***Aimal Khan***

**FSW-140\_WEEK-4\_Sales-Order-Database:**

Using the Sales Orders database, complete the queries below.

**1.Show all the information on our customers**.

1. Query: USE SalesOrders;

SELECT \* FROM customers;

1. Columns: 9
2. Expected Row Count: All
3. Screenshot: Graphical user interface, text, application, email

   Description automatically generated

**2.Show a list of states, in reverse alphabetical order, where our vendors are located, and include the names of the vendor.**

1. Query: SELECT VendName, VendState FROM vendors ORDER VendState DESC;
2. Columns: 2
3. Expected Row Count: All
4. Screenshot:
5. Graphical user interface, text, application, email

   Description automatically generated

**3.What if we adjusted the retail price of each product by increasing it 7 percent?**

1. Query: UPDATE products VALUE (RetailPrice \*1.07);
2. Columns: 6
3. Expected Row Count: All
4. Screenshot:

Graphical user interface, text, application

Description automatically generated

1. Graphical user interface, text, application

   Description automatically generated

**4.Show a list of orders made by each customer in ascending date order.**

1. Query: SELECT orders.OrderNumber, orders.OrderDate, customers.CustFirstName, customers.CustLastName FROM orders

JOIN customers ON orders.CustomerID = customers.CustomerID

ORDER BY orders.OrderDate ASC;

1. Columns: 4
2. Expected Row Count: All in orders
3. Screenshot: Graphical user interface, text, application, email

   Description automatically generated

**5.Give the names of all vendors based in Albany, Anchorage, and Dallas.**

1. Query: SELECT VendName, VendCity FROM vendors WHERE VendCity IN ('Albany', 'Anchorage', 'Dallas');
2. Columns: 2
3. Expected Row Count: 3
4. Screenshot: Graphical user interface, text, application, email

   Description automatically generated

**6.Show an alphabetized list of products with a quantity on hand greater than or equal to 30.**

1. Query: SELECT \* FROM products WHERE QuantityOnHand >=30 ORDER BY ProductName
2. Columns: 6
3. Expected Row Count: 9
4. Screenshot: Graphical user interface, text, application, email

   Description automatically generated

**7.What vendors do we work with that don’t have an email address?**

1. Query: SELECT \* FROM venders WHERE VendEMailAddress IS NULL;
2. Columns: 9
3. Expected Row Count: 5
4. Screenshot: Graphical user interface, text, application

   Description automatically generated

**8.List employees and the dates their orders shipped sorted by order date.**

1. Query: SELECT EMP.EmpFirstName, EMP.EmpLastName, orders.ShipDate, orders.OrderDate FROM employees EMP JOIN orders ON orders.EmployeeID = EMP.EmployeeID ORDER BY orders.OrderDate;
2. Columns: 4
3. Expected Row Count: 944
4. Screenshot: Graphical user interface, text, application

   Description automatically generated

**9.Show the vendors and products they supply to us for products over $75 for vendors in Texas.**

1. Query: SELECT Ven.VendName, Pro.ProductName, PV.WholesalePrice FROM vendors Ven

JOIN product\_vendors PV ON Ven.VendorID = PV.VendorID

JOIN products Pro ON PV.ProductNumber = Pro.ProductNumber

WHERE Ven.VendCity = 'Dallas' AND PV.WholesalePrice > 75;

1. Columns: 3
2. Expected Row Count: 2
3. Screenshot: Graphical user interface, text, application, email

   Description automatically generated

**10.Show employees who live in the same city and state as our vendors.**

1. Query: SELECT DISTINCT VendState, employees.\* FROM vendors JOIN employees ON VendState = EmpState;
2. Columns: 10
3. Expected Row Count: 9
4. Screenshot: Graphical user interface, text, application, email

   Description automatically generated

**11.Display customers who have no sales rep (employees) in the same state.**

1. Query: SELECT \* FROM customers WHERE CustState != 'WA' AND CustState != 'TX';
2. Columns: 9
3. Expected Row Count: 11
4. Screenshot: Graphical user interface, text, application, email

   Description automatically generated

**12.What is the average quoted price of a helmet?**

1. Query: SELECT FORMAT(AVG(QuotedPrice), 2) FROM order\_details OD

JOIN products PRO ON PRO.ProductNumber = OD.ProductNumber

WHERE ProductName LIKE '%Helmet';

1. Columns: 1
2. Expected Row Count: 1
3. Screenshot: Graphical user interface, text, application

   Description automatically generated

**13.What was the date of the earliest ship date?**

1. Query: SELECT MIN(ShipDate) FROM orders;
2. Columns: 1
3. Expected Row Count: 1
4. Screenshot: Graphical user interface, text, application, email

   Description automatically generated

**14.What is the total amount (in dollars) of orders from the state of Oregon?**

1. Query: SELECT SUM(OrderTotal) FROM orders ORD

JOIN customers CUS ON CUS.CustomerID = ORD.CustomerID

WHERE CustState = 'OR';

1. Columns: 1
2. Expected Row Count: 1
3. Screenshot: Graphical user interface, text, application, email

   Description automatically generated

**15.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.**

1. Query:

SELECT CONCAT(EmpFirstName,' ', EmpLastName) As 'Full Name',

SUM(OrderTotal) AS 'Total Sales',

SUM(QuantityOrdered) AS 'Total Items',

FORMAT(AVG(RetailPrice),2) AS 'Sales Price'

FROM employees EMP

JOIN orders ON EMP.EmployeeID = orders.EmployeeID

JOIN order\_details OD ON OD.OrderNumber = orders.OrderNumber

JOIN products PRO ON PRO.ProductNumber = OD.ProductNumber

GROUP BY orders.EmployeeID

ORDER BY 4 DESC ;

1. Columns: 4
2. Expected Row Count: 8
3. Screenshot: Graphical user interface, text, application, email

   Description automatically generated