**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.

WILLIAM CHIEN 10/25/2017

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

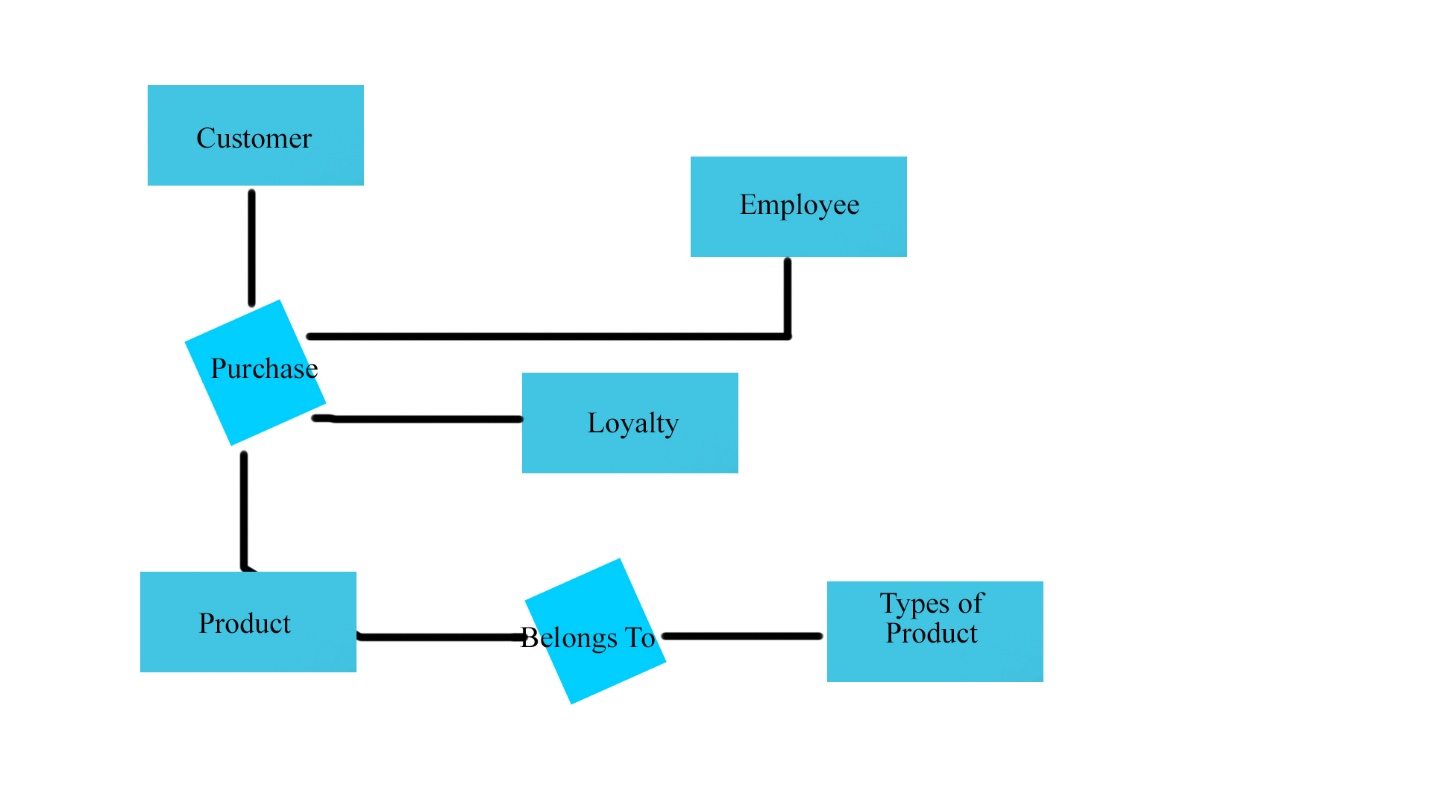
**A grocery chain needs a database designed in order to keep track of inventory and sales; the scenario to model is as follows:**

**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).**

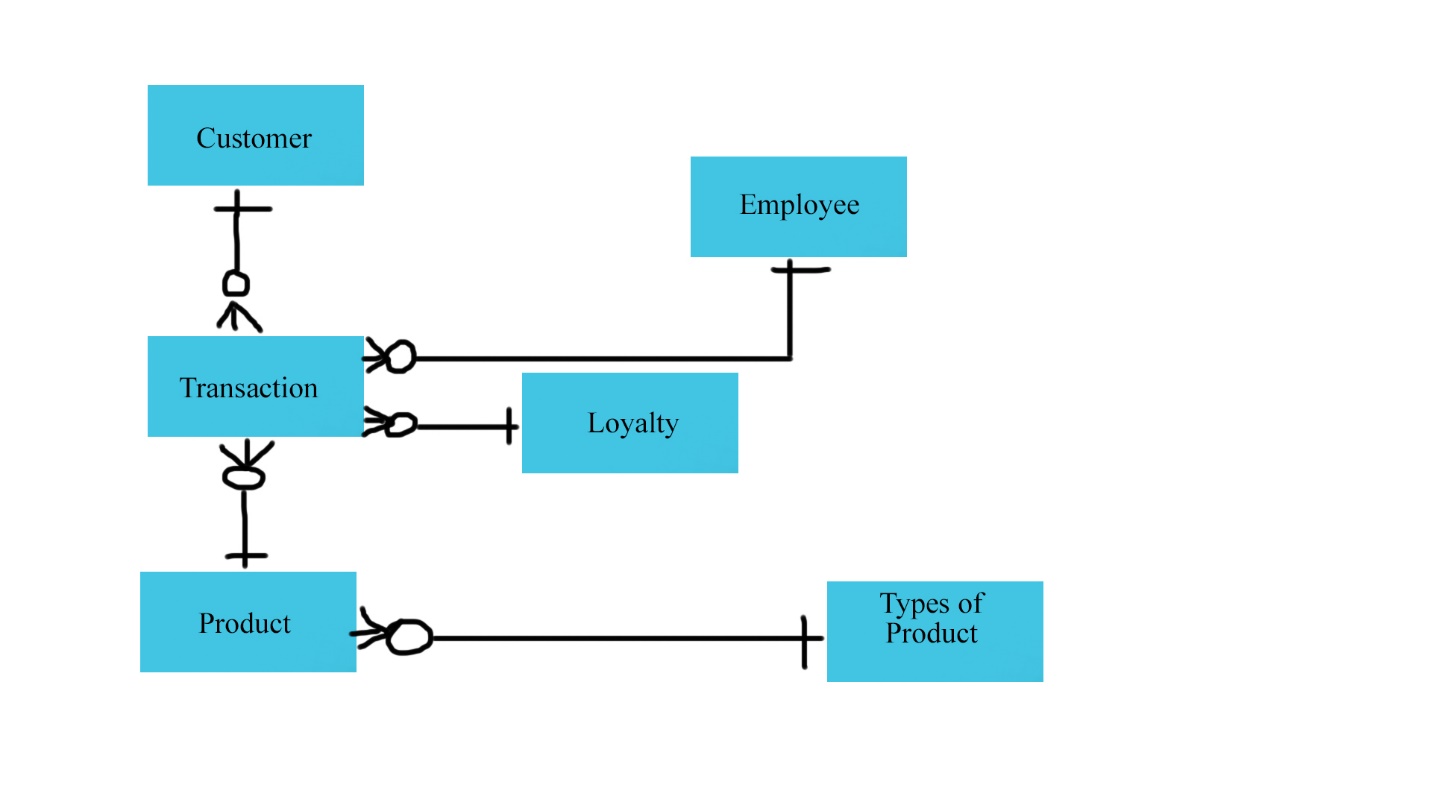
**Each product will have a current price and stock inventory count for that store; the purchase amount will be the total of all product prices times their quantity. After the customer pays, the purchase should be marked as paid, and the stock inventory count for all items in the purchase should be reduced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

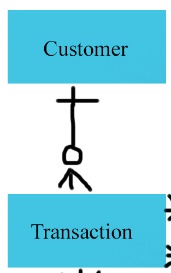


**Q3.** Taking each pair of related entities at a time, write one sentence describing the relationship cardinalities (minimum and maximum) in each direction. For example, this relationship

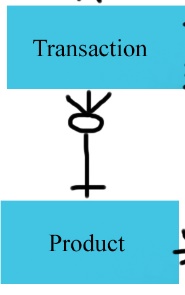


...may be stated:

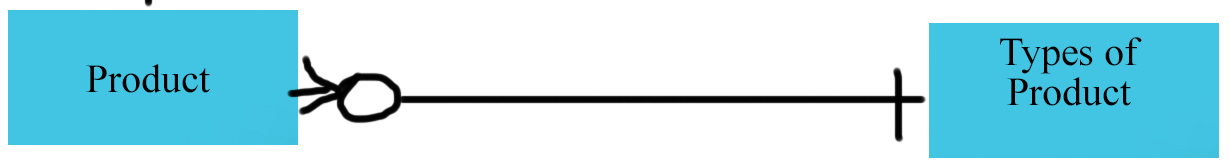
***An employee builds zero to many stoves / A stove is built by exactly one employee.***



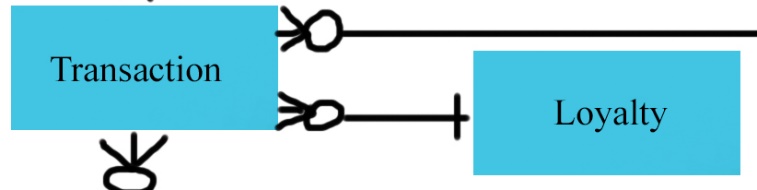
A customer makes zero to many transactions/ A transaction is made by exactly one customer.



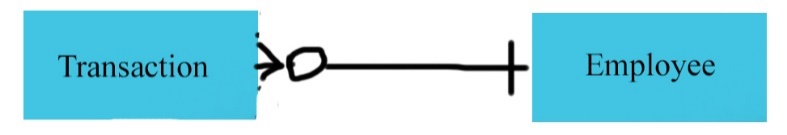
A product can be part of zero to many transactions/A transaction can have exactly one product.



A product can have exactly one type/ Types of product can have many product.



A transaction can have exactly one loyalty number/ A loyalty number can have zero to many transactions.



An employee can have zero to many transactions/ a transaction can have exactly one employee.

**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** (ID number, name, address, loyalty number)

**Employee** (ID number, *name*, position, *department*, hire date)

**Product** (Serial number, *name*, cost, *department*)

**Purchase** (Transaction ID, date, *employee ID*, *customer ID*, order total, department, cash register number, store number)

**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.

* Can a customer have more than one loyalty number? This would make a M:N relationship between Loyalty and Customer.
* Can multiple customers be put on one transaction for some reason? That would make a M:N relationship between customer and transaction.
* Can a customer order stuff online? If so, there wouldn’t be a requirement for an employee’s ID number to be on the online order.
* Can there be more than one employee listed on one transaction? That would create an M:N relationship between transaction and employee.
* Can a loyalty number have multiple customers in it? That would create an M:N relationship between customer and loyalty.

**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.

Phone: without the customer id, there would be no way to tie the phone number to the specific person.

Email: without customer id, there would be no way to tie email to the specific person.

**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.

INV\_LINE\_ITEM is an intersection entity. The entities connected to it are in 1 to many relationships.

REPAIR\_LINE\_ITEM is an intersection entity. The entities connected to it are in 1 to many relationships.

****

**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.

There are 2 different buyers for the same 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.

DATALENGTH(SKU\_DATA\_4)