1. Write/show a query that will tell whether or not any employee has a telephone number. Make sure to include the employee' first name AND last name in the query results. NOTE: No output **is** output.

**SELECT employeefirstname AS "Employee First Name",**

**employeelastname AS "Employee Last Name"**

**FROM employees**

**WHERE employeephonenumber is null**

1. Write/show a query that shows the total number of orders AND the associated employee first name AND employee last name grouped by employee ID AND ordered by total number orders descending.

**SELECT count(o.orderid) AS '# of Orders',**

**e.employeefirstname AS 'Emp First Name',**

**e.employeelastname AS 'Emp Last Name'**

**FROM orders o, employees e**

**WHERE o.employeeid = e.employeeid**

**GROUP BY e.employeeid**

**ORDER BY count(o.orderid) desc;**

1. Write/show a query that shows total quantity on hand for each category. Use aliases for good headings AND order the results by total quantity in descending order.

**SELECT distinct(p.productname) AS 'Product Name',**

**c.categorydescription AS 'Category Desc'**

**FROM products p, categories c**

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

**GROUP BY c.categorydescription**

**ORDER BY c.categorydescription, p.productname;**

1. Write/show a query that shows each separate product name (no repeats), along with the associated category description. Sort it by category description AND then by product name, both in ascending order.

**SELECT distinct(p.productname) AS 'Product Name',**

**c.categorydescription AS 'Category Desc'**

**FROM products p, categories c**

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

**ORDER BY c.categorydescription, p.productname;**

1. Write/show the same query FROM #4 above, but group results by category ID.

**SELECT distinct(p.productname) AS 'Product Name',**

**c.categorydescription AS 'Category Desc'**

**FROM products p, categories c**

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

**GROUP BY c.categorydescription**

**ORDER BY c.categorydescription, p.productname;**

1. The boss just called AND she wants a query that shows customer first name, customer last name, order number (FROM the Orders table), quantity ordered, product name, AND category description.

**SELECT cust.customerfirstname AS 'CFirst',**

**cust.customerlastname AS 'CLast',**

**o.orderID AS 'Ord#',**

**od.orderdetailquantityordered AS 'Qty',**

**p.productname AS 'Prod Name',**

**cat.categorydescription AS 'Category'**

**FROM customers cust,**

**orders o,**

**orderdetails od,**

**products p,**

**categories cat**

**WHERE cust.customerid = o.customerid**

**AND o.orderID = od.orderID**

**AND od.productID = p.productID**

**AND p.categoryid = cat.categoryid;**

1. Write/show the query to show the vendor name, product name, quantity ordered, AND quantity on hand, WHERE quantity ordered > than the quantity on hand.

**SELECT v.vendorname AS 'Vendor Name',**

**p.productname AS 'Prod Name',**

**od. orderdetailquantityordered AS 'Qty Ord',**

**p.productqty AS 'QTY On Hand'**

**FROM vendors v,**

**products p,**

**orderdetails od,**

**productvendors pv**

**WHERE v.vendorid = pv.vendorid**

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

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

**AND od.orderdetailquantityordered > p.productqty;**

1. Now how many records did you get?

**SELECT distinct(v.vendorName) AS 'Vend Name',**

**p.productName AS 'Prod Name',**

**od.orderdetailquantityordered AS 'Qty Ord',**

**p.productqty AS 'QTY On HAND'**

**FROM vendors v,**

**products p,**

**orderdetails od,**

**productvendors pv**

**WHERE v.vendorid = pv.vendorid**

**AND p.productID = pv.productID**

**AND p.productID = od.productID**

**AND od.orderdetailquantityordered > p.productqty**

**6**

1. Write/run a query that gives the customer first name, customer last name, order date, ship date, retail price, AND quoted price for those records WHERE the quoted price != equal to the retail price. Sort by order number in ascending order.

**SELECT c.customerfirstname AS 'CFirst',**

**c.customerlastname AS 'CLast',**

**date(o.orderdate) AS 'Ord Date',**

**date(o.ordershipdate) AS 'Ship Date',**

**p.productprice AS 'Retail Price',**

**od.orderdetailquotedprice AS 'Quoted Price'**

**FROM customers c,**

**orders o,**

**orderdetails od,**

**products p**

**WHERE c.customerid = o.customerid**

**AND o.orderID = od.orderID**

**AND od.productID = p.productID**

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

**ORDER BY o.orderID;**

1. Write/run the same query AS #9 above **except** add a calculated field on the end called Savings, which represents the difference between the retail price AND the quoted price. Sort on the Savings field first in descending order, followed by the customer number in ascending order. Only show the first 100 records in the query.

**SELECT c.customerfirstname AS 'CFirst',**

**c.customerlastname AS 'CLast',**

**o.orderdate AS 'Ord Date',**

**o.ordershipdate AS 'Ship Date',**

**p.productprice AS 'Retail Price',**

**od.orderdetailquotedprice AS 'Quote Price',**

**round((p.productprice - od.orderdetailquotedprice) , 2)**

**AS 'Savings'**

**FROM customers c,**

**orders o,**

**orderdetails od,**

**products p**

**WHERE c.customerid = o.customerid**

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

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

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

**ORDER BY (p.productprice - od.orderdetailquotedprice) desc,**

**o.orderID limit 100;**

1. Write/execute a query that shows only the vendor name AND product wholesale price, but only for those vendor names that either start with an ‘s’ or end with an ‘s’, **AND** have a wholesale price between 10 AND 20. Sort the results based on wholesale price descending then by vendor name ascending.

**SELECT v.vendorname AS 'Vend Name',**

**pv.productvendorwholesaleprice AS 'Wholesale Price'**

**FROM vendors v, productvendors pv**

**WHERE v.vendorid = pv.vendorid**

**AND (v.vendorname like 's%' or v.vendorname like '%s')**

**AND (pv.productvendorwholesaleprice between 10 AND 20)**

**ORDER BY pv.productvendorwholesaleprice desc;**

1. Write/execute a query that shows the customer first name, customer last name, order number, ship date (cast it as, for example (date) o.shipdate), order date (cast it as, for example (date) o.orderdate), product name, AND retail price, but only for those orders that shipped exactly one day after they were ordered. Use the date\_add() function in your answer.

**SELECT c.customerfirstname AS 'First Name',**

**c.customerlastname AS 'Last Name',**

**o.orderID AS 'Order#',**

**date(o.ordershipdate) AS 'Ship Date',**

**date(o.orderdate) AS 'Order Date',**

**p.productname AS 'Prod Name',**

**p.productprice AS 'Retail Price'**

**FROM customers c,**

**orders o,**

**orderdetails od,**

**products p**

**WHERE c.customerid = o.customerid**

**AND o.orderID = od.orderID**

**AND od.productID = p.productID**

**AND o.ordershipdate = date\_add(o.orderdate,INTERVAL 1 DAY);**