

Unidad 1 – Ejercicios Window Functions

Bases de Datos Aplicada

v1.0 – Marzo 2024



Universidad Nacional
de La Matanza

DIIT
Departamento de Ingeniería e
Investigaciones Tecnológicas

En primer lugar...

Vamos a crear nuestra base de datos y el esquema que contendrá las tablas para realizar los ejercicios

```
Use master  
go
```

```
IF NOT EXISTS ( SELECT name FROM master.dbo.sysdatabases WHERE name = 'PracticaWF')
```

```
BEGIN
```

```
    CREATE DATABASE PracticaWF  
    COLLATE Latin1_General_CI_AI;
```

```
END
```

```
go
```

```
use PracticaWF  
go
```

Esta es una verificación condicional. Verifica si existe una base de datos llamada 'PracticaWF'. Si no existe, pasa al siguiente bloque de código.

Es la configuración de ordenamiento (COLLATE) en 'Latin1_General_CI_AI' (no distingue mayúsculas de minúsculas y es insensible a acentos)

```
IF NOT EXISTS (SELECT * FROM sys.schemas WHERE name = 'tablasWF')
```

```
BEGIN
```

```
    EXEC('CREATE SCHEMA tablasWF')
```

```
END
```

```
GO
```

Esta es otra verificación condicional. Verifica si existe un esquema llamado 'tablasWF' dentro de la base de datos 'PracticaWF'. Si no existe, pasa al siguiente bloque de código.

Creación de tabla Empleados

```
IF NOT EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE
TABLE_SCHEMA = 'tablasWF' AND TABLE_NAME = 'Empleados')
BEGIN
    CREATE TABLE tablaswf.Empleados (
        EmpleadoID INT identity(1,1) primary key,
        Nombre VARCHAR(50),
        Departamento VARCHAR(50),
        Salario DECIMAL(10, 2)
    )
END
GO
```

Inserción de datos Empleados

```
INSERT INTO tablaswf.Empleados (Nombre, Departamento, Salario)  
VALUES
```

```
('Juan', 'Ventas', 3000.00),  
( 'María', 'Ventas', 2800.00),  
( 'Pedro', 'Marketing', 3200.00),  
( 'Laura', 'Marketing', 3500.00),  
( 'Carlos', 'IT', 4000.00);
```

Ejercicio 1

Enumerar, de mayor a menor, los empleados de una tabla según el salario que poseen

Resultado esperado

Results Messages					
	EmpleadoID	Nombre	Departamento	Salario	OrdenEmpleadosSalario
1	5	Carlos	IT	4000.00	1
2	4	Laura	Marketing	3500.00	2
3	3	Pedro	Marketing	3200.00	3
4	1	Juan	Ventas	3000.00	4
5	2	María	Ventas	2800.00	5

Ejercicio 2

Clasifica a los empleados del ejercicio anterior según sus salarios agrupado por departamento.

Insertamos más datos

```
INSERT INTO tablaswf.Empleados (Nombre, Departamento, Salario)  
VALUES
```

```
('Ramiro', 'Ventas', 1800.00),  
( 'Tomas', 'Ventas', 3200.00),  
( 'Erik', 'Marketing', 1477.00),  
( 'Esteban', 'Marketing', 15000.00),  
( 'Laura', 'IT', 452.00),  
( 'Romina', 'Ventas', 7855.00),  
( 'Susana', 'Ventas', 1233.00),  
( 'Mateo', 'Marketing', 4755.00),  
( 'Nicolas', 'Marketing', 1236.00),  
( 'Federico', 'IT', 260611.00),  
( 'Miguel', 'Ventas', 4688.00),  
( 'Josefina', 'Ventas', 2855.00),  
( 'Franco', 'Marketing', 7456.00),  
( 'Cesar', 'Marketing', 2555.00),  
( 'Patricio', 'IT', 4000.00)
```

Ejercicio 2

Resultado esperado

	EmpleadoID	Nombre	Departamento	Salario	Ranking
1	15	Federico	IT	260611.00	1
2	5	Carlos	IT	4000.00	2
3	20	Patricio	IT	4000.00	2
4	10	Laura	IT	452.00	4
5	9	Esteban	Marketing	15000.00	1
6	18	Franco	Marketing	7456.00	2
7	13	Mateo	Marketing	4755.00	3
8	4	Laura	Marketing	3500.00	4
9	3	Pedro	Marketing	3200.00	5
10	19	Cesar	Marketing	2555.00	6
11	8	Erik	Marketing	1477.00	7
12	14	Nicolas	Marketing	1236.00	8
13	11	Romina	Ventas	7855.00	1
14	16	Miguel	Ventas	4688.00	2
15	7	Tomas	Ventas	3200.00	3
16	1	Juan	Ventas	3000.00	4
17	17	Josefina	Ventas	2855.00	5
18	2	María	Ventas	2800.00	6
19	6	Ramiro	Ventas	1800.00	7
20	12	Susana	Ventas	1233.00	8

Ejercicio 3

Dividí a los empleados del ejercicio anterior en 4 grupos basados en su salario. El grupo 1 contendrá a los empleados con los salarios más altos, mientras que el grupo 4 incluirá a aquellos con los salarios más bajos. Esto permite asignar un grupo a cada empleado en función de su nivel de salario dentro de la empresa.

Ejercicio 3

Resultado esperado

	EmpleadoID	Nombre	Departamento	Salario	GrupoSalario
1	15	Federico	IT	260611.00	1
2	9	Esteban	Marketing	15000.00	1
3	11	Romina	Ventas	7855.00	1
4	18	Franco	Marketing	7456.00	1
5	13	Mateo	Marketing	4755.00	1
6	16	Miguel	Ventas	4688.00	2
7	5	Carlos	IT	4000.00	2
8	20	Patricio	IT	4000.00	2
9	4	Laura	Marketing	3500.00	2
10	3	Pedro	Marketing	3200.00	2
11	7	Tomas	Ventas	3200.00	3
12	1	Juan	Ventas	3000.00	3
13	17	Josefina	Ventas	2855.00	3
14	2	María	Ventas	2800.00	3
15	19	Cesar	Marketing	2555.00	3
16	6	Ramiro	Ventas	1800.00	4
17	8	Erik	Marketing	1477.00	4
18	14	Nicolas	Marketing	1236.00	4
19	12	Susana	Ventas	1233.00	4
20	10	Laura	IT	452.00	4

Ejercicio 4

Realizar una comparación de salarios entre empleados (del ejercicio anterior) para analizar la diferencia de salario entre el empleado actual y el siguiente, así como el salario del empleado anterior, dentro de un mismo departamento.

Generar una consulta que muestre, para cada empleado, su salario actual, el salario del empleado anterior en orden ascendente de salarios dentro del mismo departamento, y el salario del empleado siguiente en el mismo orden de salarios. Esta comparación permitirá visualizar cómo varían los salarios entre los empleados, mostrando la relación de los salarios con respecto al empleado anterior y posterior en términos de monto.

Ejercicio 4

Resultado esperado

	EmpleadoID	Nombre	Departamento	Salario	SalarioAnterior	SiguienteSalario
1	10	Laura	IT	452.00	0.00	4000.00
2	5	Carlos	IT	4000.00	452.00	4000.00
3	20	Patricio	IT	4000.00	4000.00	260611.00
4	15	Federico	IT	260611.00	4000.00	0.00
5	14	Nicolas	Marketing	1236.00	0.00	1477.00
6	8	Erik	Marketing	1477.00	1236.00	2555.00
7	19	Cesar	Marketing	2555.00	1477.00	3200.00
8	3	Pedro	Marketing	3200.00	2555.00	3500.00
9	4	Laura	Marketing	3500.00	3200.00	4755.00
10	13	Mateo	Marketing	4755.00	3500.00	7456.00
11	18	Franco	Marketing	7456.00	4755.00	15000.00
12	9	Esteban	Marketing	15000.00	7456.00	0.00
13	12	Susana	Ventas	1233.00	0.00	1800.00
14	6	Ramiro	Ventas	1800.00	1233.00	2800.00
15	2	María	Ventas	2800.00	1800.00	2855.00
16	17	Josefina	Ventas	2855.00	2800.00	3000.00
17	1	Juan	Ventas	3000.00	2855.00	3200.00
18	7	Tomas	Ventas	3200.00	3000.00	4688.00
19	16	Miguel	Ventas	4688.00	3200.00	7855.00
20	11	Romina	Ventas	7855.00	4688.00	0.00

Creación de tabla Clientes

```
IF NOT EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE
TABLE_SCHEMA = 'tablasWF' AND TABLE_NAME = 'Clientes')
BEGIN
    CREATE TABLE tablaswf.Clientes (
        id_cliente INT identity(1,1) PRIMARY KEY,
        nombre VARCHAR(50),
        pais VARCHAR(50)
    )
END
GO
```

Creación de tabla Pedidos

```
IF NOT EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA =  
'tablasWF' AND TABLE_NAME = 'Pedidos')  
BEGIN  
    CREATE TABLE tablaswf.Pedidos (  
        id_pedido INT PRIMARY KEY,  
        id_cliente INT,  
        fecha_pedido DATE,  
        monto DECIMAL(10, 2),  
        FOREIGN KEY (id_cliente) REFERENCES tablaswf.Clientes(id_cliente)  
    )  
END  
GO
```

Inserción de datos Clientes

```
INSERT INTO tablaswf.Clientes (nombre, pais)
VALUES
```

```
('John Doe', 'Argentina'),
('Jane Smith', 'Australia'),
('Juan García', 'Brasil'),
('Maria Hernandez', 'Canadá'),
('Michael Johnson', 'China'),
('Sophie Martin', 'Dinamarca'),
('Ahmad Khan', 'Egipto'),
('Emily Brown', 'Francia'),
('Hans Müller', 'Alemania'),
('Sofia Rossi', 'Italia'),
('Takeshi Yamada', 'Japón'),
('Javier López', 'México'),
('Eva Novak', 'Países Bajos'),
('Rafael Silva', 'Portugal'),
('Olga Petrova', 'Rusia'),
('Fernanda Gonzalez', 'España'),
('Mohammed Ali', 'Egipto'),
('Lena Schmidt', 'Alemania'),
('Yuki Tanaka', 'Japón'),
('Lucas Costa', 'Brasil');
```

Inserción de datos Pedidos

Para insertar datos en la tabla pedidos, vamos a ver una forma de generar datos aleatorios. El siguiente script inserta 100 registros en la tabla Pedidos, asignando aleatoriamente cada pedido a uno de los 20 clientes creados anteriormente en la tabla Clientes. Los montos de los pedidos se generarán de forma aleatoria entre 1000 y 6000, y las fechas de los pedidos estarán dentro del rango de tiempo definido de 2023.

Inserción de datos Pedidos

```
DECLARE @startDate DATE = '2023-01-01';
DECLARE @endDate DATE = '2023-12-31';
DECLARE @orderId INT = 1;

WHILE @orderId <= 100
BEGIN
    INSERT INTO tablaswf.Pedidos (id_pedido,id_cliente, fecha_pedido, monto)
    VALUES (
        @orderId,
        ((@orderId - 1) % 20) + 1,
        DATEADD(DAY, ABS(CHECKSUM(NEWID())) % (DATEDIFF(DAY, @startDate, @endDate) + 1),
        @startDate),
        ROUND(RAND(CHECKSUM(NEWID())) * 5000 + 1000, 2)
    );
    SET @orderId = @orderId + 1;
END
```


Ejercicio 5

Calcular el promedio de los montos de pedidos por cliente, mostrando también el monto de cada pedido y su posición relativa en comparación con el promedio de los montos para ese cliente.

Ejercicio 5

Resultado esperado

Este caso, es un ejemplo, ya que utilizamos una funcion random para cargar los datos de los pedidos

	id_pedido	id_cliente	monto	promedio_monto_cliente	posicion_rel_monto_cliente
1	21	1	1851.67	3303.356000	1
2	81	1	1881.53	3303.356000	2
3	61	1	2589.40	3303.356000	3
4	1	1	4934.38	3303.356000	4
5	41	1	5259.80	3303.356000	5
6	22	2	2013.11	3353.258000	1
7	62	2	2576.48	3353.258000	2
8	2	2	2774.18	3353.258000	3
9	82	2	4533.92	3353.258000	4
10	42	2	4868.60	3353.258000	5
11	3	3	1760.44	2980.192000	1
12	83	3	2405.06	2980.192000	2
13	43	3	2908.27	2980.192000	3
14	23	3	3650.45	2980.192000	4
15	63	3	4176.74	2980.192000	5
16	44	4	2365.78	3290.846000	1
17	64	4	2627.07	3290.846000	2
18	4	4	3475.95	3290.846000	3

Ejercicio 6

Encontrar a los tres principales clientes (por monto total de pedidos) de cada país, mostrando su nombre, país y el monto total de sus pedidos.

Ejercicio 6

Resultado esperado

Este caso, es un ejemplo, ya que utilizamos una funcion random para cargar los datos de los pedidos

	nombre	pais	monto_total_pedidos	ranking_por_pais
1	Lena Schmidt	Alemania	18949.32	1
2	Hans Müller	Alemania	14912.07	2
3	John Doe	Argentina	16516.78	1
4	Jane Smith	Australia	16766.29	1
5	Lucas Costa	Brasil	20126.23	1
6	Juan García	Brasil	14900.96	2
7	Maria Hernandez	Canadá	16454.23	1
8	Michael Johnson	China	11962.19	1
9	Sophie Martin	Dinamarca	17012.06	1
10	Mohammed Ali	Egipto	16933.97	1
11	Ahmad Khan	Egipto	12000.14	2
12	Fernanda Gonzalez	España	14956.52	1
13	Emily Brown	Francia	20452.70	1
14	Sofia Rossi	Italia	17063.69	1
15	Yuki Tanaka	Japón	9395.87	1
16	Takeshi Yamada	Japón	9293.28	2
17	Javier López	México	20367.48	1
18	Eva Novak	Países Bajos	19275.10	1
19	Rafael Silva	Portugal	18771.52	1
20	Olga Petrova	Rusia	15732.60	1

Ejercicio 7

Calcular la diferencia de monto entre un pedido y el siguiente pedido realizado por el mismo cliente, ordenado por fecha de pedido. Muestra el ID del pedido, el ID del cliente, la fecha del pedido y la diferencia de monto.

Ejercicio 7

Resultado esperado

Este caso, es un ejemplo, ya que utilizamos una funcion random para cargar los datos de los pedidos

	id_pedido	id_cliente	fecha_pedido	monto	diferencia_monto
1	41	1	2023-04-12	5259.80	-3408.13
2	21	1	2023-07-08	1851.67	737.73
3	61	1	2023-08-23	2589.40	2344.98
4	1	1	2023-11-10	4934.38	-3052.85
5	81	1	2023-12-15	1881.53	NULL
6	42	2	2023-05-27	4868.60	-2094.42
7	2	2	2023-08-10	2774.18	-197.70
8	62	2	2023-08-23	2576.48	-563.37
9	22	2	2023-08-29	2013.11	2520.81
10	82	2	2023-09-13	4533.92	NULL
11	83	3	2023-02-28	2405.06	1771.68
12	63	3	2023-03-22	4176.74	-2416.30
13	3	3	2023-05-17	1760.44	1147.83
14	43	3	2023-09-28	2908.27	742.18
15	23	3	2023-12-21	3650.45	NULL
16	4	4	2023-07-23	3475.95	-848.88
17	64	4	2023-10-01	2627.07	-261.29
18	44	4	2023-10-25	2365.78	1811.12

Ejercicio 8

Determina el percentil de monto de cada pedido en relación con todos los pedidos realizados por clientes del mismo país. Muestra el ID del pedido, el ID del cliente, el monto del pedido y su percentil.

Ejercicio 8

Resultado esperado

Este caso, es un ejemplo, ya que utilizamos una funcion random para cargar los datos de los pedidos

	id_pedido	id_cliente	pais	monto	percentil_monto
1	49	9	Alemania	1086.16	0
2	9	9	Alemania	1733.58	0.1111111111111111
3	98	18	Alemania	2391.09	0.2222222222222222
4	38	18	Alemania	2550.12	0.3333333333333333
5	89	9	Alemania	3605.04	0.4444444444444444
6	78	18	Alemania	3689.33	0.5555555555555556
7	69	9	Alemania	4097.36	0.6666666666666667
8	29	9	Alemania	4389.93	0.7777777777777778
9	58	18	Alemania	4900.52	0.8888888888888889
10	18	18	Alemania	5418.26	1
11	21	1	Argentina	1851.67	0
12	81	1	Argentina	1881.53	0.25
13	61	1	Argentina	2589.40	0.5
14	1	1	Argentina	4934.38	0.75
15	41	1	Argentina	5259.80	1
16	22	2	Australia	2013.11	0
17	62	2	Australia	2576.48	0.25
18	2	2	Australia	2774.18	0.5
19	82	2	Australia	4522.02	0.75

Ejercicio 9

Para cada cliente, muestra el ID del pedido, el número total de pedidos realizados por ese cliente, su nombre y la posición relativa de cada pedido en relación con el total de pedidos del cliente (ordenados por fecha de pedido).

Ejercicio 9

Resultado esperado

Este caso, es un ejemplo, ya que utilizamos una funcion random para cargar los datos de los pedidos

	id_pedido	id_cliente	nombre_cliente	total_pedidos_cliente	posicion_rel_pedidos_cliente
1	41	1	John Doe	5	1
2	21	1	John Doe	5	2
3	61	1	John Doe	5	3
4	1	1	John Doe	5	4
5	81	1	John Doe	5	5
6	42	2	Jane Smith	5	1
7	2	2	Jane Smith	5	2
8	62	2	Jane Smith	5	3
9	22	2	Jane Smith	5	4
10	82	2	Jane Smith	5	5
11	83	3	Juan García	5	1
12	63	3	Juan García	5	2
13	3	3	Juan García	5	3
14	43	3	Juan García	5	4
15	23	3	Juan García	5	5
16	4	4	Maria Hernandez	5	1
17	64	4	Maria Hernandez	5	2
18	44	4	Maria Hernandez	5	3