

Database Management System for an E-commerce website

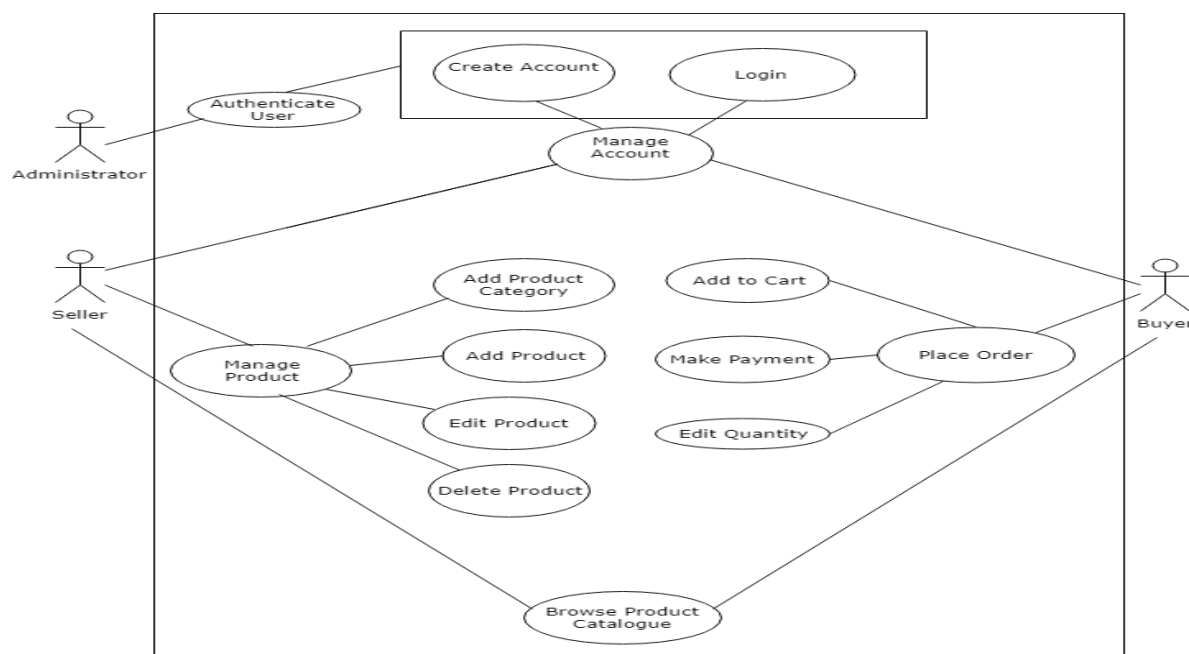
Problem Statement

An E-commerce website is a platform through which buyers and sellers can communicate with each other for buying or selling products in a faster and efficient way. Item category includes wide range of products from different domain. Every item is placed in the associated item category. A seller posts the products to be sold online with the available quantity, price and discount offered if any. Customer will buy his desired products. The system administrator will authenticate the user identity and user role (which is customer or supplier in this case). The entire model serves as a virtual shopping mall where the transaction takes place online. Current marketplace follows traditional approach which is less efficient. In the proposed model, the data will be managed and stored in a well-organized manner.

Approach

Customer and supplier can login into the system with their own credentials and has defined privileges to access the system. A customer will shortlist the selected items from a wide range of products and its category in his cart. A customer can order multiple products in a single order and in a different quantity. A supplier can post products, it's corresponding price, available quantity and any discounts if applicable. Once an order is placed, payment information is required from the customer. On successful payment, order will be shipped by supplier to the customer.

Use Case Diagram



The diagram illustrates a database schema for a retail system. The tables and their attributes are as follows:

- customer**: CustomerID INT(10) (PK), CustomerName VARCHAR(15), ContactNumber VARCHAR(12), LOGIN_DETAILS_loginID INT(10). Indexes: CustomerID, ContactNumber, LOGIN_DETAILS_loginID.
- address**: AddressID INT(10) (PK), StreetName VARCHAR(70), City VARCHAR(20), State VARCHAR(20), PostalCode VARCHAR(5), CUSTOMER_CustomerID INT(10). Indexes: AddressID, CUSTOMER_CustomerID.
- login_details**: loginID INT(10) (PK), Username VARCHAR(15), Password VARCHAR(15). Indexes: loginID, Username, Password. Triggers: login_details.
- cart**: CartID INT(10) (PK), CUSTOMER_CustomerID INT(10). Indexes: CartID, CUSTOMER_CustomerID.
- cart_has_item**: CART_CartID INT(10), CART_CUSTOMER_CustomerID INT(10), ITEM_ProductID INT(10). Indexes: CART_CartID, CART_CUSTOMER_CustomerID, ITEM_ProductID.
- order_detail**: OrderID INT(10) (PK), OrderPlacedDate DATE, SUPPLIER_SupplierID INT(10), CART_CartID INT(10), CART_CUSTOMER_CustomerID INT(10), ADDRESS_AddressID INT(10), OrderStatus VARCHAR(30). Indexes: OrderID, SUPPLIER_SupplierID, CART_CartID, CART_CUSTOMER_CustomerID, ADDRESS_AddressID, OrderStatus.
- order_has_item**: ORDER_OrderID INT(10), ORDER_CUSTOMER_CustomerID INT(10), ORDER_SUPPLIER_SupplierID INT(10), ORDER_SHIPMENT_ShippingID INT(10), ITEM_ProductID INT(10), ITEM_CATEGORY_CategoryID INT(10). Indexes: ORDER_OrderID, ORDER_CUSTOMER_CustomerID, ORDER_SUPPLIER_SupplierID, ORDER_SHIPMENT_ShippingID, ITEM_ProductID, ITEM_CATEGORY_CategoryID.
- item**: ProductID INT(10) (PK), ProductName VARCHAR(15), QuantityPerUnit INT(10), UnitPrice FLOAT, DiscountPercent FLOAT, CATEGORY_CategoryID INT(10). Indexes: ProductID, CATEGORY_CategoryID. Triggers: item. 1 more...
- category**: CategoryID INT(10) (PK), CategoryName VARCHAR(15). Indexes: CategoryID.
- shipment**: ShippingID INT(10) (PK), ShipmentStatus VARCHAR(30), SUPPLIER_SupplierID INT(10), ORDER_OrderID INT(10). Indexes: ShippingID, SUPPLIER_SupplierID, ORDER_OrderID. Triggers: shipment.
- supplier**: SupplierID INT(10) (PK), SupplierName VARCHAR(15), SupplierPhoneNo VARCHAR(12), LOGIN_DETAILS_loginID INT(10). Indexes: SupplierID, SupplierName, SupplierPhoneNo, LOGIN_DETAILS_loginID.
- supplier_has_item**: SUPPLIER_SupplierID INT(10), ITEM_ProductID INT(10), ITEM_CATEGORY_CategoryID INT(10). Indexes: SUPPLIER_SupplierID, ITEM_ProductID, ITEM_CATEGORY_CategoryID.
- card_details**: CardNumber VARCHAR(30) (PK), CardType VARCHAR(15), NameOnCard VARCHAR(15), ExpiryDate DATE. Indexes: CardNumber, CardType, NameOnCard, ExpiryDate.
- payment**: PaymentID INT(10) (PK), PaymentType VARCHAR(15), ORDER_OrderID INT(10), ORDER_CUSTOMER_CustomerID INT(10), CardDetails_CardNumber VARCHAR(30). Indexes: PaymentID, ORDER_OrderID, ORDER_CUSTOMER_CustomerID, CardDetails_CardNumber.

Relationships are indicated by lines with crow's foot notation. Key attributes are underlined. Some tables have yellow callouts: 'customerorderhistory' and 'revenuefromeachcustomer' pointing to the customer table, and '1 more...' pointing to the item table.

TRIGGERS

- 1) On successful order placed, product quantity will be reduced from the supplier stock.

Once 3 quantities of cameras are ordered successfully. A trigger Product_Quantity is initiated which will reduce the stock in Item table.

In this case, the initial stock for camera was 90. On ordering 3 cameras, the stock will be reduced to 87.

Delimiter \$\$

create trigger Product_Quantity

After INSERT on bostoncraigslist.order_has_item
for each row

Begin

set @productId = NEW.Item_ProductID;

set @newQuantity = NEW.Quantity;

update bostoncraigslist.Item set QuantityPerUnit = QuantityPerUnit -

@newQuantity

where Item.ProductID = @productId;

END;

\$\$

Initial quantity of cameras

	ProductID	ProductName	QuantityPerUnit	UnitPrice	DiscountPercent	CATEGORY_CategoryID
▶	1	closet	23	25	5	1
	2	CellPhone	196	550	25	2
	3	Camera	90	250	20	2
	4	Tshirt	497	15	0	3
	5	Jeans	78	40	10	3
	6	Dress	227	40	5	3
*	NULL	NULL	NULL	NULL	NULL	NULL

Stock after ordering 3 cameras

	ProductID	ProductName	QuantityPerUnit	UnitPrice	DiscountPercent	CATEGORY_CategoryID
▶	1	closet	23	25	5	1
	2	CellPhone	196	550	25	2
	3	Camera	87	250	20	2
	4	Tshirt	497	15	0	3
	5	Jeans	78	40	10	3
	6	Dress	227	40	5	3
*	NULL	NULL	NULL	NULL	NULL	NULL

2) Order status will be updated on change in shipment status.

Once the shipment details are updated by supplier, a trigger is called which will update the order status from the orderdetail table.

Delimiter \$\$

create trigger updateOrderStatus

After UPDATE on bostoncraiglist.shipment
for each row

Begin

UPDATE orderdetail

SET OrderStatus = "Delivered"

WHERE orderdetail.OrderID = new.ORDER_OrderID;

END;

\$\$

Order status before updating shipping details for order 7.

	OrderID	OrderPlacedDate	SUPPLIER_SupplierID	CART_CartID	CART_CUSTOMER_CustomerID	ADDRESS_AddressID	OrderStatus
▶	1	2018-11-27	3	1	1	1	In Process
	2	2018-12-01	3	2	2	2	Delivered
	3	2018-10-18	2	3	3	3	In Process
	4	2018-10-10	1	4	4	4	Delivered
	5	2018-11-27	4	1	1	1	In Process
	6	2018-12-01	2	1	1	1	In Process
	7	2018-12-01	1	3	3	3	In Process

After updating shipping information by supplier for order 7, Order status is updated.

	OrderID	OrderPlacedDate	SUPPLIER_SupplierID	CART_CartID	CART_CUSTOMER_CustomerID	ADDRESS_AddressID	OrderStatus
▶	1	2018-11-27	3	1	1	1	In Process
	2	2018-12-01	3	2	2	2	Delivered
	3	2018-10-18	2	3	3	3	In Process
	4	2018-10-10	1	4	4	4	Delivered
	5	2018-11-27	4	1	1	1	In Process
	6	2018-12-01	2	1	1	1	In Process
	7	2018-12-01	1	3	3	3	Delivered

VIEWS1) Display history of order details.

This view includes all the different orders made by all the customers, product name, quantity ordered, current order status, supplier details for the corresponding orders and the total price for that order.

```
create view CustomerOrderHistory
as
Select Customer.CustomerName AS 'Customer Name', item.productName AS
'Product Name', ordereditem.Quantity AS 'Quantity', ordereditem.ORDER_OrderID
AS 'Order Number',od.OrderStatus AS 'Order Status', supplier.SupplierName AS
'Supplier Name', supplier.SupplierPhoneNo AS 'Supplier Contact Number',
sum((ordereditem.Price*(1-(ordereditem.Discount/100))*ordereditem.Quantity)) AS
'Total Price After Discount'
from customer JOIN
order_has_item ordereditem
on customer.customerID = ordereditem.order_customer_customerID
JOIN item
ON ordereditem.Item_ProductID = item.ProductID
JOIN orderdetail od
ON ordereditem.ORDER_OrderID = od.OrderID
JOIN shipment
ON shipment.ORDER_OrderID = ordereditem.ORDER_OrderID
JOIN supplier
ON shipment.SUPPLIER_SupplierID = supplier.SupplierID
group by Customer.CustomerName,ordereditem.ORDER_OrderID;

select * from CustomerOrderHistory;
```

	Customer Name	Product Name	Quantity	Order Number	Order Status	Supplier Name	Supplier Contact Number	Total Price After Discount
►	Krina	CellPhone	1	1	In Process	Krishna	8573339874	427.5
	Kalp	Dress	3	7	Delivered	Krishna	8573339874	114
	Kaksha	Camera	3	4	Delivered	Shraddha	8573449654	600
	Krina	Jeans	2	6	In Process	Shraddha	8573449654	72
	Mansi	closet	1	2	Delivered	Naman	6178524165	23.75
	Kalp	CellPhone	2	3	In Process	Vaishu	6177536891	825

- 2) A view to generate total revenue generated from each customer and the total revenue generated by all the orders from all the customers.

create view RevenueFromEachCustomer

as

Select Customer.CustomerName AS 'Customer Name', sum((ordereditem.Price*(1-(ordereditem.Discount/100))*ordereditem.Quantity)) AS 'Total Revenue From Each Customer'

from customer JOIN

order_has_item ordereditem

on customer.customerID = ordereditem.order_customer_customerID

group by Customer.CustomerName WITH ROLLUP;

	Customer Name	Total Revenue From Each Customer
▶	Kaksha	600
	Kalp	939
	Krina	499.5
	Mansi	23.75
	NULL	2062.25

User Defined Function

- 1) User defined function to get the unit price for a particular product

DELIMITER //

Create Function PriceInOrder(ProductID int)

Returns float

DETERMINISTIC

Begin

declare price float;

select item.UnitPrice into price from item where item.ProductID = ProductID;

return price;

END;

//

Called in insert statement: order_has_item table

Insert into order_has_item values (2, 2, 3, 2, 1, 1, **PriceInOrder(order_has_item.Item_ProductID)**, 1, DiscountInOrder(order_has_item.Item_ProductID), TotalPriceInOrder(order_has_item.Item_ProductID));

2) User defined function to get the percentage discount for a particular product

```

DELIMITER //
Create Function DiscountInOrder(ProductID int)
Returns float
DETERMINISTIC
Begin
    declare discount float;
    select item.DiscountPercent into discount  from item  where item.ProductID =
ProductID;
    return discount;
END;
//

Insert into order_has_item values (2, 2, 3, 2, 1, 1,
PriceInOrder(order_has_item.Item_ProductID), 1,
DiscountInOrder(order_has_item.Item_ProductID),
TotalPrice(order_has_item.Item_ProductID)
);

```

STORED PROCEDURE1) Stored Procedure to calculate total revenue generated by all the orders at the end of the day.

```

DELIMITER $$
CREATE PROCEDURE TotalRevenuePerDay(IN OrderDate date)
BEGIN
    select ordereditem.ORDER_OrderID AS
    OrderID,sum((ordereditem.Price*(1-
    (ordereditem.Discount/100))*ordereditem.Quantity)) AS
    TotalRevenue from orderdetail JOIN
    order_has_item ordereditem
    on orderdetail.OrderID = ordereditem.Order_OrderID
    where orderdetail.OrderPlacedDate = OrderDate
    group by ordereditem.ORDER_OrderID;

END;
$$

```

	OrderID	Total Revenue Per Day
►	2	23.75
	6	72
	7	114
	NULL	209.75

2) Retrieve products based on category

```

DELIMITER $$
CREATE PROCEDURE productBasedOnCategory(IN categoryName
varchar(45))
BEGIN
    select item.ProductName, item.QuantityPerUnit from item
    JOIN category
    on item.CATEGORY_CategoryID = category.CategoryID
    where category.CategoryName = categoryName;
END;
$$

```

	ProductName	QuantityPerUnit
▶	CellPhone	196
	Camera	87

3) Get the original price and discounted price based on product Name

```

DELIMITER $$
CREATE PROCEDURE discountOnProduct(IN pname varchar(30))
BEGIN
    select ProductName, UnitPrice AS 'Original Price', (UnitPrice*(1
DiscountPercent/100))) AS 'Discounted Price' From item
    where ProductName = pname ;
END;
$$

```

	ProductName	Original Price	Discounted Price
▶	CellPhone	550	412.5

INDEX

- 1) create index prouctName_index on item(ProductName) USING BTREE;
- 2) create index categoryName_index on category(CategoryName) USING BTREE;

ROLE and PRIVILEGES

```

create user 'customer'@localhost identified by 'customer';
GRANT ALL on bostoncraiglist.cart to 'customer'@localhost;
GRANT ALL on bostoncraiglist.orderdetail to 'customer'@localhost;

```



```
GRANT ALL on bostoncraiglist.payment to 'customer'@localhost;  
GRANT ALL on bostoncraiglist.ccarddetails to 'customer'@localhost;
```

```
create user 'supplier'@localhost identified by 'supplier';  
GRANT select on bostoncraiglist.orderdetail to 'supplier'@localhost;  
GRANT ALL on bostoncraiglist.shipment to 'supplier'@localhost;  
GRANT ALL on bostoncraiglist.item to 'supplier'@localhost;
```

Eg.

When supplier tries to view cart details, it throws error

```
1  Select * from cart;
```

Output				
Action Output				
#	Time	Action	Message	Duration / Fetch
1	02:46:58	Select * from cart LIMIT 0, 1000	Error Code: 1142. SELECT command denied to user 'supplier'@'localhost' for table 'cart'	0.000 sec

➔ Dump (backup) of the database



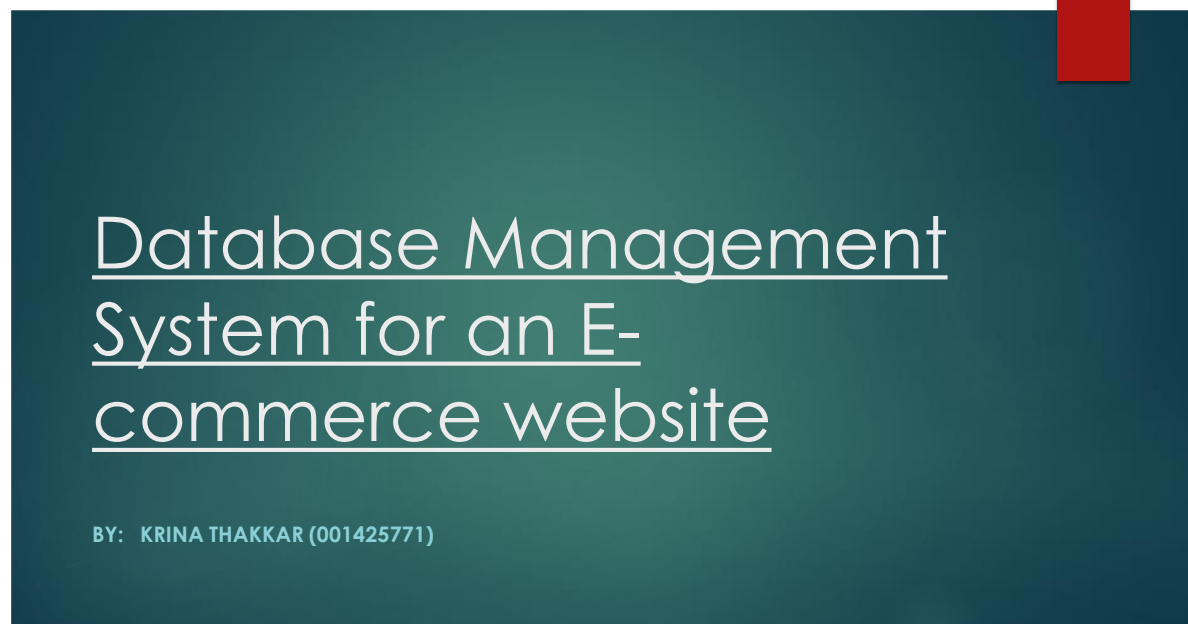
dump.sql

➔ ER Diagram



dmdd_finalProject.mwb

➔ Presentation



APPENDIX

```
-- MySQL dump 10.13 Distrib 8.0.12, for Win64 (x86_64)
--
-- Host: 127.0.0.1 Database: bostoncraiglist
-- -----
-- Server version 8.0.12

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT
*/;
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION
*/;
SET NAMES utf8 ;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS,
UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `address`
--

DROP TABLE IF EXISTS `address`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
SET character_set_client = utf8mb4 ;
CREATE TABLE `address` (
  `AddressID` int(10) unsigned NOT NULL,
  `StreetName` varchar(70) NOT NULL,
  `City` varchar(20) NOT NULL,
  `State` varchar(20) NOT NULL,
  `PostalCode` varchar(5) NOT NULL,
  `CUSTOMER_CustomerID` int(10) unsigned NOT NULL,
  PRIMARY KEY (`AddressID`,`CUSTOMER_CustomerID`),
  KEY `fk_ADDRESS_CUSTOMER1_idx` (`CUSTOMER_CustomerID`),
  CONSTRAINT `fk_ADDRESS_CUSTOMER1` FOREIGN KEY
(`CUSTOMER_CustomerID`) REFERENCES `customer` (`customerid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;
```

```
--
-- Dumping data for table `address`
--

LOCK TABLES `address` WRITE;
/*!40000 ALTER TABLE `address` DISABLE KEYS */;
INSERT INTO `address` VALUES (1,'Boylston St','Boston','MA','02215',1),(2,'Francis
St','Albany','NY','04213',2),(3,'Portland St','Los Angeles','CA','90007',3),(4,'Hemenway
St','Boston','MA','02115',4);
/*!40000 ALTER TABLE `address` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `carddetails`
--

DROP TABLE IF EXISTS `carddetails`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
SET character_set_client = utf8mb4 ;
CREATE TABLE `carddetails` (
  `CardNumber` varchar(30) NOT NULL,
  `CardType` varchar(45) NOT NULL,
  `NameOnCard` varchar(45) NOT NULL,
  `ExpiryDate` date NOT NULL,
  `CUSTOMER_CustomerID` int(10) unsigned NOT NULL,
  PRIMARY KEY (`CardNumber`,`CUSTOMER_CustomerID`),
  KEY `fk_CardDetails_CUSTOMER1_idx` (`CUSTOMER_CustomerID`),
  CONSTRAINT `fk_CardDetails_CUSTOMER1` FOREIGN KEY
(`CUSTOMER_CustomerID`) REFERENCES `customer` (`customerid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `carddetails`
--

LOCK TABLES `carddetails` WRITE;
/*!40000 ALTER TABLE `carddetails` DISABLE KEYS */;
INSERT INTO `carddetails` VALUES ('8349 5028 3593 3490','Credit Card','Krina
Thakkar','2028-11-11',1),('9535 2613 3546 9237','Debit Card','Mansi Pandya','2020-12-
12',2);
/*!40000 ALTER TABLE `carddetails` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `cart`
```

--

```
DROP TABLE IF EXISTS `cart`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
SET character_set_client = utf8mb4 ;
CREATE TABLE `cart` (
  `CartID` int(10) unsigned NOT NULL,
  `CUSTOMER_CustomerID` int(10) unsigned NOT NULL,
  PRIMARY KEY (`CartID`,`CUSTOMER_CustomerID`),
  KEY `fk_CART_CUSTOMER1_idx` (`CUSTOMER_CustomerID`),
  CONSTRAINT `fk_CART_CUSTOMER1` FOREIGN KEY
(`CUSTOMER_CustomerID`) REFERENCES `customer` (`customerid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;
```

--

-- Dumping data for table `cart`

--

```
LOCK TABLES `cart` WRITE;
/*!40000 ALTER TABLE `cart` DISABLE KEYS */;
INSERT INTO `cart` VALUES (1,1),(2,2),(3,3),(4,4);
/*!40000 ALTER TABLE `cart` ENABLE KEYS */;
UNLOCK TABLES;
```

--

-- Table structure for table `cart_has_item`

--

```
DROP TABLE IF EXISTS `cart_has_item`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
SET character_set_client = utf8mb4 ;
CREATE TABLE `cart_has_item` (
  `CART_CartID` int(10) unsigned NOT NULL,
  `CART_CUSTOMER_CustomerID` int(10) unsigned NOT NULL,
  `ITEM_ProductID` int(10) unsigned NOT NULL,
  PRIMARY KEY
(`ITEM_ProductID`,`CART_CartID`,`CART_CUSTOMER_CustomerID`),
  KEY `fk_CART_has_PRODUCT_PRODUCT1_idx` (`ITEM_ProductID`),
  KEY `fk_CART_has_PRODUCT_CART1_idx`
(`CART_CartID`,`CART_CUSTOMER_CustomerID`),
  CONSTRAINT `fk_CART_has_PRODUCT_CART1` FOREIGN KEY (`CART_CartID`,
`CART_CUSTOMER_CustomerID`) REFERENCES `cart` (`cartid`,
`customer_customerid`),
  CONSTRAINT `fk_CART_has_PRODUCT_PRODUCT1` FOREIGN KEY
(`ITEM_ProductID`) REFERENCES `item` (`ProductID`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `cart_has_item`
--

LOCK TABLES `cart_has_item` WRITE;
/*!40000 ALTER TABLE `cart_has_item` DISABLE KEYS */;
INSERT INTO `cart_has_item` VALUES
(2,2,1),(4,4,1),(1,1,2),(3,3,2),(1,1,3),(2,2,4),(4,4,5),(4,4,6);
/*!40000 ALTER TABLE `cart_has_item` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `category`
--

DROP TABLE IF EXISTS `category`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
SET character_set_client = utf8mb4 ;
CREATE TABLE `category` (
  `CategoryID` int(10) unsigned NOT NULL,
  `CategoryName` varchar(45) NOT NULL,
  PRIMARY KEY (`CategoryID`),
  KEY `categoryName_index` (`CategoryName`) USING BTREE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `category`
--

LOCK TABLES `category` WRITE;
/*!40000 ALTER TABLE `category` DISABLE KEYS */;
INSERT INTO `category` VALUES (3,'Clothes'),(2,'Electronics'),(1,'HomeAppliances');
/*!40000 ALTER TABLE `category` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `customer`
--

DROP TABLE IF EXISTS `customer`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
SET character_set_client = utf8mb4 ;
```

```
CREATE TABLE `customer` (  
  `CustomerID` int(10) unsigned NOT NULL,  
  `CustomerName` varchar(45) DEFAULT NULL,  
  `ContactNumber` varchar(12) DEFAULT NULL,  
  `LOGIN DETAILS_loginID` int(10) unsigned NOT NULL,  
  PRIMARY KEY (`CustomerID`,`LOGIN DETAILS_loginID`),  
  KEY `fk_CUSTOMER_LOGIN DETAILS1_idx` (`LOGIN DETAILS_loginID`),  
  CONSTRAINT `fk_CUSTOMER_LOGIN DETAILS1` FOREIGN KEY (`LOGIN  
DETAILS_loginID`) REFERENCES `logindetails` (`loginID`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
/*!40101 SET character_set_client = @saved_cs_client */;  
  
--  
-- Dumping data for table `customer`  
--  
  
LOCK TABLES `customer` WRITE;  
/*!40000 ALTER TABLE `customer` DISABLE KEYS */;  
INSERT INTO `customer` VALUES  
(1,'Krina','8574249361',1),(2,'Mansi','8574249370',2),(3,'Kalp','6174865297',6),(4,'Kaksh  
a','6173287954',8);  
/*!40000 ALTER TABLE `customer` ENABLE KEYS */;  
UNLOCK TABLES;  
  
--  
-- Temporary view structure for view `customerorderhistory`  
--  
  
DROP TABLE IF EXISTS `customerorderhistory`;  
/*!50001 DROP VIEW IF EXISTS `customerorderhistory` */;  
SET @saved_cs_client = @@character_set_client;  
SET character_set_client = utf8mb4;  
/*!50001 CREATE VIEW `customerorderhistory` AS SELECT  
  1 AS `Customer Name`,  
  1 AS `Product Name`,  
  1 AS `Quantity`,  
  1 AS `Order Number`,  
  1 AS `Order Status`,  
  1 AS `Supplier Name`,  
  1 AS `Supplier Contact Number`,  
  1 AS `Total Price After Discount` */;  
SET character_set_client = @saved_cs_client;  
  
--  
-- Table structure for table `item`  
--
```

```
DROP TABLE IF EXISTS `item`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
SET character_set_client = utf8mb4 ;
CREATE TABLE `item` (
  `ProductID` int(10) unsigned NOT NULL,
  `ProductName` varchar(45) DEFAULT NULL,
  `QuantityPerUnit` int(10) unsigned DEFAULT NULL,
  `UnitPrice` float DEFAULT NULL,
  `DiscountPercent` float DEFAULT '0',
  `CATEGORY_CategoryID` int(10) unsigned NOT NULL,
  PRIMARY KEY (`ProductID`,`CATEGORY_CategoryID`),
  KEY `fk_PRODUCT_CATEGORY_idx` (`CATEGORY_CategoryID`),
  KEY `prouctName_index` (`ProductName`) USING BTREE,
  CONSTRAINT `fk_PRODUCT_CATEGORY` FOREIGN KEY
(`CATEGORY_CategoryID`) REFERENCES `category` (`categoryid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `item`
--

LOCK TABLES `item` WRITE;
/*!40000 ALTER TABLE `item` DISABLE KEYS */;
INSERT INTO `item` VALUES
(1,'closet',23,25,5,1),(2,'CellPhone',196,550,25,2),(3,'Camera',87,250,20,2),(4,'Tshirt',49
7,15,0,3),(5,'Jeans',78,40,10,3),(6,'Dress',227,40,5,3);
/*!40000 ALTER TABLE `item` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `logindetails`
--

DROP TABLE IF EXISTS `logindetails`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
SET character_set_client = utf8mb4 ;
CREATE TABLE `logindetails` (
  `loginID` int(10) unsigned NOT NULL,
  `Username` varchar(45) NOT NULL,
  `Passwd` varchar(45) NOT NULL,
  PRIMARY KEY (`loginID`,`Username`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;
```



```
--
-- Dumping data for table `logindetails`
--

LOCK TABLES `logindetails` WRITE;
/*!40000 ALTER TABLE `logindetails` DISABLE KEYS */;
INSERT INTO `logindetails` VALUES
(1,'Krina','krina1'),(2,'Mansi','mansi1'),(3,'Krisha','krisha1'),(4,'Shraddha','shraddha1'),(5,'
Naman','naman1'),(6,'Kalp','kalp1'),(7,'Vaishu','vaishu1'),(8,'Kaksha','kaksha1'),(9,'shital',
'shital1'),(10,'john','john@123');
/*!40000 ALTER TABLE `logindetails` ENABLE KEYS */;
UNLOCK TABLES;
/*!50003 SET @saved_cs_client      = @@character_set_client */;
/*!50003 SET @saved_cs_results    = @@character_set_results */;
/*!50003 SET @saved_col_connection = @@collation_connection */;
/*!50003 SET character_set_client = utf8mb4 */;
/*!50003 SET character_set_results = utf8mb4 */;
/*!50003 SET collation_connection = utf8mb4_0900_ai_ci */;
/*!50003 SET @saved_sql_mode      = @@sql_mode */;
/*!50003 SET sql_mode              =
'STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION' */;
DELIMITER ;;
/*!50003 CREATE*/ /*!50017 DEFINER=`root` @`localhost`*/ /*!50003 TRIGGER
`logindetails` AFTER UPDATE ON `logindetails` FOR EACH ROW Begin
    SET @loginid = loginID;
    SET @user = username;
    INSERT INTO login_logs VALUES (@loginid,@user,now());
END */;;
DELIMITER ;
/*!50003 SET sql_mode              = @saved_sql_mode */;
/*!50003 SET character_set_client  = @saved_cs_client */;
/*!50003 SET character_set_results = @saved_cs_results */;
/*!50003 SET collation_connection = @saved_col_connection */;

--
-- Table structure for table `order_has_item`
--

DROP TABLE IF EXISTS `order_has_item`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
SET character_set_client = utf8mb4 ;
CREATE TABLE `order_has_item` (
  `ORDER_OrderID` int(10) unsigned NOT NULL,
  `ORDER_CUSTOMER_CustomerID` int(10) unsigned NOT NULL,
  `ORDER_SUPPLIER_SupplierID` int(10) unsigned NOT NULL,
  `ORDER_SHIPMENT_ShippingID` int(10) unsigned NOT NULL,
```

```

`ITEM_ProductID` int(10) unsigned NOT NULL,
`ITEM_CATEGORY_CategoryID` int(10) unsigned NOT NULL,
`Price` float NOT NULL,
`Quantity` smallint(50) unsigned NOT NULL,
`Discount` float NOT NULL,
`Total` float DEFAULT NULL,
PRIMARY KEY
(`ORDER_OrderID`,`ORDER_CUSTOMER_CustomerID`,`ORDER_SUPPLIER_Suppli
erID`,`ORDER_SHIPMENT_ShippingID`,`ITEM_ProductID`,`ITEM_CATEGORY_Categ
oryID`),
KEY `fk_ORDER_has_PRODUCT_PRODUCT1_idx`
(`ITEM_ProductID`,`ITEM_CATEGORY_CategoryID`),
KEY `fk_ORDER_has_PRODUCT_ORDER1_idx`
(`ORDER_OrderID`,`ORDER_CUSTOMER_CustomerID`,`ORDER_SUPPLIER_Suppli
erID`,`ORDER_SHIPMENT_ShippingID`),
KEY `fk_ORDER_has_PRODUCT_ORDER1`
(`ORDER_OrderID`,`ORDER_SUPPLIER_SupplierID`),
CONSTRAINT `fk_ORDER_has_PRODUCT_ORDER1` FOREIGN KEY
(`ORDER_OrderID`,`ORDER_SUPPLIER_SupplierID`) REFERENCES `orderdetail`
(`orderid`,`supplier_supplierid`),
CONSTRAINT `fk_ORDER_has_PRODUCT_PRODUCT1` FOREIGN KEY
(`ITEM_ProductID`,`ITEM_CATEGORY_CategoryID`) REFERENCES `item`
(`productid`,`category_categoryid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `order_has_item`
--

LOCK TABLES `order_has_item` WRITE;
/*!40000 ALTER TABLE `order_has_item` DISABLE KEYS */;
INSERT INTO `order_has_item` VALUES
(1,1,3,1,2,2,550,1,25,NULL),(1,1,3,1,4,3,15,1,0,NULL),(2,2,3,2,1,1,25,1,5,NULL),(3,3,2,
3,2,2,550,2,25,NULL),(4,4,1,4,3,2,250,3,20,NULL),(6,1,2,6,5,3,40,2,10,NULL),(7,3,1,7,6
,3,40,3,5,NULL);
/*!40000 ALTER TABLE `order_has_item` ENABLE KEYS */;
UNLOCK TABLES;
/*!50003 SET @saved_cs_client = @@character_set_client */;
/*!50003 SET @saved_cs_results = @@character_set_results */;
/*!50003 SET @saved_col_connection = @@collation_connection */;
/*!50003 SET character_set_client = utf8mb4 */;
/*!50003 SET character_set_results = utf8mb4 */;
/*!50003 SET collation_connection = utf8mb4_0900_ai_ci */;
/*!50003 SET @saved_sql_mode = @@sql_mode */;

```

```

/*!50003 SET sql_mode            =
'STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION' */ ;
DELIMITER ;;
/*!50003 CREATE*/ /*!50017 DEFINER=`root`@`localhost`*/ /*!50003 TRIGGER
`Product_Quantity` AFTER INSERT ON `order_has_item` FOR EACH ROW Begin
    set @productId = NEW.Item_ProductID;
    set @newQuantity = NEW.Quantity;
    update bostoncraiglist.Item set QuantityPerUnit = QuantityPerUnit - @newQuantity
    where Item.ProductID = @productID;
END */;;
DELIMITER ;
/*!50003 SET sql_mode            = @saved_sql_mode */ ;
/*!50003 SET character_set_client = @saved_cs_client */ ;
/*!50003 SET character_set_results = @saved_cs_results */ ;
/*!50003 SET collation_connection = @saved_col_connection */ ;

--
-- Table structure for table `orderdetail`
--

DROP TABLE IF EXISTS `orderdetail`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
SET character_set_client = utf8mb4 ;
CREATE TABLE `orderdetail` (
  `OrderID` int(10) unsigned NOT NULL,
  `OrderPlacedDate` date NOT NULL,
  `SUPPLIER_SupplierID` int(10) unsigned NOT NULL,
  `CART_CartID` int(10) unsigned NOT NULL,
  `CART_CUSTOMER_CustomerID` int(10) unsigned NOT NULL,
  `ADDRESS_AddressID` int(10) unsigned NOT NULL,
  `OrderStatus` varchar(30) DEFAULT 'In Process',
  PRIMARY KEY
(`OrderID`,`SUPPLIER_SupplierID`,`CART_CartID`,`CART_CUSTOMER_CustomerID`,
`ADDRESS_AddressID`),
  KEY `fk_ORDER_SUPPLIER1_idx` (`SUPPLIER_SupplierID`),
  KEY `fk_ORDER_CART1_idx` (`CART_CartID`,`CART_CUSTOMER_CustomerID`),
  KEY `fk_ORDER_ADDRESS1_idx` (`ADDRESS_AddressID`),
  CONSTRAINT `fk_ORDER_ADDRESS1` FOREIGN KEY (`ADDRESS_AddressID`)
REFERENCES `address` (`addressid`),
  CONSTRAINT `fk_ORDER_CART1` FOREIGN KEY (`CART_CartID`,
`CART_CUSTOMER_CustomerID`) REFERENCES `cart` (`cartid`,
`customer_customerid`),
  CONSTRAINT `fk_ORDER_SUPPLIER1` FOREIGN KEY (`SUPPLIER_SupplierID`)
REFERENCES `supplier` (`supplierid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

```

```
--
-- Dumping data for table `orderdetail`
--

LOCK TABLES `orderdetail` WRITE;
/*!40000 ALTER TABLE `orderdetail` DISABLE KEYS */;
INSERT INTO `orderdetail` VALUES (1,'2018-11-27',3,1,1,1,'In Process'),(2,'2018-12-01',3,2,2,2,'Delivered'),(3,'2018-10-18',2,3,3,3,'In Process'),(4,'2018-10-10',1,4,4,4,'Delivered'),(5,'2018-11-27',4,1,1,1,'In Process'),(6,'2018-12-01',2,1,1,1,'In Process'),(7,'2018-12-01',1,3,3,3,'Delivered');
/*!40000 ALTER TABLE `orderdetail` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `payment`
--

DROP TABLE IF EXISTS `payment`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
SET character_set_client = utf8mb4 ;
CREATE TABLE `payment` (
  `PaymentID` int(10) unsigned NOT NULL,
  `PaymentType` varchar(45) NOT NULL,
  `ORDER_OrderID` int(10) unsigned NOT NULL,
  `ORDER_CUSTOMER_CustomerID` int(10) unsigned NOT NULL,
  `CardDetails_CardNumber` varchar(30) NOT NULL,
  PRIMARY KEY
(`PaymentID`,`ORDER_OrderID`,`ORDER_CUSTOMER_CustomerID`,`CardDetails_CardNumber`),
  UNIQUE KEY `ORDER_CUSTOMER_CustomerID_UNIQUE`
(`ORDER_CUSTOMER_CustomerID`),
  UNIQUE KEY `CardDetails_CardNumber_UNIQUE` (`CardDetails_CardNumber`),
  KEY `fk_PAYMENT_ORDER1_idx`
(`ORDER_OrderID`,`ORDER_CUSTOMER_CustomerID`),
  KEY `fk_PAYMENT_CardDetails1_idx` (`CardDetails_CardNumber`),
  CONSTRAINT `fk_PAYMENT_CardDetails1` FOREIGN KEY
(`CardDetails_CardNumber`) REFERENCES `carddetails` (`cardnumber`),
  CONSTRAINT `fk_PAYMENT_ORDER1` FOREIGN KEY (`ORDER_OrderID`)
REFERENCES `orderdetail` (`orderid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `payment`
--
```

```
LOCK TABLES `payment` WRITE;
/*!40000 ALTER TABLE `payment` DISABLE KEYS */;
INSERT INTO `payment` VALUES (1,'card',1,1,'8349 5028 3593
3490'),(2,'card',2,2,'9535 2613 3546 9237');
/*!40000 ALTER TABLE `payment` ENABLE KEYS */;
UNLOCK TABLES;
```

```
--
-- Temporary view structure for view `revenuefromeachcustomer`
--
```

```
DROP TABLE IF EXISTS `revenuefromeachcustomer`;
/*!50001 DROP VIEW IF EXISTS `revenuefromeachcustomer`*/;
SET @saved_cs_client = @@character_set_client;
SET character_set_client = utf8mb4;
/*!50001 CREATE VIEW `revenuefromeachcustomer` AS SELECT
  1 AS `Customer Name`,
  1 AS `Total Revenue From Each Customer`*/;
SET character_set_client = @saved_cs_client;
```

```
--
-- Table structure for table `shipment`
--
```

```
DROP TABLE IF EXISTS `shipment`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
SET character_set_client = utf8mb4 ;
CREATE TABLE `shipment` (
  `ShippingID` int(10) unsigned NOT NULL,
  `ShipmentStatus` varchar(30) NOT NULL,
  `SUPPLIER_SupplierID` int(10) unsigned NOT NULL,
  `ORDER_OrderID` int(10) unsigned NOT NULL,
  PRIMARY KEY (`ShippingID`,`SUPPLIER_SupplierID`,`ORDER_OrderID`),
  KEY `fk_SHIPMENT_SUPPLIER1_idx` (`SUPPLIER_SupplierID`),
  KEY `fk_SHIPMENT_ORDER1_idx` (`ORDER_OrderID`),
  CONSTRAINT `fk_SHIPMENT_ORDER1` FOREIGN KEY (`ORDER_OrderID`)
REFERENCES `orderdetail` (`orderid`),
  CONSTRAINT `fk_SHIPMENT_SUPPLIER1` FOREIGN KEY
(`SUPPLIER_SupplierID`) REFERENCES `supplier` (`supplierid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;
```

```
--
-- Dumping data for table `shipment`
--
```

```

LOCK TABLES `shipment` WRITE;
/*!40000 ALTER TABLE `shipment` DISABLE KEYS */;
INSERT INTO `shipment` VALUES (1,'In process',1,1),(2,'Delivered',3,2),(3,'In
process',4,3),(4,'Delivered',2,4),(6,'In Process',2,6),(7,'Delivered',1,7);
/*!40000 ALTER TABLE `shipment` ENABLE KEYS */;
UNLOCK TABLES;
/*!50003 SET @saved_cs_client      = @@character_set_client */;
/*!50003 SET @saved_cs_results    = @@character_set_results */;
/*!50003 SET @saved_col_connection = @@collation_connection */;
/*!50003 SET character_set_client  = utf8mb4 */;
/*!50003 SET character_set_results = utf8mb4 */;
/*!50003 SET collation_connection = utf8mb4_0900_ai_ci */;
/*!50003 SET @saved_sql_mode      = @@sql_mode */;
/*!50003 SET sql_mode              =
'STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION' */;
DELIMITER ;;
/*!50003 CREATE*/ /*!50017 DEFINER=`root`@`localhost`*/ /*!50003 TRIGGER
`updateOrderStatus` AFTER UPDATE ON `shipment` FOR EACH ROW Begin
    UPDATE orderdetail
    SET OrderStatus = "Delivered"
    WHERE orderdetail.OrderID = new.ORDER_OrderID;
END */;;
DELIMITER ;
/*!50003 SET sql_mode              = @saved_sql_mode */;
/*!50003 SET character_set_client  = @saved_cs_client */;
/*!50003 SET character_set_results = @saved_cs_results */;
/*!50003 SET collation_connection = @saved_col_connection */;

--
-- Table structure for table `supplier`
--

DROP TABLE IF EXISTS `supplier`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
SET character_set_client = utf8mb4 ;
CREATE TABLE `supplier` (
  `SupplierID` int(10) unsigned NOT NULL,
  `SupplierName` varchar(45) DEFAULT NULL,
  `SupplierPhoneNo` varchar(12) DEFAULT NULL,
  `LOGIN DETAILS_loginID` int(10) unsigned NOT NULL,
  PRIMARY KEY (`SupplierID`,`LOGIN DETAILS_loginID`),
  KEY `fk_SUPPLIER_LOGIN DETAILS1_idx` (`LOGIN DETAILS_loginID`),
  CONSTRAINT `fk_SUPPLIER_LOGIN DETAILS1` FOREIGN KEY (`LOGIN
DETAILS_loginID`) REFERENCES `logindetails` (`loginID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `supplier`
--

LOCK TABLES `supplier` WRITE;
/*!40000 ALTER TABLE `supplier` DISABLE KEYS */;
INSERT INTO `supplier` VALUES
(1,'Krisha','8573339874',3),(2,'Shraddha','8573449654',4),(3,'Naman','6178524165',5),(4
,'Vaishu','6177536891',7);
/*!40000 ALTER TABLE `supplier` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `supplier_has_item`
--

DROP TABLE IF EXISTS `supplier_has_item`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
SET character_set_client = utf8mb4 ;
CREATE TABLE `supplier_has_item` (
  `SUPPLIER_SupplierID` int(10) unsigned NOT NULL,
  `ITEM_ProductID` int(10) unsigned NOT NULL,
  `ITEM_CATEGORY_CategoryID` int(10) unsigned NOT NULL,
  PRIMARY KEY
(`SUPPLIER_SupplierID`,`ITEM_ProductID`,`ITEM_CATEGORY_CategoryID`),
  KEY `fk_SUPPLIER_has_ITEM_ITEM1_idx`
(`ITEM_ProductID`,`ITEM_CATEGORY_CategoryID`),
  KEY `fk_SUPPLIER_has_ITEM_SUPPLIER1_idx` (`SUPPLIER_SupplierID`),
  CONSTRAINT `fk_SUPPLIER_has_ITEM_ITEM1` FOREIGN KEY (`ITEM_ProductID`,
`ITEM_CATEGORY_CategoryID`) REFERENCES `item` (`ProductID`,
`CATEGORY_CategoryID`),
  CONSTRAINT `fk_SUPPLIER_has_ITEM_SUPPLIER1` FOREIGN KEY
(`SUPPLIER_SupplierID`) REFERENCES `supplier` (`supplierid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `supplier_has_item`
--

LOCK TABLES `supplier_has_item` WRITE;
/*!40000 ALTER TABLE `supplier_has_item` DISABLE KEYS */;
INSERT INTO `supplier_has_item` VALUES
(1,3,2),(1,5,3),(2,2,2),(2,4,3),(3,1,1),(3,6,3),(4,1,1);
```

```
/*!40000 ALTER TABLE `supplier_has_item` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Dumping events for database 'bostoncraigslist'
--
--
-- Dumping routines for database 'bostoncraigslist'
--
/*!50003 DROP FUNCTION IF EXISTS `DiscountInOrder` */;
/*!50003 SET @saved_cs_client      = @@character_set_client */;
/*!50003 SET @saved_cs_results    = @@character_set_results */;
/*!50003 SET @saved_col_connection = @@collation_connection */;
/*!50003 SET character_set_client = utf8mb4 */;
/*!50003 SET character_set_results = utf8mb4 */;
/*!50003 SET collation_connection = utf8mb4_0900_ai_ci */;
/*!50003 SET @saved_sql_mode      = @@sql_mode */;
/*!50003 SET sql_mode              =
'STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION' */;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` FUNCTION `DiscountInOrder`(ProductID int)
RETURNS float
    DETERMINISTIC
Begin
    declare discount float;
    select item.DiscountPercent into discount from item where item.ProductID =
ProductID;
    return discount;
END ;;
DELIMITER ;
/*!50003 SET sql_mode              = @saved_sql_mode */;
/*!50003 SET character_set_client  = @saved_cs_client */;
/*!50003 SET character_set_results = @saved_cs_results */;
/*!50003 SET collation_connection = @saved_col_connection */;
/*!50003 DROP FUNCTION IF EXISTS `PriceInOrder` */;
/*!50003 SET @saved_cs_client      = @@character_set_client */;
/*!50003 SET @saved_cs_results    = @@character_set_results */;
/*!50003 SET @saved_col_connection = @@collation_connection */;
/*!50003 SET character_set_client  = utf8mb4 */;
/*!50003 SET character_set_results = utf8mb4 */;
/*!50003 SET collation_connection = utf8mb4_0900_ai_ci */;
/*!50003 SET @saved_sql_mode      = @@sql_mode */;
/*!50003 SET sql_mode              =
'STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION' */;
DELIMITER ;
```



```

CREATE DEFINER=`root`@`localhost` FUNCTION `PriceInOrder`(ProductID int)
RETURNS float
    DETERMINISTIC
Begin
    declare abc float;
    select item.UnitPrice into abc from item where item.ProductID = ProductID;
    return abc;
END ;;
DELIMITER ;
/*!50003 SET sql_mode            = @saved_sql_mode */;
/*!50003 SET character_set_client = @saved_cs_client */;
/*!50003 SET character_set_results = @saved_cs_results */;
/*!50003 SET collation_connection = @saved_col_connection */;
/*!50003 DROP FUNCTION IF EXISTS `TotalPrice` */;
/*!50003 SET @saved_cs_client      = @@character_set_client */;
/*!50003 SET @saved_cs_results     = @@character_set_results */;
/*!50003 SET @saved_col_connection = @@collation_connection */;
/*!50003 SET character_set_client = utf8mb4 */;
/*!50003 SET character_set_results = utf8mb4 */;
/*!50003 SET collation_connection = utf8mb4_0900_ai_ci */;
/*!50003 SET @saved_sql_mode       = @@sql_mode */;
/*!50003 SET sql_mode              =
'STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION' */;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` FUNCTION `TotalPrice`(ProductID int)
RETURNS float
    DETERMINISTIC
Begin
    declare price float;
    declare total float;
    select ((Price*(1-(Discount/100))*Quantity)) into total from order_has_item where
order_has_item.Item_ProductID=ProductID;
    return total;
END ;;
DELIMITER ;
/*!50003 SET sql_mode            = @saved_sql_mode */;
/*!50003 SET character_set_client = @saved_cs_client */;
/*!50003 SET character_set_results = @saved_cs_results */;
/*!50003 SET collation_connection = @saved_col_connection */;
/*!50003 DROP PROCEDURE IF EXISTS `discountOnProduct` */;
/*!50003 SET @saved_cs_client      = @@character_set_client */;
/*!50003 SET @saved_cs_results     = @@character_set_results */;
/*!50003 SET @saved_col_connection = @@collation_connection */;
/*!50003 SET character_set_client = utf8mb4 */;
/*!50003 SET character_set_results = utf8mb4 */;
/*!50003 SET collation_connection = utf8mb4_0900_ai_ci */;

```

```
/*!50003 SET @saved_sql_mode      = @@sql_mode */ ;
/*!50003 SET sql_mode              =
'STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION' */ ;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` PROCEDURE `discountOnProduct`(IN pname
varchar(30))
BEGIN
    select ProductName, UnitPrice AS 'Original Price',(UnitPrice*(1-
(DiscountPercent/100))) AS 'Discounted Price' From item
    where ProductName = pname ;
END ;;
DELIMITER ;
/*!50003 SET sql_mode              = @saved_sql_mode */ ;
/*!50003 SET character_set_client  = @saved_cs_client */ ;
/*!50003 SET character_set_results = @saved_cs_results */ ;
/*!50003 SET collation_connection = @saved_col_connection */ ;
/*!50003 DROP PROCEDURE IF EXISTS `FilterOnDiscount` */ ;
/*!50003 SET @saved_cs_client      = @@character_set_client */ ;
/*!50003 SET @saved_cs_results    = @@character_set_results */ ;
/*!50003 SET @saved_col_connection = @@collation_connection */ ;
/*!50003 SET character_set_client  = utf8mb4 */ ;
/*!50003 SET character_set_results = utf8mb4 */ ;
/*!50003 SET collation_connection  = utf8mb4_0900_ai_ci */ ;
/*!50003 SET @saved_sql_mode      = @@sql_mode */ ;
/*!50003 SET sql_mode              =
'STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION' */ ;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` PROCEDURE `FilterOnDiscount`(IN
categoryName varchar(45))
BEGIN
    select item.ProductName, category.CategoryName from item
    JOIN category
    on item.CATEGORY_CategoryID = category.CategoryID
    where category.CategoryName = categoryName;
END ;;
DELIMITER ;
/*!50003 SET sql_mode              = @saved_sql_mode */ ;
/*!50003 SET character_set_client  = @saved_cs_client */ ;
/*!50003 SET character_set_results = @saved_cs_results */ ;
/*!50003 SET collation_connection = @saved_col_connection */ ;
/*!50003 DROP PROCEDURE IF EXISTS `productBasedOnCategory` */ ;
/*!50003 SET @saved_cs_client      = @@character_set_client */ ;
/*!50003 SET @saved_cs_results    = @@character_set_results */ ;
/*!50003 SET @saved_col_connection = @@collation_connection */ ;
/*!50003 SET character_set_client  = utf8mb4 */ ;
/*!50003 SET character_set_results = utf8mb4 */ ;
```

```
/*!50003 SET collation_connection = utf8mb4_0900_ai_ci */ ;
/*!50003 SET @saved_sql_mode      = @@sql_mode */ ;
/*!50003 SET sql_mode              =
'STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION' */ ;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` PROCEDURE `productBasedOnCategory`(IN
categoryName varchar(45))
BEGIN
    select item.ProductName, item.QuantityPerUnit from item
    JOIN category
    on item.CATEGORY_CategoryID = category.CategoryID
    where category.CategoryName = categoryName;
END ;;
DELIMITER ;
/*!50003 SET sql_mode              = @saved_sql_mode */ ;
/*!50003 SET character_set_client  = @saved_cs_client */ ;
/*!50003 SET character_set_results = @saved_cs_results */ ;
/*!50003 SET collation_connection  = @saved_col_connection */ ;
/*!50003 DROP PROCEDURE IF EXISTS `TotalRevenuePerDay` */;
/*!50003 SET @saved_cs_client      = @@character_set_client */ ;
/*!50003 SET @saved_cs_results    = @@character_set_results */ ;
/*!50003 SET @saved_col_connection = @@collation_connection */ ;
/*!50003 SET character_set_client  = utf8mb4 */ ;
/*!50003 SET character_set_results = utf8mb4 */ ;
/*!50003 SET collation_connection  = utf8mb4_0900_ai_ci */ ;
/*!50003 SET @saved_sql_mode      = @@sql_mode */ ;
/*!50003 SET sql_mode              =
'STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION' */ ;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` PROCEDURE `TotalRevenuePerDay`(IN
OrderDate date)
BEGIN
    select ordereditem.ORDER_OrderID AS OrderID,sum((ordereditem.Price*(1-
(ordereditem.Discount/100))*ordereditem.Quantity)) AS 'Total Revenue Per Day' from
orderdetail JOIN
    order_has_item ordereditem
    on orderdetail.OrderID = ordereditem.Order_OrderID
    where orderdetail.OrderPlacedDate = OrderDate
    group by ordereditem.ORDER_OrderID with rollup;
END ;;
DELIMITER ;
/*!50003 SET sql_mode              = @saved_sql_mode */ ;
/*!50003 SET character_set_client  = @saved_cs_client */ ;
/*!50003 SET character_set_results = @saved_cs_results */ ;
/*!50003 SET collation_connection  = @saved_col_connection */ ;
```

--

-- Final view structure for view `customerorderhistory`

--

```

/*!50001 DROP VIEW IF EXISTS `customerorderhistory`*/;
/*!50001 SET @saved_cs_client      = @@character_set_client */;
/*!50001 SET @saved_cs_results    = @@character_set_results */;
/*!50001 SET @saved_col_connection = @@collation_connection */;
/*!50001 SET character_set_client  = utf8mb4 */;
/*!50001 SET character_set_results = utf8mb4 */;
/*!50001 SET collation_connection = utf8mb4_0900_ai_ci */;
/*!50001 CREATE ALGORITHM=UNDEFINED */
/*!50013 DEFINER=`root`@`localhost` SQL SECURITY DEFINER */
/*!50001 VIEW `customerorderhistory` AS select `customer`.`CustomerName` AS
`Customer Name`,`item`.`ProductName` AS `Product Name`,`ordereditem`.`Quantity`
AS `Quantity`,`ordereditem`.`ORDER_OrderID` AS `Order Number`,`od`.`OrderStatus`
AS `Order Status`,`supplier`.`SupplierName` AS `Supplier
Name`,`supplier`.`SupplierPhoneNo` AS `Supplier Contact
Number`,sum(((`ordereditem`.`Price` * (1 - (`ordereditem`.`Discount` / 100)))) *
`ordereditem`.`Quantity`)) AS `Total Price After Discount` from ((((`customer` join
`order_has_item` `ordereditem` on((`customer`.`CustomerID` =
`ordereditem`.`ORDER_CUSTOMER_CustomerID`))) join `item`
on((`ordereditem`.`ITEM_ProductID` = `item`.`ProductID`))) join `orderdetail` `od`
on((`ordereditem`.`ORDER_OrderID` = `od`.`OrderID`))) join `shipment`
on((`shipment`.`ORDER_OrderID` = `ordereditem`.`ORDER_OrderID`))) join `supplier`
on((`shipment`.`SUPPLIER_SupplierID` = `supplier`.`SupplierID`))) group by
`customer`.`CustomerName`,`ordereditem`.`ORDER_OrderID` */;
/*!50001 SET character_set_client  = @saved_cs_client */;
/*!50001 SET character_set_results = @saved_cs_results */;
/*!50001 SET collation_connection = @saved_col_connection */;

```

--

-- Final view structure for view `revenuefromeachcustomer`

--

```

/*!50001 DROP VIEW IF EXISTS `revenuefromeachcustomer`*/;
/*!50001 SET @saved_cs_client      = @@character_set_client */;
/*!50001 SET @saved_cs_results    = @@character_set_results */;
/*!50001 SET @saved_col_connection = @@collation_connection */;
/*!50001 SET character_set_client  = utf8mb4 */;
/*!50001 SET character_set_results = utf8mb4 */;
/*!50001 SET collation_connection = utf8mb4_0900_ai_ci */;
/*!50001 CREATE ALGORITHM=UNDEFINED */
/*!50013 DEFINER=`root`@`localhost` SQL SECURITY DEFINER */
/*!50001 VIEW `revenuefromeachcustomer` AS select `customer`.`CustomerName` AS
`Customer Name`,sum(((`ordereditem`.`Price` * (1 - (`ordereditem`.`Discount` / 100)))) *

```

```
`ordereditem`.`Quantity`)) AS `Total Revenue From Each Customer` from (`customer`  
join `order_has_item` `ordereditem` on((`customer`.`CustomerID` =  
`ordereditem`.`ORDER_CUSTOMER_CustomerID`))) group by  
`customer`.`CustomerName` with rollup */;  
/*!50001 SET character_set_client    = @saved_cs_client */;  
/*!50001 SET character_set_results   = @saved_cs_results */;  
/*!50001 SET collation_connection    = @saved_col_connection */;  
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;  
  
/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;  
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;  
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;  
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;  
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS  
*/;  
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;  
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
```

-- Dump completed on 2018-12-13 3:10:53