

Directions: Create stored queries and procedures.

1. The store manager suspects that someone in customer service is harvesting customer emails and selling them on the dark web. The manager tells you their suspicion. What do you suggest? Write some SQL to handle this situation, but that still permits agents to view the other customer information, including name and address.
2. Create a view containing order information, in particular: date, customer name (first and last), address, phone number, email, product id and name, quantity, and MSRP. Hint: you will need to join tables.
3. Write a statement that creates a summarized view of the average of products in each order. Include the order id.
4. Create a summarized view containing the average of products in each order, but base it on the view created in Questions 2.
5. Create a view of all customers from West Virginia.
6. Create a view, `WVOrders`, with order details from West Virginia using the view created in Problem 5. Include order id, customer id, product id, and quantity ordered.
7. Create a stored procedure that returns order information (i.e., date, customer name, address, phone number, email, product id and name, quantity, and MSRP) for a given order number. Call your procedure on Order 1. Hint: See Problem 2.
8. Update `Customers` table by setting `address2` to `NULL` if it is empty. Then write one (or more) stored procedure that accepts an order number (i.e., order id) and does the following.

(a). Return a shipping label with customer information in the following format:

```
Joe Smith
123 Park Way
City, State Zip
```

If there is a second address, that should be included as well. In particular,

```
Joe Smith
123 Park Way
Apt A.
City, State Zip
```

(b). Return the order total including 8% sales tax and a \$10.00 flat rate shipping fee.

- (c). Update the inventory levels by subtracting each product's quantity from the **Products** table.
- 9. Create, then call, a procedure that accepts two integers as input parameters and returns the sum and product.
- 10. Create indexes on the first and last name fields of the **Customers** table.