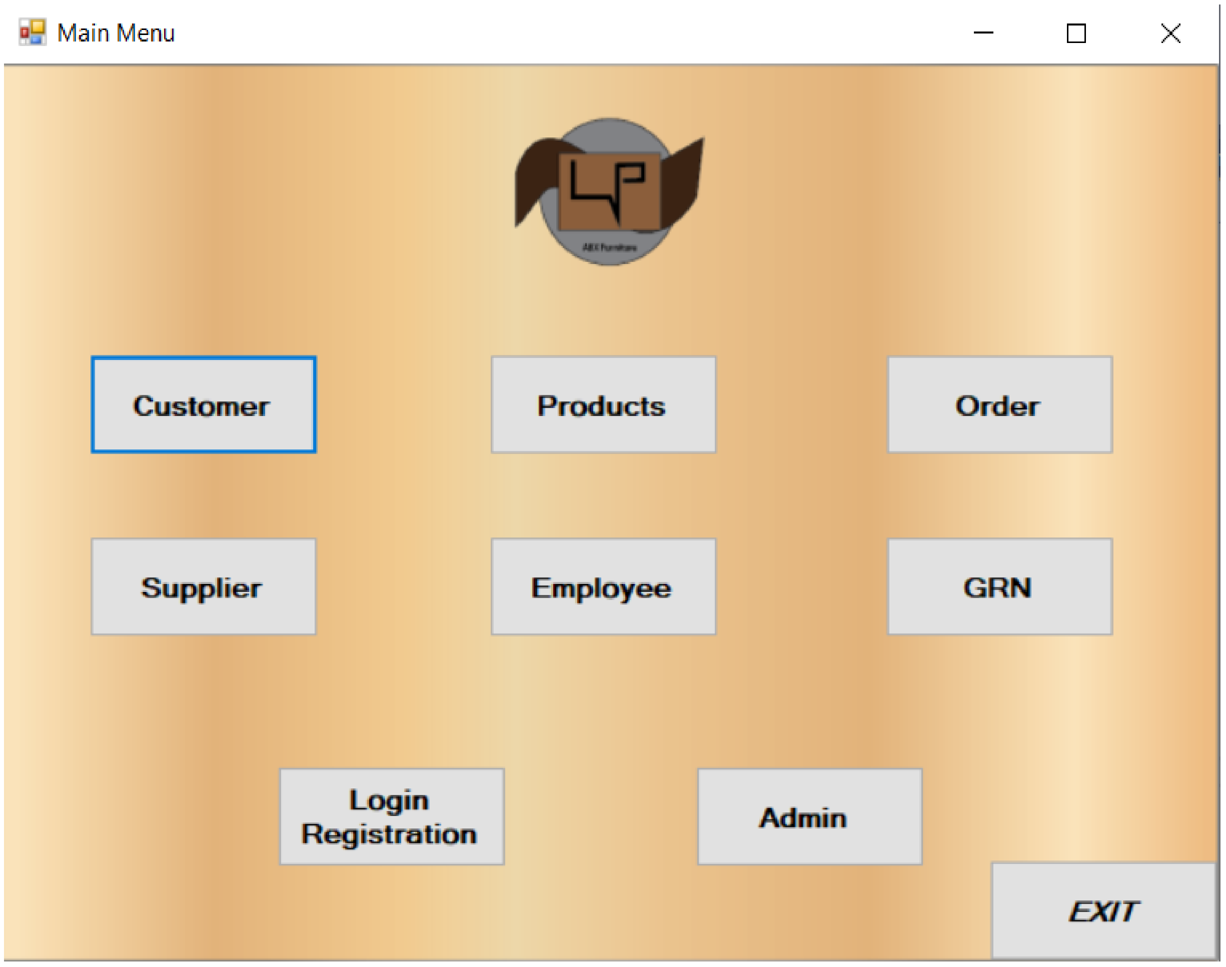
This is the Interface which Allows the user to select whether to Login as and Admin or to place an order as a customer. If the user is an admin, he/she can click the Admin button and access the login interface in order to logging in to the system by providing valid credentials.

If the user is a customer to the company, they can click the customer order button and access the Interface for placing orders.

AdminLogin Interface-



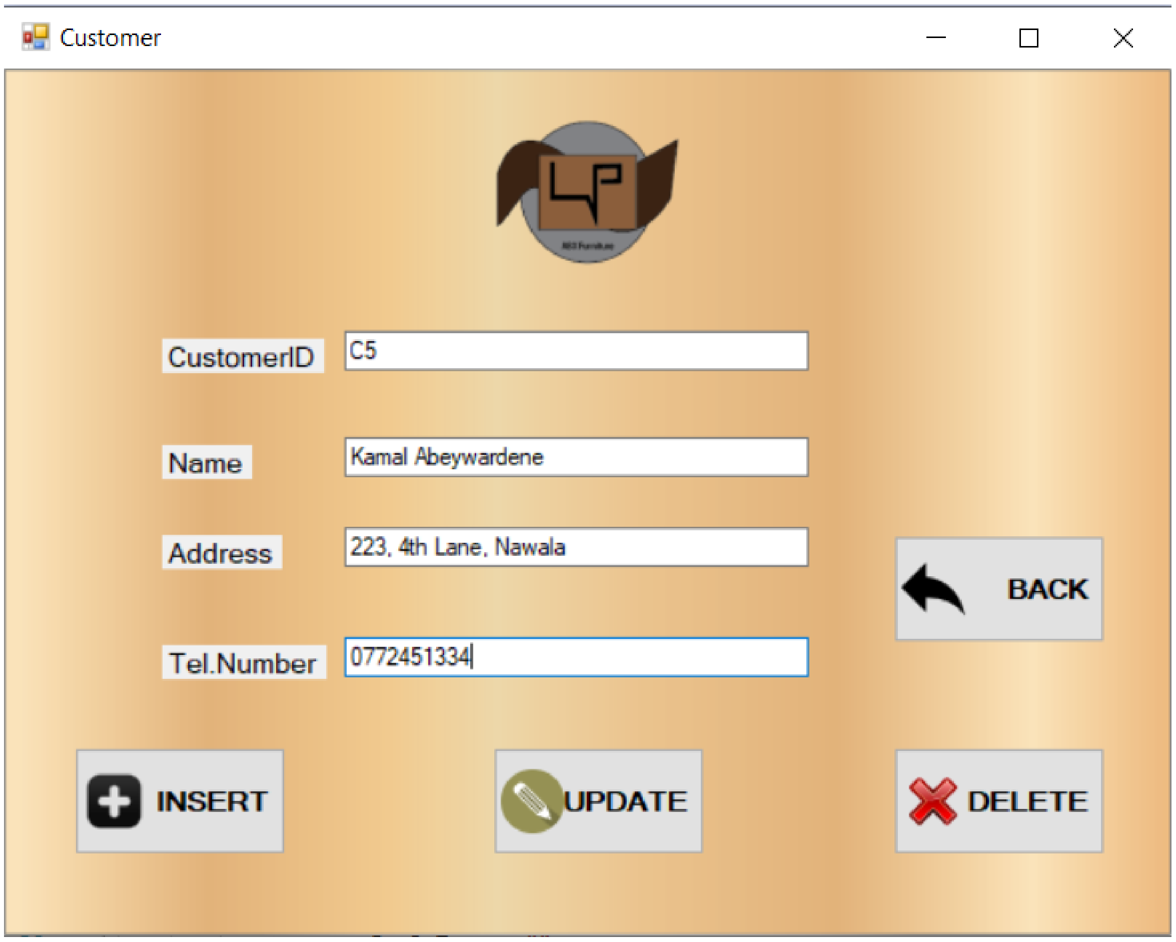
This is the interface used for logging in to the ABX Furniture Information Management System by the Admin. When the user enters valid credentials in the relevant spaces provided, he/she will be directed to the Main Menu of the System.



Main Menu-

This is the interface which allows the user to access each and every entity that they need to manipulate data in. By clicking the relevant button to the entity, the user will be directed to the interface which enables user to manipulate data in that particular entity.

1) Customer Interface-

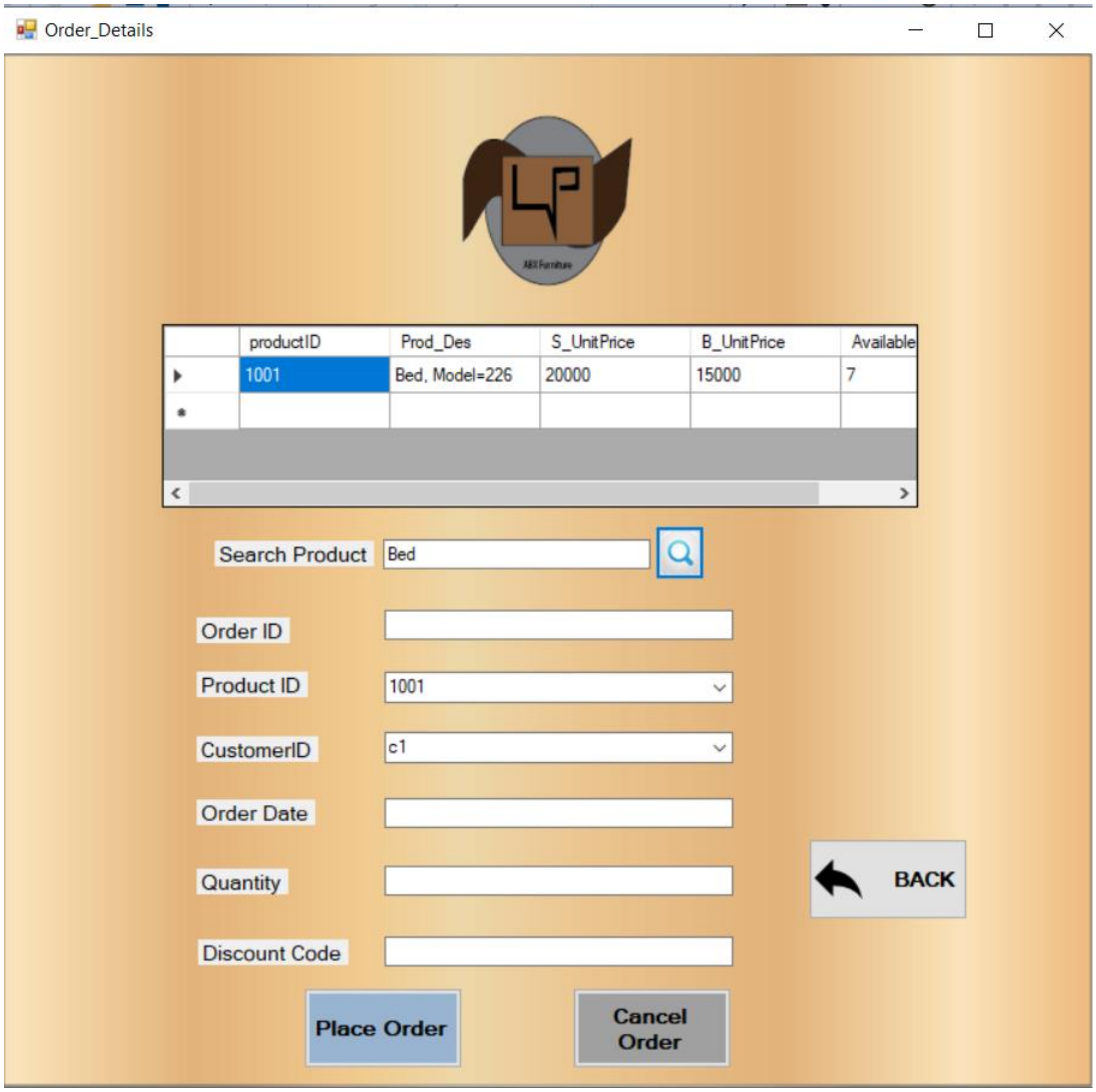

Customer

CustomerID	C5
Name	Kamal Abeywardene
Address	223, 4th Lane, Nawala
Tel.Number	0772451334

This is the interface used for manipulation of data in the customer entity. This will allow the user to Insert, Update and Delete Data After entering valid data for that particular operation.

This interface is the sample structure of other interfaces related to entities such as,

- Product.
- Supplier.
- Employee.
- Login registration.
- Admin Registration.



	productID	Prod_Des	S_UnitPrice	B_UnitPrice	Available
▸	1001	Bed, Model=226	20000	15000	7
*					

Search Product: Bed

Order ID:

Product ID: 1001

CustomerID: c1

Order Date:

Quantity:

Discount Code:

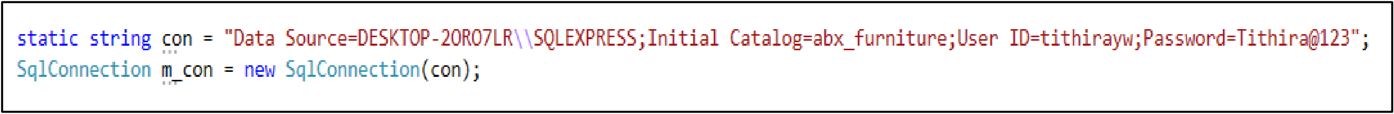
This is the interface appears when the user chooses the option of placing an order from the Main interface.This interface is a public interface.This will allow the user to search products by its name and search the product details before placing the order.

```
public DataTable searchLogin(string usrnm, string pswrd) //Method for
searching Login a record by  password and username
{
    try
    {
        string sql = "Select count(*) from login where username = '" +
        usrnm + "' and password = '" + pswrd + "'";

        SqlDataAdapter Adapt = new SqlDataAdapter(sql, m_con);

        DataTable dt = new DataTable();

        Adapt.Fill(dt);

        return dt;
    }
    finally
    {
        m_con.Close();
    }
  }
```

Shown above is the coding written in the LoginDao class in order to select a record in the Login table according to the object passed through the user interface with the intention of validating the entered data.

```
private void button1_Click(object sender, EventArgs e)
{
    try
    {
        DataTable dt = ldao.searchLogin((textBox1.Text),
        (textBox2.Text));
        if (dt.Rows[0][0].ToString() == "1") //Check whether the
                                             entered data is available
                                             in the login Registration
                                              Table
        {
            this.Hide();


            Form1 ob = new Form1();
            ob.Show();//Display the main menu
            MessageBox.Show("Login Successful", "ABX Furniture");

        }
        else
        {
            MessageBox.Show("Invalid Access", "ABX Furniture");
        }
    }
    catch
    {
        m_con.Close();
    }
/*Reference - https://www.c-sharpcorner.com/UploadFile/9582c9/login-
form-with-sql-in-C-Sharp/ - (Online website)*/



}
```

Shown above is the coding written in the Login button of Login interface.It will check whether the entered data by the user in order to login to the system is valid or not.If the entered data is valid it will direct the user to the main menu of the system.Otherwise it will not allow the user to login by displaying an error message.



```
static string con = "Data Source=DESKTOP-20RO7LR\\SQLEXPRESS;Initial Catalog=abx_furniture;User ID=tithirayw;Password=Tithira@123";
SqlConnection m_con = new SqlConnection(con);
```

This is the code which is used for connecting the system application to the database.

```
public void insertnewcustomer(Customerc cobj)//method for inserting
new records.
{
    try
    {
        string sql= "insert into customer(CustomerID,customername,
        cust_address,tel_no)values('"+cobj.getid()+"',
        '"+cobj.getname()+"','"+cobj.getaddress()+"', '"+
        cobj.gettelNO()+"')";//sql query to insert new records to the
        customer table

        SqlCommand cmd = new SqlCommand(sql, m_con);//sql command to
        connect to the database

        m_con.Open();//openning the pathway to the database

        cmd.ExecuteNonQuery();
    }
    finally
    {
        m_con.Close();//closing the pathway to the database
    }
}
```

This is the public void method written in CustomerDao class in order to insert data to the customer table in the database.

Alll the other interfaces which contains Inert function contains an insert method in the Dao class which has the same structure as above.

```
public void deleteCustomerbyID(string id)//Method for deleting records
from the database table
{
    try
    {
        string sql = "delete from customer where customerid='" + id +
        "'";//sql quey to delete records from the database

        SqlCommand cmd = new SqlCommand(sql, m_con);//sql command to
        connect to the database

        m_con.Open();//openning pathway to the database

        cmd.ExecuteNonQuery();

    }
    finally
    {
        m_con.Close();//closing the pathway of the database
    }
}
```

This is the public void method written in CustomerDao class in order to Delete data in the customer table.

Alll the other interfaces which contains delete function contains an delete method in the Dao class which has the same structure as above.

```
public void updatecustomer(Customerc cobj)//method to update data of
the customer table
{
    try
    {
        string sql = "update customer set customername=('" +
        cobj.getname() + "'),cust_address= ('" + cobj.getaddress() +
        "'), tel_no=('" + cobj.gettelNO() + "') where customerid='" +
        cobj.getid() + "'";//sql query to delete data

        SqlCommand cmd = new SqlCommand(sql, m_con);//sql command to
        connect to the database

        m_con.Open();//openning the pathway to the database

        cmd.ExecuteNonQuery();

    }
    finally
    {
        m_con.Close();//closing the pathway to the database
    }
}
```

This is the public void method written in CustomerDao class in order to Update data in the customer table.

All the other interfaces which contains Update function contains an Update method in the Dao class which has the same structure as above.

```
private void button1_Click(object sender, EventArgs e)//button click
to insert data to the database
{
    try
    {
        Customerc cob = new Customerc((textBox1.Text),
       (textBox2.Text), (textBox3.Text), (textBox4.Text));

        cdao.insertnewcustomer(cob);

        MessageBox.Show("Recorded Successfully", "Customer
        Registration");

        textBox1.Text = "";
        textBox2.Text = "";
        textBox3.Text = "";
        textBox4.Text = "";

    }
    catch
    {
        MessageBox.Show("Error", "Customer Registration");

    }
}
```

The above code is written for the insert button of the Customer Registration interface to pass the inserted values in the textbox as a Customer object to the database to record it in the table.

All the other interfaces which contains Inert function contains a code which has the same structure as above code.

```
private void button3_Click(object sender, EventArgs e)//button click
to delete data from the database
{
    try
    {
        cdao.deleteCustomerbyID(textBox1.Text);

        MessageBox.Show("Deleted Successfully", "Customer
        Registration");

        textBox1.Text = "";
        textBox2.Text = "";
        textBox3.Text = "";
        textBox4.Text = "";
    }
    catch
    {
        MessageBox.Show("Error", "Customer Registration");
    }
}
```

The above code is written in the delete button of the interface of Customer to pass the value in textBox1 to the delete method in CustomerDAO class.

All the other interfaces which contains Delete function contains a code which has the same structure as above code.

```
private void button2_Click(object sender, EventArgs e)//button click
to update data in the database
{
    try
    {
        Customerc cob = new Customerc((textBox1.Text),
        (textBox2.Text), (textBox3.Text), (textBox4.Text));

        cdao.updatecustomer(cob);

        MessageBox.Show("Updated Successfully", "Customer
        Registration");

        textBox1.Text = "";
        textBox2.Text = "";
        textBox3.Text = "";
        textBox4.Text = "";
    }
    catch
    {
        MessageBox.Show("Error", "Customer Registration");

    }


}
```

The above code is written in the Update button of the interface of customer and is used to pass the values inserted in the textboxes to the Update method of the CustomerDao class.

All the other interfaces which contains Update function contains a code which has the same structure as above code.

4)Distributed Database System

4.1)

- Storing details at the place where they are frequently used.

  Recording details regarding purchase of products from suppliers can be done in the server which is at the area which is closest to the manufacturing place of the product. Therefore, records of purchasing different products can be stored in different servers which are at different locations closest to the manufacturing place. Therefore, other than the cost which saves for communication, transportation cost can be reduced by distributing product by own vehicles of the company ABX.

  sales details of sales at a particular area can be recorded separately in different servers.

- Speed of processing.

  Other than accessing the same server again and again through the network which is cost a higher amount, the local data can be used when the data is distributed in the above manner. It will lead for minimized cost other than using a centralized system.

Advantages of DDBMS-

1) Modular Development.

   In a centralized database system, it is harder to expand the database when it's needed, But it Distributed database systems expansion can be done without any interruptions for the current functionalities [3].

2) More Reliable-

   In a centralized systems , the whole system break downs if any failure occurred for a certain component of the system. But in Distributed Database systems, less effect will be their for the whole system though a component break down [3].

3) Better Response-

If data are arranged in a proper way, the steps that have to be followed in order to perform a particular operation may be less. Therefore, the execution time for operations might be less [3].

4) Lower Communication Cost-

   If data is stored in the correct place where they are being used the most, cost for communicating data become less [3].

Disadvantages of DDBMS

1) Complexity-

   When the data becomes distributed, the arrangement of data can become more complicated than a centralize System. Therefore, it can also cause the data replication due to the complexity which will arise more problems if a particular software does not allow required replications for the system to posses [3].

2) Cost-

   When the system become distributed there will be more cost for purchasing and maintenance of the system. Also , there will be more expenditure for the labor which will need for the maintenance of the system [3].

3) Integrity control is difficult-

   When maintaining the validity and consistency of data, it may be required to change different constraints of the system. Usually, it will be needed to access large amount of data in order to manipulate those constraint. Because of that DDBMS may cost more than the centralized system when maintaining data [3].

4) Security-

   In centralized systems, only one system needs to be considered when applying the security. But in Distributed database systems security of all parts of the systems which are at different places need to be considered separately and also security should be applied for the network which all the components of the system are connected with, which will cost higher than the centralized system [3].

4.2)

Different constraints can be implemented in the database in order to maintain the concurrency of data which is common to all the servers. In the above scenario ID can be generated in a common way to all the servers using a trigger or any other method , so that it will not interrupt the consistency of the data though sales details and purchase details are recorded at different places [4].

References

[1] "Learning Hub," [Online]. Available: https://learn.g2.com/ide. [Accessed 04 12 2020].

[2] "Guru99," [Online]. Available: https://www.guru99.com/relational-algebra-dbms.html. [Accessed 03 01 2021].

[3] "Notes Station," [Online]. Available: https://notes-station.blogspot.com/2018/05/advantage-and-disadvantage-of-ddbms.html. [Accessed 08 01 2021].

[4] "Guru99," [Online]. Available: https://www.guru99.com/dbms-concurrency-control.html#:~:text=Concurrency%20Control%20in%20Database%20Management,integrity%20of%20the%20respective%20Database.. [Accessed 8 1 2021].