Melissa Hoffman

FSW140

7/1/21

For this assignment, write queries using SQL to acquire data about customers, vendors, products, and employees in a fictitious sales database. These queries will cover many of the core aspects of writing SQL to produce data for reporting and analyzing information. There may be multiple ways to produce the same results, but ensure you are returning the requested fields.

Using the Sales Orders database, complete the queries below.

1. **Show all the information on our customers**.
2. Query: SELECT \* FROM customers;
3. Columns: 9 Columns - CustomerID | CustFirstName | CustLastName | CustStreetAddress | CustCity | CustState | CustZipCode | CustAreaCode | CustPhoneNumber
4. Expected Row Count: 28 Rows
5. Screenshot:

Graphical user interface, application

Description automatically generated

1. **Show a list of states, in reverse alphabetical order, where our vendors are located, and include the names of the vendor.**
2. Query: SELECT VendName, VendState FROM vendors

ORDER BY VendState DESC;

1. Columns: 2 Columns - VendName | VendState
2. Expected Row Count: 11 ROWS
3. Screenshot: A picture containing text, screenshot, indoor, computer

   Description automatically generated
4. **What if we adjusted the retail price of each product by increasing it 7 percent?**
5. Query: SELECT productname, retailprice, (round(RetailPrice \* 1.07))

AS NewPrice FROM products;

1. Columns: 3 columnsproductname | retailprice | NewPrice
2. Expected Row Count: 40 rows
3. Screenshot:

A picture containing text, screenshot, indoor, computer

Description automatically generated

1. **Show a list of orders made by each customer in ascending date order.**
2. Query:

SELECT \* FROM orders GROUP BY CustomerID ORDER BY OrderDate ASC;

1. Columns: 6 columns

OrderNumber | OrderDate | ShipDate | CustomerID | EmployeeID | OrderTotal

1. Expected Row Count: 27 rows
2. Screenshot:

Graphical user interface, text, application

Description automatically generated

1. **Give the names of all vendors based in Albany, Anchorage, and Dallas.**
2. Query: SELECT VendName, VendCity, VendState FROM vendors  
    WHERE VendCity IN ('albany', 'anchorage', 'dallas');
3. Columns: 3 columns - VendName, VendCity, VendState
4. Expected Row Count: 3 ROWS
5. Screenshot:A computer screen shot

   Description automatically generated with low confidence
6. **Show an alphabetized list of products with a quantity on hand greater than or equal to 30.**
7. Query: SELECT \* FROM products WHERE QuantityOnHand >= 30

ORDER BY ProductName ASC;

1. Columns: 6 COLUMNS - ProductNumber| PoductName | ProductDescription | RetailPrice | QuantityOnHand | CategoryID
2. Expected Row Count: 9 ROWS
3. Screenshot:Graphical user interface, text, application

   Description automatically generated
4. **What vendors do we work with that don’t have an email address?**
5. Query: SELECT \* FROM vendors WHERE VendEMailAddress IS NULL;
6. Columns: 10 COLUMNS – VendorId | VendorName | VendStreetAddress | VendCity | VendState | VendZipCode | VendPhoneNumber | VendFaxNumber | VendWebPage |VendEMailAddress
7. Expected Row Count: 9 ROWS
8. Screenshot:Graphical user interface, text, application, email

   Description automatically generated
9. **List employees and the dates their orders shipped sorted by order date.**
10. Query: SELECT employees.EmployeeID, EmpFirstName, EmpLastName, orderdate, shipdate FROM employees

JOIN orders ON employees.EmployeeID = orders.EmployeeID

ORDER BY orders.OrderDate;

1. Columns: 5 columns – EmployeeID | EmpFirstName | EmpLastName | orderdate | shipdate
2. Expected Row Count: 944 rows
3. Screenshot:A picture containing text, screenshot, indoor

   Description automatically generated
4. **Show the vendors and products they supply to us for products over $75 for vendors in Texas.**
5. Query: SELECT vendors.VendorID, vendors.VendName, vendors.VendCity, vendors.VendState, pv.productnumber, products. productname, pv.WholesalePriceFROM vendors

JOIN product\_vendors pv ON vendors.vendorid = pv.vendorid

JOIN products ON pv.ProductNumber = products.ProductNumber

WHERE VendState LIKE 'TX' AND pv.WholesalePrice > 75;

1. Columns: 7 COLUMNS – VendorID | VendName |VendCity | VendState | productnumber | productname| WholesalePrice
2. Expected Row Count: 12 rows
3. Screenshot:A picture containing text, screenshot, indoor

   Description automatically generated
4. **Show employees who live in the same city and state as our vendors.**
5. Query: SELECT EmployeeID, EmpFirstName, EmpLastName, EmpCity, EmpState, vendors.VendName, vendors.VendCity, vendors.VendState FROM employees

INNER JOIN vendors ON employees.EmpCity = vendors.VendCity AND employees.EmpState = vendors.VendState

ORDER BY EmpState;

1. Columns: 8 columns – EmployeeID | EmpFirstName | EmpLastName | EmpCity | EmpState | VendName | VendCity | VendState
2. Expected Row Count: 2 rows
3. Screenshot:Graphical user interface, text, application

   Description automatically generated
4. **Display customers who have no sales rep (employees) in the same state.**
5. Query: SELECT DISTINCT CustFirstName, CustLastName, CustState

FROM customers

INNER JOIN employees on employees.EmpState != customers.CustState;

1. Columns: 3 columns – CustFirstName | CustLastName | CustState
2. Expected Row Count: 28 rows
3. Screenshot:A picture containing text, screenshot, indoor, computer

   Description automatically generated
4. **What is the average quoted price of a helmet?**
5. Query: SELECT AVG(round(RetailPrice)) AS 'Avg Helmet $'

FROM products

WHERE ProductName LIKE '%helmet';

1. Columns: 1 column - Avg Helmet $
2. Expected Row Count: 1 row
3. Screenshot: Graphical user interface, text, application

   Description automatically generated
4. **What was the date of the earliest ship date?**
5. Query: SELECT MIN(ShipDate) FROM orders;
6. Columns: 1 COLUMN - MIN(ShipDate)
7. Expected Row Count: 1 ROW
8. Screenshot: Graphical user interface, text, application

   Description automatically generated
9. **What is the total amount (in dollars) of orders from the state of Oregon?**
10. Query: SELECT CustState, SUM(orders.OrderTotal) AS OrderTotal

FROM customers

INNER JOIN orders ON customers.CustomerID = orders.CustomerID

WHERE CustState = 'OR';

1. Columns: 2 columns – CustState | OrderTotal
2. Expected Row Count: 1 row
3. Screenshot: Graphical user interface, text, application

   Description automatically generated
4. **Show each employee, the employee’s total sales (in dollars), the employee’s total sales item quantity, and the average item sales price ordered by the employee’s average item sales price highest to lowest.**
5. Query:

SELECT orders.employeeid, sum(quotedprice) AS TotalSales, sum(quantityordered) as QuantityOrder,

round(AVG(quotedprice)) AS AvgSales FROM order\_details

JOIN orders ON order\_details.ordernumber = orders.ordernumber

JOIN employees ON employees.employeeid = orders.employeeid

GROUP BY orders.employeeid

ORDER BY employeeid DESC;

1. Columns: 4 columns – employeeid | TotalSales | QuantityOrder | AvgSales
2. Expected Row Count: 8 rows
3. Screenshot: A computer screen shot

   Description automatically generated with low confidence