Name and Student Number: _____

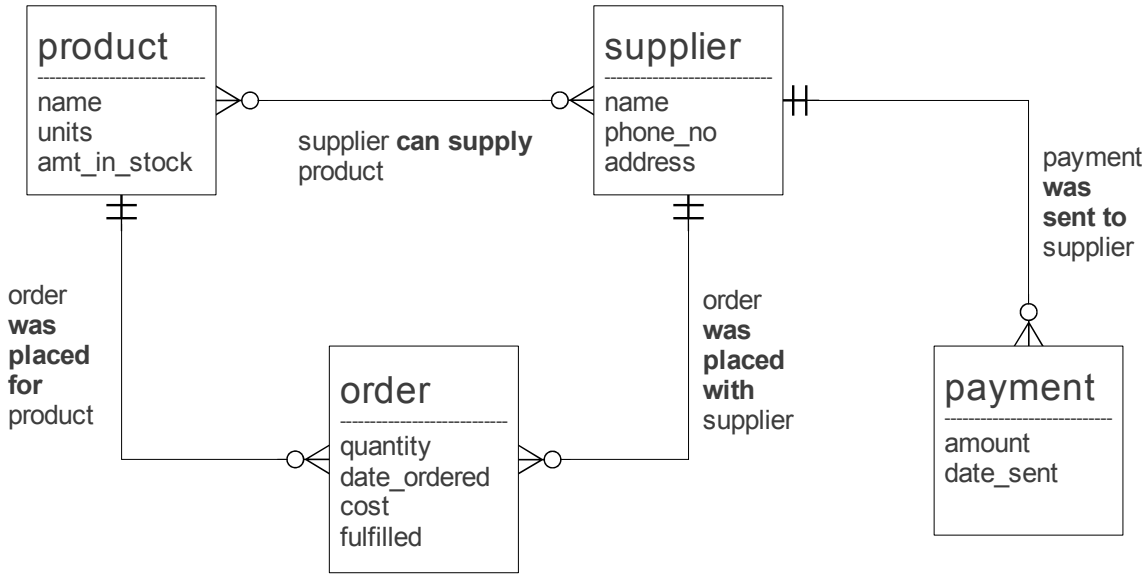
Quiz 2

This quiz consists of **two** questions.

Yoda's Yogurt now wants to expand its database. Instead of just keeping track of what yogurt they happen to have in stock at the moment, the store wants to also be able to use the database to keep track of suppliers for ingredients (milk, sugar, flavorings, etc.) and toppings (chocolate chips, M&Ms, different kinds of nuts, etc.). With the help of the new database, management will know who to call when supplies are running low. They'll also know whether an order has already been made and if so when and for what quantity. They should also be able to know whether they owe money to any of their suppliers.

1. Draw an ER diagram for the new system. The new system should be able to support all features mentioned above and does not need to do anything beyond it. (In particular, your diagram does *not* need to include the yogurt_in_stock table from the last quiz.) Make sure to label all relationships and to mark cardinality.

Here is one plausible diagram:



Notes:

- We know we need to represent suppliers somehow. One of the required features is that we'll know who to call. Where do we store that phone number? The supplier.
- We also need to represent the different toppings and ingredients. Those could in theory be two different entities, but there is actually no reason for that. There is no meaningful differences between "toppings" (such as chocolate chips) and "ingredients" (such as milk and sugar) in terms of how they appear in our database. All of them are things we use up in the course of operating the shop and need to reorder from suppliers. So, it makes sense to treat them as one entity: "product" (or some other name like that).
- The relationship between product and supplier is M:M. A supplier can supply us with more than one product and we might have multiple suppliers for the same product.
- We also need to track orders. In the most basic setup an order would represent our request to a particular supplier to send us a certain quantity of a particular ingredient at a particular time. The order entity thus

needs to be related to both “product” and “supplier”, with a 1:M relationship – M on the side of the order and 1 on the side of product and supplier.

- Finally, we need to track how much money we owe to our suppliers. This would require knowing how much money we were supposed to pay them (the total cost of all fulfilled orders) vs. how much money we already paid. This means we need to track payments somehow, hence a payment entity. Each payment needs to be associated with a particular supplier, but a supplier can get multiple payments.
- This is a minimalistic “5 minute” ER diagram for the problem. A more thorough analysis might lead us to expand it.
- This was a rather difficult problem, however, so I gave a lot of partial credit.

Common mistakes:

- Missing labels. Not labeling relationships potentially hurts you twice. First, labels are required, so failure to add them reduces your grade. Second, it is hard to understand the logic in diagrams without labels, so failure to add labels reduces the opportunities for partial credit.
- Wrong cardinality. Some diagrams suggest that we can only one issue one order to a supplier. Why can’t we go back to the same supplier if we liked them? We can only order a particular supply once. But what happens when we run out of chocolate chips we ordered yesterday? Also, whenever you are making a relationship as “1:1,” you should ask yourself if this is really what it’s supposed to be.
- Wrong attributes. One specific mistake was adding stocks of specific kinds of ingredients (“milk”, “sugar”) as attributes. Those should not appear as attribute names. Rather, they’ll be *values*, where the attribute is going to be called something like “name”.
- An ER diagram does not *need* to include primary and foreign keys. We express those things with relationships. (I didn’t not reduce your grade for including PKs and FKs, unless they were clearly incorrect.)
- Some diagrams included unnecessary or irrelevant entities. (I did not reduce the grade for this unless those entities had their own mistakes.)

2. What are primary and foreign keys and how do we use them to convert an ER diagram into a relational form? *Please write your answer on the other side of the sheet.*

A *key* is a field or a set of fields in a table that can uniquely identify a row. *(Uniquely identify here means that no two rows can have the same value for the key.)* A primary key (PK) is a key that has been chosen for identifying rows in a particular table. A foreign key (FK) is a field (or a set of fields) in a table which contain values from the PK of some other table. *(A FK is not actually a “key” in that several rows can have the same value in a foreign key.)* A FK-PK pair creates a link between two tables, with each row in one of the tables storing a value (or a set of values) that identify a specific row in the other table. This allows us to implement 1:M relationships by giving the table that represents the entity on the “M” side a FK to the table that represents the entity on the “1” side.

(The part of the answer shown in italics is for clarification. I didn’t not expect this level of detail in your answers.)