Team 4 - Ricky Genz | David Rhodes

IS 475/675 - HW #8

1. Code

```
Update     tblOrder
Set        OrderDate = '1/15/2016'
Where      OrderID = '567123';

Update     tblOrder
Set        OrderDate = '2/10/2016'
Where      OrderID = '671100';

Update     tblShipLine
Set        MethodShipped = LOWER(MethodShipped);

Update     tblCustomer
Set        State = UPPER(State);

Delete     From tblOrder
Where      OrderID Not In (Select   ol.OrderID
                           From     tblOrderLine ol);
```

1. Output

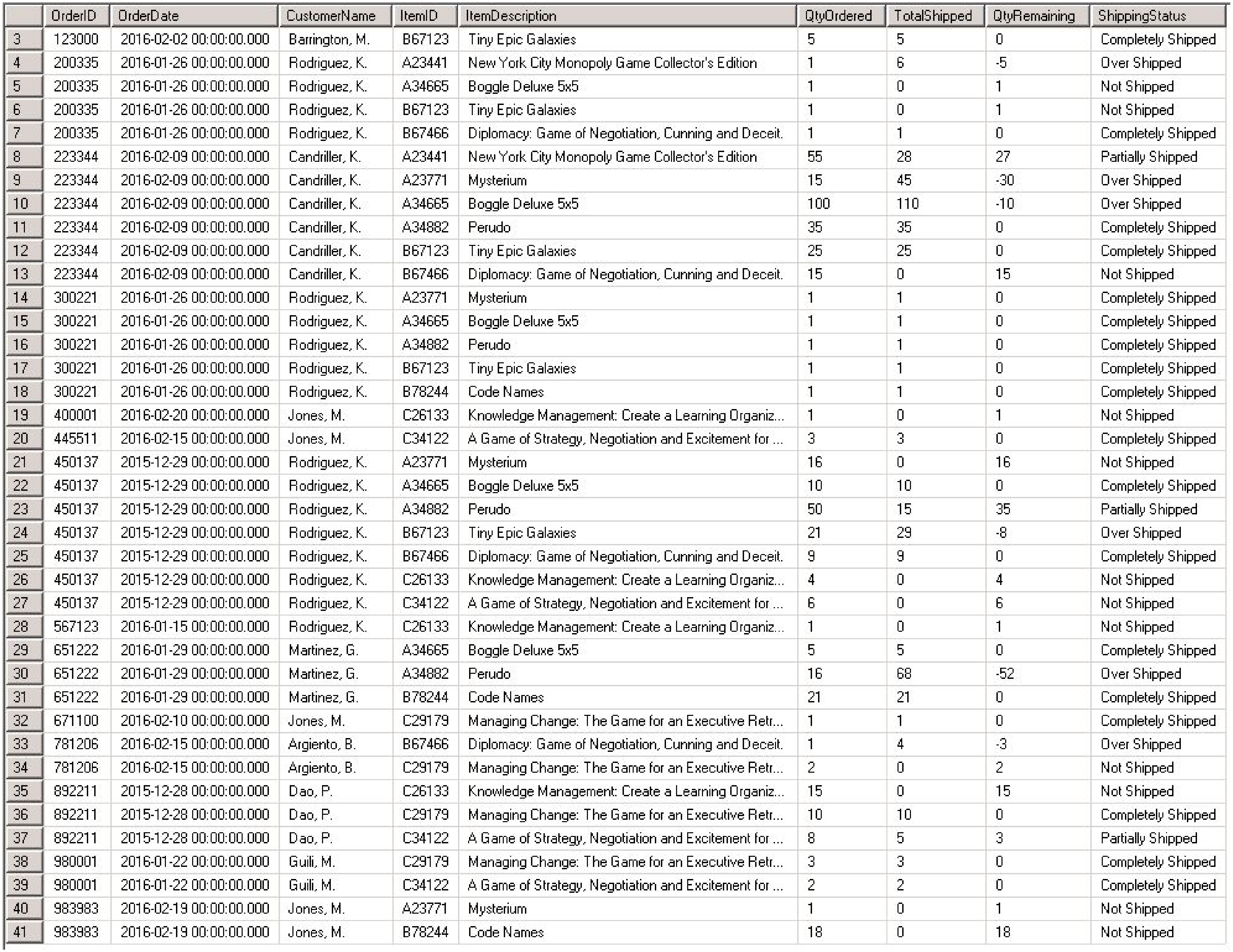
No Ouput

2. Code

```
Create ViewvShipTotal As
    Select          OrderID,
                    ItemID,
                    SUM(QtyShipped) TotalShipped
    From            tblShipLine
    Group By        OrderID, ItemID;

Select                      ol.OrderID,
                            ord.OrderDate,
                            (cus.LastName + ', ' + SUBSTRING(cus.FirstName,1,1) + '.') CustomerName,
                            ol.ItemID,
                            ite.Description ItemDescription,
                            ol.Quantity QtyOrdered,
                            ISNULL(vst.TotalShipped, 0) TotalShipped,
                            (ol.Quantity - ISNULL(vst.TotalShipped, 0)) QtyRemaining,
                            Case
                                    When ISNULL(vst.TotalShipped, 0) = 0 Then 'Not Shipped'
                                    When (ol.Quantity - ISNULL(vst.TotalShipped, 0)) = 0 Then 'Completely Shipped'
                                    When (ol.Quantity - ISNULL(vst.TotalShipped, 0)) < 0 Then 'Over Shipped'
                                    When (ol.Quantity - ISNULL(vst.TotalShipped, 0)) > 0 Then 'Partially Shipped'
                                    Else 'Unknown'
                            End ShippingStatus
From                        tblOrderLine ol
Left Outer Join     vShipTotal vst
On                          ol.OrderID = vst.OrderID And ol.ItemID = vst.ItemID
Inner Join          tblOrder ord
On                          ol.OrderID = ord.OrderID
Inner Join          tblCustomer cus
On                          ord.CustomerID = cus.CustomerID
Inner Join          tblItem ite
On                          ol.ItemID = ite.ItemID
Order By            ol.OrderID, ol.ItemID;
```

2. Output

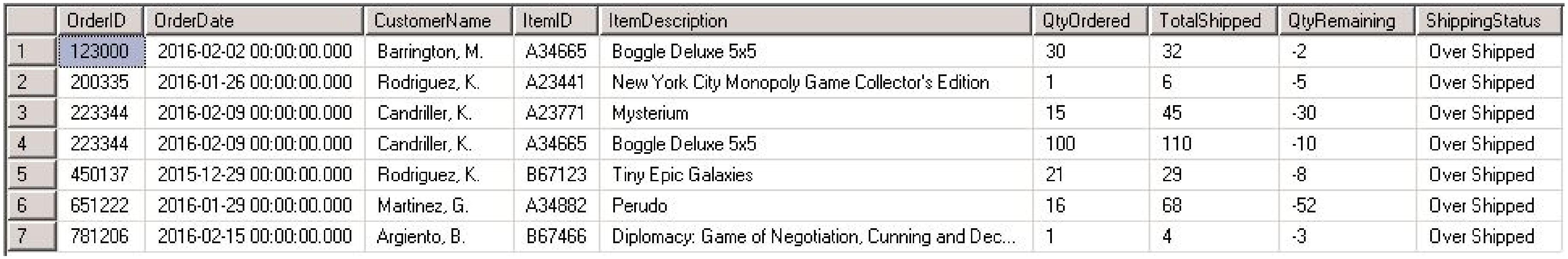


	OrderID	OrderDate	CustomerName	ItemID	ItemDescription	QtyOrdered	TotalShipped	QtyRemaining	ShippingStatus
3	123000	2016-02-02 00:00:00.000	Barrington, M.	B67123	Tiny Epic Galaxies	5	5	0	Completely Shipped
4	200335	2016-01-26 00:00:00.000	Rodriguez, K.	A23441	New York City Monopoly Game Collector's Edition	1	6	-5	Over Shipped
5	200335	2016-01-26 00:00:00.000	Rodriguez, K.	A34665	Boggle Deluxe 5x5	1	0	1	Not Shipped
6	200335	2016-01-26 00:00:00.000	Rodriguez, K.	B67123	Tiny Epic Galaxies	1	0	1	Not Shipped
7	200335	2016-01-26 00:00:00.000	Rodriguez, K.	B67466	Diplomacy: Game of Negotiation, Cunning and Deceit.	1	1	0	Completely Shipped
8	223344	2016-02-09 00:00:00.000	Candriller, K.	A23441	New York City Monopoly Game Collector's Edition	55	28	27	Partially Shipped
9	223344	2016-02-09 00:00:00.000	Candriller, K.	A23771	Mysterium	15	45	-30	Over Shipped
10	223344	2016-02-09 00:00:00.000	Candriller, K.	A34665	Boggle Deluxe 5x5	100	110	-10	Over Shipped
11	223344	2016-02-09 00:00:00.000	Candriller, K.	A34882	Perudo	35	35	0	Completely Shipped
12	223344	2016-02-09 00:00:00.000	Candriller, K.	B67123	Tiny Epic Galaxies	25	25	0	Completely Shipped
13	223344	2016-02-09 00:00:00.000	Candriller, K.	B67466	Diplomacy: Game of Negotiation, Cunning and Deceit.	15	0	15	Not Shipped
14	300221	2016-01-26 00:00:00.000	Rodriguez, K.	A23771	Mysterium	1	1	0	Completely Shipped
15	300221	2016-01-26 00:00:00.000	Rodriguez, K.	A34665	Boggle Deluxe 5x5	1	1	0	Completely Shipped
16	300221	2016-01-26 00:00:00.000	Rodriguez, K.	A34882	Perudo	1	1	0	Completely Shipped
17	300221	2016-01-26 00:00:00.000	Rodriguez, K.	B67123	Tiny Epic Galaxies	1	1	0	Completely Shipped
18	300221	2016-01-26 00:00:00.000	Rodriguez, K.	B78244	Code Names	1	1	0	Completely Shipped
19	400001	2016-02-20 00:00:00.000	Jones, M.	C26133	Knowledge Management: Create a Learning Organiz...	1	0	1	Not Shipped
20	445511	2016-02-15 00:00:00.000	Jones, M.	C34122	A Game of Strategy, Negotiation and Excitement for ...	3	3	0	Completely Shipped
21	450137	2015-12-29 00:00:00.000	Rodriguez, K.	A23771	Mysterium	16	0	16	Not Shipped
22	450137	2015-12-29 00:00:00.000	Rodriguez, K.	A34665	Boggle Deluxe 5x5	10	10	0	Completely Shipped
23	450137	2015-12-29 00:00:00.000	Rodriguez, K.	A34882	Perudo	50	15	35	Partially Shipped
24	450137	2015-12-29 00:00:00.000	Rodriguez, K.	B67123	Tiny Epic Galaxies	21	29	-8	Over Shipped
25	450137	2015-12-29 00:00:00.000	Rodriguez, K.	B67466	Diplomacy: Game of Negotiation, Cunning and Deceit.	9	9	0	Completely Shipped
26	450137	2015-12-29 00:00:00.000	Rodriguez, K.	C26133	Knowledge Management: Create a Learning Organiz...	4	0	4	Not Shipped
27	450137	2015-12-29 00:00:00.000	Rodriguez, K.	C34122	A Game of Strategy, Negotiation and Excitement for ...	6	0	6	Not Shipped
28	567123	2016-01-15 00:00:00.000	Rodriguez, K.	C26133	Knowledge Management: Create a Learning Organiz...	1	0	1	Not Shipped
29	651222	2016-01-29 00:00:00.000	Martinez, G.	A34665	Boggle Deluxe 5x5	5	5	0	Completely Shipped
30	651222	2016-01-29 00:00:00.000	Martinez, G.	A34882	Perudo	16	68	-52	Over Shipped
31	651222	2016-01-29 00:00:00.000	Martinez, G.	B78244	Code Names	21	21	0	Completely Shipped
32	671100	2016-02-10 00:00:00.000	Jones, M.	C29179	Managing Change: The Game for an Executive Retr...	1	1	0	Completely Shipped
33	781206	2016-02-15 00:00:00.000	Argiento, B.	B67466	Diplomacy: Game of Negotiation, Cunning and Deceit.	1	4	-3	Over Shipped
34	781206	2016-02-15 00:00:00.000	Argiento, B.	C29179	Managing Change: The Game for an Executive Retr...	2	0	2	Not Shipped
35	892211	2015-12-28 00:00:00.000	Dao, P.	C26133	Knowledge Management: Create a Learning Organiz...	15	0	15	Not Shipped
36	892211	2015-12-28 00:00:00.000	Dao, P.	C29179	Managing Change: The Game for an Executive Retr...	10	10	0	Completely Shipped
37	892211	2015-12-28 00:00:00.000	Dao, P.	C34122	A Game of Strategy, Negotiation and Excitement for ...	8	5	3	Partially Shipped
38	980001	2016-01-22 00:00:00.000	Guili, M.	C29179	Managing Change: The Game for an Executive Retr...	3	3	0	Completely Shipped
39	980001	2016-01-22 00:00:00.000	Guili, M.	C34122	A Game of Strategy, Negotiation and Excitement for ...	2	2	0	Completely Shipped
40	983983	2016-02-19 00:00:00.000	Jones, M.	A23771	Mysterium	1	0	1	Not Shipped
41	983983	2016-02-19 00:00:00.000	Jones, M.	B78244	Code Names	18	0	18	Not Shipped

3. Code

```
Select                      ol.OrderID,
                            ord.OrderDate,
                            (cus.LastName + ', ' + SUBSTRING(cus.FirstName,1,1) + '.') CustomerName,
                            ol.ItemID,
                            ite.Description ItemDescription,
                            ol.Quantity QtyOrdered,
                            ISNULL(vst.TotalShipped, 0) TotalShipped,
                            (ol.Quantity - ISNULL(vst.TotalShipped, 0)) QtyRemaining,
                            Case
                                    When ISNULL(vst.TotalShipped, 0) = 0 Then 'Not Shipped'
                                    When (ol.Quantity - ISNULL(vst.TotalShipped, 0)) = 0 Then 'Completely Shipped'
                                    When (ol.Quantity - ISNULL(vst.TotalShipped, 0)) < 0 Then 'Over Shipped'
                                    When (ol.Quantity - ISNULL(vst.TotalShipped, 0)) > 0 Then 'Partially Shipped'
                                    Else 'Unknown'
                            End ShippingStatus
From                        tblOrderLine ol
Left Outer Join     vShipTotal vst
On                          ol.OrderID = vst.OrderID And ol.ItemID = vst.ItemID
Inner Join          tblOrder ord
On                          ol.OrderID = ord.OrderID
Inner Join          tblCustomer cus
On                          ord.CustomerID = cus.CustomerID
Inner Join          tblItem ite
On                          ol.ItemID = ite.ItemID
Where                       (ol.Quantity - ISNULL(vst.TotalShipped, 0)) < 0
Order By            ol.OrderID, ol.ItemID;
```

3. Output



	OrderID	OrderDate	CustomerName	ItemID	ItemDescription	QtyOrdered	TotalShipped	QtyRemaining	ShippingStatus
1	123000	2016-02-02 00:00:00.000	Barrington, M.	A34665	Boggle Deluxe 5x5	30	32	-2	Over Shipped
2	200335	2016-01-26 00:00:00.000	Rodriguez, K.	A23441	New York City Monopoly Game Collector's Edition	1	6	-5	Over Shipped
3	223344	2016-02-09 00:00:00.000	Candriller, K.	A23771	Mysterium	15	45	-30	Over Shipped
4	223344	2016-02-09 00:00:00.000	Candriller, K.	A34665	Boggle Deluxe 5x5	100	110	-10	Over Shipped
5	450137	2015-12-29 00:00:00.000	Rodriguez, K.	B67123	Tiny Epic Galaxies	21	29	-8	Over Shipped
6	651222	2016-01-29 00:00:00.000	Martinez, G.	A34882	Perudo	16	68	-52	Over Shipped
7	781206	2016-02-15 00:00:00.000	Argiento, B.	B67466	Diplomacy: Game of Negotiation, Cunning and Dec...	1	4	-3	Over Shipped

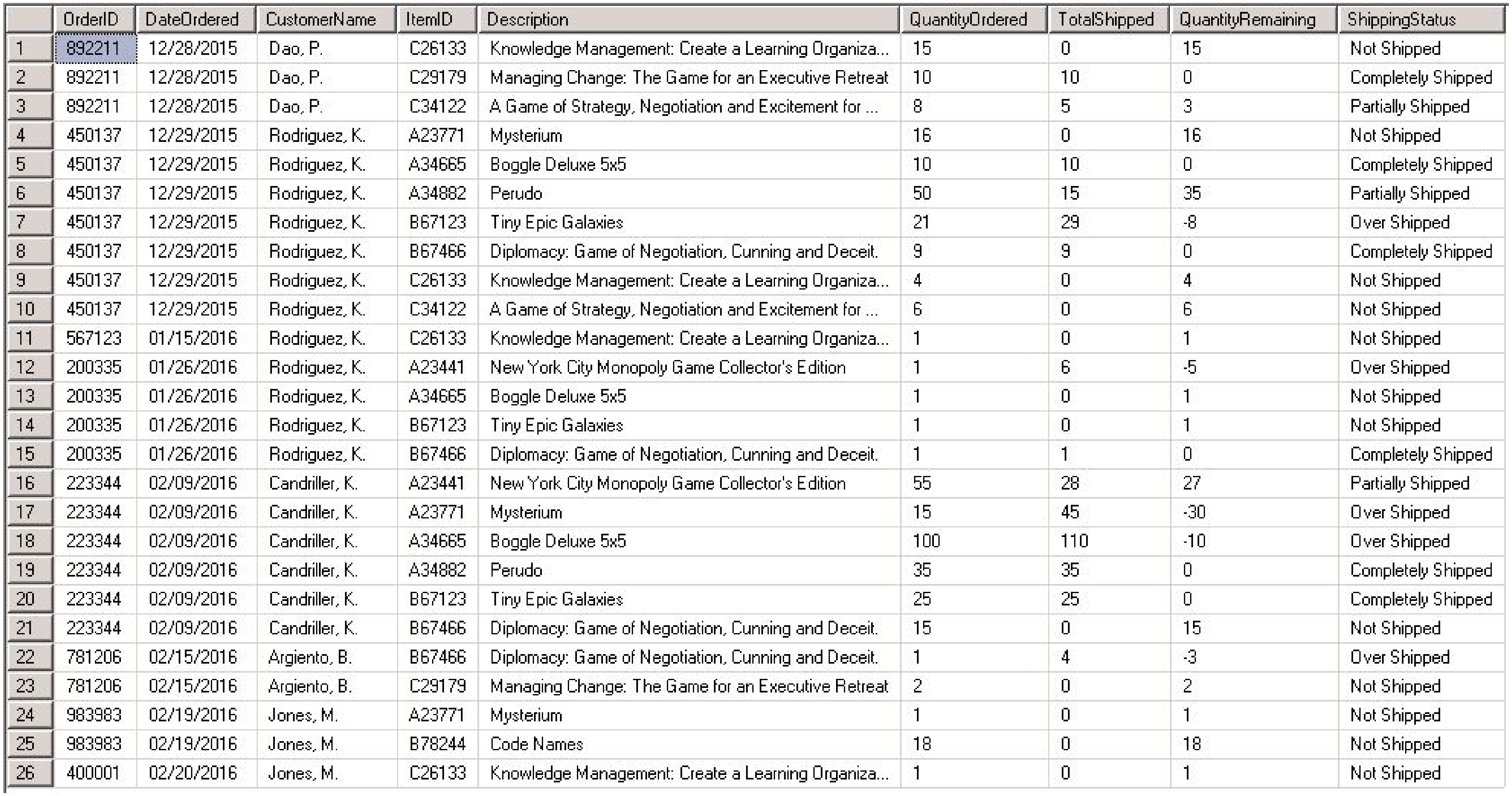
4. Code

```
Create ViewvTesting As
    Select                  ol.OrderID,
                                    Count(ol.ItemID) ItemCount
    From                    tblOrderLine ol
    Left Outer Join   vShipTotal st
    On                              ol.OrderID = st.OrderID and ol.ItemID = st.ItemID
    Where                   ((ol.Quantity - ISNULL(st.TotalShipped, 0)) < 0) or ((ol.Quantity - ISNULL(st.TotalShipped, 0)) = 0)
    Group By                ol.OrderID

Create View vOpenOrder As
    Select                  ol.OrderID,
                                    Count(ol.ItemID) TotalItems,
                                    vt.ItemCount
    From                    tblOrderLine ol
    Left Outer Join   vTesting vt
    On                              ol.OrderID = vt.OrderID
    Group By                ol.OrderID, vt.ItemCount
    Having                  count(ol.ItemID) = vt.ItemCount

Select                      ol.OrderID,
                            CONVERT(varchar, o.OrderDate, 101) DateOrdered,
                            (cus.LastName + ', ' + substring(cus.FirstName, 1, 1) + '.') CustomerName,
                            ol.ItemID,
                            it.Description,
                            ol.Quantity QuantityOrdered,
                                  ISNULL(st.TotalShipped, 0) TotalShipped,
                                  (ol.Quantity - ISNULL(st.TotalShipped, 0)) QuantityRemaining,
                                  Case
                                          When (ol.Quantity - ISNULL(st.TotalShipped, 0)) = 0 Then 'Completely Shipped'
                                          When ISNULL(st.TotalShipped, 0) = 0                                    Then 'Not Shipped'
                                          When (ol.Quantity - ISNULL(st.TotalShipped, 0)) < 0 Then 'Over Shipped'
                                          When (ol.Quantity - ISNULL(st.TotalShipped, 0)) > 0 Then 'Partially Shipped'
                                  End ShippingStatus
    From                          tblOrderLine ol
    Inner Join            tblOrder o
    On                            ol.OrderID = o.OrderID
    Inner Join            tblCustomer cus
    On                            o.CustomerID = cus.CustomerID
    Inner Join            tblItem it
    On                            ol.ItemID = it.ItemID
    Left Outer Join vShipTotal st
    On                            ol.OrderID = st.OrderID and ol.ItemID = st.ItemID
    Where                         ol.OrderID Not In (Select OrderID
                                                                  From          vOpenOrder)
    Order By OrderDate;
```

4. Output



	OrderID	DateOrdered	CustomerName	ItemID	Description	QuantityOrdered	TotalShipped	QuantityRemaining	ShippingStatus
1	892211	12/28/2015	Dao, P.	C26133	Knowledge Management: Create a Learning Organiza...	15	0	15	Not Shipped
2	892211	12/28/2015	Dao, P.	C29179	Managing Change: The Game for an Executive Retreat	10	10	0	Completely Shipped
3	892211	12/28/2015	Dao, P.	C34122	A Game of Strategy, Negotiation and Excitement for ...	8	5	3	Partially Shipped
4	450137	12/29/2015	Rodriguez, K.	A23771	Mysterium	16	0	16	Not Shipped
5	450137	12/29/2015	Rodriguez, K.	A34665	Boggle Deluxe 5x5	10	10	0	Completely Shipped
6	450137	12/29/2015	Rodriguez, K.	A34882	Perudo	50	15	35	Partially Shipped
7	450137	12/29/2015	Rodriguez, K.	B67123	Tiny Epic Galaxies	21	29	-8	Over Shipped
8	450137	12/29/2015	Rodriguez, K.	B67466	Diplomacy: Game of Negotiation, Cunning and Deceit.	9	9	0	Completely Shipped
9	450137	12/29/2015	Rodriguez, K.	C26133	Knowledge Management: Create a Learning Organiza...	4	0	4	Not Shipped
10	450137	12/29/2015	Rodriguez, K.	C34122	A Game of Strategy, Negotiation and Excitement for ...	6	0	6	Not Shipped
11	567123	01/15/2016	Rodriguez, K.	C26133	Knowledge Management: Create a Learning Organiza...	1	0	1	Not Shipped
12	200335	01/26/2016	Rodriguez, K.	A23441	New York City Monopoly Game Collector's Edition	1	6	-5	Over Shipped
13	200335	01/26/2016	Rodriguez, K.	A34665	Boggle Deluxe 5x5	1	0	1	Not Shipped
14	200335	01/26/2016	Rodriguez, K.	B67123	Tiny Epic Galaxies	1	0	1	Not Shipped
15	200335	01/26/2016	Rodriguez, K.	B67466	Diplomacy: Game of Negotiation, Cunning and Deceit.	1	1	0	Completely Shipped
16	223344	02/09/2016	Candriller, K.	A23441	New York City Monopoly Game Collector's Edition	55	28	27	Partially Shipped
17	223344	02/09/2016	Candriller, K.	A23771	Mysterium	15	45	-30	Over Shipped
18	223344	02/09/2016	Candriller, K.	A34665	Boggle Deluxe 5x5	100	110	-10	Over Shipped
19	223344	02/09/2016	Candriller, K.	A34882	Perudo	35	35	0	Completely Shipped
20	223344	02/09/2016	Candriller, K.	B67123	Tiny Epic Galaxies	25	25	0	Completely Shipped
21	223344	02/09/2016	Candriller, K.	B67466	Diplomacy: Game of Negotiation, Cunning and Deceit.	15	0	15	Not Shipped
22	781206	02/15/2016	Argiento, B.	B67466	Diplomacy: Game of Negotiation, Cunning and Deceit.	1	4	-3	Over Shipped
23	781206	02/15/2016	Argiento, B.	C29179	Managing Change: The Game for an Executive Retreat	2	0	2	Not Shipped
24	983983	02/19/2016	Jones, M.	A23771	Mysterium	1	0	1	Not Shipped
25	983983	02/19/2016	Jones, M.	B78244	Code Names	18	0	18	Not Shipped
26	400001	02/20/2016	Jones, M.	C26133	Knowledge Management: Create a Learning Organiza...	1	0	1	Not Shipped

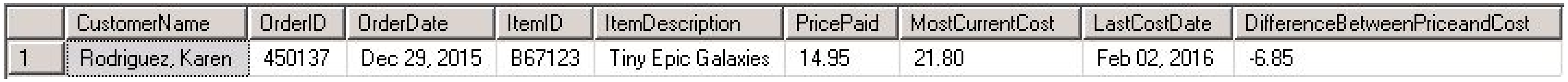
5. Code

```
Create View vTEG As
Select              ol.OrderID,
                    ol.ItemID,
                    ol.Price,
                    i.Description
From                tblOrderLine ol
```

```
Left Outer Join tblItem i
On                      ol.ItemID = i.ItemID
Where                   i.Description like 'Tiny Epic Galaxies';

Select                  (c.LastName + ', ' + c.FirstName) CustomerName,
                        ol.OrderID,
                        CONVERT(varchar, o.OrderDate, 107) OrderDate,
                        ol.ItemID,
                        i.Description ItemDescription,
                        ol.Price PricePaid,
                        ich.LastCost MostCurrentCost,
                        CONVERT(varchar, ich.LastCostDate, 107) LastCostDate,
                        (ol.Price - ich.LastCost) DifferenceBetweenPriceandCost
From                    tblOrderLine ol
Left Outer Join tblOrder o
On                      ol.OrderID = o.OrderID
Left Outer Join tblCustomer c
On                      o.CustomerID = c.CustomerID
Left Outer Join tblItem i
On                      ol.ItemID = i.ItemID
Left Outer Join tblItemCostHistory ich
On                      i.ItemID = ich.ItemID
Where                   ol.Price = (Select      MIN(Price)
                                                From    vTEG vInner)
                        and
                        ich.LastCostDate = (Select      MAX(LastCostDate)
                                                        From    tblItemCostHistory ichInner
                                                        Where   ich.ItemID = ichInner.ItemID)
```

5. Output



	CustomerName	OrderID	OrderDate	ItemID	ItemDescription	PricePaid	MostCurrentCost	LastCostDate	DifferenceBetweenPriceandCost
1	Rodriguez, Karen	450137	Dec 29, 2015	B67123	Tiny Epic Galaxies	14.95	21.80	Feb 02, 2016	-6.85

6. Code

```
Create View vPPI As
```

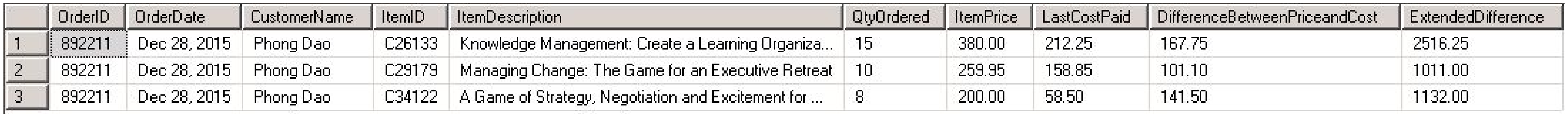
```
Select                      ol.OrderID,
                            ol.ItemID,
                            ol.Price ItemPrice,
                            ich.LastCost LastCostPaid,
                            (ol.Price - ich.LastCost) DifferenceBetweenPriceandCost,
                            ol.Quantity * (ol.Price - ich.LastCost) ExtendedDifference
From                        tblOrderLine ol
Left Outer Join tblItem i
On                          ol.ItemID = i.ItemID
Left Outer Join tblItemCostHistory ich
On                          i.ItemID = ich.ItemID
Where                       ich.LastCostDate = (Select      MAX(LastCostDate)
                                                                    From    tblItemCostHistory ichInner
                                                                    Where   ich.ItemID = ichInner.ItemID)

Create View vMPI As
Select                      v.OrderID,
                            v.ExtendedDifference
From                        vPPI v
Where                       v.ExtendedDifference = (Select  MAX(ExtendedDifference)
                                                                            From    vPPI vInner)

Select                      ol.OrderID,
                            CONVERT(varchar, o.OrderDate, 107) OrderDate,
                            (c.FirstName + ' ' + c.LastName) CustomerName,
                            ol.ItemID,
                            i.Description ItemDescription,
                            ol.Quantity QtyOrdered,
                            vPPI.ItemPrice,
                            vPPI.LastCostPaid,
                            vPPI.DifferenceBetweenPriceandCost,
                            vPPI.ExtendedDifference
From                        tblOrderLine ol
Left Outer Join     tblOrder o
On                          ol.OrderID = o.OrderID
Left Outer Join     tblCustomer c
On                          o.CustomerID = c.CustomerID
```

```
Left Outer Join tblItem i
On                      ol.ItemID = i.ItemID
Left Outer Join vPPI
On                      ol.OrderID = vPPI.OrderID and ol.ItemID = vPPI.ItemID
Where                   ol.OrderID = (Select    OrderID
                                      From         vMPI)
```

6. Output



	OrderID	OrderDate	CustomerName	ItemID	ItemDescription	QtyOrdered	ItemPrice	LastCostPaid	DifferenceBetweenPriceandCost	ExtendedDifference
1	892211	Dec 28, 2015	Phong Dao	C26133	Knowledge Management: Create a Learning Organiza...	15	380.00	212.25	167.75	2516.25
2	892211	Dec 28, 2015	Phong Dao	C29179	Managing Change: The Game for an Executive Retreat	10	259.95	158.85	101.10	1011.00
3	892211	Dec 28, 2015	Phong Dao	C34122	A Game of Strategy, Negotiation and Excitement for ...	8	200.00	58.50	141.50	1132.00

7. Code

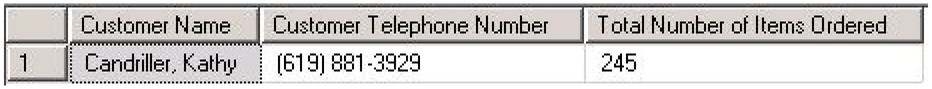
```
Create View vTIO As
Select                  c.CustomerID,
                        SUM(ol.Quantity) TotalItemsOrdered
From                    tblOrderLine ol
Left Outer Join tblOrder o
On                      ol.OrderID = o.OrderID
Left Outer Join tblCustomer c
On                      o.CustomerID = c.CustomerID
Group By           c.CustomerID

Create View vMIO As
Select                  v.CustomerID,
                        v.TotalItemsOrdered
From                    vTIO v
Where                   v.TotalItemsOrdered = (Select   MAX(TotalItemsOrdered)
                                                        From    vTIO vInner)

Select                  (c.LastName + ', ' + c.FirstName) 'Customer Name',
                        '(' + SUBSTRING(c.Phone,1,3) + ') ' + SUBSTRING(c.Phone,4,3) + '-' + SUBSTRING(c.Phone,7,4) 'Customer Telephone
Number',
                        vMIO.TotalItemsOrdered 'Total Number of Items Ordered'
```

```
From                    tblCustomer c
Left Outer Join vMIO
On                      c.CustomerID = vMIO.CustomerID
Where                   c.CustomerID = (Select      CustomerID
                                                    From          vMIO)
```

7. Output



	Customer Name	Customer Telephone Number	Total Number of Items Ordered
1	Candriller, Kathy	(619) 881-3929	245

8. Code

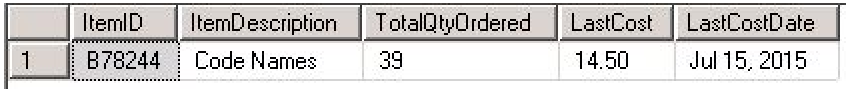
```
Create View vCIN As
Select                  ol.ItemID,
                        SUM(ol.Quantity) TotalQtyOrdered
From                    tblOrderLine ol
Left Outer Join tblOrder o
On                      ol.OrderID = o.OrderID
Left Outer Join tblCustomer c
On                      o.CustomerID = c.CustomerID
Where                   c.State like 'NV'
Group By          ol.ItemID;

Create View vMQO As
Select                  v.ItemID,
                        v.TotalQtyOrdered,
                        ich.LastCost,
                        CONVERT(varchar, ich.LastCostDate, 107) LastCostDate
From                    vCIN v
Left Outer Join tblItem i
On                      v.ItemID = i.ItemID
Left Outer Join tblItemCostHistory ich
On                      i.ItemID = ich.ItemID
Where                   v.TotalQtyOrdered = (Select    MAX(TotalQtyOrdered)
                                                           From    vCIN vInner)
```

```
                                and
                                ich.LastCostDate = (Select      MAX(LastCostDate)
                                                                      From     tblItemCostHistory ichInner
                                                                      Where   ich.ItemID = ichInner.ItemID)

    Select                      i.ItemID,
                                i.Description ItemDescription,
                                vMQO.TotalQtyOrdered,
                                vMQO.LastCost,
                                vMQO.LastCostDate
    From                        tblItem i
    Left Outer Join vMQO
    On                          i.ItemID = vMQO.ItemID
    Where                       i.ItemID = (Select      ItemID
                                                        From     vMQO)
```

8. Output



	ItemID	ItemDescription	TotalQtyOrdered	LastCost	LastCostDate
1	B78244	Code Names	39	14.50	Jul 15, 2015

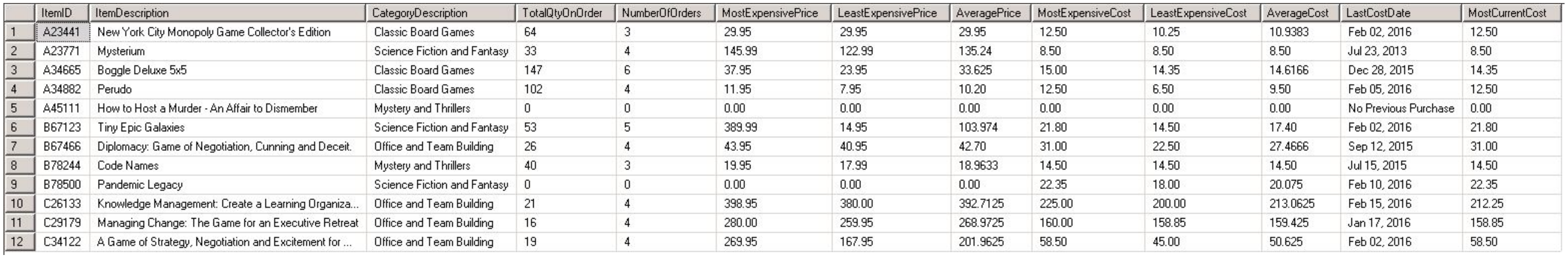
9. Code

```
    Create View vIOL As
    Select                      ol.ItemID,
                                SUM(ol.Quantity) TotalQtyOnOrder,
                                COUNT(ol.OrderID) NumberOfOrders,
                                MAX(ol.Price) MostExpensivePrice,
                                MIN(ol.Price) LeastExpensivePrice,
                                AVG(ol.Price) AveragePrice
    From                        tblOrderLine ol
    Group By            ol.ItemID;

    Create View vILC As
    Select                      ich.ItemID,
                                MAX(ich.LastCost) MostExpensiveCost,
                                MIN(ich.LastCost) LeastExpensiveCost,
                                        AVG(ich.LastCost) AverageCost,
                                        MAX(ich.LastCostDate) LastCostDate
From                                    tblItemCostHistory ich
Group By                         ich.ItemID;

Select                                  i.ItemID,
                                        i.Description ItemDescription,
                                        it.CategoryDescription,
                                        ISNULL(vIOL.TotalQtyOnOrder, 0) TotalQtyOnOrder,
                                        ISNULL(vIOL.NumberOfOrders, 0) NumberOfOrders,
                                        ISNULL(vIOL.MostExpensivePrice, 0) MostExpensivePrice,
                                        ISNULL(vIOL.LeastExpensivePrice, 0) LeastExpensivePrice,
                                        ISNULL(vIOL.AveragePrice, 0) AveragePrice,
                                        ISNULL(vILC.MostExpensiveCost, 0) MostExpensiveCost,
                                        ISNULL(vILC.LeastExpensiveCost, 0) LeastExpensiveCost,
                                        ISNULL(vILC.AverageCost, 0) AverageCost,
                                        ISNULL(CONVERT(varchar, vILC.LastCostDate, 107), 'No Previous Purchase') LastCostDate,
                                        ISNULL((Select             ich.LastCost
                                                       From               tblItemCostHistory ich
                                                       Where              ich.LastCostDate = vILC.LastCostDate and ich.ItemID = vILC.itemID), 0) MostCurrentCost
From                                    tblItem i
Left Outer Join tblItemType it
On                                      i.TypeID = it.TypeID
Left Outer Join vIOL
On                                      i.ItemID = vIOL.ItemID
Left Outer Join vILC
On                                      i.ItemID = vILC.ItemID
```

9. Output



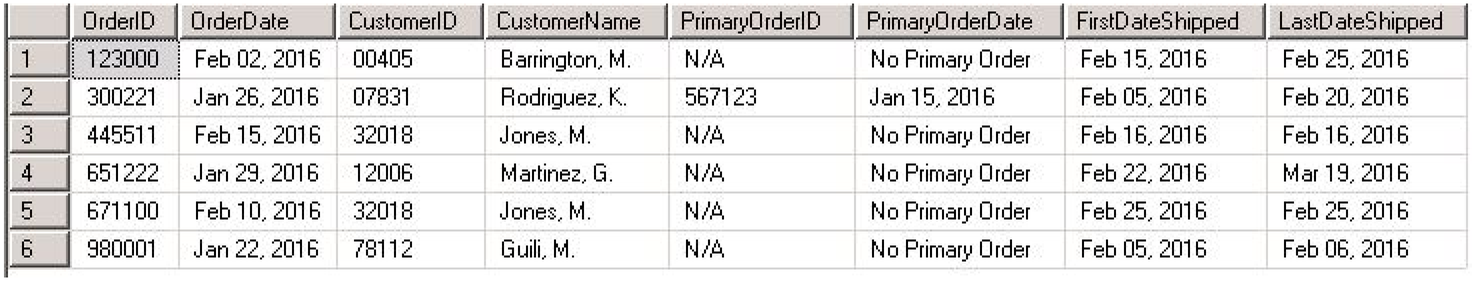
	ItemID	ItemDescription	CategoryDescription	TotalQtyOnOrder	NumberOfOrders	MostExpensivePrice	LeastExpensivePrice	AveragePrice	MostExpensiveCost	LeastExpensiveCost	AverageCost	LastCostDate	MostCurrentCost
1	A23441	New York City Monopoly Game Collector's Edition	Classic Board Games	64	3	29.95	29.95	29.95	12.50	10.25	10.9383	Feb 02, 2016	12.50
2	A23771	Mysterium	Science Fiction and Fantasy	33	4	145.99	122.99	135.24	8.50	8.50	8.50	Jul 23, 2013	8.50
3	A34665	Boggle Deluxe 5x5	Classic Board Games	147	6	37.95	23.95	33.625	15.00	14.35	14.6166	Dec 28, 2015	14.35
4	A34882	Perudo	Classic Board Games	102	4	11.95	7.95	10.20	12.50	6.50	9.50	Feb 05, 2016	12.50
5	A45111	How to Host a Murder - An Affair to Dismember	Mystery and Thrillers	0	0	0.00	0.00	0.00	0.00	0.00	0.00	No Previous Purchase	0.00
6	B67123	Tiny Epic Galaxies	Science Fiction and Fantasy	53	5	389.99	14.95	103.974	21.80	14.50	17.40	Feb 02, 2016	21.80
7	B67466	Diplomacy: Game of Negotiation, Cunning and Deceit.	Office and Team Building	26	4	43.95	40.95	42.70	31.00	22.50	27.4666	Sep 12, 2015	31.00
8	B78244	Code Names	Mystery and Thrillers	40	3	19.95	17.99	18.9633	14.50	14.50	14.50	Jul 15, 2015	14.50
9	B78500	Pandemic Legacy	Science Fiction and Fantasy	0	0	0.00	0.00	0.00	22.35	18.00	20.075	Feb 10, 2016	22.35
10	C26133	Knowledge Management: Create a Learning Organiza...	Office and Team Building	21	4	398.95	380.00	392.7125	225.00	200.00	213.0625	Feb 15, 2016	212.25
11	C29179	Managing Change: The Game for an Executive Retreat	Office and Team Building	16	4	280.00	259.95	268.9725	160.00	158.85	159.425	Jan 17, 2016	158.85
12	C34122	A Game of Strategy, Negotiation and Excitement for ...	Office and Team Building	19	4	269.95	167.95	201.9625	58.50	45.00	50.625	Feb 02, 2016	58.50

10. Code

```
Create View vOSL As
Select              sl.OrderID,
                    MIN(DateShipped) FirstDateShipped,
                    MAX(DateShipped) LastDateShipped
From                tblShipLine sl
Where               sl.OrderID In (Select   OrderID
                                    From    vOpenOrder)
Group By            sl.OrderID;

Select                      v.OrderID,
                            CONVERT(varchar, o.OrderDate, 107) OrderDate,
                            o.CustomerID,
                            (c.LastName + ', ' + substring(c.FirstName, 1, 1) + '.') CustomerName,
                            ISNULL(o.PrimaryOrderID, 'N/A') PrimaryOrderID,
                            ISNULL((Select  CONVERT(varchar, oInner.OrderDate, 107)
                                            From    tblOrder oInner
                                            Where   oInner.OrderID = o.PrimaryOrderID), 'No Primary Order') PrimaryOrderDate,
                            CONVERT(varchar, vOSL.FirstDateShipped, 107) FirstDateShipped,
                            CONVERT(varchar, vOSL.LastDateShipped, 107) LastDateShipped
From                        vOpenOrder v
Left Outer Join tblOrder o
On                          v.OrderID = o.OrderID
Left Outer Join tblCustomer c
On                          o.CustomerID = c.CustomerID
Left Outer Join vOSL
On                          v.OrderID = vOSL.OrderID;
```

10. Output



	OrderID	OrderDate	CustomerID	CustomerName	PrimaryOrderID	PrimaryOrderDate	FirstDateShipped	LastDateShipped
1	123000	Feb 02, 2016	00405	Barrington, M.	N/A	No Primary Order	Feb 15, 2016	Feb 25, 2016
2	300221	Jan 26, 2016	07831	Rodriguez, K.	567123	Jan 15, 2016	Feb 05, 2016	Feb 20, 2016
3	445511	Feb 15, 2016	32018	Jones, M.	N/A	No Primary Order	Feb 16, 2016	Feb 16, 2016
4	651222	Jan 29, 2016	12006	Martinez, G.	N/A	No Primary Order	Feb 22, 2016	Mar 19, 2016
5	671100	Feb 10, 2016	32018	Jones, M.	N/A	No Primary Order	Feb 25, 2016	Feb 25, 2016
6	980001	Jan 22, 2016	78112	Guili, M.	N/A	No Primary Order	Feb 05, 2016	Feb 06, 2016

11. Code

```
Create View vQLS As
Select                    ol.ItemID,
                          SUM((ol.Quantity - ISNULL(vst.TotalShipped, 0))) QtyRemaining
From                      tblOrderLine ol
Left Outer Join vShipTotal vst
On                        ol.OrderID = vst.OrderID and ol.ItemID = vst.ItemID
Group By          ol.ItemID

Select                    i.ItemID,
                          i.Description ItemDescription,
                          vQLS.QtyRemaining 'Total Left To Ship',
                          SUM(ISNULL(il.QtyOnHand, 0)) 'Total Available in Inventory',
                          (vQLS.QtyRemaining - SUM(ISNULL(il.QtyOnHand, 0))) QtyRemaining
From                      tblItem i
Left Outer Join tblItemLocation il
On                        i.ItemID = il.ItemID
Left Outer Join tblOrderLine ol
On                        i.ItemID = ol.ItemID
Left Outer Join vShipTotal vst
On                        i.ItemID = vst.ItemID
Left Outer Join vQLS
On                        i.ItemID = vQLS.ItemID
Group By          i.ItemID, i.Description, vQLS.QtyRemaining
```

11. Output

Unable to obtain correct result table.