Advanced Databases Project

Design Document

1/29/2017

GameStop

2004154020041540

Michael Kelly, Christopher Hennessy

Contents

[Introduction 3](#_Toc475705318)

[Enhanced Entity Relationship Diagram 4](#_Toc475705319)

[Table Design/EER Description 5](#_Toc475705320)

[List of tables / relations created from EER. 5](#_Toc475705321)

[Table Specifications 8](#_Toc475705322)

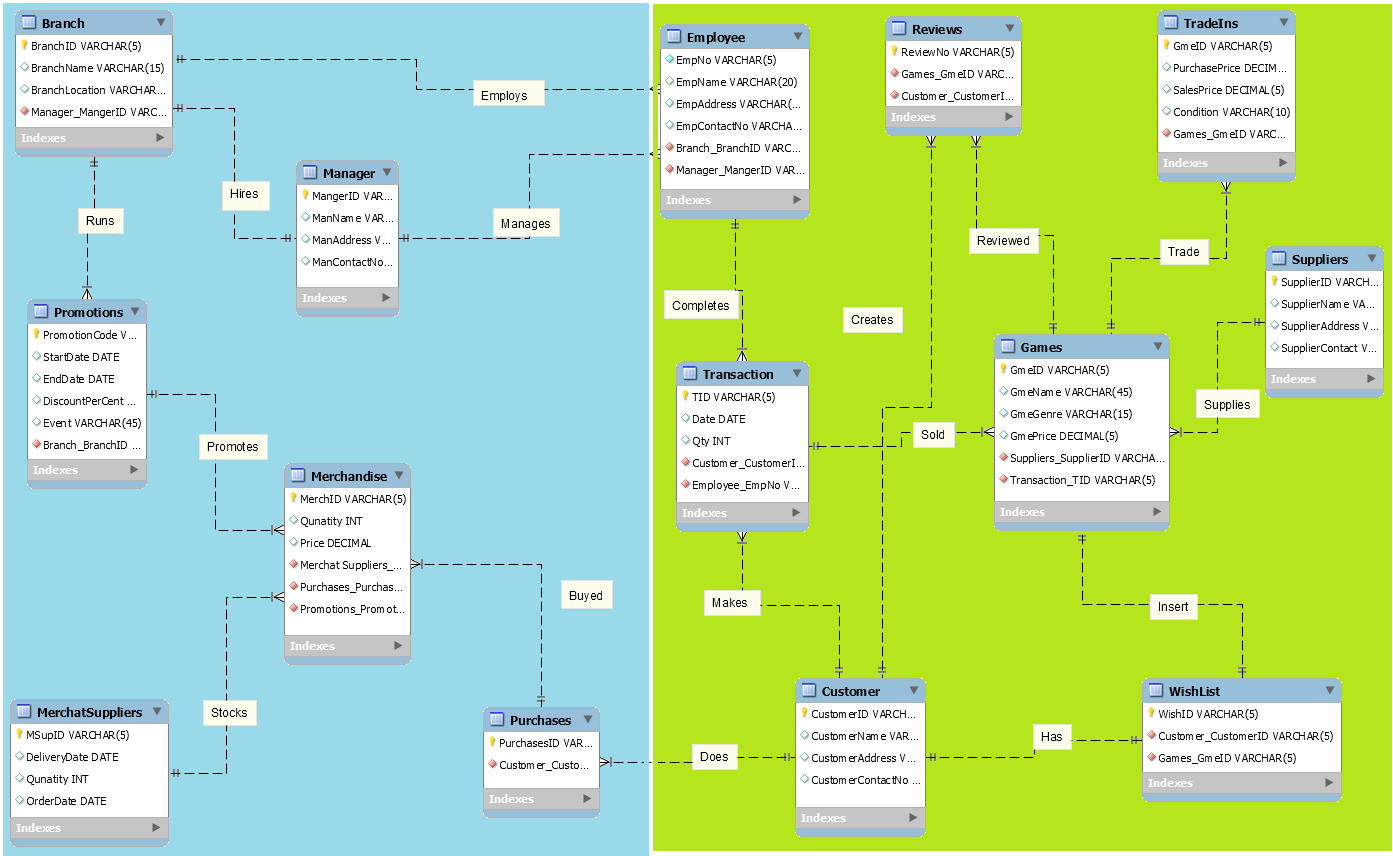
[Inserts 23](#_Toc475705323)

[Users 35](#_Toc475705324)

# Introduction

We have chosen GameStop as our business of choice to design our project about. We thought it would be a good choice as you can design a significantly large EER Model around the different factors that are associated with this business, this been the basis for the whole project. Within this project, we will include an introduction, as we are writing now. A brief description of GameStop and its benefits of implementing a database. In addition, how we attained the information for this business. Once the EER diagram is constructed, then the creation of the tables will take place such as stating the name, attributes and keys of each table that we have created. As we move forward in the project, we will be including Super/Sub type storage representation, Table design that will involve creating the relevant integrity constraints and key requirements for each table. Queries will be the next stage then, as we create ten queries, to be used in are project. Security will be introduced just before the end, to give different personal that are employed by the company access to the different aspects of the database. Then as the project concludes, I will give my opinion and feedback on the project.

# Enhanced Entity Relationship Diagram



# Table Design/EER Description

## List of tables / relations created from EER.

**Branch = { BranchID, BranchName, BranchLocation, Manger\_MangerID }**

Primary Key = BranchID

Foreign Key = Manger\_MangerID References Manger(MangerID)

**Promotions = {PromotionCode, StartDate, EndDate, DiscountPercent, Event, Branch\_BranchID**   **}**

Primary Key = PromotionCode

Foreign Key = Branch\_BranchID References Branch (BranchID)

**Merchandise = {MerchID, Quantity, Price,** **Promotions\_PromotionCode, MerchatSuppliers\_MSupID, Purchases\_PurchasesID }**

Primary Key = MerchID

Foreign Key = Promotions\_PromotionCode References Promotions (PromotionCode)

Foreign Key= MerchatSuppliers\_MSupID References MerchatSuppliers (MSupID)

Foreign Key = Purchases\_PurchasesID References Purchases (PurchasesID)

**MerchSuppliers = {MSupID, OrderDate, DeliveryDate, Quantity }**

Primary Key = MSupID

**Purchases = {PurchaseID, Customer\_CustomerID }**

Primary Key = PurchaseID

Foreign Key= Customer\_CustomerID References Customer (CustomerID)

**Manager = {MangerID, MangName, MangAddress, MangContactNo}**

Primary Key = MangerID

**Employee = {EmpNo, EmpName, EmpAddress, EmpContactNo, Branch\_BranchID, Manger\_MangerID** **}**

Primary Key = EmpNo

Foreign Key = Branch\_BranchID references Branch (BranchID)

Foreign Key = Manger\_MangerID references Manager (MangerID)

**Customer = {CustomerID, CustomerName, CustomerAddress, CustomerContactNo}**

Primary Key = CustomerID

**Transaction = {ID, Date, Qty, Employee\_EmpNo Customer\_CustomerID }**

Primary Key = TID

Foreign Key = Employee\_EmpNo references Employee (EmpNo)

Foreign Key = Customer\_CustomerID references Customer (CustomerID)

**Games = {GmeID, GmeName, GmeGenre, GmePrice, Suppliers\_SupplierID, Transaction\_TID)**

Primary Key = GmeID

Foreign Key = Suppliers\_SupplierID References Suppliers (SupplierID)

Foreign Key = Transaction\_TID Reference Transaction (TID)

**Suppliers = {SupplierID, SupplierName, SupplierAddress, SupplierContact}**

Primary Key= SupplierID

**Wishlist = {WishID,** **Customer\_CustomerID, Games\_GmeID }**

Primary Key = WishID

Foreign Key = Customer\_CustomerID references Customer (CustomerID)

Foreign Key = Games\_GmeID references Games (GmeID)

**Reviews = {ReviewNo, Customer\_CustomerID, Games\_GmeID }**

Primary Key = ReviewNo

Foreign Key = Customer\_CustomerID references Customer (CustomerID)

Foreign Key = Games\_GmeID references Games (GmeID)

**TradeIns = {GmeID, PurchasePrice, SalesPrice, Conditions}**

Primary Key = GmeID

Foreign Key = Games\_GmeID references Games (GmeID)

### Table Specifications

**Branch**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Size** | **Null/ not null** | **Default** | **Constraints** | **Index** | **Description** |
| BranchID | Varchar | 5 | Not null |  | unique | Primary key | This is the ID of each Branch. |
| BranchName | Varchar | 15 |  |  |  |  | A name for each branch of Gamestop |
| BranchLocation | Varchar | 45 |  |  |  |  | Where the shop is located address |
| Manager\_MangerID | Varchar | 5 | Not null |  | unique | Foreign Key | Unique code for Managers that work at different Stores. |

CREATE TABLE IF NOT EXISTS `GameStop`.`Branch` (

`BranchID` VARCHAR(5) NOT NULL,

`BranchName` VARCHAR(15) NULL,

`BranchLocation` VARCHAR(45) NULL,

`Manager\_MangerID` VARCHAR(5) NOT NULL,

PRIMARY KEY (`BranchID`),

UNIQUE INDEX `BranchID\_UNIQUE` (`BranchID` ASC),

INDEX `fk\_Branch\_Manager1\_idx` (`Manager\_MangerID` ASC),

CONSTRAINT `fk\_Branch\_Manager1`

FOREIGN KEY (`Manager\_MangerID`)

REFERENCES `GameStop`.`Manager` (`MangerID`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

**Promotions**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Size** | **Null/ not null** | **Default** | **Constraints** | **Index** | **Description** |
| PromotionCode | Varchar | 5 | Not null |  | unique | Primary key | This is a unique registered code for the Promotion. |
| Startdate | Date | 8 |  |  |  |  | When the offer starts |
| EndDate | Date | 8 |  |  |  |  | When the offer ends. |
| DiscountPercent | Decimal | 4 |  |  |  |  | How much will it be reduced by it will be a percentage. |
| Event | VarChar | 45 |  |  |  |  | What The event is. |
| Branch\_BranchID | VarChar | 5 | Not null |  | Unique | Foreign key | References Branch |

CREATE TABLE IF NOT EXISTS `GameStop`.`Promotions` (

`PromotionCode` VARCHAR(5) NOT NULL,

`StartDate` DATE NULL,

`EndDate` DATE NULL,

`DiscountPerCent` DECIMAL NULL,

`Event` VARCHAR(45) NULL,

`Branch\_BranchID` VARCHAR(5) NOT NULL,

PRIMARY KEY (`PromotionCode`),

UNIQUE INDEX `PromotionCode\_UNIQUE` (`PromotionCode` ASC),

INDEX `fk\_Promotions\_Branch1\_idx` (`Branch\_BranchID` ASC),

CONSTRAINT `fk\_Promotions\_Branch1`

FOREIGN KEY (`Branch\_BranchID`)

REFERENCES `GameStop`.`Branch` (`BranchID`)

ON DELETE RESTRICT

ON UPDATE RESTRICT)

ENGINE = InnoDB;

**Merchandise**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Size** | **Null/ not null** | **Default** | **Constraints** | **Index** | **Description** |
| MerchID | Varchar | 5 | Not null |  | unique | Primary key | This is a unique registered merchandise id number. |
| Quantity | Int | 5 |  |  |  |  | Number of Items. |
| Price | Decimal | 4 |  |  |  |  | How much the items will cost. |
| MerchantSuppliers\_MSupID | Varchar | 5 | Not null |  | unique | Foreign Key | Merchant Supplier id number references Merch Suppliers |
| Purchases\_PurchasesID | Varchar | 5 | Not null |  | Unique | Foreign Key | PurchasesID references Purchases. |
| Promotions\_PromotionCode | Varchar | 5 | Not null |  | unique | Foreign Key | PromotionCode References Promotions |

CREATE TABLE IF NOT EXISTS `GameStop`.`Merchandise` (

`MerchID` VARCHAR(5) NOT NULL,

`Qunatity` INT NULL,

`Price` DECIMAL NULL,

`Merchat Suppliers\_MSupID` VARCHAR(5) NOT NULL,

`Purchases\_PurchasesID` VARCHAR(5) NOT NULL,

`Promotions\_PromotionCode` VARCHAR(5) NOT NULL,

PRIMARY KEY (`MerchID`),

UNIQUE INDEX `MerchID\_UNIQUE` (`MerchID` ASC),

INDEX `fk\_Merchandise\_Merchat Suppliers\_idx` (`Merchat Suppliers\_MSupID` ASC),

INDEX `fk\_Merchandise\_Purchases1\_idx` (`Purchases\_PurchasesID` ASC),

INDEX `fk\_Merchandise\_Promotions1\_idx` (`Promotions\_PromotionCode` ASC),

CONSTRAINT `fk\_Merchandise\_Merchat Suppliers`

FOREIGN KEY (`Merchat Suppliers\_MSupID`)

REFERENCES `GameStop`.`MerchatSuppliers` (`MSupID`)

ON DELETE NO ACTION

ON UPDATE CASCADE,

CONSTRAINT `fk\_Merchandise\_Purchases1`

FOREIGN KEY (`Purchases\_PurchasesID`)

REFERENCES `GameStop`.`Purchases` (`PurchasesID`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `fk\_Merchandise\_Promotions1`

FOREIGN KEY (`Promotions\_PromotionCode`)

REFERENCES `GameStop`.`Promotions` (`PromotionCode`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

**Merchat Suppliers**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Size** | **Null/ not null** | **Default** | **Constraints** | **Index** | **Description** |
| MSupID | Varchar | 8 | Not null |  | unique | Primary key | Merchant Supplier id number |
| DeliveryDate | Date | 6 | Not null |  |  |  | Date Items Arrived |
| Quantity | Int | 5 |  |  |  |  | Number of Items. |
| OrderDate | Date | 8 |  |  |  |  | Date Stock Is Ordered. |

CREATE TABLE IF NOT EXISTS `GameStop`.`MerchatSuppliers` (

`MSupID` VARCHAR(5) NOT NULL,

`DeliveryDate` DATE NULL,

`Qunatity` INT NULL,

`OrderDate` DATE NULL,

PRIMARY KEY (`MSupID`),

UNIQUE INDEX `MSupID\_UNIQUE` (`MSupID` ASC))

ENGINE = InnoDB;

**Purchases**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Size** | **Null/ not null** | **Default** | **Constraints** | **Index** | **Description** |
| PurchaseID | Varchar | 5 | Not null |  | unique | Primary key | This is a unique number For Purchases |
| Qty | Int |  |  |  |  |  | Number Of Items Purchased. |
| Customer\_CustomerID | VarChar | 5 | Not null |  | Unique | Foreign key | CustomerID references Customer |

CREATE TABLE IF NOT EXISTS `GameStop`.`Purchases` (

`PurchasesID` VARCHAR(5) NOT NULL,

`Customer\_CustomerID` VARCHAR(5) NOT NULL,

PRIMARY KEY (`PurchasesID`),

UNIQUE INDEX `PurchasesID\_UNIQUE` (`PurchasesID` ASC),

INDEX `fk\_Purchases\_Customer1\_idx` (`Customer\_CustomerID` ASC),

CONSTRAINT `fk\_Purchases\_Customer1`

FOREIGN KEY (`Customer\_CustomerID`)

REFERENCES `GameStop`.`Customer` (`CustomerID`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

***Manager***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Size** | **Null/ not null** | **Default** | **Constraints** | **Index** | **Description** |
| MangerID | Varchar | 5 | Not null |  | unique | Primary key | This is a unique registered code for the Manager. |
| ManName | Varchar | 20 |  |  |  |  | Name Of Manager |
| ManAddress | Varchar | 45 |  |  |  |  | Managers Address. |
| ManContactNo | Varchar | 15 |  |  |  |  | Managers Contact Number. |

CREATE TABLE IF NOT EXISTS `GameStop`.`Manager` (

`MangerID` VARCHAR(5) NOT NULL,

`ManName` VARCHAR(20) NULL,

`ManAddress` VARCHAR(45) NULL,

`ManContactNo` VARCHAR(15) NULL,

PRIMARY KEY (`MangerID`),

UNIQUE INDEX `MangerID\_UNIQUE` (`MangerID` ASC))

ENGINE = InnoDB;

***Employee***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Size** | **Null/ not null** | **Default** | **Constraints** | **Index** | **Description** |
| EmpNo | Varchar | 5 | Not null |  | unique | Primary key | This is a unique registered code for the Employee. |
| EmpName | Varchar | 20 |  |  |  |  | Name Of employee |
| EmpAddress | Varchar | 45 |  |  |  |  | Employee Address. |
| EmpContactNo | Varchar | 15 | Not null |  |  | Foreign Key | Employee Contact Number. |
| Branch\_BranchID | Varchar | 5 | Not null |  |  | Foreign key | References Branch |
| Manager\_ManagerID | Varchar | 5 | Not null |  |  | Foreign key | References Manager |

CREATE TABLE IF NOT EXISTS `GameStop`.`Employee` (

`EmpNo` VARCHAR(5) NOT NULL,

`EmpName` VARCHAR(20) NULL,

`EmpAddress` VARCHAR(45) NULL,

`EmpContactNo` VARCHAR(15) NULL,

`Branch\_BranchID` VARCHAR(5) NOT NULL,

`Manager\_MangerID` VARCHAR(5) NOT NULL,

UNIQUE INDEX `EmpNo\_UNIQUE` (`EmpNo` ASC),

INDEX `fk\_Employee\_Branch1\_idx` (`Branch\_BranchID` ASC),

INDEX `fk\_Employee\_Manager1\_idx` (`Manager\_MangerID` ASC),

CONSTRAINT `fk\_Employee\_Branch1`

FOREIGN KEY (`Branch\_BranchID`)

REFERENCES `GameStop`.`Branch` (`BranchID`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `fk\_Employee\_Manager1`

FOREIGN KEY (`Manager\_MangerID`)

REFERENCES `GameStop`.`Manager` (`MangerID`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

***Customer***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Size** | **Null/ not null** | **Default** | **Constraints** | **Index** | **Description** |
| CustomerID | Varchar | 5 | Not null |  | unique | Primary key | This is a unique registered code for the Customer. |
| CustomerName | Varchar | 25 |  |  |  |  | Name Of Customer |
| CustomerAddress | Varchar | 45 |  |  |  |  | Customer Address. |
| CustomerContactNo | Varchar | 15 |  |  |  |  | Customer Contact Number. |

CREATE TABLE IF NOT EXISTS `GameStop`.`Customer` (

`CustomerID` VARCHAR(5) NOT NULL,

`CustomerName` VARCHAR(25) NULL,

`CustomerAddress` VARCHAR(45) NULL,

`CustomerContactNo` VARCHAR(15) NULL,

PRIMARY KEY (`CustomerID`),

UNIQUE INDEX `CustomerID\_UNIQUE` (`CustomerID` ASC))

ENGINE = InnoDB;

***Transaction***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Size** | **Null/ not null** | **Default** | **Constraints** | **Index** | **Description** |
| TID | Varchar | 5 | Not null |  | unique | Primary key | This is a unique registered code for the Transaction. |
| Date | Date | 8 |  |  |  |  | Date Of Transaction |
| Qty | Int |  |  |  |  |  | Number Of Items Sold |
| Customer\_CustomerID | Varchar | 5 | Not null |  |  | Foreign Key | References Customer |
| Employee\_EmpNo | Varchar | 5 | Not null |  |  | Foreign key | References Branch |

CREATE TABLE IF NOT EXISTS `GameStop`.`Transaction` (

`TID` VARCHAR(5) NOT NULL,

`Date` DATE NULL,

`Qty` INT NULL,

`Customer\_CustomerID` VARCHAR(5) NOT NULL,

`Employee\_EmpNo` VARCHAR(5) NOT NULL,

PRIMARY KEY (`TID`),

UNIQUE INDEX `TID\_UNIQUE` (`TID` ASC),

INDEX `fk\_Transaction\_Customer1\_idx` (`Customer\_CustomerID` ASC),

INDEX `fk\_Transaction\_Employee1\_idx` (`Employee\_EmpNo` ASC),

CONSTRAINT `fk\_Transaction\_Customer1`

FOREIGN KEY (`Customer\_CustomerID`)

REFERENCES `GameStop`.`Customer` (`CustomerID`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `fk\_Transaction\_Employee1`

FOREIGN KEY (`Employee\_EmpNo`)

REFERENCES `GameStop`.`Employee` (`EmpNo`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

***Games***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Size** | **Null/ not null** | **Default** | **Constraints** | **Index** | **Description** |
| GmeID | Varchar | 5 | Not null |  | unique | Primary key | This is a unique registered code for the game. |
| GmeName | Varchar | 45 |  |  |  |  | Name Of Game |
| GmeGenre | Varchar | 15 |  |  |  |  | Type Of Game |
| GmePrice | Decimal | 5 |  |  |  |  | Price Of Game. |
| Suppliers\_SupplierId | Varchar | 5 | Not null |  |  | Foreign key | References Suppliers |
| Transaction\_TID | Varchar | 5 | Not null |  |  | Foreign key | References Transaction. |

CREATE TABLE IF NOT EXISTS `GameStop`.`Games` (

`GmeID` VARCHAR(5) NOT NULL,

`GmeName` VARCHAR(45) NULL,

`GmeGenre` VARCHAR(15) NULL,

`GmePrice` DECIMAL(5) NULL,

`Suppliers\_SupplierID` VARCHAR(5) NOT NULL,

`Transaction\_TID` VARCHAR(5) NOT NULL,

PRIMARY KEY (`GmeID`),

UNIQUE INDEX `GmeID\_UNIQUE` (`GmeID` ASC),

INDEX `fk\_Games\_Suppliers1\_idx` (`Suppliers\_SupplierID` ASC),

INDEX `fk\_Games\_Transaction1\_idx` (`Transaction\_TID` ASC),

CONSTRAINT `fk\_Games\_Suppliers1`

FOREIGN KEY (`Suppliers\_SupplierID`)

REFERENCES `GameStop`.`Suppliers` (`SupplierID`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `fk\_Games\_Transaction1`

FOREIGN KEY (`Transaction\_TID`)

REFERENCES `GameStop`.`Transaction` (`TID`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

***Suppliers***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Size** | **Null/ not null** | **Default** | **Constraints** | **Index** | **Description** |
| SupplierId | Varchar | 5 | Not null |  | unique | Primary key | This is a unique registered code for the Supplier. |
| SupplierName | Varchar | 15 |  |  |  |  | Name Of Supplier |
| SupplierAddress | Varchar | 45 |  |  |  |  | Supplier Address. |
| SupplierContact | Varchar | 15 |  |  |  |  | Supplier Contact Number. |

CREATE TABLE IF NOT EXISTS `GameStop`.`Suppliers` (

`SupplierID` VARCHAR(5) NOT NULL,

`SupplierName` VARCHAR(15) NULL,

`SupplierAddress` VARCHAR(45) NULL,

`SupplierContact` VARCHAR(15) NULL,

PRIMARY KEY (`SupplierID`),

UNIQUE INDEX `SupplierID\_UNIQUE` (`SupplierID` ASC))

ENGINE = InnoDB;

***WishList***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Size** | **Null/ not null** | **Default** | **Constraints** | **Index** | **Description** |
| WishID | Varchar | 5 | Not null |  | unique | Primary key | This is a unique registered code for the WishList. |
| Customer\_CustomerID | Varchar | 5 | Not null |  |  | Foreign key | References Customer |
| Gmes\_GmeID | Varchar | 5 | Not null |  |  | Foreign key | References Games |

CREATE TABLE IF NOT EXISTS `GameStop`.`WishList` (

`WishID` VARCHAR(5) NOT NULL,

`Customer\_CustomerID` VARCHAR(5) NOT NULL,

`Games\_GmeID` VARCHAR(5) NOT NULL,

PRIMARY KEY (`WishID`),

UNIQUE INDEX `WishID\_UNIQUE` (`WishID` ASC),

INDEX `fk\_WishList\_Customer1\_idx` (`Customer\_CustomerID` ASC),

INDEX `fk\_WishList\_Games1\_idx` (`Games\_GmeID` ASC),

CONSTRAINT `fk\_WishList\_Customer1`

FOREIGN KEY (`Customer\_CustomerID`)

REFERENCES `GameStop`.`Customer` (`CustomerID`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `fk\_WishList\_Games1`

FOREIGN KEY (`Games\_GmeID`)

REFERENCES `GameStop`.`Games` (`GmeID`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

***Reviews***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Size** | **Null/ not null** | **Default** | **Constraints** | **Index** | **Description** |
| ReviewNo | Varchar | 5 | Not null |  | unique | Primary key | This is a unique registered code for the Reviews. |
| Customer\_CustomerID | Varchar | 5 | Not null |  |  | Foreign key | References Customer |
| Gmes\_GmeID | Varchar | 5 | Not null |  |  | Foreign key | References Games |

CREATE TABLE IF NOT EXISTS `GameStop`.`Reviews` (

`ReviewNo` VARCHAR(5) NOT NULL,

`Games\_GmeID` VARCHAR(5) NOT NULL,

`Customer\_CustomerID` VARCHAR(5) NOT NULL,

PRIMARY KEY (`ReviewNo`),

UNIQUE INDEX `ReviewNo\_UNIQUE` (`ReviewNo` ASC),

INDEX `fk\_Reviews\_Games1\_idx` (`Games\_GmeID` ASC),

INDEX `fk\_Reviews\_Customer1\_idx` (`Customer\_CustomerID` ASC),

CONSTRAINT `fk\_Reviews\_Games1`

FOREIGN KEY (`Games\_GmeID`)

REFERENCES `GameStop`.`Games` (`GmeID`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `fk\_Reviews\_Customer1`

FOREIGN KEY (`Customer\_CustomerID`)

REFERENCES `GameStop`.`Customer` (`CustomerID`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

***Trade-Ins***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Size** | **Null/ not null** | **Default** | **Constraints** | **Index** | **Description** |
| GmeID | Varchar | 5 | Not null |  | unique | Primary key | This is a unique registered code for the WishList. |
| PurchasePrice | Decimal | 5 |  |  |  |  | Price Purchased For From Customer. |
| SalesPrice | Decimal | 5 |  |  |  |  | Price Game To Be Sold At. |
| Condition | VarChar | 10 |  |  |  |  | Quality Of Game. |
| Gmes\_GmeID | Varchar | 5 | Not null |  |  | Foreign key | References Games |

CREATE TABLE IF NOT EXISTS `GameStop`.`TradeIns` (

`GmeID` VARCHAR(5) NOT NULL,

`PurchasePrice` DECIMAL(5) NULL,

`SalesPrice` DECIMAL(5) NULL,

`Conditions` VARCHAR(10) NULL,

`Games\_GmeID` VARCHAR(5) NOT NULL,

PRIMARY KEY (`GmeID`),

UNIQUE INDEX `GmeID\_UNIQUE` (`GmeID` ASC),

INDEX `fk\_Trade-ins\_Games1\_idx` (`Games\_GmeID` ASC),

CONSTRAINT `fk\_Trade-ins\_Games1`

FOREIGN KEY (`Games\_GmeID`)

REFERENCES `GameStop`.`Games` (`GmeID`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

### Inserts

-- -----------------------------------------------------

-- Data for table `GameStop`.`Manager`

-- -----------------------------------------------------

START TRANSACTION;

USE `GameStop`;

INSERT INTO `GameStop`.`Manager` (`MangerID`, `ManName`, `ManAddress`, `ManContactNo`) VALUES ('M001', 'Martin Ryan', 'Carlow', '4587878');

INSERT INTO `GameStop`.`Manager` (`MangerID`, `ManName`, `ManAddress`, `ManContactNo`) VALUES ('M002', 'Sean Kavanagh', 'Wexford', '4878874');

INSERT INTO `GameStop`.`Manager` (`MangerID`, `ManName`, `ManAddress`, `ManContactNo`) VALUES ('M003', 'Tom', 'Waterford', '7878845');

COMMIT;

-- -----------------------------------------------------

-- Data for table `GameStop`.`Branch`

-- -----------------------------------------------------

START TRANSACTION;

USE `GameStop`;

INSERT INTO `GameStop`.`Branch` (`BranchID`, `BranchName`, `BranchLocation`, `Manager\_MangerID`) VALUES ('B001', 'GameStop', 'Wexford', 'M001');

INSERT INTO `GameStop`.`Branch` (`BranchID`, `BranchName`, `BranchLocation`, `Manager\_MangerID`) VALUES ('B002', 'GameStop', 'Waterford', 'M002');

INSERT INTO `GameStop`.`Branch` (`BranchID`, `BranchName`, `BranchLocation`, `Manager\_MangerID`) VALUES ('B003', 'GameStop', 'Kilkenny', 'M003');

COMMIT;

-- -----------------------------------------------------

-- Data for table `GameStop`.`Promotions`

-- -----------------------------------------------------

START TRANSACTION;

USE `GameStop`;

INSERT INTO `GameStop`.`Promotions` (`PromotionCode`, `StartDate`, `EndDate`, `DiscountPerCent`, `Event`, `Branch\_BranchID`) VALUES ('PR001', '2017-03-22', '2017-03-29', 10, 'Crazy Bonanaza', 'B001');

INSERT INTO `GameStop`.`Promotions` (`PromotionCode`, `StartDate`, `EndDate`, `DiscountPerCent`, `Event`, `Branch\_BranchID`) VALUES ('PR002', '2017-03-30', '2017-04-06', 20, 'Wild Week', 'B002');

INSERT INTO `GameStop`.`Promotions` (`PromotionCode`, `StartDate`, `EndDate`, `DiscountPerCent`, `Event`, `Branch\_BranchID`) VALUES ('PR003', '2017-03-23', '2017-04-08', 15, '2 Week Fest', 'B003');

COMMIT;

-- -----------------------------------------------------

-- Data for table `GameStop`.`MerchatSuppliers`

-- -----------------------------------------------------

START TRANSACTION;

USE `GameStop`;

INSERT INTO `GameStop`.`MerchatSuppliers` (`MSupID`, `DeliveryDate`, `Qunatity`, `OrderDate`) VALUES ('MS001', '2017-03-19', 50, '2017-03-10');

INSERT INTO `GameStop`.`MerchatSuppliers` (`MSupID`, `DeliveryDate`, `Qunatity`, `OrderDate`) VALUES ('MS002', '2017-03-20', 100, '2017-03-11');

INSERT INTO `GameStop`.`MerchatSuppliers` (`MSupID`, `DeliveryDate`, `Qunatity`, `OrderDate`) VALUES ('MS003', '2017-03-21', 150, '2017-03-12');

INSERT INTO `GameStop`.`MerchatSuppliers` (`MSupID`, `DeliveryDate`, `Qunatity`, `OrderDate`) VALUES ('MS004', '2017-03-22', 60, '2017-03-13');

INSERT INTO `GameStop`.`MerchatSuppliers` (`MSupID`, `DeliveryDate`, `Qunatity`, `OrderDate`) VALUES ('MS005', '2017-03-23', 20, '2017-03-14');

COMMIT;

-- -----------------------------------------------------

-- Data for table `GameStop`.`Customer`

-- -----------------------------------------------------

START TRANSACTION;

USE `GameStop`;

INSERT INTO `GameStop`.`Customer` (`CustomerID`, `CustomerName`, `CustomerAddress`, `CustomerContactNo`) VALUES ('C001', 'Christopher Henesey', 'Kilkenny', '08712345');

INSERT INTO `GameStop`.`Customer` (`CustomerID`, `CustomerName`, `CustomerAddress`, `CustomerContactNo`) VALUES ('C002', 'Michael Kelly', 'Wexford', '08612356');

INSERT INTO `GameStop`.`Customer` (`CustomerID`, `CustomerName`, `CustomerAddress`, `CustomerContactNo`) VALUES ('C003', 'Luke Cullen', 'Galway', '02365478');

INSERT INTO `GameStop`.`Customer` (`CustomerID`, `CustomerName`, `CustomerAddress`, `CustomerContactNo`) VALUES ('C004', 'Peggy Ryan', 'Wexford', '03256984');

INSERT INTO `GameStop`.`Customer` (`CustomerID`, `CustomerName`, `CustomerAddress`, `CustomerContactNo`) VALUES ('C005', 'Mary Murphy', 'Waterford', '01425639');

INSERT INTO `GameStop`.`Customer` (`CustomerID`, `CustomerName`, `CustomerAddress`, `CustomerContactNo`) VALUES ('C006', 'Wayne Mitchell', 'Waterford', '02546895');

INSERT INTO `GameStop`.`Customer` (`CustomerID`, `CustomerName`, `CustomerAddress`, `CustomerContactNo`) VALUES ('C007', 'Donna Watson', 'Wexford', '03658755');

INSERT INTO `GameStop`.`Customer` (`CustomerID`, `CustomerName`, `CustomerAddress`, `CustomerContactNo`) VALUES ('C008', 'Sandra Martinez', 'Kilkenny', '21548554');

INSERT INTO `GameStop`.`Customer` (`CustomerID`, `CustomerName`, `CustomerAddress`, `CustomerContactNo`) VALUES ('C009', 'Jesse Bryant', 'Wexford', '48755896');

INSERT INTO `GameStop`.`Customer` (`CustomerID`, `CustomerName`, `CustomerAddress`, `CustomerContactNo`) VALUES ('C010', 'Patrick Russell', 'Dublin', '45788775');

COMMIT;

-- -----------------------------------------------------

-- Data for table `GameStop`.`Purchases`

-- -----------------------------------------------------

START TRANSACTION;

USE `GameStop`;

INSERT INTO `GameStop`.`Purchases` (`PurchasesID`, `Customer\_CustomerID`) VALUES ('P001', 'C001');

INSERT INTO `GameStop`.`Purchases` (`PurchasesID`, `Customer\_CustomerID`) VALUES ('P002', 'C002');

INSERT INTO `GameStop`.`Purchases` (`PurchasesID`, `Customer\_CustomerID`) VALUES ('P003', 'C004');

INSERT INTO `GameStop`.`Purchases` (`PurchasesID`, `Customer\_CustomerID`) VALUES ('P004', 'C003');

INSERT INTO `GameStop`.`Purchases` (`PurchasesID`, `Customer\_CustomerID`) VALUES ('P005', 'C005');

INSERT INTO `GameStop`.`Purchases` (`PurchasesID`, `Customer\_CustomerID`) VALUES ('P006', 'C006');

INSERT INTO `GameStop`.`Purchases` (`PurchasesID`, `Customer\_CustomerID`) VALUES ('P007', 'C007');

INSERT INTO `GameStop`.`Purchases` (`PurchasesID`, `Customer\_CustomerID`) VALUES ('P008', 'C008');

INSERT INTO `GameStop`.`Purchases` (`PurchasesID`, `Customer\_CustomerID`) VALUES ('P009', 'C009');

INSERT INTO `GameStop`.`Purchases` (`PurchasesID`, `Customer\_CustomerID`) VALUES ('P010', 'C010');

COMMIT;

-- -----------------------------------------------------

-- Data for table `GameStop`.`Merchandise`

-- -----------------------------------------------------

START TRANSACTION;

USE `GameStop`;

INSERT INTO `GameStop`.`Merchandise` (`MerchID`, `Qunatity`, `Price`, `Merchat Suppliers\_MSupID`, `Purchases\_PurchasesID`, `Promotions\_PromotionCode`) VALUES ('Mc001', 50, 19.99, 'MS001', 'P001', 'PR001');

INSERT INTO `GameStop`.`Merchandise` (`MerchID`, `Qunatity`, `Price`, `Merchat Suppliers\_MSupID`, `Purchases\_PurchasesID`, `Promotions\_PromotionCode`) VALUES ('Mc002', 100, 5.00, 'MS002', 'P002', 'PR001');

INSERT INTO `GameStop`.`Merchandise` (`MerchID`, `Qunatity`, `Price`, `Merchat Suppliers\_MSupID`, `Purchases\_PurchasesID`, `Promotions\_PromotionCode`) VALUES ('Mc003', 150, 9.99, 'MS003', 'P003', 'PR001');

INSERT INTO `GameStop`.`Merchandise` (`MerchID`, `Qunatity`, `Price`, `Merchat Suppliers\_MSupID`, `Purchases\_PurchasesID`, `Promotions\_PromotionCode`) VALUES ('Mc004', 60, 29.99, 'MS004', 'P004', 'PR002');

INSERT INTO `GameStop`.`Merchandise` (`MerchID`, `Qunatity`, `Price`, `Merchat Suppliers\_MSupID`, `Purchases\_PurchasesID`, `Promotions\_PromotionCode`) VALUES ('Mc005', 20, 159.99, 'MS005', 'P005', 'PR003');

COMMIT;

-- -----------------------------------------------------

-- Data for table `GameStop`.`Employee`

-- -----------------------------------------------------

START TRANSACTION;

USE `GameStop`;

INSERT INTO `GameStop`.`Employee` (`EmpNo`, `EmpName`, `EmpAddress`, `EmpContactNo`, `Branch\_BranchID`, `Manager\_MangerID`) VALUES ('E001', 'Nicole Lewis', 'Wexford', '124578', 'B001', 'M002');

INSERT INTO `GameStop`.`Employee` (`EmpNo`, `EmpName`, `EmpAddress`, `EmpContactNo`, `Branch\_BranchID`, `Manager\_MangerID`) VALUES ('E002', 'Maria Carter', 'Kilkenny', '145487', 'B003', 'M001');

INSERT INTO `GameStop`.`Employee` (`EmpNo`, `EmpName`, `EmpAddress`, `EmpContactNo`, `Branch\_BranchID`, `Manager\_MangerID`) VALUES ('E003', 'Peter Wright', 'Waterford', '887755', 'B002', 'M003');

INSERT INTO `GameStop`.`Employee` (`EmpNo`, `EmpName`, `EmpAddress`, `EmpContactNo`, `Branch\_BranchID`, `Manager\_MangerID`) VALUES ('E004', 'Larry Torres', 'Wexford', '745788', 'B001', 'M002');

INSERT INTO `GameStop`.`Employee` (`EmpNo`, `EmpName`, `EmpAddress`, `EmpContactNo`, `Branch\_BranchID`, `Manager\_MangerID`) VALUES ('E005', 'Paula Roberts', 'Kilkenny', '445566', 'B003', 'M001');

INSERT INTO `GameStop`.`Employee` (`EmpNo`, `EmpName`, `EmpAddress`, `EmpContactNo`, `Branch\_BranchID`, `Manager\_MangerID`) VALUES ('E006', 'Clarence James', 'Waterford', '663388', 'B002', 'M003');

INSERT INTO `GameStop`.`Employee` (`EmpNo`, `EmpName`, `EmpAddress`, `EmpContactNo`, `Branch\_BranchID`, `Manager\_MangerID`) VALUES ('E007', 'Diana Adams', 'Wexford', '444466', 'B001', 'M002');

INSERT INTO `GameStop`.`Employee` (`EmpNo`, `EmpName`, `EmpAddress`, `EmpContactNo`, `Branch\_BranchID`, `Manager\_MangerID`) VALUES ('E008', 'Richard Mitchell', 'Kilkenny', '665588', 'B003', 'M001');

INSERT INTO `GameStop`.`Employee` (`EmpNo`, `EmpName`, `EmpAddress`, `EmpContactNo`, `Branch\_BranchID`, `Manager\_MangerID`) VALUES ('E009', 'George Edwards', 'Waterford', '546987', 'B002', 'M003');

INSERT INTO `GameStop`.`Employee` (`EmpNo`, `EmpName`, `EmpAddress`, `EmpContactNo`, `Branch\_BranchID`, `Manager\_MangerID`) VALUES ('E010', 'Lawrence King', 'Wexford', '965478', 'B001', 'M002');

COMMIT;

-- -----------------------------------------------------

-- Data for table `GameStop`.`Transaction`

-- -----------------------------------------------------

START TRANSACTION;

USE `GameStop`;

INSERT INTO `GameStop`.`Transaction` (`TID`, `Date`, `Qty`, `Customer\_CustomerID`, `Employee\_EmpNo`) VALUES ('T001', '2017-03-23', 4, 'C006', 'E001');

INSERT INTO `GameStop`.`Transaction` (`TID`, `Date`, `Qty`, `Customer\_CustomerID`, `Employee\_EmpNo`) VALUES ('T002', '2017-04-24', 6, 'C007', 'E002');

INSERT INTO `GameStop`.`Transaction` (`TID`, `Date`, `Qty`, `Customer\_CustomerID`, `Employee\_EmpNo`) VALUES ('T003', '2017-03-25', 10, 'C008', 'E003');

INSERT INTO `GameStop`.`Transaction` (`TID`, `Date`, `Qty`, `Customer\_CustomerID`, `Employee\_EmpNo`) VALUES ('T004', '2017-03-26', 3, 'C010', 'E004');

INSERT INTO `GameStop`.`Transaction` (`TID`, `Date`, `Qty`, `Customer\_CustomerID`, `Employee\_EmpNo`) VALUES ('T005', '2017-03-27', 2, 'C001', 'E005');

INSERT INTO `GameStop`.`Transaction` (`TID`, `Date`, `Qty`, `Customer\_CustomerID`, `Employee\_EmpNo`) VALUES ('T006', '2017-03-28', 15, 'C003', 'E006');

INSERT INTO `GameStop`.`Transaction` (`TID`, `Date`, `Qty`, `Customer\_CustomerID`, `Employee\_EmpNo`) VALUES ('T007', '2017-03-29', 4, 'C002', 'E007');

INSERT INTO `GameStop`.`Transaction` (`TID`, `Date`, `Qty`, `Customer\_CustomerID`, `Employee\_EmpNo`) VALUES ('T008', '2017-03-30', 6, 'C005', 'E008');

INSERT INTO `GameStop`.`Transaction` (`TID`, `Date`, `Qty`, `Customer\_CustomerID`, `Employee\_EmpNo`) VALUES ('T009', '2017-03-31', 8, 'C004', 'E009');

INSERT INTO `GameStop`.`Transaction` (`TID`, `Date`, `Qty`, `Customer\_CustomerID`, `Employee\_EmpNo`) VALUES ('T010', '2017-04-01', 20, 'C009', 'E010');

INSERT INTO `GameStop`.`Transaction` (`TID`, `Date`, `Qty`, `Customer\_CustomerID`, `Employee\_EmpNo`) VALUES ('T011', '2017-04-02', 3, 'C003', 'E003');

INSERT INTO `GameStop`.`Transaction` (`TID`, `Date`, `Qty`, `Customer\_CustomerID`, `Employee\_EmpNo`) VALUES ('T012', '2017-04-03', 2, 'C006', 'E005');

INSERT INTO `GameStop`.`Transaction` (`TID`, `Date`, `Qty`, `Customer\_CustomerID`, `Employee\_EmpNo`) VALUES ('T013', '2017-04-04', 15, 'C005', 'E006');

COMMIT;

-- -----------------------------------------------------

-- Data for table `GameStop`.`Suppliers`

-- -----------------------------------------------------

START TRANSACTION;

USE `GameStop`;

INSERT INTO `GameStop`.`Suppliers` (`SupplierID`, `SupplierName`, `SupplierAddress`, `SupplierContact`) VALUES ('S001', 'Xbox', 'Dublin', '457854');

INSERT INTO `GameStop`.`Suppliers` (`SupplierID`, `SupplierName`, `SupplierAddress`, `SupplierContact`) VALUES ('S002', 'Playstation', 'Offaly', '863399');

INSERT INTO `GameStop`.`Suppliers` (`SupplierID`, `SupplierName`, `SupplierAddress`, `SupplierContact`) VALUES ('S003', 'Nintendo', 'Belfast', '444988');

INSERT INTO `GameStop`.`Suppliers` (`SupplierID`, `SupplierName`, `SupplierAddress`, `SupplierContact`) VALUES ('S004', 'PC', 'Dublin', '336699');

COMMIT;

-- -----------------------------------------------------

-- Data for table `GameStop`.`Games`

-- -----------------------------------------------------

START TRANSACTION;

USE `GameStop`;

INSERT INTO `GameStop`.`Games` (`GmeID`, `GmeName`, `GmeGenre`, `GmePrice`, `Suppliers\_SupplierID`, `Transaction\_TID`) VALUES ('G001', 'Tetris', 'Puzzle', 20.99, 'S001', 'T001');

INSERT INTO `GameStop`.`Games` (`GmeID`, `GmeName`, `GmeGenre`, `GmePrice`, `Suppliers\_SupplierID`, `Transaction\_TID`) VALUES ('G002', 'Deus ex', 'Action', 59.99, 'S002', 'T002');

INSERT INTO `GameStop`.`Games` (`GmeID`, `GmeName`, `GmeGenre`, `GmePrice`, `Suppliers\_SupplierID`, `Transaction\_TID`) VALUES ('G003', 'Minecraft', 'Puzzle', 39.99, 'S003', 'T003');

INSERT INTO `GameStop`.`Games` (`GmeID`, `GmeName`, `GmeGenre`, `GmePrice`, `Suppliers\_SupplierID`, `Transaction\_TID`) VALUES ('G004', 'Injustice', 'Action', 59.99, 'S004', 'T004');

INSERT INTO `GameStop`.`Games` (`GmeID`, `GmeName`, `GmeGenre`, `GmePrice`, `Suppliers\_SupplierID`, `Transaction\_TID`) VALUES ('G005', 'Black Ops', 'Shooter', 69.99, 'S002', 'T005');

INSERT INTO `GameStop`.`Games` (`GmeID`, `GmeName`, `GmeGenre`, `GmePrice`, `Suppliers\_SupplierID`, `Transaction\_TID`) VALUES ('G006', 'Gta V', 'Action', 69.99, 'S002', 'T006');

INSERT INTO `GameStop`.`Games` (`GmeID`, `GmeName`, `GmeGenre`, `GmePrice`, `Suppliers\_SupplierID`, `Transaction\_TID`) VALUES ('G007', 'Assaisn\'s Creed', 'Action', 69.99, 'S001', 'T007');

INSERT INTO `GameStop`.`Games` (`GmeID`, `GmeName`, `GmeGenre`, `GmePrice`, `Suppliers\_SupplierID`, `Transaction\_TID`) VALUES ('G008', 'Fifa 17', 'Sports', 69.99, 'S003', 'T008');

INSERT INTO `GameStop`.`Games` (`GmeID`, `GmeName`, `GmeGenre`, `GmePrice`, `Suppliers\_SupplierID`, `Transaction\_TID`) VALUES ('G009', 'Metal Gear Solid', 'Action', 59.99, 'S001', 'T009');

INSERT INTO `GameStop`.`Games` (`GmeID`, `GmeName`, `GmeGenre`, `GmePrice`, `Suppliers\_SupplierID`, `Transaction\_TID`) VALUES ('G010', 'Madden NFL', 'Sports', 69.99, 'S004', 'T010');

INSERT INTO `GameStop`.`Games` (`GmeID`, `GmeName`, `GmeGenre`, `GmePrice`, `Suppliers\_SupplierID`, `Transaction\_TID`) VALUES ('G011', 'Halo Wars', 'Strategy', 20.99, 'S004', 'T011');

COMMIT;

-- -----------------------------------------------------

-- Data for table `GameStop`.`WishList`

-- -----------------------------------------------------

START TRANSACTION;

USE `GameStop`;

INSERT INTO `GameStop`.`WishList` (`WishID`, `Customer\_CustomerID`, `Games\_GmeID`) VALUES ('W001', 'C001', 'G004');

INSERT INTO `GameStop`.`WishList` (`WishID`, `Customer\_CustomerID`, `Games\_GmeID`) VALUES ('W002', 'C003', 'G006');

INSERT INTO `GameStop`.`WishList` (`WishID`, `Customer\_CustomerID`, `Games\_GmeID`) VALUES ('W003', 'C001', 'G007');

INSERT INTO `GameStop`.`WishList` (`WishID`, `Customer\_CustomerID`, `Games\_GmeID`) VALUES ('W004', 'C005', 'G009');

INSERT INTO `GameStop`.`WishList` (`WishID`, `Customer\_CustomerID`, `Games\_GmeID`) VALUES ('W005', 'C002', 'G010');

INSERT INTO `GameStop`.`WishList` (`WishID`, `Customer\_CustomerID`, `Games\_GmeID`) VALUES ('W006', 'C003', 'G006');

INSERT INTO `GameStop`.`WishList` (`WishID`, `Customer\_CustomerID`, `Games\_GmeID`) VALUES ('W007', 'C006', 'G007');

INSERT INTO `GameStop`.`WishList` (`WishID`, `Customer\_CustomerID`, `Games\_GmeID`) VALUES ('W008', 'C001', 'G003');

COMMIT;

-- -----------------------------------------------------

-- Data for table `GameStop`.`Reviews`

-- -----------------------------------------------------

START TRANSACTION;

USE `GameStop`;

INSERT INTO `GameStop`.`Reviews` (`ReviewNo`, `Games\_GmeID`, `Customer\_CustomerID`) VALUES ('R001', 'G001', 'C001');

INSERT INTO `GameStop`.`Reviews` (`ReviewNo`, `Games\_GmeID`, `Customer\_CustomerID`) VALUES ('R002', 'G002', 'C002');

INSERT INTO `GameStop`.`Reviews` (`ReviewNo`, `Games\_GmeID`, `Customer\_CustomerID`) VALUES ('R003', 'G003', 'C003');

INSERT INTO `GameStop`.`Reviews` (`ReviewNo`, `Games\_GmeID`, `Customer\_CustomerID`) VALUES ('R004', 'G004', 'C004');

INSERT INTO `GameStop`.`Reviews` (`ReviewNo`, `Games\_GmeID`, `Customer\_CustomerID`) VALUES ('R005', 'G005', 'C005');

INSERT INTO `GameStop`.`Reviews` (`ReviewNo`, `Games\_GmeID`, `Customer\_CustomerID`) VALUES ('R006', 'G006', 'C006');

INSERT INTO `GameStop`.`Reviews` (`ReviewNo`, `Games\_GmeID`, `Customer\_CustomerID`) VALUES ('R007', 'G007', 'C007');

INSERT INTO `GameStop`.`Reviews` (`ReviewNo`, `Games\_GmeID`, `Customer\_CustomerID`) VALUES ('R008', 'G009', 'C009');

INSERT INTO `GameStop`.`Reviews` (`ReviewNo`, `Games\_GmeID`, `Customer\_CustomerID`) VALUES ('R009', 'G008', 'C008');

INSERT INTO `GameStop`.`Reviews` (`ReviewNo`, `Games\_GmeID`, `Customer\_CustomerID`) VALUES ('R010', 'G010', 'C010');

COMMIT;

-- -----------------------------------------------------

-- Data for table `GameStop`.`TradeIns`

-- -----------------------------------------------------

START TRANSACTION;

USE `GameStop`;

INSERT INTO `GameStop`.`TradeIns` (`GmeID`, `PurchasePrice`, `SalesPrice`, `Conditions`, `Games\_GmeID`) VALUES ('G001', 7.99, 15.99, 'Good', 'G001');

INSERT INTO `GameStop`.`TradeIns` (`GmeID`, `PurchasePrice`, `SalesPrice`, `Conditions`, `Games\_GmeID`) VALUES ('G002', 10.99, 20.99, 'Good', 'G002');

INSERT INTO `GameStop`.`TradeIns` (`GmeID`, `PurchasePrice`, `SalesPrice`, `Conditions`, `Games\_GmeID`) VALUES ('G003', 10.99, 20.99, 'Good', 'G003');

INSERT INTO `GameStop`.`TradeIns` (`GmeID`, `PurchasePrice`, `SalesPrice`, `Conditions`, `Games\_GmeID`) VALUES ('G004', 5.99, 20.99, 'Playable', 'G004');

INSERT INTO `GameStop`.`TradeIns` (`GmeID`, `PurchasePrice`, `SalesPrice`, `Conditions`, `Games\_GmeID`) VALUES ('G005', 10.99, 20.99, 'Playable', 'G005');

INSERT INTO `GameStop`.`TradeIns` (`GmeID`, `PurchasePrice`, `SalesPrice`, `Conditions`, `Games\_GmeID`) VALUES ('G006', 10.99, 20.99, 'Good', 'G006');

INSERT INTO `GameStop`.`TradeIns` (`GmeID`, `PurchasePrice`, `SalesPrice`, `Conditions`, `Games\_GmeID`) VALUES ('G007', 10.99, 20.99, 'Good', 'G007');

INSERT INTO `GameStop`.`TradeIns` (`GmeID`, `PurchasePrice`, `SalesPrice`, `Conditions`, `Games\_GmeID`) VALUES ('G008', 15.99, 30.99, 'Excellent', 'G008');

COMMIT;

### Queries

*List the cheapest game and call it LowestPricedGame from Games table*

Select Min(GMEPRICE) as LowestPricedGame, GmeName from Games;

# LowestPricedGame, GmeName

'21', 'Tetris'

*This sub query gets the ID of the employee from Transaction table who sold the most Quantity to be name Employee of the Month.*

SELECT Employee\_EmpNo as EmployeeOfTheMonth

FROM Transaction

WHERE Qty=

(SELECT MAX(Qty)

FROM transaction);

# EmployeeOfTheMonth

'E010'

*This retrieves the Names of all the customers from Wexford that both customers required the game be sent by mail.*

SELECT customername as CustomerFromWexford FROM Customer

WHERE CUSTOMERADDRESS like '%wex%';

# CustomerFromWexford

'Michael Kelly'

'Peggy Ryan'

'Donna Watson'

'Jesse Bryant'

*This returns the dearest game*

SELECT GMENAME as DearestGames

FROM games

WHERE GMEPRICE=

(SELECT MAX(GmePrice)

FROM games);

# DearestGames

'Black Ops'

'Gta V'

'Assaisn\'s Creed'

'Fifa 17'

'Madden NFL'

*Count the amount of transactions completed by the customer with the code C006 as they might have reached an award stage with the amount of transaction that they have done.*

SELECT COUNT(Customer\_CustomerID) AS OrdersFromCustomerc006 FROM transaction

WHERE Customer\_CustomerID='C006';

# OrdersFromCustomerc006

'2'

This will display the Employee ID from the transaction table that has sold more Qty than the average Qty sold. To keep track who is performing well in their job.

SELECT distinct Employee\_EmpNo as EmployeeID FROM transaction

WHERE QTY>(SELECT AVG(QTY) FROM Transaction);

# EmployeeID

'E003'

'E006'

'E009'

'E010'

This is just to keep record on the average amount of items sold per transaction for the books

SELECT AVG(QTY) as AverageItemsSold

FROM Transaction;

# AverageItemsSold

'7.5385'

What this query does it extracts the all the information required made by purchases on what customer bought what merchandise that would have been promoted, this is to maintain knowledge of which promotion event on what merchandise is doing the best.

select p.PromotionCode as Promotion\_Code , p.Event as Promotional\_Eent,m.MerchID as Merchandise\_ID,

t.PurchasesID, c.CustomerID

from promotions p inner join merchandise m

on p.PromotionCode = m.Promotions\_PromotionCode

inner join purchases t

on m.Purchases\_PurchasesID = t.PurchasesID

inner join customer c

on t.Customer\_CustomerID = c.CustomerID

ORDER BY c.CustomerID;

# Promotion\_Code, Promotional\_Eent, Merchandise\_ID, PurchasesID, CustomerID

'PR001', 'Crazy Bonanaza', 'Mc001', 'P001', 'C001'

'PR001', 'Crazy Bonanaza', 'Mc002', 'P002', 'C002'

'PR002', 'Wild Week', 'Mc004', 'P004', 'C003'

'PR001', 'Crazy Bonanaza', 'Mc003', 'P003', 'C004'

'PR003', '2 Week Fest', 'Mc005', 'P005', 'C005'

With this query, we are checking to see which games are listed on the wish list and must be put away.

select GmeID, GmeName as Games\_To\_Be\_Put\_Away from games g

right outer join wishlist fp

on g.GmeID = fp.Games\_GmeID;

# GmeID, Games\_To\_Be\_Put\_Away

'G003', 'Minecraft'

'G004', 'Injustice'

'G006', 'Gta V'

'G006', 'Gta V'

'G007', 'Assaisn\'s Creed'

'G007', 'Assaisn\'s Creed'

'G009', 'Metal Gear Solid'

'G010', 'Madden NFL'

This is returning information on the games that are in good or excellent condition as required by the customer who only wants the best quality of traded in games.

SELECT GmeID as ID,GMEName as Games\_In\_Good\_or\_Excellent\_Condition

FROM Games

WHERE GmeID IN (SELECT GmeID

FROM tradeins

WHERE Conditions='Good' OR Conditions='Excellent');

# ID, Games\_In\_Good\_or\_Excellent\_Condition

'G001', 'Tetris'

'G002', 'Deus ex'

'G003', 'Minecraft'

'G006', 'Gta V'

'G007', 'Assaisn\'s Creed'

'G008', 'Fifa 17'

This displays the how many customers are employed in each of the branch locations, which is important to keep track of.

SELECT branch.branchID, branch.BranchLocation,count(employee.EmpNo) AS NumberOfEmployees

FROM employee

LEFT JOIN branch

ON employee.Branch\_BranchID=branch.BranchID

GROUP BY BranchID;

# branchID, BranchLocation, NumberOfEmployees

'B001', 'Wexford', '4'

'B002', 'Waterford', '3'

'B003', 'Kilkenny', '3'

### Users

There will be three users in the database, each with their own access to the database and what they can modify. The Employee will be able to view the Games, Reviews, Trade-Ins, Wish List. The Employee does not need access to any of the financial aspects of the business that would mostly be for the manager or owner maybe. The Manager will have the same access as the Employee but be able to view all the transactions that took place and to see how well the shop is doing on a financial level. The Head of the company will have access to the entire database and be able to view everything as he would need to know where a business is succeeding or failing. He will only be mainly interested in how will the business is doing on a financial level.

**Employee**

Create User 'Employee@LocalHost' Identified BY 'Emp123'

Grant Select,insert,update on GameStop.Transaction to Employee@LocalHost

**Manager**

Create User 'Manger@LocalHost' Identified BY 'Man123'

Grant Select, Insert, Update on GameStop.Games to Manger @LocalHost

Grant Select, Insert, Update on GameStop.Reviews to Manger @LocalHost

Grant Select, Insert, Update on GameStop.Trade-Ins to Manger @LocalHost

Grant Select, Insert, Update on GameStop.WishList to Manger@LocalHost

Grant Select, Insert, Update on GameStop.Suppliers to Manger@LocalHost

Grant Select, Insert, Update on GameStop.Customer to Manger@LocalHost

Grant Select, Insert, Update on GameStop.Transaction to Manger@LocalHost

Grant Select, Insert, Update on GameStop.Employee to Manger@LocalHost

Grant Select, Insert, Update on GameStop.Promotions to Manger@LocalHost

Grant Select, Insert, Update on GameStop.Merchandise to Manger@LocalHost

Grant Select, Insert, Update on GameStop.MerchatSuppliers to Manger@LocalHost

Grant Select, Insert, Update on GameStop.Purchases to Manger@LocalHost

**CEO**

Create User 'Ceo@LocalHost' Identified BY 'Ceo123'

Grant All on GameStop. \* to Ceo@LocalHost with Grant Option;

# Views

There would be different views based on the user of the database, as an Employee would not need the same views as a manager or a Ceo. I made these views to show what each user would need to use the database for the individual tables the need.

Create View Transaction\_Emp As Select Qty,TID,Employee\_EmpNo From Transaction;

Create View TradeStock as Select SalesPrice, Conditions, GmeID from TradeIns where Conditions = "Excellent";

Create View PromStart As Select Event, StartDate, EndDateFrom Promotions where StartDate = "1/3/2017";

Create View TopSeller As Select TID, Employee\_EmpNo From Transaction where Date = "3/1/2017";

# Conclusion

In doing this project, we learned the way a database is fragmented and why you would use fragmentation in a database. We enjoyed doing this project as it gave us an understanding how to use workbench and be able to generate databases using workbench. We found forward engineering the database to be a valuable tool to know for future when we are creating are own databases. In addition, this was one of the databases projects that we enjoyed doing, as it was interesting to see how easy it is to create the whole process through forward engineering.