**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 to Blackboard 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:*

1. *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*

**CUSTOMER:**

CREATE TABLE CUSTOMER(

CustomerID Int NOT NULL AUTO\_INCREMENT PRIMARY KEY,

LastName VarChar(25) NOT NULL,

FirstName VarChar(25) NOT NULL,

Phone VarChar(12) NULL,

EmailAddress VarChar(12) NOT NULL UNIQUE,

);

ALTER TABLE CUSTOMER AUTO\_INCREMENT=1;

**PURCHASE:**

CREATE TABLE PURCHASE(

InvoiceNumber Int NOT NULL AUTO\_INCREMENT PRIMARY KEY,

InvoiceDate DateTime NULL,

PreTaxAmount Numeric(5,2) NOT NULL,

CustomerID Int NOT NULL,

CONSTRAINT PUR\_CUS\_FK FOREIGN KEY(CustomerID)

REFERENCES CUSTOMER(CustomerID)

ON UPDATE CASCADE

ON DELETE NO ACTION

);

ALTER TABLE PURCHASE AUTO\_INCREMENT=1000;

**ON UPDATE CASCADE:** If you update a CustomerID in a row of table CUSTOMER the engine will update it accordingly on all PURCHASE rows referencing this CUSTOMER (but no triggers activated on PURCHASE table, warning). The engine will track the changes for you, it's good.

**ON DELETE CASCADE:** *dangerous* : if you delete a customer row in table CUSTOMER the engine will delete as well the related PURCHASE. This is dangerous but can be used to make automatic cleanups on secondary tables (so it can be something you want, but quite certainly not for a CUSTOMER<->PURCHASE example). That’s why it’s always good to use ON DELETE NO ACTION: it will restrict the delete and will reject the operation.

**PURCHASE\_ITEM:**

CREATE TABLE PURCHASE\_ITEM(

InvoiceNumber Int NOT NULL,

InvoiceLineNumber Int NOT NULL,

ItemNumber Int NOT NULL,

RetailPrice Numeric(5,2) NOT NULL,

CONSTRAINT PI\_PRI\_KEY PRIMARY KEY (InvoiceNumber, InvoiceLineNumber),

CONSTRAINT PI\_PUR\_FK FOREIGN KEY(InvoiceNumber)

REFERENCES PURCHASE(InvoiceNumber)

ON UPDATE CASCADE

ON DELETE NO ACTION,

CONSTRAINT PI\_IT\_FK FOREIGN KEY(ItemNumber)

REFERENCES ITEM(ItemNumber)

ON UPDATE CASCADE

ON DELETE NO ACTION

);

**ON UPDATE CASCADE:** If you update a InvoiceNumber in a row of table PURCHASE the engine will update it accordingly on all PURCHASE\_ITEM rows referencing this PURCHASE(but no triggers activated on PURCHASE\_ITEM table, warning). The engine will track the changes for you, it's good.

**ON DELETE CASCADE:** *dangerous* : if you delete a purchase row in table PURCHASE the engine will delete as well the related PURCHASE\_ITEM. This is dangerous but can be used to make automatic cleanups on secondary tables. That’s why it’s always good to use ON DELETE NO ACTION: it will restrict the delete and will reject the operation.

**ITEM:**

CREATE TABLE ITEM(

ItemNumber Int NOT NULL AUTO\_INCREMENT PRIMARY KEY,

ItemDescription VarChar(50) NOT NULL,

Cost Numeric(5,2) NOT NULL,

ArtistLastName VarChar(25) NOT NULL,

ArtistFirstName VarChar(25) NOT NULL,

);

ALTER TABLE ITEM AUTO\_INCREMENT=1;

1. *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:**

*Graphical user interface, text, table

Description automatically generated*

INSERT INTO CUSTOMER VALUES(NULL, 'Stanley', 'Elizabeth', ‘555-236-7789’, '[Elizabeth.Stanley@somewhere.com](mailto:Elizabeth.Stanley@somewhere.com)');

INSERT INTO CUSTOMER VALUES(NULL, 'Price’, 'Fred', ‘555-236-0091’, 'Fred.Price[@somewhere.com](mailto:Elizabeth.Stanley@somewhere.com)');

INSERT INTO CUSTOMER VALUES(NULL, 'Becky’, 'Linda', ‘555-236-0392’, 'Linda.Becky[@somewhere.com](mailto:Elizabeth.Stanley@somewhere.com)');

INSERT INTO CUSTOMER VALUES(NULL, 'Birch', 'Pamela', ‘555-236-4493’, 'Pamela.Birch[@somewhere.com](mailto:Elizabeth.Stanley@somewhere.com)');

INSERT INTO CUSTOMER VALUES(NULL, 'Romez', ‘Ricardo', ‘555-236-3334’, 'Ricardo.Romez[@somewhere.com](mailto:Elizabeth.Stanley@somewhere.com)');

INSERT INTO CUSTOMER VALUES(NULL, 'Jackson', 'Samantha', ‘555-236-1095’, 'Samantha.Jackson@somewhere.com');

**ITEM:**

Table

Description automatically generated

INSERT INTO ITEM VALUES(NULL, 'Gold Bracelet', 120.00, 'Josephson', 'Mary');

INSERT INTO ITEM VALUES(NULL, 'Gold Necklace', 160.00, 'Baker', 'Samantha');

INSERT INTO ITEM VALUES(NULL, 'Bead Earrings', 50.00, 'Josephson', 'Mary');

INSERT INTO ITEM VALUES(NULL, 'Gold Bracelet', 180.00, 'Baker', 'Samantha');

INSERT INTO ITEM VALUES(NULL, 'Silver Necklace', 135.00, 'Baxter', 'Sam');

INSERT INTO ITEM VALUES(NULL, 'Bead Earrings', 25.00, 'Josephson', 'Mary');

INSERT INTO ITEM VALUES(NULL, 'Bead Earrings', 22.50, 'Josephson', 'Mary');

INSERT INTO ITEM VALUES(NULL, 'Gold Earrings', 50.00, 'Lintz', 'John');

INSERT INTO ITEM VALUES(NULL, 'Gold Necklace', 160.00, 'Lintz', 'John');

INSERT INTO ITEM VALUES(NULL, 'Bead Earrings', 20.00, 'Josephson', 'Mary');

INSERT INTO ITEM VALUES(NULL, 'Bead Earrings', 35.00, 'Josephson', 'Mary');

INSERT INTO ITEM VALUES(NULL, 'Bead Earrings', 45.00, 'Josephson', 'Mary');

INSERT INTO ITEM VALUES(NULL, 'Gold Necklace', 225.00, 'Lintz', 'John');

INSERT INTO ITEM VALUES(NULL, 'Silver Earrings', 55.00, 'Lintz', 'John');

INSERT INTO ITEM VALUES(NULL, 'Gold Bracelet', 200.00, 'Lintz', 'John');

INSERT INTO ITEM VALUES(NULL, 'Bead Earrings', 25.00, 'Josephson', 'Mary');

INSERT INTO ITEM VALUES(NULL, 'Bead Earrings', 45.00, 'Josephson', 'Mary');

INSERT INTO ITEM VALUES(NULL, 'Gold Bracelet', 210.00, 'Baker', 'Samantha');

INSERT INTO ITEM VALUES(NULL, 'Silver Necklace', 165.00, 'Baxter', 'Sam');

**PURCHASE:**

Table, Excel

Description automatically generated

INSERT INTO PURCHASE VALUES(NULL, '2019-05-05', 155.00, 1);

INSERT INTO PURCHASE VALUES(NULL, '2019-05-07', 203.00, 2);

INSERT INTO PURCHASE VALUES(NULL, '2019-05-11', 75.00, 3);

INSERT INTO PURCHASE VALUES(NULL, '2019-05-15', 67.00, 4);

INSERT INTO PURCHASE VALUES(NULL, '2019-05-15', 330.00, 5);

INSERT INTO PURCHASE VALUES(NULL, '2019-05-16', 25.00, 1);

INSERT INTO PURCHASE VALUES(NULL, '2019-05-25', 45.00, 3);

INSERT INTO PURCHASE VALUES(NULL, '2019-06-06', 445.00, 1);

INSERT INTO PURCHASE VALUES(NULL, '2019-06-07', 72.00, 6);

**INVOICE:**

Table

Description automatically generated

INSERT INTO PURCHASE\_ITEM VALUES(1001, 1, 1, 155.00);

INSERT INTO PURCHASE\_ITEM VALUES(1002, 1, 2, 203.00);

INSERT INTO PURCHASE\_ITEM VALUES(1003, 1, 3, 75.00);

INSERT INTO PURCHASE\_ITEM VALUES(1004, 1, 6, 35.00);

INSERT INTO PURCHASE\_ITEM VALUES(1004, 2, 7, 32.00);

INSERT INTO PURCHASE\_ITEM VALUES(1005, 1, 4, 240.00);

INSERT INTO PURCHASE\_ITEM VALUES(1005, 2, 8, 90.00);

INSERT INTO PURCHASE\_ITEM VALUES(1006, 1, 10, 25.00);

INSERT INTO PURCHASE\_ITEM VALUES(1007, 1, 11, 45.00);

INSERT INTO PURCHASE\_ITEM VALUES(1008, 1, 5, 175.00);

INSERT INTO PURCHASE\_ITEM VALUES(1008, 2, 9, 215.00);

INSERT INTO PURCHASE\_ITEM VALUES(1008, 3, 12, 55.00);

INSERT INTO PURCHASE\_ITEM VALUES(1009, 1, 14, 72.00);