ONLINE RETAIL APPLICATION DATABASE MANAGEMENT SYSTEM

Nidhi Goyal

NORTHEASTERN UNIVERSITY | INFO 6210

1.

Introduction

The term retail as it is used today often refers to both traditional in-store retail and electronic commerce or e-commerce. Before the emergence of e-commerce into the internet, customers had to visit the traditional brick and mortar shops to buy goods, and sellers had to find a place where they could sell their products. Online Retail is quite similar to physical, brick and mortar retail in the nature of selling and buying good, but it happens over computer-mediated networks. Even many large retail stores include both options physical and online purchase.

Background Information

Brief History of E-commerce

E-commerce is the activity of electronically buying or selling of products on online services or over the internet. E-commerce, which is now an integral part of many businesses, works on technologies such as mobile commerce, electronic funds transfer, online transaction processing, supply chain management, online, electronic data interchange (EDI), inventory management systems, and automated data collection systems. New technologies continue to influence the e-commerce which works in the area to boost sale revenue, to attract more customers to buy a wide range of goods and services at anytime and anywhere in the world.

Why E-commerce?

In the today's business world, no seller wants to be left behind, moreover online shopping for retail sales direct to consumers via Web sites and mobile apps, and conversational commerce via live chat, chatbots, and voice assistant. E-commerce businesses also employ third-party business-to-consumer (B2C), consumer-to-consumer (C2C) sales, business-to-business (B2B) buying and selling and electronic data interchange. The following are advantage of accepting online retail application:

1. To allow customers to buy products at their own convenience at anytime and anywhere in the world.
2. To make goods and services available 24/7.
3. To provide better customer relation in low operational cost.
4. Easy to expand business to increase revenue.
5. To reach more consumers without any physically limitations.

Objective

The aim of this project is to develop an Online Retail Management System which is a form of electronic shopping store where the seller can sell their product online and customer can browse and buy the products without any difficulty. It saves the customers time and allow them to have a good experience. Hence in this project online retail application database named Ecommerce was designed.

Scope

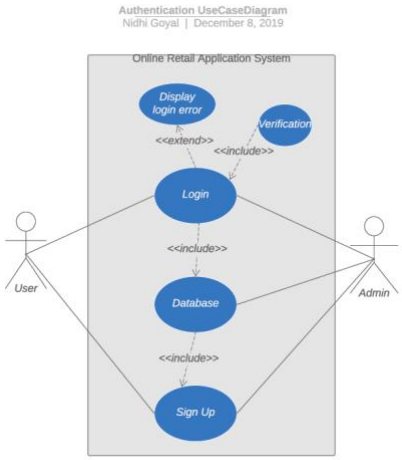
Online retail management refers to the process which helps the customers to obtain the desired resources from the retail stores on daily basis. Retail management features the option to buy goods by

the customers from the retail store through the online mode. Nowadays people do not have time to go to shops and get the desired products. They like to shop in just one stretch by few mouse clicks. So, buying desired items through online is being used a lot these days.

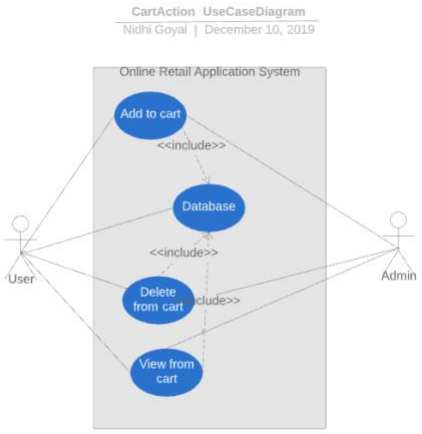
Use Cases and Use Case Diagram

Through this project, the features that can be included in the online retail application management system database are as follows:

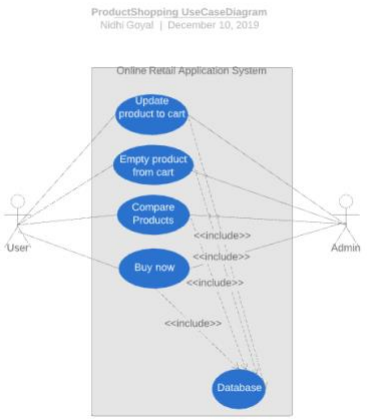
1. Customers have an option to login to the application with the username and password. Only the authenticated customers will be having access to the application. The customer need to login to the application with the unique username and the password.



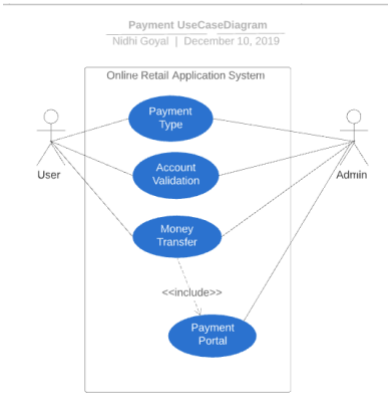
2. Customers have an option to add products to a shopping cart.
3. Customers are able to remove any product from the cart.
4. Customers are able to view products in the shopping cart.



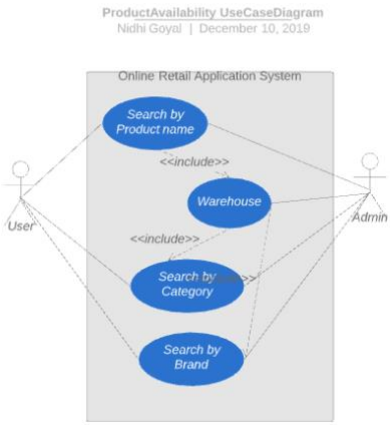
5. Customers are able to update product quantity in the cart.
6. Customer are able to empty all the products in the cart.



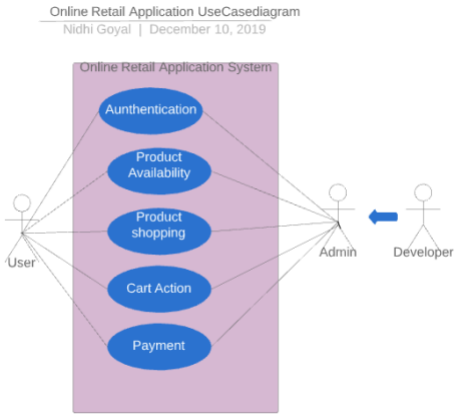
7. Customers have an option to pay through Visa Card, MasterCard and American Express.



8. Customers have an option to search product by different means.

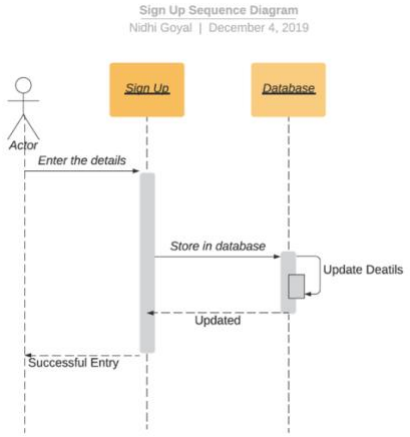


9. The admin is able to manage (add, delete and update) products and their categories.
10. The admin is able to view the lists of products, brand and categories.
11. The admin is able to view payment and order details.
12. The admin is able to view list of categories whose product is available to sell and whose quantity is more than zero.

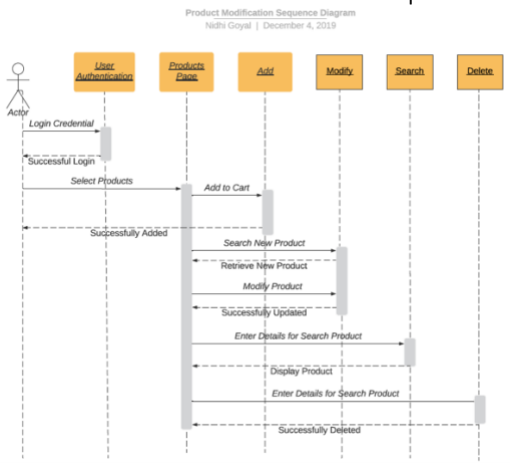


Sequence Diagram

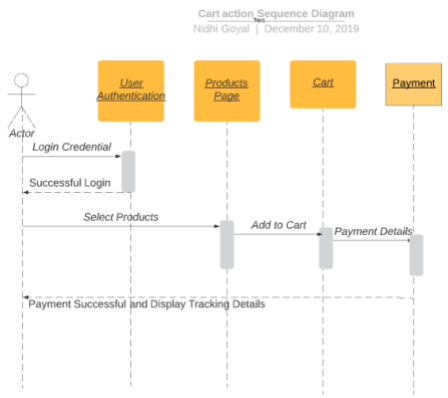
1. Signup Sequence Diagram



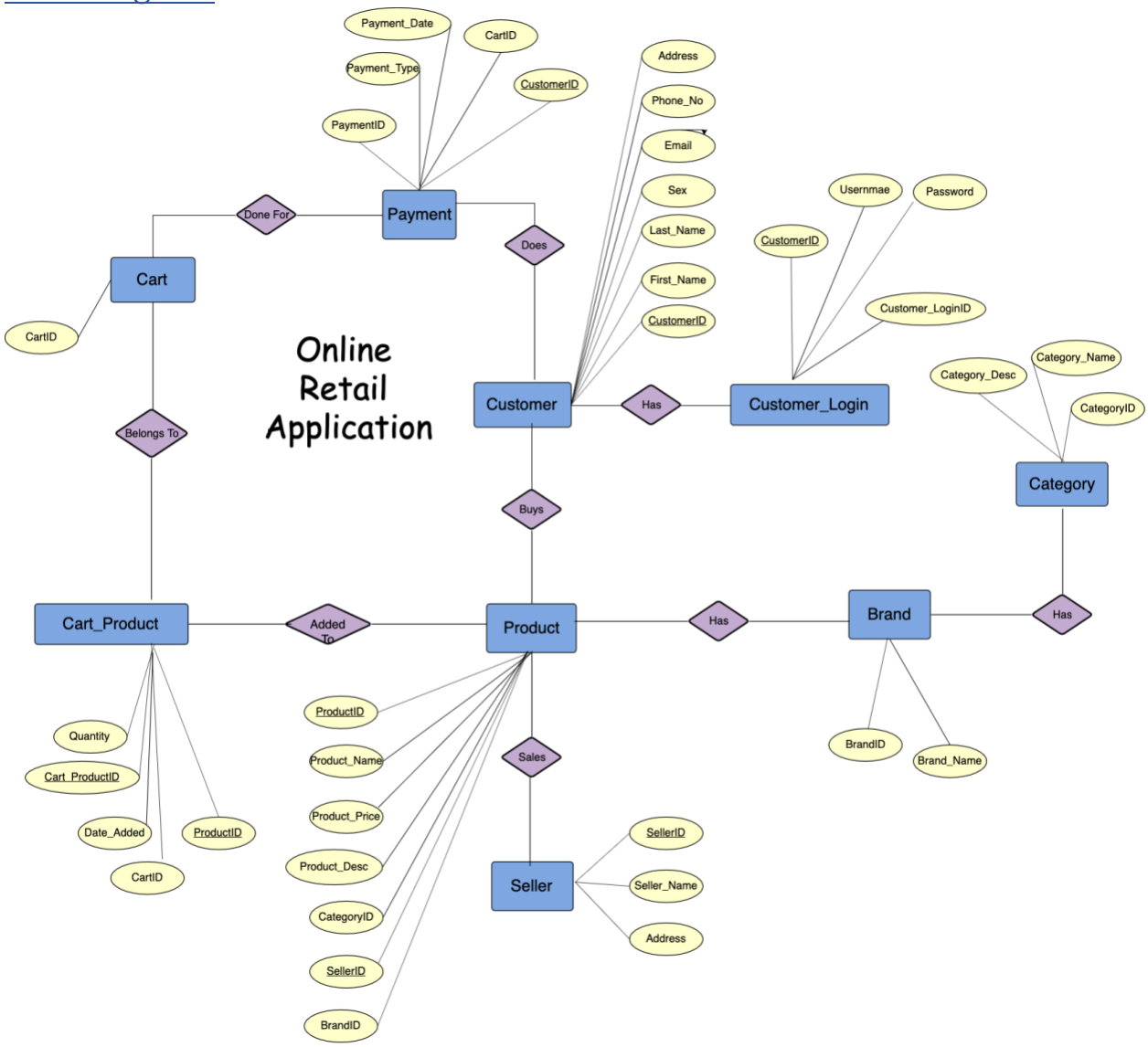
2. Product Modification Sequence Diagram



3. Cart action Sequence Diagram



2.ER Diagram



MySQL database management system is the most popular database system because of its reliability, flexibility, and speed. All the data generated by this application are managed on MySQL database system.

The above figure the Entity Relationship Diagram of the application database. The following list gives a brief explanation of these tables:

1. Customer – Contains customer data.
2. Customer_Login – Contains customer login information.
3. Category – Contains category data for the products.
4. Brand – Contains brand details for the products.
5. Seller – Contains seller information who sales product.
6. Cart – Contains cart data.
7. Product – Contains product data.
8. Payment – This contains order payment data.
9. Cart_Product– This is a bridge table for the products which is added by customer to cart.

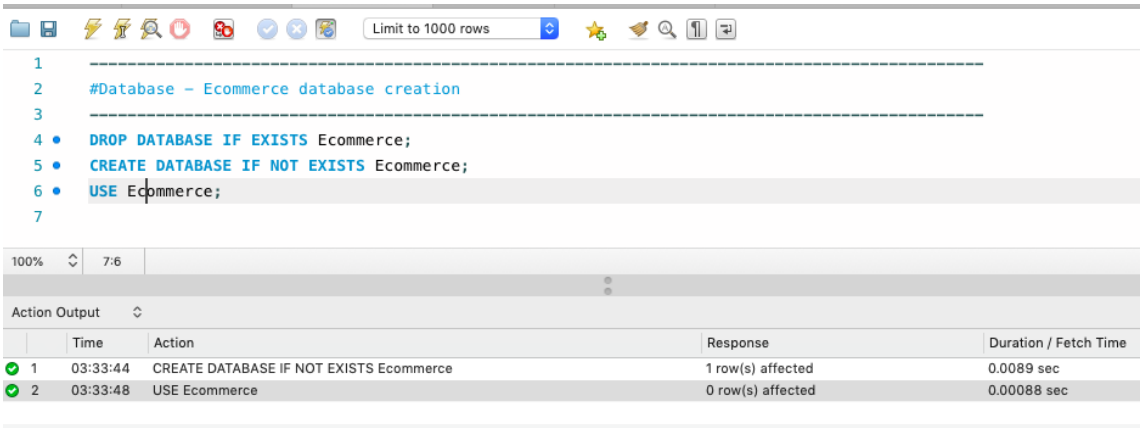
3.Procedure

SET 1: Database Creation

The CREATE DATABASE statement is used to create a database in MySQL. We can also drop table by using the DROP DATABASE keyword. Here we created 'Ecommerce' database for online retail application management system. Once we create database, we need to select that database by using USE keyword for further table creation in that schema.

MySQL syntax for Database Creation:

DROP DATABASE IF EXISTS database_name;
CREATE DATABASE IF NOT EXISTS database_name;
USE database_name;



SET 2: Table Creation

The CREATE TABLE statement is used to create a table in MySQL. We can also drop table by using the DROP TABLE keyword. We use SHOW TABLES statement to list all the tables.

MySQL syntax for Table Creation:

```
DROP TABLE IF EXISTS table_name;
CREATE TABLE IF NOT EXISTS table_name
(
        column_name1 datatype NOT NULL AUTO_INCREMENT,
        column_name2 datatype NOT NULL,
        PRIMARY KEY (column_name1)
);
```

In the Ecommerce database, we created 10 tables which are Customer, Address, Cart, Brand, Category, Product, Seller, Cart_Product, Customer_Login and Payment. In the creation of tables many MySQL Constraints were used.

MySQL Constraint:

The MySQL constraint is used to define what values can be stored in columns. The purpose of using constraints is to enforce the integrity of a database. It can be classified into two types - column level and table level. The column level constraints can apply only to one column whereas table level constraints are applied to the entire table. The constraint is declared at the time of creating a table. We used the following constraints:

1. NOT NULL - It does not allow null value in the column; each row must contain a value for that column. Because by default, a column can hold NULL values. So, when we use NOT NULL constraint, we cannot insert or update value without adding a value to that field.

2. CHECK - It used to limit the value range or define the certain values for a column.

3. UNIQUE – This ensures that all values in a column are different. We can use unique constraint multiple times in table.

4. DEFAULT - It will set a default value that is added when no other value is passed.

5. AUTO_INCREMENT - MySQL automatically increases the value of the field by 1 each time a new record is added.

6. PRIMARY KEY - Each table should have a primary key column and it uniquely identifies each record in a table. The column with PRIMARY KEY must contain UNIQUE values and cannot contain NULL values. A table can have only one primary key column and is often used with AUTO_INCREMENT.

7. FOREIGN KEY - We used this key to link between two tables. A FOREIGN KEY is a field (or collection of fields) in one table that refers to the PRIMARY KEY in another table. The table containing the foreign key is called the child table, and the table containing the candidate key is called the referenced or parent table.

---

1. Table - Address table containing address details

```
DROP TABLE IF EXISTS Address;
CREATE TABLE IF NOT EXISTS Address (
AddressID INT NOT NULL AUTO_INCREMENT,
Street VARCHAR (50) NOT NULL,
Region VARCHAR (50) NOT NULL,
City VARCHAR (50) NOT NULL,
Country VARCHAR (50) NOT NULL DEFAULT 'USA',
Postal_Code VARCHAR (10) NOT NULL,
PRIMARY KEY (AddressID)
);
```

---

2. Table - Cart table containing cart details

```
DROP TABLE IF EXISTS Cart;
CREATE TABLE IF NOT EXISTS Cart (
CartID INT NOT NULL AUTO_INCREMENT,
PRIMARY KEY (CartID)
);
```

---

3. Table - Customer table containing customer details

```
DROP TABLE IF EXISTS Customer;
CREATE TABLE IF NOT EXISTS Customer (
CustomerID INT NOT NULL AUTO_INCREMENT,
First_Name VARCHAR (50) NOT NULL,
Last_Name VARCHAR (50) NOT NULL,
Sex Enum ('Male', 'Female', 'Transgender'),
Email VARCHAR (50),
Phone_No VARCHAR (50),
AddressID INT NOT NULL,
CartID INT NOT NULL,
PRIMARY KEY (CustomerID),
FOREIGN KEY (AddressID) REFERENCES Address (AddressID),
FOREIGN KEY (CartID) REFERENCES Cart (CartID)
);
```

---

4. Table - Customer_Login table containing customer login details

---

```
DROP TABLE IF EXISTS Customer_Login;
CREATE TABLE IF NOT EXISTS Customer_Login (
Customer_LoginID INT NOT NULL AUTO_INCREMENT UNIQUE,
Username VARCHAR (20) NOT NULL,
Password VARCHAR (20) NOT NULL,
CustomerID INT NOT NULL,
PRIMARY KEY (Customer_LoginID),
FOREIGN KEY (CustomerID) REFERENCES Customer (CustomerID)
);
```

---

5. Table - Brand table containing brand details

```
DROP TABLE IF EXISTS Brand;
CREATE TABLE IF NOT EXISTS Brand (
BrandID INT NOT NULL AUTO_INCREMENT,
Brand_Name VARCHAR (50) NOT NULL UNIQUE,
PRIMARY KEY (BrandID)
);
```

---

6. Table - Category table containing category details

```
DROP TABLE IF EXISTS Category;
CREATE TABLE IF NOT EXISTS Category (
CategoryID INT NOT NULL AUTO_INCREMENT,
Category_Name VARCHAR (50) NOT NULL,
Category_DESC TEXT,
PRIMARY KEY (CategoryID)
);
```

---

7. Table - Seller table containing cart details

```
DROP TABLE IF EXISTS Seller;
CREATE TABLE Seller (
SellerID INT NOT NULL,
Seller_Name VARCHAR (50) NOT NULL,
AddressID INT NOT NULL,
PRIMARY KEY (SellerID),
FOREIGN KEY (AddressID) REFERENCES Address (AddressID));
```

---

8. Table - Product table containing product details

```
DROP TABLE IF EXISTS Product;
CREATE TABLE IF NOT EXISTS Product (
ProductID INT NOT NULL AUTO_INCREMENT,
Product_Name VARCHAR (50) NOT NULL,
Product_Price DECIMAL (8,2) NOT NULL,
Product_Description VARCHAR (255) NOT NULL DEFAULT 'No Description',
CategoryID INT NOT NULL,
BrandID INT NOT NULL,
SellerID INT NOT NULL,
PRIMARY KEY (ProductID),
FOREIGN KEY (BrandID) REFERENCES Brand (BrandID),
FOREIGN KEY (CategoryID) REFERENCES Category (CategoryID),
FOREIGN KEY (SellerID) REFERENCES Seller (SellerID)
);
```

---

9. Table - Payment table containing payment details

```
DROP TABLE IF EXISTS Payment;
CREATE TABLE IF NOT EXISTS Payment (
PaymentID INT NOT NULL AUTO_INCREMENT,
Payment_Type VARCHAR (50) NOT NULL,
Payment_Date DATETIME NOT NULL,
CustomerID INT NOT NULL,
CartID INT NOT NULL,
PRIMARY KEY (PaymentID),
FOREIGN KEY (CustomerID) REFERENCES Customer (CustomerID),
FOREIGN KEY (CartID) REFERENCES Cart (CartID));
```

---

10. Table - Cart_Product table containing product cart details

---

```
DROP TABLE IF EXISTS Cart_Product;
CREATE TABLE Cart_Product (
Cart_ProductID INT NOT NULL AUTO_INCREMENT,
Quantity INT NOT NULL CHECK (Quantity > 0),
Date_Added DATETIME NOT NULL,
CartID INT NOT NULL,
ProductID INT NOT NULL,
Primary key (Cart_ProductID),
FOREIGN KEY (CartID) REFERENCES Cart (CartID),
FOREIGN KEY (ProductID) REFERENCES Product (ProductID));
```

```sql
-----------------------------------------------------------------------------------------
#Table - Address table containing address details
-----------------------------------------------------------------------------------------

DROP TABLE IF EXISTS Address;
CREATE TABLE IF NOT EXISTS Address(
AddressID INT NOT NULL AUTO_INCREMENT,
Street VARCHAR(50) NOT NULL,
Region VARCHAR(50) NOT NULL,
City VARCHAR(50) NOT NULL,
Country VARCHAR(50) NOT NULL DEFAULT 'USA',
Postal_Code VARCHAR(10) NOT NULL,
PRIMARY KEY (AddressID)
);
```

100% 16:6

Action Output

	Time	Action	Response	Duration / Fetch Time
1	03:33:44	CREATE DATABASE IF NOT EXISTS Ecommerce	1 row(s) affected	0.0089 sec
2	03:33:48	USE Ecommerce	0 row(s) affected	0.00088 sec
3	03:34:28	CREATE TABLE IF NOT EXISTS Address(AddressID INT NOT NULL AUTO_INCREME...	0 row(s) affected	0.040 sec

Tables_in_ecommerce
Address
Brand
Cart
Cart_Product
Category
Customer
Customer_Login
Payment
Product
Seller

Result Grid

Form Editor

Field Types

Result 1

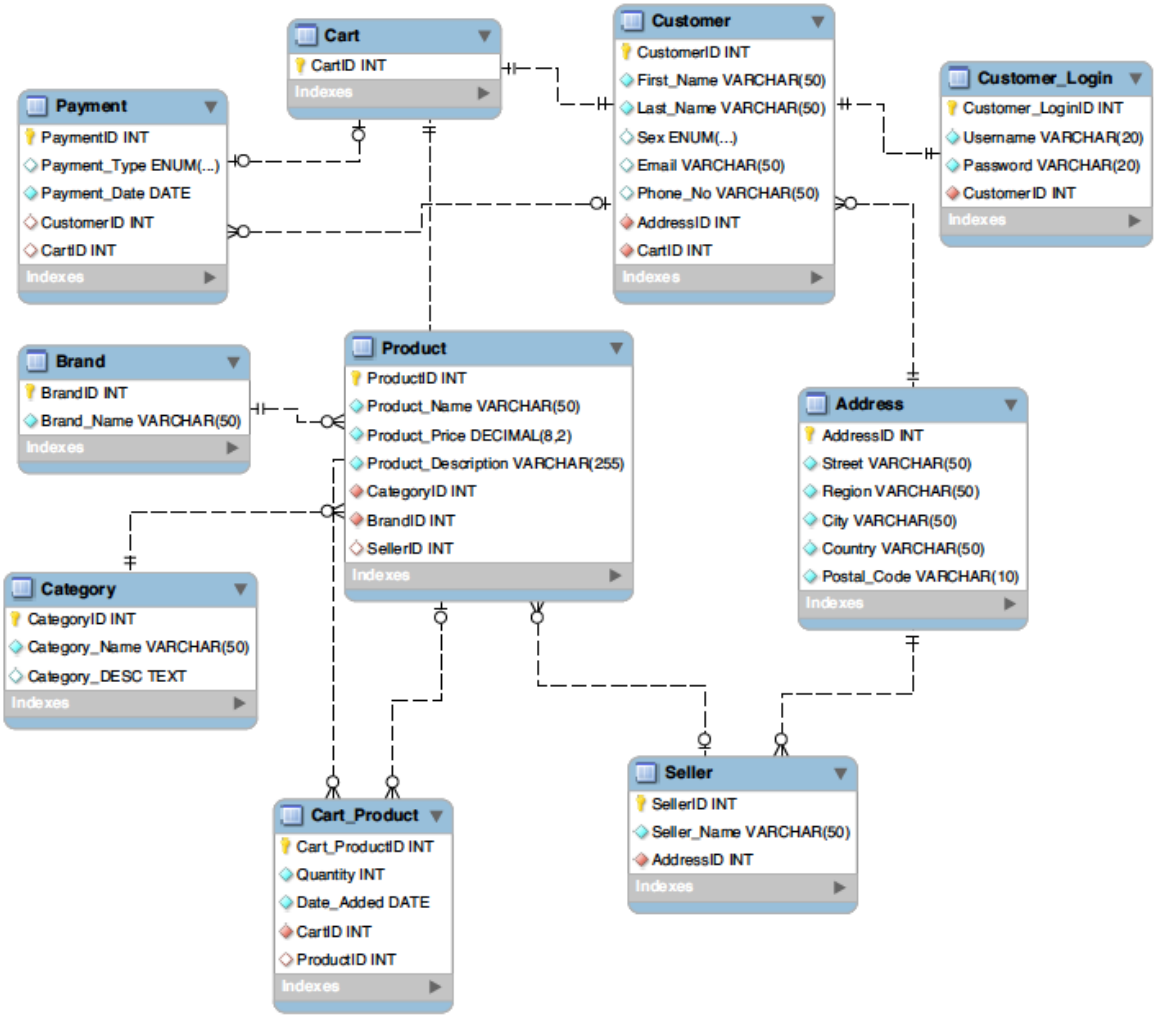
Read Only

Action Output

	Time	Action	Response
1	03:39:42	CREATE DATABASE IF NOT EXISTS Ecommerce	1 row(s) affected
2	03:39:44	USE Ecommerce	0 row(s) affected
3	03:39:48	CREATE TABLE IF NOT EXISTS Address(AddressID INT NOT NULL AUTO_INCREMENT, Street VARCHAR(50) NOT NULL, Region VARCHAR(50)...	0 row(s) affected
4	03:39:53	CREATE TABLE IF NOT EXISTS Cart(CartID INT NOT NULL AUTO_INCREMENT, PRIMARY KEY (CartID))	0 row(s) affected
5	03:39:57	CREATE TABLE IF NOT EXISTS Customer(CustomerID INT NOT NULL AUTO_INCREMENT, First_Name VARCHAR(50) NOT NULL, Last_Name V...	0 row(s) affected
6	03:40:03	CREATE TABLE IF NOT EXISTS Customer_Login(Customer_LoginID INT NOT NULL AUTO_INCREMENT UNIQUE, Username VARCHAR(20) NOT...	0 row(s) affected
7	03:40:07	CREATE TABLE IF NOT EXISTS Brand(BrandID INT NOT NULL AUTO_INCREMENT, Brand_Name VARCHAR(50) NOT NULL UNIQUE, PRIMARY K...	0 row(s) affected
8	03:40:10	CREATE TABLE IF NOT EXISTS Category(CategoryID INT NOT NULL AUTO_INCREMENT, Category_Name VARCHAR(50) NOT NULL, Category_...	0 row(s) affected
9	03:40:14	CREATE TABLE Seller(SellerID INT NOT NULL, Seller_Name VARCHAR(50) NOT NULL, AddressID INT NOT NULL, PRIMARY KEY (SellerID), FO...	0 row(s) affected
10	03:40:18	CREATE TABLE IF NOT EXISTS Product(ProductID INT NOT NULL AUTO_INCREMENT, Product_Name VARCHAR(50) NOT NULL, Product_Price...	0 row(s) affected
11	03:40:22	CREATE TABLE IF NOT EXISTS Payment(PaymentID INT NOT NULL AUTO_INCREMENT, Payment_Type ENUM('Visa', 'MasterCard', 'AmericanEx...	0 row(s) affected
12	03:40:26	CREATE TABLE Cart_Product(Cart_ProductID INT NOT NULL AUTO_INCREMENT, Quantity INT NOT NULL CHECK (Quantity > 0), Date_Added...	0 row(s) affected
13	03:40:38	SHOW TABLES	10 row(s) returned

We can create EER (Enhanced Entity Relationship) diagram through Reverse Engineering in the MySQL workbench. EER represents:

- The identifying or non-identifying relationship between tables
- One to one, one to many or many to many relationships between tables
- Optional or mandatory relationship between tables.



Alter Table

The ALTER TABLE statement is used to add, delete, or modify columns in an existing table. The ALTER TABLE statement is also used to add and drop various constraints on an existing table.

---

1. ALTER TABLE - DROP Column

ALTER TABLE Customer
DROP COLUMN Sex;

The above query drops the column name Sex from the Customer table. We can check the column list of a table by using Describe statement.

DESCRIBE Customer;

```
13 •  DESCRIBE Customer;
```

100% 19:13

Result Grid Filter Rows: Search Export:

Field	Type	Null	Key	Default	Extra
CustomerID	int(11)	NO	PRI	NULL	auto_increment
First_Name	varchar(50)	NO		NULL	
Last_Name	varchar(50)	NO	MUL	NULL	
Email	varchar(50)	NO		NULL	
Phone_No	varchar(50)	YES		NULL	
AddressID	int(11)	NO	MUL	NULL	
CartID	int(11)	NO	MUL	NULL	

---

2. ALTER TABLE - ALTER/MODIFY Column

ALTER TABLE Customer_Login
MODIFY COLUMN Password BLOB;

---

3. SQL NOT NULL on ALTER TABLE

ALTER TABLE Customer
MODIFY Email VARCHAR (50) NOT NULL;

---

4. SQL DEFAULT on ALTER TABLE

ALTER TABLE Customer_Login
ALTER Username SET DEFAULT 'No Username';

---

5. SQL AUTO INCREMENT on ALTER TABLE

ALTER TABLE Customer_Login AUTO_INCREMENT = 100;

Column	Type	Default Value	Nullable	Extra	Character Set	Collation
Customer_LoginID	int(11)		NO	auto_increment		
CustomerID	int(11)		NO			
Password	blob		YES			
Username	varchar(20)	No Username	NO		utf8mb4	utf8mb4_090...

---

6. SQL UNIQUE Constraint on ALTER TABLE

```
ALTER TABLE Category
ADD CONSTRAINT UC_Category UNIQUE (CategoryID, Category_Name);
```

---

7. SQL CHECK on ALTER TABLE

```
ALTER TABLE Cart_Product
ADD CHECK (Quantity > 0);
```

---

8. SQL AUTO INCREMENT on ALTER TABLE

```
ALTER TABLE Payment
MODIFY COLUMN Payment_Type VARCHAR (50);
```

---

9. ALTER TABLE - ALTER/MODIFY Column

```
ALTER TABLE Cart_Product
ADD Purchased VARCHAR (10) DEFAULT 'NO';
```

---

10. ALTER TABLE - UPDATE Column

```
UPDATE Cart_Product
SET Purchased = 'Yes'
WHERE Cart_ProductID IN (2, 5, 6, 9, 54, 23, 46, 89, 77, 61, 3, 83, 66, 55, 44, 33, 22, 11, 7, 82, 28, 42, 43, 49, 35, 37, 39);
```

---

11. SQL CREATE INDEX Statement

```
CREATE INDEX idx_cname
ON Customer (Last_Name, First_Name);
```

A database index is a data structure that can be created using one or more columns. The users cannot see the indexes, they are just used to speed up queries and will be used by the Database Search Engine to locate records very fast. When we insert or update in a database, we also need to insert or update the index values as well.

SET 3: Select – From – Where – Group By – Having - Order By – Limit

SELECT Statement - The SELECT statement is used to select data from a database. The result from the select statement is called result set. When we use asterisk (*) in select statement, it gives all the rows from the table.

WHERE - The WHERE clause is used to filter records based on the specified condition.

GROUP BY - The GROUP BY statement groups rows that have the same values. It often used with aggregate functions (COUNT, MAX, MIN, SUM, AVG) to group the result-set by one or more columns.

HAVING - The HAVING clause is used because we cannot use WHERE with aggregate functions.

ORDER BY – This keyword is used to sort the result-set in ascending or descending order. This sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.

LIMIT - This keyword is used to limit the number of rows returned from a result set.

MySQL Syntax

SELECT column_name(s)
FROM *table_name*
WHERE *condition*
GROUP BY column_name(s)
HAVING *condition*
ORDER BY column_name(s);

Query: To find the total number of products by product name in the product table where category id is 11 and count is greater than 3.

```
SELECT COUNT(ProductID), Product_Name
FROM Product
WHERE CategoryID = 11
GROUP BY Product_Name
HAVING COUNT(ProductID) > 3;
```

100% 1:13

Result Grid Filter Rows: Search Export:

COUNT(ProductID)	Product_Name
5	Chain

Query: To find the total number of 4 products by descending order of product name in the product table where count is greater than 3.

```
SELECT COUNT(ProductID), Product_Name
FROM Product
GROUP BY Product_Name
HAVING COUNT(ProductID) > 3
ORDER BY COUNT(ProductID) DESC
LIMIT 4;
```

100% 1:12

Result Grid Filter Rows: Search Export: Fetch rows:

COUNT(ProductID)	Product_Name
5	Chain
4	File
4	Bag
4	Mobile

SET 4: SQL Functions

1. COUNT () Function: This is an aggregate function which returns the number of records returned by a select query.

```
CREATE DEFINER=`root`@`localhost` FUNCTION `total_products_for_category`(cid INT) RETURNS int(11)
    READS SQL DATA
    DETERMINISTIC
BEGIN
DECLARE Total INT;

 SET Total = (
    SELECT COUNT(ProductID)
    FROM Product
    WHERE CategoryID = cid
    HAVING COUNT(ProductID) > 3);

RETURN Total;
END
```

Call stored function Ecommerce.total_products_for_category

Enter values for parameters of your function and click <Execute> to create an SQL editor and run the call:

cid 11 INT

Cancel Execute

```
select Ecommerce.total_products_for_category(11);
```

100% 1:1

Result Grid Filter Rows: Search Export:

Ecommerce.total_products_for_category(11)
5

2. AVG () Function: This is an aggregate function which returns the average value of an expression.

```
CREATE DEFINER=`root`@`localhost` FUNCTION `avg_price_of_product`(cid INT) RETURNS int(11)
    READS SQL DATA
    DETERMINISTIC
BEGIN
DECLARE Avg INT;

 SET Avg = (
    SELECT AVG(Product_Price)
    FROM Product
    WHERE CategoryID = cid);

RETURN Avg;
END
```

Call stored function Ecommerce.avg_price_of_product

Enter values for parameters of your function and click <Execute> to create an SQL editor and run the call:

cid 11 INT

Cancel Execute

```
select Ecommerce.avg_price_of_product(11);
```

100% 1:3

Result Grid Filter Rows: Search Export:

Ecommerce.avg_price_of_product(11)
1340

3. ADDDATE () FUNCTION - This function adds a time/date interval to a date and then returns the date.

```
SELECT ADDDATE("2019-10-09 07:49:37", INTERVAL 10 DAY);
```

100% 1:21

Result Grid Filter Rows: Search Export:

ADDDATE("2019-10-09 07:49:37", INTERVAL...
2019-10-19 07:49:37

4. Comparison Function -

```
SELECT *
FROM Customer
WHERE CartID <= 50;
```

5. LIKE Function -

```
SELECT *
FROM Customer
WHERE First_Name LIKE '%L';
```

100% 1:23

Result Grid Filter Rows: Search Edit: Export/Import:

CustomerID	First_Name	Last_Name	Email	Phone_No	AddressID	CartID
39	Gail	Keller	Nulla.dignissim@convalliserat.net	1-468-396-2667	43	36
78	Gil	Chan	orci@nuncnullavulputate.edu	663-0411	62	79
79	Nell	Frederick	dignissim@molestie.co.uk	1-123-596-0890	60	61
84	Carl	Wiggins	tellus.justo.sit@aliquetmolestie.edu	1-341-209-5348	75	73
89	Jael	Sherman	pretium.et.rutrum@adipiscinglacusUt.net	1-990-331-4409	10	24
NULL	NULL	NULL	NULL	NULL	NULL	NULL

6. SUM () Function-

```
SELECT SUM(Product_Price) AS Total_Price
FROM Product P
LIMIT 0, 1000;
```

100% 1:21

Result Grid Filter Rows: Search Export:

Total_Price
11052.00

7. MySQL INSTR Function – This Aggregate return the position of the first occurrence of a substring in a string.

```
SELECT Product_Name
FROM Product
WHERE INSTR(Product_Name,'Chain') > 0;
```

100% 32:24

Result Grid Filter Rows: Search Export:

Product_Name
Chain
Chain
Chain
Chain
Chain

8. GROUP_CONCAT Function – This aggregate function concatenate strings from a group into a string with various options such as DISTINCT, ORDER BY, and SEPARATOR.

```
SELECT GROUP_CONCAT(Country)
FROM Address;
```

100% 1:26

Result Grid Filter Rows: Search Export:

GROUP_CONCAT(Country)
Guernsey,Tanzania,Azerbaijan,American Samoa,Korea, South,Falkland Islands,Saudi Arabia,Romania,Christmas Island,Mauritania,L...

9. MySQL EXTRACT Function

```
SELECT EXTRACT(DAY FROM '2020-06-16 13:42:40') DAY;
```

100% 1:29

Result Grid Filter Rows: Search Export:

DAY
16

10. MySQL FLOOR () Function

```
SELECT FLOOR(AVG(Product_Price)) avgPrice
FROM Product
ORDER BY avgPrice;
```

100% 1:32

Result Grid Filter Rows: Search Export:

avgPrice
502

SET 5: JOIN OPERATION

1. INNER JOIN – This join matches each row in one table with every row in other tables and return rows that contain columns from both tables.

SELECT First_Name, Last_Name, Customer.CartID, Address.AddressID, Address.city, Address.Country
FROM Customer
INNER JOIN Address
ON Customer.AddressID = Address.AddressID
WHERE Address.AddressID IN (1, 2, 3);

Result Grid Filter Rows: Search Export:

First_Name	Last_Name	CartID	AddressID	city	Country
Basia	Brennan	1	1	Belfast	Guernsey
Ezra	Herrera	82	1	Belfast	Guernsey
Graham	Fowler	71	2	Bama	Tanzania
Xavier	Walls	9	3	Fuenlabrada	Azerbaijan

2. LEFT JOIN - The LEFT JOIN keyword returns all records from the left table, and the matched records from the right table. The result is NULL from the right side, if there is no match.

```
SELECT Address.AddressID, Customer.First_Name
FROM Address
LEFT JOIN Customer
ON Address.AddressID = Customer.AddressID
ORDER BY Customer.First_Name;
```

45	NULL
66	NULL
64	NULL
61	NULL
22	NULL
46	NULL
58	NULL
57	NULL
48	NULL
55	NULL
51	NULL
81	Adele
26	Aladdin
67	Alden
59	Alisa
84	Alma
76	Althea
49	Alvin
77	Amela

3. RIGHT JOIN - The RIGHT JOIN keyword returns all records from the right table, and the matched records from the left table. The result is NULL from the left side, if there is no match.

```
SELECT Payment.PaymentID, Customer.First_Name, Customer.Last_Name
FROM Payment
RIGHT JOIN Customer
ON Payment.CustomerID = Customer.CustomerID
ORDER BY Payment.CustomerID;
```

NULL	Kelly	Warren
NULL	Alisa	Watkins
NULL	Raya	Webster
NULL	Carl	Wiggins
NULL	Connor	Wilkinson
7	Walter	Huber
10	Keelie	Baker
34	Keelie	Baker
5	Macon	Roman
22	Conan	Branch
20	Lilah	Dodson
9	Libby	Sandoval
47	Libby	Sandoval
29	Winifred	Houston
35	Winifred	Houston
30	Lenore	Norton
38	Chester	Rocha
24	Erich	Watkins

4. FULL JOIN - The FULL OUTER JOIN keyword returns all records when there is a match in left or right table records. It gives very large result-sets. FULL OUTER JOIN also known as FULL JOIN.

```
SELECT First_Name, Last_Name, Payment_Type
FROM Customer
JOIN Payment
ON Customer.CustomerID = Payment.CustomerID;
```

First_Name	Last_Name	Payment_Type
Alvin	Church	VisaCard
Xavier	Walls	VisaCard
Althea	Hurley	MasterCard
Lucian	Kaufman	AmericanExpress
Macon	Roman	VisaCard
Tyler	Yates	AmericanExpress
Walter	Huber	MasterCard
Ezra	Herrera	VisaCard
Libby	Sandoval	MasterCard
Keelie	Baker	VisaCard
Garrison	Cole	AmericanExpress
Garrison	Kaufman	MasterCard
Aladdin	Malone	AmericanExpress
Olympia	Barker	MasterCard
Aladdin	Malone	VisaCard
Olympia	Barker	AmericanExpress
Wayne	Oneill	MasterCard
Xavier	Walls	AmericanExpress
Winifred	Rush	MasterCard
Lilah	Dodson	MasterCard

5. CROSS JOIN - It will produce rows which combine each row from the left table with each row from the right table, so It will return the Cartesian product of rows from tables in the join. We can use CROSS JOIN explicitly or implicitly Within SELECT statement.

```
SELECT *
FROM Customer
CROSS JOIN Address;
```

CustomerID	First_Name	Last_Name	Email	Phone_No	AddressID	CartID	AddressID	Street	Region	City
1	Walter	Huber	magnis@ipsumDonec.co.uk	1-833-261-7822	12	2	1	8385 Lorem Avenue	Ulster	Belfa
2	Adele	Mccormick	tincidunt@mollis.net	904-1519	81	3	1	8385 Lorem Avenue	Ulster	Belfa
3	Suki	Kinney	a.malesuada.id@maurissapien.org	1-903-715-1597	29	5	1	8385 Lorem Avenue	Ulster	Belfa
4	Doris	Hawkins	porttitor.eros@velsapienimperdiet.org	1-408-236-1613	52	3	1	8385 Lorem Avenue	Ulster	Belfa
5	Keelie	Baker	Nulla@mauris.com	575-8226	43	4	1	8385 Lorem Avenue	Ulster	Belfa
6	Macon	Roman	Integer.vulputate.risus@nonnisi.com	1-186-493-8603	68	6	1	8385 Lorem Avenue	Ulster	Belfa
7	Iona	Colon	molestie.tellus@Seddiam.ca	219-2586	41	7	1	8385 Lorem Avenue	Ulster	Belfa
8	Justin	Mckinney	malesuada.malesuada@blanditmattisCras.co.uk	986-4511	28	8	1	8385 Lorem Avenue	Ulster	Belfa
9	Ross	Mclaughlin	convallis.convallis@Aliquamgravida.co.uk	512-0245	81	87	1	8385 Lorem Avenue	Ulster	Belfa
10	Felicia	Hanson	pede@nonummyacfeugiat.edu	1-945-979-8176	50	50	1	8385 Lorem Avenue	Ulster	Belfa
11	Ian	Navarro	turpis.non@ipsumsodalespurus.com	1-308-132-0987	15	15	1	8385 Lorem Avenue	Ulster	Belfa
12	Beck	Gill	elit.sed.consequat@laciniaat.org	617-1122	56	56	1	8385 Lorem Avenue	Ulster	Belfa
13	Marny	Medina	sed.pede.Cum@blanditenim.ca	1-626-939-7832	21	21	1	8385 Lorem Avenue	Ulster	Belfa
14	Connor	Wilkinson	vulputate@sollicitudin.net	112-0588	54	54	1	8385 Lorem Avenue	Ulster	Belfa
15	Conan	Branch	penatibus.et.magnis@quisarcuvel.com	166-6792	65	56	1	8385 Lorem Avenue	Ulster	Belfa
16	Silas	Baird	Duis.at@euismodac.ca	1-475-433-1204	20	2	1	8385 Lorem Avenue	Ulster	Belfa
17	Lilah	Dodson	a.arcu.Sed@egestashendrerit.co.uk	987-8835	26	62	1	8385 Lorem Avenue	Ulster	Belfa
18	Libby	Sandoval	Fusce.aliquet@duiCraspellentesque.net	579-4773	21	12	1	8385 Lorem Avenue	Ulster	Belfa
19	Elmo	Randall	nunc@diamPellentesquehabitant.ca	374-5940	38	83	1	8385 Lorem Avenue	Ulster	Belfa
20	Winifred	Houston	Cras.vulputate.velit@ultrices.edu	424-9180	56	65	1	8385 Lorem Avenue	Ulster	Belfa
21	Lenore	Norton	dictum.Phasellus.in@risus.ca	1-267-925-1945	28	82	1	8385 Lorem Avenue	Ulster	Belfa
22	Hasad	Hess	a.facilisis@est.co.uk	600-5403	91	19	1	8385 Lorem Avenue	Ulster	Belfa
23	Hasad	Beach	imperdiet@acmiclaifend.com	289-8344	53	35	1	8385 Lorem Avenue	Ulster	Belfa

SET 6: Performing Queries and Sub-queries

1. Query for Customer who wants to see details of product present in the cart

```
SELECT *
FROM Product
WHERE ProductID IN (
        SELECT ProductID
        FROM Cart_Product
        WHERE CartID IN (
                SELECT CartID
                FROM Customer
                WHERE CustomerID = 8));
```

ProductID	Product_Name	Product_Price	Product_Description	CategoryID	BrandID	SellerID
22	Bag	250.00	New Only	20	38	34
14	File	4.00	New Only	23	47	3
NULL	NULL	NULL	NULL	NULL	NULL	NULL

2. Query for Customer who wants to see order history

```
SELECT ProductID, Quantity
FROM Cart_Product
WHERE Purchased = 'YES'
AND CartID IN (
        SELECT CartID
        FROM Customer
        WHERE CustomerID = 5);
```

ProductID	Quantity
20	4

3. Query for Customer who wants to modify the cart

```
DELETE FROM Cart_Product
WHERE (ProductID = 32
AND CartID IN (
        SELECT CartID
        FROM Customer
        WHERE CustomerID = 55));
```

4. Query for Customer wants to see filtered products on the basis of Luggage & Travel Accessories

```
SELECT ProductID, Product_Name, Product_Price, Product_Description, SellerID
FROM Product
WHERE CategoryID IN (
        SELECT CategoryID
        FROM Category
        WHERE Category_Name = 'Luggage & Travel Accessories');
```

ProductID	Product_Name	Product_Price	Product_Description	SellerID
20	Bag	200.00	New Only	34
21	Bag	150.00	Used	36
22	Bag	250.00	New Only	34
23	Bag	100.00	New Only	23
NULL	NULL	NULL	NULL	NULL

5. Query for System admin who wants to see what the product are purchased on the particular date

```
SELECT ProductID
FROM Cart_Product
WHERE (Purchased = 'Yes' AND Date_added = '2019-05-15 22:24:39');
```

ProductID
27
14
24
15
15

SET 7: USER and PRIVILEGES

In the Ecommerce database, we assigned four different users which are Admin, Tester, Developer, and User with different privilege settings. This section grants the user's various permission levels in order to ensure data security.

1. MySQL syntax for User Creation

CREATE USER 'user' IDENTIFIED BY 'password';
Where user - the name of the MySQL user which will be created and password – the password which we want to assign to that user

2. MySQL syntax to grant permission for a user

GRANT ALL ON database. * TO 'user'@'localhost';
or
GRANT ALL PRIVILEGES ON database. * TO 'user'@'localhost';

By using this command, we grant all privileges of database to user. These are following MySQL privileges which are most commonly used in Ecommerce database:

- ALL PRIVILEGES – grants all privileges to the MySQL user
- CREATE – allows the user to create databases and tables
- DROP - allows the user to drop databases and tables
- DELETE - allows the user to delete rows from specific MySQL table
- INSERT - allows the user to insert rows into specific MySQL table
- SELECT - allows the user to read the database
- UPDATE - allows the user to update table row
- EXECUTE - allows use of statements that execute stored routines (stored procedures and functions)
- INDEX - allows use of statements that create or drop (remove) indexes
- CREATE ROUTINE - allows use of statements that create stored routines (stored procedures and functions).
- ALTER ROUTINE – allows use of statements that alter or drop stored routines (stored procedures and functions).
- CREATE VIEW – allows use of create view statement
- REFERENCES – allows creation of a foreign key constraint requires the references privilege for the parent table.
- TRIGGER – allows trigger operation (INSERT, DELETE, UPDATE)

3. Other MySQL syntax

SELECT USER FROM mysql.USER; This command gives the list of all created users in the database.

SELECT USER FROM mysql.USER LIMIT 0, 1000;

ALTER USER 'root'@'localhost' IDENTIFIED BY '1234'; By using this command we can change the user information.

FLUSH PRIVILEGES; This command is used to take effect and the privileges to be saved. So, this command should be executed at the end.

DROP USER user_name; This command can drop user from the database.

DROP USER Admin;
DROP USER Tester;
DROP USER Developer;
DROP USER User;

CREATE USER 'Admin' IDENTIFIED BY 'admin';

GRANT ALL ON Ecommerce.* TO 'Admin';

CREATE USER 'Tester' IDENTIFIED BY 'tester';

GRANT Execute ON PROCEDURE Ecommerce.price_filter TO 'Tester';
GRANT Execute ON PROCEDURE Ecommerce.verify_customer_login TO 'Tester';
GRANT SELECT ON Ecommerce.* TO 'Tester';
GRANT INSERT, DELETE ON Ecommerce. * TO 'Tester';
GRANT CREATE TEMPORARY TABLES ON Ecommerce.* TO 'Tester';

CREATE USER 'Developer' IDENTIFIED BY 'developer';

GRANT SELECT, INSERT, DELETE ON Ecommerce. * TO 'Developer';
GRANT CREATE, ALTER, INDEX, DROP, REFERENCES ON Ecommerce. * TO 'Developer';
GRANT CREATE VIEW, CREATE ROUTINE, ALTER ROUTINE, TRIGGER ON Ecommerce. * TO 'Developer';

CREATE USER 'User ' IDENTIFIED BY 'user ';

GRANT SELECT ON Ecommerce. * TO 'User';
GRANT DELETE, UPDATE, INSERT ON Ecommerce.Cart_Product TO 'User';
GRANT DELETE, UPDATE, INSERT ON Ecommerce.Customer_Login TO 'User';
GRANT DELETE, UPDATE, INSERT ON Ecommerce.Payment TO 'User';
GRANT UPDATE, INSERT ON Ecommerce.Customer TO 'User';
GRANT UPDATE, INSERT ON Ecommerce.Address TO 'User';
GRANT Execute ON PROCEDURE Ecommerce.price_filter TO 'User';
GRANT Execute ON PROCEDURE Ecommerce.verify_customer_login TO 'User';

	Time	Action	Response
1	04:53:00	CREATE USER 'Admin' IDENTIFIED BY 'admin'	0 row(s) affected
2	04:53:03	GRANT ALL ON Ecommerce.* TO 'Admin'	0 row(s) affected
3	04:53:05	CREATE USER 'Tester' IDENTIFIED BY 'tester'	0 row(s) affected
4	04:53:13	GRANT Execute ON PROCEDURE Ecommerce.price_filter TO 'Tester'	0 row(s) affected
5	04:53:13	GRANT Execute ON PROCEDURE Ecommerce.verify_customer_login TO 'Tester'	0 row(s) affected
6	04:53:13	GRANT SELECT ON Ecommerce.* TO 'Tester'	0 row(s) affected
7	04:53:13	GRANT INSERT, DELETE ON Ecommerce.* TO 'Tester'	0 row(s) affected
8	04:53:13	GRANT CREATE TEMPORARY TABLES ON Ecommerce.* TO 'Tester'	0 row(s) affected
9	04:53:17	CREATE USER 'Developer' IDENTIFIED BY 'developer'	0 row(s) affected
10	04:53:22	GRANT CREATE VIEW, CREATE ROUTINE, ALTER ROUTINE, TRIGGER ON Ecommerce.* TO 'Developer'	0 row(s) affected
11	04:53:25	CREATE USER 'User ' IDENTIFIED BY 'user '	0 row(s) affected
12	04:53:33	GRANT SELECT ON Ecommerce.* TO 'User'	0 row(s) affected
13	04:53:33	GRANT DELETE, UPDATE, INSERT ON Ecommerce.Cart_Product TO 'User'	0 row(s) affected
14	04:53:33	GRANT DELETE, UPDATE, INSERT ON Ecommerce.Customer_Login TO 'User'	0 row(s) affected
15	04:53:33	GRANT DELETE, UPDATE, INSERT ON Ecommerce.Payment TO 'User'	0 row(s) affected
16	04:53:33	GRANT UPDATE, INSERT ON Ecommerce.Customer TO 'User'	0 row(s) affected
17	04:53:33	GRANT UPDATE, INSERT ON Ecommerce.Address TO 'User'	0 row(s) affected
18	04:53:33	GRANT Execute ON PROCEDURE Ecommerce.price_filter TO 'User'	0 row(s) affected
19	04:53:33	GRANT Execute ON PROCEDURE Ecommerce.verify_customer_login TO 'User'	0 row(s) affected
20	04:54:42	FLUSH PRIVILEGES	0 row(s) affected

SET 8: STORED PROCEDURE

MySQL stored the procedure is an executable database object that contains one or more SQL statements. A procedure has a name, a parameter list, and SQL statements.

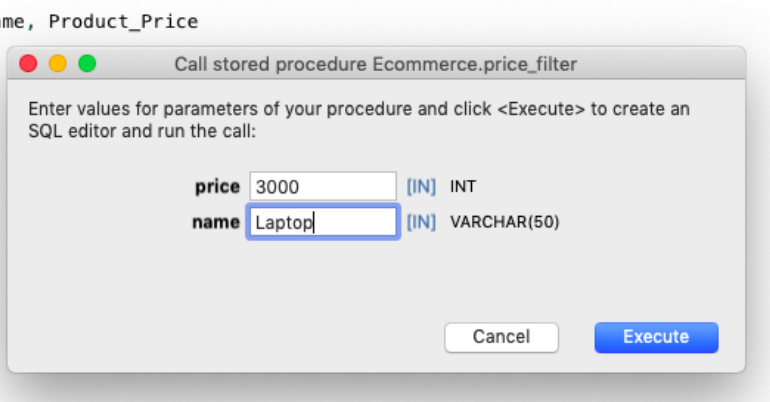
Advantage of the stored procedure-

- Stored procedures are fast. So, if we have a repetitive task that requires checking, looping, multiple statements, and no user interaction, we can do it with a single call to a procedure that's stored on the server.
- Stored procedures are portable, and it will write in SQL. So, we can run on every platform that MySQL runs on, without obliging you to install an additional runtime-environment package or set permissions for program execution in the operating system.
- Stored procedures can restrict and control access to a database. In this way, it can prevent both accidental errors and malicious damage in the database.

Here we created following stored procedure:

1. price filter stored procedure is created to sort data on the basis of Product_Price and Product_Name. When we enter the name and price in this procedure, it will give the value of product whose price is less than the entered price and matches the name.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `price_filter`(IN price INT, IN name VARCHAR(50))
    READS SQL DATA
BEGIN
    SELECT ProductID, Product_Name, Product_Price
    FROM Product
    WHERE Product_Price < PRICE
    AND Product_Name = name;
END
```




```
call Ecommerce.price_filter(3000, 'Laptop');
```

100% 1:3

Result Grid Filter Rows: Search Export:

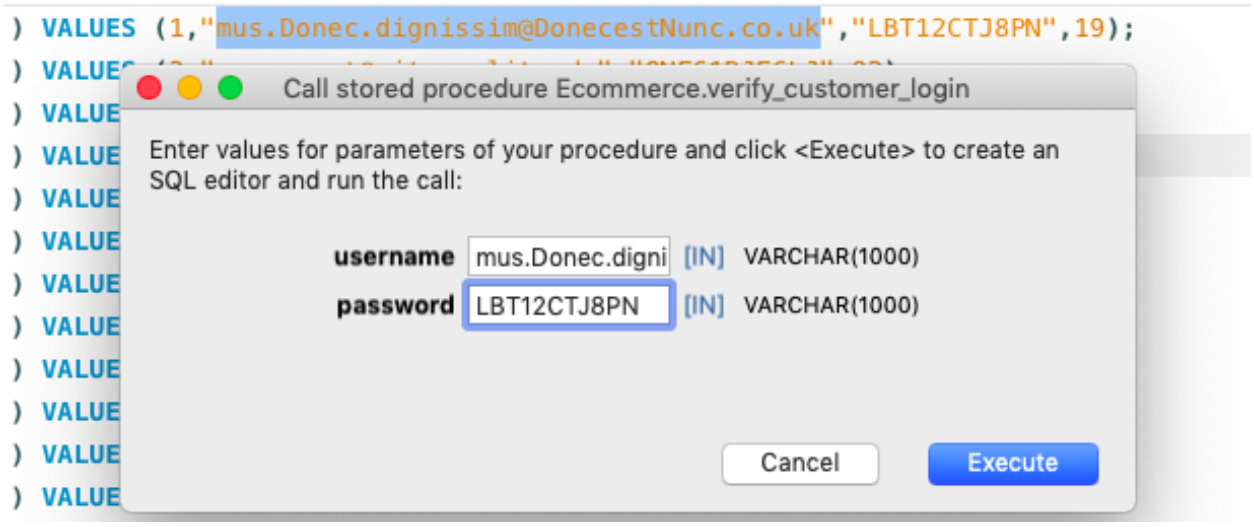
ProductID	Product_Name	Product_Price
34	Laptop	1000.00

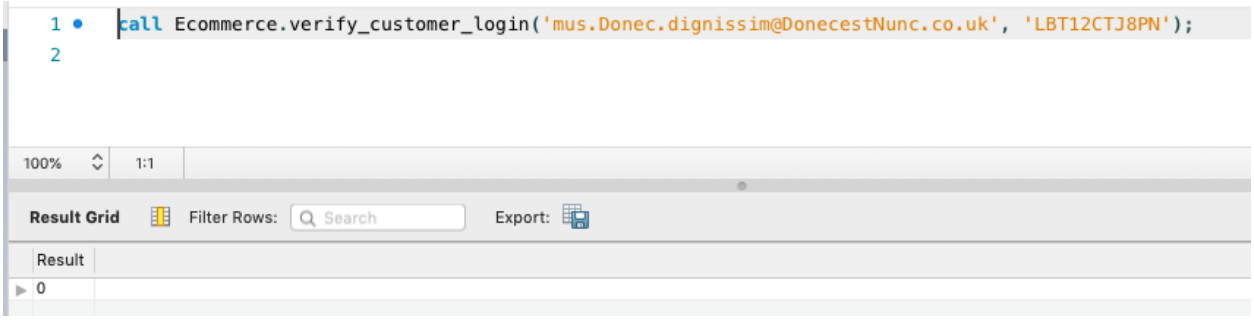
2. verify_customer_login stored procedure helps in authentication of customer login. When customers sign up to the database, their credentials save in the system. Whenever he/she wants to login, this stored procedure will call to check.

```
USE `Ecommerce`;
DROP procedure IF EXISTS `verify_customer_login`;

DELIMITER $$
USE `Ecommerce`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `verify_customer_login`(IN username VARCHAR(1000
    READS SQL DATA
BEGIN
    SELECT IF(count(*) > 0, 0, 1) AS Result
    FROM Customer_Login
    WHERE Username = username
    AND Password = password;
END$$

DELIMITER ;
```





SET 9: MySQL TRANSACTIONS

1. COMMIT STATEMENT –

In order to use a transaction, WE first have to break the SQL statements into logical portions and determine when data should be committed or rolled back.

The following illustrates the step of creating a new customer:

First, start a transaction by using the START TRANSACTION statement.

Next, select the latest customerID from the customer table and use the next customerID number as the new entry.

Then, insert a new customer field values into the customer table.

Finally, commit the transaction using the COMMIT statement.

```
START TRANSACTION;

SELECT *
FROM Customer;

INSERT INTO Customer(First_Name, Last_Name, Email, Phone_No, AddressID, CartID) VALUES
('Nidhi', 'Goyal', 'goyal.ni@husky.neu.edu', '2016260890',23, 24);

COMMIT;
```

100% 3:3

Result Grid Filter Rows: Search Edit: Export/Import:

CustomerID	First_Name	Last_Name	Email	Phone_No	AddressID	CartID
87	Yoshi	Mason	vitae@Namnulla.org	1-128-611-6044	53	91
88	Alvin	Church	magna.nec@PhasellusornareFusce.net	693-4893	49	51
89	Jael	Sherman	pretium.et.rutrum@adipiscinglacusUt.net	1-990-331-4409	10	24
90	Eugenia	Barnett	habitant.morbi@orciUt.com	1-785-467-0895	7	27
91	Giacomo	Gray	torquent.per@duilectusrutrum.org	194-5951	37	97
92	Harrison	Avery	cursus@atiaculisquis.ca	1-397-335-8662	24	34
93	Hedwig	Watson	ad@consequatlectus.net	1-661-691-4228	47	56
94	Jocelyn	Golden	hendrerit@risus.net	289-2667	23	11
95	Hop	Dudley	pede.ultrices.a@convallis.net	711-1244	26	13
96	Tyler	Yates	et@vitaesodales.com	419-1032	21	46
97	Stella	Skinner	magna@cubiliaCuraeDonec.edu	372-2714	39	38
98	Lesley	Burris	aliquet@dolorFuscefeugiat.co.uk	1-960-523-9013	50	59
99	Inez	Walters	amet.risus@lectus.ca	877-8780	84	87
100	Aretha	Boone	aliquet@etrisusQuisque.co.uk	1-104-163-2303	81	77
101	Nidhi	Goyal	goyal.ni@husky.neu.edu	2016260890	23	24
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Customer 2 Apply

Action Output

	Time	Action	Response	Duration
48	15:40:45	START TRANSA...	0 row(s) affected	0.019 se
49	15:40:48	SELECT * FRO...	100 row(s) returned	0.090 se
50	15:40:57	INSERT INTO C...	1 row(s) affected	0.109 se
51	15:41:00	COMMIT	0 row(s) affected	0.012 se

2. ROLLBACK STATEMENT-

```
START TRANSACTION;

DELETE FROM Customer
WHERE CustomerID = 101;

SELECT COUNT(*) FROM Customer;

ROLLBACK;
```

100% 1:16

Result Grid Filter Rows: Search Export:

COUNT(*)
101

Result 4

Action Output

	Time	Action	Response
1	15:49:48	START TRANSACTION	0 row(s) affected
2	15:49:50	DELETE FROM Customer WHERE CustomerID = 101	1 row(s) affected
3	15:49:53	SELECT COUNT(*) FROM Customer LIMIT 0, 1000	1 row(s) returned
4	15:50:00	ROLLBACK	0 row(s) affected
5	15:50:05	SELECT COUNT(*) FROM Customer LIMIT 0, 1000	1 row(s) returned

MySQL Syntax for LOCK Tables

LOCK TABLES table_name [READ | WRITE];

Example: LOCK TABLES Customer_Login READ;

MySQL Syntax for UNLOCK Tables

UNLOCK TABLES;

SET10: MySQL TRIGGERS

The MySQL trigger is a database object that is associated with a table. It will be activated automatically when a specified change operation is performed on a specified table. Triggers are useful for tasks such as enforcing business rules, validating input data, and keeping an audit trail. We created triggers to manage and monitor tables during insert, update or delete.

Advantage of Trigger-

- Triggers are used to check the integrity of data.
- Triggers handle errors from the database layer.
- Triggers give another way to run scheduled tasks. By using triggers, we don't have to wait for the scheduled events to run because the triggers are invoked automatically before or after a change is made to the data in a table.
- Triggers can be useful for auditing the data changes in tables.

Here we created following triggers:

1. Payment_BEFORE_INSERT- This trigger automatically triggers when customers is going to pay. If PaymentID is blank, then it shows 'SQLSTATE 45000'. '45000' is generic SQL error which means unhandled user defined exception.

```
DROP TRIGGER IF EXISTS `Ecommerce`.`Payment_BEFORE_INSERT`;

DELIMITER $$
USE `Ecommerce`$$
CREATE DEFINER = CURRENT_USER TRIGGER `Ecommerce`.`Payment_BEFORE_INSERT`
BEFORE INSERT ON `Payment`
FOR EACH ROW
BEGIN
IF NEW.PaymentID = ' ' THEN
SIGNAL SQLSTATE '45000';
END IF;
END$$
DELIMITER ;
```

2. Customer_BEFORE_INSERT_NULL -It triggers when customers inserts value in customer table and if customers don't put value in Email and Phone_No column then it automatically sets Null value for those columns.

```
DROP TRIGGER IF EXISTS `Ecommerce`.`Customer_BEFORE_INSERT_NULL`;

DELIMITER $$
USE `Ecommerce`$$
CREATE DEFINER = CURRENT_USER TRIGGER `Ecommerce`.`Customer_BEFORE_INSERT_NULL`
BEFORE INSERT ON `Customer`
FOR EACH ROW
BEGIN
IF NEW.Email = '' THEN
SET NEW.Email = NULL;
ELSEIF
NEW.Phone_No = '' THEN
SET NEW.Phone_No = NULL;
END IF;
END$$
DELIMITER ;
```

SET11: MySQL VIEW

The MySQL view is a simple to select statement that gets the inner join result and the view is always going to be up to date whenever we run the view statement. We can also update the table through the view. If we update any value in a column in the view, then it automatically updates the table. So, view not only gives us the latest data, but it also allows us to put data in.

There are some restrictions when we update the view, updateable the view can't include Aggregate functions, GROUP BY clause, HAVING Clause, UNION Clause, DISTINCT keyword, LEFT and RIGHT JOIN and Subqueries. For a view to be updatable, there must be a one-to-one relationship between the rows in the view and the rows in the underlying table. If we change in the structure of the table then it breaks the view.

Here we created following views:

1. The following view gives the list of all customers who used VISA card as Payment_Type when purchasing product on online retail application.

```sql
USE `Ecommerce`;
CREATE  OR REPLACE VIEW `customer_use_VISACard` AS
SELECT C.CustomerID, C.First_Name, C.Last_Name, P.Payment_Type
FROM Payment P
JOIN Customer C
WHERE C.CustomerID = P.CustomerID
AND Payment_Type = 'VisaCard';;
```

Result-

```sql
SELECT * FROM Ecommerce.customer_use_VISACard;
```

100% 1:1

Result Grid Filter Rows: Search Export:

CustomerID	First_Name	Last_Name	Payment_Type
88	Alvin	Church	VisaCard
41	Xavier	Walls	VisaCard
6	Macon	Roman	VisaCard
36	Ezra	Herrera	VisaCard
5	Keelie	Baker	VisaCard
44	Aladdin	Malone	VisaCard
83	Hedley	Dunlap	VisaCard
26	Erich	Watkins	VisaCard
67	Beverly	Schneider	VisaCard
21	Lenore	Norton	VisaCard
33	Victor	Alford	VisaCard
5	Keelie	Baker	VisaCard
86	Todd	Cline	VisaCard
24	Chester	Rocha	VisaCard
78	Gil	Chan	VisaCard
45	Elijah	Callahan	VisaCard
94	Jocelyn	Golden	VisaCard
32	Shea	Leblanc	VisaCard

2. The following view gives the list of all customers who used Master card as Payment_Type when purchasing product on online retail application.

```sql
USE `ecommerce`;
CREATE  OR REPLACE VIEW `customer_use_MasterCard` AS
SELECT C.CustomerID, C.First_Name, C.Last_Name, P.Payment_Type
FROM Payment P
JOIN Customer C
WHERE C.CustomerID = P.CustomerID
AND Payment_Type = 'MasterCard';;
```

Result –

```
SELECT * FROM Ecommerce.customer_use_MasterCard;
```

100% 1:1

Result Grid Filter Rows: Search Export:

CustomerID	First_Name	Last_Name	Payment_Type
52	Althea	Hurley	MasterCard
1	Walter	Huber	MasterCard
18	Libby	Sandoval	MasterCard
34	Garrison	Kaufman	MasterCard
49	Olympia	Barker	MasterCard
54	Wayne	Oneill	MasterCard
48	Winifred	Rush	MasterCard
17	Lilah	Dodson	MasterCard
70	Ursula	Farley	MasterCard
95	Hop	Dudley	MasterCard
87	Yoshi	Mason	MasterCard
51	Robin	Shepard	MasterCard
85	Graham	Fowler	MasterCard
30	Lani	Roman	MasterCard
42	Unity	Levy	MasterCard
87	Yoshi	Mason	MasterCard

3. The following view gives the list of all customers who used AmericanExpress as Payment_Type when purchasing product on online retail application.

```
USE `ecommerce`;
CREATE  OR REPLACE VIEW `customer_use_AmericanExpress` AS
SELECT C.CustomerID, C.First_Name, C.Last_Name, P.Payment_Type
FROM Payment P
JOIN Customer C
WHERE C.CustomerID = P.CustomerID
AND Payment_Type = 'AmericanExpress';;
```

Result –

```
SELECT * FROM Ecommerce.customer_use_AmericanExpress;
```

100% 1:1

Result Grid Filter Rows: Search Export:

CustomerID	First_Name	Last_Name	Payment_Type
69	Lucian	Kaufman	AmericanExpress
96	Tyler	Yates	AmericanExpress
65	Garrison	Cole	AmericanExpress
44	Aladdin	Malone	AmericanExpress
49	Olympia	Barker	AmericanExpress
41	Xavier	Walls	AmericanExpress
15	Conan	Branch	AmericanExpress
93	Hedwig	Watson	AmericanExpress
20	Winifred	Houston	AmericanExpress
29	Alma	Melton	AmericanExpress
20	Winifred	Houston	AmericanExpress
87	Yoshi	Mason	AmericanExpress
34	Garrison	Kaufman	AmericanExpress
90	Eugenia	Barnett	AmericanExpress
45	Elijah	Callahan	AmericanExpress
18	Libby	Sandoval	AmericanExpress

4.SQL CODING QUESTION

1. Write a query to find how much product sold on the particular date

```
SELECT COUNT(ProductID), Date_added
FROM Cart_Product
WHERE Purchased = 'Yes' AND Date_added = '2019-05-15 22:24:39'
GROUP BY Date_added;
```

COUNT(ProductID)	Date_added
5	2019-05-15 22:24:39

2. Write a query for Customer who want to know the total price present in the cart

```
SELECT SUM (Quantity * Product_Price) AS Total_Price
FROM Product P
JOIN Cart_Product C
ON P. ProductID = C. Cart_ProductID
WHERE C. ProductID IN (
        SELECT ProductID
        FROM Cart_Product
        WHERE Purchased = 'Yes'
        AND CartID IN (
                SELECT CartID
                FROM Customer
                WHERE CustomerID = 8));
```

Total_Price
12.00

3. Write a query to show the details of the customer who has not purchased any thing

```
SELECT * FROM Customer
WHERE CustomerID NOT IN (
        SELECT CustomerID
        FROM Payment);
```

CustomerID	First_Name	Last_Name	Email	Phone_No	AddressID	CartID
4	Doris	Hawkins	porttitor.eros@velsapienimperdiet.org	1-408-236-1613	52	3
7	Iona	Colon	molestie.tellus@Seddiam.ca	219-2586	41	7
8	Justin	Mckinney	malesuada.malesuada@blanditmattisCras.co.uk	986-4511	28	8
9	Ross	Mclaughlin	convallis.convallis@Aliquamgravida.co.uk	512-0245	81	87
10	Felicia	Hanson	pede@nonummyacfeugiat.edu	1-945-979-8176	50	50

4. Write a query to Count total active customer i.e. who has purchased any thing

```
SELECT COUNT(CustomerID) AS Active
FROM Customer
WHERE CustomerID IN (
        SELECT CustomerID
        FROM Payment);
```

Active
40

5. Write a query to find total sales of the online retail application

```
SELECT SUM (Quantity * Product_Price) AS Total_Profit
FROM Product P
INNER JOIN Cart_Product C
ON P. ProductID = C. ProductID
WHERE Purchased = 'Yes';
```

Total_Profit
19960.00

References

https://en.wikipedia.org/wiki/E-commerce
https://kyup.com/tutorials/create-new-user-grant-permissions-mysql/
https://dev.mysql.com/doc/
https://www.tutorialspoint.com/sql/index.htm
https://www.w3schools.com/sql/default.asp

MySQL CODE

```
-- MySQL dump 10.13  Distrib 8.0.17, for macos10.14 (x86_64)
--
-- Host: 127.0.0.1    Database: Ecommerce
-- ------------------------------------------------------
-- Server version          8.0.17

/*!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 */;
/*!50503 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 */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `Address` (
  `AddressID` int(11) NOT NULL AUTO_INCREMENT,
  `Street` varchar(50) NOT NULL,
  `Region` varchar(50) NOT NULL,
  `City` varchar(50) NOT NULL,
  `Country` varchar(50) NOT NULL DEFAULT 'USA',
  `Postal_Code` varchar(10) NOT NULL,
  PRIMARY KEY (`AddressID`)
) ENGINE=InnoDB AUTO_INCREMENT=101 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
/*!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,'8385 Lorem
Avenue','Ulster','Belfast','Guernsey','69473'),(2,'1177 Molestie Street','BO','Bama','Tanzania','75-
817'),(3,'913-461 Tristique Avenue','MA','Fuenlabrada','Azerbaijan','5938'),(4,'P.O. Box 508, 9002
Vel, Rd.','BE','Berlin','American Samoa','8108'),(5,'P.O. Box 507, 4954 In Road','M','Cork','Korea,
South','59-700'),(6,'Ap #422-5278 A Rd.','L','Dublin','Falkland Islands','28798'),(7,'928-2152 Et,
Ave','KP','Bydgoszcz','Saudi Arabia','582310'),(8,'213-5304 Scelerisque
Street','OK','Balfour','Romania','8382'),(9,'Ap #939-7794 Integer
Ave','Arequipa','Arequipa','Christmas Island','987363'),(10,'340-9459 Enim
Road','U','Belfast','Mauritania','98076'),(11,'P.O. Box 296, 1158 Lorem, Ave','Utah','Salt Lake
City','Lithuania','104200'),(12,'Ap #744-1669 Diam Avenue','Jönköpings
län','Värnamo','Moldova','786865'),(13,'272-3807 Adipiscing.
Rd.','HH','Hamburg','Lebanon','53147'),(14,'Ap #769-7592 Egestas
St.','Aragón','Huesca','Paraguay','245550'),(15,'Ap #968-7008 Sed,
St.','UT','Dehradun','Oman','T9C 7G5'),(16,'Ap #173-2655 Adipiscing Rd.','North
Island','Palmerston North','Latvia','65166'),(17,'Ap #585-5636 Nec,
St.','Sląskie','Gliwice','Uruguay','3781'),(18,'9727 Tincidunt Road','East Java','Probolinggo','Czech
Republic','0167 NK'),(19,'P.O. Box 235, 4375 Arcu. Avenue','Provence-Alpes-Côte
d\'Azur','Avignon','Kiribati','8345'),(20,'P.O. Box 620, 8061 Lacus. Rd.','Noord
```

Holland','Amsterdam','Mongolia','65188'),(21,'1306 Dui. Rd.','Gyeonggi','Guri','French Guiana','9070'),(22,'Ap #794-2401 In Rd.','NI','Gisborne','Senegal','05-280'),(23,'P.O. Box 489, 3231 Aliquam Street','Ktn','Villach','Saudi Arabia','6935'),(24,'P.O. Box 155, 3246 Imperdiet, Street','KP','Bydgoszcz','Saint Kitts and Nevis','81-372'),(25,'8588 Nascetur Road','Van','Bostaniçi','Iran','348928'),(26,'536-838 Eu Ave','NE','Lincoln','Libya','58-498'),(27,'P.O. Box 854, 2067 Nec Street','Western Australia','Armadale','Uganda','377325'),(28,'P.O. Box 858, 5538 Gravida St.','GB','Diamer','Togo','746583'),(29,'Ap #245-6784 Tincidunt. St.','HI','Kailua','Saint Lucia','98139'),(30,'1968 Nunc. St.','Zuid Holland','Spijkenisse','Cocos (Keeling) Islands','5267'),(31,'818-9396 Et Rd.','Antioquia','Medellín','Kuwait','88748'),(32,'P.O. Box 478, 6748 Vitae Rd.','Missouri','Jefferson City','Mongolia','44448'),(33,'P.O. Box 680, 6539 Nullam St.','Ceuta','Ceuta','Reunion','1148'),(34,'2476 Tellus. Rd.','Gl','Winterswijk','Bahamas','15485-218'),(35,'Ap #200-360 Sodales Avenue','SK','Milestone','Mali','16290'),(36,'Ap #641-1035 Eu, Road','OV','Tielrode','Pakistan','552868'),(37,'P.O. Box 618, 1611 Ac Av.','Wie','Vienna','Cambodia','Z6122'),(38,'P.O. Box 968, 5565 Tempus Ave','JH','Mango','Falkland Islands','HZ68 8FD'),(39,'P.O. Box 769, 4430 Dapibus St.','BR','Saint-Brieuc','Guam','800643'),(40,'637-304 Et, Avenue','ANT','Itagüí','Bahrain','412779'),(41,'Ap #493-5032 Congue St.','Gyeonggi','Incheon','Gambia','253424'),(42,'P.O. Box 524, 2231 Orci Av.','Adana','Adana','Papua New Guinea','6850 CH'),(43,'732-1136 Nec Road','New South Wales','Bathurst','Italy','9371 JW'),(44,'P.O. Box 715, 4839 Vivamus Rd.','Swiętokrzyskie','Ostrowiec Świętokrzyski','New Zealand','199176'),(45,'985-3581 Dictum Av.','Provence-Alpes-Côte d\'Azur','Martigues','Sri Lanka','88773-924'),(46,'9472 Ipsum Street','Antioquia','Rionegro','Azerbaijan','31113'),(47,'Ap #372-2860 Malesuada Ave','CV','Torrevieja','Solomon Islands','91815'),(48,'Ap #380-2541 Sem Road','UP','Kanpur Cantonment','Algeria','17283'),(49,'P.O. Box 296, 3717 Blandit St.','Istanbul','Istanbul','Peru','17116'),(50,'703-7811 Sed St.','Comunitat Valenciana','Valéncia','Guam','2257'),(51,'450-8477 Placerat. Street','South Island','Gore','Libya','1933'),(52,'586-8501 Urna Street','Niedersachsen','Braunschweig','France','75126'),(53,'963-6924 Id Av.','C','Galway','Belarus','629971'),(54,'136-7364 At, Street','V','Villa Alemana','Yemen','56343'),(55,'Ap #964-5331 Integer St.','SAR','Armungia','Virgin Islands, British','J9C 7G4'),(56,'418-8265 Fringilla St.','Gye','Uijeongbu','South Georgia and The South Sandwich Islands','66-288'),(57,'1636 Dui. St.','Rio Grande do Sul','Santa Maria','Estonia','859398'),(58,'Ap #845-1374 Sem, Rd.','Gye','Sacheon','Uganda','5669'),(59,'4154 Dolor Road','NW','Leverkusen','Korea, North','6182 TE'),(60,'P.O. Box 994, 5145 Non, Av.','NO','Lille','Bouvet Island','40101'),(61,'Ap #225-5062 Ornare Road','KIR','Kirov','Liechtenstein','07909-570'),(62,'P.O. Box 628, 3364 Aliquet Av.','CAM','Ceppaloni','Switzerland','98619-364'),(63,'310-8793 Integer St.','XV','General Lagos','Maldives','24881'),(64,'2461 Primis Street','Jambi','Jambi','Iceland','25108'),(65,'664-1745 Risus Av.','Berlin','Berlin','Comoros','3628'),(66,'P.O. Box 717, 7293 At, St.','Kirkcudbrightshire','Castle Douglas','Kiribati','485112'),(67,'495-8045 Dolor St.','BC','Nanaimo','United Arab Emirates','45436-269'),(68,'Ap #107-4526 Euismod St.','Anambra','Awka','Uganda','64568'),(69,'Ap #355-343 Quisque St.','South Chungcheong','Dangjin','Moldova','8328'),(70,'Ap #184-8304 Nisi

Rd.','Ank','Kızılcahamam','Uzbekistan','Z9936'),(71,'P.O. Box 982, 813 Mattis. Av.','Provence-Alpes-Côte d\'Azur','Marseille','Guinea-Bissau','141696'),(72,'774-4040 Nunc Street','Guanajuato','Salamanca','Gibraltar','G1W 9M0'),(73,'P.O. Box 100, 1082 A Rd.','Sindh','Sialkot','Dominican Republic','79294'),(74,'P.O. Box 690, 8428 Odio. Av.','Tver Oblast','Tver','Brunei','300338'),(75,'2237 Nunc Ave','SC','Chapecó','Samoa','134113'),(76,'P.O. Box 815, 2593 Quisque St.','Kerala','Thalassery','Korea, South','978888'),(77,'4054 Consectetuer, Road','AP','Nandyal','Bermuda','68123'),(78,'3922 Leo, Av.','ANT','Envigado','Virgin Islands, British','27065-690'),(79,'Ap #666-9714 Mi St.','AN','Vosselaar','Burkina Faso','P6 5YE'),(80,'637-9424 Arcu. Avenue','Khyber Pakhtoonkhwa','Kohat','Vanuatu','703466'),(81,'503-4203 Ut Street','IL','Vitry-sur-Seine','Timor-Leste','4720'),(82,'6793 Neque. Rd.','Samsun','Vezirköprü','Chad','42268'),(83,'P.O. Box 905, 3101 Proin Av.','ATL','Sabanalarga','Iran','50489'),(84,'632-6284 Est Street','RM','Quilicura','Timor-Leste','53377'),(85,'9052 Lacus. Av.','Quebec','Lachine','Tanzania','07914'),(86,'897-5831 Ante. Rd.','Wie','Vienna','Niue','6073 MT'),(87,'Ap #335-3419 Id, St.','C','Galway','Kiribati','922989'),(88,'Ap #614-1551 In St.','SI','Queenstown','Ukraine','R2V 4Z6'),(89,'476-6740 Non, St.','Bali','Denpasar','Dominican Republic','06758-933'),(90,'118-9379 Orci Ave','Baja California','La Paz','Taiwan','770567'),(91,'P.O. Box 323, 4177 Semper Rd.','Washington','Spokane','Guyana','14977'),(92,'3482 Ac Avenue','RM','El Monte','Bahrain','65860'),(93,'Ap #177-4590 Dolor. Road','Kerala','Trivandrum','Malawi','9940 AW'),(94,'Ap #190-124 Tincidunt Rd.','ON','Osgoode','Bonaire, Sint Eustatius and Saba','SU7P 2HN'),(95,'748-9651 Mauris Street','Atlántico','Malambo','United States','33458'),(96,'Ap #365-4256 Auctor St.','Veracruz','Coatzacoalcos','Namibia','5375 OG'),(97,'7582 In Rd.','Balochistan','Sherani','Lesotho','686311'),(98,'5914 Magna. Ave','Noord Brabant','Ravenstein','Indonesia','148239'),(99,'7909 Montes, Av.','SO','Sokoto','Albania','59826-003'),(100,'P.O. Box 414, 2308 Vel Avenue','West Java','Bandung','Cayman Islands','68978');
/*!40000 ALTER TABLE `Address` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `Brand`
--

DROP TABLE IF EXISTS `Brand`;
/*!40101 SET @saved_cs_client   = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `Brand` (
  `BrandID` int(11) NOT NULL AUTO_INCREMENT,
  `Brand_Name` varchar(50) NOT NULL,
  PRIMARY KEY (`BrandID`),
  UNIQUE KEY `Brand_Name` (`Brand_Name`)
) ENGINE=InnoDB AUTO_INCREMENT=51 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

```
--
-- Dumping data for table `Brand`
--

LOCK TABLES `Brand` WRITE;
/*!40000 ALTER TABLE `Brand` DISABLE KEYS */;
INSERT INTO `Brand` VALUES (47,'3M™ office supplies'),(48,'AbilityOne® office
supplies'),(10,'Addidas'),(39,'American Tourister'),(40,'Amway'),(24,'Apple'),(49,'AT-A-GLANCE®
office supplies'),(1,'Bang & Olufsen'),(50,'Boardwalk® office supplies'),(27,'Bose'),(7,'Calvin
Klein'),(30,'Canon'),(42,'Cartier'),(22,'Cello'),(2,'Clarion'),(36,'Columbia'),(35,'Contigo'),(28,'David
Backham'),(44,'David Yurman'),(43,'Harry Winston'),(33,'Holsum
Bread'),(26,'HP'),(32,'Iphone'),(4,'JBL'),(34,'Kellogs'),(18,'Lancome'),(25,'LG'),(17,'Mac'),(16,'Mayb
eline'),(8,'Michael Kors'),(3,'MTX Audio'),(13,'Natures
Valley'),(14,'Nestle'),(12,'Nike'),(29,'Nikon'),(9,'Nush'),(5,'Panasonic'),(11,'Puma'),(46,'Renolds'),(3
7,'Samsonite'),(31,'Samsung'),(20,'Skechers'),(23,'sony'),(21,'Staples'),(15,'Starbucks'),(19,'Tatcha
'),(41,'Tiffany & Co'),(45,'Van Cleef & Arpels'),(6,'Victoria Secreat'),(38,'VIP');
/*!40000 ALTER TABLE `Brand` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `Cart`
--

DROP TABLE IF EXISTS `Cart`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `Cart` (
  `CartID` int(11) NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`CartID`)
) ENGINE=InnoDB AUTO_INCREMENT=101 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
/*!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),(2),(3),(4),(5),(6),(7),(8),(9),(10),(11),(12),(13),(14),(15),(16),(17),(18),(19),(20),(21),(22),(23),(2
4),(25),(26),(27),(28),(29),(30),(31),(32),(33),(34),(35),(36),(37),(38),(39),(40),(41),(42),(43),(44),(4
5),(46),(47),(48),(49),(50),(51),(52),(53),(54),(55),(56),(57),(58),(59),(60),(61),(62),(63),(64),(65),(6
```

```
6),(67),(68),(69),(70),(71),(72),(73),(74),(75),(76),(77),(78),(79),(80),(81),(82),(83),(84),(85),(86),(8
7),(88),(89),(90),(91),(92),(93),(94),(95),(96),(97),(98),(99),(100);
/*!40000 ALTER TABLE `Cart` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `Cart_Product`
--

DROP TABLE IF EXISTS `Cart_Product`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `Cart_Product` (
  `Cart_ProductID` int(11) NOT NULL AUTO_INCREMENT,
  `Quantity` int(11) NOT NULL,
  `Date_Added` datetime NOT NULL,
  `CartID` int(11) NOT NULL,
  `ProductID` int(11) NOT NULL,
  `Purchased` varchar(10) DEFAULT 'NO',
  PRIMARY KEY (`Cart_ProductID`),
  KEY `CartID` (`CartID`),
  KEY `ProductID` (`ProductID`),
  CONSTRAINT `cart_product_ibfk_1` FOREIGN KEY (`CartID`) REFERENCES `cart` (`CartID`),
  CONSTRAINT `cart_product_ibfk_2` FOREIGN KEY (`ProductID`) REFERENCES `product`
(`ProductID`),
  CONSTRAINT `cart_product_chk_1` CHECK ((`Quantity` > 0)),
  CONSTRAINT `cart_product_chk_2` CHECK ((`Quantity` > 0))
) ENGINE=InnoDB AUTO_INCREMENT=101 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `Cart_Product`
--

LOCK TABLES `Cart_Product` WRITE;
/*!40000 ALTER TABLE `Cart_Product` DISABLE KEYS */;
INSERT INTO `Cart_Product` VALUES (1,2,'2020-08-06 05:31:37',33,33,'NO'),(3,2,'2019-11-14
04:16:40',77,21,'Yes'),(5,3,'2019-05-15 22:24:39',15,27,'Yes'),(6,4,'2019-05-15
22:24:39',26,14,'Yes'),(7,4,'2019-05-15 22:24:39',25,24,'Yes'),(9,2,'2020-01-18
22:00:40',42,16,'Yes'),(11,4,'2019-05-15 22:24:39',54,15,'Yes'),(14,1,'2019-05-15
22:24:39',34,22,'NO'),(16,2,'2019-05-15 22:24:39',13,25,'NO'),(17,1,'2020-08-11
02:30:57',74,18,'NO'),(18,1,'2019-05-15 22:24:39',55,27,'NO'),(21,4,'2019-05-15
22:24:39',64,15,'NO'),(22,4,'2020-10-18 11:05:27',32,26,'Yes'),(23,4,'2019-05-15
```

```
22:24:39',38,15,'Yes'),(24,2,'2019-11-13 16:49:10',87,33,'NO'),(25,4,'2019-05-15
22:24:39',45,31,'NO'),(26,4,'2020-11-19 19:21:44',34,32,'NO'),(27,1,'2019-10-06
04:23:47',99,22,'NO'),(28,1,'2019-04-25 02:56:52',62,29,'Yes'),(29,1,'2020-03-23
21:25:32',1,23,'NO'),(30,3,'2019-01-12 03:00:49',35,17,'NO'),(31,3,'2019-12-15
22:41:54',77,32,'NO'),(32,3,'2019-07-05 04:06:14',33,14,'NO'),(33,3,'2020-10-15
01:21:10',73,23,'Yes'),(34,4,'2020-07-12 02:47:34',46,16,'NO'),(37,1,'2020-04-16
19:09:51',1,18,'Yes'),(38,4,'2019-11-26 06:02:41',98,16,'NO'),(39,4,'2019-11-07
06:00:20',91,19,'Yes'),(42,4,'2020-10-06 07:32:16',97,28,'Yes'),(46,1,'2020-07-07
04:00:32',40,32,'Yes'),(47,4,'2020-08-18 05:03:15',12,30,'NO'),(49,4,'2018-12-19
13:30:06',4,20,'Yes'),(52,2,'2020-05-29 19:00:54',3,17,'NO'),(54,2,'2020-04-24
02:02:51',18,21,'Yes'),(57,4,'2020-10-18 06:58:10',67,25,'NO'),(59,2,'2020-05-06
03:33:37',71,33,'NO'),(62,2,'2020-03-22 19:54:01',25,33,'NO'),(63,1,'2020-03-14
14:46:50',78,19,'NO'),(64,1,'2019-07-05 05:02:43',9,17,'NO'),(67,2,'2020-08-22
17:43:16',98,18,'NO'),(68,1,'2020-07-26 12:40:18',100,16,'NO'),(70,2,'2019-02-27
21:39:31',8,22,'NO'),(71,3,'2019-04-25 06:09:43',51,18,'NO'),(72,4,'2020-04-06
04:57:20',53,14,'NO'),(74,3,'2020-02-23 03:06:31',57,26,'NO'),(75,3,'2020-11-19
06:33:39',43,29,'NO'),(76,1,'2019-10-08 19:03:16',76,27,'NO'),(77,4,'2020-10-22
11:00:38',78,15,'Yes'),(78,2,'2019-07-14 12:49:56',49,34,'NO'),(79,4,'2019-11-20
04:58:52',93,21,'NO'),(81,4,'2020-01-27 01:23:46',14,25,'NO'),(82,3,'2020-10-02
00:39:54',49,14,'Yes'),(83,2,'2020-09-18 03:18:51',8,14,'Yes'),(84,4,'2019-06-14
02:39:59',93,33,'NO'),(85,4,'2019-10-15 20:54:33',86,33,'NO'),(88,2,'2020-10-27
18:41:30',39,28,'NO'),(91,4,'2019-08-08 15:12:09',85,27,'NO'),(94,3,'2020-11-08
06:18:29',40,34,'NO'),(96,4,'2019-04-24 02:51:03',51,33,'NO'),(97,3,'2020-11-29
23:52:01',93,18,'NO'),(98,1,'2019-01-30 15:47:42',12,31,'NO'),(100,4,'2019-07-20
20:35:30',100,17,'NO');
/*!40000 ALTER TABLE `Cart_Product` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `Category`
--

DROP TABLE IF EXISTS `Category`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `Category` (
  `CategoryID` int(11) NOT NULL AUTO_INCREMENT,
  `Category_Name` varchar(50) NOT NULL,
  `Category_DESC` text,
  PRIMARY KEY (`CategoryID`),
  UNIQUE KEY `UC_Category` (`CategoryID`,`Category_Name`)
) ENGINE=InnoDB AUTO_INCREMENT=26 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
/*!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 (1,'Automotive & Powersports','New, Certified Refurbished,
Used, Collectible'),(2,'Baby Products','New only'),(3,'Beauty','New only'),(4,'Books','New,
used'),(5,'Business Products (B2B)','New, Certified Refurbished, Used'),(6,'Camera & Photo','New,
Certified Refurbished, Used'),(7,'Cell Phones','New, Certified Refurbished, Used,
Unlocked'),(8,'Clothing & Accessories','New only'),(9,'Collectible
Coins','Collectible'),(10,'Electronics (Accessories)','New, Certified Refurbished,
Used'),(11,'Fashion Jewelry','New only'),(12,'Fine Jewelry','New only'),(13,'Fine
Art','Collectible'),(14,'Grocery & Gourmet Food','New only'),(15,'Handmade','New
only'),(16,'Health & Personal Care','New only'),(17,'Historical & Advertising
Collectibles','Collectible'),(18,'Home & Garden','New, Certified Refurbished, Used,
Collectible'),(19,'Industrial & Scientific','New only'),(20,'Luggage & Travel Accessories','New
only'),(21,'Music','New, Used, Collectible'),(22,'Musical Instruments','New, Certified Refurbished,
Used, Collectible'),(23,'Office Products','Professionals only'),(24,'Shoes, Handbags &
Sunglasses','New only'),(25,'Outdoors','New, Certified Refurbished, Used');
/*!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 */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `Customer` (
  `CustomerID` int(11) NOT NULL AUTO_INCREMENT,
  `First_Name` varchar(50) NOT NULL,
  `Last_Name` varchar(50) NOT NULL,
  `Email` varchar(50) NOT NULL,
  `Phone_No` varchar(50) DEFAULT NULL,
  `AddressID` int(11) NOT NULL,
  `CartID` int(11) NOT NULL,
  PRIMARY KEY (`CustomerID`),
  KEY `AddressID` (`AddressID`),
  KEY `CartID` (`CartID`),
  KEY `idx_cname` (`Last_Name`,`First_Name`),
```

```
  CONSTRAINT `customer_ibfk_1` FOREIGN KEY (`AddressID`) REFERENCES `address`
(`AddressID`),
  CONSTRAINT `customer_ibfk_2` FOREIGN KEY (`CartID`) REFERENCES `cart` (`CartID`)
) ENGINE=InnoDB AUTO_INCREMENT=102 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
/*!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,'Walter','Huber','magnis@ipsumDonec.co.uk','1-833-261-
7822',12,2),(2,'Adele','Mccormick','tincidunt@mollis.net','904-
1519',81,3),(3,'Suki','Kinney','a.malesuada.id@maurissapien.org','1-903-715-
1597',29,5),(4,'Doris','Hawkins','porttitor.eros@velsapienimperdiet.org','1-408-236-
1613',52,3),(5,'Keelie','Baker','Nulla@mauris.com','575-
8226',43,4),(6,'Macon','Roman','Integer.vulputate.risus@nonnisi.com','1-186-493-
8603',68,6),(7,'Iona','Colon','molestie.tellus@Seddiam.ca','219-
2586',41,7),(8,'Justin','Mckinney','malesuada.malesuada@blanditmattisCras.co.uk','986-
4511',28,8),(9,'Ross','Mclaughlin','convallis.convallis@Aliquamgravida.co.uk','512-
0245',81,87),(10,'Felicia','Hanson','pede@nonummyacfeugiat.edu','1-945-979-
8176',50,50),(11,'Ian','Navarro','turpis.non@ipsumsodalespurus.com','1-308-132-
0987',15,15),(12,'Beck','Gill','elit.sed.consequat@laciniaat.org','617-
1122',56,56),(13,'Marny','Medina','sed.pede.Cum@blanditenim.ca','1-626-939-
7832',21,21),(14,'Connor','Wilkinson','vulputate@sollicitudin.net','112-
0588',54,54),(15,'Conan','Branch','penatibus.et.magnis@quisarcuvel.com','166-
6792',65,56),(16,'Silas','Baird','Duis.at@euismodac.ca','1-475-433-
1204',20,2),(17,'Lilah','Dodson','a.arcu.Sed@egestashendrerit.co.uk','987-
8835',26,62),(18,'Libby','Sandoval','Fusce.aliquet@duiCraspellentesque.net','579-
4773',21,12),(19,'Elmo','Randall','nunc@diamPellentesquehabitant.ca','374-
5940',38,83),(20,'Winifred','Houston','Cras.vulputate.velit@ultrices.edu','424-
9180',56,65),(21,'Lenore','Norton','dictum.Phasellus.in@risus.ca','1-267-925-
1945',28,82),(22,'Hasad','Hess','a.facilisis@est.co.uk','600-
5403',91,19),(23,'Hasad','Beach','imperdiet@acmieleifend.com','289-
8344',53,35),(24,'Chester','Rocha','libero.Integer@consectetuercursus.edu','486-
7320',16,61),(25,'Timon','Atkins','ullamcorper.nisl@euismodac.co.uk','860-
6954',25,52),(26,'Erich','Watkins','elementum.purus@Curabiturdictum.com','624-
6167',75,57),(27,'Amela','Griffith','odio.Aliquam.vulputate@sapienimperdiet.com','867-
5576',77,77),(28,'Remedios','England','lectus.quis.massa@sociis.org','1-556-785-
5984',31,13),(29,'Alma','Melton','feugiat.non.lobortis@lectus.net','1-498-237-
8027',84,48),(30,'Lani','Roman','Proin.nisl@vitaeeratvel.net','131-
4045',94,49),(31,'Basia','Brennan','pede.ultrices@miac.com','747-
```

4858',1,1),(32,'Shea','Leblanc','dictum.eu.placerat@eratnequenon.com','1-796-712-
0246',5,10),(33,'Victor','Alford','Aliquam.rutrum@posuerecubiliaCurae.net','773-
7680',72,27),(34,'Garrison','Kaufman','Aenean@dictumsapien.net','907-
3031',30,34),(35,'Azalia','Slater','penatibus@urna.com','745-
6970',16,61),(36,'Ezra','Herrera','ornare.In.faucibus@urnasuscipit.edu','330-
2304',1,82),(37,'Gisela','Blackwell','nibh.enim@morbi.co.uk','354-
9065',27,72),(38,'Dean','Garrison','arcu@In.com','954-
8957',42,24),(39,'Gail','Keller','Nulla.dignissim@convalliserat.net','1-468-396-
2667',43,36),(40,'Venus','Schmidt','aliquet@Nullasempertellus.org','1-496-811-
3484',52,35),(41,'Xavier','Walls','eu.odio@felis.com','1-379-328-
1240',3,9),(42,'Unity','Levy','molestie.Sed@Duissit.com','998-
2828',86,68),(43,'Laith','Montgomery','lorem.ipsum.sodales@metusfacilisislorem.org','1-798-
677-1119',100,100),(44,'Aladdin','Malone','congue.In.scelerisque@dolorvitaedolor.co.uk','1-560-
348-2505',26,29),(45,'Elijah','Callahan','dolor.tempus@massaSuspendisse.net','885-
1684',21,23),(46,'Alisa','Watkins','Curabitur.ut.odio@estac.edu','1-328-792-
3174',59,95),(47,'Wayne','Burks','scelerisque@anteblanditviverra.org','1-962-893-
5327',89,45),(48,'Winifred','Rush','rutrum.magna@torquentperconubia.ca','933-
3585',6,46),(49,'Olympia','Barker','pede.malesuada@necimperdietnec.org','569-
2750',77,48),(50,'Martin','Mccray','sit.amet.consectetuer@egestashendrerit.edu','622-
7367',17,19),(51,'Robin','Shepard','Vivamus.nibh@Suspendisse.net','276-
9574',88,82),(52,'Althea','Hurley','ac.arcu@odio.com','1-799-586-
3451',76,63),(53,'Cara','Hammond','eros@Aliquamvulputate.net','655-
9037',14,15),(54,'Wayne','Oneill','turpis@tellusfaucibusleo.net','842-
5177',41,41),(55,'Wyoming','Conrad','sapien@quispede.ca','1-833-400-
4579',100,46),(56,'Kelly','Mcmahon','risus.at@tortor.org','954-
7636',12,17),(57,'Megan','Valencia','tempus.eu@mollisInteger.ca','514-
9135',60,60),(58,'Kelly','Warren','fringilla.Donec.feugiat@semper.co.uk','1-178-639-
5214',82,70),(59,'Marcia','Davidson','elit.pellentesque.a@ipsumprimis.com','131-
3716',47,72),(60,'Alden','Clarke','dictum.mi@dui.com','1-225-978-
9517',67,56),(61,'Lane','Mendez','sem.mollis.dui@lobortisaugue.co.uk','1-201-685-
6174',63,33),(62,'Channing','Hicks','eleifend.Cras.sed@veliteget.net','253-
4795',67,44),(63,'Lance','Mosley','consequat.lectus@placerategetvenenatis.co.uk','1-180-420-
1176',69,55),(64,'Skyler','Mcdaniel','tincidunt.Donec@etmalesuadafames.edu','943-
0548',60,66),(65,'Garrison','Cole','accumsan@Sed.edu','1-287-578-
0081',24,43),(66,'Daria','Ross','Sed.congue@atnisi.co.uk','825-
3118',85,55),(67,'Beverly','Schneider','arcu@Proin.com','1-579-764-
5734',100,69),(68,'Jelani','Cobb','Proin.non@gravidaAliquam.net','542-
2348',91,79),(69,'Lucian','Kaufman','lorem.tristique@duinec.co.uk','367-
7794',30,89),(70,'Ursula','Farley','quam.a.felis@nequetellusimperdiet.edu','919-
1991',17,18),(71,'Imelda','Decker','consectetuer.ipsum@pharetra.ca','671-
6356',10,12),(72,'Shad','Vincent','eu@odio.co.uk','107-
8067',11,13),(73,'Sheila','Goff','nibh.enim.gravida@turpisnonenim.edu','1-506-205-
6547',9,14),(74,'Inez','Marks','mollis.Duis@lectussit.co.uk','182-
3570',37,39),(75,'Raya','Webster','mi.enim@quisurnaNunc.net','766-

```
3227',93,38),(76,'Elaine','Kim','bibendum@feugiatSednec.edu','606-
4012',56,58),(77,'Freya','Leon','per@ac.co.uk','1-486-142-
7290',47,49),(78,'Gil','Chan','orci@nuncnullavulputate.edu','663-
0411',62,79),(79,'Nell','Frederick','dignissim@molestie.co.uk','1-123-596-
0890',60,61),(80,'Harper','Patel','arcu.iaculis@enimmi.com','460-
2352',18,17),(81,'Leigh','Aguirre','enim.Nunc.ut@arcueu.net','264-
5977',99,94),(82,'Dustin','Nguyen','dapibus@molestie.edu','1-613-323-
9131',13,42),(83,'Hedley','Dunlap','vel.pede@ultrices.com','1-409-477-
4970',37,55),(84,'Carl','Wiggins','tellus.justo.sit@aliquetmolestie.edu','1-341-209-
5348',75,73),(85,'Graham','Fowler','molestie@sit.edu','776-
6997',2,71),(86,'Todd','Cline','habitant@Praesent.co.uk','1-329-689-
0834',86,81),(87,'Yoshi','Mason','vitae@Namnulla.org','1-128-611-
6044',53,91),(88,'Alvin','Church','magna.nec@PhasellusornareFusce.net','693-
4893',49,51),(89,'Jael','Sherman','pretium.et.rutrum@adipiscinglacusUt.net','1-990-331-
4409',10,24),(90,'Eugenia','Barnett','habitant.morbi@orciUt.com','1-785-467-
0895',7,27),(91,'Giacomo','Gray','torquent.per@duilectusrutrum.org','194-
5951',37,97),(92,'Harrison','Avery','cursus@atiaculisquis.ca','1-397-335-
8662',24,34),(93,'Hedwig','Watson','ad@consequatlectus.net','1-661-691-
4228',47,56),(94,'Jocelyn','Golden','hendrerit@risus.net','289-
2667',23,11),(95,'Hop','Dudley','pede.ultrices.a@convallis.net','711-
1244',26,13),(96,'Tyler','Yates','et@vitaesodales.com','419-
1032',21,46),(97,'Stella','Skinner','magna@cubiliaCuraeDonec.edu','372-
2714',39,38),(98,'Lesley','Burris','aliquet@dolorFuscefeugiat.co.uk','1-960-523-
9013',50,59),(99,'Inez','Walters','amet.risus@lectus.ca','877-
8780',84,87),(100,'Aretha','Boone','aliquet@etrisusQuisque.co.uk','1-104-163-
2303',81,77),(101,'Nidhi','Goyal','goyal.ni@husky.neu.edu','2016260890',23,24);
/*!40000 ALTER TABLE `Customer` 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              =
'ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FO
R_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION' */ ;
DELIMITER ;;
/*!50003 CREATE*/ /*!50017 DEFINER=`root`@`localhost`*/ /*!50003 TRIGGER
`Customer_BEFORE_INSERT_NULL` BEFORE INSERT ON `customer` FOR EACH ROW BEGIN
IF NEW.Email = '' THEN
SET NEW.Email = NULL;
ELSEIF
```

```
NEW.Phone_No = '' THEN
SET NEW.Phone_No = NULL;
END IF;
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 `Customer_Login`
--

DROP TABLE IF EXISTS `Customer_Login`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `Customer_Login` (
  `Customer_LoginID` int(11) NOT NULL AUTO_INCREMENT,
  `Username` varchar(20) NOT NULL DEFAULT 'No Username',
  `Password` blob,
  `CustomerID` int(11) NOT NULL,
  PRIMARY KEY (`Customer_LoginID`),
  UNIQUE KEY `Customer_LoginID` (`Customer_LoginID`),
  KEY `CustomerID` (`CustomerID`),
  CONSTRAINT `customer_login_ibfk_1` FOREIGN KEY (`CustomerID`) REFERENCES `customer`
(`CustomerID`)
) ENGINE=InnoDB AUTO_INCREMENT=100 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `Customer_Login`
--

LOCK TABLES `Customer_Login` WRITE;
/*!40000 ALTER TABLE `Customer_Login` DISABLE KEYS */;
INSERT INTO `Customer_Login` VALUES (1,'mus.Donec.dignissim@',_binary
'LBT12CTJ8PN',19),(2,'consequat@vitaevelit',_binary
'ONF61BJE6LJ',82),(3,'ipsum@Aeneaneuismodm',_binary
'ORC71GNF5ZA',30),(4,'consectetuer.adipisc',_binary
'GCO49XPY4VJ',59),(5,'sollicitudin@nonleoV',_binary
'LNG36DFT3SS',78),(6,'laoreet@semsemperera',_binary
'BQZ03DAV3OZ',70),(7,'blandit@lobortismaur',_binary
```

'PFV97OVT3MJ',7),(8,'faucibus.id.libero@a',_binary
'ZCB95WWA4AF',73),(9,'non.justo.Proin@sedo',_binary
'CWM00MAE8OL',100),(10,'semper@duinec.ca',_binary
'EVO93FIQ2AX',63),(11,'Sed@posuereatvelit.c',_binary
'PVJ34ZAD5NS',15),(12,'lacus@infaucibus.net',_binary
'QWJ47GRG4TE',96),(13,'sapien@purusgravidas',_binary
'IFJ32GUT8VE',39),(14,'sodales.elit.erat@Qu',_binary
'NIE98GWW2LS',2),(15,'Integer.urna@non.com',_binary
'EYJ40NAK9BY',90),(16,'nisl.sem@torquentper',_binary
'HZV62WIH9CX',61),(17,'faucibus.lectus@quam',_binary
'ZRX78EMI4AN',100),(18,'egestas.urna.justo@v',_binary
'EPB17WZI6QZ',24),(19,'feugiat.non@lectusan',_binary
'LUS80VBN3VW',54),(20,'lobortis.ultrices.Vi',_binary
'BRF15FPZ1KA',23),(21,'erat.vel@vestibulumM',_binary
'DDA95RTM7FB',61),(22,'at.velit@facilisisfa',_binary
'HBZ24PXG0OQ',33),(23,'tempor.erat@enimnisl',_binary
'PAM43DSD6KZ',81),(24,'quam.Curabitur.vel@c',_binary
'QYP31OWI7UE',71),(25,'ac.arcu@facilisislor',_binary
'BVX20PPN7VJ',3),(26,'pede.ac.urna@sodales',_binary
'FTX16GRV4BM',48),(27,'eget@blanditviverra.',_binary
'CBX24KYD9UQ',37),(28,'nostra.per@quamPelle',_binary
'QTV28YOX3HG',35),(29,'sed.consequat.auctor',_binary
'ZBP39NJA0BF',16),(30,'risus.Quisque@sedest',_binary
'VSO65XTR9EH',16),(31,'lobortis.ultrices@ul',_binary
'ACS85TOH6MN',53),(32,'at.egestas@Nunclectu',_binary
'QGX15OVU4JH',90),(33,'penatibus.et@aliquet',_binary
'RWP84UOX6DD',52),(34,'volutpat.ornare.faci',_binary
'RCW46UUK6UO',8),(35,'tristique@facilisisn',_binary
'QRX39LRO8FD',45),(36,'ac@vitae.org',_binary
'TZD84PFF4ID',6),(37,'Vestibulum.ante@auct',_binary
'NNV93NKJ2PX',21),(38,'nec@Phasellusinfelis',_binary
'HPF50FCC2UD',75),(39,'erat.eget.tincidunt@',_binary
'XOF68LFM6XY',70),(40,'libero.Proin@cursus.',_binary
'BXQ00JOB6YH',3),(41,'mus@Quisquefringilla',_binary
'DJN02BNF3OR',27),(42,'pede.ac@dignissimMae',_binary
'UFA75GMA1UF',9),(43,'nulla@augue.com',_binary
'YWO03LWR7WN',29),(44,'non.egestas@elitdict',_binary
'ZKB86TSP9WP',78),(45,'dapibus.gravida@disp',_binary
'HWB84LXQ1VD',32),(46,'Duis@dictumeuplacera',_binary
'CNK46JUV6PW',22),(47,'amet.ultricies@massa',_binary
'WZO56VFU8OR',75),(48,'arcu@netus.ca',_binary
'MXQ39AVI8SO',97),(49,'tincidunt@sapien.org',_binary
'OEM12CZM0EV',43),(50,'gravida.mauris.ut@hy',_binary
'ZFU93YCJ8UM',65),(51,'consequat@Morbi.ca',_binary

'EYB87BTY4IT',53),(52,'Ut.nec.urna@pedemale',_binary
'BDN93FXV1PC',51),(53,'ipsum@nec.ca',_binary
'IQM97LQR1HP',9),(54,'Nulla@eulacus.org',_binary
'GIJ47RXM7BP',67),(55,'Fusce.mi@velconvalli',_binary
'SDD98FHR0JB',53),(56,'mauris.sit.amet@euli',_binary
'KMN29VFC4UN',19),(57,'neque@ultrices.net',_binary
'AJR72FPE8JU',79),(58,'augue@vitaemauris.co',_binary
'ODT56GSW2NY',31),(59,'orci.quis@sociosqu.c',_binary
'MCQ61ZUT1XL',72),(60,'ipsum.primis.in@mole',_binary
'WNR76EMD2EB',87),(61,'convallis.in@seddui.',_binary
'QWW64ADB5GU',63),(62,'dapibus.quam@magnaCr',_binary
'DCX75FJK4VK',41),(63,'Vestibulum.accumsan.',_binary
'WAG57UZC5LO',35),(64,'ipsum@nonummyFusce.n',_binary
'ZIV40WWV3BF',99),(65,'Duis@malesuadafamesa',_binary
'NDK47YJF5HB',53),(66,'mauris.Morbi.non@Qui',_binary
'WAZ20YEK8QF',94),(67,'egestas.a@quamvelsap',_binary
'UEW37RGG1CY',83),(68,'elementum.dui@Aenean',_binary
'XDU14WXB8KJ',96),(69,'Vivamus@ametdapibusi',_binary
'DGT80OPF3RH',14),(70,'urna.Nunc@Donecfeugi',_binary
'NIJ46XJY2JK',35),(71,'eu.dolor@nec.net',_binary
'HJR87KZB4SJ',56),(72,'interdum@velnisl.ca',_binary
'RED11MUK2ET',25),(73,'ullamcorper.Duis@ac.',_binary
'EBR88FCE4YM',98),(74,'turpis@nisiCum.net',_binary
'SYJ31KWI1RJ',51),(75,'ornare.In.faucibus@c',_binary
'FQW77FTK9UK',8),(76,'arcu@metus.edu',_binary
'TYM87MLA4QI',1),(77,'tincidunt.orci@at.co',_binary
'UMT41YNE0JT',26),(78,'lectus@feugiat.co.uk',_binary
'MNL17JSW6XB',8),(79,'nec.eleifend@Etiamim',_binary
'ZED39LBG6LQ',94),(80,'non.magna.Nam@euturp',_binary
'KYM12XWS0AH',16),(81,'dui@dui.ca',_binary
'AEC18CJU4CC',84),(82,'Phasellus.vitae.maur',_binary
'YYB28CNK7UY',66),(83,'sit.amet.ultricies@d',_binary
'AZT19EJH9LL',81),(84,'sed.hendrerit@noneni',_binary
'PAJ81LIH2KB',97),(85,'tempus.risus@afacili',_binary
'LKP00ANS3BI',100),(86,'ornare.elit.elit@tur',_binary
'UHP99QRP2SZ',40),(87,'et@Cras.edu',_binary
'ZZY68PMQ9IZ',32),(88,'id@duiSuspendisseac.',_binary
'YNI82VVM6UP',62),(89,'dolor@nectempus.org',_binary
'OOL20MQF0DR',30),(90,'Cras.lorem.lorem@atp',_binary
'XOW91GON0IR',69),(91,'in@tinciduntnequevit',_binary
'LWP60PKP6OK',55),(92,'Nam.ligula@Namportti',_binary
'JDE73MMC5XE',29),(93,'amet.ante@non.edu',_binary
'HBL13YOL2EA',91),(94,'montes.nascetur@asce',_binary
'UMU75IFR9GV',27),(95,'mi.felis.adipiscing@',_binary

```
'TIK54ZCI6CT',55),(96,'Mauris@Cras.ca',_binary
'AJC65CLF2DK',72),(97,'Aliquam.rutrum.lorem',_binary
'RSV14HXC3FV',83),(98,'Aliquam.ultrices@lec',_binary
'OAX02COI5UI',33),(99,'mi.felis.adipiscing@',_binary
'NAC72VSV0TZ',46),(100,'lectus@molestietellu',_binary 'LYF52KZN4OL',64);
/*!40000 ALTER TABLE `Customer_Login` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Temporary view structure for view `customer_use_americanexpress`
--

DROP TABLE IF EXISTS `customer_use_americanexpress`;
/*!50001 DROP VIEW IF EXISTS `customer_use_americanexpress`*/;
SET @saved_cs_client     = @@character_set_client;
/*!50503 SET character_set_client = utf8mb4 */;
/*!50001 CREATE VIEW `customer_use_americanexpress` AS SELECT
 1 AS `CustomerID`,
 1 AS `First_Name`,
 1 AS `Last_Name`,
 1 AS `Payment_Type`*/;
SET character_set_client = @saved_cs_client;

--
-- Temporary view structure for view `customer_use_mastercard`
--

DROP TABLE IF EXISTS `customer_use_mastercard`;
/*!50001 DROP VIEW IF EXISTS `customer_use_mastercard`*/;
SET @saved_cs_client     = @@character_set_client;
/*!50503 SET character_set_client = utf8mb4 */;
/*!50001 CREATE VIEW `customer_use_mastercard` AS SELECT
 1 AS `CustomerID`,
 1 AS `First_Name`,
 1 AS `Last_Name`,
 1 AS `Payment_Type`*/;
SET character_set_client = @saved_cs_client;

--
-- Temporary view structure for view `customer_use_visacard`
--

DROP TABLE IF EXISTS `customer_use_visacard`;
/*!50001 DROP VIEW IF EXISTS `customer_use_visacard`*/;
```

```
SET @saved_cs_client     = @@character_set_client;
/*!50503 SET character_set_client = utf8mb4 */;
/*!50001 CREATE VIEW `customer_use_visacard` AS SELECT
 1 AS `CustomerID`,
 1 AS `First_Name`,
 1 AS `Last_Name`,
 1 AS `Payment_Type`*/;
SET character_set_client = @saved_cs_client;

--
-- Table structure for table `Payment`
--

DROP TABLE IF EXISTS `Payment`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `Payment` (
  `PaymentID` int(11) NOT NULL AUTO_INCREMENT,
  `Payment_Type` varchar(50) DEFAULT NULL,
  `Payment_Date` datetime NOT NULL,
  `CustomerID` int(11) NOT NULL,
  `CartID` int(11) NOT NULL,
  PRIMARY KEY (`PaymentID`),
  KEY `CustomerID` (`CustomerID`),
  KEY `CartID` (`CartID`),
  CONSTRAINT `payment_ibfk_1` FOREIGN KEY (`CustomerID`) REFERENCES `customer`
(`CustomerID`),
  CONSTRAINT `payment_ibfk_2` FOREIGN KEY (`CartID`) REFERENCES `cart` (`CartID`)
) ENGINE=InnoDB AUTO_INCREMENT=51 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
/*!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,'VisaCard','2020-10-17 07:50:46',88,76),(2,'VisaCard','2020-
11-08 07:21:39',41,65),(3,'MasterCard','2020-06-16 13:42:40',52,53),(4,'AmericanExpress','2019-
09-23 04:11:01',69,86),(5,'VisaCard','2019-10-09 07:49:37',6,83),(6,'AmericanExpress','2019-08-
22 15:16:31',96,74),(7,'MasterCard','2020-01-20 14:07:27',1,52),(8,'VisaCard','2020-07-11
15:09:43',36,20),(9,'MasterCard','2020-04-11 15:32:12',18,22),(10,'VisaCard','2019-07-05
12:56:15',5,70),(11,'AmericanExpress','2020-01-14 05:37:32',65,4),(12,'MasterCard','2020-05-12
```

```
13:34:26',34,9),(13,'AmericanExpress','2019-11-17 01:18:35',44,79),(14,'MasterCard','2020-11-
20 05:46:44',49,97),(15,'VisaCard','2019-05-15 00:54:41',44,23),(16,'AmericanExpress','2019-08-
26 00:23:11',49,32),(17,'MasterCard','2019-01-12 06:25:13',54,24),(18,'AmericanExpress','2019-
10-07 12:57:39',41,75),(19,'MasterCard','2019-07-15 02:33:08',48,42),(20,'MasterCard','2019-
07-23 15:46:08',17,26),(21,'VisaCard','2019-11-23 14:13:39',83,35),(22,'AmericanExpress','2019-
07-03 00:18:13',15,47),(23,'MasterCard','2019-05-23 00:10:58',70,48),(24,'VisaCard','2019-05-28
04:25:51',26,42),(25,'MasterCard','2019-06-12 09:03:27',95,67),(26,'MasterCard','2019-03-16
09:38:56',87,12),(27,'AmericanExpress','2019-05-12 01:59:53',93,54),(28,'VisaCard','2020-10-21
16:54:51',67,40),(29,'AmericanExpress','2019-09-13 08:23:10',20,32),(30,'VisaCard','2020-01-06
10:54:25',21,15),(31,'VisaCard','2019-12-12 16:02:09',33,25),(32,'AmericanExpress','2018-12-22
13:45:16',29,41),(33,'MasterCard','2020-10-22 06:04:22',51,99),(34,'VisaCard','2019-12-10
23:03:14',5,34),(35,'AmericanExpress','2019-12-30 21:26:14',20,85),(36,'VisaCard','2019-07-26
00:16:03',86,96),(37,'MasterCard','2020-06-26 15:14:39',85,2),(38,'VisaCard','2019-09-14
07:11:38',24,43),(39,'AmericanExpress','2020-11-24
06:48:24',87,1),(40,'AmericanExpress','2019-10-16 16:46:33',34,73),(41,'VisaCard','2019-09-23
14:45:19',78,59),(42,'MasterCard','2019-06-01 20:08:10',30,86),(43,'AmericanExpress','2019-04-
12 12:42:51',90,65),(44,'VisaCard','2020-03-25 22:37:59',45,1),(45,'MasterCard','2020-01-24
11:09:42',42,90),(46,'AmericanExpress','2020-04-27
17:28:21',45,73),(47,'AmericanExpress','2020-07-25 04:41:12',18,63),(48,'VisaCard','2020-08-23
11:10:57',94,42),(49,'MasterCard','2020-02-22 03:00:52',87,89),(50,'VisaCard','2019-11-19
18:15:20',32,6);
/*!40000 ALTER TABLE `Payment` 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              =
'ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FO
R_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION' */ ;
DELIMITER ;;
/*!50003 CREATE*/ /*!50017 DEFINER=`root`@`localhost`*/ /*!50003 TRIGGER
`Payment_BEFORE_INSERT` BEFORE INSERT ON `payment` FOR EACH ROW BEGIN
IF NEW.PaymentID = ' ' THEN
SIGNAL SQLSTATE '45000';
END IF;
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 `Product`
--

DROP TABLE IF EXISTS `Product`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `Product` (
  `ProductID` int(11) NOT NULL AUTO_INCREMENT,
  `Product_Name` varchar(50) NOT NULL,
  `Product_Price` decimal(8,2) NOT NULL,
  `Product_Description` varchar(255) NOT NULL DEFAULT 'No Description',
  `CategoryID` int(11) NOT NULL,
  `BrandID` int(11) NOT NULL,
  `SellerID` int(11) NOT NULL,
  PRIMARY KEY (`ProductID`),
  KEY `BrandID` (`BrandID`),
  KEY `CategoryID` (`CategoryID`),
  KEY `SellerID` (`SellerID`),
  CONSTRAINT `product_ibfk_1` FOREIGN KEY (`BrandID`) REFERENCES `brand` (`BrandID`),
  CONSTRAINT `product_ibfk_2` FOREIGN KEY (`CategoryID`) REFERENCES `category`
(`CategoryID`),
  CONSTRAINT `product_ibfk_3` FOREIGN KEY (`SellerID`) REFERENCES `seller` (`SellerID`)
) ENGINE=InnoDB AUTO_INCREMENT=43 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `Product`
--

LOCK TABLES `Product` WRITE;
/*!40000 ALTER TABLE `Product` DISABLE KEYS */;
INSERT INTO `Product` VALUES (14,'File',4.00,'New Only',23,47,3),(15,'File',8.00,'New
Only',23,48,6),(16,'Pen',2.00,'New Only',23,22,8),(17,'Pen',4.00,'New
Only',23,22,21),(18,'File',8.00,'New Only',23,50,76),(19,'File',8.00,'New
Only',23,49,89),(20,'Bag',200.00,'New
Only',20,37,34),(21,'Bag',150.00,'Used',20,37,36),(22,'Bag',250.00,'New
Only',20,38,34),(23,'Bag',100.00,'New
Only',20,39,23),(24,'Mobile',670.00,'Refurbished',7,32,12),(25,'Mobile',470.00,'Refurbished',7,31
,11),(26,'Chain',1000.00,'New Only',11,41,87),(27,'Chain',1200.00,'New
Only',11,42,90),(28,'Chain',1500.00,'New Only',11,43,26),(29,'Chain',1800.00,'New
```

```
Only',11,44,87),(30,'Chain',1200.00,'New Only',11,45,54),(31,'Pen',4.00,'New
Only',23,46,64),(32,'Pencil',4.00,'New Only
Only',23,46,45),(33,'Mobile',500.00,'Refurbished',7,31,16),(34,'Laptop',1000.00,'Refurbished',10,
24,42),(35,'Mobile',970.00,'New Only',7,32,23);
/*!40000 ALTER TABLE `Product` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `Seller`
--

DROP TABLE IF EXISTS `Seller`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `Seller` (
  `SellerID` int(11) NOT NULL,
  `Seller_Name` varchar(50) NOT NULL,
  `AddressID` int(11) NOT NULL,
  PRIMARY KEY (`SellerID`),
  KEY `AddressID` (`AddressID`),
  CONSTRAINT `seller_ibfk_1` FOREIGN KEY (`AddressID`) REFERENCES `address` (`AddressID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `Seller`
--

LOCK TABLES `Seller` WRITE;
/*!40000 ALTER TABLE `Seller` DISABLE KEYS */;
INSERT INTO `Seller` VALUES (1,'Carl Andrews',45),(2,'Quamar Payne',72),(3,'Kameko
Gregory',60),(4,'Arthur Franklin',6),(5,'Anthony Suarez',48),(6,'Amelia Franks',11),(7,'Dalton
Ball',86),(8,'Candice Callahan',59),(9,'Adara Mays',28),(10,'Kalia Randolph',73),(11,'Victor
Acevedo',2),(12,'Robert Klein',17),(13,'Zeph Shepard',94),(14,'Mira Tate',23),(15,'Declan
Watkins',67),(16,'Marny Burns',35),(17,'Justine Conley',5),(18,'Sonya Potts',31),(19,'Shoshana
Baker',97),(20,'Harriet Hinton',30),(21,'Caldwell Rose',2),(22,'Montana Rutledge',21),(23,'Barrett
Macdonald',99),(24,'Christen Dejesus',2),(25,'Byron Shepherd',62),(26,'Adria
Mcguire',88),(27,'Marvin Boyd',8),(28,'Dylan Silva',98),(29,'Daquan Salazar',30),(30,'Wyoming
Cortez',2),(31,'Anne Johnston',40),(32,'Emma Quinn',8),(33,'Kelly Travis',75),(34,'Halla
Aguirre',38),(35,'Lester Mckinney',31),(36,'Alika Skinner',96),(37,'Fleur Mayer',34),(38,'Helen
Hall',89),(39,'Cassady Higgins',37),(40,'Tamara Hall',16),(41,'Deirdre Levy',17),(42,'Pascale
Blair',26),(43,'Catherine Moses',75),(44,'Emerson Fry',2),(45,'Maxwell Burris',39),(46,'Lee
Lamb',7),(47,'Alfonso Pollard',43),(48,'Castor Glenn',11),(49,'Justina Russell',31),(50,'Christian
Haney',47),(51,'Echo Atkinson',16),(52,'Sara Rhodes',84),(53,'Jakeem Wheeler',29),(54,'Deborah
```

```
Warner',95),(55,'Cadman Yates',29),(56,'Odessa Boyer',53),(57,'Barbara Kaufman',35),(58,'Gavin
Dejesus',4),(59,'Finn Stone',92),(60,'Victor Brown',35),(61,'Drew Harrington',14),(62,'Vaughan
Wilkins',18),(63,'Barry Riddle',17),(64,'Roary Chaney',56),(65,'Kyla Dodson',42),(66,'Carly
Castillo',98),(67,'Acton Patel',78),(68,'Inga Gray',56),(69,'Arsenio Solomon',26),(70,'Mariam
Dunlap',27),(71,'Joy Bullock',73),(72,'Prescott Compton',76),(73,'Ali Hyde',54),(74,'Madeson
Hickman',83),(75,'Cally Maldonado',69),(76,'Fiona Fleming',7),(77,'Simon Holder',22),(78,'Desirae
Hutchinson',31),(79,'Owen Mendez',8),(80,'Diana Burgess',100),(81,'Carissa
Beach',44),(82,'Lester Blankenship',13),(83,'Erasmus Mills',81),(84,'Hamish Kent',81),(85,'Suki
Parks',14),(86,'Timothy Sheppard',68),(87,'Venus Mcgee',92),(88,'Gannon Keller',4),(89,'Lionel
Larson',18),(90,'Lillith Rush',100),(91,'Holmes Morrow',63),(92,'Piper Mcintosh',94),(93,'Zia
Stevenson',57),(94,'Vincent Zamora',87),(95,'Charde Bullock',82),(96,'Amity
Sampson',6),(97,'Lacey Lott',83),(98,'Dora Beasley',68),(99,'Germaine Harrell',21),(100,'Dillon
Watts',18);
/*!40000 ALTER TABLE `Seller` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Dumping events for database 'Ecommerce'
--

--
-- Dumping routines for database 'Ecommerce'
--
/*!50003 DROP FUNCTION IF EXISTS `avg_price_of_product` */;
/*!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              = 'NO_AUTO_VALUE_ON_ZERO' */ ;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` FUNCTION `avg_price_of_product`(cid INT) RETURNS
int(11)
    READS SQL DATA
    DETERMINISTIC
BEGIN
DECLARE Avg INT;

 SET Avg = (
            SELECT AVG(Product_Price)
            FROM Product
            WHERE CategoryID = cid);
```

```
RETURN Avg;
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 `total_products_for_category` */;
/*!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              = 'NO_AUTO_VALUE_ON_ZERO' */ ;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` FUNCTION `total_products_for_category`(cid INT) RETURNS
int(11)
    READS SQL DATA
    DETERMINISTIC
BEGIN
DECLARE Total INT;

 SET Total = (
            SELECT COUNT(ProductID)
            FROM Product
            WHERE CategoryID = cid
            HAVING COUNT(ProductID) > 3);

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 FUNCTION IF EXISTS `total_products_of_seller` */;
/*!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              = 'NO_AUTO_VALUE_ON_ZERO' */ ;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` FUNCTION `total_products_of_seller`(sellerid INT) RETURNS
int(11)
    READS SQL DATA
    DETERMINISTIC
BEGIN
            DECLARE total INT;
            SELECT COUNT(Quantity) INTO total
            FROM Cart_Product CP
    INNER JOIN Product P
    INNER JOIN Seller S
    WHERE CP.ProductID = P.ProductID
    AND P.SellerID = S.SellerID
            AND S.SellerID = sellerid;
            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 `price_filter` */;
/*!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              = 'NO_AUTO_VALUE_ON_ZERO' */ ;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` PROCEDURE `price_filter`(IN price INT, IN name
VARCHAR(50))
    READS SQL DATA
BEGIN
            SELECT ProductID, Product_Name, Product_Price
    FROM Product
    WHERE Product_Price < PRICE
    AND Product_Name = name;
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 `verify_customer_login` */;
/*!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              = 'NO_AUTO_VALUE_ON_ZERO' */ ;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` PROCEDURE `verify_customer_login`(IN username
VARCHAR(1000), IN password VARCHAR(1000))
    READS SQL DATA
BEGIN
            SELECT IF(count(*) > 0, 0, 1) AS Result
    FROM Customer_Login
    WHERE Username = username
    AND Password = password;
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 `customer_use_americanexpress`
--

/*!50001 DROP VIEW IF EXISTS `customer_use_americanexpress`*/;
/*!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 `customer_use_americanexpress` AS select `C`.`CustomerID` AS
`CustomerID`,`C`.`First_Name` AS `First_Name`,`C`.`Last_Name` AS
```

```
`Last_Name`,`P`.`Payment_Type` AS `Payment_Type` from (`payment` `P` join `customer` `C`)
where ((`C`.`CustomerID` = `P`.`CustomerID`) and (`P`.`Payment_Type` = 'AmericanExpress')) */;
/*!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 `customer_use_mastercard`
--

/*!50001 DROP VIEW IF EXISTS `customer_use_mastercard`*/;
/*!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 `customer_use_mastercard` AS select `C`.`CustomerID` AS
`CustomerID`,`C`.`First_Name` AS `First_Name`,`C`.`Last_Name` AS
`Last_Name`,`P`.`Payment_Type` AS `Payment_Type` from (`payment` `P` join `customer` `C`)
where ((`C`.`CustomerID` = `P`.`CustomerID`) and (`P`.`Payment_Type` = 'MasterCard')) */;
/*!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 `customer_use_visacard`
--

/*!50001 DROP VIEW IF EXISTS `customer_use_visacard`*/;
/*!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 `customer_use_visacard` AS select `C`.`CustomerID` AS
`CustomerID`,`C`.`First_Name` AS `First_Name`,`C`.`Last_Name` AS
`Last_Name`,`P`.`Payment_Type` AS `Payment_Type` from (`payment` `P` join `customer` `C`)
where ((`C`.`CustomerID` = `P`.`CustomerID`) and (`P`.`Payment_Type` = 'VisaCard')) */;
```

```
/*!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 2019-12-10 17:03:58
```