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
4. Expected Row Count: 28
5. Screenshot:
6. **Show a list of states, in reverse alphabetical order, where our vendors are located, and include the names of the vendor.**
7. Query: SELECT VendName, VendState FROM vendors
8. ORDER BY VendState Desc;
9. Columns: 2
10. Expected Row Count: 11
11. Screenshot:
12. **What if we adjusted the retail price of each product by increasing it 7 percent?**
13. Query: UPDATE Products SET RetailPrice = RetailPrice \* 1.07;
14. Columns: 6
15. Expected Row Count: 40
16. Screenshot:
17. **Show a list of orders made by each customer in ascending date order.**
18. Query: SELECT \* FROM orders
19. ORDER BY OrderDate ASC;
20. Columns: 6
21. Expected Row Count: 944
22. Screenshot:
23. **Give the names of all vendors based in Albany, Anchorage, and Dallas.**
24. Query: SELECT VendName from vendors
25. WHERE VendCity = 'Anchorage' OR VendCity = 'Albany' OR VendCity = 'Dallas';
26. Columns: 1
27. Expected Row Count: 3
28. Screenshot:
29. **Show an alphabetized list of products with a quantity on hand greater than or equal to 30.**
30. Query: SELECT \* FROM products
31. WHERE QuantityOnHand >=30
32. ORDER BY ProductName ASC;
33. Columns: 6
34. Expected Row Count: 9
35. Screenshot:
36. **What vendors do we work with that don’t have an email address?**
37. Query: SELECT \* FROM Vendors
38. WHERE VendEMailAddress IS NULL;
39. Columns: 10
40. Expected Row Count: 5
41. Screenshot:
42. **List employees and the dates their orders shipped sorted by order date.**
43. Query: SELECT \* FROM employees;
44. SELECT \* FROM orders;
45. SELECT e.EmpFirstName, o.EmployeeID, o.ShipDate
46. FROM employees AS e, orders AS o
47. WHERE e.EmployeeID = o.employeeID;
48. Columns: 3
49. Expected Row Count: 944
50. Screenshot:
51. **Show the vendors and products they supply to us for products over $75 for vendors in Texas.**
52. Query: SELECT \* FROM vendors;
53. SELECT \* FROM product\_vendors;
54. SELECT \* FROM products;
55. SELECT v.VendName, p.ProductName, pv.ProductNumber, v.VendState
56. FROM vendors AS v, products AS p, product\_vendors AS pv
57. WHERE v.VendorID = pv.ProductNumber
58. AND RetailPrice > 75
59. AND VendState = 'TX';
60. Columns: 4
61. Expected Row Count: 36
62. Screenshot:
63. **Show employees who live in the same city and state as our vendors.**
64. Query: SELECT \* FROM customers;
65. SELECT \* FROM employees;
66. SELECT c.CustFirstName, c.CustState, e.EmpState
67. FROM customers AS c, employees AS e
68. WHERE c.CustState != e.EmpState;
69. Columns: 3
70. Expected Row Count: 163
71. Screenshot:
72. **Display customers who have no sales rep (employees) in the same state.**
73. Query: SELECT \* FROM customers;
74. SELECT \* FROM orders;
75. SELECT \* FROM order\_details;
76. SELECT c.CustFirstName, c.CustState, e.EmpState
77. FROM customers AS c, employees AS e
78. WHERE e.EmployeeID IS NULL
79. AND c.CustState != e.EmpState;
80. Columns: 0
81. Expected Row Count: 0
82. Screenshot:
83. **What is the average quoted price of a helmet?**
84. Query: SELECT AVG(RetailPrice)
85. FROM Products
86. WHERE ProductName LIKE 'h%';
87. Columns: 1
88. Expected Row Count: 1
89. Screenshot:
90. **What was the date of the earliest ship date?**
91. Query: SELECT MIN(ShipDate) FROM orders;
92. Columns: 1
93. Expected Row Count: 1
94. Screenshot:
95. **What is the total amount (in dollars) of orders from the state of Oregon?**
96. Query:
97. Columns:
98. Expected Row Count:
99. Screenshot:
100. **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.**
101. Query: SELECT \* FROM order\_details;
102. SELECT \* FROM products;
103. SELECT e.EmpFirstName, e.EmpLastName, sum(o.OrderTotal), count(od.QuantityOrdered), AVG(o.OrderTotal)
104. FROM employees AS e
105. LEFT JOIN orders AS o
106. ON o.EmployeeID = e.EmployeeID
107. LEFT JOIN order\_details AS od
108. ON od.OrderNumber = o.OrderNumber AND o.OrderTotal AND o.OrderTotal
109. WHERE od.OrderNumber = o.OrderNumber AND od.QuantityOrdered
110. ORDER BY e.EmpFirstName;
111. Columns:
112. Expected Row Count:
113. Screenshot: