

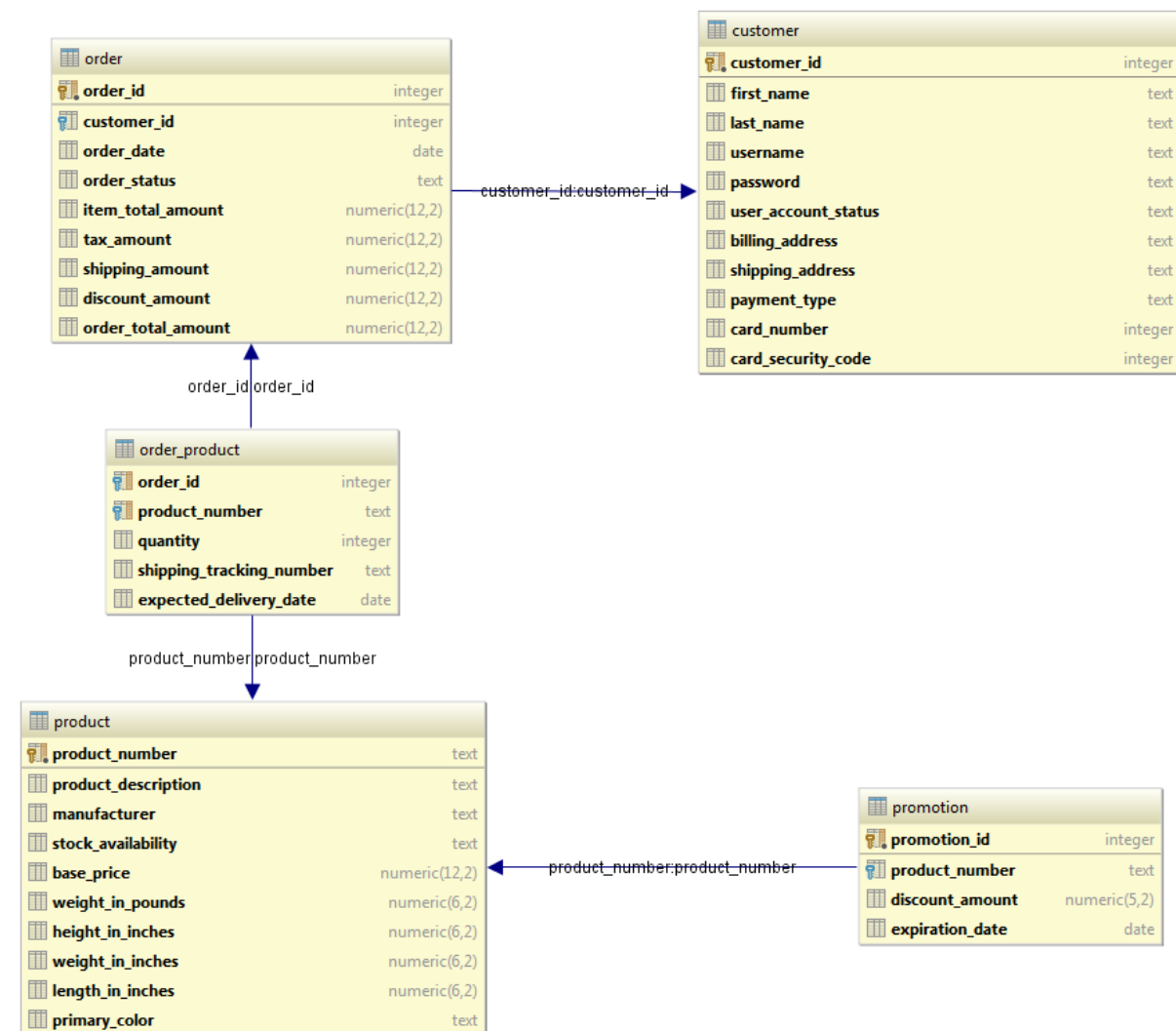
BACKGROUND

The Business Technology or "BizTech" team at Nodal Exchange is responsible for testing the software applications that support our business. As many of our tests require advanced database skills to set up and validate test cases, it is important for team members to be proficient SQL writers. This exercise aims to replace a first-round interview by testing your SQL skills against an example data model.

It is expected that interested candidates spend roughly 30-60 minutes on the exercise (equivalent to the time spent in a first-round interview). You are welcome to spend more, but our experience has been that if you spend significantly more than one hour, then there might be some strong fit questions that will likely show up during the in-person interview round.

MODEL

Below is a model of the database that an e-Commerce site uses to track customers, orders, and products. Explanations of the tables and the symbols used can be found on the next page.



MODEL GUIDE

The elements of each table are displayed as follows:

- Table Name is the very top row in the table, with gray background and a table icon on the left.
- Primary Key is the line below the table name, with yellow background and a table icon with an orange key in it on the left. There is a line below the PK. Not all tables have a PK.
- Other columns are listed below the PK. Each column has a simple table icon on the left. A blue key in this simple table icon indicates that the column is a Foreign Key from another table.

With PK and other columns, data type is listed on the right-hand side. Numeric values use a (precision, scale) notation where precision is the total number of significant digits in the number and scale is the number of decimal places. For example, a data type of *numeric (3, 2)* indicates that the number has two decimal places and, therefore, one number before the decimal.

An arrow between the tables indicates a Foreign Key. The table with the open end of the arrow has a column that refers to a column in the table with the arrowhead. The names of these columns are displayed along the arrow; for simplicity, all FKs use the same column name as the source column.

TABLE GUIDE

- *CUSTOMER* - Contains information for the site's customers, including biographical information, site login credentials, and saved payment and delivery information.
- *ORDER* - Contains information on each order placed on the site. Each record has the customer who placed the order, the date it was placed, and its status (e.g., "Shipped"). It also has the total amount charged as well as the individual components of that charge stored in their own columns.
- *PRODUCT* - Contains the products listed for purchase on the site. The *base_price* is the price of the product before discounts and additional fees, such as taxes, are applied.
- *ORDER_PRODUCT* - Lists products that were purchased as part of each order (i.e. an association table). A given order may be comprised of one or more products.
- *PROMOTION* - Contains the discounts that are or have previously been offered on products. There could be many or no promotions offered on a given product at any time.

ASSIGNMENT

1. Write a query that retrieves all orders made by the customer with *customer_id* 123456, from most to least recently placed.
2. Write a query that shows the date of the most recently placed order for each customer in the database.
3. Write a query that gives the total number of products purchased over the previous 12 months.
4. Write a query that gives the three products yielding the highest revenue among all purchases over the previous 30 days.
5. Write a query or script that validates the total amount charged for each order is correct, assuming: (a) the *base_price* of a product is constant over time; (b) a constant tax rate of 10%; and (c) a flat shipping fee of \$5 for each type of product shipped as part of an order (e.g., so that ordering three of one product incurs the same shipping fee as ordering just one of the product).
6. Describe some ways in which you would improve upon the given data model. What additional data fields would you capture? What logical flaws would you correct and how? In which ways would you extend the model?