QM-7093-01 ENTERPRISE DATA SYSTEMS
CASE STUDY (CS-6) – NOAH L. SCHRICK - 1492657

Instructor: Dr. Ismail Abdulrashid,

Instructions:

In this case study, **M1_CH02** database needs to be used. You can find the Review Questions in the textbook at the end of Chapter 6: Questions A-E of the Morgan Importing Project

- Answer questions A through E of the Morgan Importing Project found on page 321.
- Create and set up the M1_CH02 database with the given SQL files.
- Do not include the result table unless specifically directed to.
- Include at least one line of white space between answers.
- Submit thru Harvey drop box
- Deliverable: You are expected to submit
    - A single SQL script file (.sql) prepared and saved in SQL Server Management Studio that includes your SQL statements that answer each of the questions in order.
    - This word file that you copied all of your SQL script (no result tables) from your SQL file.
- Each query should start with a comment line that looks like the following (last character corresponding to question number):
    - /* *** CS6-MI.A *** */
    - This line should follow the SQL statement that is your answer to the particular question (e.g. B, C, D, ...)
- You should include at least one line of white space between your answer SQL statements
- Do not include the result table unless it is said so!
- Check Harvey for the due date!

Your answer should look like this:

```
/* Your Names-Group Name */
/* *** CS1-2.17 *** */
SELECT      SKU, SKU_Description
FROM        INVENTORY;

/* *** CS1-2.18 *** */
SELECT      SKU, SKU_Description
FROM        INVENTORY;

/* *** CS1-2.19 *** */
SELECT      SKU, SKU_Description
FROM        INVENTORY;
```

Please write your solution below:

/* Noah L. Schrick */

/* *** CS6-MI.A *** */

EMPLOYEE

Column Name	Type	Key	NULL Status	Remarks
Employee_ID	Int	Primary Key	NOT NULL	Surrogate Key
Employee_Fname	Char(25)	Alternate Key	NOT NULL	AK.1
Employee_Lname	Char(25)	Alternate Key	NOT NULL	AK.2
Department	Char(10)	No	NOT NULL	IN('Receiving', 'Purchasing')

STORE

Column Name	Type	Key	NULL Status	Remarks
StoreID	Int	Primary Key	NOT NULL	
City	Char (30)	No	NOT NULL	
Country	Char(30)	No	NOT NULL	
Phone	E.164 FORMAT STRING	No	NULL	[+][country code][area code][local phone number]
Email	Char(30)	No	NULL	

ITEM

Column Name	Type	Key	NULL Status	Remarks
ItemID	Int	Primary Key	NOT NULL	
StoreID	Int	Foreign Key	NULL	May no longer be carried
PurchaserID	Int	Foreign Key	NULL	
PurchaseDate	Date	No	NULL	
Description	Char(30)	No	NULL	
Quantity	Int	No	NOT NULL	
CostPer	Double	No	NOT NULL	

SHIPPER

Column Name	Type	Key	NULL Status	Remarks
ShipperID	Int	Primary Key	NOT NULL	
ShipperName	Char(30)	Alternate Key	NOT NULL	AK.1
Phone	E.164 FORMAT STRING	No	NULL	[+][country code][area code][local phone number]
Email	Char(30)	No	NULL	

SHIPMENT

Column Name	Type	Key	NULL Status	Remarks
ShipmentID	Int	Composite Key	NOT NULL	Surrogate Key
ShipmentItemID	Int	Composite Key, Foreign Key	NOT NULL	Several items may be purchased without all being in the same shipment
ShipperID	Int	Foreign Key	NOT NULL	
ShipDate	Date	No	NULL	
ETADate	Char(30)	No	NULL	
InsuranceValue	Int	No	NULL	

SHIPPED_ITEM

Column Name	Type	Key	NULL Status	Remarks
ShipmentItemID	Int	Primary Key	NOT NULL	
ShipmentID	Int	Foreign Key	NOT NULL	
ItemID	Int	Foreign Key	NOT NULL	

SHIPMENT_RECEIPT

Column Name	Type	Key	NULL Status	Remarks
ShippedItemID	Int	Composite Key, Foreign Key	NOT NULL	
ShipmentID	Int	Composite Key, Foreign Key	NOT NULL	
ReceivedBy	Int	Foreign Key	NOT NULL	Employee ID
ReceiptID	Int	No	NOT NULL	
Condition	Char(30)	No	NULL	
ShipDate	Date	No	NULL	
ETADate	Char(30)	No	NULL	

/* *** CS6-MI.B *** */

Weak entities are represented through the listing of “Composite Key” in a table’s Key column.

There are a few examples, mainly when it comes to shipping:

1) Shipment

   Since any number of items can be on a shipment, it is insufficient to have ShipmentID as the primary key. It is not a feasible, nor wise, design choice to have multiple item columns for a single ShipmentID to maintain uniqueness.

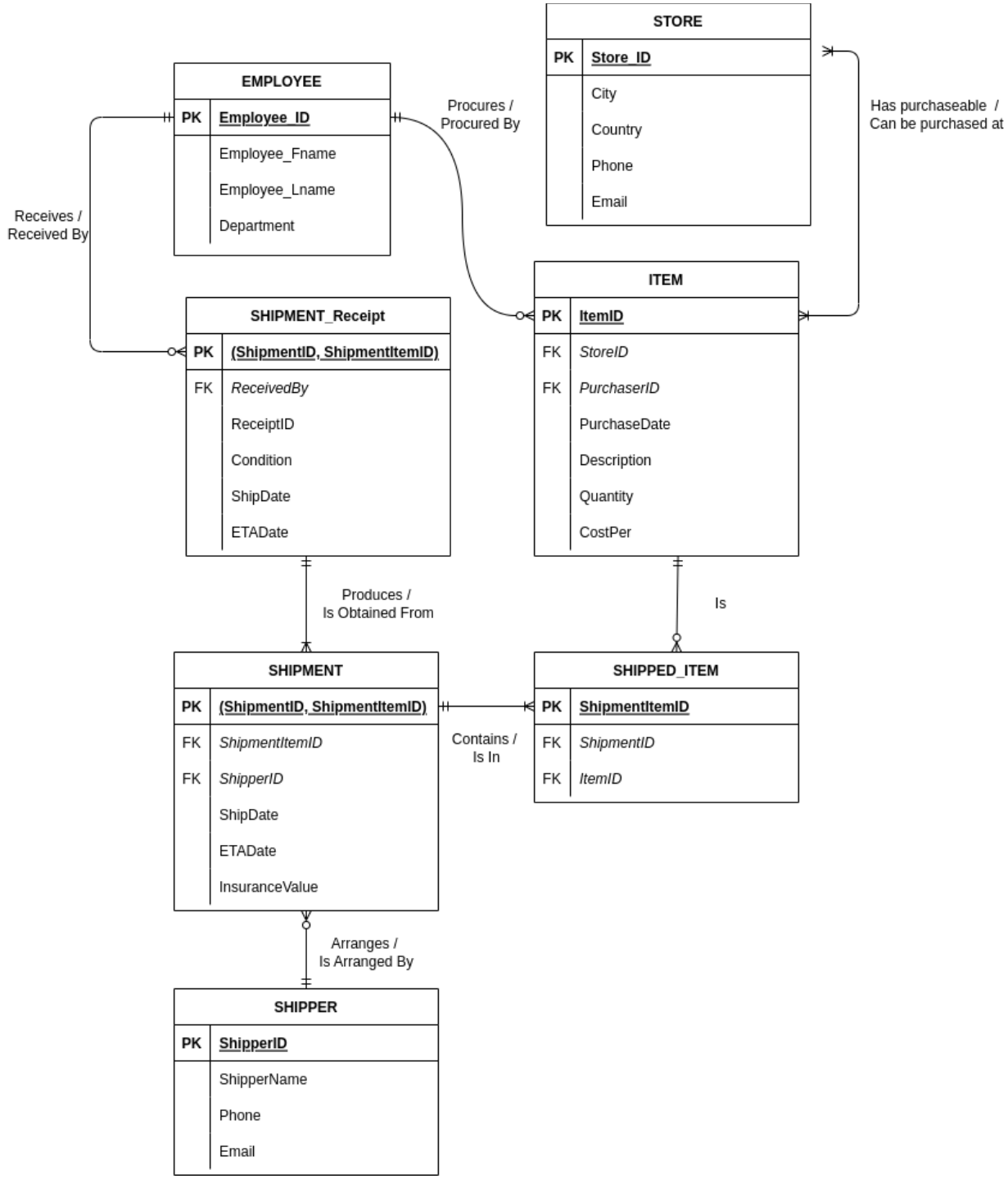
2) Shipment_Receipt

   Similar to the Shipment table, the Shipment_Receipt table also uses a composite key to handle the Condition column for any number of items on a shipment.

/* *** CS6-MI.C *** */

Supertypes and subtypes were not used for this database design.

/* *** CS6-MI.D *** */


STORE
PK
Store_ID
City
Country
Phone
Email
EMPLOYEE
PK
Employee_ID
Employee_Fname
Employee_Lname
Department
Procures /
Procured By
Has purchaseable /
Can be purchased at
Receives /
Received By
ITEM
PK
ItemID
FK
StoreID
FK
PurchaserID
PurchaseDate
Description
Quantity
CostPer
SHIPMENT_Receipt
PK
(ShipmentID, ShipmentItemID)
FK
ReceivedBy
ReceiptID
Condition
ShipDate
ETADate
Produces /
Is Obtained From
Is
SHIPMENT
PK
(ShipmentID, ShipmentItemID)
FK
ShipmentItemID
FK
ShipperID
ShipDate
ETADate
InsuranceValue
Contains /
Is In
SHIPPED_ITEM
PK
ShipmentItemID
FK
ShipmentID
FK
ItemID
Arranges /
Is Arranged By
SHIPPER
PK
ShipperID
ShipperName
Phone
Email

/* *** CS6-MI.E *** */

Minimum cardinalities present in this design:

- M-M
- M-O
- O-M

M-O and M-M

Parent Required	Action on Parent	Action on Child
Insert	None	Get a parent. Prohibit.
Modify	Cascading updates. Prohibit.	Allow if new FK matches existing parent. Prohibit.
Delete	Cascade delete. Prohibit.	None.

O-O and M-M

Child Required	Action on Parent	Action on Child
Insert	Get a child. Prohibit	None.
Modify	Update FK of >= 1 child. Prohibit.	Allow if not last child. Prohibit or find replacement.
Delete	None.	Allow if not last child. Prohibit or find replacement.