

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.

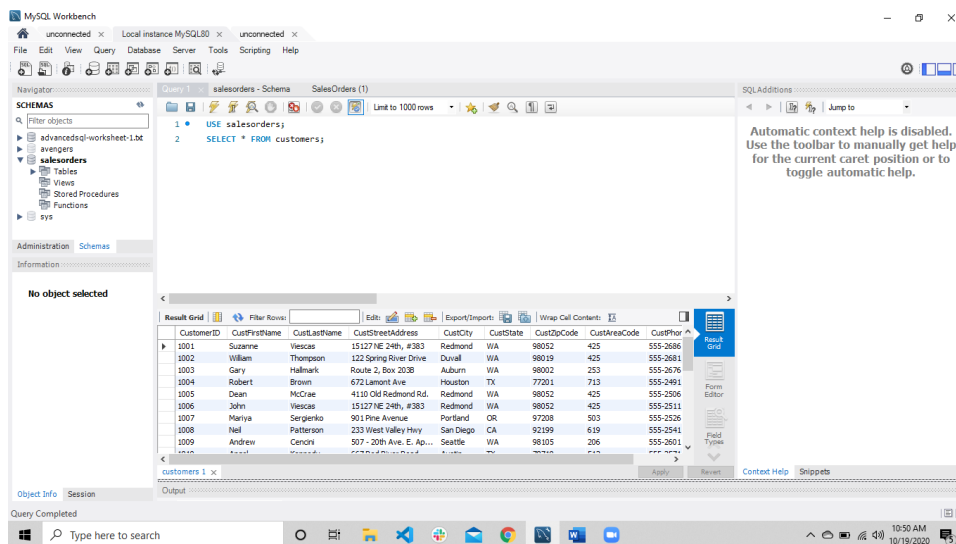
**1. Show all the information on our customers.**

a) Query: `SELECT * FROM customers;`

b) Columns: 9 Columns

c) Expected Row Count: 28 Rows

d) Screenshot:



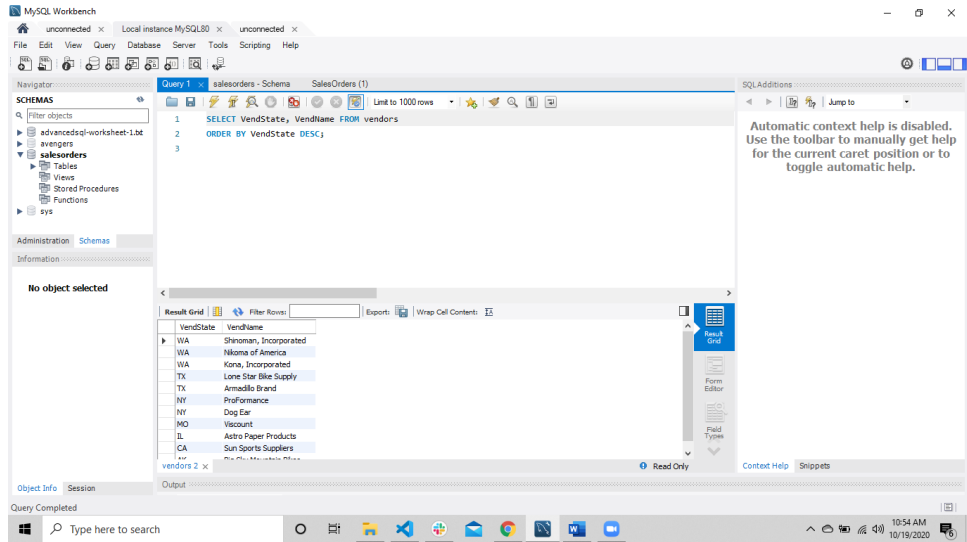
**2. Show a list of states, in reverse alphabetical order, where our vendors are located, and include the names of the vendor.**

a) Query: `SELECT VendState, VendName FROM vendors ORDER BY VendState DESC;`

b) Columns: 2 Columns

c) Expected Row Count: 11 Rows

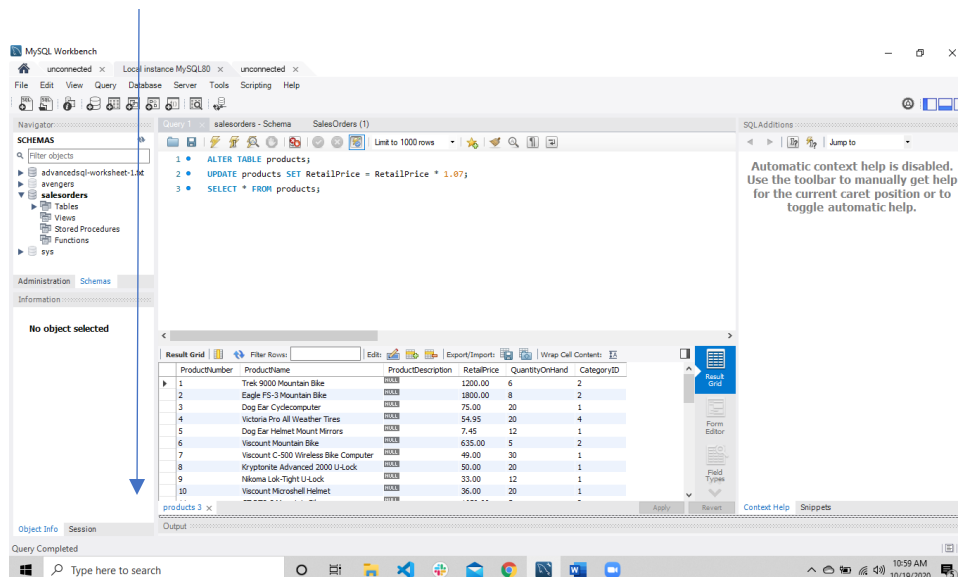
d) Screenshot:



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

- Query: 

```
ALTER TABLE products;
UPDATE products SET RetailPrice = RetailPrice * 1.07;
SELECT * FROM products;
```
- Columns: 6 Columns
- Expected Row Count: 40 Rows
- Screenshot:



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

- Query: 

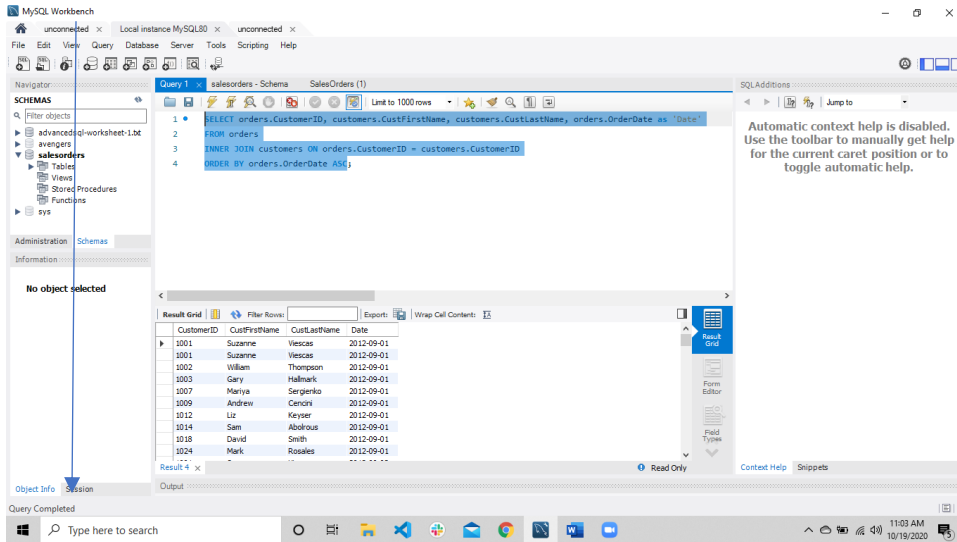
```
SELECT orders.CustomerID, customers.CustFirstName,
customers.CustLastName, orders.OrderDate as 'Date'
FROM orders
```

```
INNER JOIN customers ON orders.CustomerID = customers.CustomerID
ORDER BY orders.OrderDate ASC;
```

b) Columns: 4 Columns

c) Expected Row Count: 944 Rows

d) Screenshot:



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

a) Query: 

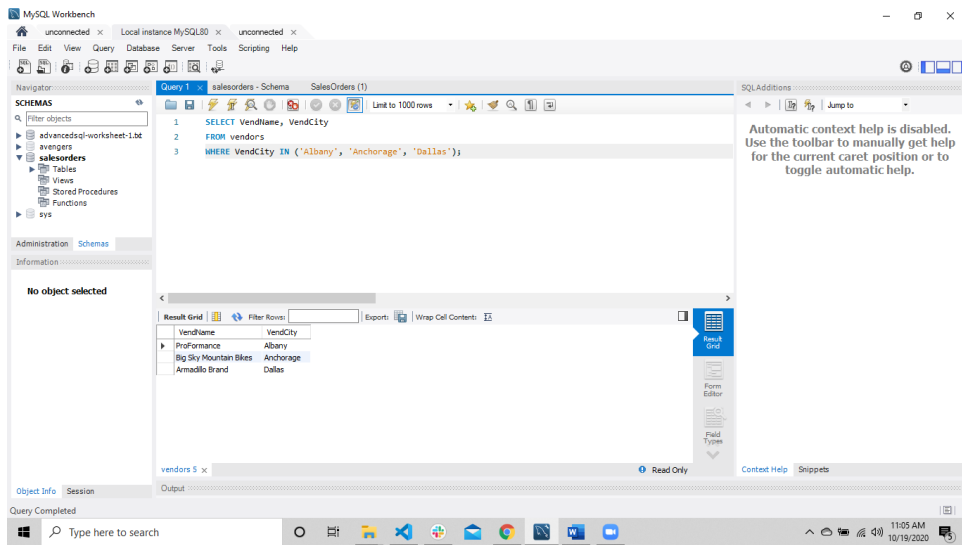
```
SELECT VendName, VendCity
FROM vendors
WHERE VendCity IN ('Albany', 'Anchorage', 'Dallas');
```

b) Columns: 2

c) Expected Row Count: 3 Rows

d) Screenshot:

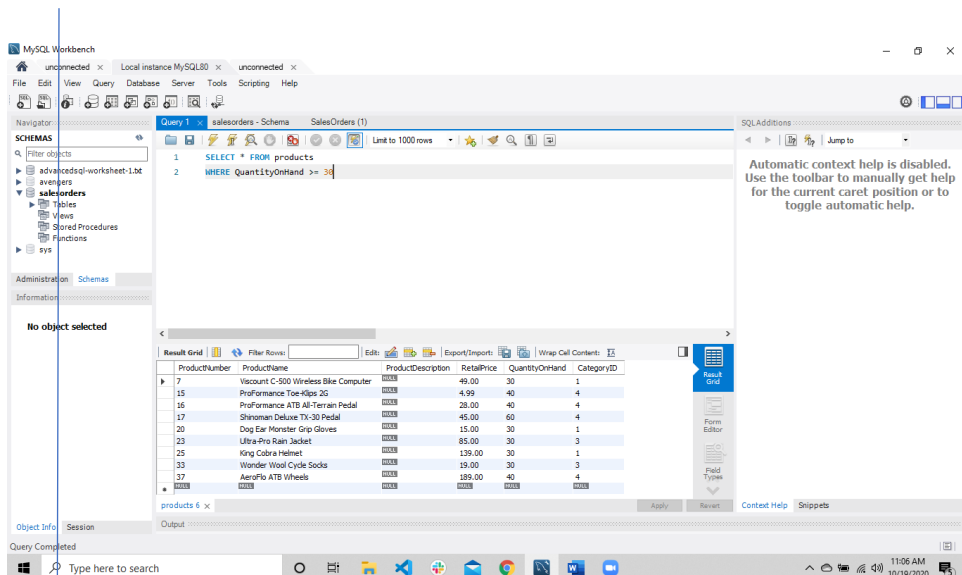




6. Show an alphabetized list of products with a quantity on hand greater than or equal to 30.

- Query:  

```
SELECT * FROM products
WHERE QuantityOnHand >= 30;
```
- Columns: 6 Columns
- Expected Row Count: 9 Rows
- Screenshot:



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

- Query: 

```
SELECT * FROM vendors
WHERE VendEmailAddress IS NULL;
```

b) Columns: 10 Columns

c) Expected Row Count: 5 Rows

d) Screenshot:

The screenshot shows a database query tool interface. At the top, a SQL query is entered:

```
1 • SELECT * FROM vendors
2 WHERE VendEmailAddress IS NULL;
```

Below the query, a "Result Grid" displays the results of the query. The grid has 10 columns: VendorID, VendName, VendStreetAddress, VendCity, VendState, VendZipCode, VendPhoneNumber, VendFaxNumber, VendWebPage, and VendEmailAddress. The results show 5 rows of data, all with NULL values for VendEmailAddress.

VendorID	VendName	VendStreetAddress	VendCity	VendState	VendZipCode	VendPhoneNumber	VendFaxNumber	VendWebPage	VendEmailAddress
6	Big Sky Mountain Bikes	Glacier Bay South	Anchorage	AK	99209	(907) 222-1234	(907) 222-1235	NULL	NULL
7	Dog Ear	575 Madison Ave.	New York	NY	10003	(212) 888-9876	(212) 888-9877	NULL	NULL
8	Sun Sports Suppliers	PO Box 8082	Santa Monica	CA	91003	(310) 777-9876	(310) 777-9877	NULL	NULL
9	Lone Star Bike Supply	7402 Kingman Drive	El Paso	TX	79915	(915) 666-9876	(915) 666-9877	NULL	NULL
11	Astro Paper Products	5639 N. Riverside	Chicago	IL	60637	(312) 555-9876	(312) 555-9875	NULL	NULL

At the bottom, the "Output" pane shows the execution details:

```
1 17:32:18 SELECT * FROM vendors WHERE VendEmailAddress IS NULL LIMIT 0, 1000
```

The message "5 row(s) returned" is displayed.

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

a) Query: 

```
SELECT orders.EmployeeID, employees.EmpFirstName,
employees.EmpLastName, orders.OrderDate as 'Date'
FROM orders
INNER JOIN employees ON orders.EmployeeID = employees.EmployeeID
ORDER BY orders.OrderDate ASC;
```

b) Columns: 4 Columns

c) Expected Row Count: 944 Rows

d) Screenshot:

The screenshot shows a SQL query editor with the following query:

```

1 SELECT orders.EmployeeID, employees.EmpFirstName, employees.EmpLastName, orders.OrderDate as 'Date'
2 FROM orders
3 INNER JOIN employees ON orders.EmployeeID = employees.EmployeeID
4 ORDER BY orders.OrderDate ASC;

```

The results are displayed in a table with the following columns: EmployeeID, EmpFirstName, EmpLastName, and Date. The table contains 23 rows of data, showing a list of employees and their order dates.

EmployeeID	EmpFirstName	EmpLastName	Date
701	Ann	Patterson	2012-09-01
702	Mary	Thompson	2012-09-01
703	Matt	Berg	2012-09-01
703	Matt	Berg	2012-09-01
703	Matt	Berg	2012-09-01
707	Kathryn	Patterson	2012-09-01
707	Kathryn	Patterson	2012-09-01
708	Susan	McLain	2012-09-01
708	Susan	McLain	2012-09-01
708	Susan	McLain	2012-09-01
701	Ann	Patterson	2012-09-02
702	Mary	Thompson	2012-09-02
704	Carol	Viescas	2012-09-02
704	Carol	Viescas	2012-09-02
706	David	Viescas	2012-09-02
706	David	Viescas	2012-09-02

The output section shows the following message:

```

1 17:43:36 SELECT orders.EmployeeID, employees.EmpFirstName, employees.EmpLastName, orders.OrderDate as 'Date' FROM orders INNER JOIN employee... 944 row(s) returned

```

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

- Query: 

```
SELECT vendors.VendName, products.ProductName, product_vendors.WholesalePrice, vendors.VendState FROM products JOIN product_vendors ON product_vendors.ProductNumber = products.ProductNumber JOIN vendors ON vendors.VendorID = product_vendors.VendorID WHERE product_vendors.WholesalePrice > 75 AND vendors.VendState LIKE 'TX';
```
- Columns: 4 Columns
- Expected Row Count: 12 Rows
- Screenshot:



The screenshot shows a SQL query execution interface. The query is as follows:

```

1  SELECT vendors.VendName, products.ProductName, product_vendors.WholesalePrice, vendors.VendState
2  FROM products JOIN product_vendors ON product_vendors.ProductNumber = products.ProductNumber
3  JOIN vendors ON vendors.VendorID = product_vendors.VendorID
4  WHERE product_vendors.WholesalePrice > 75 AND vendors.VendState LIKE 'TX';

```

The results are displayed in a table with the following columns: VendName, ProductName, WholesalePrice, and VendState. There are 12 rows of data.

VendName	ProductName	WholesalePrice	VendState
Lone Star Bike Supply	Trek 9000 Mountain Bike	854.22	TX
Lone Star Bike Supply	Eagle FS-3 Mountain Bike	1477.81	TX
Lone Star Bike Supply	Viscount Mountain Bike	448.73	TX
Lone Star Bike Supply	GT RTS-2 Mountain Bike	1178.65	TX
Lone Star Bike Supply	Eagle SA-120 Clipless Pedals	101.22	TX
Lone Star Bike Supply	King Cobra Helmet	105.29	TX
Lone Star Bike Supply	Cosmic Elite Road Warrior Wheels	122.78	TX
Lone Star Bike Supply	AeroFlo ATB Wheels	124.89	TX
Lone Star Bike Supply	Cycle-Doc Pro Repair Stand	136.98	TX
Lone Star Bike Supply	Ultimate Export 2G Car Rack	136.35	TX
Armadillo Brand	King Cobra Helmet	101.58	TX
Armadillo Brand	Cycle-Doc Pro Repair Stand	128.65	TX

The interface also shows an "Output" section with the following message:

```

# Time Action Message
1 18:03:17 SELECT vendors.VendName, products.ProductName, product_vendors.WholesalePrice, vendors.VendState FROM products JOIN product_vendors... 12 row(s) returned

```

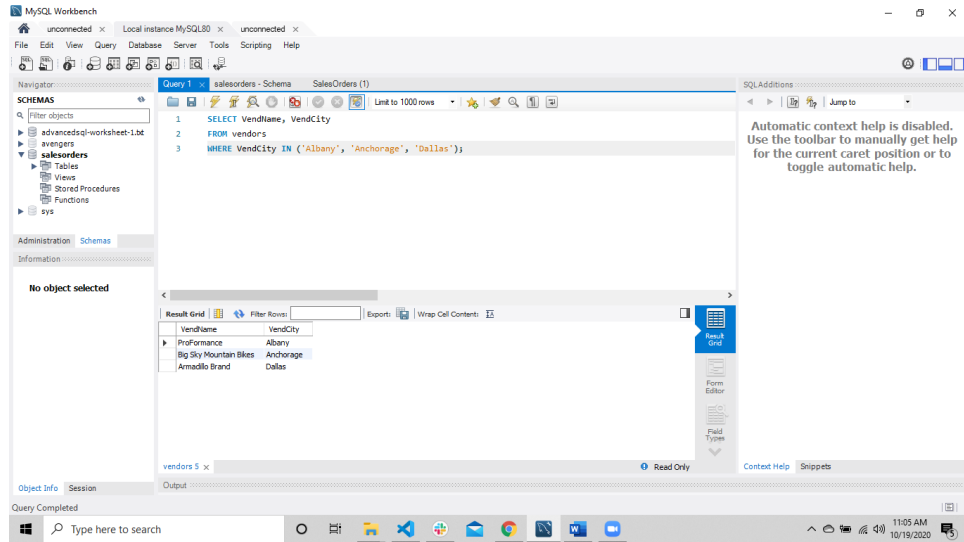
10). Show employees who live in the same city and state as our vendors.

- Query:
 

```

SELECT employees.EmpFirstName, employees.EmpLastName,
employees.EmpCity, employees.EmpState, vendors.VendName,
vendors.VendCity, vendors.VendState
FROM employees
INNER JOIN vendors ON employees.EmpCity = vendors.VendCity AND
employees.EmpState = vendors.VendState;

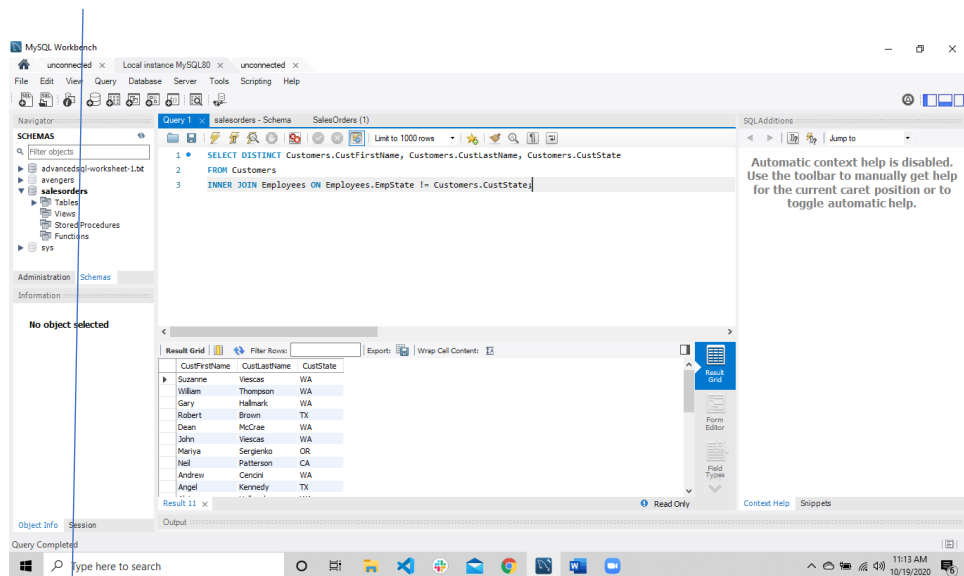
```
- Columns: 7 Columns
- Expected Row Count: 2 Rows
- Screenshot:



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

- Query:  

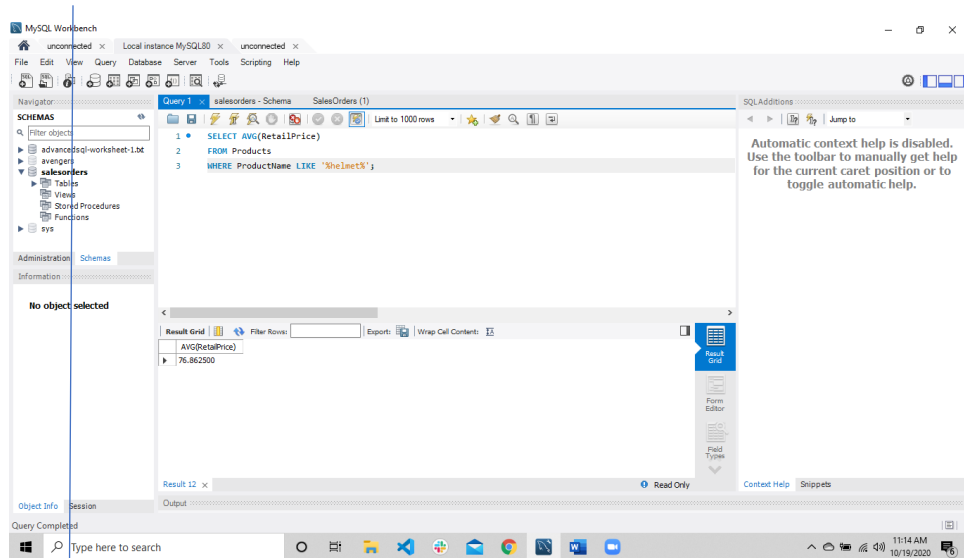
```
SELECT DISTINCT Customers.CustFirstName, Customers.CustLastName,
Customers.CustState
FROM Customers
INNER JOIN Employees ON Employees.EmpState != Customers.CustState;
```
- Columns: 3 Columns
- Expected Row Count: 28 Rows
- Screenshot:



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

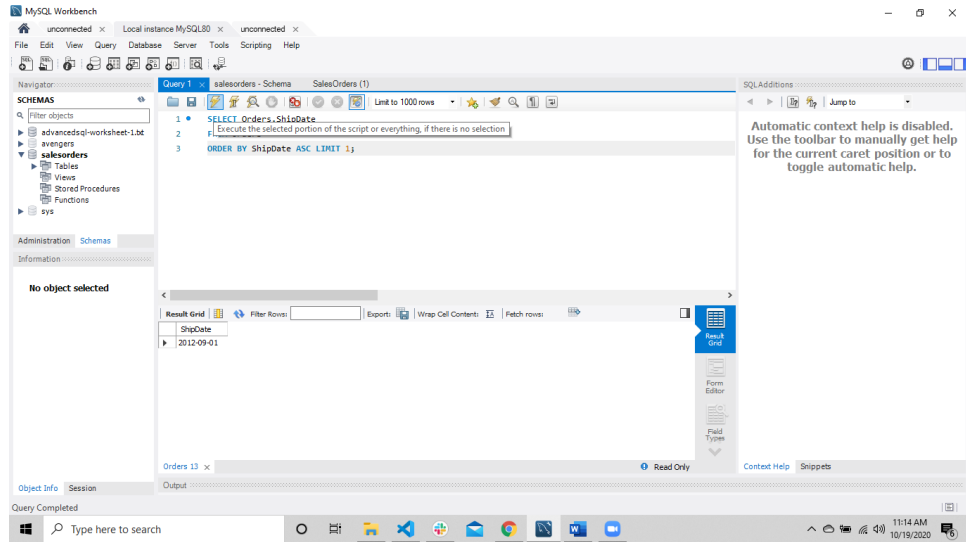


- a) Query:  
`SELECT AVG(RetailPrice)  
FROM Products  
WHERE ProductName LIKE '%helmet%';`
- b) Columns: 1 Columns
- c) Expected Row Count: 1 Row
- d) Screenshot:



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

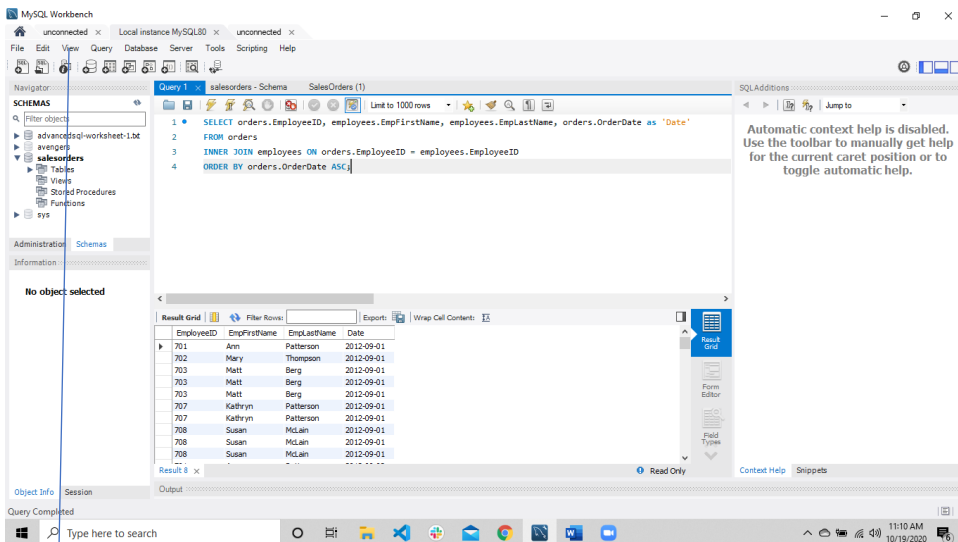
- a) Query:  
`SELECT Orders.ShipDate  
FROM Orders  
ORDER BY ShipDate ASC LIMIT 1;`
- b) Columns: 1 Column
- c) Expected Row Count: 1 Row
- d) Screenshot:



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

- Query:  

```
SELECT customers.CustState, SUM(orders.OrderTotal)
FROM customers
INNER JOIN orders ON customers.CustomerID = orders.CustomerID WHERE
CustState = 'OR';
```
- Columns: 2 Columns
- Expected Row Count: 1 Row
- Screenshot:



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.

a) Query:

```
SELECT employees.EmpFirstName, employees.EmployeeID,  
orders.OrderTotal, orders.EmployeeID, order_details.QuantityOrdered,  
AVG(QuotedPrice)  
FROM employees  
INNER JOIN orders ON employees.EmployeeID = orders.EmployeeID  
INNER JOIN order_details ON orders.OrderNumber =  
order_details.OrderNumber;
```

b) Columns: 6 Columns

c) Expected Row Count: 1000 Rows

d) Screenshot:

