



INSTITUTO POLITÉCNICO  
NACIONAL

UNIDAD PROFESIONAL  
INTERDISCIPLINARIA EN  
INGENIERÍA Y TECNOLOGÍAS



# **Bases de Datos Distribuidas**

## **Práctica 1:**

### **“REPASO DE CONSULTAS SQL Y USO DE SERVIDORES VINCULADOS ”**

Integrantes:

De la Rosa Valdes Itztli Fernanda

Luviano Moreno Alondra

Boleta:

2024640062

2023640692

Ejercicio 1. Encuentra los 10 productos más vendidos en 2014, mostrando nombre del producto, cantidad total vendida y nombre del cliente.

```
SELECT TOP 10
    P.Name AS [Nombre del Producto],
    SUM(SOD.OrderQty) AS [Cantidad Total Vendida],
    PER.FirstName + ' ' + PER.LastName AS [Nombre del Cliente]
FROM Production.Product P
JOIN Sales.SalesOrderDetail SOD ON P.ProductID = SOD.ProductID
JOIN Sales.SalesOrderHeader SOH ON SOD.SalesOrderID = SOH.SalesOrderID
JOIN Sales.Customer C ON SOH.CustomerID = C.CustomerID
JOIN Person.Person PER ON C.PersonID = PER.BusinessEntityID
WHERE YEAR(SOH.OrderDate) = 2014
GROUP BY P.Name, PER.FirstName, PER.LastName
ORDER BY [Cantidad Total Vendida] DESC;
```

	Nombre del Producto	Cantidad Total Vendida	Nombre del Cliente
1	Road-350-W Yellow, 48	42	Kevin Liu
2	Women's Mountain Shorts, L	41	Richard Bready
3	Classic Vest, S	39	Pilar Ackerman
4	Women's Mountain Shorts, S	38	Richard Bready
5	Women's Mountain Shorts, L	38	James Haugh
6	Women's Mountain Shorts, S	35	Kathleen Garza
7	Classic Vest, S	35	Jon Grande
8	Women's Mountain Shorts, L	32	François Ferrier
9	Classic Vest, S	32	Alexander Berger
10	Short-Sleeve Classic Jersey, XL	32	Min Su

1. Una vez resuelta la consulta: agrega el precio unitario promedio (AVG(UnitPrice)) y filtra solo productos con ListPrice > 1000.

```
SELECT TOP 10
    P.Name AS [Nombre del Producto],
    SUM(SOD.OrderQty) AS [Cantidad Total Vendida],
    AVG(SOD.UnitPrice) AS [Precio Unitario Promedio], -- AVG(UnitPrice)
    PER.FirstName + ' ' + PER.LastName AS [Nombre del Cliente]
FROM Production.Product P
JOIN Sales.SalesOrderDetail SOD ON P.ProductID = SOD.ProductID
JOIN Sales.SalesOrderHeader SOH ON SOD.SalesOrderID = SOH.SalesOrderID
JOIN Sales.Customer C ON SOH.CustomerID = C.CustomerID
JOIN Person.Person PER ON C.PersonID = PER.BusinessEntityID
WHERE YEAR(SOH.OrderDate) = 2014
    AND P.ListPrice > 1000 --ListPrice > 1000
GROUP BY P.Name, PER.FirstName, PER.LastName
ORDER BY [Cantidad Total Vendida] DESC;
```

	Nombre del Producto	Cantidad Total Vendida	Precio Unitario Promedio	Nombre del Cliente
1	Road-350-W Yellow, 48	42	893.0197	Kevin Liu
2	Road-350-W Yellow, 48	24	986.5742	Kirk DeGrasse
3	Road-350-W Yellow, 42	22	1003.5841	Kevin Liu
4	Mountain-200 Black, 38	21	1354.0441	Kathleen Garza
5	Road-350-W Yellow, 40	19	935.5445	Robin McGuigan
6	Road-350-W Yellow, 48	19	1020.594	Jon Grande
7	Road-350-W Yellow, 48	17	1003.5841	Nate Sun
8	Touring-1000 Yellow, 60	17	1406.6013	Megan Davis
9	Touring-1000 Yellow, 60	17	1406.6013	Terry Eminhizer
10	Mountain-200 Black, 38	16	1262.2445	Holly Dickson

Se utilizaron 5 tablas diferentes (Product, SalesOrderDetail, SalesOrderHeader, Customer y Person) para vincular la información del producto con los datos y la identidad del cliente.

Se empleó SUM para contabilizar las cantidades vendidas y AVG para determinar el valor promedio de venta por unidad.

El uso de YEAR(OrderDate) permitió aislar exclusivamente las operaciones del año 2014, mientras que el filtro en ListPrice se enfocó en el precio de los productos como lo requirió la segunda consulta.

Ejercicio 2: Lista los empleados que han vendido más que el promedio de ventas por empleado en el territorio 'Northwest'.

```

SELECT
    BusinessEntityID,
    SalesYTD
FROM Sales.SalesPerson
WHERE TerritoryID = (SELECT TerritoryID FROM Sales.SalesTerritory WHERE Name = 'Northwest')
AND SalesYTD > (
    SELECT AVG(SalesYTD)
    FROM Sales.SalesPerson
    WHERE TerritoryID = (SELECT TerritoryID FROM Sales.SalesTerritory WHERE Name = 'Northwest')
);

```

1. Requisito adicional: aplicar subconsultas.
2. Una vez resuelta la consulta convierte la subconsulta en un CTE (Common Table Expresión).
3. Documenta la solución inicial y solución con la variante solicitada.

```

WITH PromedioNorthwest AS (
    -- (CTE)
    SELECT AVG(SalesYTD) AS PromedioVentas
    FROM Sales.SalesPerson
    WHERE TerritoryID = (SELECT TerritoryID FROM Sales.SalesTerritory WHERE Name = 'Northwest')
)
SELECT
    sp.BusinessEntityID,
    sp.SalesYTD
FROM Sales.SalesPerson sp
CROSS JOIN PromedioNorthwest pnw
WHERE sp.TerritoryID = (SELECT TerritoryID FROM Sales.SalesTerritory WHERE Name = 'Northwest')
AND sp.SalesYTD > pnw.PromedioVentas;

```

	BusinessEntityID	SalesYTD
1	283	1573012.9383
2	284	1576562.1966

En la solución inicial, se hizo uso de una consulta con WHERE para determinar la media de la columna SalesYTD. Para la variante solicitada, se implementó una CTE (Common Table Expression), la cual define un conjunto de resultados temporal y nombrado que mejora la modularidad del código. De igual manera haciendo uso de JOIN y CROSS JOIN para vincular los datos de las diferentes tablas usadas para las consultas.

Ejercicio 3: Calcula ventas totales por territorio y año, mostrando solo aquellos con más de 5 órdenes y ventas > \$1,000,000, ordenado por ventas descendente.

```

SELECT
    ST.Name AS Territorio,
    YEAR(SOH.OrderDate) AS Anio,
    SUM(SOH.TotalDue) AS VentasTotales,
    COUNT(SOH.SalesOrderID) AS NumeroOrdenes
FROM Sales.SalesOrderHeader SOH
JOIN Sales.SalesTerritory ST ON SOH.TerritoryID = ST.TerritoryID
GROUP BY ST.Name, YEAR(SOH.OrderDate)
HAVING COUNT(SOH.SalesOrderID) > 5
AND SUM(SOH.TotalDue) > 1000000
ORDER BY VentasTotales DESC;

```

	Territorio	Anio	VentasTotales	NumeroOrdenes
1	Southwest	2013	10239209.3403	2725
2	Southwest	2012	9329154.3425	777
3	Canada	2013	7010449.6994	1884
4	Northwest	2013	6759500.6713	2053
5	Canada	2012	6599971.0217	460
6	Northwest	2012	5325813.0562	510
7	Australia	2013	4702404.0504	3015
8	Southwest	2014	4437517.8076	2383
9	France	2013	4271019.2663	1273
10	United Kingdom	2013	4068178.6672	1528

1. Una vez resuelta la consulta agrega desviación estándar de ventas

```

SELECT
    ST.Name AS Territorio,
    YEAR(SOH.OrderDate) AS Anio,
    SUM(SOH.TotalDue) AS VentasTotales,
    COUNT(SOH.SalesOrderID) AS NumeroOrdenes,
    STDEV(SOH.TotalDue) AS DesviacionVentas --Desviación
FROM Sales.SalesOrderHeader SOH
JOIN Sales.SalesTerritory ST ON SOH.TerritoryID = ST.TerritoryID
GROUP BY ST.Name, YEAR(SOH.OrderDate)
HAVING COUNT(SOH.SalesOrderID) > 5
      AND SUM(SOH.TotalDue) > 1000000
ORDER BY VentasTotales DESC;

```

2. Documenta la solución inicial y solución con la variante solicitada.

En este ejercicio hicimos uso de GROUP BY y HAVING. La lógica implementada agrupa las transacciones por YEAR y Territory, aplicando funciones de agregación para visualizar las ventas y contabilizar el número de órdenes. HAVING nos permite imponer restricciones sobre los resultados de las funciones agregadas, filtrando grupos que no alcanzan los \$1,000,000 o el mínimo de actividad. La inclusión de la desviación estándar (STDEV) aporta un análisis de dispersión sobre el TotalDue.

Ejercicio 4: Encuentra vendedores que han vendido TODOS los productos de la categoría "Bikes".

```
SELECT
    SOH.SalesPersonID,
    COUNT(DISTINCT P.ProductID) AS ProductosDistintosVendidos
FROM Sales.SalesOrderHeader SOH
JOIN Sales.SalesOrderDetail SOD ON SOH.SalesOrderID = SOD.SalesOrderID
JOIN Production.Product P ON SOD.ProductID = P.ProductID
JOIN Production.ProductSubcategory PSC ON P.ProductSubcategoryID = PSC.ProductSubcategoryID
JOIN Production.ProductCategory PC ON PSC.ProductCategoryID = PC.ProductCategoryID
WHERE PC.Name = 'Bikes' AND SOH.SalesPersonID IS NOT NULL
GROUP BY SOH.SalesPersonID
HAVING COUNT(DISTINCT P.ProductID) = (
    SELECT COUNT(P2.ProductID)
    FROM Production.Product P2
    JOIN Production.ProductSubcategory PSC2 ON P2.ProductSubcategoryID = PSC2.ProductSubcategoryID
    JOIN Production.ProductCategory PC2 ON PSC2.ProductCategoryID = PC2.ProductCategoryID
    WHERE PC2.Name = 'Bikes'
);
```

	SalesPersonID	ProductosDistintosVendidos
1	276	97
2	277	97
3	279	97
4	281	97
5	282	97

1. Cambia a categoría "Clothing" (ID=4).

```
SELECT
    SOH.SalesPersonID,
    PC.Name AS Categoria
FROM Sales.SalesOrderHeader SOH
JOIN Sales.SalesOrderDetail SOD ON SOH.SalesOrderID = SOD.SalesOrderID
JOIN Production.Product P ON SOD.ProductID = P.ProductID
JOIN Production.ProductSubcategory PSC ON P.ProductSubcategoryID = PSC.ProductSubcategoryID
JOIN Production.ProductCategory PC ON PSC.ProductCategoryID = PC.ProductCategoryID
WHERE PC.ProductCategoryID = 4 --Clothing
AND SOH.SalesPersonID IS NOT NULL
GROUP BY SOH.SalesPersonID, PC.Name;
```

## 2. Cuenta cuántos productos por categoría maneja cada vendedor.

```
SELECT
    SOH.SalesPersonID,
    PC.Name AS Categoria,
    COUNT(DISTINCT P.ProductID) AS CantidadProductosManejados
FROM Sales.SalesOrderHeader SOH
JOIN Sales.SalesOrderDetail SOD ON SOH.SalesOrderID = SOD.SalesOrderID
JOIN Production.Product P ON SOD.ProductID = P.ProductID
JOIN Production.ProductSubcategory PSC ON P.ProductSubcategoryID = PSC.ProductSubcategoryID
JOIN Production.ProductCategory PC ON PSC.ProductCategoryID = PC.ProductCategoryID
WHERE PC.ProductCategoryID = 4
AND SOH.SalesPersonID IS NOT NULL
GROUP BY SOH.SalesPersonID, PC.Name
ORDER BY CantidadProductosManejados DESC;
```

## 3. Documenta la solución inicial y solución con las variantes solicitadas.

Para determinar qué vendedores han cubierto la totalidad de la categoría "Bikes", se comparó el conteo de COUNT DISTINCT de los productos vendidos por cada empleado contra el total de registros en el catálogo de dicha categoría. En las variantes solicitadas, se transitó hacia el uso de identificadores numéricos (ProductCategoryID = 4). El conteo final por categoría fue con el mismo comando solo que encapsulando la variación.

Ejercicio 5: Determinar el producto más vendido de cada categoría de producto, considerando el escenario de que el esquema SALES se encuentra en una instancia (servidor) A y el esquema PRODUCTION en otra instancia (servidor) B.

```
SELECT Categoria, Producto, TotalVendido
FROM (
    SELECT
        PC.Name AS Categoria,
        P.Name AS Producto,
        SUM(SOD.OrderQty) AS TotalVendido,
        ROW_NUMBER() OVER(PARTITION BY PC.Name ORDER BY SUM(SOD.OrderQty) DESC) as Ranking
    FROM Sales.SalesOrderDetail SOD -- LOCAL
    JOIN [26.7.159.60].[AdventureWorks2022].[Production].[Product] P
        ON SOD.ProductID = P.ProductID
    JOIN [26.7.159.60].[AdventureWorks2022].[Production].[ProductSubcategory] PS
        ON P.ProductSubcategoryID = PS.ProductSubcategoryID
    JOIN [26.7.159.60].[AdventureWorks2022].[Production].[ProductCategory] PC
        ON PS.ProductCategoryID = PC.ProductCategoryID
    GROUP BY PC.Name, P.Name
) AS Resultados
WHERE Ranking = 1;
```

27 %

Results Messages

	Categoria	Producto	TotalVendido
1	Accessories	Water Bottle - 30 oz.	6815
2	Bikes	Mountain-200 Black, 38	2977
3	Clothing	AWC Logo Cap	8311
4	Components	LL Road Frame - Black, 52	1581

La consulta mostrada en la imagen utiliza una subconsulta con una función de ventana denominada ROW\_NUMBER() para filtrar el producto más exitoso de cada categoría. Primero, realiza una unión de tablas JOINS para consolidar la información de productos y ventas, calculando el total vendido mediante un agrupamiento. La clave del proceso es la instrucción OVER(PARTITION BY PC.Name ORDER BY SUM(...) DESC), la cual asigna un ranking numérico a cada producto dentro de su respectiva categoría, posicionando al que tiene mayores ventas en el número uno. Finalmente, la consulta externa aplica un filtro WHERE Ranking = 1, lo que permite que el resultado final muestra exclusivamente el producto líder por cada grupo, eliminando el resto de los registros secundarios y generando un reporte de "best-sellers" limpio y preciso.

Procesos												
	P...	Inici...	Bas...	Esta...	Com...	Aplic...	Tiempo d...	Tipo...	Rec...	B...	B...	Uso ...
51	1	Alondra	master			SQL Ser...	0					32 LAPT...
52	1	Alondra	master	SUSPEN...	SELECT	SQL Ser...	9810	XE_LIVE...				0 LAPT...
53	1	Alondra	master			SQL Ser...	0					32 LAPT...
54	1	Alondra	master	SUSPEN...	SELECT	SQL Ser...	7812	XE_LIVE...				0 LAPT...
57	1	NT SER...	master			SQLServ...	0					32 LAPT...
58	1	Alondra	master			Microsoft...	0					32 LAPT...
59	1	Alondra	master			Microsoft...	0					40 LAPT...
75	1	Alondra	master			Microsoft...	0					48 FER
75	1	Alondra	master			Microsoft...	0					48 FER
77	1	Alondra	tempdb	RUNNING	SELECT	SQL Ser...	0					40 LAPT...

Conexión establecida de ambas bases de datos.

Dado que la práctica se realizó originalmente en la red local de la escuela y no se conservaron las capturas necesarias, se utilizó Radmin VPN para recrear ese entorno de forma remota. Esta herramienta crea una red local virtual (VLAN) que permite conectar nuestra computadora con el servidor mediante una IP privada (26.x.x.x), simulando una conexión física de laboratorio. Esto nos permitió acceder nuevamente a la base de datos desde casa.