Name(s)\_\_Ryder Dale Walton && Chance Turner\_\_

DB Homework 11

You may complete this with one partner.

Due Tues, Nov 29th

(20 pts)

Turn In: This document, with SQL statements copied below each question.

1. Refer to figure P8.19 on p. 421. Modify the Ch08\_LargeCo\_SQL script with constraints (where appropriate) to implement the following:

* Primary Keys
* Foreign Key (to avoid tedious hassles, do not create the FK constraints between employee and department)
* For the Invoice-Line FK relationship, create a Cascade Delete/Update

Syntax Examples for constraints: <https://msdn.microsoft.com/en-us/library/ms189049.aspx>

Create a new empty database in Sql Server and test your script. You will need to make some minor changes to get the PK and FK constraints to work:

* Make sure the primary key fields are specified as NOT NULL;
* Make sure the FK field and the PK it references have identical definitions (e.g., one can’t say NUMBER and the other say NUMBER(6,0)…the types must be identical.

CREATE TABLE lgbrand (

brand\_id numeric(4,0) NOT NULL,

brand\_name VARCHAR(100) NULL,

brand\_type VARCHAR(20) NULL,

CONSTRAINT PK\_LGBRAND PRIMARY KEY (BRAND\_ID)

);

CREATE TABLE lgproduct (

prod\_sku VARCHAR(15) NOT NULL,

prod\_descript VARCHAR(255) NULL,

prod\_type VARCHAR(255) NULL,

prod\_base VARCHAR(255) NULL,

prod\_category VARCHAR(255) NULL,

prod\_price NUMERIC(10,2) NULL,

prod\_qoh NUMERIC(10,0) NULL,

prod\_min NUMERIC(10,0) NULL,

brand\_id NUMERIC(4,0) NULL,

CONSTRAINT PK\_LGPRODUCT PRIMARY KEY (PROD\_SKU),

CONSTRAINT FK\_LGBRAND\_LGPRODUCT FOREIGN KEY (BRAND\_ID)

REFERENCES LGBRAND (BRAND\_ID)

ON DELETE CASCADE

ON UPDATE CASCADE

);

CREATE TABLE lgvendor (

vend\_id NUMERIC(6,0) NOT NULL,

vend\_name VARCHAR(255) NULL,

vend\_street VARCHAR(50) NULL,

vend\_city VARCHAR(50) NULL,

vend\_state VARCHAR(2) NULL,

vend\_zip VARCHAR(5) NULL,

CONSTRAINT PK\_LGVENDOR PRIMARY KEY (VEND\_ID)

);

CREATE TABLE lgsupplies (

prod\_sku VARCHAR(15) NOT NULL,

vend\_id NUMERIC(6,0) NOT NULL,

CONSTRAINT PK\_LGSUPPLIES PRIMARY KEY (PROD\_SKU, VEND\_ID),

CONSTRAINT FK\_LGSUPPLIES\_LGPRODUCT FOREIGN KEY (PROD\_SKU)

REFERENCES LGPRODUCT (PROD\_SKU)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT FK\_LGSUPPLIES\_LGVENDOR FOREIGN KEY (VEND\_ID)

REFERENCES LGVENDOR (VEND\_ID)

ON DELETE CASCADE

ON UPDATE CASCADE

);

CREATE TABLE lgdepartment (

dept\_num NUMERIC(5,0) NOT NULL,

dept\_name VARCHAR(50) NOT NULL,

dept\_mail\_box VARCHAR(3) NULL,

dept\_phone VARCHAR(9) NULL,

emp\_num NUMERIC(6,0) NULL,

CONSTRAINT PK\_LGDEPARTMENT PRIMARY KEY (DEPT\_NUM),

);

CREATE TABLE lgemployee (

emp\_num NUMERIC(6,0) NOT NULL,

emp\_fname VARCHAR(20) NULL,

emp\_lname VARCHAR(25) NOT NULL,

emp\_email VARCHAR(25) NOT NULL,

emp\_phone VARCHAR(20) NULL,

emp\_hiredate DATE NOT NULL,

emp\_title VARCHAR(45) NOT NULL,

emp\_comm NUMERIC(2,2) NULL,

dept\_num NUMERIC(5,0) NULL,

CONSTRAINT PK\_LGEMPLOYEE PRIMARY KEY (EMP\_NUM),

CONSTRAINT FK\_LGEMPLOYEE\_LGDEPARTMENT FOREIGN KEY (DEPT\_NUM)

REFERENCES LGDEPARTMENT (DEPT\_NUM)

);

CREATE TABLE lgsalary\_history (

emp\_num NUMERIC(6,0) NOT NULL,

sal\_from DATE NOT NULL,

sal\_end DATE NULL,

sal\_amount NUMERIC(10,2) NULL,

CONSTRAINT PK\_LGSALARYHISTORY PRIMARY KEY (EMP\_NUM, SAL\_FROM),

CONSTRAINT FK\_LGSALARYHISTORY\_LGEMPLOYEE FOREIGN KEY (EMP\_NUM)

REFERENCES LGEMPLOYEE (EMP\_NUM)

ON DELETE CASCADE

ON UPDATE CASCADE

);

CREATE TABLE lgcustomer (

cust\_code NUMERIC(38,0) NOT NULL,

cust\_fname VARCHAR(20) NOT NULL,

cust\_lname VARCHAR(20) NOT NULL,

cust\_street VARCHAR(70) NULL,

cust\_city VARCHAR(50) NULL,

cust\_state CHAR(2) NULL,

cust\_zip CHAR(5) NULL,

cust\_balance NUMERIC(8,2) NULL,

CONSTRAINT PK\_LGCUSTOMER PRIMARY KEY (CUST\_CODE)

);

CREATE TABLE lginvoice (

inv\_num NUMERIC(38,0) NOT NULL,

inv\_date DATE NULL,

cust\_code NUMERIC(38,0) NULL,

inv\_total NUMERIC(11,2) NULL,

employee\_id NUMERIC(6,0) NULL,

CONSTRAINT PK\_LGINVOICE PRIMARY KEY (INV\_NUM),

CONSTRAINT FK\_LGINVOICE\_LGCUSTOMER FOREIGN KEY (CUST\_CODE)

REFERENCES LGCUSTOMER (CUST\_CODE),

CONSTRAINT FK\_LGINVOICE\_LGEMPLOYEE FOREIGN KEY (EMPLOYEE\_ID)

REFERENCES LGEMPLOYEE (EMP\_NUM)

);

CREATE TABLE lgline (

inv\_num NUMERIC (38,0) NOT NULL,

line\_num NUMERIC NOT NULL,

prod\_sku VARCHAR(15) NULL,

line\_qty NUMERIC NULL,

line\_price NUMERIC(8,2) NULL,

CONSTRAINT PK\_LGLINE PRIMARY KEY (INV\_NUM, LINE\_NUM),

CONSTRAINT FK\_LGLINE\_LGINVOICE FOREIGN KEY (INV\_NUM)

REFERENCES LGINVOICE (INV\_NUM)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT FK\_LGLINE\_LGPRODUCT FOREIGN KEY (PROD\_SKU)

REFERENCES LGPRODUCT (PROD\_SKU)

);

--Only including the Create statements. Otherwise, this document would be over 630 pages long. We didn’t do anything with the inserts anyway.

Write sql server statements as needed to answer the following questions. Copy/paste the sql you use for each question below.

1. How many line items are associated with invoice 106?

SELECT COUNT(L.line\_num) AS Line\_Items

FROM lgline L

where L.inv\_num = 106;

1. Delete invoice 106, and verify that the associated line items are removed. Show the sql below.

DELETE FROM lginvoice

WHERE inv\_num = 106;

SELECT \* FROM lgline; --used this to verify they were gone

1. Show a sql statement below that will cause an error due to one of your foreign key constraints. Copy/paste the sql and the error message below.

DELETE FROM lgdepartment

WHERE dept\_num = 200;

Msg 547, Level 16, State 0, Line 1

The DELETE statement conflicted with the REFERENCE constraint "FK\_LGEMPLOYEE\_LGDEPARTMENT". The conflict occurred in database "Homework\_11", table "dbo.lgemployee", column 'dept\_num'.

The statement has been terminated.

1. Show a sql statement below that will cause an error due to one of your primary key constraints. Copy/paste the sql and the error message below.

INSERT INTO lgdepartment(dept\_num)

VALUES (200); -- got same error whether I inserted null or 200

Msg 515, Level 16, State 2, Line 1

Cannot insert the value NULL into column 'dept\_name', table 'Homework\_11.dbo.lgdepartment'; column does not allow nulls. INSERT fails.

The statement has been terminated.

1. Write a sql statement that will list all customer codes who have not purchased anything.

SELECT cust\_code

FROM lgcustomer

WHERE cust\_code NOT IN (SELECT cust\_code

FROM lginvoice);

1. Write a sql statement that will list all brands for which we do not have products.

SELECT \*

FROM lgbrand

WHERE brand\_id NOT IN (SELECT brand\_id

FROM lgproduct);

1. Refer to your results from the previous two problems. Write a console application that will use two sql delete statements to delete one of the inactive customers and one of the unused brands (you may hard-code the specific customer and brand to delete). This must use Dapper, and it must be done via two Execute statements. Copy/paste your source code below.

using System;

using Dapper;

using System.Data.SqlClient;

using System.Collections.Generic;

using System.Data;

using System.Linq;

namespace HW\_11

{

class Program

{

static void Main(string[] args)

{

//Connection String to the database as a whole.

const string CONNECTION\_STRING = @"Data Source = .\SQLEXPRESS;" +

@"Initial Catalog = Homework\_11;" +

@"Integrated Security = True;" +

@"Connect Timeout = 15;" +

@"Encrypt = False;" +

@"TrustServerCertificate = True;" +

@"ApplicationIntent = ReadWrite;" +

@"MultiSubnetFailover = False";

//Dapper Database Request to delete unused customer.

using (IDbConnection conn = new SqlConnection(CONNECTION\_STRING))

{

conn.Execute("DELETE FROM lgcustomer WHERE cust\_code = 393");

}

//Dapper Database Request to delete unused brand.

using (IDbConnection conn = new SqlConnection(CONNECTION\_STRING))

{

conn.Execute("DELETE FROM lgbrand WHERE brand\_id = 24");

}

}

}

} // Verified that this worked by checking with the queries from 6 and 7

1. Repeat the previous question, but *after* deleting an inactive customer, delete a brand that ***is*** referenced by a product (this should cause a run time error). Both deletes should be performed within the same transaction, as illustrated here: <http://stackoverflow.com/questions/24650710/proper-way-of-using-begintransaction-with-dapper-idbconnection>. You must use a try…catch to handle the error gracefully.
   1. Note: the Execute() statement accepts the transaction as an optional final parameter.
   2. Verify that *neither* table was modified by the deletion (this should be handled by the transaction).
   3. You need to wrap the transaction.Rollback() in *another* try/catch, since the rollback itself will fail if the transaction was already automatically rolled back by the database.
   4. Copy/paste your source code below, and insert a screen-shot of the error message produced by your program.

using System;

using Dapper;

using System.Data.SqlClient;

using System.Collections.Generic;

using System.Data;

using System.Linq;

namespace HW\_11

{

class Program

{

static void Main(string[] args)

{

//Connection String to the database as a whole.

const string CONNECTION\_STRING = @"Data Source = .\SQLEXPRESS;" +

@"Initial Catalog = Homework\_11;" +

@"Integrated Security = True;" +

@"Connect Timeout = 15;" +

@"Encrypt = False;" +

@"TrustServerCertificate = True;" +

@"ApplicationIntent = ReadWrite;" +

@"MultiSubnetFailover = False";

using(IDbConnection conn = new SqlConnection(CONNECTION\_STRING))

{

conn.Open();

using (var tran = conn.BeginTransaction())

{

try

{

// DB requests go here

conn.Execute("DELETE FROM lgcustomer WHERE cust\_code = 730", transaction: tran);

conn.Execute("DELETE FROM lgbrand WHERE brand\_id = 23", transaction: tran);

tran.Commit();

} catch (Exception Ex) //handle any exception

{

Console.WriteLine(Ex.Message);

try

{

tran.Rollback(); // If an exception is raised by the FK\_Constraint, then enter this block.

Console.WriteLine("Transaction rolled back.");

}

catch (SqlException SqlEx) // handle only rollback exception

{

Console.WriteLine(SqlEx.Message);

Console.WriteLine("The transaction was already rolled back by the DBMS.");

}

}

}

}

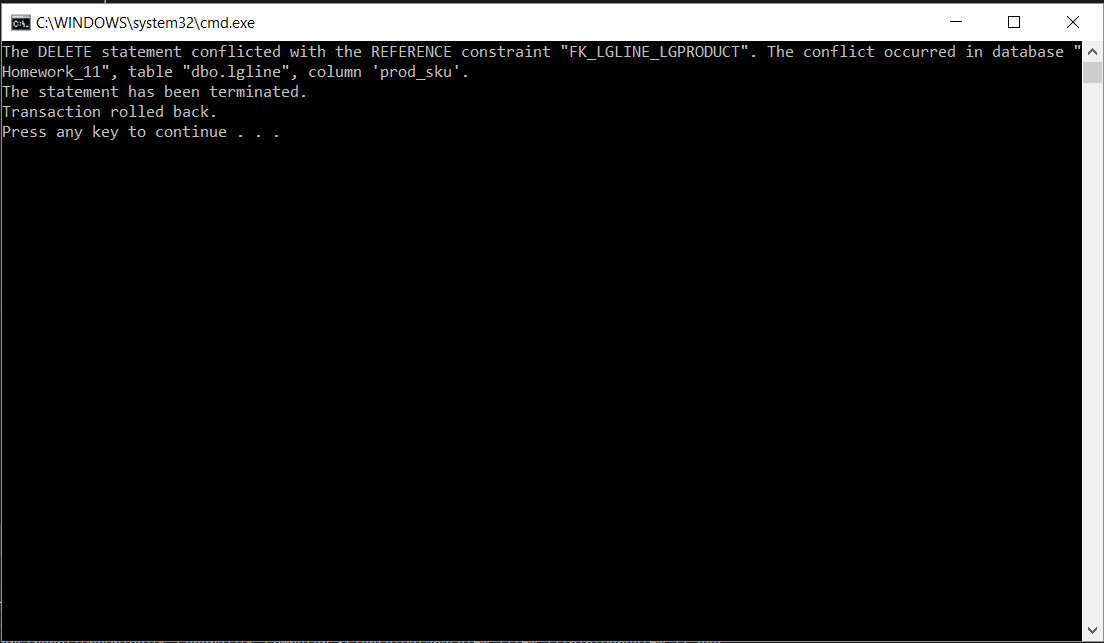
}

}

}

// Checked the Database using the 6 and 7 queries again. Neither were deleted.

// Screenshot



**Turn In:**

This document to bboard, containing sql and C# source code.