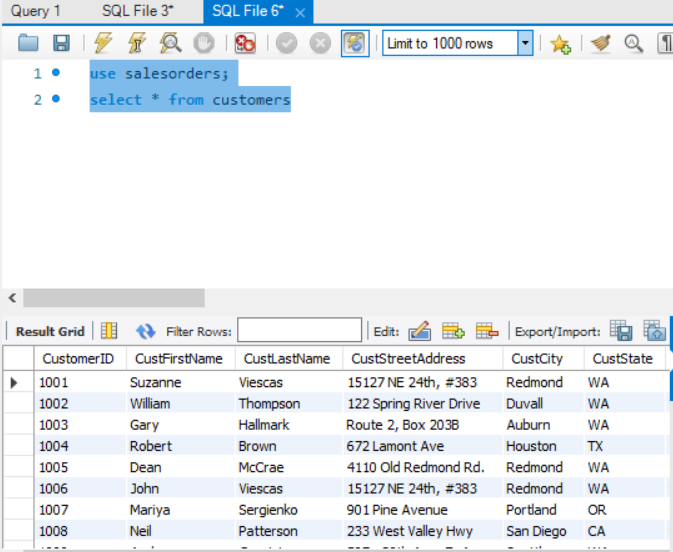
For this assignment, write queries using SQL to acquire data about customers, vendors, products, and employees in a fictitious sales database. These queries will cover many of the core aspects of writing SQL to produce data for reporting and analyzing information. There may be multiple ways to produce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

**use salesorders;  
select \* from customers**

1. Columns: 9
2. Expected Row Count: 22
3. Screenshot:

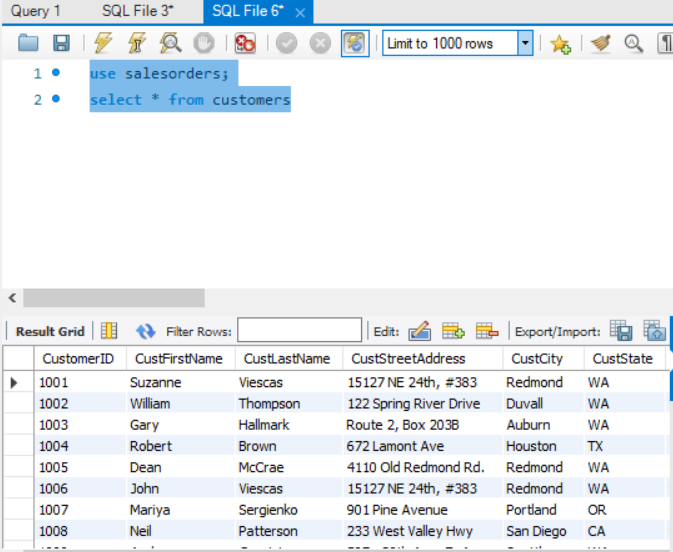


1. **Show a list of states, in reverse alphabetical order, where our vendors are located, and include the names of the vendor.**
2. Query:

**select vendstate, vendname from vendors**

**order by vendstate DESC;**

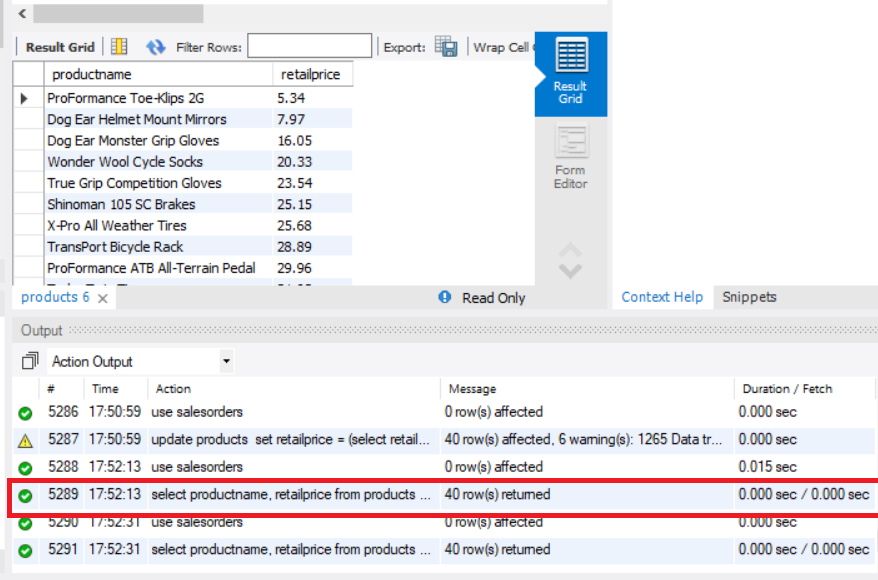
1. Columns: 2
2. Expected Row Count: 11
3. Screenshot:



1. **What if we adjusted the retail price of each product by increasing it 7 percent?**
2. Query:

**use salesorders;  
update products   
set retailprice = (select retailprice + retailprice \* .07)**

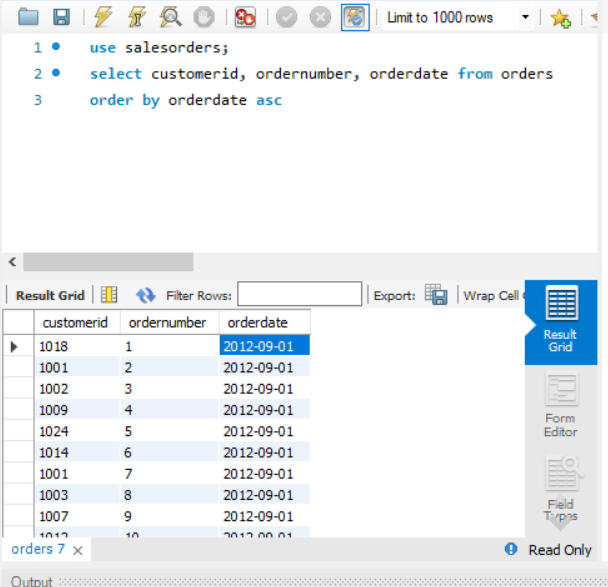
1. Columns: 0
2. Expected Row Count: 40
3. Screenshot:



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

**select customerid, ordernumber, orderdate from orders  
order by orderdate asc;**

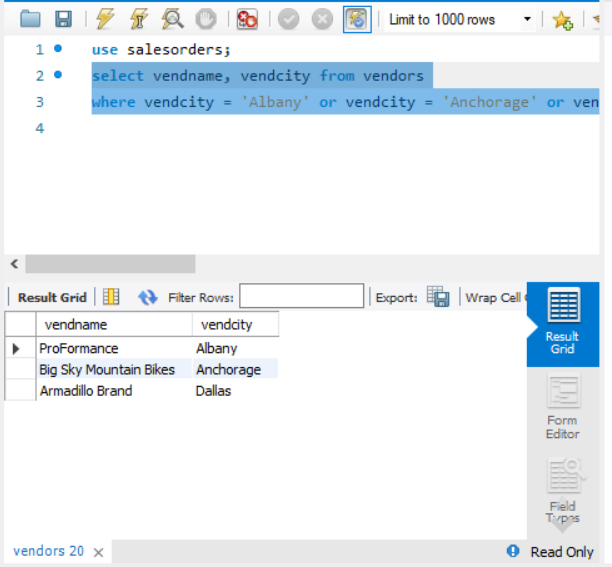
1. Columns: 3
2. Expected Row Count: 944
3. Screenshot:



1. **Give the names of all vendors based in Albany, Anchorage, and Dallas.**
2. Query:

**select vendname, vendcity from vendors  
 where vendcity = 'Albany' or vendcity = 'Anchorage' or vendcity = ‘Dallas'**

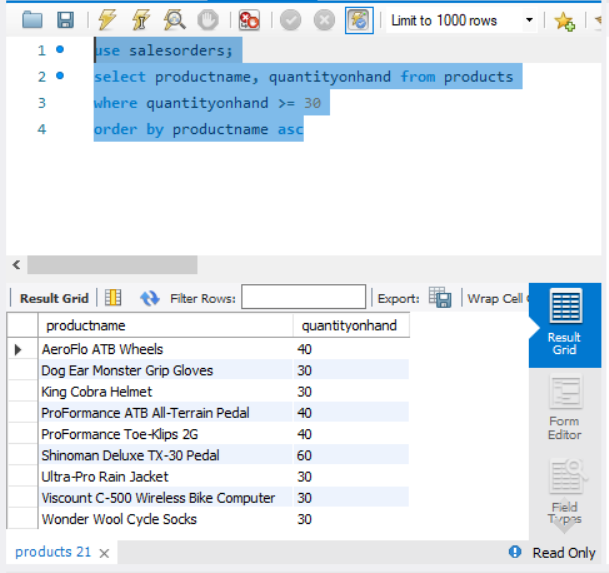
1. Columns: 2
2. Expected Row Count: 3
3. Screenshot:



1. **Show an alphabetized list of products with a quantity on hand greater than or equal to 30.**
2. Query:

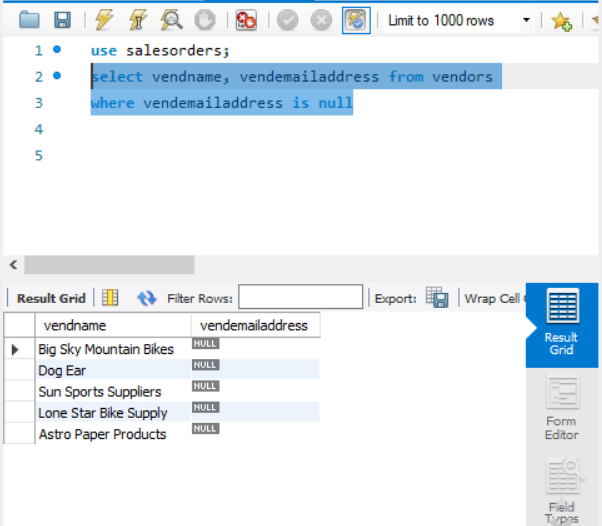
**use salesorders;  
 select productname, quantityonhand from products  
 where quantityonhand >= 30  
 order by productname asc**

1. Columns: 2
2. Expected Row Count: 9
3. Screenshot:



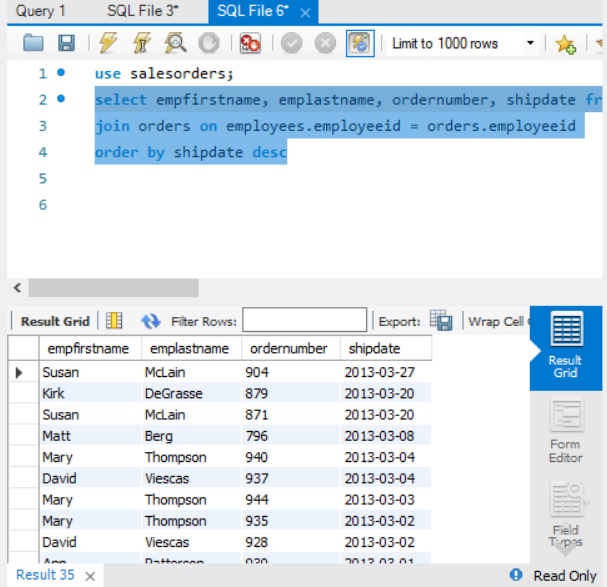
1. **What vendors do we work with that don’t have an email address?**
2. Query:

**select vendname, vendemailaddress from vendors  
 where vendemailaddress is null;**

1. Columns: 2
2. Expected Row Count: 5
3. Screenshot:
4. **List employees and the dates their orders shipped sorted by order date.**
5. Query:

**select empfirstname, emplastname, ordernumber, shipdate from employees   
 join orders on employees.employeeid = orders.employeeid  
 order by shipdate desc;**

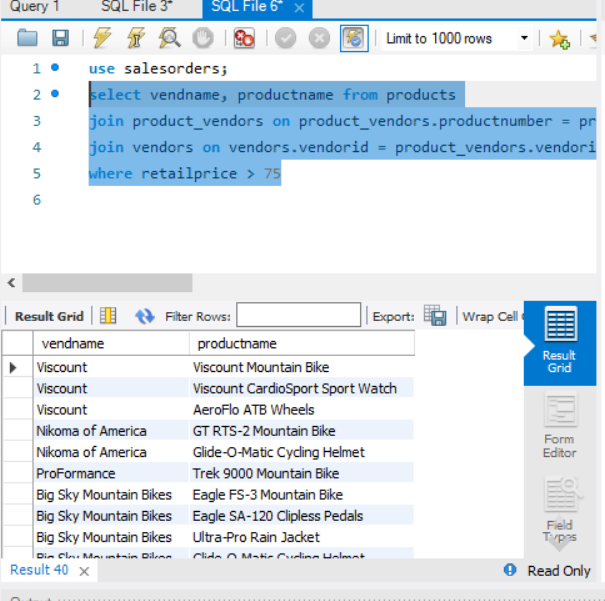
1. Columns: 4
2. Expected Row Count: 944
3. Screenshot:



1. **Show the vendors and products they supply to us for products over $75 for vendors in Texas.**
2. Query:

**select vendname, productname from products  
 join product\_vendors on product\_vendors.productnumber = products.productnumber  
 join vendors on vendors.vendorid = product\_vendors.vendorid  
 where retailprice > 75;**

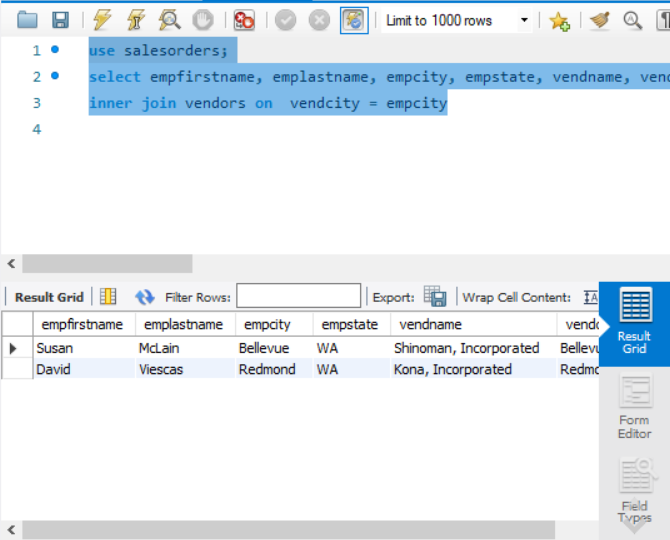
1. Columns: 2
2. Expected Row Count: 32
3. Screenshot:



1. **Show employees who live in the same city and state as our vendors.**
2. Query:

**use salesorders;  
select empfirstname, emplastname, empcity, empstate, vendname, vendcity, vendstate from employees  
inner join vendors on vendcity = empcity**

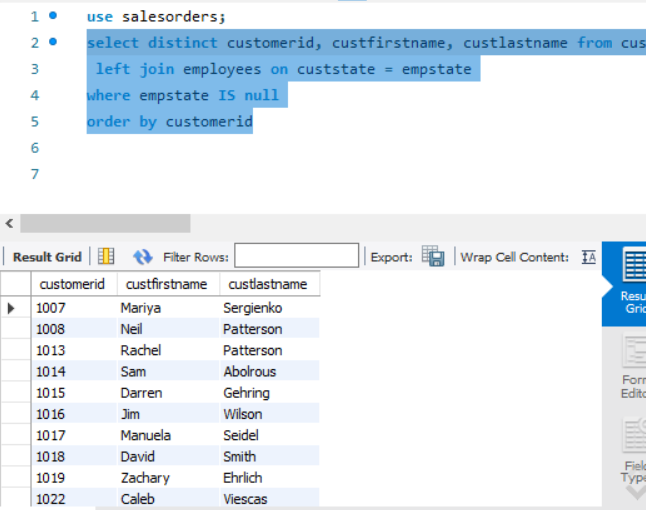
1. Columns: 7
2. Expected Row Count: 2
3. Screenshot:



1. **Display customers who have no sales rep (employees) in the same state.**
2. Query:

**select distinct customerid, custfirstname, custlastname from customers  
 left join employees on custstate = empstate  
 where empstate IS null  
 order by customerid**

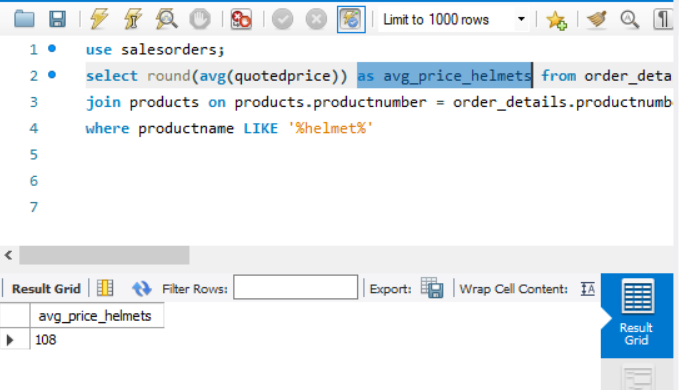
1. Columns: 3
2. Expected Row Count: 11
3. Screenshot:



1. **What is the average quoted price of a helmet?**
2. Query:

**select round(avg(quotedprice))as avg\_price\_helmets from order\_details  
 join products on products.productnumber = order\_details.productnumber  
 where productname LIKE '%helmet%'**

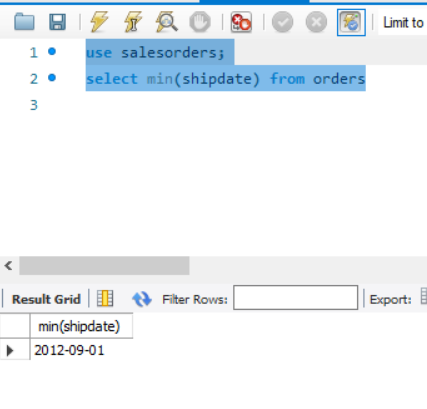
1. Columns: 1
2. Expected Row Count: 1
3. Screenshot:



1. **What was the date of the earliest ship date?**
2. Query:

**use salesorders;  
 select min(shipdate) from orders**

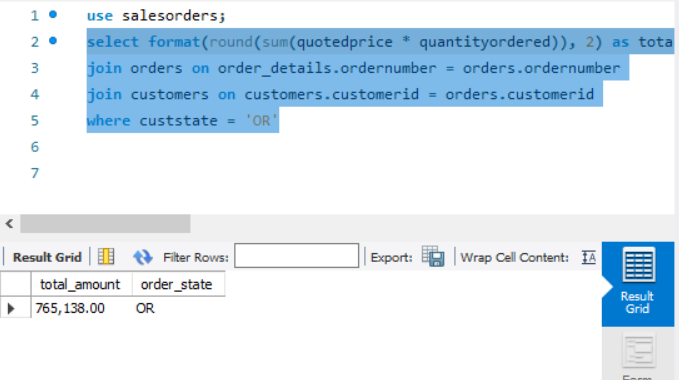
1. Columns: 1
2. Expected Row Count: 1
3. Screenshot:



1. **What is the total amount (in dollars) of orders from the state of Oregon?**
2. Query:

**select format(round(sum(quotedprice \* quantityordered)), 2) as total\_amount, custstate as order\_state from order\_details  
 join orders on order\_details.ordernumber = orders.ordernumber  
 join customers on customers.customerid = orders.customerid  
 where custstate = 'OR'**

1. Columns: 2
2. Expected Row Count: 1
3. Screenshot:



1. **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.**
2. Query:

**select orders.employeeid, sum(quotedprice) as total\_sales\_amount, sum(quantityordered), round(avg(quotedprice)) as avg\_sale\_price 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 asc**

1. Columns: 3
2. Expected Row Count: 8
3. Screenshot:

