1. Write/show two queries: one that shows how many order DATEs FROM the orders table are incorrect (i.e. they are 0000-00-0000 00:00:00) and one that shows how many ship DATEs FROM the orders table are incorrect (i.e. they are 0000-00-0000 00:00:00). For each question, show the order number and the order DATE or ship DATE.

**SELECT orderid AS 'Order #', orderdate AS 'Order DATE'**

**FROM orders WHERE orderdate = '0000-00-00 00:00:00';**

**SELECT orderid AS 'Order #', ordershipdate AS 'Ship DATE'**

**FROM orders WHERE ordershipdate = '0000-00-00 00:00:00';**

1. Assuming that your answer to question #1 above is not zero for either table, write/show two queries; one that will change the value of any order DATE that is currently 0000-00-00 to the current DATE, and one that will change the value of any ship DATE that is currently 0000-00-00 to the current DATE. Do **not** physically enter the current DATE into either query. Rather, use one of the MySQL DATE functions that provide the current DATE.

The results WERE 0, but here are the queries:

**UPDATE orders**

**SET orderdate = current\_timestamp()**

**WHERE orderdate = '0000-00-00 00:00:00';**

**UPDATE orders**

**SET ordershipdate = current\_timestamp()**

**WHERE ordershipdate = '0000-00-00 00:00:00';**

1. Write/show a query that shows the total number of customers per state. Include both the state name and the COUNT in your answer. Order the result by the COUNT in descending order. Write/show a query that shows the total number of employees per state. Include both the state name and the COUNT in your answer. Order the result by the COUNT in descending order.

**SELECT customerstate AS 'State',**

**COUNT(customerstate) AS 'COUNT FROM State'**

**FROM customers**

**GROUP BY customerstate**

**ORDER BY COUNT(customerstate) desc;**

**SELECT employeestate AS 'State',**

**COUNT(employeestate) AS 'COUNT FROM State'**

**FROM employees GROUP BY employeestate**

**ORDER BY COUNT(employeestate) desc;**

1. Write/show a query that shows each category description, along with a COUNT of the number of products in that category description. Sort it by the COUNT of the number of products descending and then category description ascending.

**SELECT c.categorydescription AS 'Description',**

**COUNT(p.productid) AS 'COUNT'**

**FROM products p, categories c**

**WHERE p.categoryid = c.categoryid**

**GROUP BY c.categorydescription**

**ORDER BY COUNT(p.productid) desc, c.categorydescription;**

1. Write/show a query that shows the biggest discrepancy (DIFFerence) between the order DATE and the ship DATE. Write/show another query that shows how many records meet that criteria. Or, write it AS one query with a subquery.

**SELECT MAX(DATEDIFF(ordershipdate, orderdate)) AS 'Biggest DATE Difference'**

**FROM orders;**

**SELECT \* FROM orders WHERE DATEDIFF(ordershipdate, orderdate) = 4;**

1. Write/show a query that shows the smallest discrepancy (DIFFerence) between the order DATE and the ship DATE. Write/show another query that shows how many records meet that criteria. Or, write it AS one query with a subquery.

**SELECT MIN(DATEDIFF(ordershipdate, orderdate)) AS 'Smallest DATE Difference'**

**FROM orders;**

**SELECT \* FROM orders WHERE DATEDIFF(ordershipdate, orderdate) = 0;**

7 & 8. This is such a big query that it is #7 **AND** #8. You must SELECT one field FROM each of the tables AS follows: order DATE (cast AS DATE), ship DATE (cast AS DATE), quantity ordered, wholesale price, retail price, quoted price, category description, and vendor name WHERE the vendor name begins with an ‘A’ and the quoted price is greater than 150.00. Order the results by category description ascending, then quoted price descending. (I got 114 records).

**SELECT DATE(o.orderdate) AS 'ORDDATE',**

**DATE(o.ordershipdate) AS 'SHIPDATE',**

**od.orderdetailquantityordered AS 'QTY',**

**pv.productvendorwholesaleprice AS 'WHOPRICE',**

**p.productprice AS 'RETPRICE',**

**od.orderdetailquotedprice AS 'QUOTPRICE',**

**c.categorydescription AS 'CATDESC',**

**v.vendorname AS 'VENDNAME'**

**FROM orders o, orderdetails od,**

**products p, categories c,**

**productvendors pv,**

**vendors v**

**WHERE o.orderid = od.orderid**

**AND od.productid = p.productid**

**AND p.categoryid = c.categoryid**

**AND p.productid = pv.productid**

**AND pv.vendorid = v.vendorid**

**AND v.vendorname like 'a%'**

**AND od.orderdetailquotedprice > 150**

**ORDER BY c.categorydescription,**

**od.orderdetailquotedprice desc;**

9 & 10.This is such a big query that it is #9 **AND** #10 (6 points). You must SELECT one field FROM each of the tables AS follows: customerid, employeeid, shipDATE (cast AS DATE), wholesaleprice, retailprice, quotedprice, category description, and vendorid WHERE the quotedprice is not equal to the retailprice and the wholesaleprice is between 125.00 and 130.00 and the employeeid is equal to 705. Order the results by categorydescription, then customerid, both in ascending order.

**SELECT cu.customerid AS 'CUSTID',**

**e.employeeid AS 'EMPID',**

**DATE(o.orderdate) AS 'ORDDATE',**

**pv.productvendorwholesaleprice AS 'WHOPRICE',**

**p.productprice AS 'RETPRICE',**

**od.orderdetailquotedprice AS 'QUOTPRICE',**

**ca.categorydescription AS 'CATDESC',**

**v.vendorid AS 'VENDID'**

**FROM customers cu,**

**employees e,**

**orders o,**

**orderdetails od,**

**products p,**

**categories ca,**

**productvendors pv,**

**vendors v**

**WHERE cu.customerid = o.customerid**

**AND o.employeeid = e.employeeid**

**AND o.orderid = od.orderid**

**AND od.productid = p.productid**

**AND p.categoryid = ca.categoryid**

**AND p.productid = pv.productid**

**AND pv.vendorid = v.vendorid**

**AND od.orderdetailquotedprice <> p.productprice**

**AND pv.productvendorwholesaleprice between 125.00 and 130.00**

**AND e.employeeid = 705;**