**DATABASE DESIGN PROJECT**

**ONLINE RETAIL APPLICATION DATABASE**

**BY**

**Student Name**

**Table of Contents**

[**Executive Summary** 3](#_Toc103185569)

[**Overview of the Database Project.** 4](#_Toc103185570)

[**1. Business Explanation** 4](#_Toc103185571)

[**2.Objectives** 5](#_Toc103185572)

[**3.High level design description** 6](#_Toc103185573)

[**4.Physical and logical model of the Database.** 7](#_Toc103185574)

[**4.1 Tables to be implemented with their respective Usage.** 7](#_Toc103185575)

[**4.2 Entity-Relationship Diagram** 13](#_Toc103185576)

[**Create statements of the respective tables to be implemented** 14](#_Toc103185577)

[**Functional dependencies of the above Relations** 18](#_Toc103185578)

[**SAMPLE DATA** 19](#_Toc103185579)

[**5.View definitions and sample output** 21](#_Toc103185580)

[**6.Reports, queries and sample output** 23](#_Toc103185581)

[**6.** **Conclusion** 24](#_Toc103185582)

# **Executive Summary**

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.

# **Overview of the Database Project.**

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 afore-Mentioned information anticipated to be stored in the Database opts to be of relevance for retailing potentials.

# **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. CustomerBranches, managers table, settings, purchasedOrder, 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 an item. The Customer will provide bank account number and bank name (can have Multiple account number). After Registration, each customer will have a unique customerid, userid, and password.A Customer can purchase one or more item in different quantities. The 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 bank account is required to settle the bill. The items can be ordered to one or more suppliers.

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.

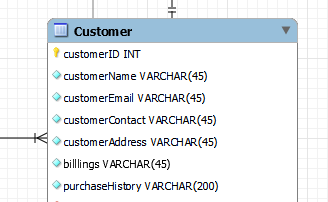
# **4. SQL Tables, functionalities, ER diagram and The Data Descriptions.**

## **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.

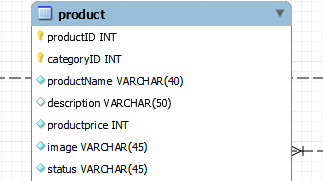
1. **Customer relation**

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.



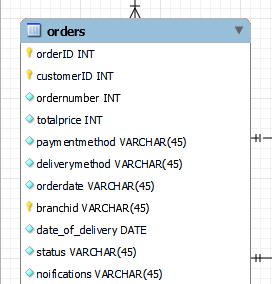
1. **Product relation**

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.

****

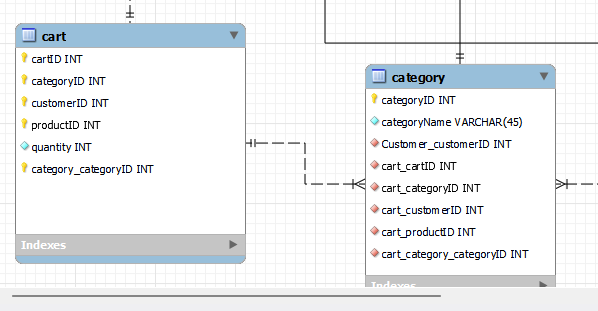
1. **Orders relation**

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



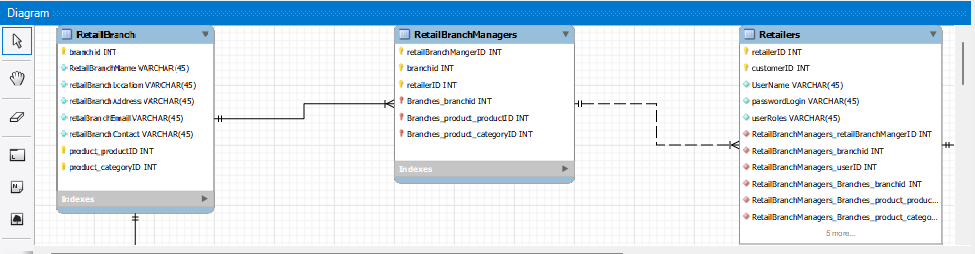
1. **Cart and Category relation tables**

Cart table will be used for storing relational information from the top three Online Retail schemas of editors in elation to the client and the photograph taken. The schemas below clearly depicts the attributes of both carts’ and Products’ entity.



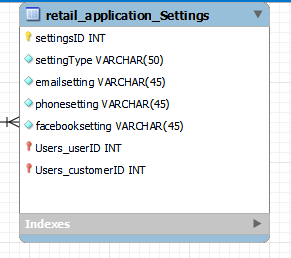
1. **RetailBranch, RetailBranchManagers and Retailers’ relations**

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.



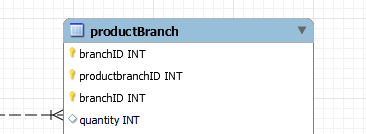
1. **Retail application Settings Table**

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: -



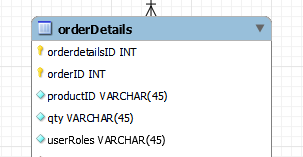
1. **ProductBranch relation**

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.

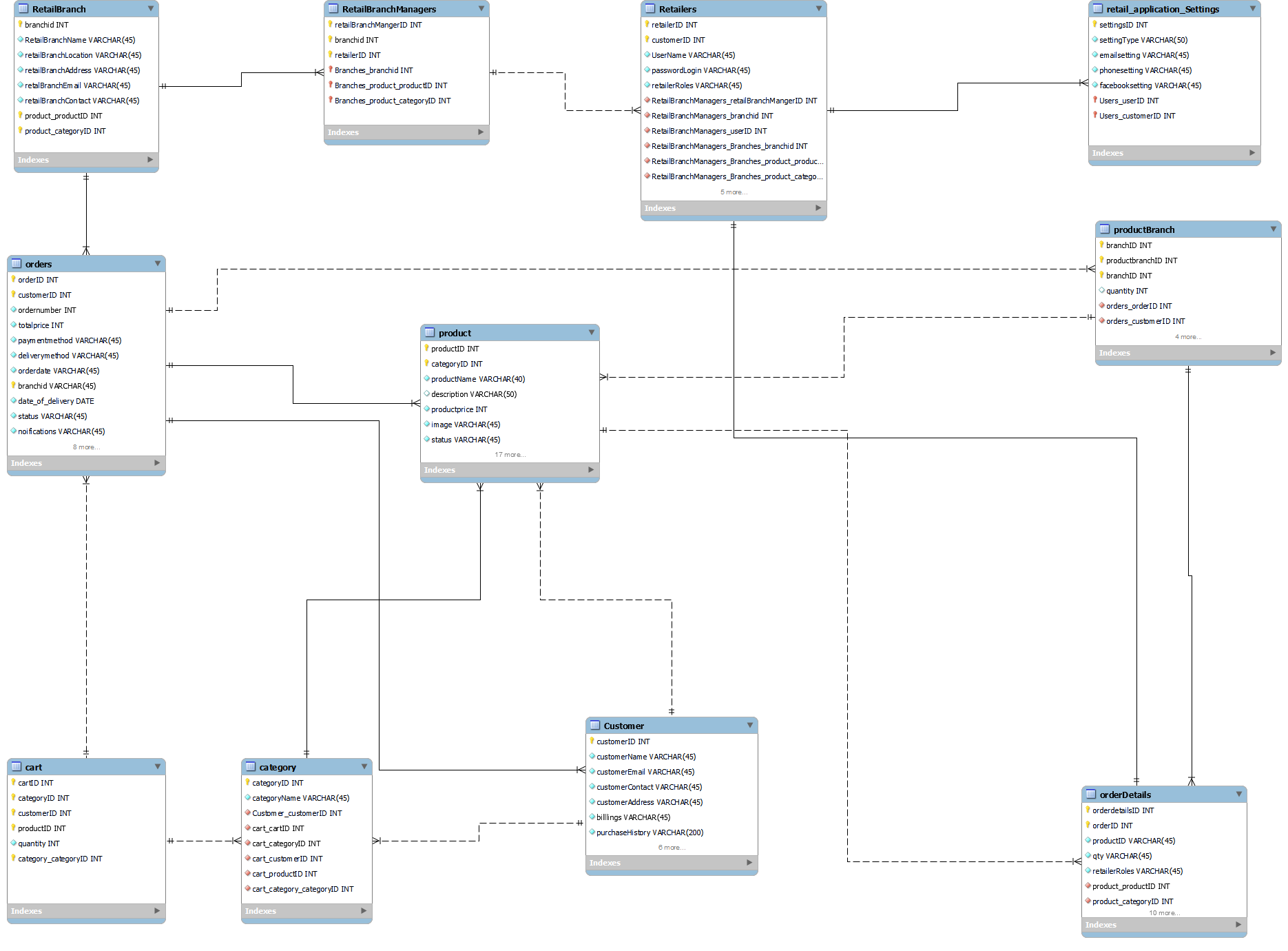


1. **OrderDetails relation**

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**

**Create statement for table client**

*CREATE TABLE IF NOT EXISTS `client` (*

*`clientid` INT NOT NULL,*

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

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

*`NumberOfPictures` INT(20) NOT NULL,*

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

*`photographerID` INT NOT NULL,*

*PRIMARY KEY (`clientid`, `photographerID`))*

*ENGINE = InnoDB*

**Create statement for table photo**

*CREATE TABLE IF NOT EXISTS `mydb`.`photo` (*

*`photoid` INT NOT NULL,*

*`clientid` INT NOT NULL,*

*`photographerID` INT NOT NULL,*

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

*`studioID` INT NOT NULL,*

*PRIMARY KEY (`photoid`, `studioID`, `clientid`),*

*CONSTRAINT `clientID`*

*FOREIGN KEY (`photoid`)*

*REFERENCES `mydb`.`client` (`clientid`)*

*ON DELETE NO ACTION*

*ON UPDATE NO ACTION,*

*CONSTRAINT `studioID`*

*FOREIGN KEY (`photoid`)*

*REFERENCES `mydb`.`Studio` (`studioID`)*

*ON DELETE NO ACTION*

*ON UPDATE NO ACTION)*

*ENGINE = InnoDB*

**Create statement for table studio**

*CREATE TABLE IF NOT EXISTS `mydb`.`Studio` (*

*`studioID` INT NOT NULL,*

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

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

*`NumberOfPictures` INT(20) NOT NULL,*

*`clientID` INT NOT NULL,*

*PRIMARY KEY (`studioID`, `clientID`))*

*ENGINE = InnoDB*

**Create statement for table photographer**

*CREATE TABLE IF NOT EXISTS `mydb`.`photographer` (*

*`photographerid` INT NOT NULL,*

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

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

*`NumberOfPictures` INT(20) NOT NULL,*

*`studiorID` INT NOT NULL,*

*PRIMARY KEY (`photographerid`, `studiorID`),*

*INDEX `studiorID\_idx` (`studiorID` ASC) VISIBLE,*

*CONSTRAINT `studiorID`*

*FOREIGN KEY (`studiorID`)*

*REFERENCES `mydb`.`Studio` (`studioID`)*

*ON DELETE NO ACTION*

*ON UPDATE NO ACTION)*

*ENGINE = InnoDB*

**Create statement for table cameraEquipment**

*CREATE TABLE IF NOT EXISTS `cameraEquipment` (*

*`cameraid` int(11) NOT NULL AUTO\_INCREMENT,*

*`cameraName` varchar(50) NOT NULL,*

*`workType` varchar(50) NOT NULL,*

*`editorid` INT NOT NULL,*

*PRIMARY KEY (`cameraid`),*

*FOREIGN KEY (editorid) REFERENCES editor(editorid)*

*);*

**Create statement for table editor**

*CREATE TABLE IF NOT EXISTS `mydb`.`editor` (*

*`editorid` INT NOT NULL,*

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

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

*`NumberOfPictures` INT(20) NOT NULL,*

*`clientID` INT NOT NULL,*

*PRIMARY KEY (`editorid`, `clientID`),*

*INDEX `clientID\_idx` (`clientID` ASC) VISIBLE,*

*CONSTRAINT `clientID`*

*FOREIGN KEY (`clientID`)*

*REFERENCES `mydb`.`client` (`clientid`)*

*ON DELETE NO ACTION*

*ON UPDATE NO ACTION)*

*ENGINE = InnoDB*

**Create statement for table RoleType**

*CREATE TABLE IF NOT EXISTS `mydb`.`RoleTypes` (*

*`roleTypeID` INT NOT NULL,*

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

*PRIMARY KEY (`roleTypeID`))*

*ENGINE = InnoDB*

**Create statement for table filmProducer**

*CREATE TABLE IF NOT EXISTS `mydb`.`FilmProducer` (*

*`producerID` INT NOT NULL,*

*`filmTitleID` INT NOT NULL,*

*`StudioID` INT NOT NULL,*

*PRIMARY KEY (`producerID`, `StudioID`, `filmTitleID`))*

*ENGINE = InnoDB*

**Create statement for table filmActors**

*CREATE TABLE IF NOT EXISTS `mydb`.`filmActors` (*

*`actorsID` INT NOT NULL,*

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

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

*`Studio\_studioID` INT NOT NULL,*

*`Studio\_clientID` INT NOT NULL,*

*PRIMARY KEY (`actorsID`, `Studio\_studioID`, `Studio\_clientID`),*

*INDEX `fk\_filmActors\_Studio1\_idx` (`Studio\_studioID` ASC, `Studio\_clientID` ASC) VISIBLE,*

*CONSTRAINT `fk\_filmActors\_Studio1`*

*FOREIGN KEY (`Studio\_studioID` , `Studio\_clientID`)*

*REFERENCES `mydb`.`Studio` (`studioID` , `clientID`)*

*ON DELETE NO ACTION*

*ON UPDATE NO ACTION)*

*ENGINE = InnoDB*

**Create statement for table filmActorsRoles**

*CREATE TABLE IF NOT EXISTS `mydb`.`filmActorsRoles` (*

*`filmTitleID` INT NOT NULL,*

*`actorsID` INT NOT NULL,*

*`roleTypeID` INT NOT NULL,*

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

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

*PRIMARY KEY (`filmTitleID`, `actorsID`, `roleTypeID`),*

*CONSTRAINT `actorsID`*

*FOREIGN KEY (`filmTitleID` , `actorsID` , `roleTypeID`)*

*REFERENCES `mydb`.`filmActors` (`actorsID` , `actorsID` , `actorsID`)*

*ON DELETE NO ACTION*

*ON UPDATE NO ACTION)*

*ENGINE = InnoDB*

**Create statement for table filmGenre**

*CREATE TABLE IF NOT EXISTS `filmGenre` (*

*`filmGenreid` int(11) NOT NULL AUTO\_INCREMENT,*

*`filmGenre` varchar(45) NOT NULL,*

*`producerid` int(11) NOT NULL,*

*`certificateid` int(50) NOT NULL,*

*PRIMARY KEY (`filmGenreid`),*

*FOREIGN KEY (producerid) REFERENCES filmProducer(producerid),*

*FOREIGN KEY (certificateid) REFERENCES legalCertificates(certificateid)*

*);*

**Create statement for table legalCertificate**

*CREATE TABLE IF NOT EXISTS `legalCertificates` (*

*`certificateid` int(50) NOT NULL AUTO\_INCREMENT,*

*`certificateName` varchar(45) NOT NULL,*

*`photoid` INT NOT NULL,*

*`studioid` int(11) NOT NULL,*

*PRIMARY KEY (`certificateid`),*

*FOREIGN KEY (photoid) REFERENCES photo(photoid),*

*FOREIGN KEY (studioid) REFERENCES studio(studioid)*

*);*

**Create statement for table filmTitle**

*CREATE TABLE IF NOT EXISTS `filmTitle` (*

*`filmTitleid` INT NOT NULL AUTO\_INCREMENT,*

*`filmTitle` varchar(50) NOT NULL,*

*`filmStory` varchar(50) NOT NULL,*

*`filmReleasedDate` varchar(50) NOT NULL,*

*`filmDuration` varchar(50) NOT NULL,*

*`filmGenre` varchar(50) NOT NULL,*

*`filmCertificate` int(11) NOT NULL,*

*`filmAdditionalDetails` int(11) NOT NULL,*

*PRIMARY KEY (`filmTitleid`)*

*);*

### **FUNCTIONAL DEPENDENCIES OF THE ABOVE 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**->filmTitle, filmStory, filmReleasedDate, filmDuration, filmGenre, filmCertificate,filmAdditionalDetails
* **ProductID**->filmTitleid, studioid
* **branchID**->actorsFullname,actorsScript
* **settingsID**->clientfullname, address, NumberOfPictures,occassionType,photographyid
* **categoryID**->fullname, address, NumberOfPictures, Studioid
* **cartID**->studioName, location, NumberOfPictures,clientid
* **retailersID**->cameraName, workType,editorid
* **RetailbranchManagerID**->editorfullname,address, numbeOfPictures

**SAMPLE DATA**

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 CameraEquipment Table***

INSERT INTO `cameraequipment` (`cameraid`, `cameraName`, `workType`, `editorid`) VALUES

(1, 'Myworks', 'intermediate', 1),

(2, 'Rhynoodle', 'productivity', 2),

(3, 'Dynazzy', 'Right-sized', 3),

(4, 'Cogilith', 'object-oriented', 4),

(5, 'Layo', 'impactful', 5),

(6, 'Realpoint', 'Profit-focused', 6),

(7, 'Tagcat', 'functionalities', 7),

(8, 'Flashdog', 'secured line', 8),

(9, 'DabZ', 'complexity', 9),

(10, 'Yombu', 'human-resource', 10);

***Inserting sample Data into Client***

INSERT INTO `client` (`clientid`, `fullname`, `Address`, `NumberOfPictures`, `occassionType`, `photographerid`) VALUES

(1, 'Lennie Jaquiss', '96 Meadow Vale Trail', 36, '', 1),

(2, 'Raine Barensky', '265 Northview Terrace', 10, '', 2),

(3, 'Orv De Ambrosi', '563 Del Sol Plaza', 36, '', 3),

(4, 'Mathew Medgewick', '20 Hollow Ridge Trail', 87, '', 4),

(5, 'Karly Teulier', '114 Talmadge Parkway', 78, '', 5),

(6, 'Corliss Goudie', '6 Helena Avenue', 38, '', 6),

(7, 'Osbourn Sealeaf', '63 Springview Crossing', 11, '', 7),

(8, 'Kassie Dumbrell', '20 New Castle Lane', 26, '', 8),

(9, 'Terence Spurdon', '8764 Crownhardt Drive', 61, '', 9),

(10, 'Jefferson Scotson', '3 Jenna Lane', 26, '', 10);

***Inserting Data into Editor Table***

INSERT INTO `editor` (`editorid`, `fullname`, `Address`, `NumberOfPictures`, `clientid`) VALUES

(1, 'Monique', '64 Del Sol Plaza', 965, 1),

(2, 'Cristy', '2 Lindbergh Lane', 87, 2),

(3, 'Carlyle', '64554 Sycamore Street', 921, 3),

(4, 'Sibylla', '5522 Del Mar Trail', 708, 4),

(5, 'Romain', '9 Morning Circle', 759, 5),

(6, 'Breena', '91437 Hoard Drive', 681, 6),

(7, 'Benjy', '04015 Cherokee Pass', 318, 7),

(8, 'Selena', '005 Kinsman Alley', 843, 8),

(9, 'Shani', '062 Hoepker Hill', 213, 9),

(10, 'Willy', '7360 Novick Avenue', 639, 10);

***Inserting data into Editor Table***

INSERT INTO `filmactors` (`actorsid`, `studioid`, `actorsFullName`, `actorsScript`) VALUES

(1, 1, 'Quillan Bennoe', ''),

(2, 2, 'Zachary Westbrook', ''),

(3, 3, 'Ximenez Dauney', ''),

(4, 4, 'Selinda Ashall', ''),

(5, 5, 'Barbra Shakelade', ''),

(6, 6, 'Sam Hrihorovich', ''),

(7, 7, 'Millie Anderson', ''),

(8, 8, 'Zita Millichap', ''),

(9, 9, 'Ofelia Bostock', ''),

(10, 10, 'Isobel Unitt', '');

***Inserting Data into FilmActorsRoles***

INSERT INTO `filmactorsroles` (`filmTitleid`, `actorsid`, `roleTypesid`, `characterName`, `characterDescription`) VALUES

(1, 1, 1, 'Ricca Burrows', 'Function-based object-oriented structure'),

(2, 2, 2, 'Daffi Bellee', 'Automated local access'),

(3, 3, 3, 'Dayle Chrestien', 'Realigned transitional analyzer'),

(4, 4, 4, 'Corilla Hostan', 'Sharable 3rd generation moratorium'),

(5, 5, 5, 'Tera Macek', 'Pre-emptive client-driven protocol'),

(6, 6, 6, 'Flore Mathon', 'Fundamental 24 hour alliance'),

(7, 7, 7, 'Kathie Bernaciak', 'Integrated analyzing utilisation'),

(8, 8, 8, 'Jamison Woltering', 'Function-based motivating installation'),

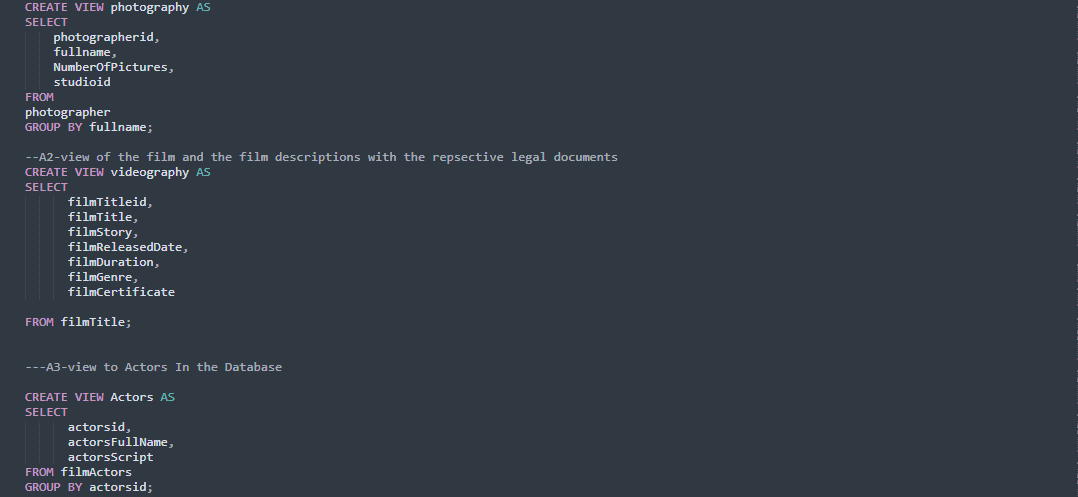
(9, 9, 9, 'Angeli Tregaskis', 'Public-key cohesive architecture'),

(10, 10, 10, 'Jarid Heathorn', 'Organized modular data-warehouse');

# **5.View definitions and sample output**

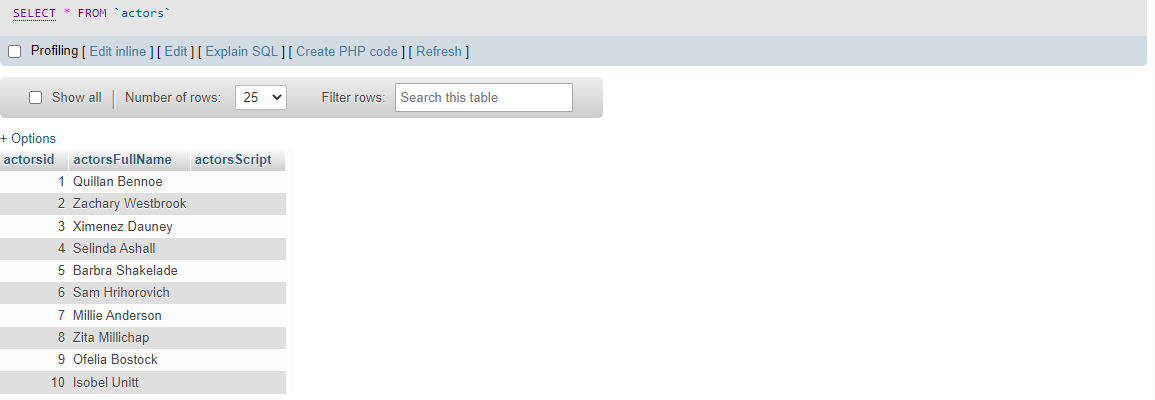
The following SQL codes were used to create and views from the relational Database.

***Screen capture of SQL codes***

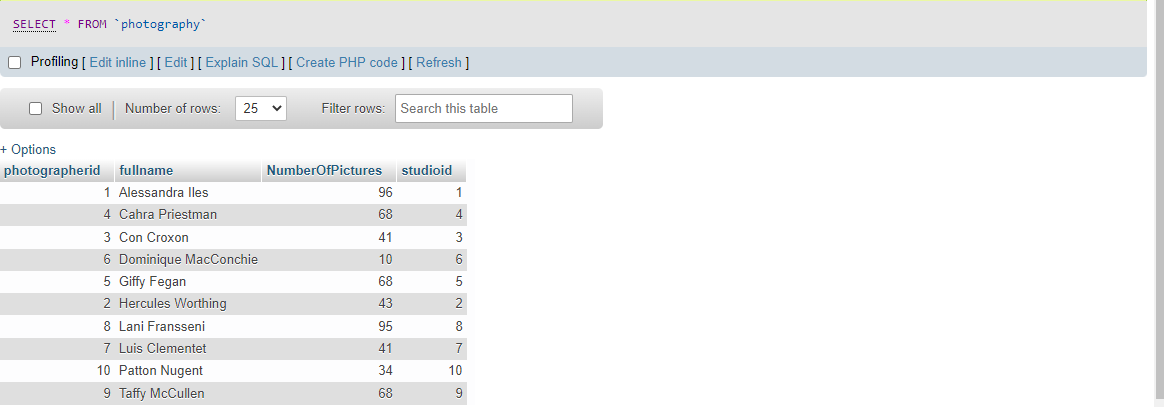


**Sample Output of The Above Created Views.**

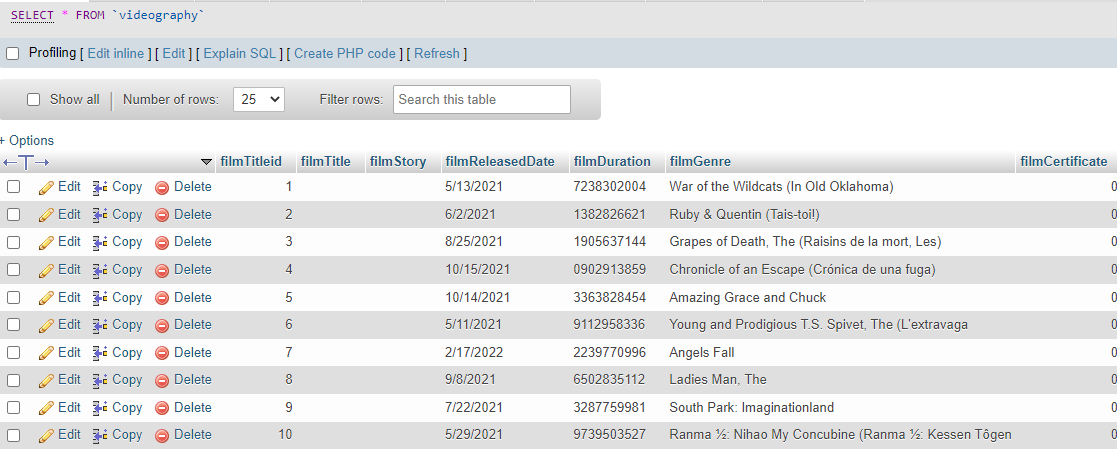
Sample output of Actors View

****

Sample output of Photography View

****

Sample Output of video Graphic View

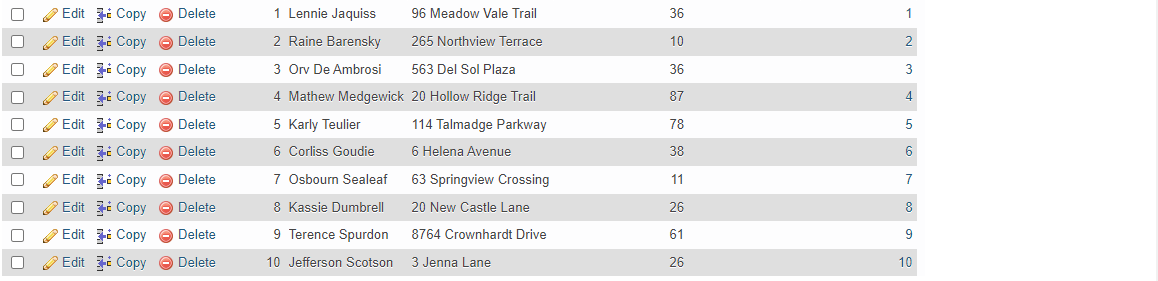
****

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

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

select \* FROM client;

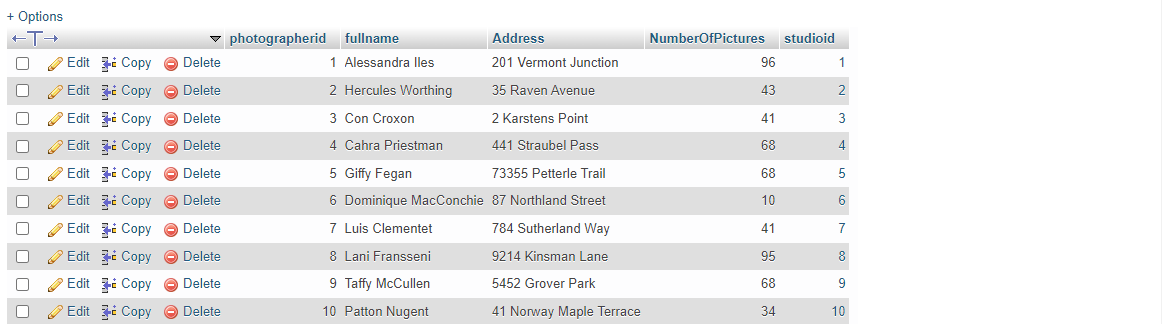
output

****

**2. Sample Queries to select all from photographer Table**

SELECT \* FROM photographer;

Output

****

**3. Sample Queries to select and combine information from filmproducers’ and editors’ Table**

SELECT filmsProduced, Studioid FROM filmproducer

UNION

SELECT fullname, NumberOfPictures FROM editor;

Output

****

# **Conclusion**

Advanced art Database management system is implemented to completion using the set of data modelling Technologies. The employed tools are MySQL Workbench, and Xampp MySQL server. Other environments are, Visual Studio. Enhanced Entity relational Diagram was created using MySQL workbench, whereas implementation of SQL codes was done using XAMPP server. Therefore, with the afore-developed Database system, Entertainment enterprises will be in a position to integrate this Relational Database system with their respective Software to Suite their needs