QM-7093-01 ENTERPRISE DATA SYSTEMS
EXAM 2 – 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 G found on pages 187-189 in Chapter 3 for the Queen Anne Curiosity Shop.
- Answer Questions A through G found on pages 209-210 in Chapter 4 for the Queen Anne Curiosity Shop.
- Answer Questions A and B found on page 262 in Chapter 5 for the Writer's Patrol Case.
- Answer Questions A through E found on page 265 in Chapter 5 for the Queen Anne Curiosity Shop.
- Answer Questions A through E found on page 321 in Chapter 6 for the Queen Anne Curiosity Shop.
- 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
- 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 */

Questions A through G found on pages 187-189 in Chapter 3 for the Queen Anne Curiosity Shop.

/* *** E2 – 3A *** */

(LastName, FirstName) → Phone Number

(InvoiceDate, InvoiceItem) → Price, Tax, Total

(Price, Tax) → Total

(PurchaseItem, PurchaseDate) → PurchasePrice

(Phone, PurchaseItem) → Vendor

/* *** E2 – 3B *** */

1. Not valid. LastName does not have the uniqueness nor the ability to determine any other attributes.

2. Not valid. LastName and FirstName do not have the uniqueness nor the ability to determine any other attributes.

3. Not valid. Phone does not have the uniqueness nor the ability to determine any other attributes.

4. Not valid. LastName, FirstName, and InvoiceDate do not have the uniqueness nor the ability to determine any other attributes. (EX: There are two occurrences of (Shire, Robert, 14-Dec-17)).

5. Not valid. LastName, FirstName, and InvoiceItem do not have the uniqueness nor the ability to determine any other attributes. (EX: There are two occurrences of (Goodyear, Katherine, Antique Chair) that have different totals)

6. Not valid. The two tables do not have an applicable foreign key.

7. Not valid. InvoiceDate as a foreign key is unable to determine InvoiceItem, Price, Tax, or Total.

8. Not valid. LastName, FirstName as primary keys, and InvoiceDate and InvoiceItem are not unique enough nor do they have the ability to determine Price, Tax, or Total. For example, there are two entries with the given keys that have different Price, Tax, and Total:

Goodyear, Katherine, 15-Jan-18, Antique Chair, 1250.00, 103.75, 1353.75

Goodyear, Katherine, 15-Jan-18, Antique Chair, 1750.00, 145.251, 895.25

/* *** E2 – 3C *** */

CUSTOMER(**<u>CustomerID</u>**, LastName, FirstName, Phone, Email)

SALE(**<u>SaleID</u>**, InvoiceDate, InvoiceItem, Price, Tax, Total, *CustomerID*)

Only one primary key is needed for each table, rather than needing composite keys.

/* *** E2 – 3D *** */

SALE(**<u>SaleID</u>**, InvoiceDate, *InvoiceItem*, CustomerID, Quantity, Tax, Total*)*

SALE_ITEM(**<u>InvoiceItem</u>**, Price, Vendor)

This allows for the item and its price to be stored separately alongside its vendor, and for the tax and quantity to be stored with the sale table. Since the tax and total are dependent on the sale itself and not the base item, they should be stored separate from the sale item table.

/* *** E2 – 3E *** /

1. Not valid. PurchaseItem is not unique nor does it have the ability to determine other attributes.

2. Valid for this set of data, but not valid long term. In this data, PurchaseItem and PurchasePrice are unique enough to determine all other attributes. However, this is not sustainable. If another item was purchased at the same location for the same price at a different date, it would no longer be valid.

3. Not valid. PurchaseItem and PurchaseDate are not able to determine other attributes. Two Antique Desks were purchased at 7-Nov-17, but at different purchase prices.

4. Not valid. PurchaseItem and Vendor are not able to determine other attributes.

5. Not valid. No specified foreign keys, and the composite keys are unable to determine other attributes.

6. Not valid. No specified foreign keys, and the composite keys are unable to determine other attributes.

7. Not valid. PurchaseItem, PurchaseDate, and Vendor are not unique enough nor do they have the ability to determine other attributes. For example, there were multiple Antique Desks purchased on 7-Nov-17 from European Specialities.

/* *** E2 – 3F *** */

PURCHASE(**PurchaseID,** PurchaseItem, PurchasePrice, PurchaseDate, *VendorID*)

VENDOR(**VendorID,** Vendor, Phone)

Only one primary key is needed for each table, instead of composite keys.

/* *** E2 – 3G *** */

SALE(***<u>SaleID</u>***, InvoiceDate, *InvoiceItem*, CustomerID, Quantity, Tax, Total*)*

SALE_ITEM(***<u>InvoiceItem</u>***, Price, Vendor)

PURCHASE(**<u>PurchaseID,</u>** *PurchaseItem*, PurchasePrice, PurchaseDate, *VendorID*)

VENDOR(**VendorID,** Vendor, Phone)

Answer Questions A through G found on pages 209-210 in Chapter 4 for the Queen Anne Curiosity Shop.

/* *** E2 – 4A *** */

1.

ZIP → (City, State)

(LastName, FirstName) → EmailAddress

(LastName, FirstName) → Phone

CustomerID → (LastName, FirstName, EmailAddress, Address, ZIP, Phone, ReferredBy)

Address → ZIP

Address → (City, State)

SaleID → (CustomerID, InvoiceDate, PreTaxTotal, Tax, Total)

(PreTaxTotal, Tax) → Total

(Vendor, Phone) → VendorID

(PurchaseItem, PurchasePrice) → VendorID

(PurchaseID, PurchaseItem) → (PurchasePrice, PurchaseDate, VendorID)

2.

PurchaseDate → → PurchaseItem

PurchaseDate → → PurchasePrice

3.

CustomerID in CUSTOMER

For this set of data, but not for long-term actual usage: LastName

For this set of data, but not for long-term actual usage: FirstName

For this set of data, but not for long-term actual usage: Phone

For this set of data, but not for long-term actual usage: Email

For this set of data, but not for long-term actual usage: Address

SaleID in SALE

For this set of data, but not for long-term actual usage: Total

For this set of data, but not for long-term actual usage: Tax

PurchaseID in SALE_ITEM

PurchaseID in PURCHASE

VendorID in VENDOR

For this set of data, but not for long-term actual usage: Vendor

For this set of data, but not for long-term actual usage: Phone

4.

CustomerID in CUSTOMER

SaleID in SALE

SaleItemID in SALE_ITEM

PurchaseID in PURCHASE

VendorID in VENDOR

5.

CustomerID in SALE

SaleID and PurchaseID in SALE_ITEM

VendorID in PURCHASE

/* *** E2 – 4B *** */

Will multiple people with the same last name be able to make purchases?

Will multiple people with the same first name be able to make purchases?

Will family members be able to use another family member’s phone number for purchases?

Will family members be able to use another family member’s email address for purchases?

Will family members be able to ship to the same address as another family member?

Will multiple customers buy the same set of items?

Will multiple customers buy different sets of items that add to the same total?

Will you use multiple vendors that have the same name?

Will subsidiary vendor companies list the main parent company’s phone number?

/* *** E2 – 4C *** */

No new tables need to be created, though some tables should be altered to remove columns.

The PURCHASE table can suffice without PurchaseItem. Since PurchaseID is a foreign key in SALE_ITEM, we can safely remove this column without losing data.

/* *** E2 – 4D *** */

Removing PurchaseItem from the PURCHASE table can handle the multivalue, multicolumn problem.

/* *** E2 – 4E *** */

They do not currently have the inconsistent data problem, however it is possible that it could occur. Enforcing database rules with referential integrity or checking for misspellings can alleviate this problem. It is unlikely that ZIP codes will change, but through normalization, a vendor changing its address or phone will not lead to issues.

/* *** E2 – 4F *** */

They do not currently have a null (missing) value data problem, however it is possible that it could occur. Enforcing database rules such as NOT NULL can alleviate this problem when appropriate.

/* *** E2 – 4E *** */

They do not currently have a general-purpose remarks problem, and it is unlikely they will have this issue unless tables are deliberately altered.

Answer Questions A and B found on page 262 in Chapter 5 for the Writer's Patrol Case.

/* *** E2 – 5A *** */

OWNER(**<u>LastName, FirstName</u>**, MI, Address, State, City, Zip, *LicNum*)

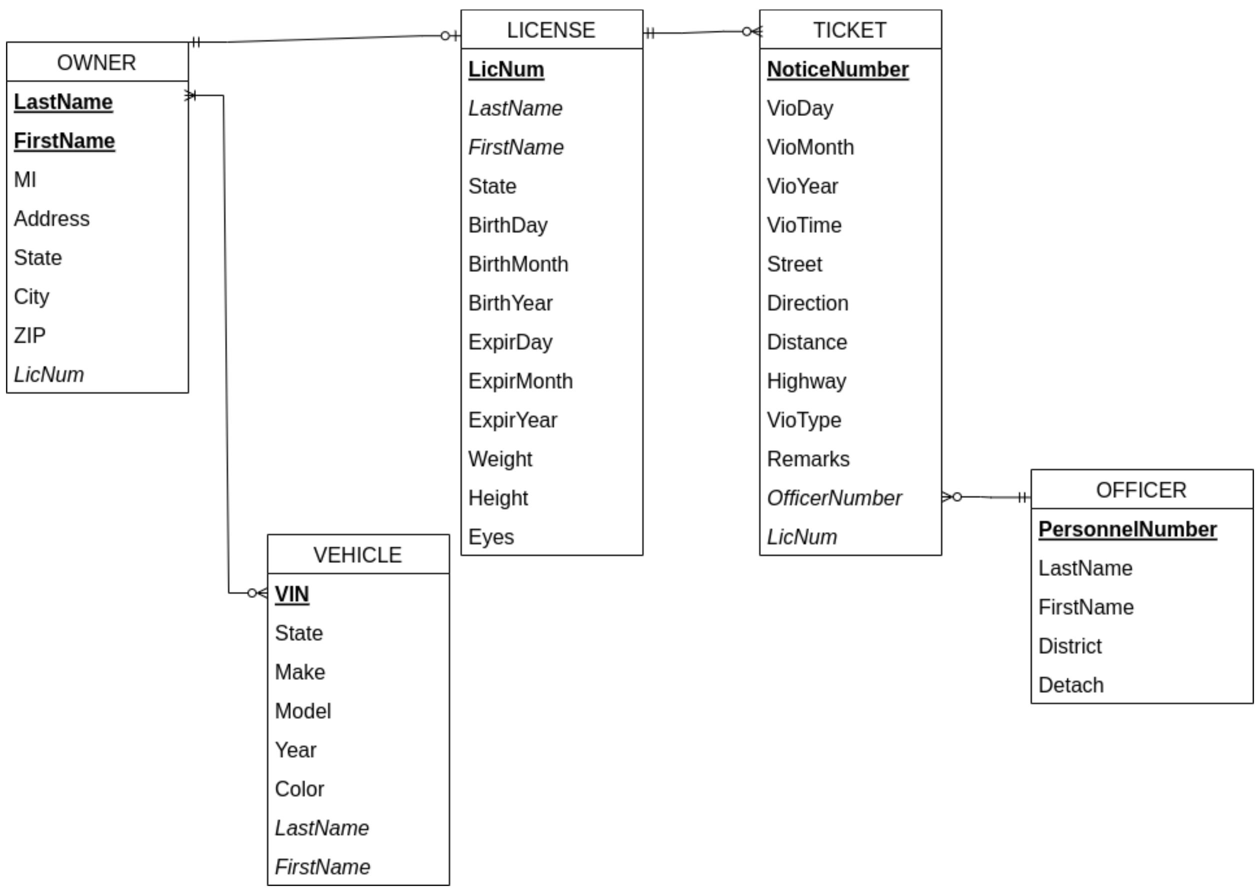
LICENSE(**<u>LicNum,</u>** *LastName, FirstName*, State, BirthDay, BirthMonth, BirthYear, ExpirDay, ExpirMonth, ExpirYear, Weight, Height, Eyes)

VEHICLE(**<u>VIN</u>**, State, Make, Model, Year, Color, *LastName, FirstName*)

OFFICER(**<u>PersonnelNumber,</u>** LastName, FirstName, District, Detach)

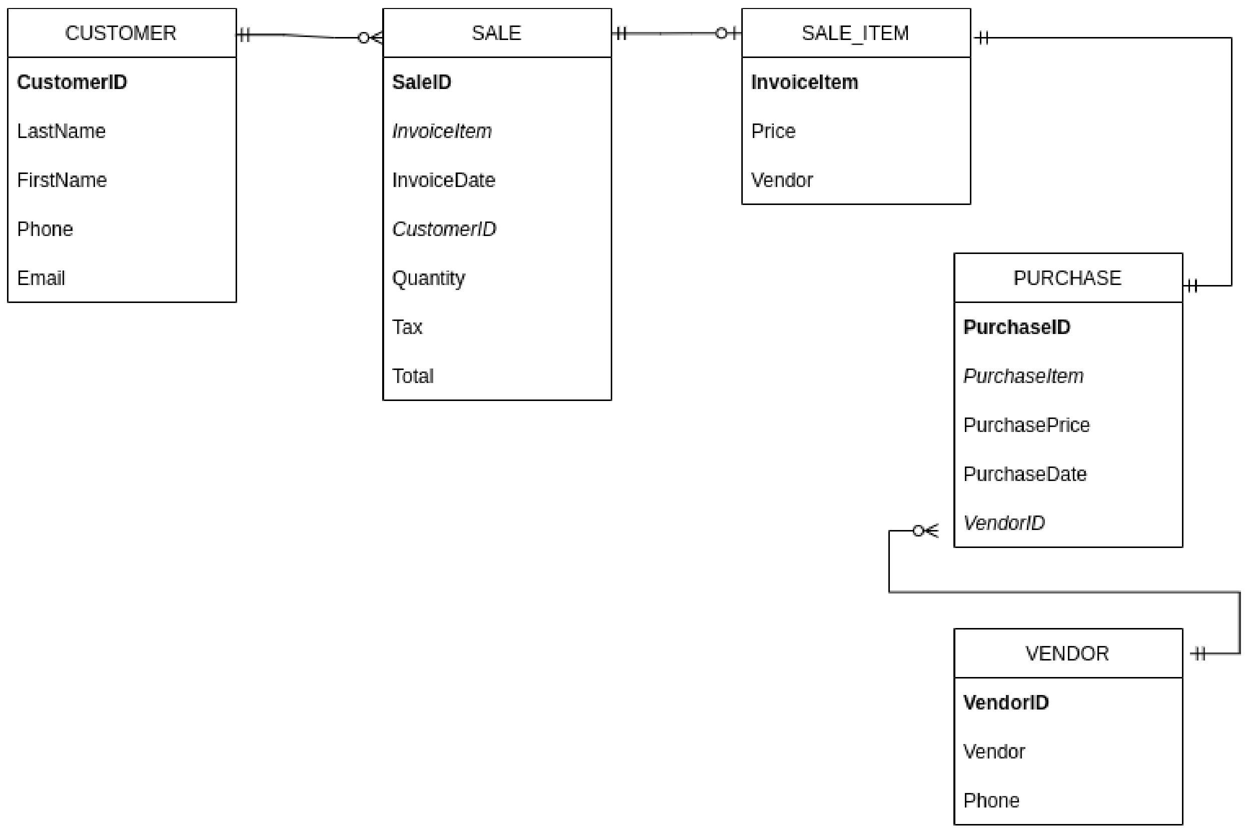
TICKET(**<u>NoticeNumber,</u>** VioDay, VioMonth, VioYear, VioTime, Street, Direction, Distance, Highway, VioType, Remarks, *OfficerNumber, LicNum*)

/* *** E2 – 5B *** */


OWNER
LastName
FirstName
MI
Address
State
City
ZIP
LicNum
LICENSE
LicNum
LastName
FirstName
State
BirthDay
BirthMonth
BirthYear
ExpirDay
ExpirMonth
ExpirYear
Weight
Height
Eyes
TICKET
NoticeNumber
VioDay
VioMonth
VioYear
VioTime
Street
Direction
Distance
Highway
VioType
Remarks
OfficerNumber
LicNum
OFFICER
PersonnelNumber
LastName
FirstName
District
Detach
VEHICLE
VIN
State
Make
Model
Year
Color
LastName
FirstName

Answer Questions A through E found on page 265 in Chapter 5 for the Queen Anne Curiosity Shop.

/*** * E2 – 5A *** */



A sale can only have one customer associated with it, and a sale requires a person to sell to.

A customer may register with a company, but may never decide to buy anything. They could also make multiple purchases.

Due to the way the database is currently designed, a sale can only have at most one sale item, since each item must be entered uniquely as its own column. A sale may not include an item.

However, a sale item is present in all sales, and since they are unique, they may only be associated with one sale.

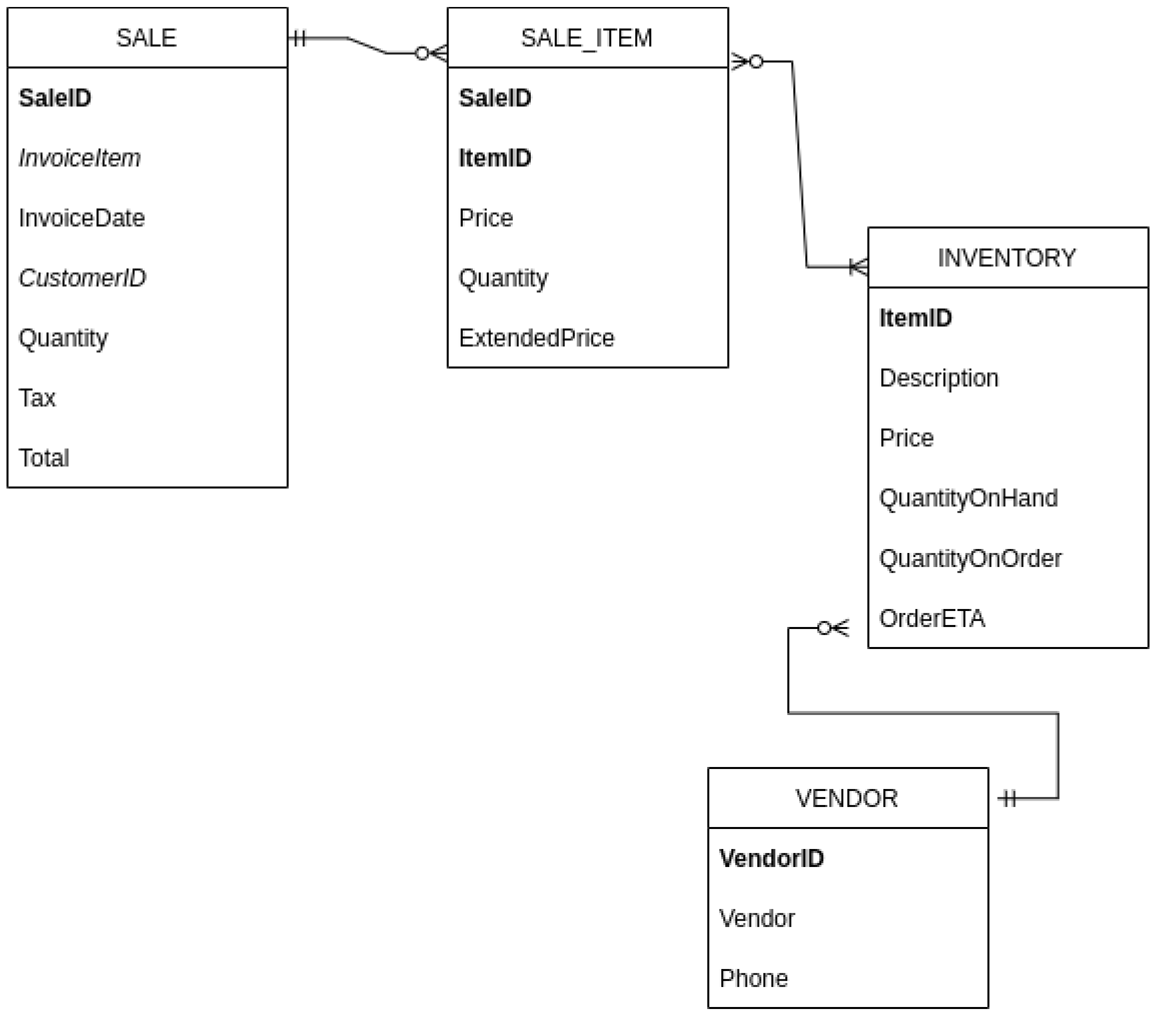
For this same reason, an item must be present in a purchase from a vendor, and it has to be unique, limiting its maximum cardinality to 1.

This applies to purchases and items as well.

A purchase only has one item, and as a result must only have one and exactly one vendor.

A vendor may be involved in zero, or many purchases.

/* *** E2 – 5B *** */



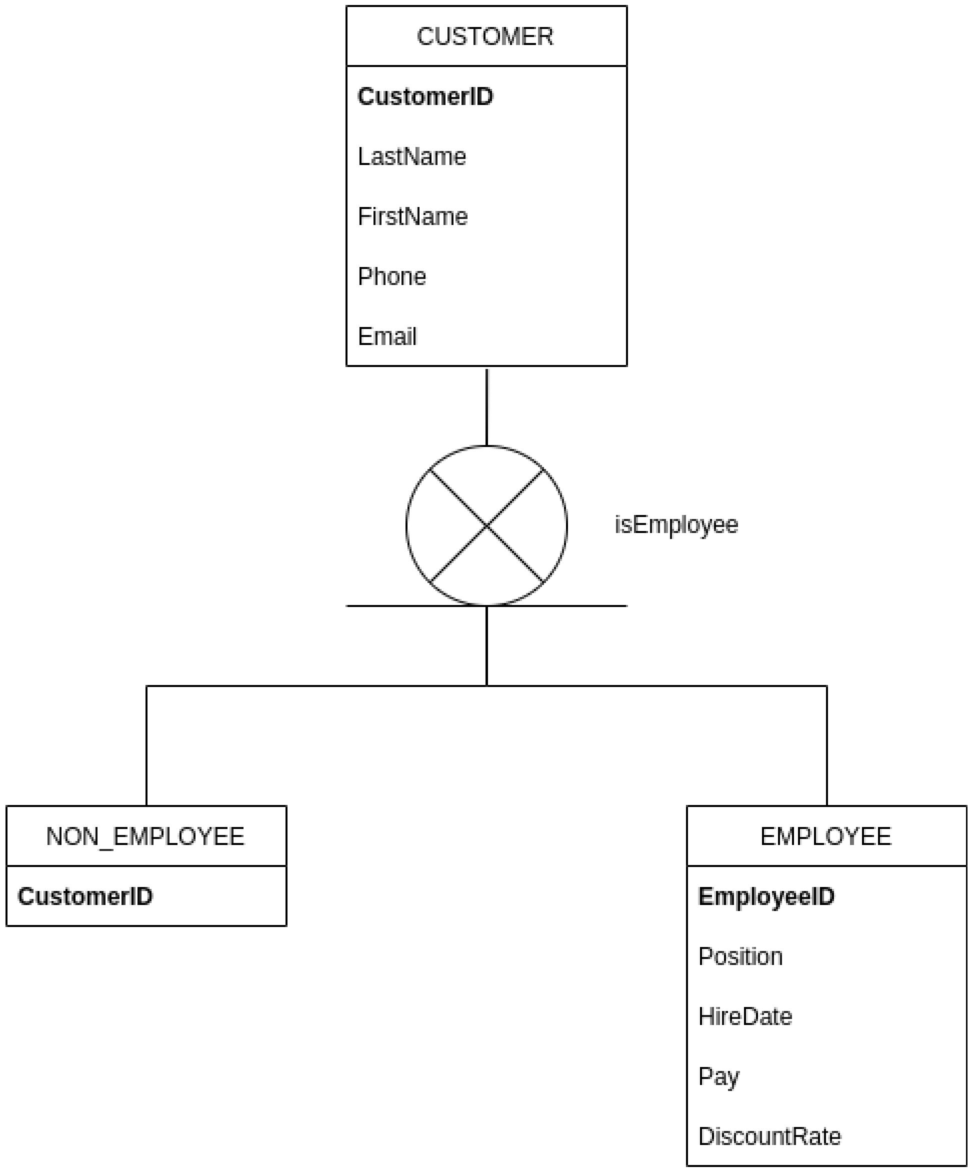
Relevant Removals: PURCHASES table – unnecessary with the existence of order attributes in INVENTORY table.

Relevant Additions: INVENTORY table – used to keep track of the number of items and their status per the shop requests.

Modifications:

- Changing the SALE_ITEM table primary key to a composite key with SaleID and ItemID. Since numerous items may now be purchased, it is important to maintain uniqueness by joining it with an item identifier.
- Maximum cardinality of SALE and SALE_ITEM: multiple items may now be purchased simultaneously in a sale.
- Cardinality of SALE_ITEM and PURCHASE: if an item is to be sold, it must exist in the INVENTORY table, and it is possible there can be multiple. An item in the INVENTORY table may not be a sold item, but there may also be many items sold.

/* *** E2 – 5C *** */

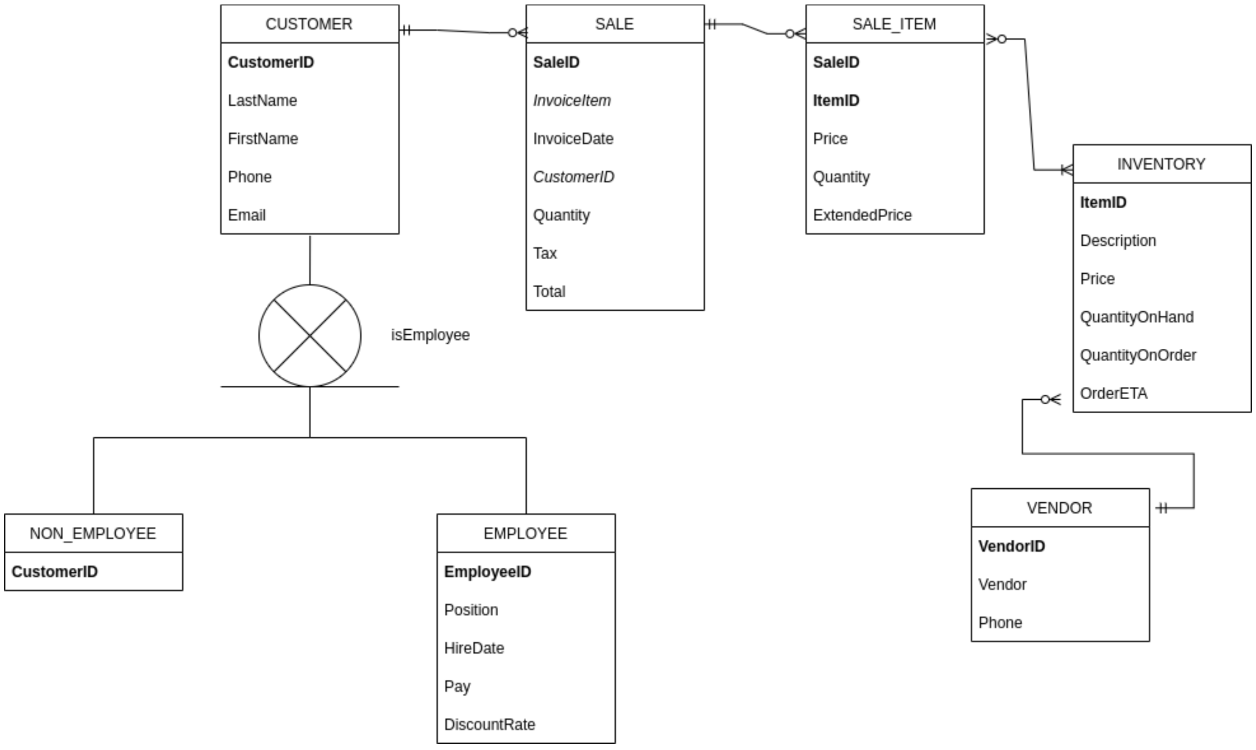

CUSTOMER
CustomerID
LastName
FirstName
Phone
Email
isEmployee
NON_EMPLOYEE
CustomerID
EMPLOYEE
EmployeeID
Position
HireDate
Pay
DiscountRate

Discriminator: isEmployee

Exclusive – A person is either an employee or they are not. It is not possible to be inclusive.

Customer is the supertype – both employees and non-employees would absorb its attributes.

/* *** E2 – 5D *** */



No additional modifications were required to combine Part B and Part C. The cardinality between CUSTOMER and SALE are unmodified, since each sale needs to be tied to exactly one customer, and a customer may make zero or many purchases.

/* *** E2 – 5E *** */

- Create a set of validation data, and insert it into the database.
- Modify, add, and remove validation data from the database.
- Enforce proper rules, such as data types, numeric ranges, null values, uniqueness, etc.
- Create scripts and queries to check the accuracy of the model.
- Utilize the database server software to check for common errors or misspellings.

Answer Questions A through E found on page 321 in Chapter 6 for the Queen Anne Curiosity Shop.

/* *** E2 – 6A *** */

CUSTOMER

Column Name	Type	Key	NULL Status	Remarks
CustomerID	Int	Primary	Not Null	Surrogate
LastName	Char(25)	No	Not Null	
FirstName	Char(25)	No	Not Null	
Phone	E.164 FORMAT STRING	No	Null	
Email	Char(100)	No	Null	
IsEmployee	Bool	No	Not Null	Exclusive Discriminator

EMPLOYEE

Column Name	Type	Key	NULL Status	Remarks
EmployeeID	Int	Primary	Not Null	Surrogate
Position	Char(25)	No	Not Null	
HireDate	Date	No	Not Null	
Pay	Int	No	Not Null	
DiscountRate	Char(100)	No	Null	

NON_EMPLOYEE

Column Name	Type	Key	NULL Status	Remarks
CustomerID	Int	Primary	Not Null	Surrogate

SALE

Column Name	Type	Key	NULL Status	Remarks
SaleID	Int	Primary	Not Null	Surrogate
InvoiceItem	Char(25)	Foreign	Not Null	
InvoiceDate	Date	No	Not Null	
CustomerID	Int	Foreign	Null	
Quantity	Int	No	Null	
Tax	Numeric	No	Null	
Total	Numeric	No	Not Null	

SALE_ITEM

Column Name	Type	Key	NULL Status	Remarks
SaleID	Int	Primary, Foreign	Not Null	
ItemID	Char(25)	Primary, Foreign	Not Null	
Price	Numeric	No	Null	
Quantity	Int	No	Null	
ExtendedPrice	Numeric	No	Null	

INVENTORY

Column Name	Type	Key	NULL Status	Remarks
ItemID	Int	Primary	Not Null	Surrogate
Description	Char(100)	No	Not Null	UNIQUE AK.1
Price	Numeric	No	Null	
QuantityOnHand	Int	No	Null	
QuantityOnOrder	Int	No	Null	
OrderETA	Date	No	Null	
VendorID	Int	Foreign	Not Null	

VENDOR

Column Name	Type	Key	NULL Status	Remarks
VendorID	Int	Primary	Not Null	Surrogate
Vendor	Char(25)	No	Null	
Phone	E.164 FORMAT STRING	No	Null	

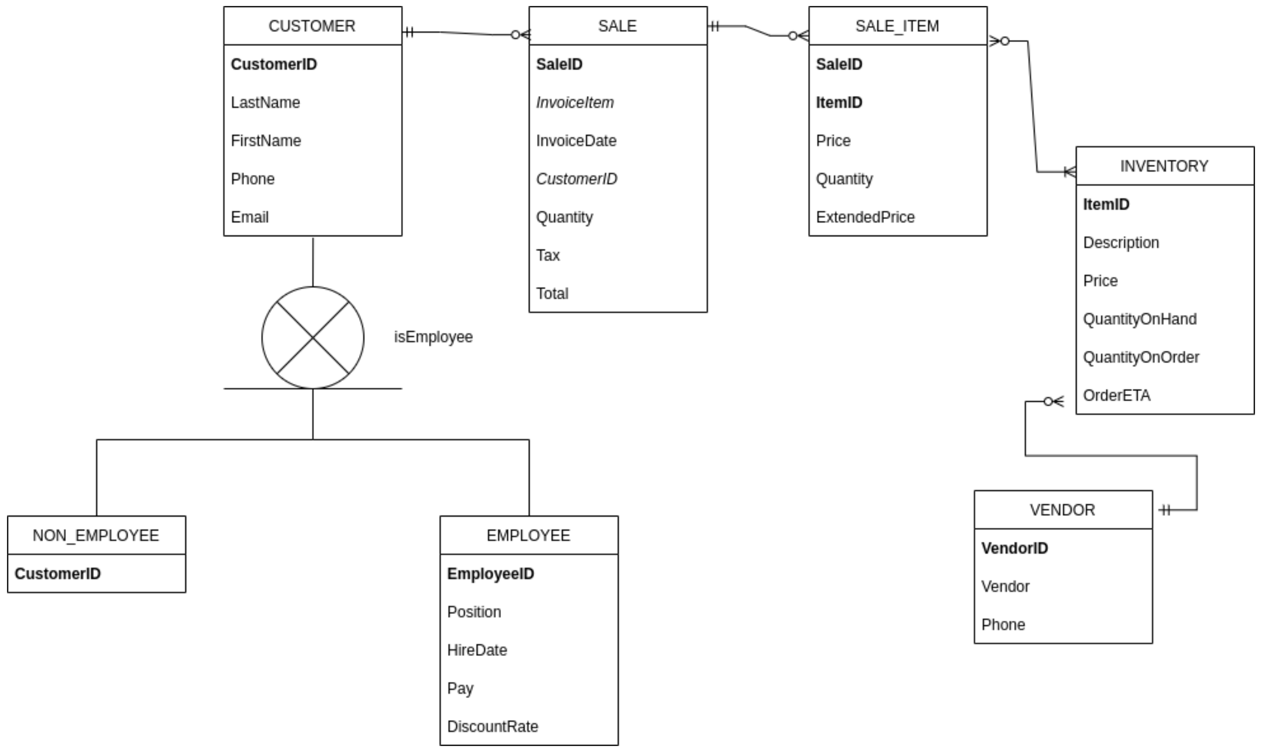
/* *** E2 – 6B *** */

SALE_ITEM is a weak entity since it is ID-Dependent. It is represented by having both of its primary keys (SaleID and ItemID) exist also as foreign keys.

/* *** E2 – 6C *** */

EMPLOYEE and NON_EMPLOYEE are subtypes of the CUSTOMER supertype. The CUSTOMER table has an attribute titled “isEmployee” which exists as an exclusive discriminator. EMPLOYEE and NON_EMPLOYEE absorb the attributes of CUSTOMER.

/* *** E2 – 6D *** */



/* *** E2 – 6E *** */

Relationship			**Cardinality**	
Parent	**Child**	**Type**	**Min**	**Max**
Customer	Sale	Nonidentifying	Mandatory-to-Optional	One-to-Many
Sale	Sale_Item	Identifying	Mandatory-to-Optional	One-to-Many
Inventory	Sale_Item	Identifying	Mandatory-to-Optional	One-to-Many
Vendor	Inventory	Nonidentifying	Mandatory-to-Optional	One-to-Many

Customer (Parent) is Required	**Action on Customer (Parent)**	**Action on Sale (Child)**
Insert	None.	Get a parent.
Modify key or foreign key	Prohibit. Surrogate key is used as primary key.	Prohibit. Surrogate key is used as primary key.
Delete	Prohibit if SaleID exists with specified Customer. Allow if no SaleIDs exist with specified Customer.	None.

Sale (Parent) is Required	**Action on Sale (Parent)**	**Action on Sale_Item (Child)**
Insert	None.	Get a parent.
Modify key or foreign key	Prohibit. Surrogate key is used as primary key.	Prohibit. Surrogate key is used as primary key.
Delete	Cascade delete.	None.

Inventory (Parent) is Required	**Action on Inventory (Parent)**	**Action on Sale_Item (Child)**
Insert	None.	Get a parent.
Modify key or foreign key	Prohibit. Surrogate key is used as primary key.	Prohibit. Surrogate key is used as primary key.
Delete	Prohibit if SaleIDs exist for specified Inventory. Allow if no SaleIDs exist for specified Inventory.	None.

Vendor(Parent) is Required	**Action on Vendor (Parent)**	**Action on Inventory (Child)**
Insert	None.	Get a parent.
Modify key or foreign key	Prohibit. Surrogate key is used as primary key.	Prohibit. Surrogate key is used a primary key.
Delete	Cascade delete.	None.