

Suppose that you have designed a database for Morgan Importing that has the following tables:

EMPLOYEE (EmployeeID, LastName, FirstName, Department, Phone, Fax, EmailAddress)

STORE (StoreName, City, Country, Phone, Fax, EmailAddress, Contact)

ITEM (ItemID, StoreName, PurchasingAgentID, PurchaseDate, ItemDescription, Category, PriceUSD)

SHIPPER (ShipperID, ShipperName, Phone, Fax, EmailAddress, Contact)

SHIPMENT (ShipmentID, ShipperID, PurchasingAgentID, ShipperInvoiceNumber, Origin, Destination, ScheduledDepartureDate, ActualDepartureDate, EstimatedArrivalDate)

SHIPMENT_ITEM (ShipmentID, ShipmentItemID, ItemID, InsuredValue)

SHIPMENT_RECEIPT (ReceiptNumber, ShipmentID, ItemID, ReceivingAgentID, ReceiptDate, ReceiptTime, ReceiptQuantity, isReceivedUndamaged, DamageNotes)

- A. Do you think STORE should have a surrogate key? If so, create it and make required Adjustments in the design. If not, explain why not or make any other adjustments to STORE and other tables that you think are appropriate.***

Answer:

Yes, STORE should have a surrogate key. A surrogate key is any column or set of columns that can be declared as the primary key instead of a "real" or natural key. A surrogate is a candidate key.

The table StoreName has a primary key which maybe large and used throughout the database. Also, the primary key should be unique and can't be reused after deletion. Eg: if any StoreName has been deleted then it's not possible to add it again. Therefore we need surrogate key.

The design of the store should be changed and below are the changes we need to do:

Alter table Store

Add StoreID Int identity(1,1)

Alter table item

drop foreign_key StoreName

Alter table item

add foreign_key StoreID

References Store(StoreID)

B. Specify NULL/NOT NULL constraints for each table column.

<i>Table</i>	<i>NULL column</i>	<i>NOT NULL Columns</i>
EMPLOYEE	Department	EmployeeID
	Phone	LastName
	Fax	FirstName
		Emailaddress
STORE	City	StoreID
	Phone	StoreName
	Fax	Emailaddress
	Contact	Country
ITEM	PurchaseDate	ItemID
	ItemDescription	StoreID
	Category	PurchasingAgentId
	PriceUSD	
SHIPPER	ShipperName	ShipperID
	Phone	Emailaddress
	Fax	
	Contact	
SHIPMENT	ShipperInvoiceNumber	ShipmentID
	Origin	ShipperID
	Destination	PurchasingAgentId
	ScheduledDepartureDate	
	ActualDepartureDate	
	EstimatedArrivalDate	
SHIPMENT_ITEM	InsuredValue	ShipmentID
		ShipmentItemID
		ItemID
SHIPMENT_RECEIPT	ReceiptDate	ReceiptNumber
	ReceiptTime	ShipmentID
	ReceiptQuantity	ItemID
	isReceivedUndamaged	
	DamageNotes	
	ReceivingAgentID	

C. Specify alternate keys, if any.**Answer:**

We have already created the necessary primary, foreign and surrogate keys. There are no alternate keys which are required in this database.

D. State relationships as implied by foreign keys, and specify the maximum and minimum Cardinality of each relationship. Justify your choices.

Answer:

The second relation is between shipper table and shipment table. The maximum cardinality is N and minimum is 1.

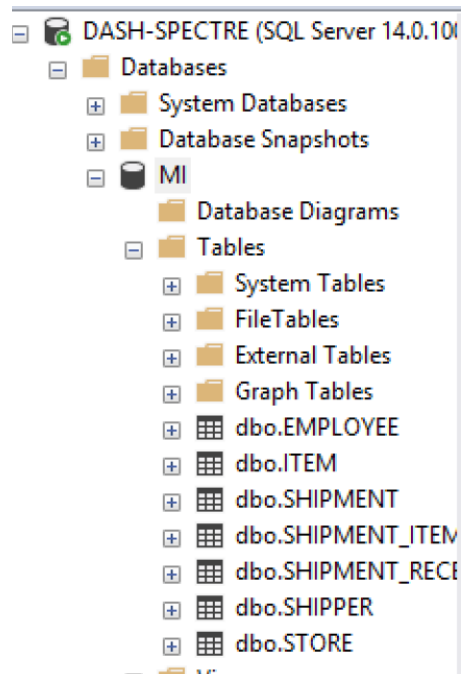
Shipper can have multiple shipments, but a shipment can have only one shipper.

E. Explain how you will enforce the minimum cardinalities in your answer to part D. Use referential integrity actions for required parents, if any. Use Figure 6-29(b) as a boilerplate for required children, if any.

Answer:

We can enforce minimum cardinality by either creating a trigger which can check whether child is placed in the table before the parent is placed. We can also use a unique constraint ie) enforce a rule on the child row.

F. Create a database named MI in your DBMS.



- G. Write *CREATE TABLE* statements for each of the tables using your answers to parts A–E, as necessary. If you decided to use a *StoreID* surrogate key, set the first value to 1000 and increment by 50. Set the first value of *EmployeeID* and *ShipperID* to 1 and increment it by 1. Set the first value of *ItemID* to 500 and increment it by 5. Set the first value of *ShipmentID* to 100 and increment it by 1. *ReceiptNumber* should start at 200001 and increment by 1. Use *FOREIGN KEY* constraints to create appropriate referential integrity constraints. Set *UPDATE* and *DELETE* behavior in accordance with your referential integrity action design. Set the default value of *InsuredValue* to 100. Write a constraint that *STORE.Country* be limited to seven countries (Hong Kong, India, Japan, Peru, Philippines, Singapore, United States).**
- **New Table *EMPLOYEE* is created which is as shown below:**

```
CREATE TABLE EMPLOYEE
(
  EmployeeID int identity (1,1),
  LastName VARCHAR (20) Not null,
  FirstName VARCHAR (20) not null,
  Department VARCHAR (20),
  Phone VARCHAR (20),
  Fax VARCHAR (20),
  EmailAddress VARCHAR (50) not null,
  Constraint pk1 primary key(EmployeeID)
)
```

Results		Messages				
EmployeeID	LastName	FirstName	Department	Phone	Fax	EmailAddress

- **New Table *STORE* is created which is as shown below:**

```
CREATE TABLE STORE
(
  StoreId int identity (1000,50),
  StoreName VARCHAR (20) Not NULL,
  City VARCHAR (20),
  Country VARCHAR (30) Not NULL,
  Phone VARCHAR (20),
  Fax VARCHAR (20),
  EmailAddress VARCHAR (50) Not NULL,
  Contact VARCHAR (20),
  Constraint pk2 primary key(StoreId),
  Constraint Country_ch CHECK (Country='Hong Kong' or Country='India' or
  Country='Japan' or Country='Peru' or Country='Philippines' or
  Country='Singapore' or Country='United States')
)
```

Results		Messages					
StoreId	StoreName	City	Country	Phone	Fax	EmailAddress	Contact

- **New Table ITEM is created which is as shown below:**

```
CREATE TABLE ITEM
(
  ItemID int identity (500,5),
  StoreId INT,
  PurchasingAgentID INT not NULL,
  PurchaseDate DATE,
  ItemDescription VARCHAR (50),
  Category VARCHAR (20),
  PriceUSD float,
  Constraint pk4 primary key(ItemID),
  Constraint fk1 foreign key(StoreId) references STORE(StoreId) ON DELETE
  CASCADE ON UPDATE CASCADE
)
```

Results		Messages				
ItemID	StoreId	PurchasingAgentID	PurchaseDate	ItemDescription	Category	PriceUSD

- **New Table SHIPPER is created which is as shown below:**

```
CREATE TABLE SHIPPER
(
  ShipperID int identity (1,1),
  ShipperName VARCHAR (20),
  Phone VARCHAR (20),
  Fax VARCHAR (20),
  EmailAddress VARCHAR (50) not NULL,
  Contact VARCHAR (20),
  Constraint pk5 primary key(ShipperID)
)
```

Results		Messages			
ShipperID	ShipperName	Phone	Fax	EmailAddress	Contact

- **New Table SHIPMENT which is created is as shown below:**

```
CREATE TABLE SHIPMENT
(
  ShipmentID int identity (100,1),
  ShipperID INT,
  PurchasingAgentID INT not NULL,
  ShipperInvoiceNumber INT,
  Origin VARCHAR (20),
  Destination VARCHAR (20),
  ScheduledDepartureDate DATE,
  ActualDepartureDate DATE,
  EstimatedArrivalDate DATE,
  Constraint pk6 primary key (ShipmentID),
  Constraint fk2 foreign key (ShipperID) references SHIPPER (ShipperID) ON DELETE
  CASCADE ON UPDATE CASCADE
)
```

Results Messages

ShipmentID	ShipperID	PurchasingAgentID	ShipperInvoiceNumber	Origin	Destination	ScheduledDepartureDate	ActualDepartureDate	EstimatedArrivalDate
------------	-----------	-------------------	----------------------	--------	-------------	------------------------	---------------------	----------------------

- **New Table SHIPMENT_ITEM is created which is as shown below:**

```
CREATE TABLE SHIPMENT_ITEM
(
  ShipmentID INT,
  ShipmentItemID INT,
  ItemID INT,
  InsuredValue INT DEFAULT 100,
  Constraint pk7 primary key (ShipmentItemID),
  Constraint fk3 foreign key (ShipmentID) references SHIPMENT (ShipmentID) ON
  DELETE CASCADE ON UPDATE CASCADE,
  Constraint fk4 foreign key (ItemID) references ITEM (ItemID) ON DELETE CASCADE
  ON UPDATE CASCADE
)
```

Results		Messages	
ShipmentID	ShipmentItemID	ItemID	InsuredValue

- New Table SHIPMENT_RECEIPT is created which is as shown below:

```
CREATE TABLE SHIPMENT_RECEIPT
(
  ReceiptNumber int identity (200001,1),
  ShipmentID INT,
  ItemID INT,
  ReceivingAgentID INT,
  ReceiptDate DATE,
  ReceiptTime TIME,
  ReceiptQuantity INT,
  isReceivedUndamaged VARCHAR (20),
  DamageNotes VARCHAR (50),
  Constraint pk9 primary key(ReceiptNumber),
  Constraint fk5 foreign key(ShipmentID) references SHIPMENT(ShipmentID) ON
  DELETE CASCADE ON UPDATE CASCADE,
  Constraint fk6 foreign key(ItemID) references ITEM(ItemID)ON DELETE CASCADE ON
  UPDATE CASCADE
)
```

Results		Messages						
ReceiptNumber	ShipmentID	ItemID	ReceivingAgentID	ReceiptDate	ReceiptTime	ReceiptQuantity	isReceivedUndamaged	DamageNotes

H. Write INSERT statements to insert the data shown in Figures 7-60, 7-61, 7-62, 7-63, 7-64, 7-65, and 7-66.

EMPLOYEE TABLE (7-60)

- `INSERT INTO EMPLOYEE (LastName,FirstName,Department,Phone,Fax,EmailAddress)
VALUES ('Morgan','James','Executive','310-208-1401','310-208-1499',
'James.Morgan@morganimporting.com');`
- `INSERT INTO EMPLOYEE (LastName,FirstName,Department,Phone,Fax,EmailAddress)
VALUES ('Morgan','Jessica','Executive','310-208-1402','310-208-1499',
'Jessica.Morgan@morganimporting.com');`
- `INSERT INTO EMPLOYEE (LastName,FirstName,Department,Phone,Fax,EmailAddress)
VALUES ('Williams','David','Purchasing','310-208-1434','310-208-1498',
'David.Williams@morganimporting.com');`
- `INSERT INTO EMPLOYEE (LastName,FirstName,Department,Phone,Fax,EmailAddress)
VALUES ('Wright','James','Receiving','310-208-1456','310-208-1497',
'James.Wright@morganimporting.com');`
- `INSERT INTO EMPLOYEE (LastName,FirstName,Department,Phone,Fax,EmailAddress)
VALUES ('Douglas','Tom','Receiving','310-208-1457','310-208-1497',
'Tom.Douglas@morganimporting.com');`

Results		Messages					
	EmployeeID	LastName	FirstName	Department	Phone	Fax	EmailAddress
1	1	Morgan	James	Executive	310-208-1401	310-208-1499	James.Morgan@morganimporting.com
2	2	Morgan	Jessica	Executive	310-208-1402	310-208-1499	Jessica.Morgan@morganimporting.com
3	3	Williams	David	Purchasing	310-208-1434	310-208-1498	David.Williams@morganimporting.com
4	4	Wright	James	Receiving	310-208-1456	310-208-1497	James.Wright@morganimporting.com
5	5	Douglas	Tom	Receiving	310-208-1457	310-208-1497	Tom.Douglas@morganimporting.com

✓ Query executed successfully. | DASH-SPECTRE (14.0 RTM)

STORE TABLE (7-61)

- **INSERT INTO** STORE (StoreName, City, Country, Phone, Fax, EmailAddress, Contact)
VALUES ('Eastern Sales', 'Singapore', 'Singapore', '65-543-1233', '65-543-1239', 'Sales@EasternSales.com.sg', 'Jeremy');
- **INSERT INTO** STORE (StoreName, City, Country, Phone, Fax, EmailAddress, Contact)
VALUES ('Eastern Treasures', 'Manila', 'Philippines', '63-2-654-2344', '63-2-654-2349', 'Sales@EasternTreasures.com.ph', 'Gracielle');
- **INSERT INTO** STORE (StoreName, City, Country, Phone, Fax, EmailAddress, Contact)
VALUES ('Jade Antiques', 'Singapore', 'Singapore', '65-543-3455', '65-543-3459', 'Sales@JadeAntiques.com.sg', 'Swee Lai');
- **INSERT INTO** STORE (StoreName, City, Country, Phone, Fax, EmailAddress, Contact)
VALUES ('Andes Treasures', 'Lima', 'Peru', '51-14-765-4566', '51-14-765-4569', 'Sales@AndesTreasures.com.pe', 'Juan Carlos');
- **INSERT INTO** STORE (StoreName, City, Country, Phone, Fax, EmailAddress, Contact)
VALUES ('Eastern Treasures', 'New Delhi', 'India', '91-11-987-6788', '91-11-987-6789', 'Sales@EasternTreasures.com.in', 'Deepinder');
- **INSERT INTO** STORE (StoreName, City, Country, Phone, Fax, EmailAddress, Contact)
VALUES ('European Imports', 'New York City', 'United States', '800-432-8766', '800-432-8769', 'Sales@EuropeanImports.com.sg', 'Marcello');

Results		Messages						
	StoreId	StoreName	City	Country	Phone	Fax	EmailAddress	Contact
1	1000	Eastern Sales	Singapore	Singapore	65-543-1233	65-543-1239	Sales@EasternSales.com.sg	Jeremy
2	1050	Eastern Treasures	Manila	Philippines	63-2-654-2344	63-2-654-2349	Sales@EasternTreasures.com.ph	Gracielle
3	1100	Jade Antiques	Singapore	Singapore	65-543-3455	65-543-3459	Sales@JadeAntiques.com.sg	Swee Lai
4	1150	Andes Treasures	Lima	Peru	51-14-765-4566	51-14-765-4569	Sales@AndesTreasures.com.pe	Juan Carlos
5	1200	Eastern Treasures	New Delhi	India	91-11-987-6788	91-11-987-6789	Sales@EasternTreasures.com.in	Deepinder
6	1250	European Imports	New York City	United States	800-432-8766	800-432-8769	Sales@EuropeanImports.com.sg	Marcello

✓ Query executed successfully. DASH-SPECTRE (14.0 RTM) DASH-SPECTRE\ddalvi (54)

ITEM TABLE (7-62)

- `INSERT INTO ITEM (StoreID, PurchasingAgentID, PurchaseDate,ItemDescription,Category,PriceUSD)`
`VALUES (1050,101,'12/10/2014','Antique Large Bureaus','Furniture',13415);`
- `INSERT INTO ITEM (StoreID, PurchasingAgentID, PurchaseDate,ItemDescription,Category,PriceUSD)`
`VALUES (1050,102,'12/12/2014','Porcelain Lamps','Lamps',13300);`
- `INSERT INTO ITEM (StoreID, PurchasingAgentID, PurchaseDate,ItemDescription,Category,PriceUSD)`
`VALUES (1200,104,'12/15/2014','Gold Rim Design China','Tableware',38500);`
- `INSERT INTO ITEM (StoreID, PurchasingAgentID, PurchaseDate,ItemDescription,Category,PriceUSD)`
`VALUES (1200,104,'12/16/2014','Gold Rim Design Serving Dishes','Tableware',3200);`
- `INSERT INTO ITEM (StoreID, PurchasingAgentID, PurchaseDate,ItemDescription,Category,PriceUSD)`
`VALUES (1050,102,'4/7/2015','QE Dining Set','Furniture',14300);`
- `INSERT INTO ITEM (StoreID, PurchasingAgentID, PurchaseDate,ItemDescription,Category,PriceUSD)`
`VALUES (1100,103,'5/18/2015','Misc Linen','Linens',88545);`
- `INSERT INTO ITEM (StoreID, PurchasingAgentID, PurchaseDate,ItemDescription,Category,PriceUSD)`
`VALUES (1000,103,'5/19/2015','Large Masks','Decorations',22135);`
- `INSERT INTO ITEM (StoreID, PurchasingAgentID, PurchaseDate,ItemDescription,Category,PriceUSD)`
`VALUES (1100,104,'5/20/2015','Willow Design China','Tableware',147575);`
- `INSERT INTO ITEM (StoreID, PurchasingAgentID, PurchaseDate,ItemDescription,Category,PriceUSD)`
`VALUES (1100,104,'5/20/2015','Willow Design Serving Dishes','Tableware',12040);`
- `INSERT INTO ITEM (StoreID, PurchasingAgentID, PurchaseDate,ItemDescription,Category,PriceUSD)`
`VALUES (1150,102,'6/14/2015','Woven Goods','Decorations',1200);`
- `INSERT INTO ITEM (StoreID, PurchasingAgentID, PurchaseDate,ItemDescription,Category,PriceUSD)`
`VALUES (1150,101,'6/16/2015','Antique Leather Chairs','Furniture',5375);`
- `INSERT INTO ITEM (StoreID, PurchasingAgentID, PurchaseDate,ItemDescription,Category,PriceUSD)`
`VALUES (1100,104,'7/15/2015','Willow Design Serving Dishes','Tableware',4500);`
- `INSERT INTO ITEM (StoreID, PurchasingAgentID, PurchaseDate,ItemDescription,Category,PriceUSD)`
`VALUES (1000,103,'7/17/2015','Large Bureau','Furniture',9500);`

- `INSERT INTO ITEM (StoreID, PurchasingAgentID, PurchaseDate, ItemDescription, Category, PriceUSD)`
`VALUES (1100, 104, '7/20/2015', 'Brass Lamps', 'Lamps', 1200);`

Results

Messages

	ItemID	StoreID	PurchasingAgentID	PurchaseDate	ItemDescription	Category	PriceUSD
1	500	1050	101	2014-12-10	Antique Large Bureaus	Furniture	13415
2	505	1050	102	2014-12-12	Porcelain Lamps	Lamps	13300
3	510	1200	104	2014-12-15	Gold Rim Design China	Tableware	38500
4	515	1200	104	2014-12-16	Gold Rim Design Serving Dishes	Tableware	3200
5	520	1050	102	2015-04-07	QE Dining Set	Furniture	14300
6	525	1100	103	2015-05-18	Misc Linen	Linens	88545
7	530	1000	103	2015-05-19	Large Masks	Decorations	22135
8	535	1100	104	2015-05-20	Willow Design China	Tableware	147575
9	540	1100	104	2015-05-20	Willow Design Serving Dishes	Tableware	12040
10	545	1150	102	2015-06-14	Woven Goods	Decorations	1200
11	550	1150	101	2015-06-16	Antique Leather Chairs	Furniture	5375
12	555	1100	104	2015-07-15	Willow Design Serving Dishes	Tableware	4500
13	560	1000	103	2015-07-17	Large Bureau	Furniture	9500
14	565	1100	104	2015-07-20	Brass Lamps	Lamps	1200

Query executed successfully.

DASH-SPECTRE (14.0 RTM)

SHIPPER TABLE (7-63)

- `Insert into Shipper values ('ABC Trans-Oceanic', '800-234-5656', '800-234-5659', 'Sales@ABCTransOceanic.com', 'Jonathan')`
- `Insert into Shipper values ('International', '800-123-8898', '800-123-8899', 'Sales@International.com', 'Marylin')`
- `Insert into Shipper values ('Worldwide', '800-123-4567', '800-123-4569', 'Sales@worldwide.com', 'Jose')`

Results

Messages

	ShipperID	ShipperName	Phone	Fax	EmailAddress	Contact
1	1	ABC Trans-Oceanic	800-234-5656	800-234-5659	Sales@ABCTransOceanic.com	Jonathan
2	2	International	800-123-8898	800-123-8899	Sales@International.com	Marylin
3	3	Worldwide	800-123-4567	800-123-4569	Sales@worldwide.com	Jose

✓

Query executed successfully.

DASH-SPECTRE (14.0 RTM)

SHIPMENT TABLE (7-64)

- **INSERT INTO SHIPMENT**
VALUES (1, 103, 2010651, 'Manila', 'Los Angeles', '10-Dec-14', '10-Dec-14', '15-Mar-15')
- **INSERT INTO SHIPMENT**
VALUES (1, 104, 2011012, 'Hong Kong', 'Seattle', '10-Jan-15', '12-Jan-15', '20-Mar-15')
- **INSERT INTO SHIPMENT**
VALUES (3, 103, 49100300, 'Manila', 'Los Angeles', '05-May-15', '05-May-15', '17-Jun-15')
- **INSERT INTO SHIPMENT**
VALUES (2, 104, 399400, 'Singapore', 'Portland', '02-Jun-15', '04-Jun-15', '17-Jun-15')
- **INSERT INTO SHIPMENT**
VALUES (3, 103, 84899440, 'Lima', 'Los Angeles', '10-Jul-15', '10-Jul-15', '28-Jul-15')
- **INSERT INTO SHIPMENT**
VALUES (2, 104, 488955, 'Singapore', 'Portland', '05-Aug-15', '09-Aug-15', '11-Sep-15')

Results		Messages							
	ShipmentID	ShipperID	PurchasingAgentID	ShipperInvoiceNumber	Origin	Destination	ScheduledDepartureDate	ActualDepartureDate	EstimatedArrivalDate
1	100	1	103	2010651	Manila	Los Angeles	2014-12-10	2014-12-10	2015-03-15
2	101	1	104	2011012	Hong Kong	Seattle	2015-01-10	2015-01-12	2015-03-20
3	102	3	103	49100300	Manila	Los Angeles	2015-05-05	2015-05-05	2015-06-17
4	103	2	104	399400	Singapore	Portland	2015-06-02	2015-06-04	2015-06-17
5	104	3	103	84899440	Lima	Los Angeles	2015-07-10	2015-07-10	2015-07-28
6	105	2	104	488955	Singapore	Portland	2015-08-05	2015-08-09	2015-09-11

Query executed successfully. DASH-SPECTRE (14.0 RTM) DASH-SPECTRE\ddalvi (54) MI 00:00:00

SHIPMENT_ITEM TABLE (7-65)

- **INSERT INTO SHIPMENT_ITEM** (ShipmentID, ShipmentItemID, ItemID, InsuredValue)
VALUES (100, 1, 505, 15000);
- **INSERT INTO SHIPMENT_ITEM** (ShipmentID , ShipmentItemID, ItemID, InsuredValue)
VALUES (101, 2, 510, 40000);
- **INSERT INTO SHIPMENT_ITEM** (ShipmentID , ShipmentItemID, ItemID, InsuredValue)
VALUES (102, 3, 520, 15000);
- **INSERT INTO SHIPMENT_ITEM** (ShipmentID , ShipmentItemID, ItemID, InsuredValue)
VALUES (103, 4, 525, 90000);
- **INSERT INTO SHIPMENT_ITEM** (ShipmentID , ShipmentItemID, ItemID, InsuredValue)
VALUES (104, 5, 530, 25000);

- `INSERT INTO SHIPMENT_ITEM (ShipmentID, ShipmentItemID,ItemID,InsuredValue)
VALUES (105,6,535,150000);`

Results		Messages		
	ShipmentID	ShipmentItemID	ItemID	InsuredValue
1	100	1	500	15000
2	101	2	510	40000
3	102	3	520	15000
4	103	4	525	90000
5	104	5	530	25000
6	105	6	535	150000

✓ Query executed successfully.

SHIPMENT RECEIPT TABLE

- `INSERT INTO SHIPMENT_RECEIPT (ShipmentID, ItemID, ReceivingAgentID,
ReceiptDate, ReceiptTime,ReceiptQuantity,isReceivedUndamaged,DamageNotes)
VALUES (100,500,105, '17-Mar-15', '10:00 AM',3, 'Yes',NULL);`
- `INSERT INTO SHIPMENT_RECEIPT (ShipmentID, ItemID, ReceivingAgentID,
ReceiptDate, ReceiptTime,ReceiptQuantity,isReceivedUndamaged,DamageNotes)
VALUES (100,505,105, '17-Mar-15', '10:00 AM',50, 'Yes',NULL);`
- `INSERT INTO SHIPMENT_RECEIPT (ShipmentID, ItemID, ReceivingAgentID,
ReceiptDate, ReceiptTime,ReceiptQuantity,isReceivedUndamaged,DamageNotes)
VALUES (101,510,105, '23-Mar-15', '03:30 PM',100, 'Yes',NULL);`
- `INSERT INTO SHIPMENT_RECEIPT (ShipmentID, ItemID, ReceivingAgentID,
ReceiptDate, ReceiptTime,ReceiptQuantity,isReceivedUndamaged,DamageNotes)
VALUES (101,515,105, '23-Mar-15', '03:30 PM',10, 'Yes',NULL);`
- `INSERT INTO SHIPMENT_RECEIPT (ShipmentID, ItemID, ReceivingAgentID,
ReceiptDate, ReceiptTime,ReceiptQuantity,isReceivedUndamaged,DamageNotes)
VALUES (102,520,106, '19-Jun-15', '10:15 AM',1, 'No', 'One Leg on one chair
broken');`
- `INSERT INTO SHIPMENT_RECEIPT (ShipmentID, ItemID, ReceivingAgentID,
ReceiptDate, ReceiptTime,ReceiptQuantity,isReceivedUndamaged,DamageNotes)
VALUES (103,525,106, '20-Jul-15', '02:20 AM',1000, 'Yes',NULL);`
- `INSERT INTO SHIPMENT_RECEIPT (ShipmentID, ItemID, ReceivingAgentID,
ReceiptDate, ReceiptTime,ReceiptQuantity,isReceivedUndamaged,DamageNotes)
VALUES (103,530,106, '20-Jul-15', '02:20 AM',100, 'Yes', NULL);`

- `INSERT INTO SHIPMENT_RECEIPT (ShipmentID, ItemID, ReceivingAgentID, ReceiptDate, ReceiptTime, ReceiptQuantity, isReceivedUndamaged, DamageNotes) VALUES (103, 535, 106, '20-Jul-15', '2:20 AM', 100, 'Yes', NULL);`
- `INSERT INTO SHIPMENT_RECEIPT (ShipmentID, ItemID, ReceivingAgentID, ReceiptDate, ReceiptTime, ReceiptQuantity, isReceivedUndamaged, DamageNotes) VALUES (103, 540, 106, '20-Jul-15', '2:20 AM', 10, 'Yes', NULL);`
- `INSERT INTO SHIPMENT_RECEIPT (ShipmentID, ItemID, ReceivingAgentID, ReceiptDate, ReceiptTime, ReceiptQuantity, isReceivedUndamaged, DamageNotes) VALUES (104, 545, 105, '29-Jul-15', '9:00 PM', 100, 'Yes', NULL);`
- `INSERT INTO SHIPMENT_RECEIPT (ShipmentID, ItemID, ReceivingAgentID, ReceiptDate, ReceiptTime, ReceiptQuantity, isReceivedUndamaged, DamageNotes) VALUES (104, 550, 105, '29-Jul-15', '9:00 PM', 5, 'Yes', NULL);`
- `INSERT INTO SHIPMENT_RECEIPT (ShipmentID, ItemID, ReceivingAgentID, ReceiptDate, ReceiptTime, ReceiptQuantity, isReceivedUndamaged, DamageNotes) VALUES (105, 555, 106, '14-Sep-15', '2:45 PM', 4, 'Yes', NULL);`
- `INSERT INTO SHIPMENT_RECEIPT (ShipmentID, ItemID, ReceivingAgentID, ReceiptDate, ReceiptTime, ReceiptQuantity, isReceivedUndamaged, DamageNotes) VALUES (105, 560, 106, '14-Sep-15', '2:45 PM', 1, 'Yes', NULL);`
- `INSERT INTO SHIPMENT_RECEIPT (ShipmentID, ItemID, ReceivingAgentID, ReceiptDate, ReceiptTime, ReceiptQuantity, isReceivedUndamaged, DamageNotes) VALUES (105, 560, 106, '14-Sep-15', '2:45 PM', 10, 'No', 'Base of one lamp scratched');`

Results		Messages							
	ReceiptNumber	ShipmentID	ItemID	ReceivingAgentID	ReceiptDate	ReceiptTime	ReceiptQuantity	isReceivedUndamaged	DamageNotes
1	200001	100	500	105	2015-03-17	10:00:00.0000000	3	Yes	NULL
2	200002	100	505	105	2015-03-17	10:00:00.0000000	50	Yes	NULL
3	200003	101	510	105	2015-03-23	15:30:00.0000000	100	Yes	NULL
4	200004	101	515	105	2015-03-23	15:30:00.0000000	10	Yes	NULL
5	200005	102	520	106	2015-06-19	10:15:00.0000000	1	No	One Leg on one chair broken
6	200006	103	525	106	2015-07-20	02:20:00.0000000	1000	Yes	NULL
7	200007	103	530	106	2015-07-20	02:20:00.0000000	100	Yes	NULL
8	200008	103	535	106	2015-07-20	02:20:00.0000000	100	Yes	NULL
9	200009	103	540	106	2015-07-20	02:20:00.0000000	10	Yes	NULL
10	200010	104	545	105	2015-07-29	21:00:00.0000000	100	Yes	NULL
11	200011	104	550	105	2015-07-29	21:00:00.0000000	5	Yes	NULL
12	200012	105	555	106	2015-09-14	14:45:00.0000000	4	Yes	NULL
13	200013	105	560	106	2015-09-14	14:45:00.0000000	1	Yes	NULL
14	200014	105	560	106	2015-09-14	14:45:00.0000000	10	No	Base of one lamp scratched

Query executed successfully.

DASH-SPECTRE (14.0 RTM) | DASH-SPECTRE\ddalvi (54) | MI | 00:00:00

- I. Write an **UPDATE** statement to change values of **STORE.City** from **New York City** to **NYC**.

```
update STORE
set City='NYC'
where City='New York City'
Select * from STORE
```

	StoreID	StoreName	City	Country	Phone	Fax	EmailAddress	Contact
1	1000	Eastern Sales	Singapore	Singapore	65-543-1233	65-543-1239	Sales@EasternSales.com.sg	Jeremy
2	1050	Eastern Treasures	Manila	Philippines	63-2-654-2344	63-2-654-2349	Sales@EasternTreasures.com.ph	Gracielle
3	1100	Jade Antiques	Singapore	Singapore	65-543-3455	65-543-3459	Sales@JadeAntiques.com.sg	Swee Lai
4	1150	Andes Treasures	Lima	Peru	51-14-765-4566	51-14-765-4569	Sales@AndesTreasures.com.pe	Juan Carlos
5	1200	Eastern Treasures	New Delhi	India	91-11-987-6788	91-11-987-6789	Sales@EasternTreasures.com.in	Deepinder
6	1250	European Imports	NYC	United States	800-432-8766	800-432-8769	Sales@EuropeanImports.com.sg	Marcello

Query executed successfully. DASH-SPECTRE (14.0 RTM) DASH-SPECT

- J. Create and **INSERT** new data records to record a **SHIPMENT** and the **SHIPMENT_ITEMS** for that **SHIPMENT**. Then write a **DELETE** statement(s) to delete that **SHIPMENT** and all of the items on that **SHIPMENT**. How many **DELETE** statements did you have to use? Why?

```
Insert into Shipment values (2,104,511700,'Singapore','Portland','15-Sep-12',NULL,NULL)
```

	ShipmentID	ShipperID	PurchasingAgentID	ShipperInvoiceNumber	Origin	Destination	ScheduledDepartureDate	ActualDepartureDate	EstimatedArrivalDate
1	100	1	103	2010651	Manila	Los Angeles	2014-12-10	2014-12-10	2015-03-15
2	101	1	104	2011012	Hong Kong	Seattle	2015-01-10	2015-01-12	2015-03-20
3	102	3	103	49100300	Manila	Los Angeles	2015-05-05	2015-05-05	2015-06-17
4	103	2	104	399400	Singapore	Portland	2015-06-02	2015-06-04	2015-06-17
5	104	3	103	84899440	Lima	Los Angeles	2015-07-10	2015-07-10	2015-07-28
6	105	2	104	488955	Singapore	Portland	2015-08-05	2015-08-09	2015-09-11
7	106	2	104	511700	Singapore	Portland	2012-09-15	NULL	NULL
8	107	2	104	511700	Singapore	Portland	2012-09-15	NULL	NULL

Query executed successfully. DASH-SPECTRE (14.0 RTM) DASH-SPECTRE\ddalvi (54) MI 00:00:00

```
Insert into Shipment_item values (107,7,565,1500)
```

	ShipmentID	ShipmentItemID	ItemID	InsuredValue
1	100	1	500	15000
2	101	2	510	40000
3	102	3	520	15000
4	103	4	525	90000
5	104	5	530	25000
6	105	6	535	150000
7	107	7	565	1500

Query executed successfully.

We have to use only one Delete statement because Shipment_Item table has a foreign key ShipmentID with Primary key in the table Shipment. Also, we have used On Delete cascade statement when creating the foreign key table.

```
Delete from Shipment
Where ShipmentID=107
```

Results		Messages							
	ShipmentID	ShipperID	PurchasingAgentID	ShipperInvoiceNumber	Origin	Destination	ScheduledDepartureDate	ActualDepartureDate	EstimatedArrivalDate
1	100	1	103	2010651	Manila	Los Angeles	2014-12-10	2014-12-10	2015-03-15
2	101	1	104	2011012	Hong Kong	Seattle	2015-01-10	2015-01-12	2015-03-20
3	102	3	103	49100300	Manila	Los Angeles	2015-05-05	2015-05-05	2015-06-17
4	103	2	104	399400	Singapore	Portland	2015-06-02	2015-06-04	2015-06-17
5	104	3	103	84899440	Lima	Los Angeles	2015-07-10	2015-07-10	2015-07-28
6	105	2	104	488955	Singapore	Portland	2015-08-05	2015-08-09	2015-09-11

Query executed successfully.

DASH-SPECTRE (14.0 RTM)

DASH-SPECTRE\ddalv (54)


MI

00:00:00

Results

Messages

	ShipmentID	ShipmentItemID	ItemID	InsuredValue
1	100	1	500	15000
2	101	2	510	40000
3	102	3	520	15000
4	103	4	525	90000
5	104	5	530	25000
6	105	6	535	150000

 Query executed successfully.

- K. Write an SQL statement to create a view called *PurchaseSummaryView* that shows only *ITEM.ItemID*, *ITEM.PurchaseDate*, *ITEM.ItemDescription*, and *ITEM.PriceUSD*. Run the statement to create the view, and then test the view with an appropriate SQL *SELECT* statement.**

```
Create view PurchaseSummaryView as
Select ItemID,PurchaseDate,ItemDescription,PriceUSD
from Item
select * from PurchaseSummaryView
```

Results		Messages		
	ItemID	PurchaseDate	ItemDescription	PriceUSD
1	500	2014-12-10	Antique Large Bureaus	13415
2	505	2014-12-12	Porcelain Lamps	13300
3	510	2014-12-15	Gold Rim Design China	38500
4	515	2014-12-16	Gold Rim Design Serving Dishes	3200
5	520	2015-04-07	QE Dining Set	14300
6	525	2015-05-18	Misc Linen	88545
7	530	2015-05-19	Large Masks	22135
8	535	2015-05-20	Willow Design China	147575
9	540	2015-05-20	Willow Design Serving Dishes	12040
10	545	2015-06-14	Woven Goods	1200
11	550	2015-06-16	Antique Leather Chairs	5375
12	555	2015-07-15	Willow Design Serving Dishes	4500
13	560	2015-07-17	Large Bureau	9500
14	565	2015-07-20	Brass Lamps	1200

✓ Query executed successfully.

- L. *Create and test a user-defined function named StoreContactAndPhone that combines two parameters named StoreContact and ContactPhone into a concatenated data field formatted StoreContact: ContactPhone (including the colon and space).*

```
CREATE FUNCTION dbo.StoreContactAndPhone
( @varStoreContact char(25),
  @varStorePhone char(25) )
RETURNS Varchar(60)
AS
BEGIN
DECLARE @varContactAndPhone varchar(60);
SELECT @varContactAndPhone=RTRIM(@varStoreContact)+' ':'+RTRIM(@varStorePhone);
RETURN @varContactAndPhone;
END
```

```
Select StoreName, dbo.StoreContactAndPhone(Contact,Phone) as
ContactPhone, ItemID, PurchaseDate, PriceUSD
From Store S, Item I
Where S.StoreID=I.StoreID
```

Results		Messages			
	StoreName	ContactPhone	ItemID	PurchaseDate	PriceUSD
2	Eastern Treasures	Gracielle: 63-2-654-2344	505	2014-12-12	13300
3	Eastern Treasures	Deepinder: 91-11-987-6788	510	2014-12-15	38500
4	Eastern Treasures	Deepinder: 91-11-987-6788	515	2014-12-16	3200
5	Eastern Treasures	Gracielle: 63-2-654-2344	520	2015-04-07	14300
6	Jade Antiques	Swee Lai: 65-543-3455	525	2015-05-18	88545
7	Eastern Sales	Jeremy: 65-543-1233	530	2015-05-19	22135
8	Jade Antiques	Swee Lai: 65-543-3455	535	2015-05-20	147575
9	Jade Antiques	Swee Lai: 65-543-3455	540	2015-05-20	12040
10	Andes Treasures	Juan Carlos: 51-14-765-4566	545	2015-06-14	1200
11	Andes Treasures	Juan Carlos: 51-14-765-4566	550	2015-06-16	5375
12	Jade Antiques	Swee Lai: 65-543-3455	555	2015-07-15	4500
13	Eastern Sales	Jeremy: 65-543-1233	560	2015-07-17	9500
14	Jade Antiques	Swee Lai: 65-543-3455	565	2015-07-20	1200

✓ Query executed successfully.

M. Write an SQL statement to create a view called *StorePurchaseHistoryView* that shows *STORE.StoreName*, *STORE.Phone*, *STORE.Contact*, *ITEM.ItemID*, *ITEM.PurchaseDate*, *ITEM.ItemDescription*, and *ITEM.PriceUSD*. Run the statement to create the view, and then test the view with an appropriate SQL *SELECT* statement.

```
Create view StorePurchaseHistoryView as
SELECT S.StoreName, S.Phone, S.Contact, I.ItemID, I.PurchaseDate,
I.ItemDescription, I.PriceUSD
FROM STORE AS S, ITEM AS I
WHERE S.StoreID=I.StoreID;

SELECT * FROM StorePurchaseHistoryView
```

Results		Messages					
	StoreName	Phone	Contact	ItemID	PurchaseDate	ItemDescription	PriceUSD
1	Eastern Treasures	63-2-654-2344	Gracielle	500	2014-12-10	Antique Large Bureaus	13415
2	Eastern Treasures	63-2-654-2344	Gracielle	505	2014-12-12	Porcelain Lamps	13300
3	Eastern Treasures	91-11-987-6788	Deepinder	510	2014-12-15	Gold Rim Design China	38500
4	Eastern Treasures	91-11-987-6788	Deepinder	515	2014-12-16	Gold Rim Design Serving Dishes	3200
5	Eastern Treasures	63-2-654-2344	Gracielle	520	2015-04-07	QE Dining Set	14300
6	Jade Antiques	65-543-3455	Swee Lai	525	2015-05-18	Misc Linen	88545
7	Eastern Sales	65-543-1233	Jeremy	530	2015-05-19	Large Masks	22135
8	Jade Antiques	65-543-3455	Swee Lai	535	2015-05-20	Willow Design China	147575
9	Jade Antiques	65-543-3455	Swee Lai	540	2015-05-20	Willow Design Serving Dishes	12040
10	Andes Treasures	51-14-765-4566	Juan Carlos	545	2015-06-14	Woven Goods	1200
11	Andes Treasures	51-14-765-4566	Juan Carlos	550	2015-06-16	Antique Leather Chairs	5375
12	Jade Antiques	65-543-3455	Swee Lai	555	2015-07-15	Willow Design Serving Dishes	4500
13	Eastern Sales	65-543-1233	Jeremy	560	2015-07-17	Large Bureau	9500
14	Jade Antiques	65-543-3455	Swee Lai	565	2015-07-20	Brass Lamps	1200

Query executed successfully. DASH-SPECTRE (14.0 RTM)

- N. Write an SQL statement to create a view called *StoreContactPurchaseHistoryView* that shows *STORE.StoreName*, the concatenated result of *STORE.Phone* and *STORE.Contact* from the *StoreContactAndPhone* function, *ITEM.ItemID*, *ITEM.PurchaseDate*, *ITEM.ItemDescription*, and *ITEM.PriceUSD*. Run the statement to create the view, and then test the view with an appropriate SQL *SELECT* statement.

```
Create View StoreContactPurchaseHistoryView as
Select StoreName, dbo.StoreContactAndPhone(Contact,Phone) as
ContactPhone,ItemID,PurchaseDate,PriceUSD
From Store S,Item I
Where S.StoreID=I.StoreID
```

```
SELECT * from StoreContactPurchaseHistoryView
```

Results		Messages			
	StoreName	ContactPhone	ItemID	PurchaseDate	PriceUSD
1	Eastern Treasures	Gracielle: 63-2-654-2344	500	2014-12-10	13415
2	Eastern Treasures	Gracielle: 63-2-654-2344	505	2014-12-12	13300
3	Eastern Treasures	Deepinder: 91-11-987-6788	510	2014-12-15	38500
4	Eastern Treasures	Deepinder: 91-11-987-6788	515	2014-12-16	3200
5	Eastern Treasures	Gracielle: 63-2-654-2344	520	2015-04-07	14300
6	Jade Antiques	Swee Lai: 65-543-3455	525	2015-05-18	88545
7	Eastern Sales	Jeremy: 65-543-1233	530	2015-05-19	22135
8	Jade Antiques	Swee Lai: 65-543-3455	535	2015-05-20	147575
9	Jade Antiques	Swee Lai: 65-543-3455	540	2015-05-20	12040
10	Andes Treasures	Juan Carlos: 51-14-765-4566	545	2015-06-14	1200
11	Andes Treasures	Juan Carlos: 51-14-765-4566	550	2015-06-16	5375
12	Jade Antiques	Swee Lai: 65-543-3455	555	2015-07-15	4500
13	Eastern Sales	Jeremy: 65-543-1233	560	2015-07-17	9500
14	Jade Antiques	Swee Lai: 65-543-3455	565	2015-07-20	1200

Query executed successfully.

- O. Write an SQL statement to create a view called *StoreHistoryView* that sums the *PriceUSD* column of *StorePurchaseHistoryView* for each store into a column named *TotalPurchases*. Run the statement to create the view, and then test the view with an appropriate SQL *SELECT* statement. (Hint: Assume unique store names.)

```
CREATE VIEW StoreHistoryView as
SELECT StoreName, SUM(PriceUSD) AS TotalPurchases
FROM StorePurchaseHistoryView
GROUP BY StoreName;
SELECT * FROM StoreHistoryView
```

Results Messages		
	StoreName	TotalPurchases
1	Andes Treasures	6575
2	Eastern Sales	31635
3	Eastern Treasures	82715
4	Jade Antiques	253860

✓ Query executed successfully.

- P. Write an SQL statement to create a view called *MajorSources* that uses *StoreHistoryView* and selects only those stores that have *TotalPurchases* greater than 100000. Run the statement to create the view, and then test the view with an appropriate SQL *SELECT* statement.

```
CREATE VIEW MajorSources as
SELECT StoreName, TotalPurchases
FROM StoreHistoryView
WHERE TotalPurchases > 100000
GROUP BY StoreName, TotalPurchases;

Select * from MajorSources
```

Results Messages		
	StoreName	TotalPurchases
1	Jade Antiques	253860

✓ Query executed successfully.