HW#5: Creating and Populating Tables in a Database using SQL Code

Overview

The purpose of this homework assignment is to help you learn how to create and populate tables with data using the SQL programming language. We will use the SQL Server 2012 database management system for this and the next three homework assignments. You will use the same tables for the remaining homework assignments (HW#6-8) so you must complete this homework assignment in order to do the remaining homework assignments.

The database you will create, populate and access represents part of the online customer ordering system for an organization called "Life's a Game," an online puzzle and game distributor. Life's a Game uses the Internet to distribute board games and puzzles directly to the public. The company specializes in games and puzzles that are challenging and a bit different from those available in brick and mortar distributors. The database is a transaction processing database to capture and store all current customer orders for the organization.

Your job for this homework assignment is to create nine tables in your existing database on SQL Server as described in this document and populate those tables with the data included on the pages following the table descriptions. **Please do not make up different data** – you must use the data and format for the data included in this document. I want everyone in class to be working from exactly the same test dataset.

Methods of Table Creation and Population

I will explain in class how to create and populate tables via SQL. Chapter 11 of your SQL text explains how to create tables, while Chapter 7 explains how to populate tables. You must use SQL CREATE commands stored in SQL files to create your tables. This will help you understand the use of constraints (especially the referential integrity constraint) within a database.

A simple, but somewhat tedious method of table population is to code SQL INSERT commands. You must write one INSERT statement for each row of each table that you wish to populate. If you save your INSERT statements in a script file (a script file is just a plain old ASCII file created via a text editor such as WordPad, NotePad or EditPad), then you will be able to re-populate your tables quickly and easily in case there are problems with the data in the future. Coding SQL INSERT statements will also allow you to easily create the same tables in more than one database if you are working as part of a team. I recommend that you use SQL INSERT statements. This could be made easier by using a product like Excel to take the tables in this document, add the appropriate INSERT statements, and remove any formatting.

It is OK to use the SQL Server data input screens to populate your tables. To do this, highlight and right click the table name in SQL Server 2012 Management Studio. Select "open table" from the window. You will then see a grid displayed on the screen for data entry. All data entered through this grid will be committed to disk when you do any other function via SQL Server. The problem with this approach is that you will NOT be able to quickly and easily rebuild your tables if they were built incorrectly.

Feel free to use whatever method you want to populate your tables - just make sure you understand what you are doing to populate the tables accurately.

Note: Inaccurate output on HW#'s 6-8 will be graded as wrong - even if the error is a result of a simple data input error on this assignment. You are responsible for the accuracy of your test data set!!

Assignment Issues

A few issues concerning this assignment are discussed below.

1. **Do not create a database.** We have created a database for you on the COB server – each student has his/her own database created automatically – so there is no need to issue a "create database" SQL command. For this assignment, you are creating tables in an existing database. Your starting place for this assignment is to issue "CREATE TABLE" commands in SQL to create each of the tables.

2. **Be careful with quotation marks.** Microsoft Word and other word processing packages use "smart quotes" which are pretty quotation marks that curl forwards and backwards. The SQL programming language does not like smart quotes – so check your work to make sure that the quotation marks are straight and not curved. SQL prefers single quotation marks for almost all of the syntax.

3. **Don't use spaces in table names or column (field) names.** Really. Don't do it. If you love the look of spaces, use underscores or dashes. Don't use blank spaces in names. Feel free to use upper and/or lower case in your table and field names. SQL is not case sensitive in table and field names.

4. **Plan how to populate the tables.** Data must be entered into parent tables before it is entered into child tables. Think about the concept of referential integrity and its implementation through foreign keys before populating the tables. An example of the concept is discussed below.

   The business rule for the relationship between customer and order is: An order is placed by an existing customer so the customer must be entered into the customer table before the order can be entered into the order table. The customer is the "parent" table and the order is the "child" table in this relationship. This business rule is established through a foreign key of the customerID in the order table that includes a referential integrity constraint. When a new order is entered in the order table (the child), the DBMS will check the customer table (the parent) to make sure the customer exists before permitting the entry into the order table. If the customer does not exist, the DBMS will not allow the order to be entered into the order table. This is a referential integrity constraint placed on the order table.

   An issue for a database designer or administrator is deciding the order that the tables must be created and populated in a database, since the tables are related to one another. A solution to this issue is to create the tables without referential integrity constraints, populate the tables, and then add the referential integrity constraints. A problem with that solution is that the DBMS will not validate the data as you populate the tables. For example, if you enter an order with an invalid customerID (let's say you make a typo), the DBMS will not be able to validate the customerID and will let you enter the invalid data. Thus, you are responsible for validating the data as entered if you create the tables without referential integrity constraints.

   Another solution is to differentiate parent tables and child tables by looking at the ERD. Sometimes it can be difficult to differentiate parent from child because a table may be a parent to some tables while also being a child to other tables. For example, the TblOrderLine table is a child table when related to the TblOrder and TblItem tables, but it is a parent table when related to the TblShipLine table. So the TblOrderLine table must be populated after the TblOrder and TblItem tables, but before the TblShipLine table. This will take planning and an understanding of referential integrity on your part, but it works!

5. **Do not include some of the formatting characters in data. Do not put parentheses or dashes in telephone numbers.** We are going to use the telephone number as a way to learn how to use the SQL substring (SUBSTRING) and (CHARINDEX) commands so please input the telephone number without any formatting characters. Do not put commas or dollar signs in data representing dollars. Always include decimal points where applicable in numeric data. A decimal point is not a formatting character; you must include the decimal point to get the correct numeric amount if there are digits (other than zeros) after the decimal point.

6. **Enter the capitalization exactly as shown.** Some of the data are upper case and others are lower. Please enter the data exactly as shown on this handout.

7. **Enter character fields with character strings.** Computers differentiate character strings from numbers, even if numeric characters are stored in a field declared as a character data type. For example, the OrderID in this database is a character field (char) type. If you enter a number in a character field (meaning that you enter the data without quotation marks surrounding it) then SQL Server will store it as a number. So if the OrderID is 00722 and you enter it as a number when using the INSERT command (without quotes) then SQL Server will store it as 722. If you enter it as a character string, '00722' then SQL Server will store the leading zeroes. Enter all character data with quotation marks, even if that data is numeric.
8. **Enter "blank" fields as null values.** A blank, space and null are **not** the same value. They are each stored as different characters in a database. Be sure to enter data using the word NULL when entering data that is shown as blank on the pages at the end of this narrative. The word NULL should not have quotations around it because it is a value rather than a word.
9. **Enter the data accurately.** I grade your query output based on how well it matches my generated output. If your answers do not match my output, then I will take off points even if the problem is purely a data entry error. Check your data and make sure it is accurate!
10. **Keep in mind that this assignment is the base for the next three assignments.** This is not a standalone assignment. You will use the database created and populated in this assignment to complete additional assignments.
11. **Each group member needs the tables built in his/her own database.** If you are working in a group, each group member should create and populate the tables in his/her own database on SQL Server. Each group member will need a copy of this database for the second exam, so create it now.

Deliverables

Please turn in the following paper-based deliverables for grading of HW#5:

1) **The "CREATE TABLE" statements for each table.**

2) **A discussion of how you populated the tables** (examples: "Used the data input mode through SQL Server", or "converted the Word tables to Excel and used Excel concatenate function to add the INSERT statements and quotation marks"). Do NOT turn in the SQL INSERT statements.

3) **A result table for each table.** Do a "SELECT * from table_name" in SQL for each table in the database. Example:

```
SELECT      *
FROM        tblOrder;
```

4) **A time sheet** listing time spent on the assignment by date. List time in half-hour increments. For example, if you worked on the assignment on 02/22/2016 for four and a half hours, then list the date and 4.5 hours on the time sheet. A blank time sheet is provided on the next page. If you are working in a group, each group member will turn in an individual time sheet. Please don't inflate or deflate the amount of time spent working – it won't affect your grade. I'm trying to get an accurate picture of how much time is necessary to complete each assignment. I'm also trying to understand if it takes more or less time to work in a team on these assignments.

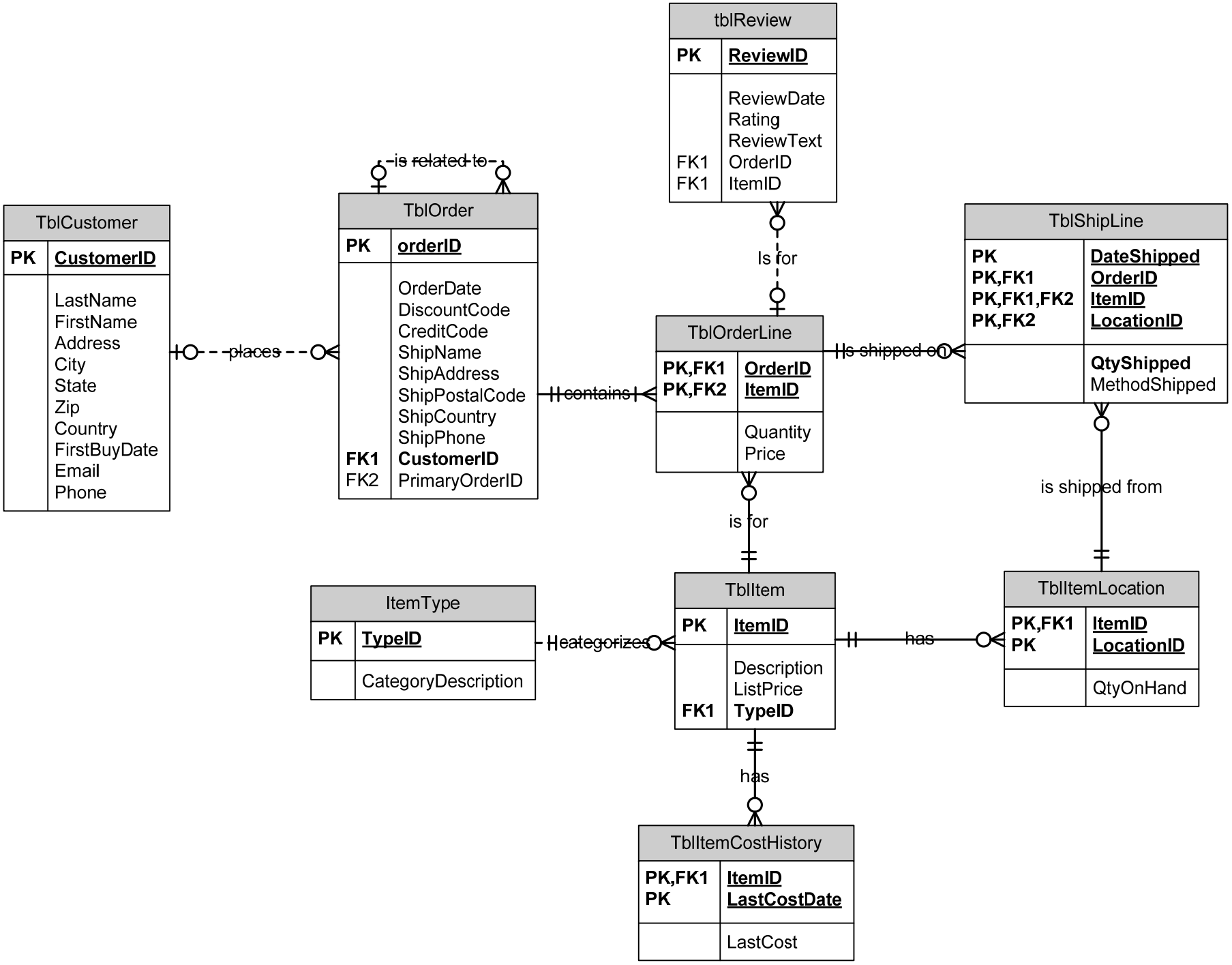
There is no need to turn in any computer-based output for HW#5. Print a paper-based copy of the above 1-4. I prefer that you simply clip the pages together with a paper clip and put all the pages in your envelope.

IS 475/675 HW#5 Time Sheet

Name:

Date	Hours		Date	Hours

Total Time Spent =


tblReview
PK ReviewID
ReviewDate
Rating
ReviewText
FK1 OrderID
FK1 ItemID
is related to
TblCustomer
PK CustomerID
LastName
FirstName
Address
City
State
Zip
Country
FirstBuyDate
Email
Phone
places
TblOrder
PK orderID
OrderDate
DiscountCode
CreditCode
ShipName
ShipAddress
ShipPostalCode
ShipCountry
ShipPhone
FK1 CustomerID
FK2 PrimaryOrderID
contains
Is for
TblOrderLine
PK,FK1 OrderID
PK,FK2 ItemID
Quantity
Price
Is shipped on
TblShipLine
PK DateShipped
PK,FK1 OrderID
PK,FK1,FK2 ItemID
PK,FK2 LocationID
QtyShipped
MethodShipped
is shipped from
is for
ItemType
PK TypeID
CategoryDescription
categorizes
TblItem
PK ItemID
Description
ListPrice
FK1 TypeID
has
TblItemLocation
PK,FK1 ItemID
PK LocationID
QtyOnHand
has
TblItemCostHistory
PK,FK1 ItemID
PK LastCostDate
LastCost

Structures for Tables

Table: TblCustomer				
Attribute Name	Data Type & Size	Primary Key	Foreign Key	Other Constraints
CustomerID	char(5)	yes	no	no null value
LastName	varchar(30)	no	no	no null value
FirstName	varchar(20)	no	no	null value OK
Address	varchar(30)	no	no	no null value
City	varchar(20)	no	no	no null value
State	char(2)	no	no	no null value
Zip	varchar(12)	no	no	no null value
Country	varchar(15)	no	no	null value OK
FirstBuyDate	datetime	no	no	null value OK
Email	varchar(60)	no	no	null value OK
Phone	char(15)	no	no	no null value

Table: TblOrder				
Attribute Name	Data Type & Size	Primary Key	Foreign Key	Other Constraints
OrderID	char(6)	yes	no	no null value
OrderDate	datetime	no	no	no null value
DiscountCode	char(2)	no	no	null value OK valid discount code include 02, 03, 04, 06, 08, 10, A1 and B3
CreditCode	char(3)	no	no	no null value
ShipName	varchar(30)	no	no	null value OK
ShipAddress	varchar(30)	no	no	null value OK
ShipZip	varchar(12)	no	no	null value OK
ShipCountry	varchar(30)	no	no	null value OK
ShipPhone	char(15)	no	no	null value OK
CustomerID	char(5)	no	yes - reference tblcustomer	no null value
PrimaryOrderID	Char(6)	NO	Yes – reference tblorder	Null value OK

Table: tblItem				
Attribute Name	Data Type & Size	Primary Key	Foreign Key	Other Constraints
ItemID	char(6)	yes	no	no null value
Description	varchar(300)	no	no	null value OK
ListPrice	money	no	no	no null value listprice must be greater than $5.
TypeID	Int	No	Yes – references tblItemType	No null value

Table: tblOrderLine				
Attribute Name	Data Type & Size	Primary Key	Foreign Key	Other Constraints
OrderID	char(6)	yes	yes – reference TblOrder	no null value
ItemID	char(6)	yes	yes – reference TblItem	no null value
Quantity	int	no	no	no null value, value must be greater than 0.
Price	money	no	no	no null value, value must be greater than 0.

Table: TblShipLine				
Attribute Name	Data Type & Size	Primary Key	Foreign Key	Other Constraints
DateShipped	datetime	yes	no	no null value
OrderID	char(6)	yes	yes – reference TblOrderLine	no null value
ItemID	char(6)	yes	yes – reference TblOrderLine yes – reference TblItemLocation	no null value
LocationID	char(2)	yes	yes – reference TblItemLocation	no null value
QtyShipped	int	no	no	no null value
MethodShipped	varchar(30)	no	no	no null value

Table: TblItemLocation				
Attribute Name	Data Type & Size	Primary Key	Foreign Key	Other Constraints
ItemID	char(6)	yes	yes – reference TblItem	no null value
LocationID	char(2)	yes	no	no null value
QtyOnHand	int	no	no	null value OK

Table: tblItemCostHistory				
Attribute Name	Data Type & Size	Primary Key	Foreign Key	Other Constraints
ItemID	char(6)	yes	yes – reference tblItem	no null value
LastCostDate	datetime	yes	no	no null value
LastCost	money	no	no	no null value

Table: tblItemType				
Attribute Name	Data Type & Size	Primary Key	Foreign Key	Other Constraints
TypeID	Int	yes	no	no null value
CategoryDescription	Varchar(50)	No	no	Null value OK

Table: tblReview				
Attribute Name	Data Type & Size	Primary Key	Foreign Key	Other Constraints
ReviewID	Int – this should be an identity field that will automatically increment	yes	no	no null value
ReviewDate	datetime	No	no	Null value OK
Rating	Int	No	No	Rating should be a value of 1,2,3,4, or 5
ReviewText	Varchar(500)	no	no	Null value OK
OrderID	Char(6)	No	Yes – references tblorderline	No null value
ItemID	Char(6)	No	Yes – references tblorderline	No null value

Data for Table Population

TblItem

tblItem			
itemID	**Description**	**Listprice**	**typeid**
A23441	New York City Monopoly Game Collector's Edition	$29.95	10
A23771	Mysterium	$132.99	12
A34665	Boggle Deluxe 5x5	$34.95	10
A34882	Perudo	$10.95	10
A45111	How to Host a Murder - An Affair to Dismember	$28.95	15
B67123	Tiny Epic Galaxies	$39.99	12
B67466	Diplomacy: Game of Negotiation, Cunning and Deceit.	$43.95	20
B78244	Code Names	$19.95	15
B78500	Pandemic Legacy	$59.99	12
C26133	Knowledge Management: Create a Learning Organization	$395.95	20
C29179	Managing Change: The Game for an Executive Retreat	$259.95	20
C34122	A Game of Strategy, Negotiation and Excitement for Office Retreats	$169.95	20

tblItemType

tblItemType	
typeid	**CategoryDescription**
10	Classic Board Games
11	Word Games
12	Science Fiction and Fantasy
15	Mystery and Thrillers
16	Romance
17	Comedy and Jokes
18	Risque and Adult Material
20	Office and Team Building

TBLCUSTOMER

tblcustomer										
customerid	lastname	firstname	address	city	state	zip	country	firstbuydate	email	phone
00405	Barrington	Margaret	1765 Roundtree Pkwy	reno	nv	89509-1454	USA	7/12/2006	barry@hotmail.com	7757464561
00625	Dao	Phong	341 West Park	Fresno	CA	96137		9/2/2015	daop@gmail.com	8582138982
06774	Phillips	Kendall	44512 Sawbuck Path	SPARKS	nv	89432	USA	8/12/2000	flipper@gmail.com	7753324636
07831	Rodriguez	Karen	4589 Marthiam	Chico	CA	97111		11/6/2008	rodriguez@aol.com	8193821828
08892	Twillers	Bethany	P.O.Box 5661	san jose	ca	98123	USA	4/1/2000	twillbeth@yahoo.com	8098291838
12001	Cranston	Brittany	12 Sandstone	Sparks	NV	89431		4/12/2014	britters@ccb.com	7753312199
12006	Martinez	Guadalupe	223 North Pinetree Drive	Reno	NV	89511		2/14/2012	gmartinez@ccu.edu	7758837612
21142	Candriller	Kathy	2 Sedgeway	Laguna Beach	CA	94567		12/11/2015	kriller@mbu.edu	6198813929
21143	Jackson	Janice	2341 Bramble Bush Drive	Sparks	NV	89431-0112		5/6/2014	jj@isp.all.com	7753317188
29188	Polanski	Tiffany	5778 Battlemount Ct.	RENO	NV	89507		8/23/2011	skipole@gmail.com	7757465771
30192	Chen	Lian	2319 Crest Dr	REno	Nv	89503-0113		8/30/1999	jester@here.com	7757218991
32018	Jones	Martin	10 South Wilders	reno	nv	89503-8912	USA	2/12/2016	mmm@hard.com	7753314838
32817	Foster	Ben	318 Western Ave	SAN diego	ca	92381		8/15/1999	fosterben@aol.com	8583284483
38817	Argiento	Bud	1001 Catchway	Anaheim	CA	95113		3/12/2015	bud@cservices.com	8037718991
78112	Guili	Mary Anne	4457 Meridith St	Irvine	CA	97128		10/9/2012	ma@san.rr.com	6195621334

TBLORDER

tblorder										
orderID	orderdate	discountcode	creditcode	shipname	shipaddress	shipzip	shipcountry	shipphone	customerid	primaryorderid
123000	2/2/2016	A1	111						00405	
200335	1/26/2016		111	Cordwin	Arch 162 Stamford Brook	W6 0SE	UK	441817417500	07831	567123
223344	2/9/2016		231						21142	
300221	1/26/2016	03	231	Baron Mancos	251 Western Avenue Suite 1a	08776	USA	2109003005	07831	567123
400001	2/20/2016	B3	111						32018	
445511	2/15/2016	02	444						32018	
450137	12/29/2015		444	Frandsen LLC	435 Caminito Corriente	92129	USA	6551223298	07831	
567123	1/26/2016		444	Jenkins Corporation	2276 Brentell Street Suite 201	92128	USA	8583440669	07831	
651222	1/29/2016		111						12006	
671100	2/19/2016		111						32018	
675990	12/28/2015		111	Karen Nelson	601 Comet View Ct.	90056	USA	7018902330	00625	892211
781206	2/15/2016	06	231	Carrington-Smythe	231 Dulwich St. Wellington	TA21 0AB	UK	02088887009	38817	
892211	12/28/2015		111						00625	
980001	1/22/2016	04	444						78112	
983983	2/19/2016		111	Ender Industries	3011 Marsh Dr.	67455	USA	3558992111	32018	671100

NOTE: Blank data should be entered as NULL values in the database. Do not put quotation marks around the word NULL.

TBLORDERLINE

tblOrderLine			
orderID	itemID	quantity	price
123000	A23441	8	$29.95
123000	A34665	30	$37.95
123000	B67123	5	$389.99
200335	A23441	1	$29.95
200335	A34665	1	$34.95
200335	B67123	1	$39.99
200335	B67466	1	$43.95
223344	A23441	55	$29.95
223344	A23771	15	$122.99
223344	A34665	100	$23.95
223344	A34882	35	$7.95
223344	B67123	25	$34.95
223344	B67466	15	$40.95
300221	A23771	1	$145.99
300221	A34665	1	$35.95
300221	A34882	1	$10.95
300221	B67123	1	$39.99
300221	B78244	1	$19.95
400001	C26133	1	$395.95
445511	C34122	3	$269.95
450137	A23771	16	$135.99
450137	A34665	10	$31.00
450137	A34882	50	$9.95
450137	B67123	21	$14.95
450137	B67466	9	$41.95
450137	C26133	4	$398.95
450137	C34122	6	$167.95
567123	C26133	1	$395.95
651222	A34665	5	$37.95
651222	A34882	16	$11.95
651222	B78244	21	$17.99
671100	C29179	1	$259.95
781206	B67466	1	$43.95
781206	C29179	2	$280.00
892211	C26133	15	$380.00
892211	C29179	10	$259.95
892211	C34122	8	$200.00
980001	C29179	3	$275.99
980001	C34122	2	$169.95
983983	A23771	1	$135.99
983983	B78244	18	$18.95

TBLITEMLOCATION

tblItemLocation		
itemID	locationID	qtyonhand
A23441	10	11
A23441	20	23
A23441	30	25
A23771	10	6
A23771	20	4
A23771	30	5
A34665	10	141
A34882	10	40
A34882	30	55
A45111	10	2
A45111	20	3
A45111	30	2
B67123	10	22
B67123	20	28
B67123	30	6
B67123	40	4
B67466	10	0
B67466	20	3
B67466	30	4
B78244	20	22
B78500	20	8
B78500	30	1
C29179	10	15
C29179	20	15
C34122	10	16
C34122	20	21

TBLSHIPLINE (continued on two pages)

tblShipLine					
DateShipped	orderID	itemID	locationID	qtyshipped	methodshipped
1/19/2016	450137	B67123	20	8	ups
1/23/2016	892211	C29179	10	6	UPS
1/23/2016	892211	C34122	10	2	UPS
1/23/2016	892211	C34122	20	3	UPS
1/27/2016	450137	B67123	20	3	UPS
1/27/2016	450137	B67123	30	14	Ups
1/27/2016	450137	B67123	40	4	ups
2/5/2016	300221	A23771	20	1	Ups
2/5/2016	300221	B78244	20	1	UPS
2/5/2016	980001	C34122	10	2	FEDEX
2/6/2016	980001	C29179	10	3	FEDex
2/10/2016	223344	A23441	10	3	UPS
2/10/2016	223344	A23441	20	20	UPS
2/10/2016	223344	A23771	10	8	ups
2/10/2016	223344	A23771	20	2	UPS
2/10/2016	223344	A34665	10	45	UPs
2/10/2016	223344	A34882	10	15	fedEX
2/10/2016	223344	B67123	10	13	UPS
2/10/2016	300221	A34665	10	1	UPS
2/11/2016	223344	A23771	30	5	UPS
2/12/2016	223344	A23441	20	5	Ups
2/12/2016	223344	A34665	10	65	ups
2/12/2016	223344	A34882	10	20	UPS
2/12/2016	300221	A34882	10	1	UPs
2/15/2016	123000	A23441	20	8	UPS
2/15/2016	123000	A34665	10	32	UPS
2/16/2016	445511	C34122	10	3	FeDEX
2/19/2016	200335	A23441	20	6	UPS
2/19/2016	781206	B67466	30	4	Delta Freight
2/20/2016	300221	B67123	10	1	UpS
2/21/2016	200335	B67466	10	1	Delta Freight
2/21/2016	450137	A34665	10	10	FeDeX
2/22/2016	223344	B67123	10	12	UPS
2/22/2016	651222	A34882	30	32	UPS
2/22/2016	892211	C29179	10	4	UPS
2/25/2016	123000	B67123	10	5	UPS
2/25/2016	671100	C29179	10	1	UPS
2/29/2016	651222	A34882	30	10	UPS

tblShipLine					
DateShipped	orderID	itemID	locationID	qtyshipped	methodshipped
3/1/2016	223344	A23771	30	30	UPS
3/1/2016	450137	B67466	20	9	FedEx
3/1/2016	651222	A34882	30	10	UPS
3/2/2016	450137	A34882	30	15	UpS
3/2/2016	651222	A34665	10	2	UPS
3/2/2016	651222	A34882	10	5	UPS
3/4/2016	651222	A34665	10	3	UPS
3/5/2016	651222	A34882	10	10	FedEX
3/6/2016	651222	B78244	20	21	ups
3/19/2016	651222	A34882	10	1	UPS

TBLReview

dbo_tblreview					
reviewID	reviewdate	rating	reviewtext	orderid	itemid
	2/12/2016	1	This is the worst company I've ever dealt with. They still have not shipped my order and it was placed on January 26. I will never buy anything from them again. Ever.	200335	A34665
	2/14/2016	1	I don't know how this company stays in business. They can't ship anything on time, and their game selection isn't all that great	450137	C34122
	2/18/2016	5	Shipping was fast, the game was in great shape, and I will order from them again. Highly recommend	300221	A23771
	2/16/2016	4	This is one of the few companies that will ship internationally fairly cost effectively. They are slow, but at least shipping isn't a ridiculous amount.	223344	A23771
	1/15/2016	3	They are slow. They overship. They undership. They have the best shipping internationally, but nothing spectacular if you are shipping to the U.S. Buyer beware.	445511	C34122

TBLITEMCOSTHISTORY

tblItemCostHistory		
itemID	lastcostdate	lastcost
A23441	1/6/2013	$10.25
A23441	5/12/2013	$10.50
A23441	9/23/2014	$10.88
A23441	7/7/2015	$11.15
A23441	1/5/2016	$10.35
A23441	2/2/2016	$12.50
A23771	7/23/2013	$8.50
A34665	1/12/2015	$15.00
A34665	7/15/2015	$14.50
A34665	12/28/2015	$14.35
A34882	2/19/2014	$6.50
A34882	2/5/2016	$12.50
B67123	8/8/2015	$14.50
B67123	10/21/2015	$18.80
B67123	1/10/2016	$14.50
B67123	2/2/2016	$21.80
B67466	7/7/2014	$22.50
B67466	6/28/2015	$28.90
B67466	9/12/2015	$31.00
B78244	7/15/2015	$14.50
B78500	3/21/2015	$18.95
B78500	7/15/2015	$18.00
B78500	11/25/2015	$21.00
B78500	2/10/2016	$22.35
C26133	7/13/2014	$200.00
C26133	7/23/2015	$225.00
C26133	9/18/2015	$215.00
C26133	2/15/2016	$212.25
C29179	8/7/2015	$160.00
C29179	1/17/2016	$158.85
C34122	2/6/2013	$45.00
C34122	2/6/2014	$48.00
C34122	2/6/2015	$51.00
C34122	2/2/2016	$58.50