< SQL Queries >

1. Basic Queries (SELECT, ORDER BY, JOIN)

Create a query that shows all employees and all order ids with the order date. Sort by Order Date from most recent to the oldest.

```
SELECT e.firstname, e.lastname, o.orderid, o.orderdate
FROM HR.Employees AS e JOIN Sales.Orders AS o
ON e.empid = o.empid
ORDER BY o.orderdate DESC;
```

Modify the query produced in exercise 2 to include the Manager's First and Last Name. Make sure all employees are selected, even the ones who don't have a manager.

```
SELECT e.empid, e.firstname, e.lastname, m.firstname AS mgrfirstname, m.lastname AS
mgrlastname
FROM HR.Employees AS e LEFT OUTER JOIN HR.Employees AS m
ON e.mgrid = m.empid;
```

Create a query that would list all employees' names with all customers that they have served. The result should only contain two columns: employee full name and customer full name. Eliminate duplicate rows.

```
SELECT DISTINCT e.firstname + N' ' + e.lastname AS e.fullname, c.companyname

FROM HR.Employees AS e JOIN Sales.Orders AS o

On o.empid = e.empid

JOIN Sales.Customers AS c

On c.custid = o.custid;
```

- Write a query that would show the employee id, employee full name (in one field), and the employee manager's full name (in one field). Do not show employees who do not have a manger.

```
SELECT DISTINCT e.empid, e.firstname + N'' + e.lastname AS empfullname, m.firstname + N'' +
m.lastname AS mgrfullname

FROM HR.Employees AS e JOIN HR.Employees AS m ON e.mgrid = m.empid;
```

2. Queries with more conditions

- Create a query that would show how many products are being produced in each country. Apply a limit to only show countries that produce 5 products or more. Show the countries that produce the most on top.

```
SELECT s.country, COUNT(*) AS cnt FROM Production.Products AS p

JOIN Production.Suppliers AS s ON s.supplierid = p.supplierid

GROUP BY s.country Having COUNT(*) >= 5

ORDER BY cnt DESC;
```

- Create a product summary query that will show the product name, the minimum, maximum and average order quantity received for each year the product was sold.

```
SELECT YEAR(o.orderdate), p.productname, Min(od.qty) AS Min, Max(od.qty) AS Max,
AVG(od.qty) AS Avg

FROM Sales.Orderdetails AS od

JOIN Production.Products AS p ON p.productid = od.productid

JOIN Sales.Orders AS o ON o.orderid = od.orderid

GROUP BY YEAR(o.orderdate), p.productname

ORDER BY YEAR(o.orderdate);
```

3. Queries with more complicated structure

- Write a query that pulls unique countries from the suppliers table. Use this query as a derived table (in other words, a sub query in the FROM clause) to produce a list of countries with the continent they below to. Use CASE statement in your query. The result should only contain two columns: Country and Continent. Place the continents and countries in alphabetical order.

```
SELECT s.country,
      (CASE
          WHEN s.country = 'Australia' THEN 'Oceania'
          WHEN s.country = 'Brazil' THEN 'South America'
          WHEN s.country = 'Canada' THEN 'North America'
          WHEN s.country = 'Denmark' THEN 'Europe'
          WHEN s.country = 'Finland' THEN 'Europe'
          WHEN s.country = 'France' THEN 'Europe'
          WHEN s.country = 'Germany' THEN 'Europe'
```

```
            WHEN s.country = 'Italy' THEN 'Europe'
            WHEN s.country = 'Japan' THEN 'Asia'
            WHEN s.country = 'Netherlands' THEN 'Europe'
            WHEN s.country = 'Norway' THEN 'Europe'
            WHEN s.country = 'Spain' THEN 'Europe'
            WHEN s.country = 'Singapore' THEN 'Asia'
            WHEN s.country = 'Sweden' THEN 'Europe'
            WHEN s.country = 'UK' THEN 'Europe'
            WHEN s.country = 'USA' THEN 'North America'
            ELSE 'Unknown Continent'
        END
        ) Continent
        FROM (SELECT DISTINCT country FROM Production.Suppliers) AS s
        GROUP BY s.country;
```

- Create a query that would combine the queries in 1st and 3rd questions, do not apply the limit in query 1. The result should include the continent, the count of countries, total count of products per continent, and the average product price.

```
SELECT
      (CASE
            WHEN s.country = 'Australia' THEN 'Oceania'
            WHEN s.country = 'Brazil' THEN 'South America'
            WHEN s.country = 'Canada' THEN 'North America'
            WHEN s.country = 'Denmark' THEN 'Europe'
            WHEN s.country = 'Finland' THEN 'Europe'
            WHEN s.country = 'France' THEN 'Europe'
            WHEN s.country = 'Germany' THEN 'Europe'
            WHEN s.country = 'Italy' THEN 'Europe'
            WHEN s.country = 'Japan' THEN 'Asia'
            WHEN s.country = 'Netherlands' THEN 'Europe'
            WHEN s.country = 'Norway' THEN 'Europe'
            WHEN s.country = 'Spain' THEN 'Europe'
            WHEN s.country = 'Singapore' THEN 'Asia'
            WHEN s.country = 'Sweden' THEN 'Europe'
            WHEN s.country = 'UK' THEN 'Europe'
            WHEN s.country = 'USA' THEN 'North America'
            ELSE 'Unknown Continent'
        END
        ) Continent,
        COUNT(s.country) AS NumCountries,
        SUM (AmountOfProducts) AS TotalProducts,
        AVG(AvgPrice) AS AvgPrice

FROM (SELECT DISTINCT country FROM Production.Suppliers) AS s
    LEFT JOIN
     (  SELECT s.country Country, COUNT(p.supplierid) AmountOfProducts,
AVG(p.unitPrice) AvgPrice
        FROM Production.Products p
```

```
                JOIN Production.Suppliers s ON p.supplierid =
s.supplierid
        GROUP BY s.country
      ) ds ON s.country = ds.Country
GROUP BY
    CASE
      WHEN s.country = 'Australia' THEN 'Oceania'
      WHEN s.country = 'Brazil' THEN 'South America'
      WHEN s.country = 'Canada' THEN 'North America'
      WHEN s.country = 'Denmark' THEN 'Europe'
      WHEN s.country = 'Finland' THEN 'Europe'
      WHEN s.country = 'France' THEN 'Europe'
      WHEN s.country = 'Germany' THEN 'Europe'
      WHEN s.country = 'Italy' THEN 'Europe'
      WHEN s.country = 'Japan' THEN 'Asia'
      WHEN s.country = 'Netherlands' THEN 'Europe'
      WHEN s.country = 'Norway' THEN 'Europe'
      WHEN s.country = 'Spain' THEN 'Europe'
      WHEN s.country = 'Singapore' THEN 'Asia'
      WHEN s.country = 'Sweden' THEN 'Europe'
      WHEN s.country = 'UK' THEN 'Europe'
      WHEN s.country = 'USA' THEN 'North America'
      ELSE 'Unknown Continent'
      END
```

- Create a query that would show a **sum**mary of order value (unit price * quantity) for each mana ger (should include sales for all employees reporting to this manager) per order year and order month. Display the year and the month in one column (i.e. '2017-07'). Limit the result to only sh ow first 6 months of year 2007.)

```
SELECT t.mgrfullname, t.orderyear, t.ordermonth, SUM(OrderValue) AS
Sales

FROM(

SELECT DISTINCT e.empid, e.firstname + N'' + e.lastname AS empfullname,
m.firstname + N'' + m.lastname AS mgrfullname, od.unitprice*od.qty AS
OrderValue,

Year(o.orderdate) AS orderyear, Month(o.orderdate) AS ordermonth

FROM HR.Employees AS e JOIN HR.Employees AS m ON e.mgrid = m.empid

JOIN Sales.Orders AS o ON o.empid = e.empid

JOIN Sales.Orderdetails AS od ON od.orderid = o.orderid) AS t

WHERE o.orderdate BETWEEN '20061231' AND '20070701'

GROUP BY t.mgrfullname, t.orderyear, t.ordermonth;
```

- Write a query that would return the following information: company name, contact name, city, customer country, order id, order date, product name, supplier country (name this field: Origin), unit price, qty, Discounted Order Line Price = unit price * order qty * (1-discount %), Total Discount = figure out the formula

  N.B. This query will be used throughout the assignment as derived table, make sure you name the columns right

```
SELECT c.companyname, c.city, c.country as CustCountry, o.orderid,
o.orderdate, p.productname, s.country as Origin, od.unitprice, od.qty,
od.unitprice*od.qty*(1-discount) AS DiscountedOrderLPrice,
od.unitprice*od.qty*discount AS TotalDiscount
FROM Sales.Customers AS c JOIN Sales.Orders AS o On o.custid = c.custid
     JOIN Sales.Orderdetails AS od On od.orderid = o.orderid
     JOIN Production.Products AS p On p.productid = od.productid
     JOIN Production.Suppliers AS s On s.supplierid = p.supplierid;
```

- Situation: every time you ship a product from one country to another, you have to pay customs clearance fee. Assuming you're shipping from supplier to customer direct and the clearance fee being of 5% of discounted price, create a query that will show inter-country discounted amounts, and custom fee. Make sure to exclude the shipments within the same country. The result should contain the following columns: country of origin, destination country, total discounted amount, customs clearance amount.

```
SELECT t.custid, t.CustCountry, t.Origin, SUM(TotalDiscount) AS
TotalDiscountedAmount, SUM(DiscountedOrderPrice*0.05) AS CustomFee

FROM(
SELECT c.custid, c.companyname, c.city, c.country as CustCountry, o.orderid,
o.orderdate, p.productname, s.country as Origin, od.unitprice, od.qty,
od.unitprice*od.qty*(1-discount) AS DiscountedOrderLPrice,
od.unitprice*od.qty*discount AS TotalDiscount
FROM Sales.Customers AS c JOIN Sales.Orders AS o On o.custid = c.custid
     JOIN Sales.Orderdetails AS od On od.orderid = o.orderid
     JOIN Production.Products AS p On p.productid = od.productid
     JOIN Production.Suppliers AS s On s.supplierid = p.supplierid
) AS t
```

```
WHERE c.CustCountry <> t.Origin

GROUP BY t.custid, t.CustCountry, t.Origin;
```

- Create an order summary per customer, sowing the total saving the customer accumulated per month. The query should return the company name, order year, order month (spelled out), Total Savings

```
SELECT t.custid, t.companyname, Year(t.orderdate) AS orderyear,
Month(t.orderdate) AS ordermonth,

SUM(t.TotalDiscount) AS TotalSavings


FROM(

SELECT c.custid, c.companyname, c.city, c.country as CustCountry, o.orderid,
o.orderdate, p.productname, s.country as Origin, od.unitprice, od.qty,

od.unitprice*od.qty*(1-discount) AS DiscountedOrderLPrice,

od.unitprice*od.qty*discount AS TotalDiscount

FROM Sales.Customers AS c JOIN Sales.Orders AS o On o.custid = c.custid

     JOIN Sales.Orderdetails AS od On od.orderid = o.orderid

     JOIN Production.Products AS p On p.productid = od.productid

     JOIN Production.Suppliers AS s On s.supplierid = p.supplierid

) AS t

GROUP BY t.custid, t.companyname, Year(t.orderdate), Month(t.orderdate);
```

4. Other various queries and functions(view, pivot table)

- Create a view using the results of query 3. Call the view: vw_MonthlyCustomerSavings. Make sure to verify if the view exists before you create it, and drop it if so. Submit both the drop and create view statements.

```
Create view vw_MonthlyCustoSavings

AS

SELECT t.custid, t.companyname, t.origin, Year(t.orderdate) AS orderyear,
Month(t.orderdate) AS ordermonth,
```

```
SUM(t.TotalDiscount) AS TotalSavings, SUM(DiscountedOrderLPrice) AS
DiscountedAmount


FROM(

SELECT c.custid, c.companyname, c.city, c.country as CustCountry, o.orderid,
o.orderdate, p.productname, s.country as origin, od.unitprice, od.qty,

od.unitprice*od.qty*(1-discount) AS DiscountedOrderLPrice,

od.unitprice*od.qty*discount AS TotalDiscount

FROM Sales.Customers AS c JOIN Sales.Orders AS o On o.custid = c.custid

     JOIN Sales.Orderdetails AS od On od.orderid = o.orderid

     JOIN Production.Products AS p On p.productid = od.productid

     JOIN Production.Suppliers AS s On s.supplierid = p.supplierid


) AS t

GROUP BY t.custid, t.companyname, t.origin, Year(t.orderdate),
Month(t.orderdate);

GO


SELECT companyname, origin, orderyear FROM vw_MonthlyCustoSavings;
```

- Modify query # 3 to accept parameter customer_id. Submit both, parameter definition and query using this parameter. Assign any customer to the parameter.

```
Create Procedure Sales.MonthlyCustomers

(@custid AS INT)

AS

Select companyname, orderyear, TotalSavings

FROM vw_MonthlyCustomerSavings

WHERE custid = @custid

ORDER BY orderyear;

GO


EXEC Sales.MonthlyCustomers @custid = 5;
```

- Using results of query 1, create a query that would display country of origin and total discounted amount per year. The years should run across and have a total column and a grand total row. Use either case statement or PIVOT function.

  I strongly suggest you try to write this query, as similar query will be on the exam!

  The result should look as follows(colors applied in excel for clarity purpose only):

  In part; using the view of #4,

```
SELECT origin, [2006], [2007], [2008]
FROM (SELECT origin, orderyear, DiscountedAmount FROM vw_MonthlyCustoSavings)
AS d
PIVOT (SUM(DiscountedAmount) For orderyear IN ([2006], [2007], [2008])) AS pvt
ORDER BY origin;
```

Origin	2006	2007	2008	Total
Finland	4336.7400000	12681.1375000	11424.8500000	28442.7275000
USA	20223.3650000	56929.8725000	44950.2275000	122103.4650000
Italy	16920.6500000	56333.0500000	25226.0750000	98479.7750000
Brazil	556.7400000	1630.1250000	2317.5000000	4504.3650000
Netherlands	97.2800000	4212.7625000	1016.7500000	5326.7925000
Germany	29340.8350000	95152.0660000	72957.1155000	197450.0165000
Australia	31481.7750000	91942.5575000	48662.2130000	172086.5455000
Sweden	4381.5200000	17563.1500000	9923.4500000	31868.1200000
UK	14019.9100000	33381.1000000	31031.0300000	78432.0400000
Canada	14421.8700000	41746.9650000	27573.3725000	83742.2075000
Norway	8873.2600000	22287.3500000	11980.9000000	43141.5100000
France	47871.0000000	112030.7600000	117652.3700000	277554.1300000
Japan	5079.5400000	26393.7525000	13789.8025000	45263.0950000
Spain	1952.6400000	15247.2400000	7959.5500000	25159.4300000
Denmark	1175.5200000	6337.2750000	2708.3800000	10221.1750000
Singapore	7351.3250000	23216.0400000	11450.2800000	42017.6450000
Grand Total	208083.9700000	617085.2035000	440623.8660000	1265793.0395000