**Question-1**

**MIS 531: SQL-2 Part-1 solution**

| SELECT 'Customer' as Type, c.CompanyName as "Company Name",  count(o.OrderID) as "Order Count",  coalesce(to\_char(max(o.OrderDate),'YYYY-MM-DD'), '-') as "Last Order Date" FROM corp.customers c  LEFT OUTER JOIN corp.orders o on (c.CustID = o.CustID)  GROUP BY CompanyName, 'Customer'  HAVING count(o.OrderID) <= 3  UNION  SELECT 'Shipper'as Type, s.Companyname as "Company Name",  count(o2.OrderID) as "Order Count",  coalesce(to\_char(max(o2.OrderDate), 'YYYY-MM-DD'),'-') as "Last Order Date" FROM corp.shippers s  LEFT OUTER JOIN corp.orders o2 on (s.ShipperID = o2.ShipVia)  GROUP BY Companyname, 'Supplier'  HAVING count(o2.OrderID) <= 50  ORDER BY "Last Order Date" desc, type  ; |
| --- |

Note: in GROUP BY the 'Customer' / 'Supplier' piece is optional (works with / without)

**Question-2**

| SELECT  e.employeeid as "Employee ID",  e.titleofcourtesy || ' ' || e.firstname || ' ' || e.lastname as "Employee Name",  e.hiredate as "Start Date",  coalesce(emp.title, 'No manager') as "Manager Title",  (select to\_char(avg(shippeddate-orderdate),'99.99') from corp.orders o2 where o2.employeeid = e.employeeID) as "Avg Time to Ship",  count(distinct o.orderid) as "Number of Orders",  to\_char(coalesce(sum(od.unitprice \* od.quantity \* (1-od.discount)),0), '$900,000.99') as "Total Revenue" FROM corp.employees e  left outer join corp.employees emp on (e.reportsto = emp.employeeid)  left outer join corp.orders o on (e.employeeid = o.employeeid)  join corp.order\_details od on (o.orderid = od.orderid)  GROUP BY e.employeeid, e.titleofcourtesy, e.firstname, e.lastname, e.hiredate, emp.title HAVING count(distinct o.orderid) > 60  ORDER BY e.employeeid  ; |
| --- |

**Question-3**

| WITH pl\_qtr as (  SELECT  CASE WHEN o.orderdate between '1-JAN-2020' and '31-MAR-2020' then 'Q1'  WHEN o.orderdate between '1-APR-2020' and '30-JUN-2020' then 'Q2'  WHEN o.orderdate between '1-JUL-2020' and '30-SEP-2020' then 'Q3'  ELSE 'Q4'  END AS quarters,  pl.pl\_name product\_line,  (od.unitprice \* od.quantity \* (1-od.discount)) AS sales  FROM corp.orders o  join corp.order\_details od on o.orderid = od.orderid  join corp.products p on od.productid = p.productid  join corp.product\_lines pl on p.pl\_id = pl.pl\_id  WHERE o.orderdate BETWEEN '1-JAN-2020' and '31-DEC-2020'  ),  salespiv AS (  SELECT \*  FROM pl\_qtr  PIVOT (  sum(sales) FOR  product\_line IN ('Beverages' "A", 'Condiments' "B", 'Confections' "C", 'Dairy Products' "D", 'Produce' "E")  )  )  SELECT quarters as "Quarters",  to\_char(A,'$99,999.99') as "Sales - Beverages",  to\_char(B,'$99,999.99') as "Sales - Condiments",  to\_char(C,'$99,999.99') as "Sales - Confections",  to\_char(D,'$99,999.99') as "Sales - Dairy Products",  to\_char(E,'$99,999.99') as "Sales - Produce",  to\_char(A + B + C + D+ E, '$999,999.99') as "Total Sales"  FROM salespiv  ORDER BY quarters  ; |
| --- |

The case statement may be simplified to: 'Q'||to\_char(o.orderdate,'Q')

**Question-4**

| WITH kindlow as (  SELECT pl\_id,  count(DISTINCT p.productid) as numlow,  sum(od.unitprice \* od.quantity) as revlow  FROM corp.products p  LEFT OUTER JOIN corp.order\_details od ON p.productid = od.productid  WHERE p.unitprice <= 26  GROUP BY pl\_id  ),  kindhigh as (  SELECT pl\_id,  count(DISTINCT p2.productid) as numhigh,  sum(od2.unitprice \* od2.quantity) as revhigh  FROM corp.products p2  LEFT OUTER JOIN corp.order\_details od2 ON p2.productid = od2.productid  WHERE p2.unitprice > 26  GROUP BY pl\_id  ),  r21 as (  SELECT pl\_id,  sum(od3.unitprice \* od3.quantity) as rev21  FROM corp.products p3  JOIN corp.order\_details od3 ON p3.productid = od3.productid  JOIN corp.orders o3 on o3.orderid = od3.orderid  WHERE extract(year from o3.orderdate) = extract(year from sysdate)  GROUP BY pl\_id  )  SELECT pl\_name as "Line Name",  numlow as "Low Price",  to\_char(revlow , '$999,999.99') as "Low Price Revenue",  numhigh as "High Price",  to\_char(revhigh , '$999,999.99') as "High Price Revenue",  to\_char((revhigh\*100/(revlow+revhigh)), '99.99') ||'%' as "Percent High-price Revenue", to\_char(rev21 , '$999,999.99') as "Revenue YTD"  FROM corp.product\_lines pl  JOIN kindhigh kh on pl.pl\_id = kh.pl\_id  JOIN kindlow kl on pl.pl\_id = kl.pl\_id  JOIN r21 on pl.pl\_id = r21.pl\_id  ; |
| --- |

Q4 Alternative solution

| WITH kindlow as (  SELECT pl\_id,  count(DISTINCT p.productid) as numlow,  sum(od.unitprice \* od.quantity) as revlow  FROM corp.products p  LEFT OUTER JOIN corp.order\_details od ON p.productid = od.productid  WHERE p.unitprice <= 26  GROUP BY pl\_id  ),  kindhigh as (  SELECT pl\_id,  count(DISTINCT p2.productid) as numhigh,  sum(od2.unitprice \* od2.quantity) as revhigh  FROM corp.products p2  LEFT OUTER JOIN corp.order\_details od2 ON p2.productid = od2.productid  WHERE p2.unitprice > 26  GROUP BY pl\_id  )  SELECT pl\_name as "Line Name",  numlow as "Low Price",  to\_char(revlow , '$999,999.99') as "Low Price Revenue",  numhigh as "High Price",  to\_char(revhigh , '$999,999.99') as "High Price Revenue",  to\_char((revhigh\*100/(revlow+revhigh)), '99.99') ||'%' as "Percent High-price Revenue", to\_char(sum(od3.unitprice \* od3.quantity) , '$999,999.99') as "Revenue YTD"  FROM corp.product\_lines pl  JOIN kindhigh kh on pl.pl\_id = kh.pl\_id  JOIN kindlow kl on pl.pl\_id = kl.pl\_id  JOIN corp.products p3 on pl.pl\_id = p3.pl\_id  JOIN corp.order\_details od3 ON p3.productid = od3.productid  JOIN corp.orders o3 on o3.orderid = od3.orderid  WHERE extract(year from o3.orderdate) = extract(year from sysdate)  GROUP BY pl.pl\_id, pl\_name, revhigh, revlow, numlow, numhigh  ; |
| --- |

**Question-5**

WITH sales\_rank\_table as (

SELECT e.employeeid as empid,

sum((od.unitprice \* od.quantity \* (1-od.discount))) as total\_sales\_amount,

RANK() OVER (order by sum((od.unitprice \* od.quantity \* (1-od.discount))) desc Nulls last) as sales\_rank FROM corp.employees e

LEFT OUTER JOIN corp.orders o on e.employeeid = o.employeeid

LEFT OUTER JOIN corp.order\_details od on o.orderid = od.orderid

GROUP BY e.employeeid

),

order\_rank\_table as (

SELECT e2.employeeid as e2\_empid,

count(o2.orderid) as totalord,

rank() over (order by count(o2.orderid) desc) as order\_rank

FROM corp.employees e2

LEFT OUTER JOIN corp.orders o2 on o2.employeeid = e2.employeeid

GROUP BY e2.employeeid

)

SELECT e.employeeid as "Employee ID",

e.lastname || ', ' || e.firstname as "Employee Name",

CASE

WHEN to\_char((months\_between(sysdate, e.hiredate))/12, '99.99') >= 10 then 'Platinum Employee' WHEN to\_char((months\_between(sysdate, e.hiredate))/12, '99.99') >= 5

AND to\_char((months\_between(sysdate, e.hiredate))/12, '99.99') < 10 then 'Gold Employee' WHEN to\_char((months\_between(sysdate, e.hiredate))/12, '99.99') >= 3

AND to\_char((months\_between(sysdate, e.hiredate))/12, '99.99') < 5 then 'Silver Employee' ELSE 'None'

END as "Employee Loyalty",

totalord as "Total Orders",

to\_char(coalesce(st.total\_sales\_amount, 0), '$9,999,990.99') as "Total Sales in Dollars", ot.order\_rank as "Ranking based on Orders",

st.sales\_rank as "Ranking based on Sales"

FROM corp.employees e

JOIN sales\_rank\_table st on st.empid = e.employeeid

JOIN order\_rank\_table ot on ot.e2\_empid = e.employeeid

WHERE (months\_between(sysdate, e.hiredate))/12 < 20

GROUP BY e.employeeid, e.firstname, e.lastname, e.hiredate, st.total\_sales\_amount, st.sales\_rank, ot.order\_rank, ot.totalord

ORDER BY lastname

;

**Question-6**

| --weekly change refers to the difference between last week and the current week.  WITH jan AS (  SELECT (last\_day('31-JAN-2020') + 2 - LEVEL) AS janday  FROM DUAL  CONNECT BY LEVEL <= 39  ),  totalorders AS (  SELECT o.orderdate order\_date,  count(o.orderid) orders  FROM corp.orders o  WHERE orderdate between '25-DEC-2019' and '01-FEB-2020'  GROUP BY o.orderdate  ),  orderdata AS (  SELECT to\_char(janday, 'DD-MON-YYYY') AS "OrderDate",  coalesce(totalorders.orders,0) AS "Orders Today",  coalesce(LAG(totalorders.orders,7) OVER (ORDER by jan.janday),0) AS "Last Week", coalesce(LAG(totalorders.orders,1) OVER (ORDER by jan.janday),0) AS "Previous Day", coalesce(LEAD(totalorders.orders,1) OVER (ORDER by jan.janday),0) AS "Next Day", coalesce(totalorders.orders,0) -  coalesce(LAG(totalorders.orders,7) OVER (ORDER by jan.janday),0) as "Weekly Change", coalesce(totalorders.orders,0) -  coalesce(LAG(totalorders.orders,1) OVER (ORDER by jan.janday),0) AS "Daily Change" FROM jan  LEFT OUTER JOIN totalorders on jan.janday=totalorders.order\_date  )  SELECT \*  FROM orderdata  WHERE to\_date("OrderDate",'DD-MON-YYYY') between '01-JAN-2020' and '31-JAN-2020'  UNION  SELECT 'Total',  SUM("Orders Today"), null, null, null, null, null  FROM orderdata  WHERE to\_date("OrderDate",'DD-MON-YYYY') between '01-JAN-2020' and '31-JAN-2020'  ORDER BY 1  ; |
| --- |

The connect by level <=39 allows us to get data from Dec-25 to Feb-1.

We need the extra days to get the previous day / week values for Jan-1 (and next day for Jan-31).

Alternative

| --weekly change refers to the difference between last week and the current week.  WITH jan(janday) AS (  SELECT to\_date('25-DEC-2019','DD-MON-YYYY') AS janday  FROM DUAL  UNION ALL  SELECT janday + 1  FROM jan  WHERE janday <= to\_date('01-FEB-2020')  ),  totalorders AS (  SELECT o.orderdate order\_date,  count(o.orderid) orders  FROM corp.orders o  WHERE orderdate between '25-DEC-2019' and '01-FEB-2020'  GROUP BY o.orderdate  ),  orderdata AS (  SELECT to\_char(janday, 'DD-MON-YYYY') AS "OrderDate",  coalesce(totalorders.orders,0) AS "Orders Today",  coalesce(LAG(totalorders.orders,7) OVER (ORDER by jan.janday),0) AS "Last Week", coalesce(LAG(totalorders.orders,1) OVER (ORDER by jan.janday),0) AS "Previous Day", coalesce(LEAD(totalorders.orders,1) OVER (ORDER by jan.janday),0) AS "Next Day", coalesce(totalorders.orders,0) -  coalesce(LAG(totalorders.orders,7) OVER (ORDER by jan.janday),0) as "Weekly Change", coalesce(totalorders.orders,0) -  coalesce(LAG(totalorders.orders,1) OVER (ORDER by jan.janday),0) AS "Daily Change" FROM jan  LEFT OUTER JOIN totalorders on jan.janday=totalorders.order\_date  )  SELECT \*  FROM orderdata  WHERE to\_date("OrderDate",'DD-MON-YYYY') between '01-JAN-2020' and '31-JAN-2020'  UNION  SELECT 'Total',  SUM("Orders Today"), null, null, null, null, null  FROM orderdata  WHERE to\_date("OrderDate",'DD-MON-YYYY') between '01-JAN-2020' and '31-JAN-2020'  ORDER BY 1  ; |
| --- |