1. List the date and the day of the week for the day that is 65 days after today. Today is the day on which the query is run, not the day on which you wrote the query. So do not enter any specific date in the query.

select to\_char(SysDate+65, 'mm/dd/yyyy, Day') from dual;

1. Show a list of employees hired in April of any year. Display ename and hire date in the format as shown below. Use aliases to rename the output columns as shown. Do not use LIKE.

select ename, to\_char(hiredate, 'Day, Mon, dd, yyyy') from emp where to\_char(hiredate, 'Mon') = 'Apr';

1. Find all orders placed in January 2017 or February 2017. Show Ordid, orderdate. (Must use to\_char to do this problem to receive full credit. Do not use LIKE.)

select ordId, orderdate from ord where to\_char(orderdate, 'Mon, yyyy') in ('Jan, 2017', 'Feb, 2017');

1. Show custId, customer name, OrdId, and order date for all orders that were shipped on the same date as orderdate.

select ord.custid, customer.name, ord.ordid, ord.orderdate from ord inner join customer on ord.custid = customer.custid where ord.orderdate = ord.shipdate;

1. Show all orders placed in 2017 that ordered products with the words “GUIDE TO TENNIS” in their descriptions.

select ord.ordid, product.prodid, product.descrip from ord inner join item on ord.ordid = item.ordid inner join product on product.prodid = item.prodid where to\_char(ord.orderdate, 'yyyy') = '2017' and product.descrip like '%GUIDE TO TENNIS%';

1. Find the employee who represented a customer whose street number (part of address) is 574. Show ename. (Correct query should return ALLEN as answer.)

select emp.ename from emp inner join customer on empno = repid where address like '574%';

1. Find the total number of products sold in each year for each product. (Hint: need to use the to\_char function to get Year and put it in Group By)

select to\_char(orderdate, 'yyyy') year, sum(item.qty) Quantity ,prodid from ord inner join item on item.ordid = ord.ordid group by prodid,to\_char(orderdate, 'yyyy');

1. Find the total number of products sold for product 100870, the total number of products sold for product 100860, and the total number of products sold for product 100861.

select product.prodid, sum(item.qty) SoldCount from product inner join item on item.prodid = product.prodid where product.prodid in (100870, 100860, 100861) group by product.prodid;

1. Find the total number of products sold for products 100870, 100860, and 100861 combined.

select sum(item.qty) SoldCount from product inner join item on item.prodid = product.prodid where product.prodid in (100870, 100860, 100861);

1. What is the average sale price (acutalprice in item table) for all products with ‘ACE’ in their descriptions? You should return one number as the answer. (28.37)

select round(avg(item.actualprice), 2) avg\_sale\_price from product inner join item on item.prodid = product.prodid where product.descrip like '%ACE%';

1. Find all products (prodid) that have sold more than 1000 in total quantity, counting only products sold at an actualprice>2.5.

select prodid, sum(Qty) from item where actualprice > 2.5 group by prodid having sum(qty) >=1000;

1. Find customers who have generated more than 10,000 in revenue (that is, total across all orders exceed 10,000). Show custid and this total revenue for each such customer.

select custid, sum(total) from ord group by custid having sum(total) > 10000;

1. Find the total revenue across all orders that were placed in Jan 2017, Feb 2017, or March 2017,. Use Ord table.  (Should get 94140.9)

select sum(total) TotalRevenue from ord where to\_char(orderdate, 'Mon yyyy') in ('Jan 2017', 'Feb 2017', 'Mar 2017');

1. Find the total revenue across all orders that were placed in Jan 2017, Feb 2017, or March 2017, counting only orders with a total greater than 5000. Use Ord table.  (Should get 82570)

select sum(total) TotalRevenue from ord where to\_char(orderdate, 'Mon yyyy') in ('Jan 2017', 'Feb 2017', 'Mar 2017') and total > 5000;

1. Give a list of employees who represent 3 or more customers. Show employee name and number of customer that employee represents. Correct Query should return the following output.

select emp.ename, count(customer.custid) from emp inner join customer on emp.empno = customer.repid group by (emp.ename) having count(customer.custid) >=3;

1. For price table, show ProdId+Startdate combination values that are repeated.

select prodid, startdate, count(\*) from price group by (prodid, startdate) having count(\*) > 1;