Database Concepts Assignment Set 3

Instructions:

1. Please type the answers below the questions directly. You may insert tables or figures. Scans of handwritten papers are not acceptable.
2. When it is done, rename the file to firstname-lastname.docx, export it into a PDF file, and submit it online by the deadline.
3. Academic integrity is strictly reinforced. Detected plagiarized works will receive zero points and potentially a failure of the whole course.

Problem description:

James River Jewelry is a small jewelry shop. While James River Jewelry does sell typical jewelry purchased from jewelry vendors, including such items as rings, necklaces, earrings, and watches, it specializes in hard-to-find Asian jewelry. Although some Asian jewelry is manufactured jewelry purchased from vendors in the same manner as the standard jewelry is obtained, many of the Asian jewelry pieces are often unique single items purchased directly from the artisan who created the piece (the term "manufactured" would be an inappropriate description of these pieces). James River Jewelry has a small but loyal clientele, and it wants to further increase customer loyalty by creating a frequent buyer program. In this program, after every 10 purchases, a customer will receive a credit equal to 50 percent of the average of his or her 10 most recent purchases. This credit must be applied to the next (or 11th) purchase.

Assume that James River designs a database with the following tables.

CUSTOMER (CustomerID, LastName, FirstName, Phone, EmailAddress)

PURCHASE (InvoiceNumber, InvoiceDate, PreTaxAmount, *CustomerID*)

PURCHASE_ITEM (*InvoiceNumber*, InvoiceLineNumber, *ItemNumber*, RetailPrice)

ITEM (ItemNumber, ItemDescription, Cost, ArtistLastName, ArtistFirstName)

The referential integrity constraints are:

CustomerID in PURCHASE must exist in CustomerID in CUSTOMER

InvoiceNumber in PURCHASE_ITEM must exist in InvoiceNumber in PURCHASE

ItemNumber in PURCHASE_ITEM must exist in ItemNumber in ITEM

Assume that CustomerID of CUSTOMER, ItemNumber of ITEM, and InvoiceNumber of PURCHASE

are all surrogate keys with values as follows:

CustomerID	***Start at 1***	***Increment by 1***
InvoiceNumber	***Start at 1001***	***Increment by 1***
ItemNumber	***Start at 1***	***Increment by 1***

Data for the James River Jewelry tables is shown in the following figures. These tables, referential integrity constraints, and data are used as the basis for the SQL statements you will create in the exercises that follow.

Write SQL statements and answer questions for this database as follows:

A. *Write CREATE TABLE statements for each of these tables including foreign key constraints for the relationships in each of these tables. Make your own assumptions regarding cascading deletions and justify those assumptions*

TABLE: CUSTOMER

```
CREATE TABLE CUSTOMER(
   CustomerID      INT PRIMARY KEY AUTO_INCREMENT,
   LastName        VARCHAR(25) NOT NULL,
   FirstName       VARCHAR(25) NOT NULL,
   Phone           VARCHAR(12) DEFAULT NULL,
   Email           VARCHAR(50) UNIQUE NOT NULL
);
SET @@auto_increment_increment=1;
```

TABLE: PURCHASE

```
CREATE TABLE PURCHASE(
   InvoiceNumber   INT PRIMARY KEY AUTO_INCREMENT,
   InvoiceDate     DATETIME DEFAULT NULL,
   AmountBeforeTax NUMERIC(5,2) NOT NULL,
   CustID          INT NOT NULL,
```

```
    FOREIGN KEY(CustID) REFERENCES CUSTOMER(CustomerID)
    ON UPDATE CASCADE ON DELETE NO ACTION
);
SET @@auto_increment_increment=1000;
```

ON UPDATE CASCADE: When you modify a CustomerID in the CUSTOMER table, the system will automatically adjust corresponding entries in the PURCHASE table that reference this particular CUSTOMER. Note: this action does not activate any triggers on the PURCHASE table. The system efficiently manages these updates for you.

ON DELETE CASCADE: Use this with caution. Deleting a record from the CUSTOMER table will also remove associated records from the PURCHASE table. While this can be beneficial for automatic cleanup of related tables, it may not be the ideal choice, especially in a CUSTOMER to PURCHASE relationship context. It's generally advisable to opt for ON DELETE NO ACTION, which prevents unintended deletions by rejecting such operations.

TABLE: PURCHASE_ITEM

```
CREATE TABLE PURCHASE_ITEM(
    InvNumber       INT NOT NULL,
    LineNumber      INT NOT NULL,
    ProductID       INT NOT NULL,
    Price           NUMERIC(5,2) NOT NULL,
    PRIMARY KEY (InvNumber, LineNumber),
    FOREIGN KEY(InvNumber) REFERENCES PURCHASE(InvoiceNumber)
    ON UPDATE CASCADE ON DELETE NO ACTION,
    FOREIGN KEY(ProductID) REFERENCES ITEM(ItemNumber)
    ON UPDATE CASCADE ON DELETE NO ACTION
);
```

ON UPDATE CASCADE: When you modify an InvoiceNumber in the PURCHASE table, the system will automatically adjust corresponding entries in the PURCHASE_ITEM table linked to that particular PURCHASE. Please be aware that this won't activate any triggers in the PURCHASE_ITEM table. The system ensures these updates are handled seamlessly.

ON DELETE CASCADE: Use with caution. Erasing a record from the PURCHASE table will concurrently remove its associated records from the PURCHASE_ITEM table. Though this feature can be helpful for auto-cleaning related tables, its unintended consequences could be problematic. Therefore, it's typically recommended to select ON DELETE NO ACTION, which blocks and declines such deletion actions.

TABLE: ITEM

```
CREATE TABLE ITEM(
   ItemNumber       INT PRIMARY KEY AUTO_INCREMENT,
   Description      VARCHAR(50) NOT NULL,
   UnitCost         NUMERIC(5,2) NOT NULL,
   ArtistLast       VARCHAR(25) NOT NULL,
   ArtistFirst      VARCHAR(25) NOT NULL
);
SET @@auto_increment_increment=1;
```

B. *Write SQL statements to insert the data shown in the following figures into these tables. Assume that surrogate key column values will be supplied by the DBMS.*

CUSTOMER (CustomerID, LastName, FirstName, Phone, EmailAddress)

PURCHASE (InvoiceNumber, InvoiceDate, PreTaxAmount, *CustomerID*)

PURCHASE_ITEM (*InvoiceNumber*, InvoiceLineNumber, *ItemNumber*, RetailPrice)

ITEM (ItemNumber, ItemDescription, Cost, ArtistLastName, ArtistFirstName)

CUSTOMER:

CustomerID	LastName	FirstName	Phone	EmailAddress
1	Stanley	Elizabeth	555-236-7789	Elizabeth.Stanley@somewhere.com
2	Price	Fred	555-236-0091	Fred.Price@somewhere.com
3	Becky	Linda	555-236-0392	Linda.Becky@somewhere.com
4	Birch	Pamela	555-236-4493	Pamela.Birch@somewhere.com
5	Romez	Ricardo	555-236-3334	Ricardo.Romez@somewhere.com
6	Jackson	Samantha	555-236-1095	Samantha.Jackson@somewhere.com

INSERT INTO CUSTOMER (CustomerID, LastName, FirstName, Phone, EmailAddress) VALUES
('1', 'Stanley', 'Elizabeth', '555-236-7789', 'Elizabeth.Stanley@somewhere.com'),
('2', 'Price', 'Fred', '555-236-0091', 'Fred.Price@somewhere.com'),
('3', 'Becky', 'Linda', '555-236-0392', 'Linda.Becky@somewhere.com'),
('4', 'Birch', 'Pamela', '555-236-4493', 'Pamela.Birch@somewhere.com'),
('5', 'Romez', 'Ricardo', '555-236-3334', 'Ricardo.Romez@somewhere.com'),
('6', 'Jackson', 'Samantha', '555-236-1095', 'Samantha.Jackson@somewhere.com');

ITEM:

ItemNumber	ItemDescription	Cost	ArtistLastName	ArtistFirstName
1	Gold Bracelet	120.00	Josephson	Mary
2	Gold Necklace	160.00	Baker	Samantha
3	Bead Earrings	50.00	Josephson	Mary
4	Gold Bracelet	180.00	Baker	Samantha
5	Silver Necklace	135.00	Baxter	Sam
6	Bead Earrings	25.00	Josephson	Mary
7	Bead Earrings	22.50	Josephson	Mary
8	Gold Earrings	50.00	Lintz	John
9	Gold Necklace	160.00	Lintz	John
10	Bead Earrings	20.00	Josephson	Mary
11	Bead Earrings	35.00	Josephson	Mary
12	Bead Earrings	45.00	Josephson	Mary
13	Gold Necklace	225.00	Lintz	John
14	Silver Earrings	55.00	Lintz	John
15	Gold Bracelet	200.00	Lintz	John
16	Bead Earrings	25.00	Josephson	Mary
17	Bead Earrings	45.00	Josephson	Mary
18	Gold Bracelet	210.00	Baker	Samantha
19	Silver Necklace	165.00	Baxter	Sam

INSERT INTO ITEM (ItemNumber, ItemDescription, Cost, ArtistLastName, ArtistFirstName) VALUES
('1', 'Gold Bracelet', 120.00, 'Josephson', 'Mary'),
('2', 'Gold Necklace', 160.00, 'Baker', 'Samantha'),
('3', 'Bead Earrings', 50.00, 'Josephson', 'Mary'),
('4', 'Gold Bracelet', 180.00, 'Baker', 'Samantha'),
('5', 'Silver Necklace', 135.00, 'Baxter', 'Sam'),
('6', 'Bead Earrings', 25.00, 'Josephson', 'Mary'),
('7', 'Bead Earrings', 22.50, 'Josephson', 'Mary'),
('8', 'Gold Earrings', 50.00, 'Lintz', 'John'),
('9', 'Gold Necklace', 160.00, 'Lintz', 'John'),
('10', 'Bead Earrings', 20.00, 'Josephson', 'Mary'),
('11', 'Bead Earrings', 35.00, 'Josephson', 'Mary'),
('12', 'Bead Earrings', 45.00, 'Josephson', 'Mary'),

(‘13’, 'Gold Necklace', 225.00, 'Lintz', 'John'),
(‘14’, 'Silver Earrings', 55.00, 'Lintz', 'John'),
(‘15’, 'Gold Bracelet', 200.00, 'Lintz', 'John'),
(‘16’, 'Bead Earrings', 25.00, 'Josephson', 'Mary'),
(‘17’, 'Bead Earrings', 45.00, 'Josephson', 'Mary'),
(‘18’, 'Gold Bracelet', 210.00, 'Baker', 'Samantha'),
(‘19’, 'Silver Necklace', 165.00, 'Baxter', 'Sam');

PURCHASE:

InvoiceNumber	InvoiceDate	PreTaxAmount	CustomerID
1001	2019-05-05	155.00	1
1002	2019-05-07	203.00	2
1003	2019-05-11	75.00	3
1004	2019-05-15	67.00	4
1005	2019-05-15	330.00	5
1006	2019-05-16	25.00	1
1007	2019-05-25	45.00	3
1008	2019-06-06	445.00	1
1009	2019-06-07	72.00	6

INSERT INTO PURCHASE (InvoiceNumber, InvoiceDate, PreTaxAmount, CustomerID) VALUES
(‘1001’, ’'2019-05-05', 155.00, 1),
(‘1002’, '2019-05-07', 203.00, 2),
(‘1003’, '2019-05-11', 75.00, 3),
(‘1004’, '2019-05-15', 67.00, 4),
(‘1005’, '2019-05-15', 330.00, 5),
(‘1006’, '2019-05-16', 25.00, 1),
(‘1007’, '2019-05-25', 45.00, 3),
(‘1008’, '2019-06-06', 445.00, 1),
(‘1009’, '2019-06-07', 72.00, 6);

INVOICE:

InvoiceNumber	InvoiceLineNumber	ItemNumber	RetailPrice
1001	1	1	155.00
1002	1	2	203.00
1003	1	3	75.00
1004	1	6	35.00
1004	2	7	32.00
1005	1	4	240.00
1005	2	8	90.00
1006	1	10	25.00
1007	1	11	45.00
1008	1	5	175.00
1008	2	9	215.00
1008	3	12	55.00
1009	1	14	72.00

INSERT INTO PURCHASE_ITEM (InvoiceNumber, InvoiceLineNumber, ItemNumber, RetailPrice) VALUES
(1001, 1, 1, 155.00),
(1002, 1, 2, 203.00),
(1003, 1, 3, 75.00),
(1004, 1, 6, 35.00),
(1004, 2, 7, 32.00),
(1005, 1, 4, 240.00),
(1005, 2, 8, 90.00),
(1006, 1, 10, 25.00),
(1007, 1, 11, 45.00),
(1008, 1, 5, 175.00),
(1008, 2, 9, 215.00),
(1008, 3, 12, 55.00),
(1009, 1, 14, 72.00);