**CS5200 DATABASE THEORY AND APPLICATIONS**

**FALL 2022**

**ONLINE RETAIL APPLICATION DATABASE**

**GROUP MEMBERS.**

|  |  |
| --- | --- |
| **NAME** | **ID** |
| **1. IAKSHMI KUNDANA KUNDURU** | **700734275** |
| **2. MANIDEEPIKA SOMA** | **700734192** |
| **3. VINITHA PATTEPURAM** | **700739902** |
| **4. PRADEEP REDDY GANGIDI** | **700743703** |

**Table of Contents**

[**1. Project Description** 3](#_Toc120216522)

[**2. Objectives** 4](#_Toc120216523)

[**3. Database Design Process and the General Database Structure** 4](#_Toc120216524)

[**4. SQL Tables, functionalities, ER diagram and The Data Descriptions.** 5](#_Toc120216525)

[**4.1 SQL Tables and Functionality Descriptions of Online Retail Schemas.** 5](#_Toc120216526)

[**4.2 Database Entity-Relationship Diagram** 10](#_Toc120216527)

[**SQL Create statements of the tables to be implemented** 10](#_Toc120216528)

[**Functional Dependencies of The Above Relations** 15](#_Toc120216529)

[**Sample Data** 16](#_Toc120216530)

[**5.Reports, queries and sample output** 19](#_Toc120216531)

[**6.** **Conclusion** 22](#_Toc120216532)

[**7.** **References** 22](#_Toc120216533)

# **1. Project Description**

Businesses Environment needs to focus on database technologies for the visibility of their daily Retailing activities. Commercially-oriented Databases, is applicable in Point of Sale, e-commerce site and customer Management application suite. Thus, to keep Valuable Data about Retailers, customers and shareholders, integration Of an Online Retail Application Database with the respective Shopping Based Applications is necessary. Moreover, customer Visibility on best Retailers, Previous purchases, enhanced retail outlet Acquisition, increasing customer base and retention of the “loyal” customers by the retailers, is the core use of the online Database application.

Online Retail Database, is a virtual storage system for the Retailers and the Point of Sale Transactions. Critical Retail sales Information are kept in the Database to either drive an E-commerce site or any customer Management applications. This Database Project tends to store information that can easily be inspected and Logs Of files created from it to track the information streaming in the Database. Integration of Retail Application with the Back-end Database is the core reason that paves way for keeping tracks, managing trends and predicting future Events that would unfold in the field of Retailing and Point of Sale services in a super-business Environment.

Retailers, Customers, Producers, Business, Business Groups among others, are the top target individuals who would utilize the Database. Therefore, Online Retail Application Database keeps track of all the retail business information fetched for the purpose of allocating retailer’s resources accordingly. Relational Databases tends to be necessary for virtual retailers. With customer Information, storage and analyzation is the biggest flex for Virtual Commercial Applications. Via this Database application project, various attributes and their data types is held for future Commercial Use. Instances of Information that will be stored is, personal user contacts, their names, user expenditures, Purchasing History and demographic details.

The Database will be implemented to enhance storage and tracking of Online Retailers data, customer information, products details among others, Thus, Online Retail Application Database will enhance automation of record Keeping in the Field of Economics, trade and retail services by the General Public.

# **2. Objectives**

Provision of Quality Customer and retailers’ services dwells solely on the Database system that ought to be blended in Customer Retail Application suite. Both Retailers and Potential Customers are the Main target of this e-Retail Database application Project. Therefore, this project is aimed at building a back-end database application to store Valuable Information about customers and retailers, their respective data will include and not limited to email addresses, the General population and the purchasing History.

**Main Objectives of Online Retail application Database**

1. To reduce time taken by retailers manage customer information.
2. To pave way for analytical success of the collected Retail Information.
3. To convert a seemingly meaningless customer and retailer’s information, by improving both its quality and consistency.

# **3. Database Design Process and the General Database Structure**

This project is anticipated to have over ten Relational tables, with nearly all the cardinalities except many to many relationships. Aggregation, Association, the implementation of Primary Keys, Foreign Keys and the database Normalization up-to third Normal Form will as well be implemented. RetailersBranches, retailer Barnch managers table, settings, purchased Order, Retail categories customer, retailer, orderedDetails, cart, category and table customer among others are the Relational Tables to be Implemented in this Project.

**Database Structure to be implemented.**

A customer can register to purchase a product. The Customer will provide methods of payment account number and usernames. After Registration, each customer will have a unique customerid, userid, and password.A Customer can purchase one or more products in different quantities. The product items can have different Classes based on their prices. Based on the quantity, the price of the item and discount (if any) on the purchased items, the bill will be generated. A payment method is required to settle the bill. The items can be ordered to one or more retailers.

Table Products, branches, users, ordered details, customer, settings, managers, RetailCategory, and cart will implement one to many Relationship cardinalities. Thus, they will be normalized into the Third Normal form with foreign Keys and primary keys to reference more than one Table. SQL View creations will as well be implemented to allow query Operations of specific Information.

# **4. SQL Tables, functionalities, ER diagram and The Data Descriptions.**

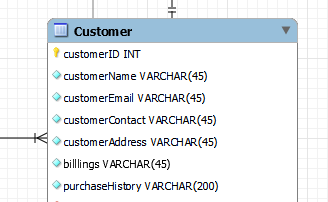
## **4.1 SQL Tables and Functionality Descriptions of Online Retail Schemas.**

Online Retail Application Database for keeping all customer and retail informationwill contain the following relational Table.

1. **Customer relation**

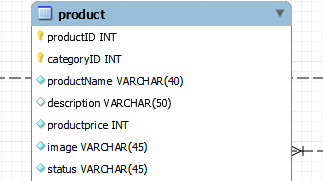
This table will be entrusted for storing and organizing information of customers making Purchases and transacting with the Retailers. Thus, all the Relational information of customers will be stored in this Schema. The schema below depicts the properties of information to be stored in the relation.

CustomerID is the Primary key column with other columns that stores custsomers Persona Information and their Purchasing History.



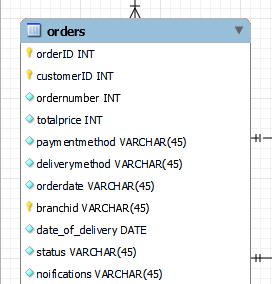
1. **Product relation**

This relation will be used in the online Retail application Database for storing product’s information based on the tuple characteristics shown in the schema below. In this schema, ProductID will be the primary key and CategoryID the foreign Key to reference the category of Purchased Products.

****

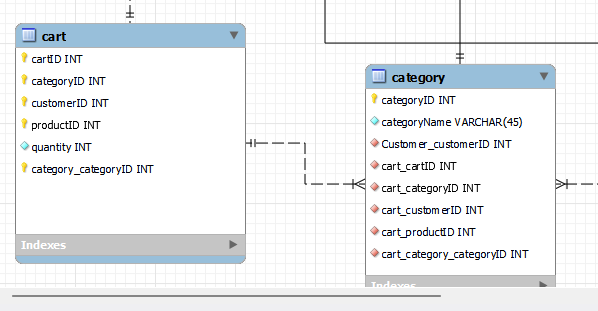
1. **Orders relation**

This table will be used for storing all the information of the orders processed before and after Purchases.



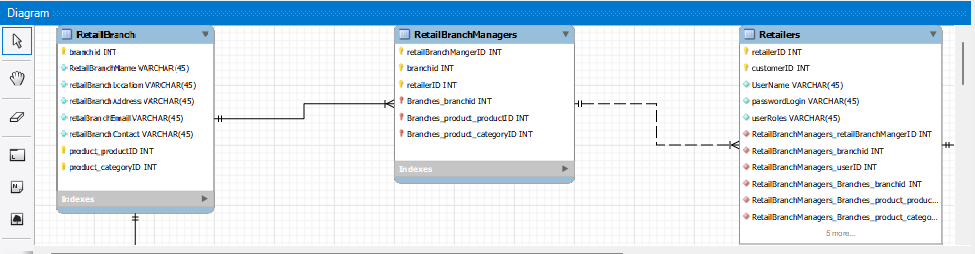
1. **Cart and Category relations**

Cart table will be used for storing relational information from the top three Online Retail schemas whereas category relation will be used to store the product categories both in the cart and orders table. The schemas below clearly depict the attributes of both carts’ and Products’ entity.



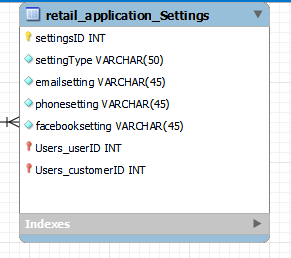
1. **RetailBranch, RetailBranchManagers and Retailers’ relations**

This schema will be implemented to ascertain the relational tables for keeping the users history and bioinformation in the e-retail application Database project. RetailBranch table stores information about branches that retails products dealt in by both small scale and large scale or wholesale retailers, whereas the other schema is used for keeping and tracking bio-data of both branch managers and retailers. BranchID, retailBranchMangerID and RetailerID are the primary Unique keys whereas branchID, and CustomerID in the later mentioned Tables, are the foreign Keys that references customer and branch tables.



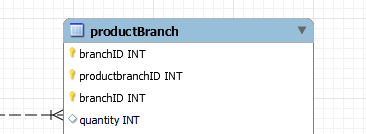
1. **Retail application Settings Table**

This table will be vital for the storage and keeping Track Of all Account Settings in the system. Thus, it keeps track of CRUD operations (create, read, update and delete) on retail information in the system. SettingsID is the primary Key. Its Attributes will be as shown in the schema below: -



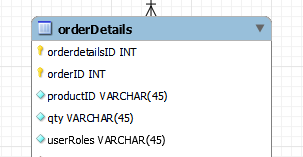
1. **ProductBranch relation**

This Tables will be used in relational database to pave way for insertion of Products information and referencing branch data with the branchID as the foreign Key. Schema is as shown below.

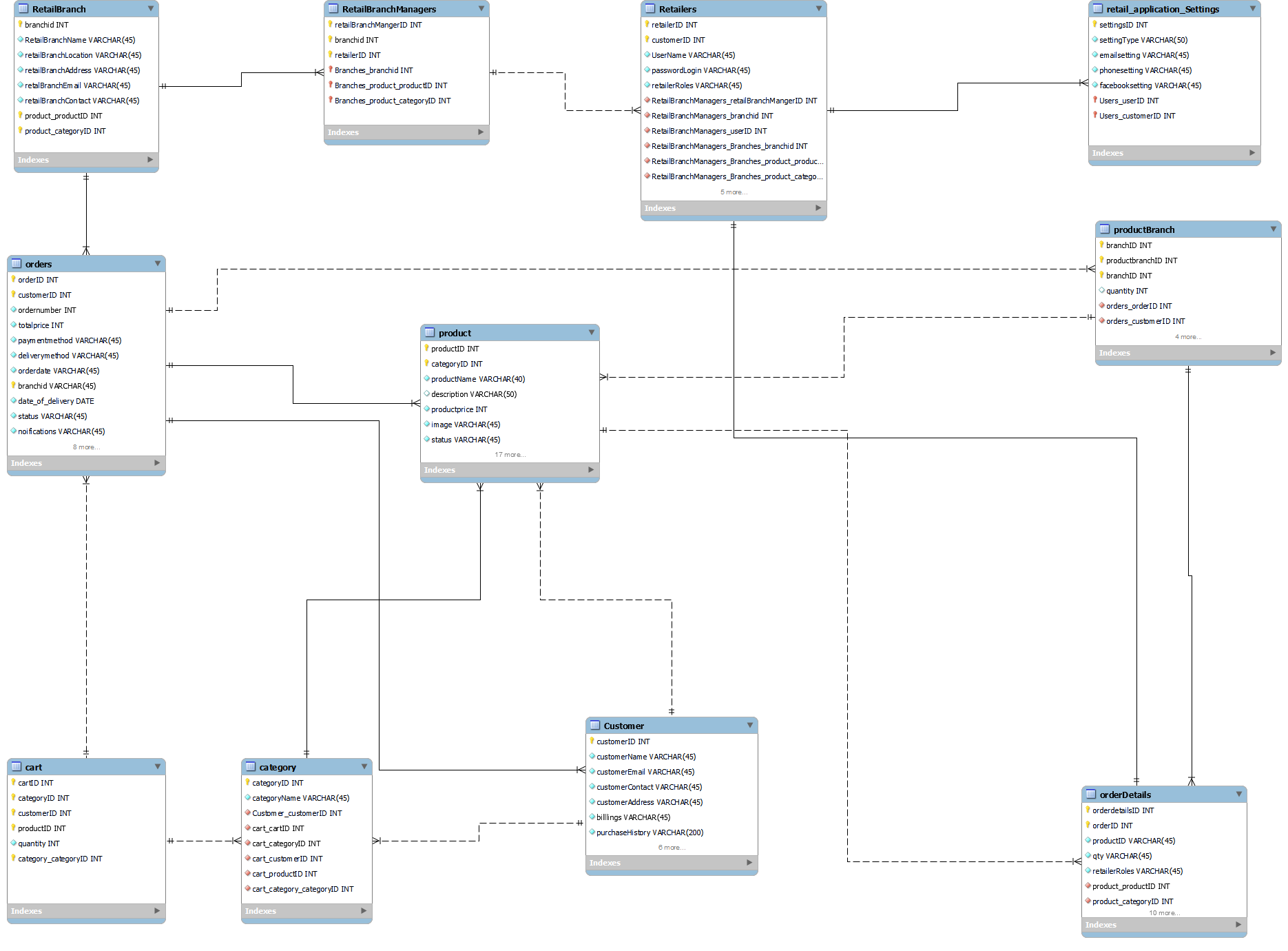


1. **OrderDetails relation**

This relation will be used for storing orders detailed Information in the developed relational virtual Retail application system. The following illustration is a schema table of the above-mentioned relation. orderID and orderDetailsID is the Primary Key and the Foreign Key respectively. OrderID (FK), fetches all data and purchase History of the ordered products, their category, quantity, among others.



**4.2 Database Entity-Relationship Diagram**



*Figure. Online Retail Application Database ERD.*

### **SQL Create statements of the tables to be implemented**

1. **Create statement for table customer**

*CREATE TABLE IF NOT EXISTS `Customer` (*

*`customerID` INT NOT NULL,*

*`customerName` VARCHAR(45) NOT NULL,*

*`customerEmail` VARCHAR(45) NOT NULL,*

*`customerContact` VARCHAR(45) NOT NULL,*

*`customerAddress` VARCHAR(45) NOT NULL,*

*`billlings` VARCHAR(45) NOT NULL,*

*`purchaseHistory` VARCHAR(200) NOT NULL,*

*PRIMARY KEY (`customerID`))*

*ENGINE = InnoDB;*

**Create statement for table retailBranch**

*CREATE TABLE IF NOT EXISTS `RetailBranch` (*

*`branchid` INT NOT NULL,*

*`RetailBranchName` VARCHAR(45) NOT NULL,*

*`retailBranchLocation` VARCHAR(45) NOT NULL,*

*`retailBranchAddress` VARCHAR(45) NOT NULL,*

*`retalBranchEmail` VARCHAR(45) NOT NULL,*

*`retailBranchContact` VARCHAR(45) NOT NULL,*

*PRIMARY KEY (`branchid`))*

*ENGINE = InnoDB;*

**Create statement for table retailBranchManagers**

*CREATE TABLE IF NOT EXISTS `RetailBranchManagers` (*

*`retailBranchMangerID` INT NOT NULL,*

*`branchid` INT NOT NULL,*

*`retailerID` INT NOT NULL,*

*PRIMARY KEY (`retailBranchMangerID`, `branchid`, `retailerID`))*

*ENGINE = InnoDB;*

**Create statement for table retailers**

*CREATE TABLE IF NOT EXISTS `Retailers` (*

*`retailerID` INT NOT NULL,*

*`customerID` INT NOT NULL,*

*`UserName` VARCHAR(45) NOT NULL,*

*`passwordLogin` VARCHAR(45) NOT NULL,*

*`retailerRoles` VARCHAR(45) NOT NULL,*

*PRIMARY KEY (`retailerID`, `customerID`))*

*ENGINE = InnoDB;*

**Create statement for table cart**

*CREATE TABLE IF NOT EXISTS `cart` (*

*`cartID` INT NOT NULL,*

*`categoryID` INT NOT NULL,*

*`customerID` INT NOT NULL,*

*`productID` INT NOT NULL,*

*`quantity` INT NOT NULL,*

*PRIMARY KEY (`cartID`, `categoryID`, `customerID`, `productID`))*

*ENGINE = InnoDB;*

**Create statement for table category**

*CREATE TABLE IF NOT EXISTS `category` (*

*`categoryID` INT NOT NULL,*

*`categoryName` VARCHAR(45) NOT NULL,*

*`Customer\_customerID` INT NOT NULL,*

*`cart\_cartID` INT NOT NULL,*

*`cart\_categoryID` INT NOT NULL,*

*`cart\_customerID` INT NOT NULL,*

*`cart\_productID` INT NOT NULL,*

*`cart\_category\_categoryID` INT NOT NULL,*

*PRIMARY KEY (`categoryID`))*

*ENGINE = InnoDB;*

**Create statement for table orderDetails**

*CREATE TABLE IF NOT EXISTS `orderDetails` (*

*`orderdetailsID` INT NOT NULL,*

*`orderID` INT NOT NULL,*

*`productID` VARCHAR(45) NOT NULL,*

*`qty` VARCHAR(45) NOT NULL,*

*`retailerRoles` VARCHAR(45) NOT NULL,*

*PRIMARY KEY (`orderdetailsID`, `orderID`, `productID`))*

*ENGINE = InnoDB;*

**Create statement for table orders**

*CREATE TABLE IF NOT EXISTS `orders` (*

*`orderID` INT NOT NULL,*

*`customerID` INT NOT NULL,*

*`ordernumber` INT NOT NULL,*

*`totalprice` INT NOT NULL,*

*`paymentmethod` VARCHAR(45) NOT NULL,*

*`deliverymethod` VARCHAR(45) NOT NULL,*

*`orderdate` VARCHAR(45) NOT NULL,*

*`branchid` VARCHAR(45) NOT NULL,*

*`date\_of\_delivery` DATE NOT NULL,*

*`status` VARCHAR(45) NOT NULL,*

*`noifications` VARCHAR(45) NOT NULL,*

*PRIMARY KEY (`orderID`, `customerID`, `branchid`))*

*ENGINE = InnoDB;*

**Create statement for table product**

*CREATE TABLE IF NOT EXISTS `product` (*

*`productID` INT NOT NULL,*

*`categoryID` INT NOT NULL,*

*`productName` VARCHAR(40) NOT NULL,*

*`description` VARCHAR(50) NULL,*

*`productprice` INT NOT NULL,*

*`image` VARCHAR(45) NOT NULL,*

*`status` VARCHAR(45) NOT NULL,*

*PRIMARY KEY (`productID`, `categoryID`))*

*ENGINE = InnoDB;*

**Create statement for table productBranch**

*CREATE TABLE IF NOT EXISTS `productbranch` (*

*`branchID` INT NOT NULL,*

*`productbranchID` INT NOT NULL,*

*`branchID` INT NOT NULL,*

*`quantity` INT NULL,*

*PRIMARY KEY (`branchID`, `productbranchID`, `branchID`))*

*ENGINE = InnoDB;*

**Create statement for table retail-application-setings**

*CREATE TABLE IF NOT EXISTS `retail\_application\_Settings` (*

*`settingsID` INT NOT NULL,*

*`settingType` VARCHAR(50) NOT NULL,*

*`emailsetting` VARCHAR(45) NOT NULL,*

*`phonesetting` VARCHAR(45) NOT NULL,*

*`facebooksetting` VARCHAR(45) NOT NULL,*

*PRIMARY KEY (`settingsID`))*

*ENGINE = InnoDB;*

### **Functional Dependencies of The Relations**

The attributes used in this relational database project, consists of several other non-primitive attributes which are transitive and dependent on the primary key and creates non-trivial dependencies with the respective derived super key. The following are the depicted unique or different determinants and dependent variable keys used in the created Enhanced entity relational Database Above. From the create statement formulated above the following are the depicted functional dependencies extracted: -

* **CustomerID**->customerName, CustomerEmail, CustomerContact, customerAddress, billings, PurchaseHistory
* **ProductID**->categoryID, productName,description, productPrice,image,status
* **productbranchID**->branchID,quanity
* **settingsID**->, settingType, emailsetting,phonesetting,facebooksetting
* **categoryID**->categoryName,, OrderDetailsID
* **cartID**->studioName, location, NumberOfPictures,clientid
* **retailersID**->cameraName, workType,editorid
* **orderDetailsID**-> ordersID, productid,qty,retailersRoles
* **RetailbranchManagerID**->BranchID, RetailID
* **ordersID**->customerID, branchID, ordernumber, totalPrice, paymentMethod, deliveryMethod, orderdate, date\_of\_delivery

**Sample Data Insertion.**

The following SQL statements showcases the insertion of data set into the relations to store information of film makers and photographer in their respective table.

***Inserting data into the retailers Table***

insert into retailers (retailerID, customerID, UserName, passwordLogin, retailerRoles) values (1, 1, 'Rhodie Swetmore', '7DoIrDR6L', 'Engineer');

insert into retailers (retailerID, customerID, UserName, passwordLogin, retailerRoles) values (2, 2, 'Jobie Gillow', 'siDbBoQ3', 'Construction Worker');

insert into retailers (retailerID, customerID, UserName, passwordLogin, retailerRoles) values (3, 3, 'Dominic Slatten', '0NHuKLlxffl', 'Construction Expeditor');

insert into retailers (retailerID, customerID, UserName, passwordLogin, retailerRoles) values (4, 4, 'Conrado Palk', 'AIFOYOkKO', 'Estimator');

insert into retailers (retailerID, customerID, UserName, passwordLogin, retailerRoles) values (5, 5, 'Ebeneser Phear', '5eTxeSWl', 'Supervisor');

***Inserting sample Data into Cart table***

insert into cart (cartID, categoryID, customerID, productID, quantity) values (1, 1, 1, 1, 646);

insert into cart (cartID, categoryID, customerID, productID, quantity) values (2, 2, 2, 2, 847);

insert into cart (cartID, categoryID, customerID, productID, quantity) values (3, 3, 3, 3, 213);

insert into cart (cartID, categoryID, customerID, productID, quantity) values (4, 4, 4, 4, 830);

***Inserting Data into orders Table***

insert into orders (orderID, customerID, ordernumber, totalprice, paymentmethod, deliverymethod, orderdate, branchID, date\_of\_delivery, status, noifications) values (1, 1, 1, 396, 'jcb', 'Electric Utilities: Central', '5/24/2022', 1, '2022:10:27', 'active', '-1/2');

insert into orders (orderID, customerID, ordernumber, totalprice, paymentmethod, deliverymethod, orderdate, branchID, date\_of\_delivery, status, noifications) values (2, 2, 2, 273, 'diners-club-enroute', 'Air Freight/Delivery Services', '11/29/2021', 2, '2022:5:11', 'inactive', 'get');

insert into orders (orderID, customerID, ordernumber, totalprice, paymentmethod, deliverymethod, orderdate, branchID, date\_of\_delivery, status, noifications) values (3, 3, 3, 114, 'jcb', 'Oil & Gas Production', '6/10/2022', 3, '2022:8:16', 'active', '00˙Ɩ$-');

insert into orders (orderID, customerID, ordernumber, totalprice, paymentmethod, deliverymethod, orderdate, branchID, date\_of\_delivery, status, noifications) values (4, 4, 4, 422, 'visa-electron', 'Biotechnology', '11/18/2022', 4, '2022:6:29', 'Tennessee', 'shipped');

insert into orders (orderID, customerID, ordernumber, totalprice, paymentmethod, deliverymethod, orderdate, branchID, date\_of\_delivery, status, noifications) values (5, 5, 5, 141, 'maestro', 'n/a', '8/25/2022', 5, '2022:1:19', 'active', '-1E2');

***Inserting data into product Table***

insert into product (productID, categoryID, productName, description, productprice, image, status) values (1, 1, 1, 'Auto Parts:O.E.M.', 58, 'http://dummyimage.com/222x100.png', 'SMP');

insert into product (productID, categoryID, productName, description, productprice, image, status) values (2, 2, 2, 'n/a', 163, 'http://dummyimage.com/203x100.png', 'NEE^I');

insert into product (productID, categoryID, productName, description, productprice, image, status) values (3, 3, 3, 'Finance: Consumer Services', 524, 'http://dummyimage.com/237x100.png', 'ELVT');

insert into product (productID, categoryID, productName, description, productprice, image, status) values (4, 4, 4, 'Property-Casualty Insurers', 550, 'http://dummyimage.com/149x100.png', 'L');

insert into product (productID, categoryID, productName, description, productprice, image, status) values (5, 5, 5, 'Life Insurance', 565, 'http://dummyimage.com/242x100.png', 'AEK');

***Inserting Data into customer Table***

insert into Customer (customerID, customerName, customerEmail, customerContact, customerAddress, billings, purchaseHistory) values (1, 'Saxe Birkmyre', 'sbirkmyre0@shareasale.com', '118-473-7446', '71 Shasta Parkway', '', '');

insert into Customer (customerID, customerName, customerEmail, customerContact, customerAddress, billings, purchaseHistory) values (2, 'Lynsey MacLaren', 'lmaclaren1@surveymonkey.com', '743-154-5659', '0029 Norway Maple Lane', '', '');

insert into Customer (customerID, customerName, customerEmail, customerContact, customerAddress, billings, purchaseHistory) values (3, 'Eryn Manion', 'emanion2@wikia.com', '579-861-3803', '85 Artisan Road', '', '');

insert into Customer (customerID, customerName, customerEmail, customerContact, customerAddress, billings, purchaseHistory) values (4, 'Rebeca Weldrake', 'rweldrake3@boston.com', '753-213-5621', '7449 Sunfield Alley', '', '');

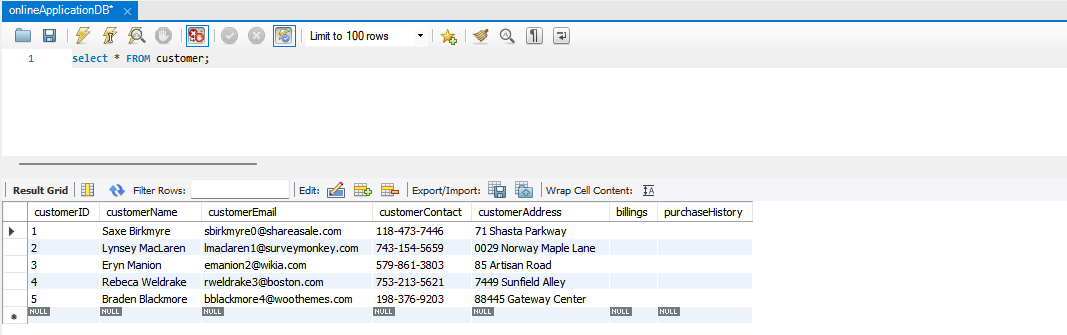
insert into Customer (customerID, customerName, customerEmail, customerContact, customerAddress, billings, purchaseHistory) values (5, 'Braden Blackmore', 'bblackmore4@woothemes.com', '198-376-9203', '88445 Gateway Center', '', '');

# **5.Reports, queries and sample output**

1. **Sample Queries to select all from customer Table**

select \* FROM customer;

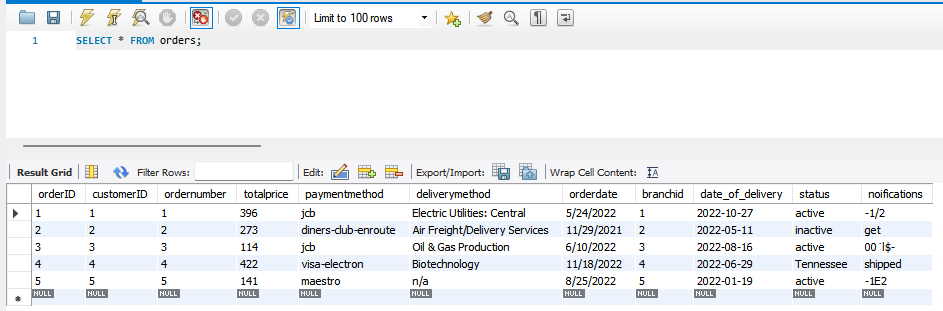
output

****

1. **Sample Queries to select all orders stored in the orders Table**

SELECT \* FROM orders;

Output

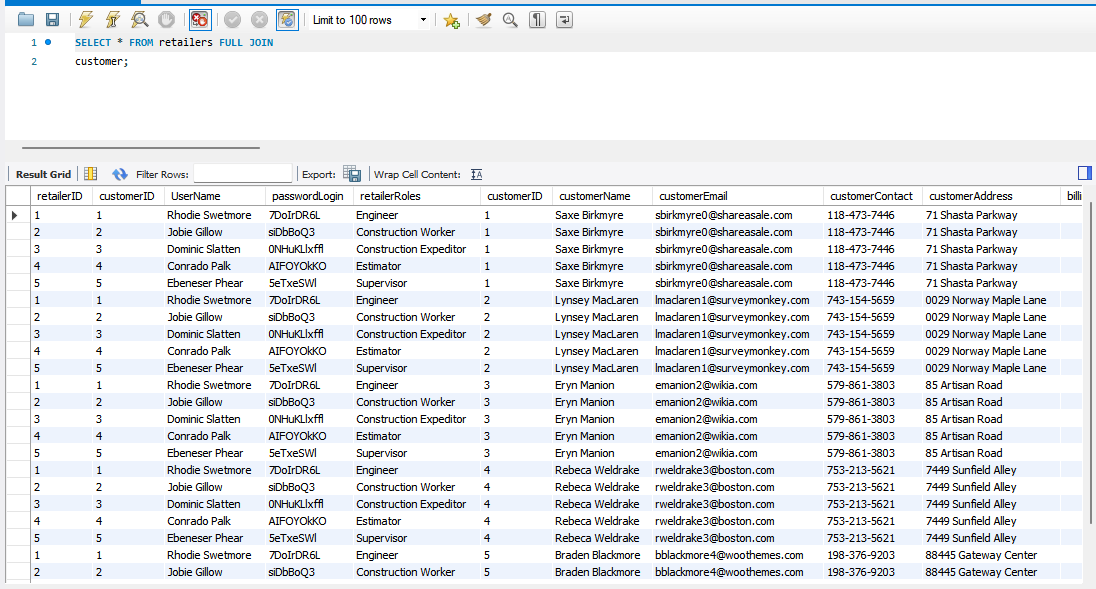
****

1. **Sample Queries to select and combine information from customers’ and retailers’ Table**

SELECT \* FROM retailers FULL JOIN

customer;

Output

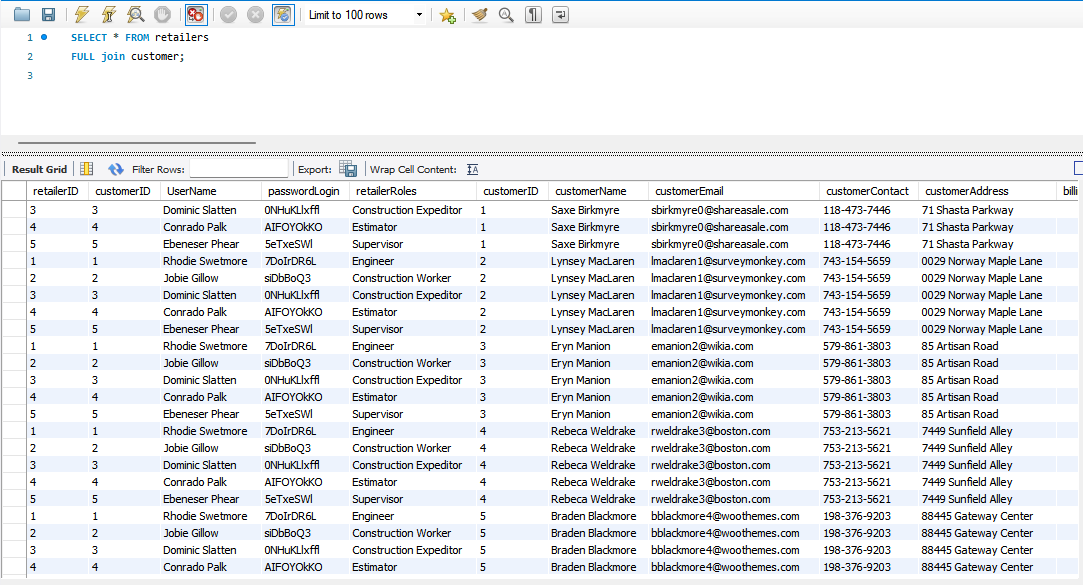
****

1. **Query to Select Online Customer and Retailers Information**

SELECT \* FROM retailers

FULL join customer;

output

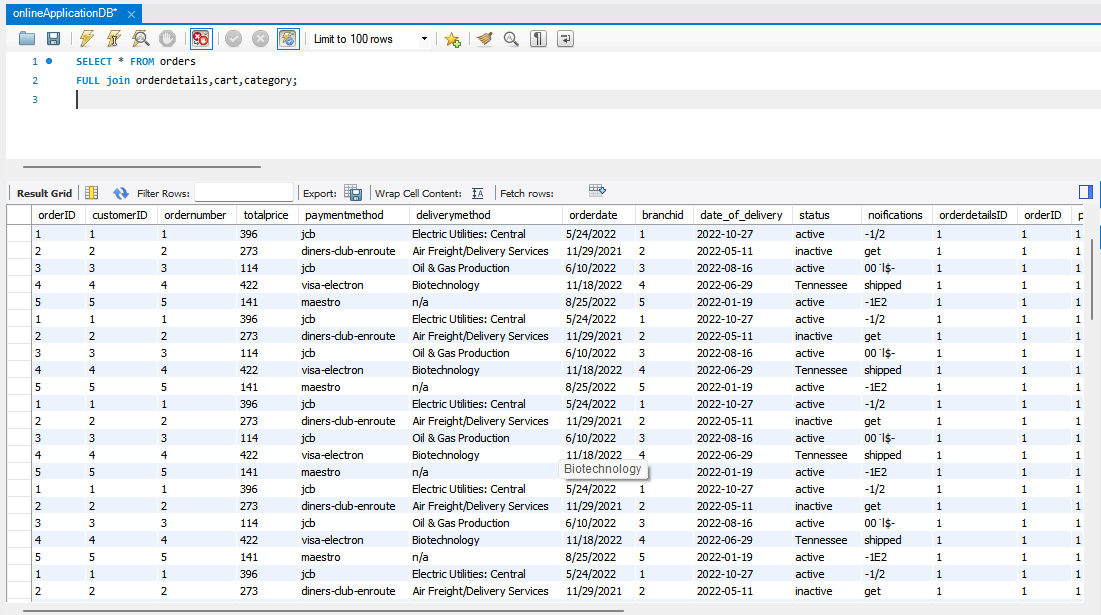
****

1. **Query to select Orders, Order details, category and cart summary data Information**

SELECT \* FROM orders

FULL join orderdetails, cart, category;

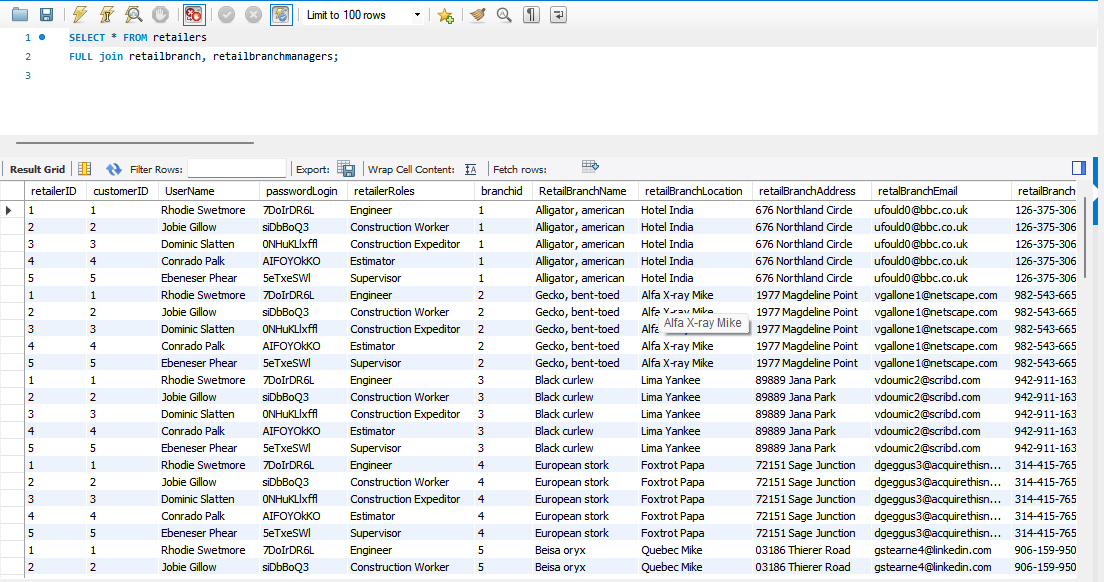
output



1. **Retailers, retailbranch and RetailBranchMangers Information**

SELECT \* FROM retailers

FULL join retailbranch, retailbranchmanagers;



# **Conclusion**

Online Retail Application Database is implemented to completion using the set of data modelling Technologies. The employed tools are MySQL Workbench server. Other environments are, Visual Studio and notepad text Editor. Entity relational Diagram and implementation of SQL codes, was done in MySQL workbench. Therefore, with the afore-developed Database technology, retailers in the field of Economics will be in a position to integrate this Relational Database system with their respective online Retail Application Software-Suite that will meet their needs.

# **References**

1. Kvet, M., Stasko, J., Wang, Y. L., Lima, A., & Gavrilović, T. (2022, April). Online Retailing and Shopping: An Academical Simulation to Databases. In *2022 31st Conference of Open Innovations Association (FRUCT)* (pp. 153-159). IEEE.
2. Coronel, C., & Morris, S. (2016). *Database systems: design, implementation, & management*. Cengage Learning.
3. https://blog.saleslayer.com/how-to-create-and-optimize-a-product-database-for-your-store