Online Shopping system

A PROJECT REPORT

In partial fulfillment for the course

Of

ADVANCE DATABASE MANAGEMENT AND DATABASE DESIGN

NORTHEASTERN UNIVERSITY BOSTON

MASSACHUSSETS

APRIL 2016

Table of contents

1. Introduction

1.1 Abstract

Best Buy is an American multinational consumer electronics corporation. It operates in the United States, Mexico, and Canada.

Best Buy is leading global retailer offering electronic gadgets like laptops, cell phones and accessories such as USB cables and hard disk etc. The products are available for purchase in store as well as online. The brand focusses on marketing strategies by advertising in newspapers, popular websites and banners across United States.

1.2 Purpose of Database

Currently businesses are facing challenges in the order management. Keeping record of products, customers, supplier and orders is a huge task. An efficient database is a necessity in such a business. Redundant data free database occupies less amount of space and works much efficiently. This project deals with the whole life cycle of the order management.

Orders in such businesses can be of two types. One order placed to the supplier and the other order they receive from the customers. In both the cases, tracking order is very important. The main factor along with the order is the product availability. Hence, this project provides a database to store data of product availability and the required products by analyzing the inventory. Once the order is placed, order tracking is of much importance to know the status of order. Products at times are also returned back from the customers. Proper information of the product while order and product while return must be registered along with the reason for return. Also another important thing to be recorded is the finance covering the expenses in orders to supplier and the income from the orders from the customers.

1.3 <u>Project Goals</u>

With the development of e-commerce websites and emerging use of it by customers to gratify their shopping needs, the management of best buy realizes that in order to support an emerging business with an online existence, a new proficient database for order management will be required. Some of the database needs are listed below.

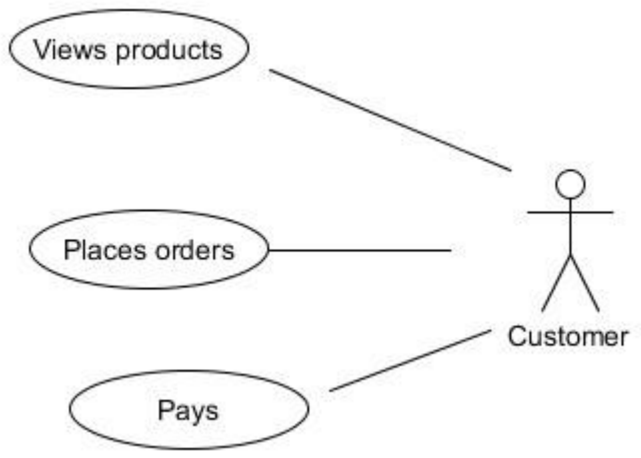
1. Associating the customer details to the customer id for effective storage and retrieval of the respective customer products.
2. Associating the orders table to product inventory so that when order is placed for particular products it reflects the available count of product in product inventory.
3. Also Linking the order return table with product inventory so that if order is returned, it again reflects the product availability count.
4. Manufacturer is linked with the product table

2. Roles and Responsibilities

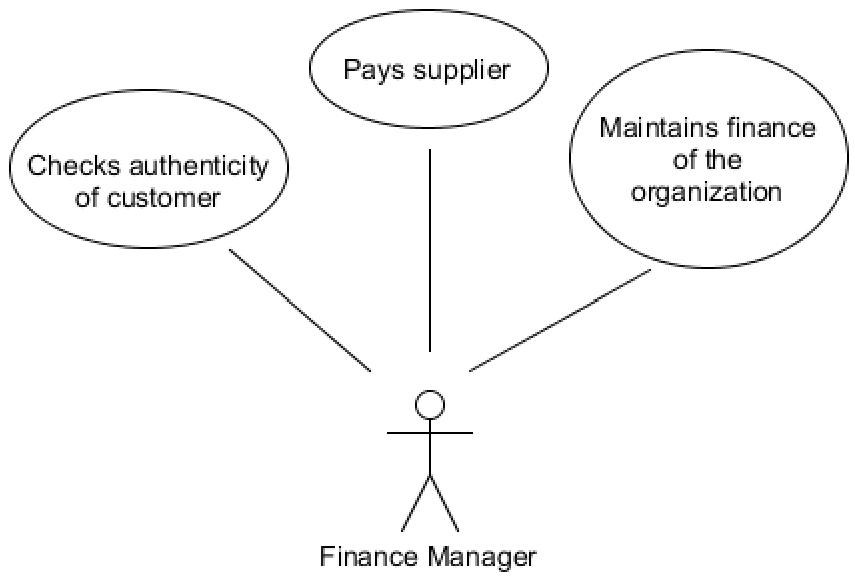
Roles	**Responsibilities**
Customer	□ Views the products □ Places Order □ Pays
Finance Manager	□ Checks authenticity of customer □ Pays the supplier □ Maintains finance of the organization
Order Manager	□ Keeps track of the orders
Product Manager	□ Keeps track of products □ Updates the cost □ Notifies the procurement manager in case of shortage of ordered products
Procurement Manager	□ Orders from supplier for products □ Checks the products received □ Forwards the bills to finance manager
Supplier	□ Receives the product request □ Sends the products □ Sends the cost of products sent
Inventory Manager	□ Maintains the incoming and outgoing count of products □ Notifies the procurement manager in case of shortage
Shipment Manager	□ Records the orders to be shipped □ Maintains status of the shipment
Returns Manager	□ Manages the returns of the orders

3. Use Case Diagrams

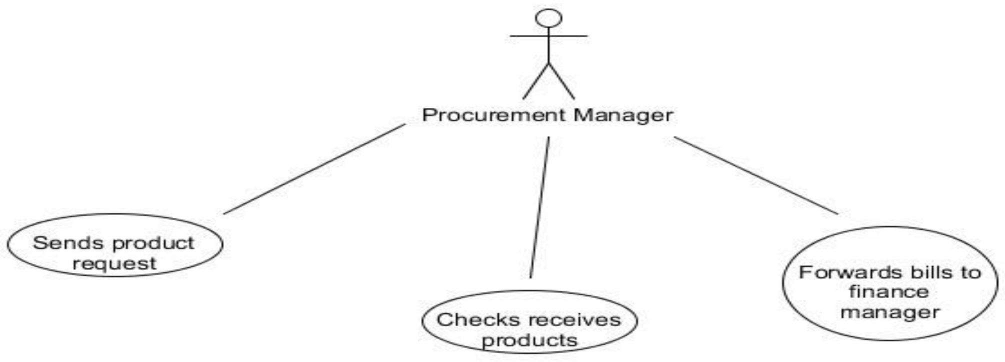
Customer:



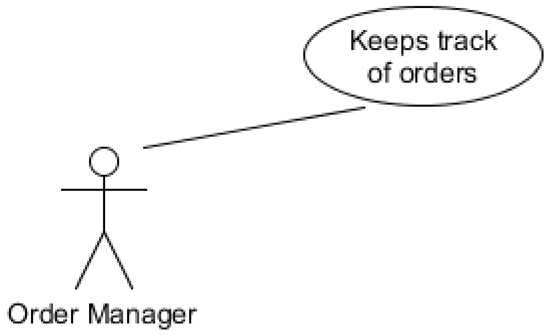
Finance Manager:


Pays supplier
Checks authenticity of customer
Maintains finance of the organization
Finance Manager

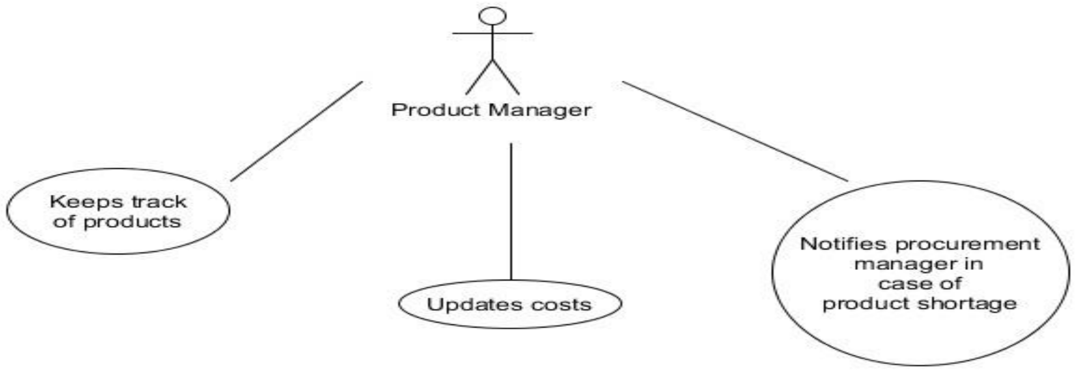
Procurement Manager:



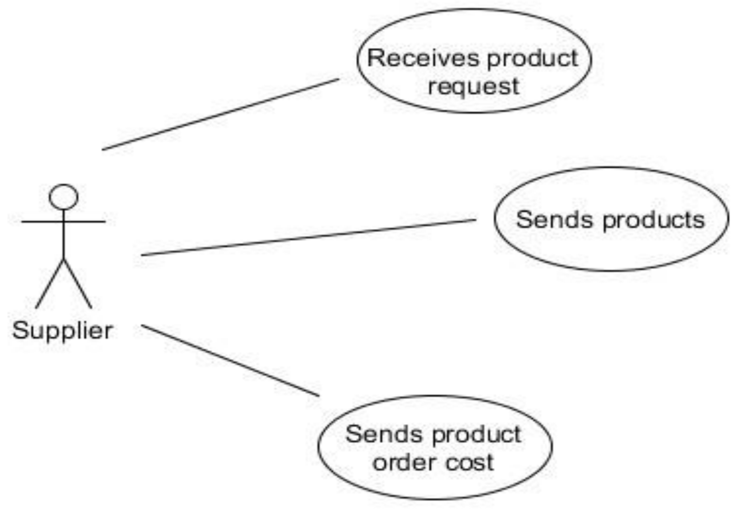
Order Manager:



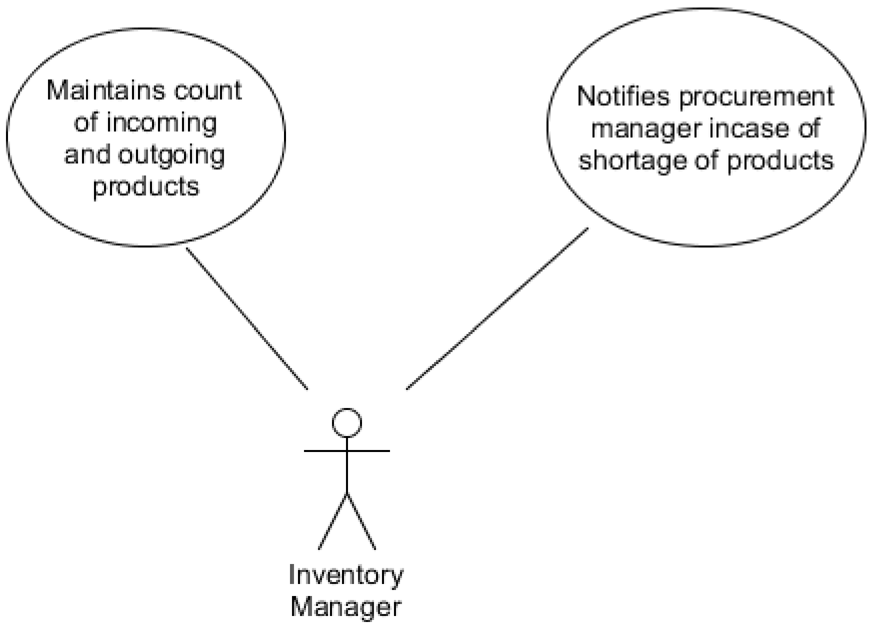
Product Manager:



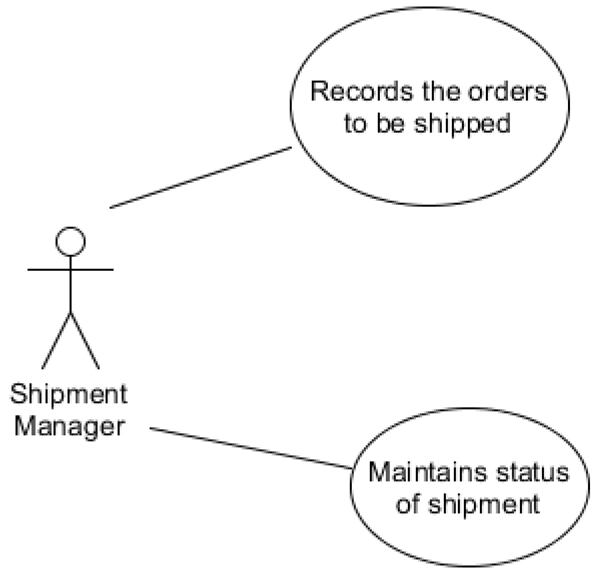
Supplier:



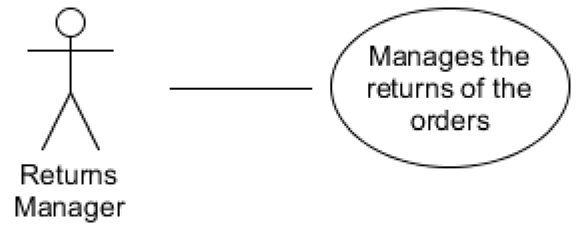
Inventory Manager:


Maintains count of incoming and outgoing products
Notifies procurement manager incase of shortage of products
Inventory Manager

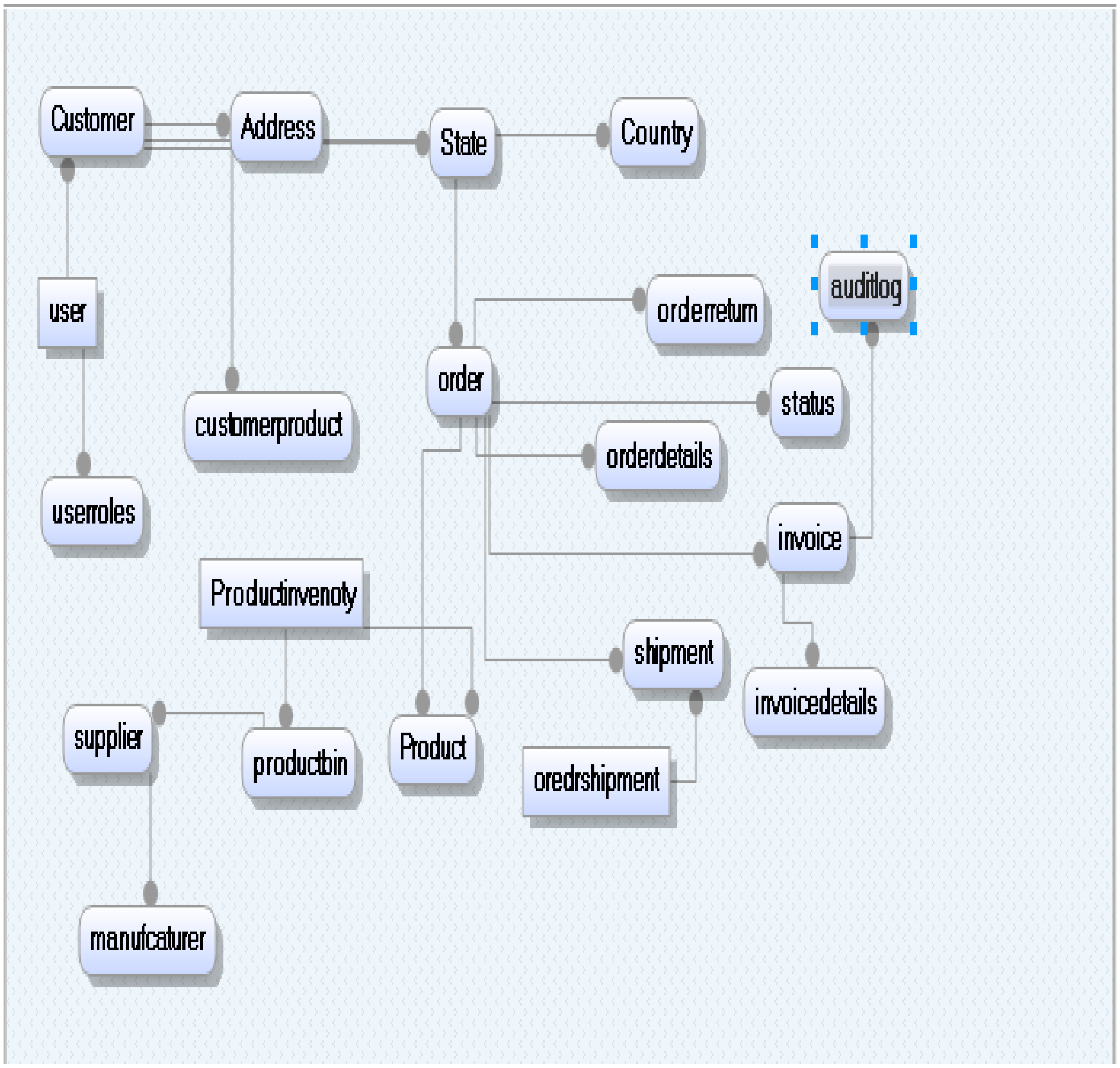
Shipment Manager:


Records the orders to be shipped
Shipment Manager
Maintains status of shipment

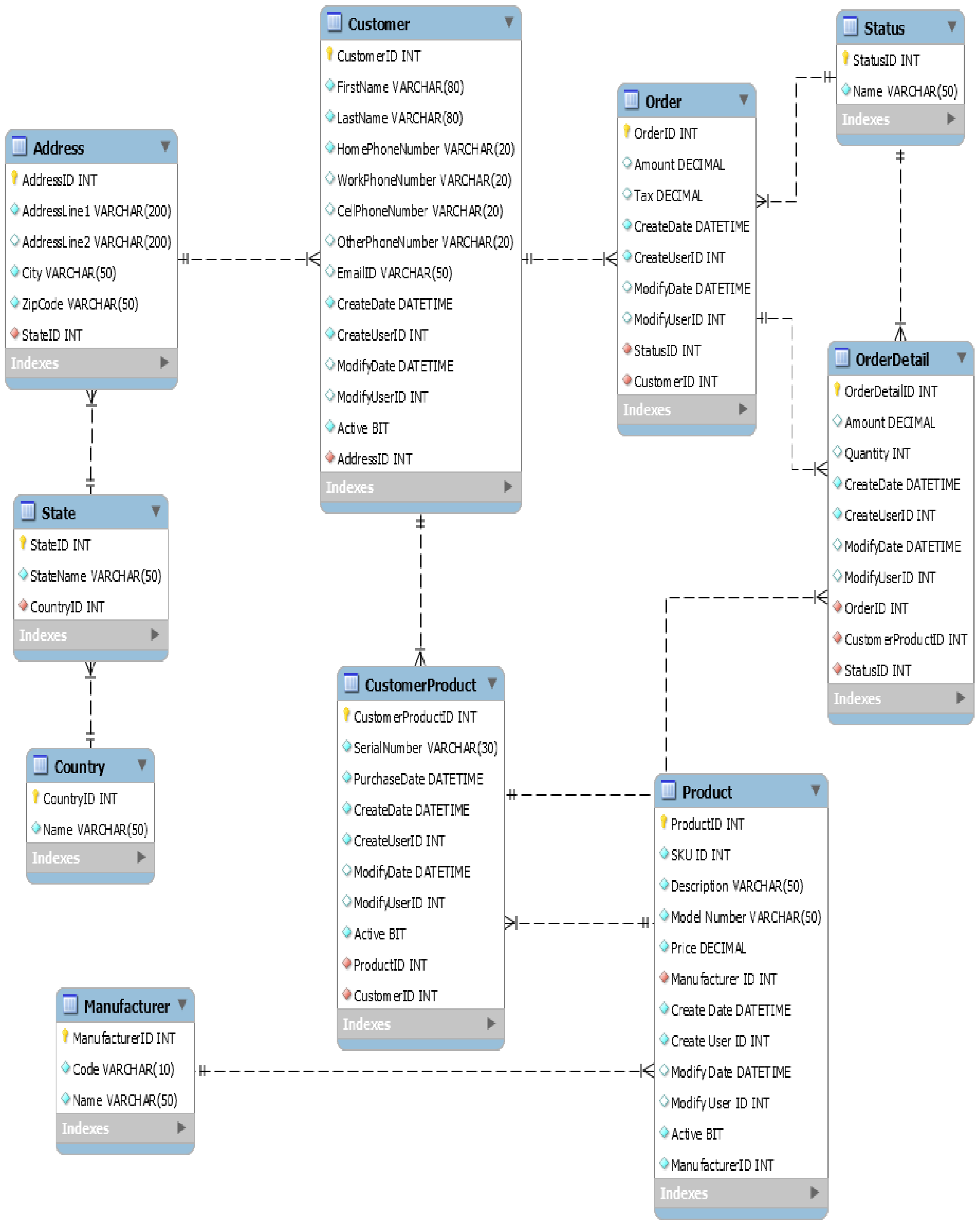
Returns Manager:



4. Conceptual ER Diagram



4.1.Logical ER Diagram



5. Normalization

Database normalization is the process of organizing the fields and tables of a relational database to minimize redundancy. Normalization usually involves dividing large tables into smaller tables and defining relationships between them.
In this project's database, all the tables are normalized. It involves 1NF, 2NF and 3NF.

Using 1 NF:
A table is in 1NF if it is free from multi-valued attributes. In this database system all the tables contain only single values.

Using 2NF:
A table is in 2NF if it is in 1NF and all the partial dependencies have been removed.
Attributes which have partial dependent on one part of the composite key alone are removed and formed as a separate table.

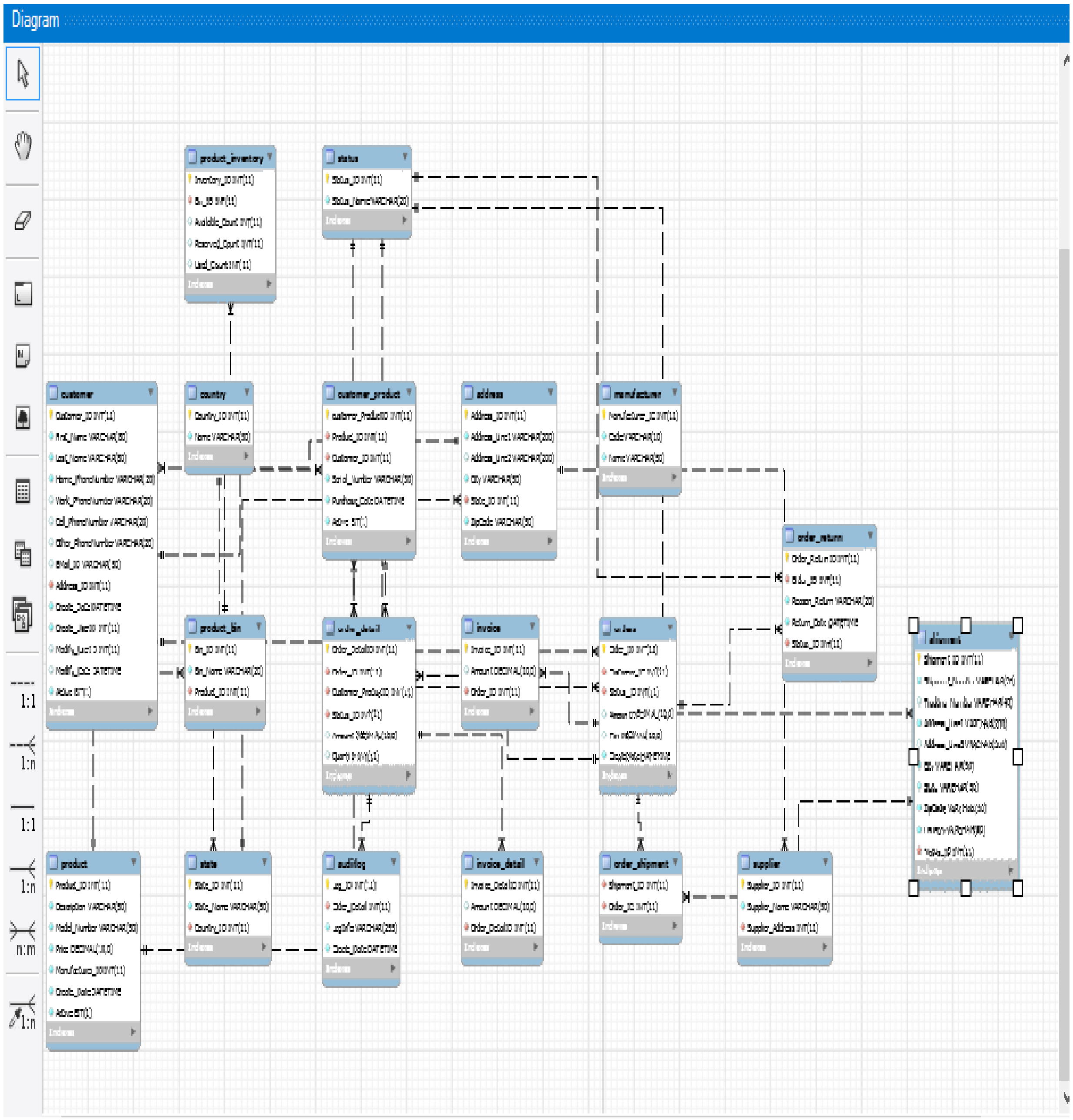
Using 3NF:
A table is in 3NF if it is in 2NF and no non-key field depends on a key which is not a primary key.

Consider the address table down:
This overcomes the update and deletion anomalies as the State_ID is the foreign key here, hence the delivery address of a customer is prevented from changing.

Address_ID	Address_Line1	Address_Line2	City	State_ID	ZipCode
1	A	B	Mumbai	1	400101

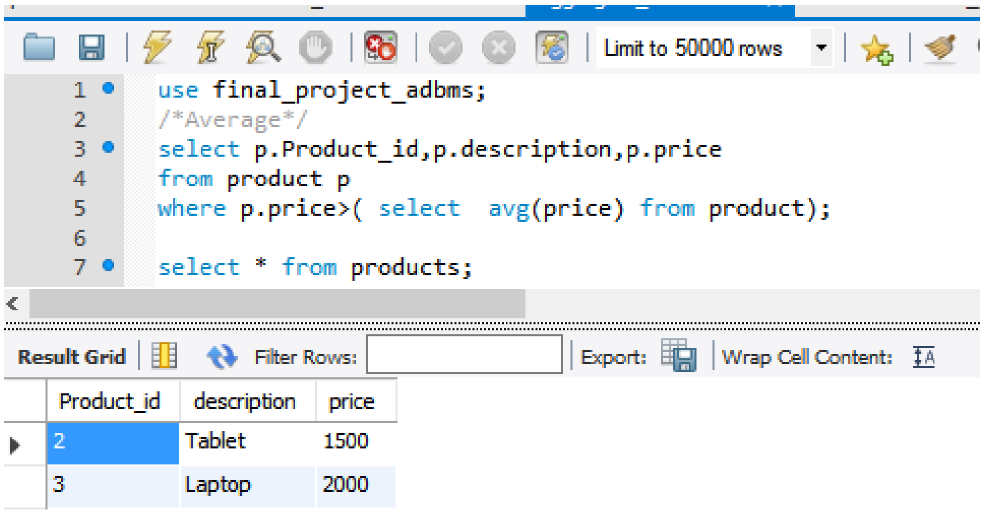
All the fields are dependent only on the primary key field which is Address_ID, and this shows that the table is in 3NF.

6. Final ER Diagram



7. Queries

1. **Aggregate Function Average:** Below query uses average function to give us all the products which has price greater than average price of products.



Limit to 50000 rows

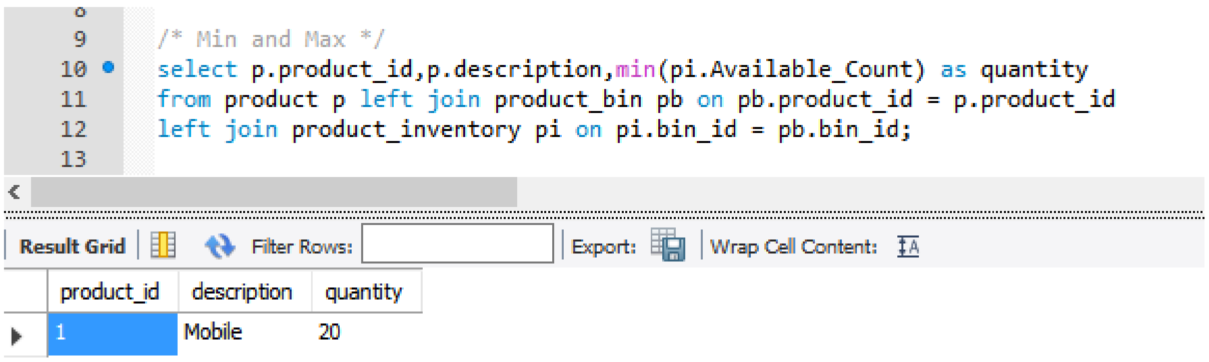
```
use final_project_adbms;
/*Average*/
select p.Product_id,p.description,p.price
from product p
where p.price>( select  avg(price) from product);

select * from products;
```

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

Product_id	description	price
2	Tablet	1500
3	Laptop	2000

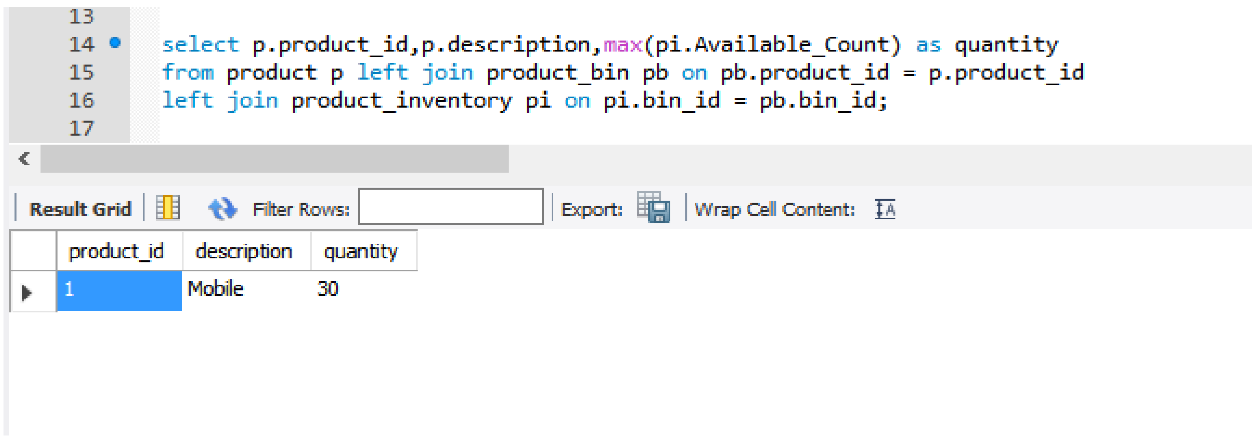
2. **Aggregate function Min & Max:** Below query descries the min and max function. Which gives us the product with min and max quantity.



```
/* Min and Max */
select p.product_id,p.description,min(pi.Available_Count) as quantity
from product p left join product_bin pb on pb.product_id = p.product_id
left join product_inventory pi on pi.bin_id = pb.bin_id;
```

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

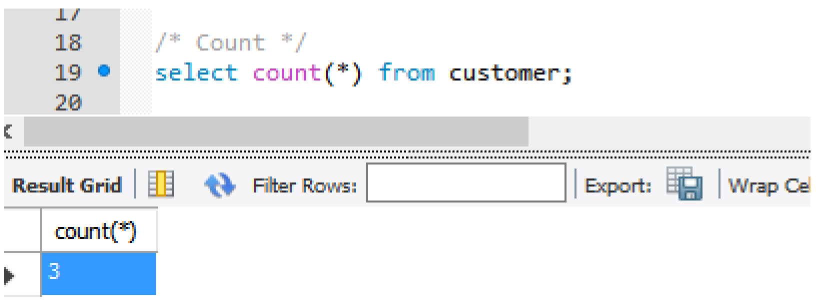
product_id	description	quantity
1	Mobile	20



```
select p.product_id,p.description,max(pi.Available_Count) as quantity
from product p left join product_bin pb on pb.product_id = p.product_id
left join product_inventory pi on pi.bin_id = pb.bin_id;
```

product_id	description	quantity
1	Mobile	30

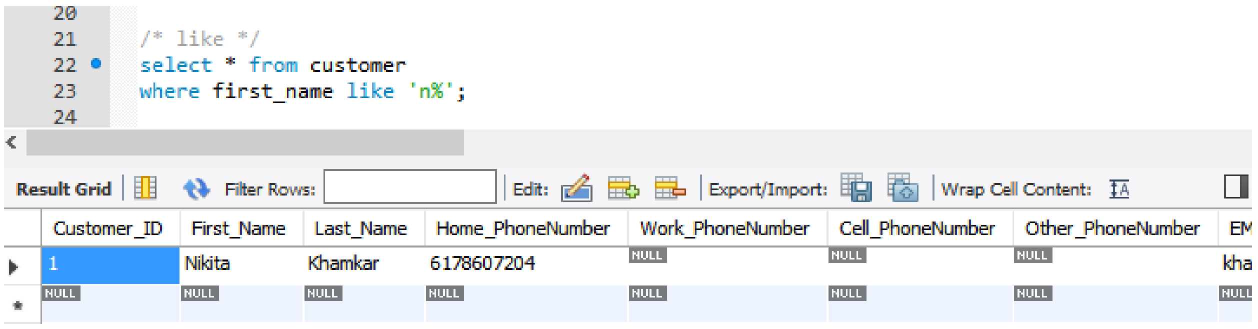
3. **Aggregate Function Count :** Count function give us the count of records in table



```
/* Count */
select count(*) from customer;
```

count(*)
3

4. **Aggregate Function Like:** Like function gives the records starting with that particular type mentioned in like statement



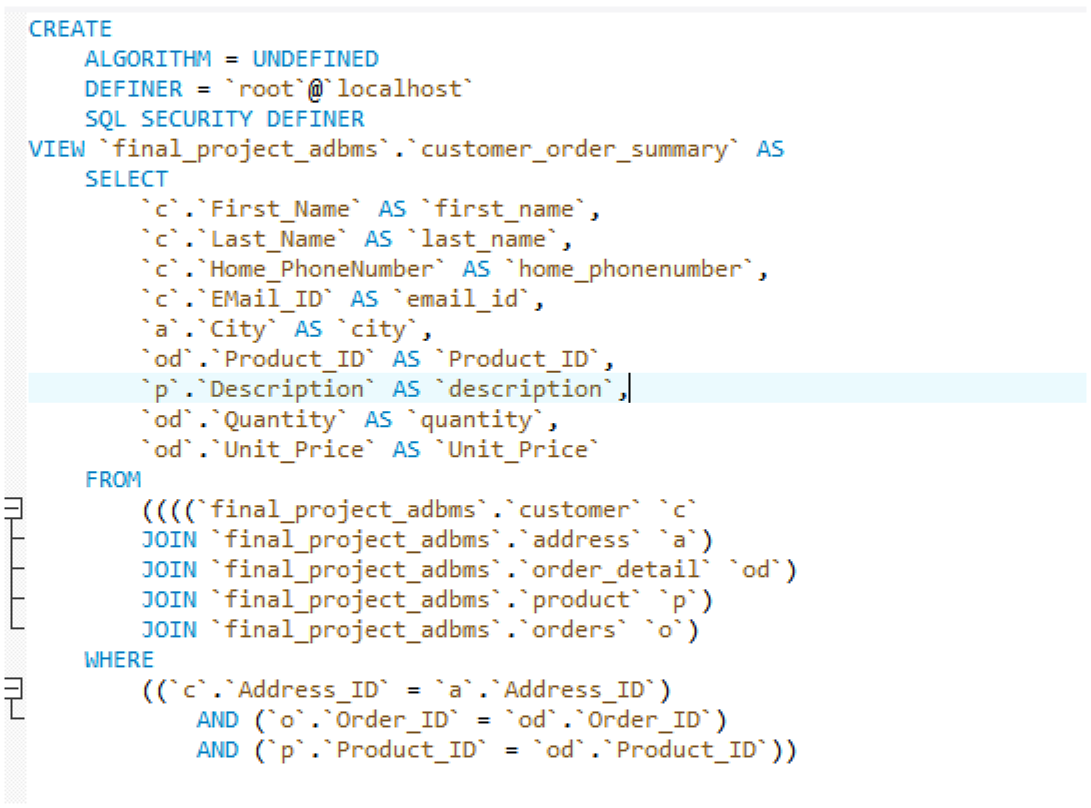
```
/* like */
select * from customer
where first_name like 'n%';
```

	Customer_ID	First_Name	Last_Name	Home_PhoneNumber	Work_PhoneNumber	Cell_PhoneNumber	Other_PhoneNumber	EM
	1	Nikita	Khamkar	6178607204	NULL	NULL	NULL	kha
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

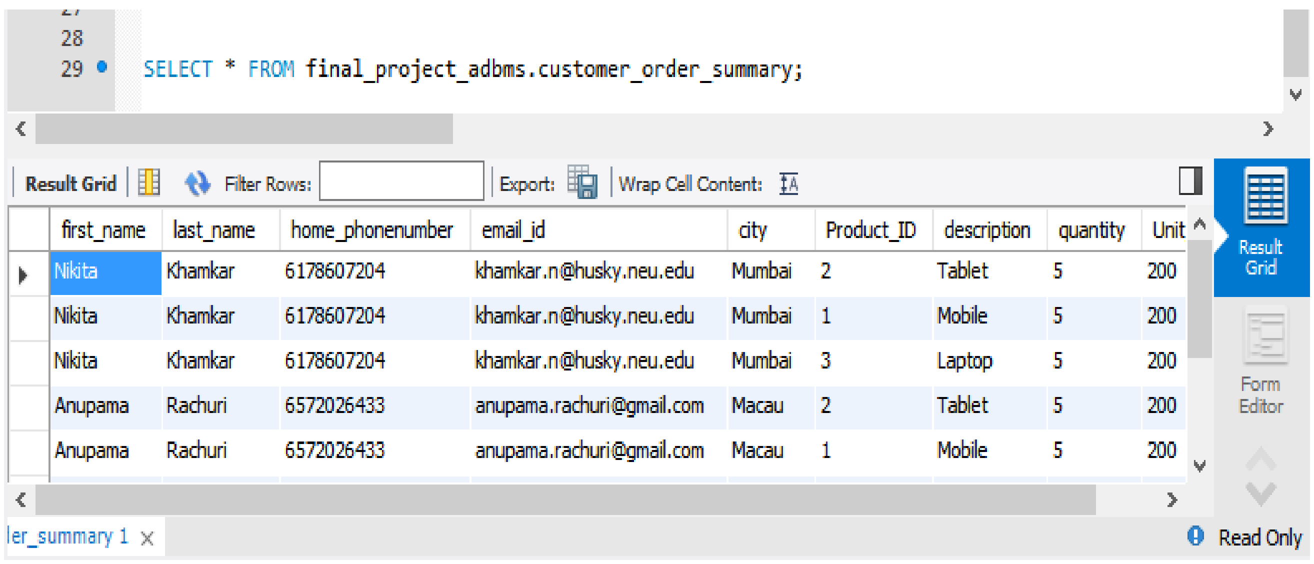
8. Views

1. Customer order summary view:

It gives us the order summary of customer without any joins.



```
CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `final_project_adbms`.`customer_order_summary` AS
    SELECT
        `c`.`First_Name` AS `first_name`,
        `c`.`Last_Name` AS `last_name`,
        `c`.`Home_PhoneNumber` AS `home_phonenumber`,
        `c`.`EMail_ID` AS `email_id`,
        `a`.`City` AS `city`,
        `od`.`Product_ID` AS `Product_ID`,
        `p`.`Description` AS `description`,
        `od`.`Quantity` AS `quantity`,
        `od`.`Unit_Price` AS `Unit_Price`
    FROM
        ((((`final_project_adbms`.`customer` `c`
        JOIN `final_project_adbms`.`address` `a`)
        JOIN `final_project_adbms`.`order_detail` `od`)
        JOIN `final_project_adbms`.`product` `p`)
        JOIN `final_project_adbms`.`orders` `o`)
    WHERE
        ((`c`.`Address_ID` = `a`.`Address_ID`)
            AND (`o`.`Order_ID` = `od`.`Order_ID`)
            AND (`p`.`Product_ID` = `od`.`Product_ID`))
```

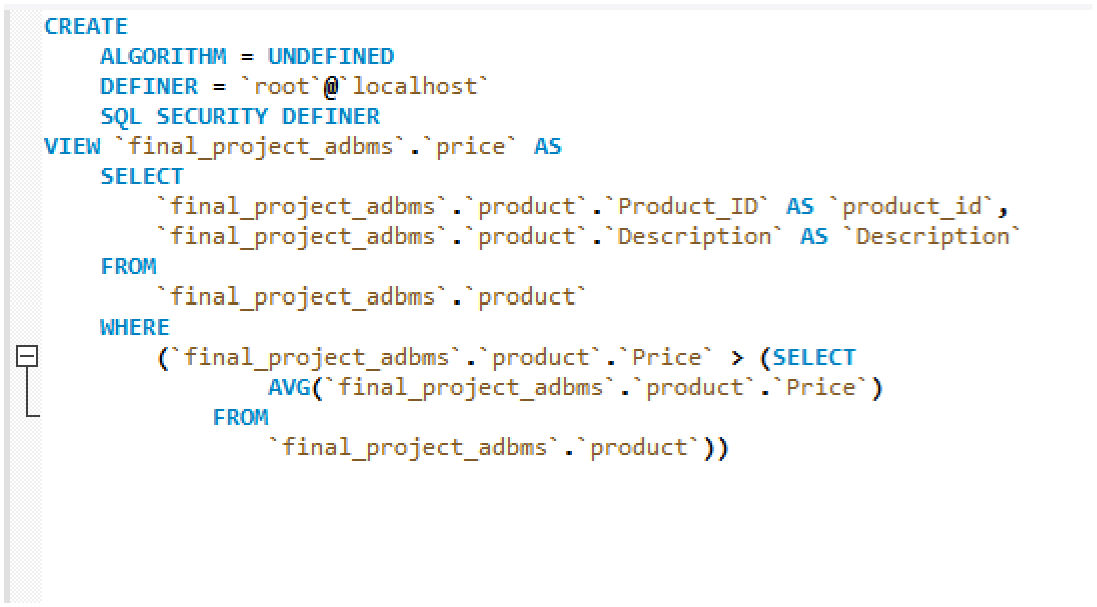


```
SELECT * FROM final_project_adbms.customer_order_summary;
```

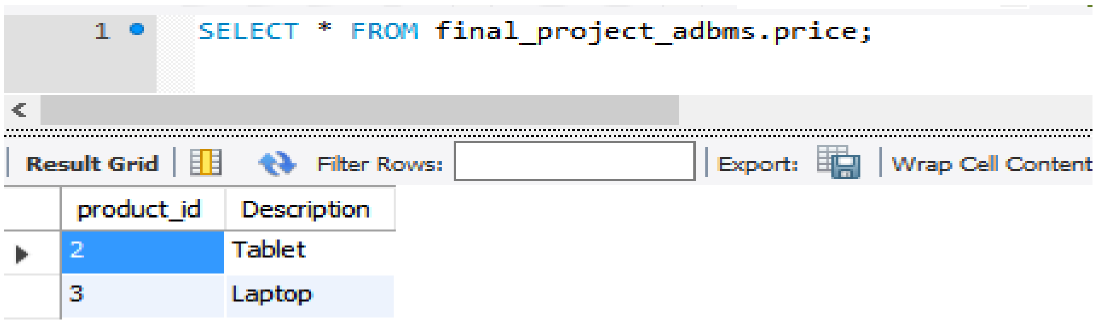
first_name	last_name	home_phonenumber	email_id	city	Product_ID	description	quantity	Unit
Nikita	Khamkar	6178607204	khamkar.n@husky.neu.edu	Mumbai	2	Tablet	5	200
Nikita	Khamkar	6178607204	khamkar.n@husky.neu.edu	Mumbai	1	Mobile	5	200
Nikita	Khamkar	6178607204	khamkar.n@husky.neu.edu	Mumbai	3	Laptop	5	200
Anupama	Rachuri	6572026433	anupama.rachuri@gmail.com	Macau	2	Tablet	5	200
Anupama	Rachuri	6572026433	anupama.rachuri@gmail.com	Macau	1	Mobile	5	200

2. **Price View:**

It gives us the price of the products without any insight view of product and price table, it just displays the price of product.



```
CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `final_project_adbms`.`price` AS
    SELECT
        `final_project_adbms`.`product`.`Product_ID` AS `product_id`,
        `final_project_adbms`.`product`.`Description` AS `Description`
    FROM
        `final_project_adbms`.`product`
    WHERE
        (`final_project_adbms`.`product`.`Price` > (SELECT
                AVG(`final_project_adbms`.`product`.`Price`)
            FROM
                `final_project_adbms`.`product`))
```



```
SELECT * FROM final_project_adbms.price;
```

product_id	Description
2	Tablet
3	Laptop

10.Procedures

1. **P_INSERT_CUSTOMER**: To insert details for an Customer and associated address of customer.

This procedure will add details of customer in customer table and associated address of customer in address table respectively.

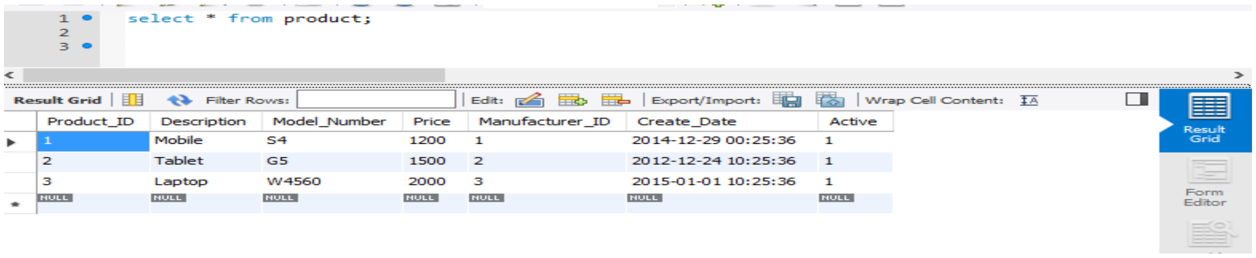
For example:

Call P_INSERT_CUSTOMER (4,6,’Varsha’,’Yalpale’,’6178607204’,null,null,null, ‘varsha.n@husky.neu.edu’,’FGH’,’LKH’,’PUNE’,1,02110,4);

This will add an entry to the tables with given information in stored procedure.

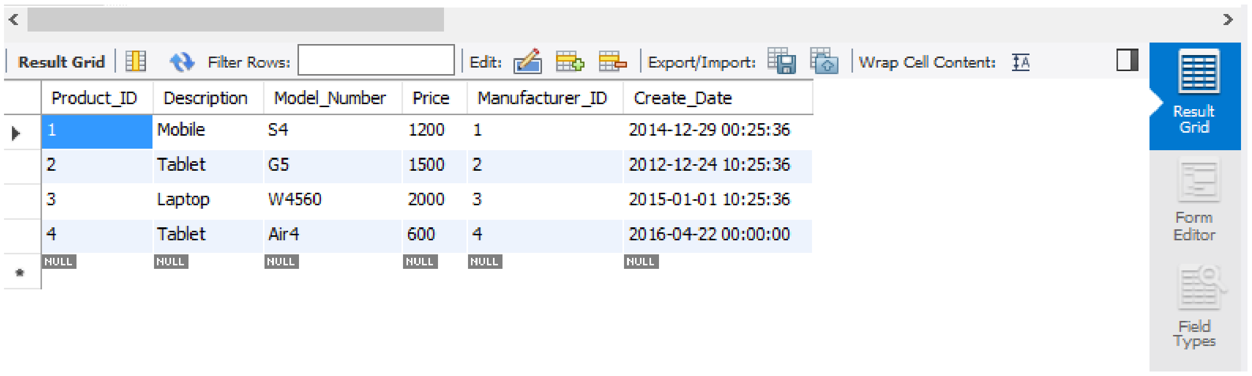
2. **P_INSERT_PRODUCT:** To enter the product details.

Call P_INSERT_PRODUCT (4,’Tablet’,Air4,600,4);



```
select * from product;
```

Product_ID	Description	Model_Number	Price	Manufacturer_ID	Create_Date	Active
1	Mobile	S4	1200	1	2014-12-29 00:25:36	1
2	Tablet	G5	1500	2	2012-12-24 10:25:36	1
3	Laptop	W4560	2000	3	2015-01-01 10:25:36	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL



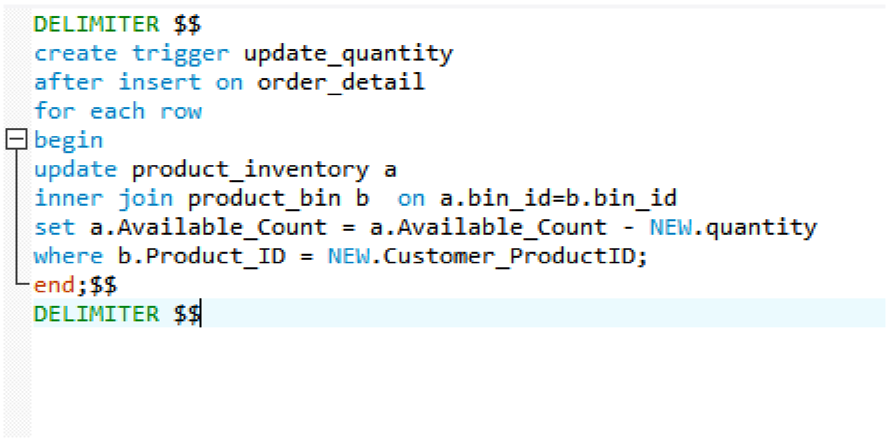
Product_ID	Description	Model_Number	Price	Manufacturer_ID	Create_Date
1	Mobile	S4	1200	1	2014-12-29 00:25:36
2	Tablet	G5	1500	2	2012-12-24 10:25:36
3	Laptop	W4560	2000	3	2015-01-01 10:25:36
4	Tablet	Air4	600	4	2016-04-22 00:00:00
NULL	NULL	NULL	NULL	NULL	NULL

3. **P_ORDER_SUBTOTAL**: Finding cost of particular order

This procedures finds cost of particular product based on product and order details.

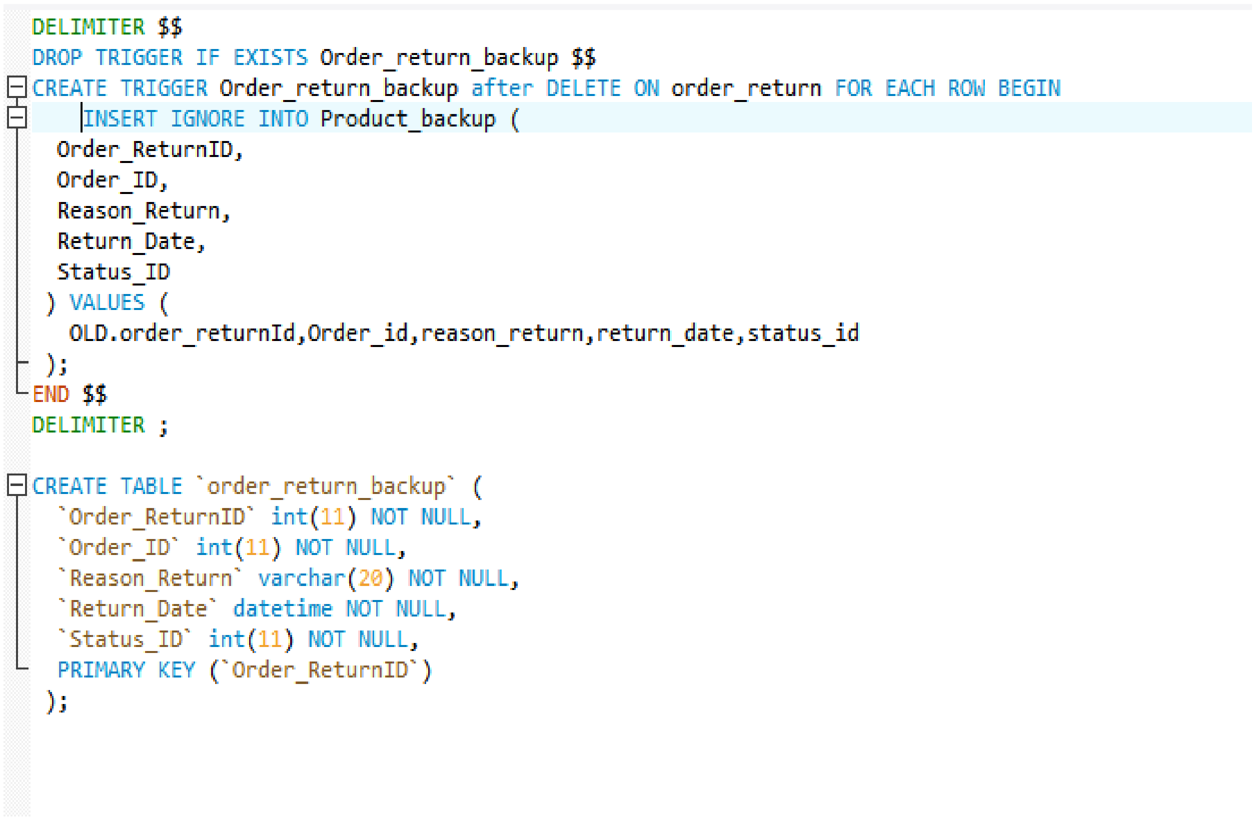
11. Triggers

1. Trigger **T_UPDATE_QUANTITY:** for updating the product quantity in product inventory table after order is placed.



```
DELIMITER $$
create trigger update_quantity
after insert on order_detail
for each row
begin
update product_inventory a
inner join product_bin b  on a.bin_id=b.bin_id
set a.Available_Count = a.Available_Count - NEW.quantity
where b.Product_ID = NEW.Customer_ProductID;
end;$$
DELIMITER $$
```

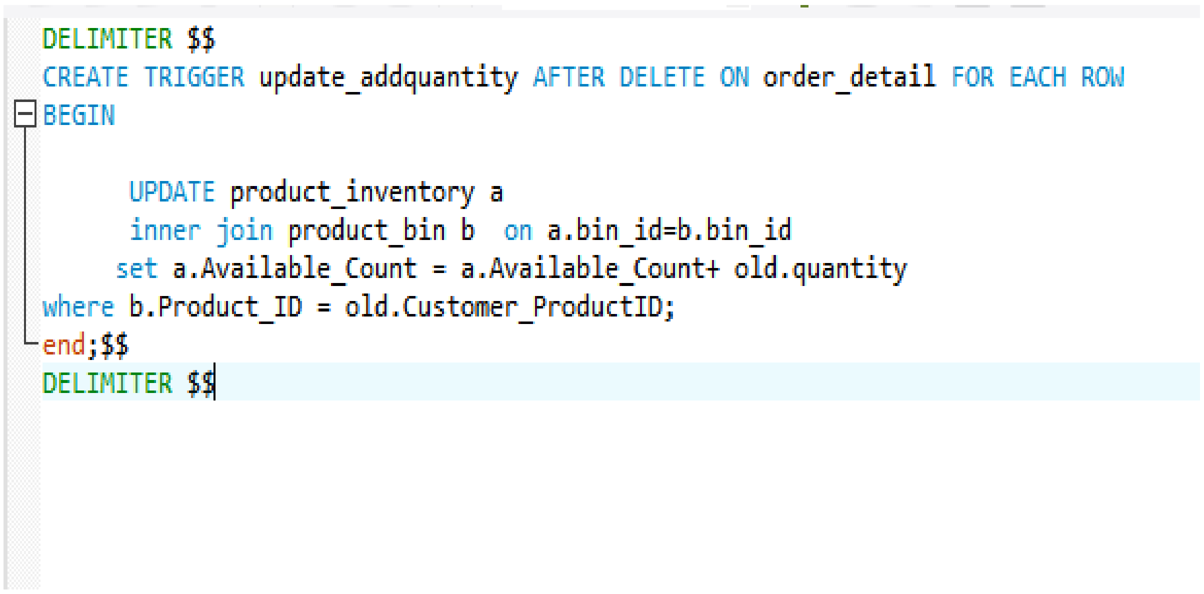
2.Trigger **T_ORDER_RETURN_BACKUP:** For taking order return back up
This trigger will populate the data in order return backup table after deleting any data from order return.



```
DELIMITER $$
DROP TRIGGER IF EXISTS Order_return_backup $$
CREATE TRIGGER Order_return_backup after DELETE ON order_return FOR EACH ROW BEGIN
    INSERT IGNORE INTO Product_backup (
  Order_ReturnID,
  Order_ID,
  Reason_Return,
  Return_Date,
  Status_ID
 ) VALUES (
   OLD.order_returnId,Order_id,reason_return,return_date,status_id
 );
END $$
DELIMITER ;

CREATE TABLE `order_return_backup` (
  `Order_ReturnID` int(11) NOT NULL,
  `Order_ID` int(11) NOT NULL,
  `Reason_Return` varchar(20) NOT NULL,
  `Return_Date` datetime NOT NULL,
  `Status_ID` int(11) NOT NULL,
  PRIMARY KEY (`Order_ReturnID`)
 );
```

3. Trigger **T_addProduct_Quanity:** For updating product inventor table

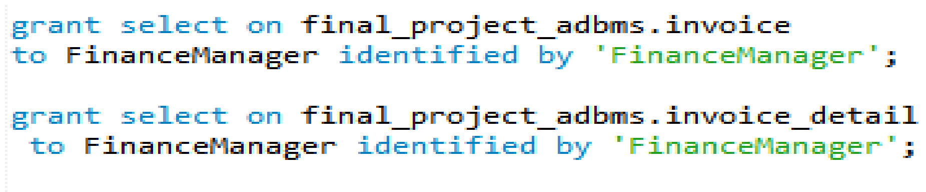


```
DELIMITER $$
CREATE TRIGGER update_addquantity AFTER DELETE ON order_detail FOR EACH ROW
BEGIN

      UPDATE product_inventory a
      inner join product_bin b  on a.bin_id=b.bin_id
     set a.Available_Count = a.Available_Count+ old.quantity
where b.Product_ID = old.Customer_ProductID;
end;$$
DELIMITER $$
```

12 Grants

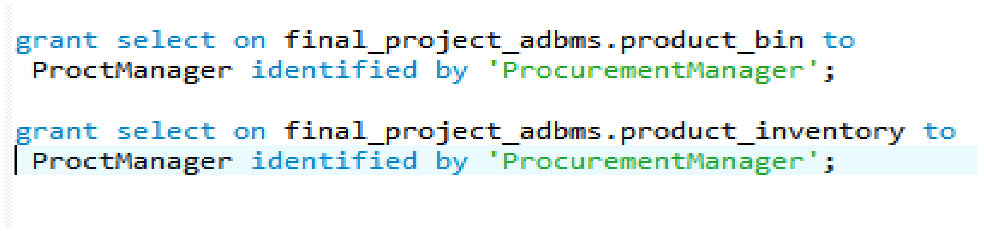
1. **Grant to Finance Manager** to have full access to invoice and invoice details table to keep track of finanace.



```
grant select on final_project_adbms.invoice
to FinanceManager identified by 'FinanceManager';

grant select on final_project_adbms.invoice_detail
 to FinanceManager identified by 'FinanceManager';
```

2. **Grant to procurement manager** to have full access to product and product inventory table.



```
grant select on final_project_adbms.product_bin to
 ProctManager identified by 'ProcurementManager';

grant select on final_project_adbms.product_inventory to
 ProctManager identified by 'ProcurementManager';
```

13. Backup & Recovery

For backup below script is used to backup my final_project_adbms database.

#mysqldump – u root – final_project_adbms > final_project_adbmsbackup.sql

With this script a backup file of database is created with new name.
Also, Data Export utility is available In Mysql for backup.

For Restoration, below script is used.

#mysql – u root – p final_project_adbms < final_project_adbmsNew.sql

14. Security

Database Design and Security is the most important criteria that needs to be considered. It is necessary to set up privileges in order to restrict acces to people using the database in addition to setting up the paswords and physical security.

Several ways to secure database are:

1. GRANT option: Granting privileges is an important aspect of database security.

2. If you want to disable network access to your database server , use below command line skip-networking

3. We can delete the test database that is created by default in Mysql as this database is accessible to all anonymous users.
   Drop database test

4. Use passwords for users created. This is most basic security.

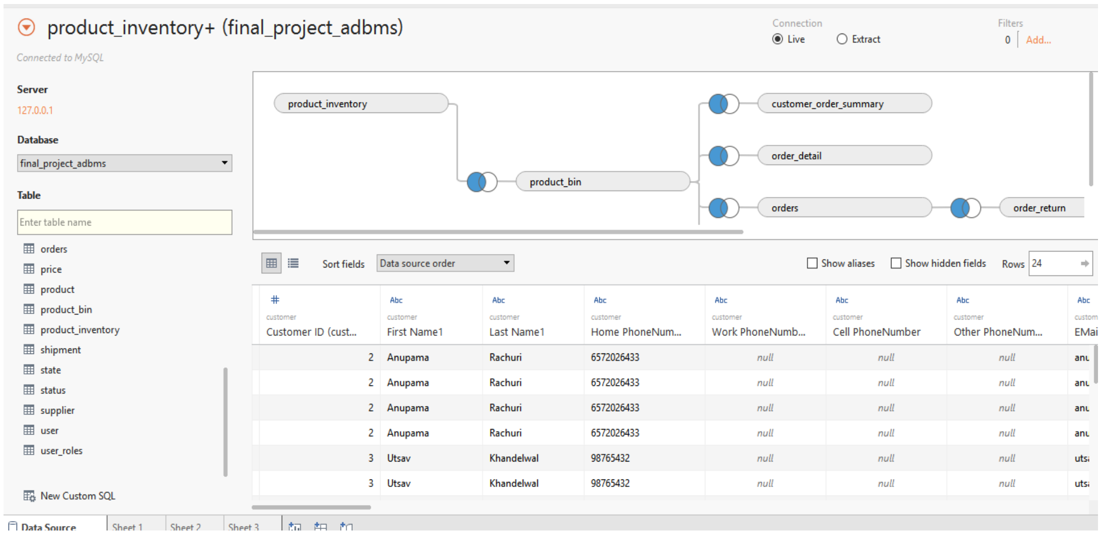
15. Business Intelligence

Business Intelligence plays an important role in any business as it allows the business users to analyze the data which are represented in form of graphs and charts. This gives an overall idea about how exactly the business works and where it is now. I have used Tableau for Business Intelligence and below mention are some of deliverables.

By using the data in the database, the business user can look into

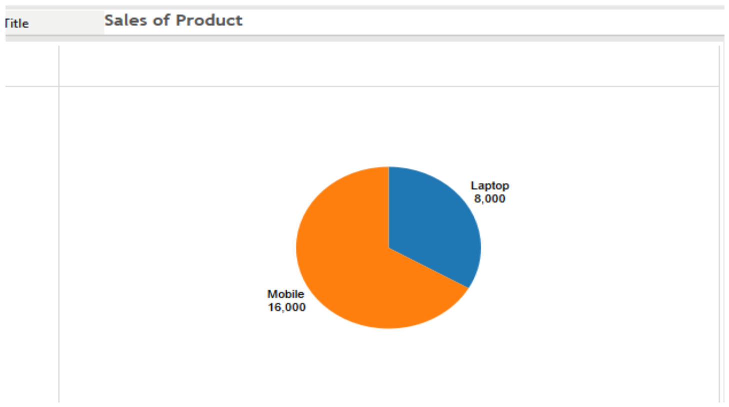
- Sales of a particular product along with the profit
- Highest selling products
- Frequent customers of the company
- Sales trend
- Products by Manufacturer and available quantity

1. Connection of Database to Tableau

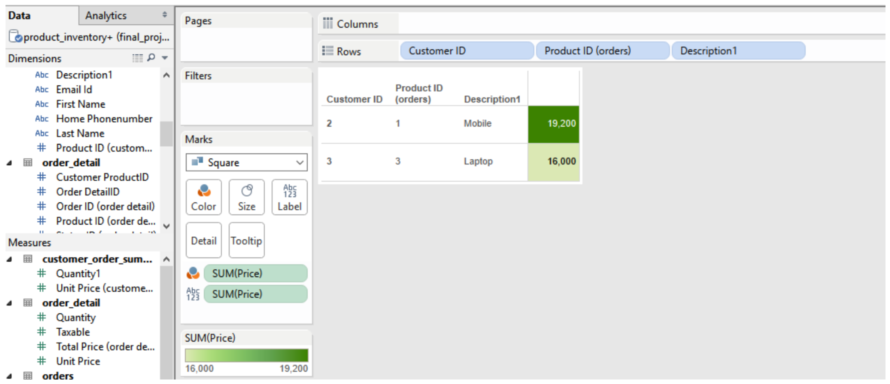


Customer ID (cust...	First Name1	Last Name1	Home PhoneNum...	Work PhoneNumb...	Cell PhoneNumber	Other PhoneNum...	EMai
2	Anupama	Rachuri	6572026433	null	null	null	anu
2	Anupama	Rachuri	6572026433	null	null	null	anu
2	Anupama	Rachuri	6572026433	null	null	null	anu
2	Anupama	Rachuri	6572026433	null	null	null	anu
3	Utsav	Khandelwal	98765432	null	null	null	uts
3	Utsav	Khandelwal	98765432	null	null	null	uts

2. Highest Selling Product and Price



3. Customer order summary



product_inventory+ (final_proj...

Dimensions
- Description1
- Email Id
- First Name
- Home Phonenumber
- Last Name
- Product ID (custom...
- order_detail
  - Customer ProductID
  - Order DetailID
  - Order ID (order detail)
  - Product ID (order de...

Measures
- customer_order_sum...
  - Quantity1
  - Unit Price (custome...
- order_detail
  - Quantity
  - Taxable
  - Total Price (order de...
  - Unit Price
- orders

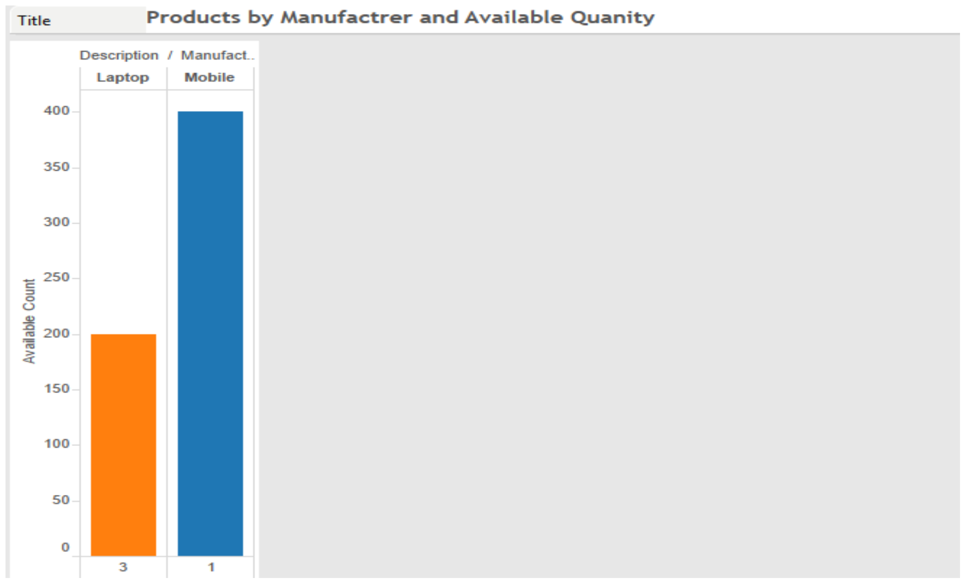
Rows: Customer ID | Product ID (orders) | Description1

Marks: SUM(Price) (Color), SUM(Price) (Label)

SUM(Price): 16,000 – 19,200

Customer ID	Product ID (orders)	Description1	
2	1	Mobile	19,200
3	3	Laptop	16,000

4. Products by manufacturer and available quanity



16. Conclusion

The report comprises of functional specifications and configurations, and design related prototype explained in detailed. The database is designed considering data integrity and security standards. The naming conventions are given as per standard rules and regulations. The database is designed using set business constraints and logic based upon the basic business requirements.