**Lab Questions (9 @5 points each for a total of 45)**

Your answers for each question should fill at least 1/2 page. You may use any drawing application to create the designs and then copy/paste the pictures into this document. If you use Visio, just use the flowchart shapes; you may be able to use the Word flowchart auto shapes; you might need to scan your drawing to a .jpg or .bmp or whatever and then paste it here.

**---------------------------------------------------------------------**

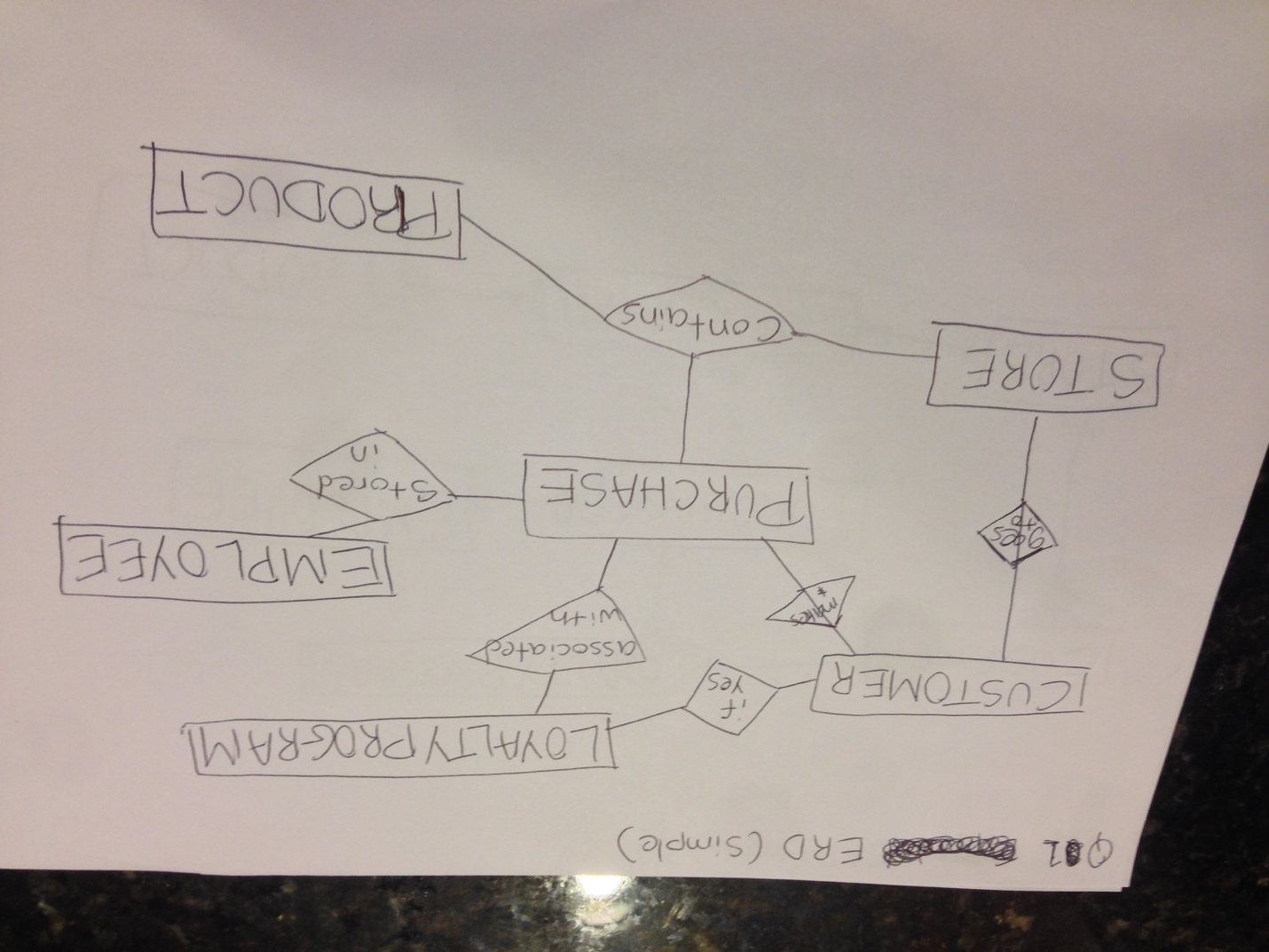
**A grocery chain needs a database designed in order to keep track of inventory and sales; the scenario to model is as follows:**

**A customer goes to one of the stores to make a purchase, which contains one or more products. The employee ID of the cashier will be stored for the purchase; if the customer is a member of the loyalty program, his membership ID will be associated with the purchase as well (although this is not required).**

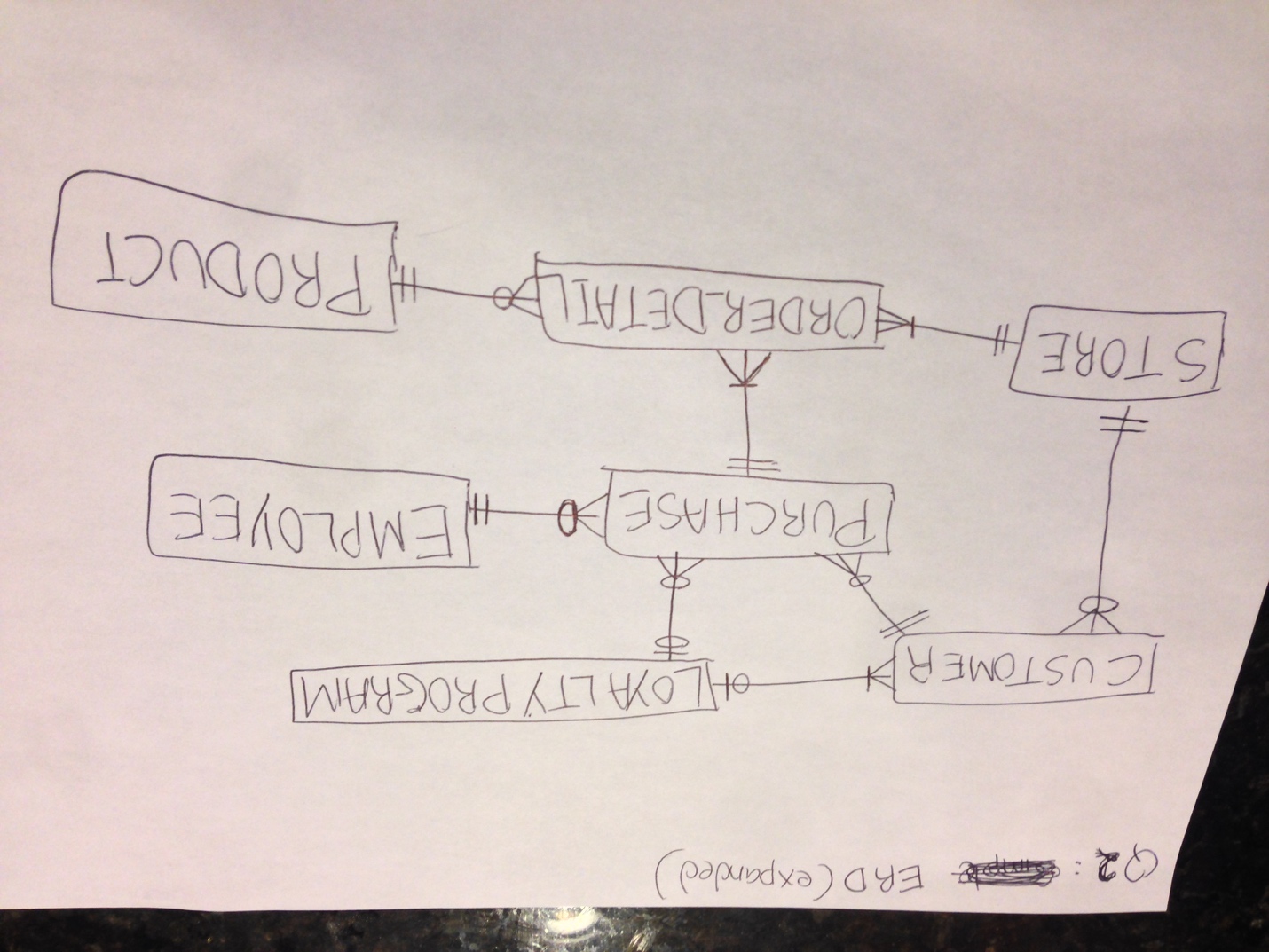
**Each product will have a current price and stock inventory count for that store; the purchase amount will be the total of all product prices times their quantity. After the customer pays, the purchase should be marked as paid, and the stock inventory count for all items in the purchase should be reduced accordingly.**

**---------------------------------------------------------------------**

**Q1**. Read through the above scenario and determine which entities are involved (identify all of the *things* and decide whether they are entity classes or attributes). Create a simpleERD to show only the **entities** and the **relationships** between them (see the first lecture example). Name the relationships (e.g. “employs”, “schedules”, etc) but do not show the relationship cardinalities nor any of the attributes. It is expected and perfectly OK to have at least one **many-to-many** and/or **ternary** relationship here, because these will be changed in the next step.



**Q2.** Expand the ERD from the previous question by adding the minimum and maximum relationship cardinalities to the model; eliminate any ternary and/or many-to-many relationships from the previous ERD by creating an intersection entity between them. Show optional/mandatory cardinalities and use crows-feet notation to display the “many” sides. Include cardinality for **all** relationships between two entities.



**Q3.** Taking each pair of related entities at a time, write one sentence describing the relationship cardinalities (minimum and maximum) in each direction. For example, this relationship



...may be stated:

***An employee builds zero to many stoves / A stove is built by exactly one employee.***

**A customer make zero to many purchases order / A purchase order can have exactly one customer.**

**A customer can or cannot be Loyalty Program member / A Loyalty Program can have zero to many customer.**

**A store can have zero to many customers / A customer can be at exactly one store.**

**A purchase can include zero to one loyalty program member / The loyalty program members can make zero to many purchases.**

**Exactly one employee can be stored in each purchase / Employee will ring one to many purchases unless that employee is helping with shelving or manual labor.**

**A purchase can include one to many ordered items (order\_detail) of one to many quantities / Each ordered item can relate to exactly one purchase order.**

**One product can be included zero to many times in an ordered item / An order item can have exactly one product.**

**A store will have one to many quantities of ordered items / An order item is purchased at exactly one store.**

**Q4.** Turn four of your entities into relations. Select an attribute(s) to represent the primary key; display this first, underlined. Include other attributes you would expect to find for this entity. Lastly, include any foreign keys which reference other entities; display these in italics.

Example from the FiredUp database:

**STOVE** (SerialNumber, Type, Version, DateOfManufacture, Color, *FK\_EmpID*)

**CUSTOMER** (CustomerID, FirstName, LastName, *FK\_LPMemberID, FK\_StoreID*)

**STORE** (StoreID, Address, PhoneNbr)

**PURCHASE** (PurchaseID, *FK\_CustomerID*, *FK\_EmployeeID, FK\_LPMemberID, FK\_OrderDetailID,* TotalPrice)

**PRODUCT** (ProductID, ProductName, Price, InventoryCount)

**ORDER\_DETAIL** (OrderDetailID, *FK\_StoreID, FK\_PurchaseID, FK\_ProductID,* Quantity)

**EMPLOYEE** (EmployeeID, FirstName, LastName)

**LOYALTY\_PROGRAM** (LPMemberID, PhoneNbr, Points)

**Q5.** What assumptions did you make when you were creating the ERD? Turn these assumptions into questions that you would ask the client in order to continue the design process. Include at least **five** questions. Be sure to address any ambiguities in the scenario that might affect your design.

Example from the FiredUp database: Can you have an INVOICE that doesn't include an EMPLOYEE (for example, an online sale with no sales associate)? This would affect the minimum cardinality of the relationship between INVOICE and EMPLOYEE.

Another example: Can two or more sales associates both be listed on a single INVOICE? This would create an M:N relationship between INVOICE and EMPLOYEE that would require an intersection entity to resolve.

**Question 1 : Does the can loyalty program membership extend to the household? I had to make any assumption that the member can provide either member ID or phone number to relate it to the purchase order.**

**Question 2: Will all stores use the same database? I will assume they will use the same database and that is why we need the store table to identify each store.**

**Question 3: Will the store id information be the receipt? I have seen many chain stores, like CVS, include the store id for customer wants to return item. It is connected to the Order Detail table that can be queried in a receipt.**

**Question 4: Can you have a customer make multiple purchase? I assume and I know I have it past because I forgot something.**

**Question 5: Can any employee work it multiple stores? I assume so. The purchase order is connected to the store so I didn’t connect employee to the store as employee is already connected to the purchase order.**

**Question 6: Can employee not make any purchases? I assume that they can be employee but does only shelving.**

**Q6.** A weak entity is an entity whose existence depends upon another entity. Examine the ERD for the FiredUp database and identify two entities that can be considered to be weak (note: this ERD does *not* use rounded-corner representation of weak entities; you will have to identify them logically). Explain why each of these entities is weak.

**After examining the ERD FiredUp database, I identified EMAIL and PHONE tables are child of the parent table: CUSTOMER. The minimum cardinality from the EMAIL table to CUSTOMER table is always one and same goes for PHONE to CUSTOMER table.**

**Q7.** Again, examine the FiredUp ERD. Identify two intersection entities (i.e. entities which were created to break apart a **N:M** relationship). Explain what tells you that these are intersection entities.

**After examining the ERD FiredUp database, I identified REPAIR\_LINE\_ITEM and INV\_LINE\_ITEM entities were created to break apart many to many relationship. The entity, REPAIR\_LINE\_ITEM was created because STOVE\_REPAIR many need one to many PART and each type of PART can be used for one to many STOVE\_REPAIR. As for the entity, INV\_LINE\_ITEM, it was created because INVOICE can include one to many PART for the STOVE and PART can be sold in one to many quantity.**

****

**Q8.** Examine the data in the relations in Figure 3-17, above. Explain why we can say that Buyer determines Department, but Department does not determine Buyer.

**Buyer determines Department and not the other way around because each Department have one or more Buyers. According the data above, each buyer belongs to only one Department.**

**Q9.** Assume the types of the columns in the SKU\_DATA\_4 table in Q7 are:

|  |  |  |
| --- | --- | --- |
| Colum Name | Data Type | Size |
| SKU | NUMERIC(6, 0) | 5 Bytes per row |
| SKU\_Description | CHAR(128) | 128 Bytes per row |
| Buyer | CHAR(64) | 64 Bytes per row |

Assume the store has 500,000 products. Ignoring any additional overhead, how large would the SKU\_DATA\_4 table be in Bytes? Show how you calculated this number.

**500,000 products are 500,000 instances or rows. Each row has 3 columns with 197 Bytes per row.**

**Total\_bytes\_SKU\_DATA\_4 = 500,000 \* (5 + 128 + 64) = 98,500,000 Bytes**