CREATE TABLE BusinessCustomer(

username varchar(20) NOT NULL,

userpassword varchar(30),

company\_name varchar(30),

street varchar(20),

city varchar(20),

state varchar(20),

zipcode varchar(10),

business\_category varchar(10),

company\_income REAL,

PRIMARY KEY(username));

INSERT INTO BusinessCustomer VALUES(

"firstcompany","123456","firstcompany",

"Fifth", "Pittsburgh",

"PA","15001",

"IT", 2000000);

INSERT INTO BusinessCustomer VALUES(

"secondcompany","654321"," secondcompany ",

"Fifth", "Pittsburgh",

"PA","15001",

"Media", 500000)

CREATE TABLE HomeCustomer(

username VARCHAR(20) NOT NULL,

userpassword varchar(30),

firstname VARCHAR(10),

lastname VARCHAR(10),

street VARCHAR(20),

city VARCHAR(20),

state VARCHAR(20),

zipcode VARCHAR(10),

marriage\_status VARCHAR(20),

gender VARCHAR(10),

age INT,

income REAL,

PRIMARY KEY(username));

INSERT INTO HomeCustomer VALUES(

"10002","123123","David","White",

"Sixth", "Pittsburgh",

"PA","15012",

"married","male",38,50000);

INSERT INTO HomeCustomer VALUES(

"10004","321321","Casey","Green",

"Forbes", "Pittsburgh",

"PA","15052",

"unmarried","female",21,18000)

CREATE TABLE Products(

productID varchar(10) NOT NULL,

product\_name varchar(10),

inventory\_amount int,

price REAL,

brand varchar(20),

product\_description varchar(100),

PRIMARY KEY(productID)

);

INSERT INTO Products VALUES(

"001","iphone4",20,300,"Apple","This is a phone from Apple");

INSERT INTO Products VALUES(

"002","Galaxy3",50,200,"Samsung","This is a phone from Samsung")

CREATE TABLE administrator(

admin\_name varchar(10) NOT NULL,

admin\_password varchar(30),

PRIMARY KEY(admin\_name)

);

INSERT INTO administrator VALUES(

"admin","admin");

CREATE TABLE Salesperson(

sales\_username varchar(10) NOT NULL,

sales\_password varchar(30),

sales\_name varchar(15),

address varchar(20),

email varchar(20),

job\_title varchar(15),

store\_assigned varchar(10),

salary int,

PRIMARY KEY(sales\_username),

FOREIGN KEY(store\_assigned) REFERENCES Store(storeID)

ON UPDATE CASCADE

ON DELETE SET NULL);

INSERT INTO Salesperson VALUES(

"001","123123","Joey","Penn","joey@gmail.com","salesman",

"01",1200);

INSERT INTO Salesperson VALUES(

"002","123123","Pheobe","Penn","Pheobe@gmail.com","manager",

"01",1800);

INSERT INTO Salesperson VALUES(

"003","123123","Monica","Penn","monica@gmail.com","region manager",

"01",2500);

INSERT INTO Salesperson VALUES(

"004","123123","Rachel","Centre","rachel@gmail.com","salesman",

"02",1300);

CREATE TABLE store(

storeID varchar(10),

address varchar(20),

manager varchar(10),

num\_of\_salespersons int,

region varchar(10),

PRIMARY KEY (storeID),

FOREIGN KEY(manager) REFERENCES Salesperson(salesID)

ON UPDATE CASCADE,

FOREIGN KEY(region) REFERENCES Region(regionID)

ON UPDATE CASCADE

);

\*if manager or region is deleted, the delete action will be rejected. It has to update first, then delete.

INSERT INTO store VALUES(

"01","Penn","Pheobe",8,"shadyside");

INSERT INTO store VALUES(

"02","Centre",null,12,"shadyside");

CREATE TABLE Region(

regionID varchar(10) NOT NULL,

region\_name varchar(20),

region\_manager varchar(10),

PRIMARY KEY(regionID),

FOREIGN KEY(region\_manager) REFERENCES Salesperson(salesID)

ON UPDATE CASCADE

ON DELETE SET NULL);

\*region manager can be non-available when the previous manager was fired.

CREATE TABLE Transaction(

orderID varchar(10) NOT NULL,

productID varchar(10),

salesperson varchar(10),

quantity int,

customer varchar(20),

PRIMARY KEY(orderID, productID),

FOREIGN KEY(orderID) REFERENCES Orders(orderID)

ON DELETE CASCADE

ON UPDATE CASCADE,

FOREIGN KEY(productID) REFERENCES Products(productID)

ON UPDATE CASCADE,

FOREIGN KEY(salesperson) REFERENCES Salespersons(salesID)

ON DELETE SET NULL

ON UPDATE CASCADE,

FOREIGN KEY(customer) REFERENCES Customers(customerID)

);

CREATE TABLE shoppingcart(

productID varchar(10) NOT NULL,

product\_name varchar(10),

customerID varchar(20),

quantity int,

price real,

PRIMARY KEY(customerID,productID),

FOREIGN KEY(customerID) REFERENCES customer(customerID),

FOREIGN KEY(productID) REFERENCES customer(productID)

);