CS 386/586 Exercise 4

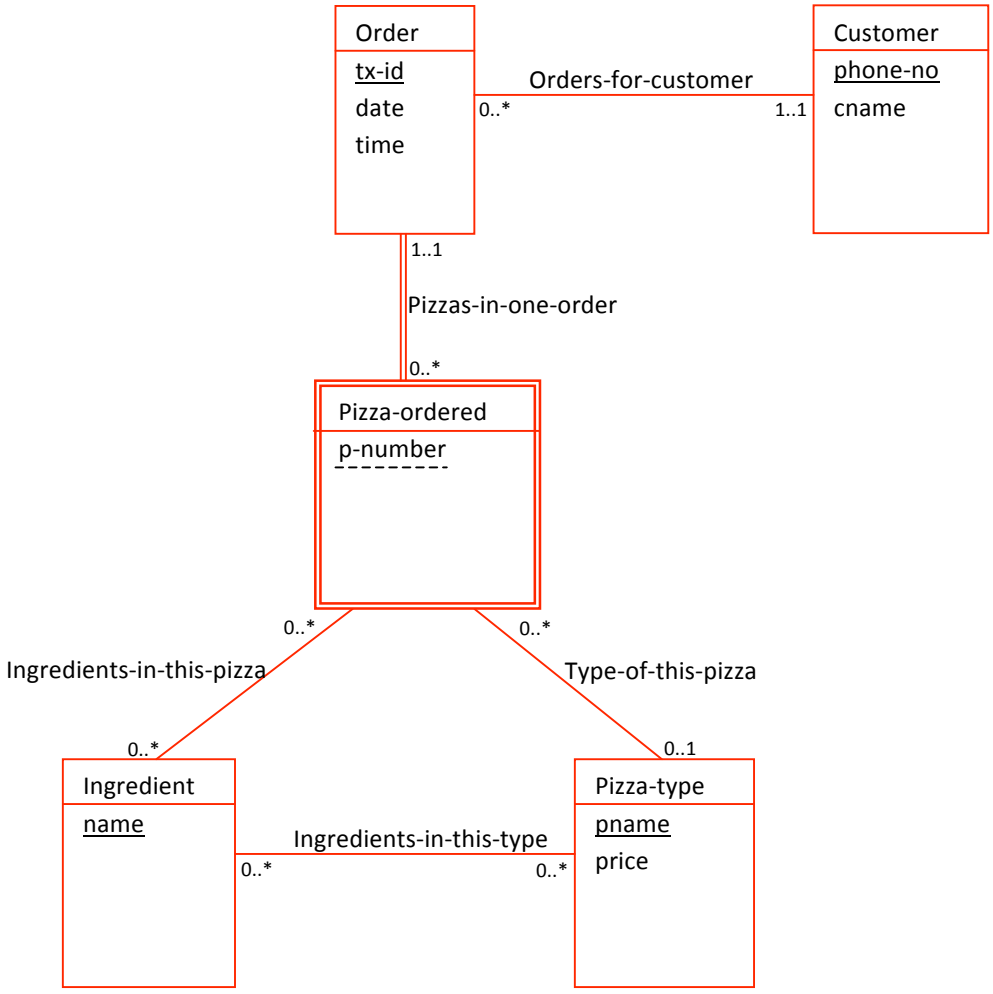
Work through the exercises Undergraduate students, please turn in your paper (with papers from all other exercises) at the end of the term.

CS 386/586 Fall 2007 Exercise 15 November 15, 2007 Undergraduate students, please turn in your paper (with papers from all other exercises) at the end of the term.

1. Draw an ER diagram for the database needed by a pizza store, described below.

The pizza store sells pizzas that you can take home and bake in your own oven. The pizza store offers some number of ingredients (like pepperoni, onion, bell pepper, Canadian bacon, etc.). The pizza store offers some number of pre-defined pizza types (like a Hawaiian pizza type is one with Canadian bacon and pineapple). The pizza store also allows you to choose your own ingredients (without choosing a pizza type). Assume that there is only one size pizza at this pizza store. The pizza store keeps track of individual pizza orders – indicating either the pizza type of the order or the specific ingredients requested by the customer – for each pizza that is ordered. The pizza store keeps track of individual orders – that include one or more pizza orders. And finally the pizza store keeps track of customers. Each order is connected to a customer.

Draw your ER diagram. Then turn this page over to see an ER diagram that I drew for the pizza store.



Note that it is important to have an Entity for Order (representing all the things that one customer buys during one visit to the pizza store) and also an entity for Pizza-order (representing one particular pizza that a customer bought). I have made the following assumptions. Each Pizza-ordered has either one Pizza-type or a set of ingredients. If a Pizza-ordered has a Pizza-type, then it won't have additional ingredients and it won't omit any of the ingredients that normally come up this Pizza-type. (You could allow this and consider how the ERD needs to be changed.) Note that the constraint that a given Pizza-ordered has EITHER a Pizza-type OR a set of Ingredients is not expressible in an ERD. You would have to rely on the application program to enforce that constraint.

Notice the double-line used for Pizza-ordered entity and the double-line used for the Pizzas-in-one-order relationship. This is used to indicate that Pizza-ordered is a weak entity with the Order as the corresponding strong entity and Pizzas-in-one-order as the identifying relationship. The p-number attribute in the Pizza-ordered entity is just a sequence number (1, 2, …) to indicate whether it was the first, second, or third pizza in a particular Order. This means that p-number is a partial key for the Pizza-ordered entity.

2. Explain why the Pizza-ordered entity is a weak entity. (And notice that a weak entity can participate in other (regular) relationships, in addition to participating in the identifying relationship.)

3. Define the tables that you need in order to represent this ERD in a relational database schema. Be sure to include all attributes that you need in your tables. Be sure to indicate all keys and foreign keys.

For each of the following tables, answer the following questions.

a. Explain what this table is about. In other words, explain what this table represents in the application. (Hint: look at the key for this table and then consider how the remaining attributes in the table describe the rows in this table.)
b. List all the FDs implied by the key(s) for this table.
c. List all the FDs from the original given list of FDs for this table that are NOT implied by a key.
d. Choose one of the FDs listed in part c. and decompose this table into two tables using the decomposition algorithm for normalization.
e. If there are any remaining FDs (i.e., troublesome FDs) that are not implied by the key(s), continue the decomposition process, step-by-step, until there are no remaining FDs that are not implied by the key.
f. Give one reason why the designer might choose NOT to decompose (i.e., choose NOT to normalize) the original table.

This application is a pizza store with several pre-defined pizza types (like Hawaiian that includes Canadian bacon and pineapple as its only ingredients). The employee is the clerk who took the order for the pizza.

1. Pizza-order (customer-id, pizza-type-id, customer-phone-no, date, time, employee-id, price)

   with the FDs: customer-id → customer-phone-no pizza-type-id → price

2. Pizza-order(customer-id, pizza-type-id, date, time, employee-id, employee-name) with the FD: employee-id → employee-name

3. Table(customer-id, employee-id, total-sales, employee-ssn) This table has two keys listed below. Typically, we don't use underlining for two keys because it is difficult to express what is going on. (customer-id, employee-id) is a key for this table (customer-id, employee-ssn) is a key for this table with the FDs: employee-id → employee-ssn employee-ssn → employee-id