

EXERCISE 9-1

Here we have the ever-useful “customer orders product” example again. Design the table that will represent the `Order` class in Figure A-37. Consider constraints, primary and foreign keys, and updating rules.

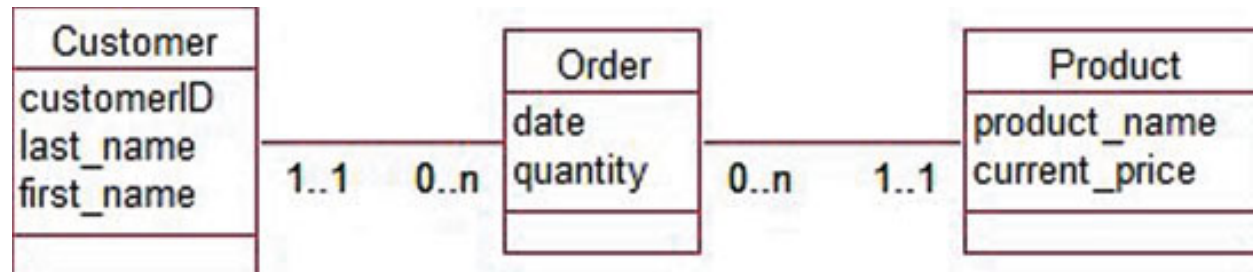


Figure A-37. Model for customers placing orders

Let’s assume that `customerID` and `product_name` are primary keys for the `Product` and `Customer` tables. To represent the two relationships, these primary keys will be included as new foreign key fields in the `Order` table. We will have (foreign keys in italics):

`Order` (*customer*, *product*, date, quantity)

What is the primary key? A customer might order a product more than once, possibly on the same day and even for the same quantity. (You might order three dozen savories for your party, realize you’ve miscalculated the appetites of certain of your guests and immediately order another three dozen.) This is a (fairly rare) case where no combination of attributes can be guaranteed to be unique and a generated order number needs to be added. All

attributes are necessary and so should have a NOT NULL constraint. If the date is when the order is placed then it should not be in the future—but if it is the date the order is to take effect, maybe a future date makes sense—let’s not tie this down just now. As for updating rules, we must choose Disallow Delete (the default). If a customer or product is deleted we will still want to retain the order for accounting purposes (and as discussed in Chapter 9, we will probably want to retain the product and customer as well). A Nullify Delete would render the order information useless as we would be left with data such as, “On a particular date someone ordered six of something.”

The SQL for creating the table would be something like that shown in Listing A-2.

Listing A-2. SQL to Create Order Table – Delete Disallowed is the Default Updating Rule

```
CREATE TABLE Order    (order_num INT PRIMARY KEY,  
                        product INT NOT NULL FOREIGN KEY REFERENCES Product,  
                        customer INT NOT NULL FOREIGN KEY REFERENCES Customer,  
                        quantity INT NOT NULL,date DATE NOT NULL,);
```

EXERCISE 9-2

Think about the options available for setting up tables for makes and models of cars at a car sale yard.

We have makes, models, types, and actual cars. If we wind right back to Chapter 2, it is useful to think about attributes by which users might like to sort or search and have these as separate classes. (We did this for genus in the plant case so we could ensure all the Eucalyptus species were spelled correctly). Here we are likely to want to be able to find information about all Fords, or all Siestas, or even all sedans or all blue cars. An initial model may look like the one in Figure [A-38](#).