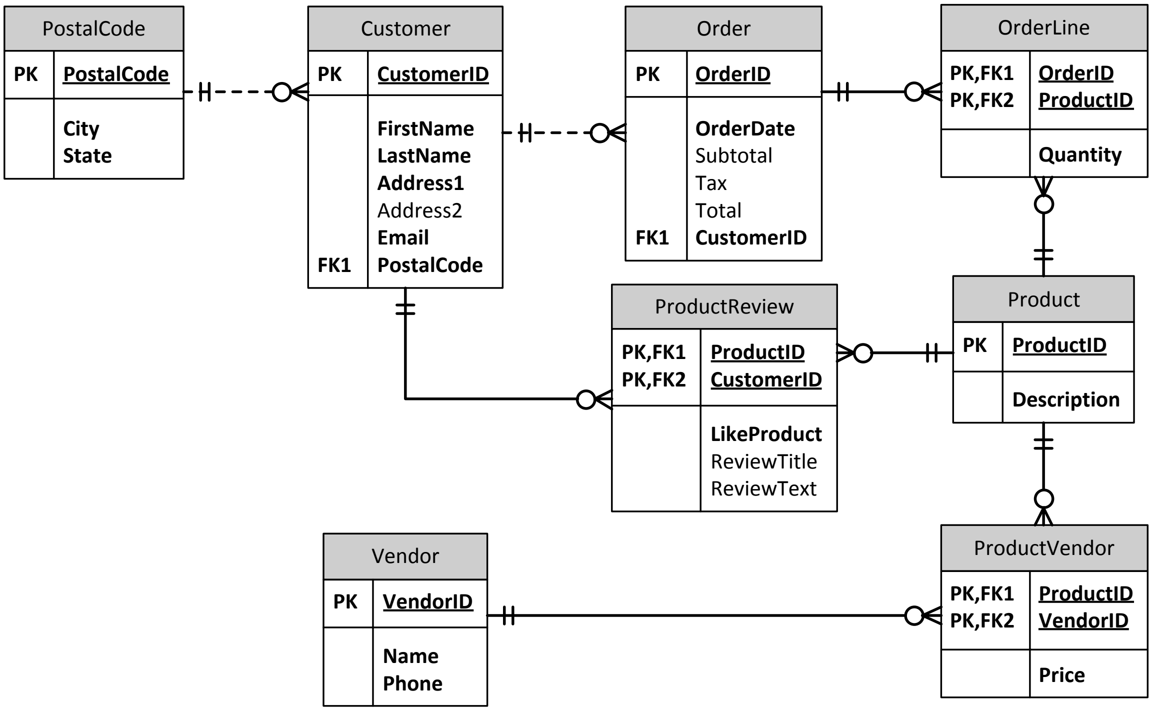
Overview

A fully normalized relational database is optimized to prevent modification problems, but this optimization typically increases the structural complexity of the relational data model, thereby decreasing query performance. *Denormalization* is the process of altering a relational data model to implement controlled data redundancy such that query performance will be improved. Denormalization can be accomplished in a number of ways, including by (1) merging two or more tables together, (2) by adding additional attributes to tables that will help to avoid computationally intensive *join* operations, or (3) by adding additional attributes to tables that store the pre-computed results of common mathematical tasks, thus avoiding the need to repeatedly perform such calculations at run-time.

This group assignment requires that you denormalize a relational data model in order to support the specific query needs of an organization.

Details

Below is a normalized (BCNF) data model that is designed to support the operations of an online retailer:



- The retailer now has over 3 million customers, and ships an average of 30,000 packages per day (i.e., approximately one package every 2.9 seconds). Every time a package is shipped, a mailing label must be generated that requires that the customer's contact information be extracted from the database. This is accomplished with the following SQL query (using CustomerID = 133 as an example):

```
SELECT C.FirstName, C.LastName, C.Address1, C.Address2,
       PC.City, PC.State, PC.PostalCode
  FROM Customer C INNER JOIN PostalCode PC
       ON C.PostalCode = PC.PostalCode
 WHERE C.CustomerID = 133;
```

- The online retailer's website allows customers to create and read product reviews, and the ProductReview table now contains over 10 million unique reviews. During the review process, a customer indicates whether they "like" a product, and this information is stored as a Bit (i.e., an integer data type whose value can only be a one or a zero) in the LikeProduct attribute (such that a value of *1* indicates that the customer likes the product, while a value of *0* indicates that the customer does not like the product). Whenever a website visitor views a product page, a query is run against the database to determine the percentage of customers who "like" the product. This is accomplished with the following SQL query (using ProductID = 5824 as an example). Note that values of LikeProduct in the query are multiplied by 1.0. This is a quick way of instructing the database engine to treat the values as decimals, thus avoiding integer division (which may produce inaccurate results). The *CAST* or *CONVERT* functions could also be used for this purpose, but I'm lazy, so I like this approach. ☺

```
SELECT AVG(LikeProduct * 1.0) AS LikePercentage
  FROM ProductReview
 WHERE ProductID = 5824;
```

- The online retailer retains several vendors for each product, and each vendor may charge a different price for a given product. This approach allows the retailer to use a dynamic pricing model that ensures that product prices remain competitive. Specifically, the lowest current vendor price for a given product is always chosen as the basis for computing the price that will be shown to customers on the retailer's website. For example, if the prices charged by three different vendors for a new smartphone were $540, $529, and $562, then the retailer would want to use $529 (the current minimum) as the basis for computing the price that it will charge its customers for the smartphone. This is accomplished with the following SQL query (using ProductID = 7741 as an example):

```
SELECT ProductID, Description,
       MinPrice = (SELECT MIN(Price)
                     FROM ProductVendor
                    WHERE ProductID = P.ProductID)
  FROM Product P
 WHERE ProductID = 7741;
```

Tasks

Your tasks for this assignment are to:

1. Create a denormalized entity-relationship data model using Microsoft Visio that will improve the performance of the business tasks that currently rely on the three queries above.
2. While working on your ERD, keep a record of:
   a. The changes that you made to the original data model, and
   b. Any new business rules that the online retailer will need to implement in order to maintain the integrity of the data in the denormalized database

Deliverables

To ensure that you receive credit for this assignment, please complete the following tasks:

1. Complete the Group Assignment Participation Form for this assignment (available on the course website). **Each group member who contributed to the assignment should complete this task.**
2. Assemble your Visio ER diagram, your record of data model modifications, and your list of new business rules from Tasks 1 and 2 above into a single Word document. Note that after selecting your ER diagram in Visio and copying it to the clipboard, you can paste the diagram into Word as an image by selecting *Paste Special >> Picture (Enhanced Metafile)*. After creating your Word document, submit the file using the appropriate link on the course website. **Each group should submit just one file.**