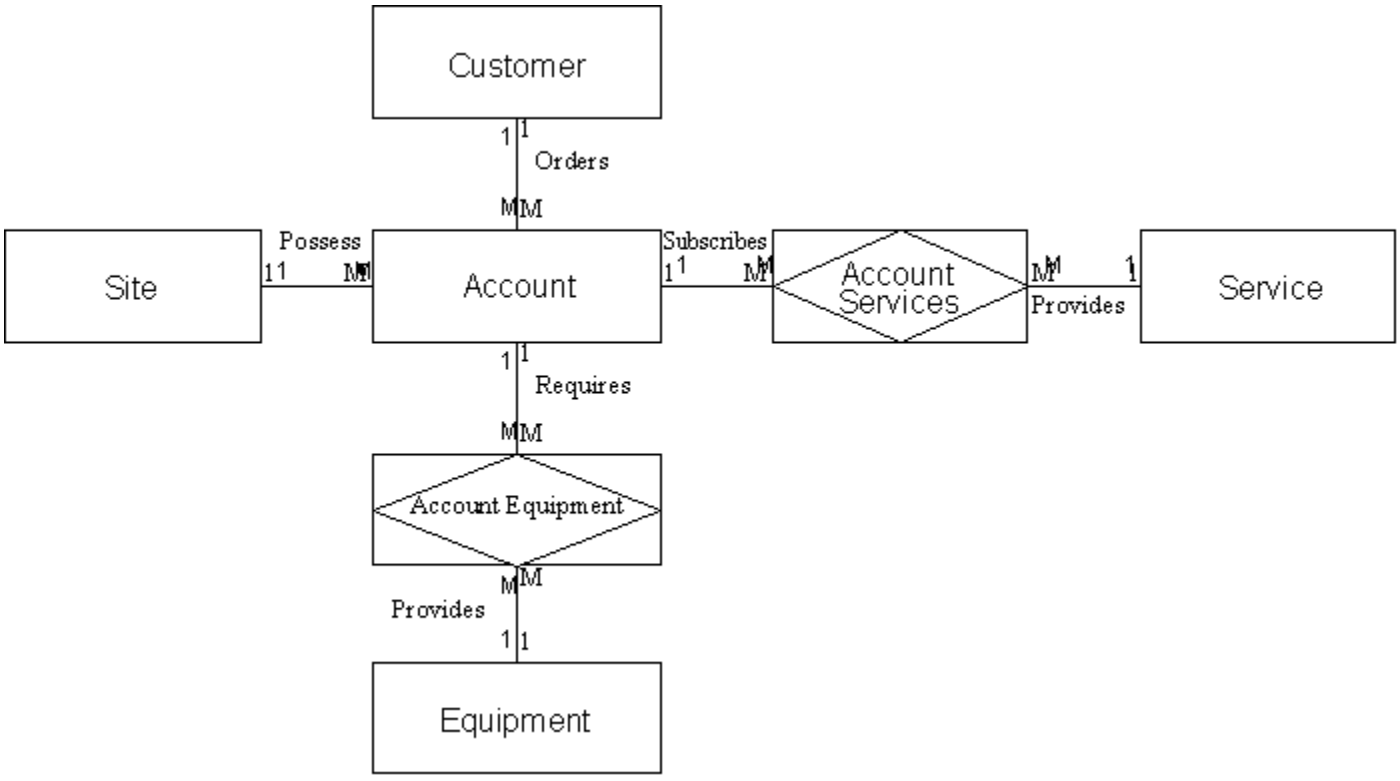
Assignment 4 Discussion

The following is a review of **two good ER Diagrams, two OK examples**, and **eight examples of flawed diagrams** that have been received from students. Comments about how the assignment was graded is at the end of the discussion.

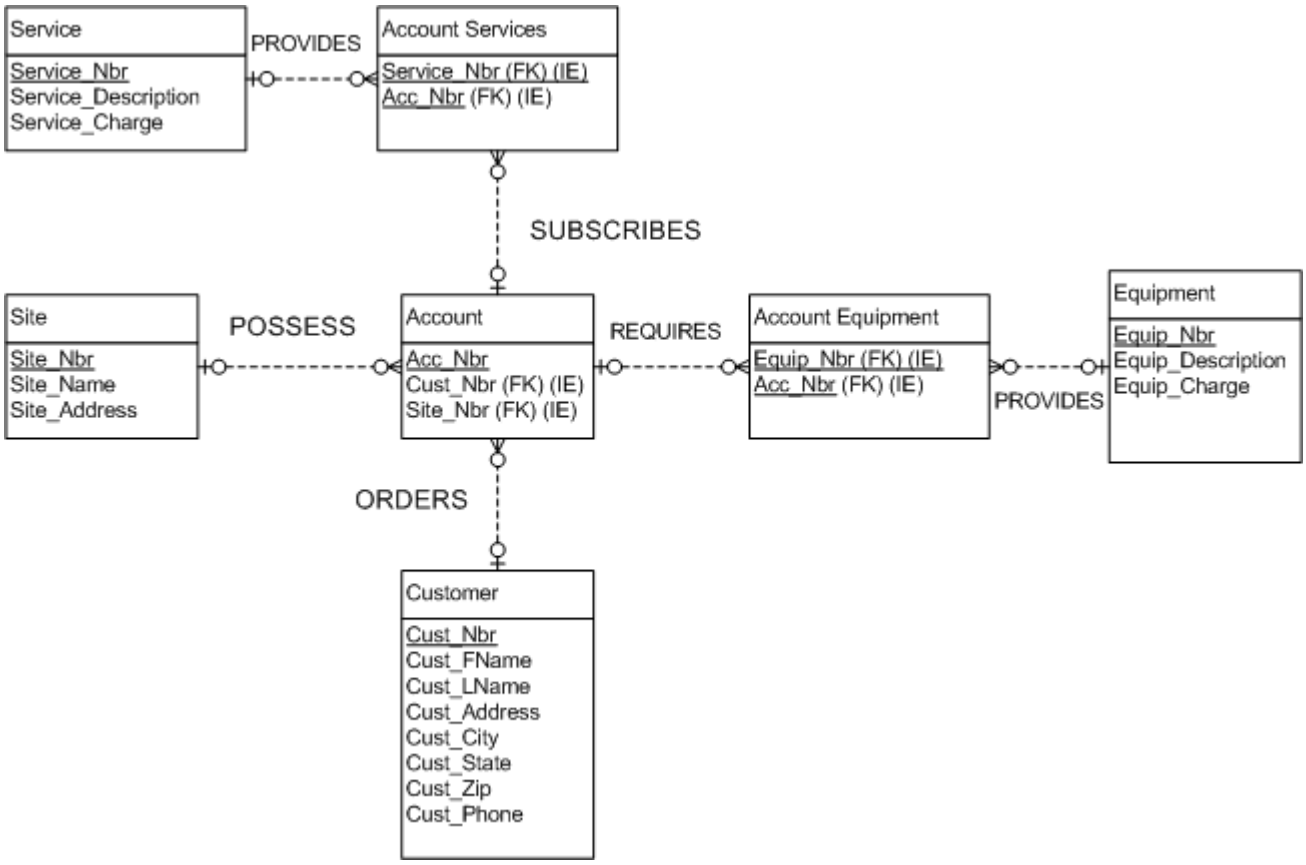
Consider the importance of understanding the business requirements/rules when evaluating each of the examples.

Good, One

There were two good possible solutions. Both required the recognition that the cardinality between Account and Service and Account & Equipment were many to many. The difference between them is the way in which equipment was handled. This first example uses composite entities to remove the Many to Many cardinalities. A composite is an entity that acts like a bridge between entities to eliminate a Many to Many. The composite entity uses the Primary Key from both of the parent entities as the Primary Key in the Composite entity.

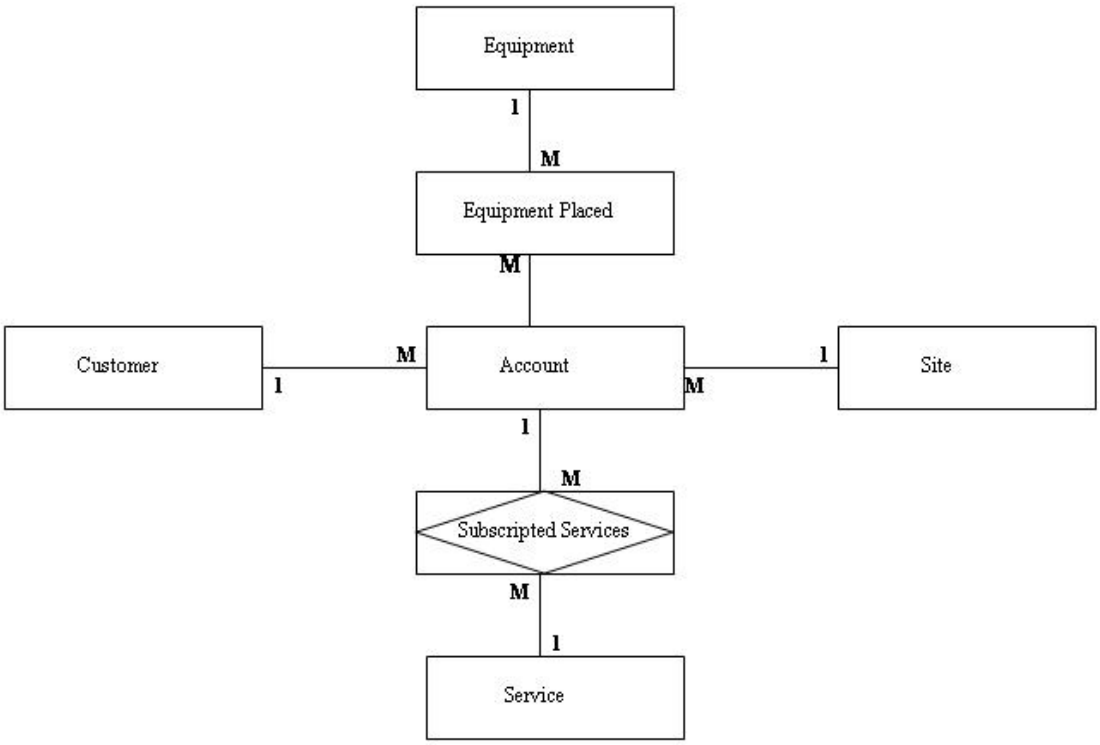


This is the same diagram but, uses the crow's foot diagramming method.



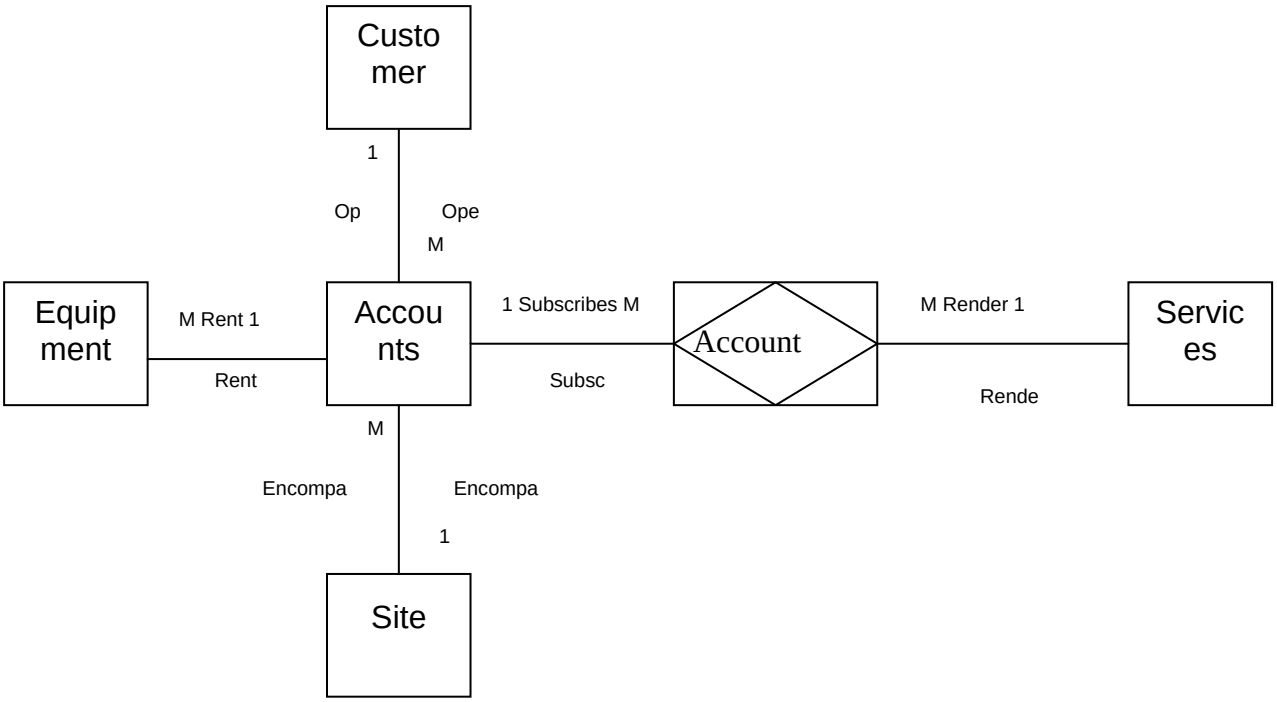
Good, Two

The second good example does not use a composite entity in the Account - Equipment association. The Equipment Placed entity is similar to a composite, but it does not use both Primary Keys as its Primary Key. What it uses is its own unique identifier as its Primary Key. In this case, the serial number could be used to uniquely identify each of the pieces of equipment that are being placed at a customer site. The serial number could then be used as the Primary Key and the entity is no longer a composite, but a normal entity.



OK, one

Equipment?



The above diagram indicates that Equipment has a one to many relationship with Account. This is an acceptable solution if an unique identifier (serial number) would be used for each piece of equipment however, it will create a lot of redundant data in the database.

The purpose of the Equipment entity would be to contain common data about the same type of equipment, such as: model, make, descriptions, manufacturer, cost, charges, etc.. This data allows for the correct charges to be applied to the account for the equipment used. For a type of equipment, such as a decoder box, there would only need to be a single entity listing for that type of equipment.

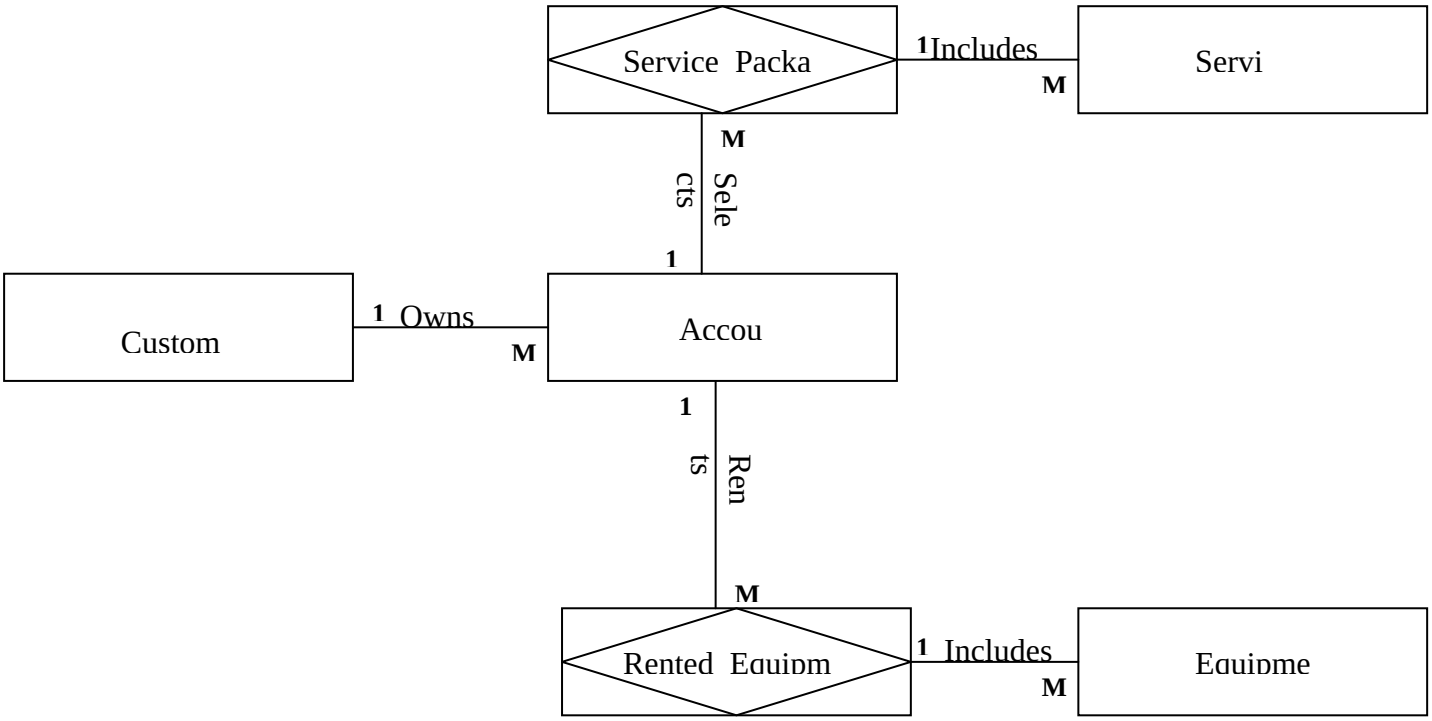
Although each piece of equipment can be uniquely identified with a serial number, using the unique identification as the reason for having an Equipment entity would create all of the common data to be repeated for each individual piece. Imagine if there were 50,000 decoder boxes being used. If each entity represented the individual decoder box, then there would be 50,0000 entities and growing for each decoder box being used. This would mean that the basic information (model, make, descriptions, manufacture, cost, charges) would be duplicated for each entity.

The solution requires having another entity that would hold the common data for each type (model, description, etc.). Then anothrer entity that would use the serial number as the unique identifier. This would be modeled as the Good, Two example shows. Another way would be to

would be to have a single Equipment entity and then place the serial number for the actual piece of equipment being used as an attribute of the composite entity that would be created for the association between Account and Equipment. By doing this the individual pieces of equipment are assigned to the account that is using it and there is not a lot of redundant data being stored unnecessarily. This is modeled in the Good, One example.

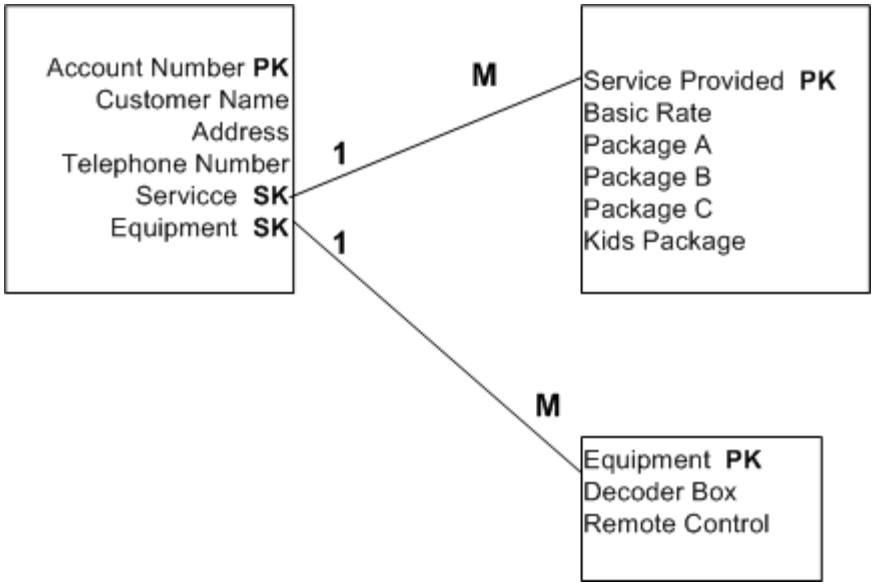
OK, two

Site?



This diagram does not have a Site entity. Where would the site be? It is possible that the site could be located as an attribute in the Account entity. This would then allow for each Account to have its own site. Yet, the business rules state that a site can have more than one Account. Including the site as an attribute of the Account entity would be sufficient, but it would not keep track of the equipment that is located per each site and increase data redundancy.

Flawed Example 1:



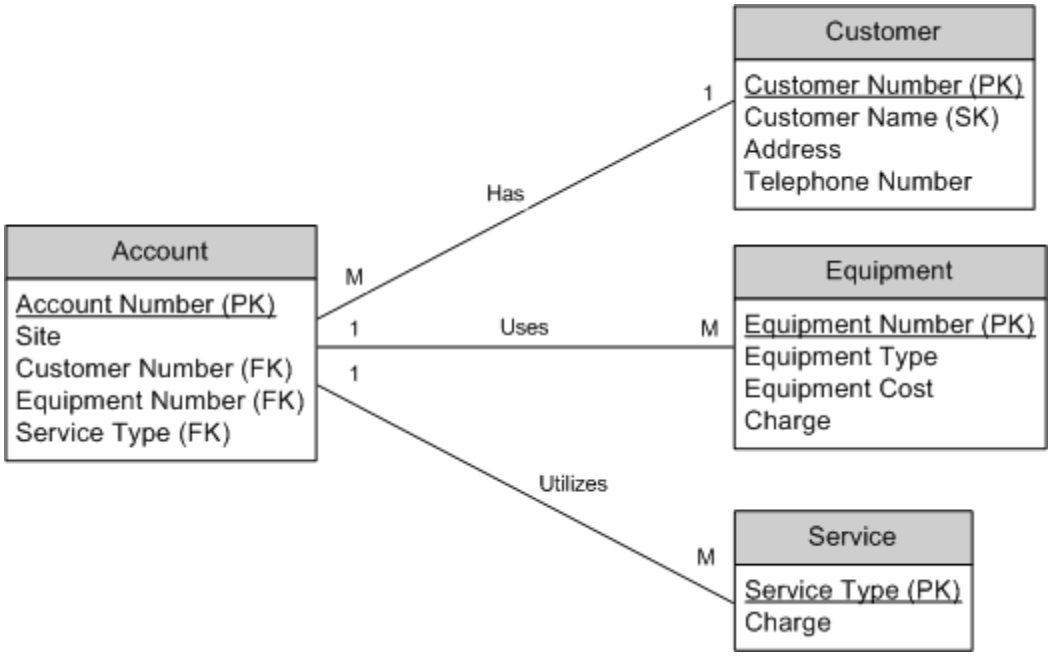
Consider what is in error with this ER Diagram:
There will be **repeating data** in the Account Entity (I think that is what is called, although there is no name for the entity) as a customer can have many accounts. Is this acceptable, well it could be if you really thought that there would not be that many times a customer will set up multiple accounts.

The relationship between both Service and Equipment are **many to many** as an account can have many different services and equipment. This would cause for the current entity to repeat itself to allow for multiple services to be associated with an account, or **repeating groups**. How does the diagram account for that? Would Account have to have many different attributes to hold the number of services that would be ordered? How do you know the exact amount?

In SERVICE PROVIDED, the types of services (Basic Rate, Package A, etc.) are each indicated as an attributes. This would limit the number of services offered to only the ones indicated. Every time a new service would be offered the structure of the entity would have to be changed. The types of services are actually values that an attribute within the entity would hold (Service Description, for an example).

Also notice that there is **no association** provided for the relationships. Why are they related?

Flawed Example 2:

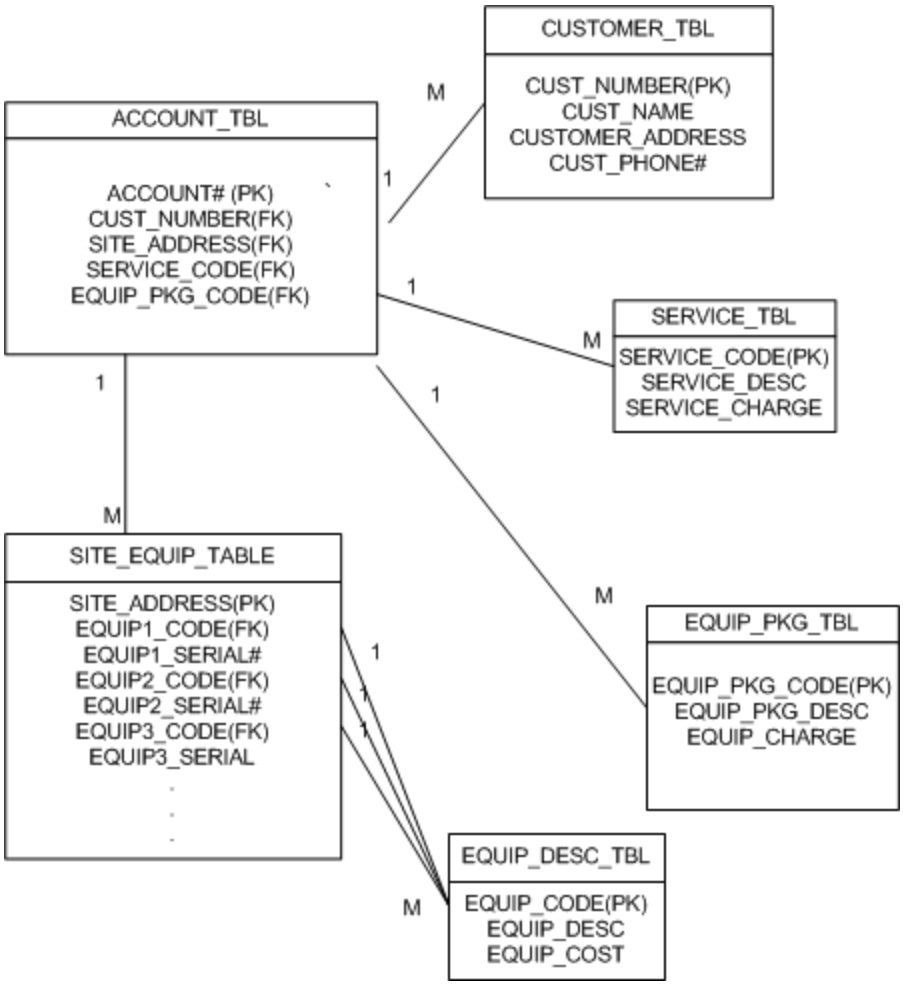


This one is better than Example 1, as it has created a Customer Entity, but there still remains a One to Many relationship between Account and Service. The requirements state that each account can have more than one service, but the SERVICE entity will have to be able to have a relationship with more than one ACCOUNT, otherwise each service could only be ordered by one account.

For the One to Many declared, is the foreign key in the correct location? If there were a one to many relationship between service and account, then the SERVICE entity should have Customer Number as an Attribute. This attribute would then be the foreign key.

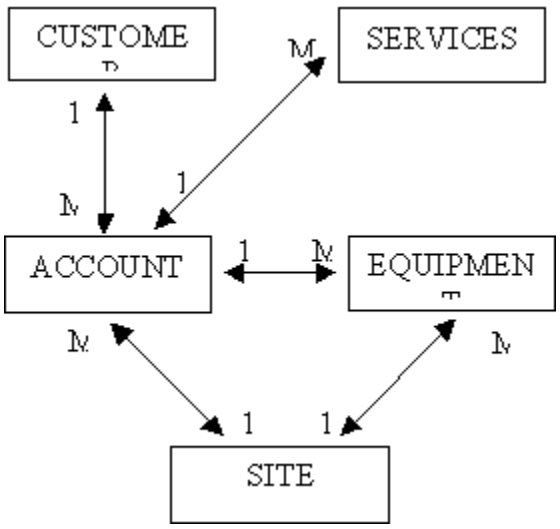
Notice that there is no SITE entity. Is there a need for a site entity? The requirements say that a site may have more than one account, but if you interpret the requirements it does not necessarily mean that a site is separate from where the customer, that is billed for the service, is located. An apartment building (site) could have many different accounts. The cable company would provide a single cable into the building and would then feed to the different accounts in the building. Would you need to have a site entity? As this diagram displays you don't, but the requirements are indicating a relationship between site and entity.

Flawed Example 3:



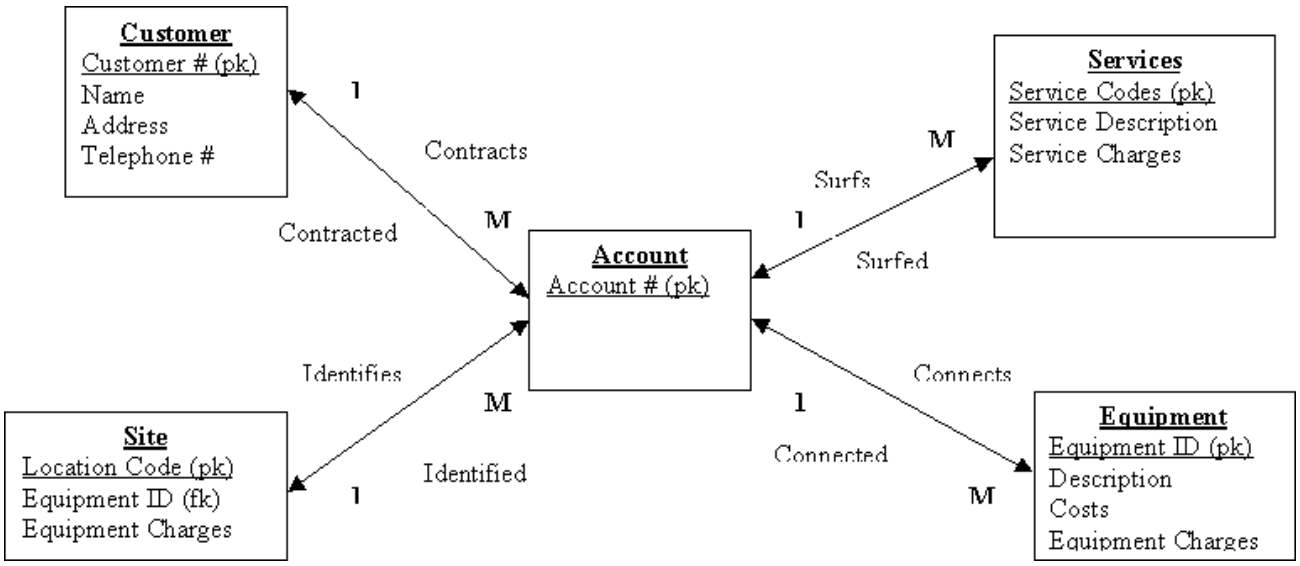
This has the same problems as Example 2, but has added a SITE entity. What is the relationship between Account and Site? The requirements say that a site can have many different accounts, but the diagram shown says that a site can have many sites. What is the association between the entities? What are the two Equipment entities doing? How can the Equipment Package be related to the Account, but the Equipment being used be related to the site? The requirements say that Equipment is linked to the account and not the site.

Flawed Example 4:



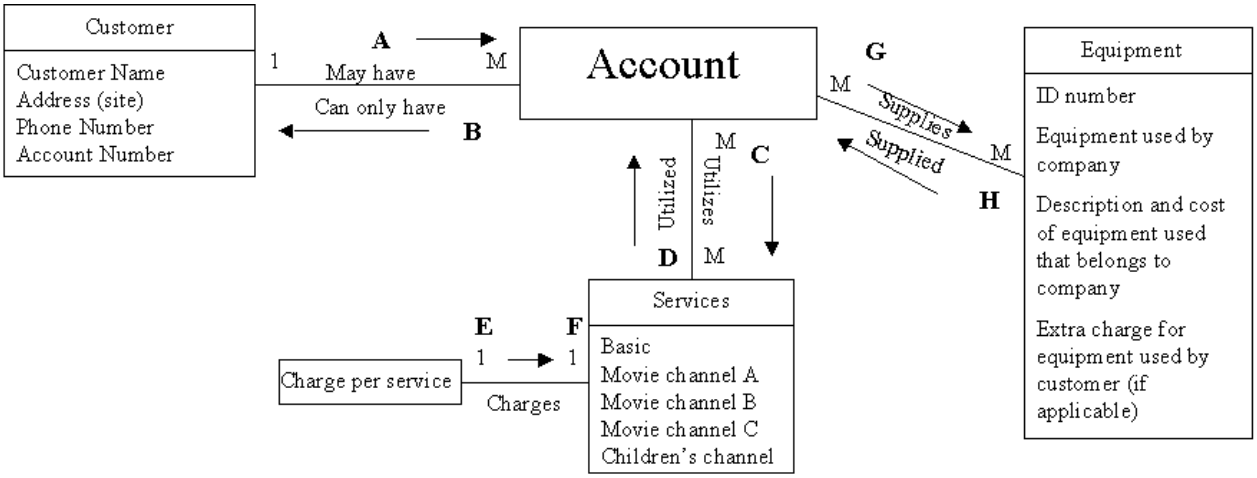
Similar to Example 3, there are no associations described and EQUIPMENT is being related to both SITE and ACCOUNT. What are the associations?

Flawed Example 5:



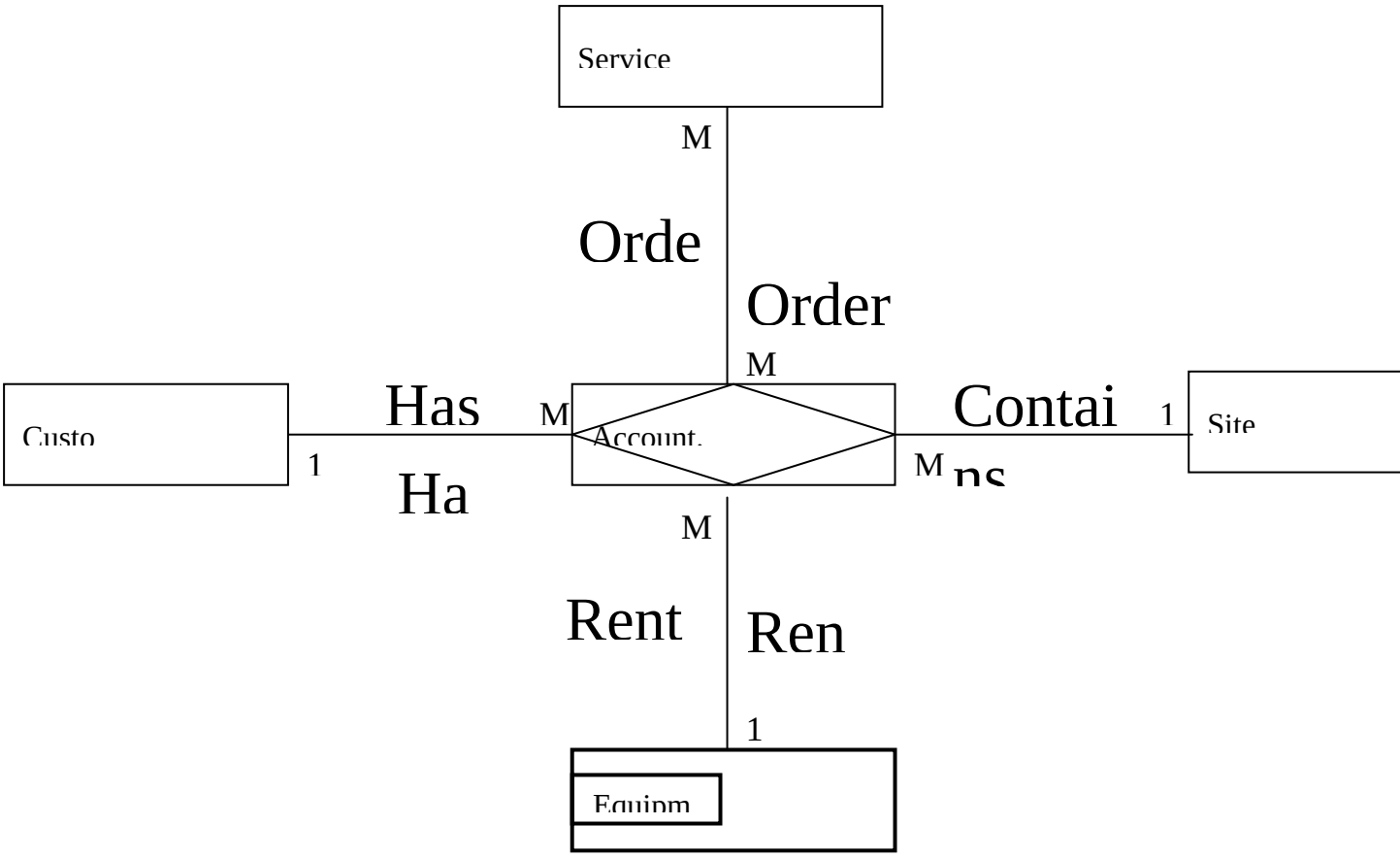
This is better than example 2, but still One to Many relationships, and would foreign keys need to be located anywhere? If you were to relate a Customer entity to an Account entity you would need to have the Account Number in the CUSTOMER entity. But would you need a foreign key? If you wanted to guarantee that you would have the correct account numbers in the CUSTOMER entity, and not allow for any erroneous data, you would have a foreign key to protect **referential integrity**.

Flawed Example 6:



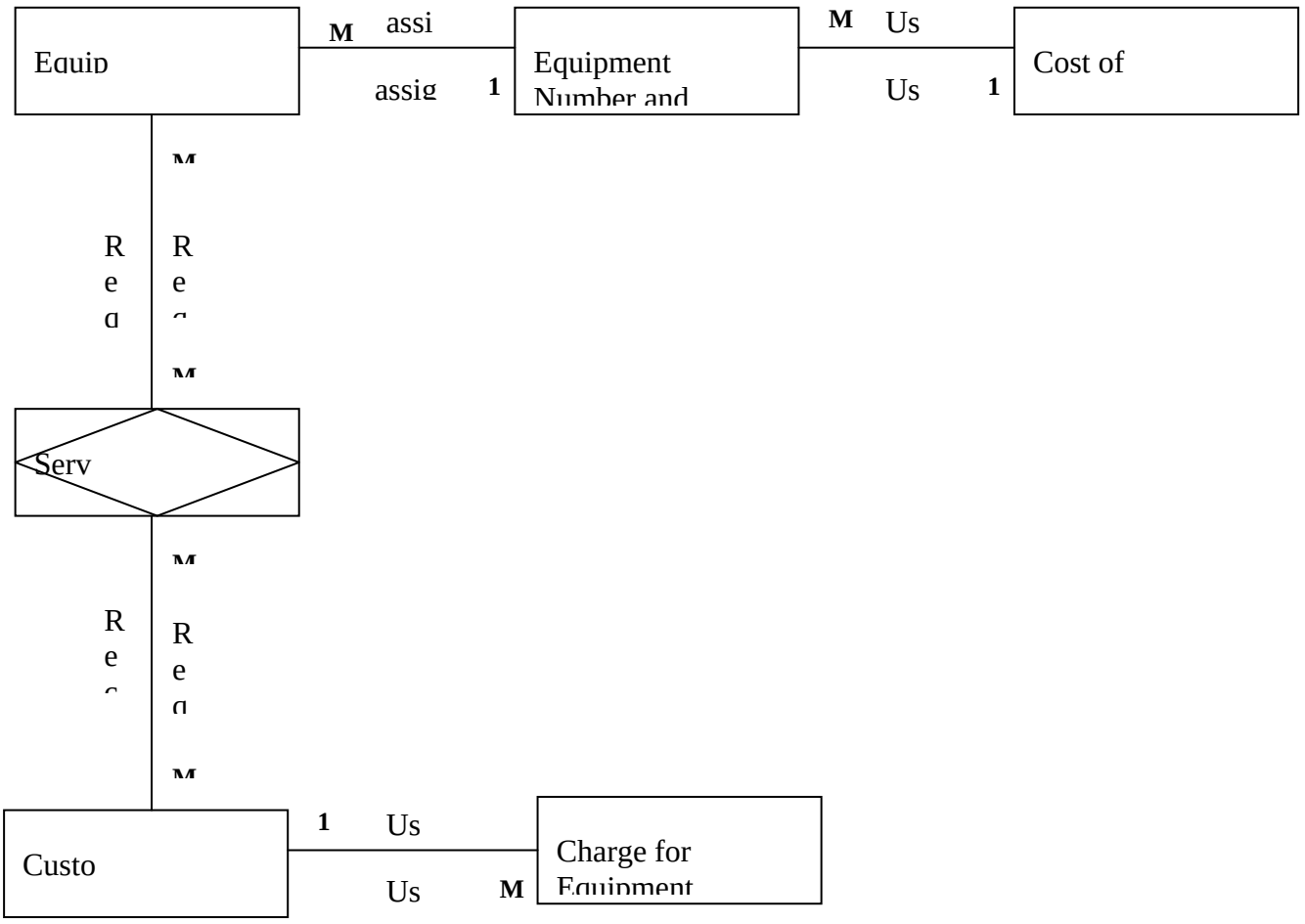
This one is close as it has the main entities (customer, account, equipment, services), but it allows for the many to many relationship. Site is listed as an attribute of customer which could work, but it would require that a customer could have only one site. In addition, the charge per service would be an attribute, not an entity.

Flawed Example 7:



This example is similar to Flawed Example #5 as that it has everything related to account. The basic concept that everything is related to the account is correct. However, a composite entity is created from two entities to correct for a many to many relationship. Account cannot be a composite for all the other entities and it would have a lot of repeating data in this fashion.

Flawed Example 8:



The problems with this should be immediately visible. Charges, Equipment Number, and Costs are **attribute**s of either Equipment or Service. There is no Account entity. Service is not a composite and is not related to Customer or Equipment. The relationship between Customer & Service and Equipment & Service is listed as a many to many and a composite is designed to eliminate that relationship.

Grading:

Students received the full score if the they provided an ER diagram using a standard ER Diagram technique with either the entities as diagramed in the Good or OK examples. Points were deducted for each of the following reasons:

- Missing or unnecessary entities
- Labeling attributes as entities
- Wrong associations
- Wrong Cardinalities

Improper use of Diagram Techniques

Please contact the instructor directly if you have any questions about your grade.