Jay Lee

100357736

CPSC 2221 – 02 Lab 2

1. **Query for creating tables**
2. **Create Tables Query & Foreign Key**

DROP TABLE IF EXISTS Region;

CREATE TABLE Region(

RegionID char(1) PRIMARY KEY,

RegionName varchar(35) NOT NULL

);

DROP TABLE IF EXISTS Store;

CREATE TABLE Store(

StoreID varchar(10) PRIMARY KEY,

StoreZip char(6) NOT NULL,

RegionID char(1) NOT NULL,

FOREIGN KEY(RegionID) REFERENCES Region(RegionID)

);

DROP TABLE IF EXISTS Vendor;

CREATE TABLE Vendor(

VendorID char(2) PRIMARY KEY,

VendorName varchar(35) NOT NULL

);

DROP TABLE IF EXISTS Category;

CREATE TABLE Category(

CategoryID char(2) PRIMARY KEY,

CategoryName varchar(35) NOT NULL

);

DROP TABLE IF EXISTS Product;

CREATE TABLE Product(

ProductID char(3) PRIMARY KEY,

ProductName varchar(35) NOT NULL,

ProductPrice int NOT NULL,

VendorID char(2) NOT NULL,

CategoryID char(2) NOT NULL,

FOREIGN KEY(VendorID) REFERENCES Vendor(vendorid),

FOREIGN KEY (CategoryID) REFERENCES Category(CategoryID)

);

DROP TABLE IF EXISTS Customer;

CREATE TABLE Customer(

CustomerID char(7) PRIMARY KEY,

CustomerName varchar(35) NOT NULL,

CustomerZip char(6) NOT NULL

);

DROP TABLE IF EXISTS SalesTransaction;

CREATE TABLE SalesTransaction(

TID char(4) PRIMARY KEY,

CustomerID char(7) NOT NULL,

StoreID varchar(10) NOT NULL,

TDate date,

FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),

FOREIGN KEY (StoreID) REFERENCES Store(StoreID)

);

DROP TABLE IF EXISTS Soldvia;

CREATE TABLE Soldvia(

ProductID char(3) NOT NULL,

TID char(4) NOT NULL,

FOREIGN KEY (ProductID) REFERENCES Product(ProductID),

FOREIGN KEY (TID) REFERENCES SalesTransaction(TID),

NoOfItems int NOT NULL,

PRIMARY KEY (ProductID, TID)

);

1. **Query for creating tables**

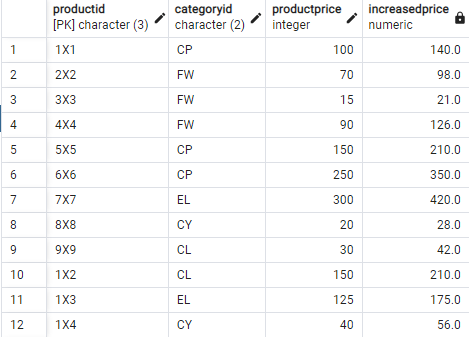
Queries:

1. SELECT \*   
   FROM Product;  
   Table

   Description automatically generated
2. SELECT VendorID, VendorName   
   FROM Vendor;  
   Table

   Description automatically generated
3. SELECT CustomerName, CustomerZip   
   FROM Customer;  
   Table

   Description automatically generated
4. SELECT ProductName, ProductID, CategoryID, ProductPrice   
   FROM Product;  
   Table

   Description automatically generated
5. SELECT ProductID, CategoryID, ProductPrice, ProductPrice \* 1.4 AS IncreasedPrice   
   FROM Product;  
   
6. SELECT ProductID, ProductName, VendorID, CategoryID, ProductPrice  
   FROM Product   
   WHERE ProductPrice >= 100;  
   Graphical user interface

   Description automatically generated with medium confidence
7. SELECT ProductID, ProductName, ProductPrice   
   FROM Product   
   WHERE CategoryID = 'FW' AND ProductPrice <= 200;  
   Table

   Description automatically generated
8. SELECT VendorID   
   FROM Vendor   
   GROUP BY VendorID;  
   Table

   Description automatically generated
9. SELECT AVG(ProductPrice) AS AvgPrice   
   FROM Product;  
   Graphical user interface, application

   Description automatically generated
10. SELECT count(\*) AS NumberOfProducts   
    FROM Product;  
    Graphical user interface, text, application

    Description automatically generated
11. SELECT COUNT(DISTINCT VendorID) AS NumberOfVendors   
    FROM Product;  
    Graphical user interface, text, application

    Description automatically generated
12. SELECT COUNT(\*) AS NumberOfProducts, AVG(ProductPrice) AS AVGPrice, MIN(ProductPrice) AS Lowest, MAX(ProductPrice) AS Highest   
    FROM Product   
    WHERE CategoryID = 'CP';  
    Graphical user interface, text, application, table

    Description automatically generated
13. SELECT ProductID, ProductName, CategoryID, ProductPrice   
    FROM Product   
    WHERE CategoryID = 'FW'   
    ORDER BY ProductPrice DESC;  
    Graphical user interface, table

    Description automatically generated with medium confidence
14. SELECT ProductID AS "PRODUCTID", COUNT(NoOfItems) AS "Total Sold"   
    FROM Soldvia   
    GROUP BY ProductId;  
    Table

    Description automatically generated