

BASES DE DATOS

CASO PRACTICO I UD 5



ALUMNO CESUR

24/25

Alejandro Muñoz de la Sierra

PROFESOR

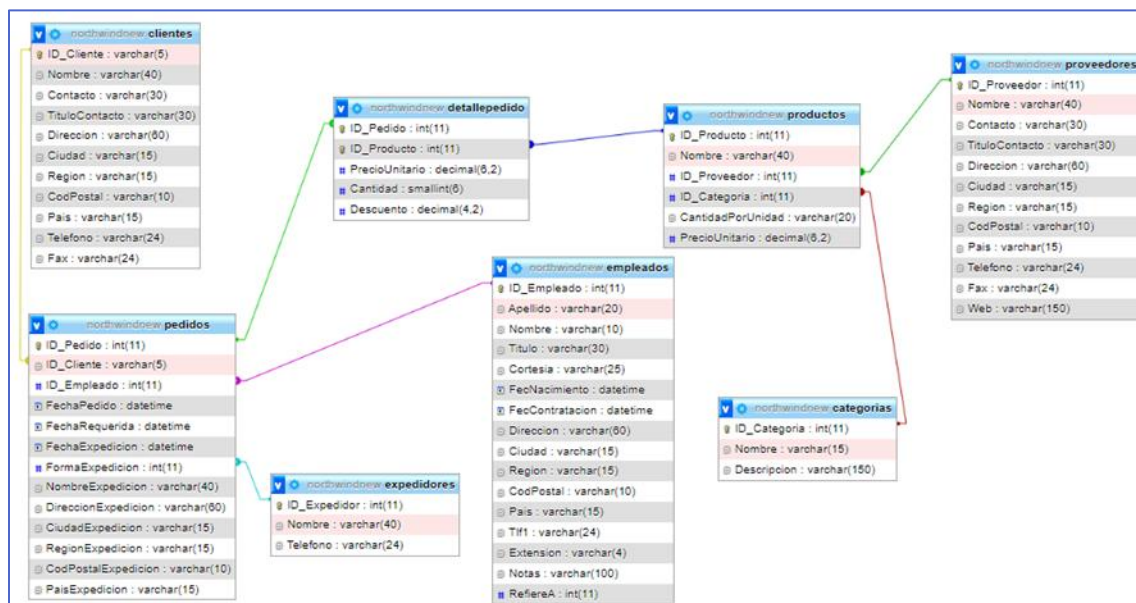
Inmaculada Morales Quesada

CASO PRÁCTICO 1

CONSULTAS AVANZADAS Y TRANSACCIONES

CONTEXTO

En tu empresa, realizas múltiples consultas en bases de datos internas, y muchas de ellas son consultas avanzadas. Esta semana se ha incorporado a tu departamento un desarrollador nuevo, y te han pedido que lo ayudes. Para ello, y dada la siguiente estructura de base de datos:



Para crear las tablas e insertar los registros para la tarea, puedes ejecutar el script adjunto: [Recurso caso](#)

Te piden que resuelvas los siguientes ejercicios:

1) Mostrar: código de la categoría, el nombre de la categoría, cod.Producto, nombre del producto y precio.

SELECT

```
c.ID_Categoria AS Codigo_Categoria,  
c.Nombre AS Nombre_Categoria,  
p.ID_Producto AS Codigo_Producto,  
p.Nombre AS Nombre_Producto,  
p.PrecioUnitario AS Precio
```

FROM

Categorias c

JOIN

Productos p ON c.ID_Categoria = p.ID_Categoria;

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'SCHEMAS' pane displays the 'northwindnew' database structure, including tables like 'categorias' and 'productos'. The 'categorias' table is expanded, showing columns: ID_Categoria (int(11) AI PK), Nombre (varchar(15)), and Description (varchar(150)).

The main pane displays a SQL query in 'SQL File 16*':

```
1 • SELECT  
2     c.ID_Categoria AS Codigo_Categoria,  
3     c.Nombre AS Nombre_Categoria,  
4     p.ID_Producto AS Codigo_Producto,  
5     p.Nombre AS Nombre_Producto,  
6     p.PrecioUnitario AS Precio  
7 FROM  
8     Categorias c  
9 JOIN  
10    Productos p ON c.ID_Categoria = p.ID_Categoria;
```

Below the query, the 'Result Grid' shows the results of the query. The columns are: Codigo_Categoria, Nombre_Categoria, Codigo_Producto, Nombre_Producto, and Precio. The results are as follows:

Codigo_Categoria	Nombre_Categoria	Codigo_Producto	Nombre_Producto	Precio
1	Beverages	1	Chai	18.00
1	Beverages	2	Chang	19.00
1	Beverages	24	Guaraná Fantástica	4.50
1	Beverages	34	Sasquatch Ale	14.00
1	Beverages	35	Steeleye Stout	18.00
1	Beverages	38	Côte de Blaye	263.50
1	Beverages	39	Chartreuse verte	18.00
1	Beverages	43	Ipoh Coffee	46.00
1	Beverages	67	Laughing Lumberjack Lager	14.00
1	Beverages	70	Outback Lager	15.00

2) Mostrar: código, nombre, ciudad y país de proveedor, código, nombre y precio del producto.

SELECT

```
pr.ID_Proveedor AS Codigo_Proveedor,  
pr.Nombre AS Nombre_Proveedor,  
pr.Ciudad AS Ciudad_Proveedor,  
pr.Pais AS Pais_Proveedor,  
p.ID_Producto AS Codigo_Producto,  
p.Nombre AS Nombre_Producto,  
p.PrecioUnitario AS Precio_Producto
```

FROM

Proveedores pr

JOIN

Productos p ON pr.ID_Proveedor = p.ID_Proveedor;

The screenshot shows a SQL IDE interface with a query editor and a results grid. The query is a SELECT statement joining the 'proveedores' and 'productos' tables. The results grid displays the following data:

	Codigo_Proveedor	Nombre_Proveedor	Ciudad_Proveedor	Pais_Proveedor	Codigo_Producto	Nombre_Producto	Precio_Producto
1	1	Exotic Liquids	London	UK	1	Chai	18.00
1	1	Exotic Liquids	London	UK	2	Chang	19.00
1	1	Exotic Liquids	London	UK	3	Aniseed Syrup	10.00
2	2	New Orleans Cajun Delights	New Orleans	USA	4	Chef Anton's Cajun Seasoning	22.00
2	2	New Orleans Cajun Delights	New Orleans	USA	5	Chef Anton's Gumbo Mix	21.35
2	2	New Orleans Cajun Delights	New Orleans	USA	65	Louisiana Fiery Hot Pepper Sauce	21.05
2	2	New Orleans Cajun Delights	New Orleans	USA	66	Louisiana Hot Spiced Okra	17.00

3) Mostrar: núm. de la orden, fecha, persona de contacto del cliente y el nombre y apellidos del empleado

SELECT

```
pe.ID_Pedido AS Numero_Orden,  
pe.FechaPedido AS Fecha,  
c.Contacto AS Persona_Contacto_Cliente,  
e.Nombre AS Nombre_Empleado,  
e.Apellido AS Apellido_Empleado
```

FROM

```
Pedidos pe
```

JOIN

```
Clientes c ON pe.ID_Cliente = c.ID_Cliente
```

JOIN

```
Empleados e ON pe.ID_Empleado = e.ID_Empleado;
```

The screenshot shows a SQL IDE interface with a Navigator on the left, a SQL editor in the center, and a Result Grid at the bottom. The Navigator shows a database named 'northwindnew' with tables 'categorias', 'clientes', 'detallepedido', 'empleados', 'expedidores', and 'pedidos'. The 'pedidos' table is selected, showing its columns: ID_Pedido, ID_Cliente, ID_Empleado, FechaPedido, FechaRequerida, and FormaExpedicion. The SQL editor contains the following query:

```
1 • SELECT  
2     pe.ID_Pedido AS Numero_Orden,  
3     pe.FechaPedido AS Fecha,  
4     c.Contacto AS Persona_Contacto_Cliente,  
5     e.Nombre AS Nombre_Empleado,  
6     e.Apellido AS Apellido_Empleado  
7 FROM  
8     Pedidos pe  
9 JOIN  
10    Clientes c ON pe.ID_Cliente = c.ID_Cliente  
11 JOIN  
12    Empleados e ON pe.ID_Empleado = e.ID_Empleado;  
13
```

The Result Grid shows the following data:

Numero_Orden	Fecha	Persona_Contacto_Cliente	Nombre_Empleado	Apellido_Empleado
10258	1996-07-17	Roland Mendel	Nancy	Davolio
10270	1996-01-06	Pirkko Koskitalo	Nancy	Davolio
10275	1996-07-07	Giovanni Rovelli	Nancy	Davolio
10285	1996-01-05	Horst Kloss	Nancy	Davolio
10292	1996-12-21	Anabela Domingues	Nancy	Davolio
10293	1996-09-01	Miguel Angel Paolino	Nancy	Davolio
10304	1996-10-10	Miguel Angel Paolino	Nancy	Davolio
10306	1996-10-12	Alejandra Camino	Nancy	Davolio

4) Visualizar el mínimo precio de los productos, mostrar el nombre de la categoría y el producto

SELECT

```
c.Nombre AS Nombre_Categoria,  
p.Nombre AS Nombre_Producto,  
MIN(p.PrecioUnitario) AS Precio_Minimo
```

FROM

Categorias c

JOIN

Productos p ON c.ID_Categoria = p.ID_Categoria

GROUP BY

c.Nombre, p.Nombre;

The screenshot shows a SQL Enterprise Manager interface. On the left, the 'SCHEMAS' pane displays a tree view of the 'northwindnew' database, including tables like 'categorias', 'clientes', 'detallepedido', 'empleados', 'expedidores', 'pedidos', 'productos', and 'proveedores'. The 'categorias' table is expanded, showing columns 'ID_Categoria', 'Nombre', and 'Descripcion'. The main pane displays a SQL query in 'SQL File 19*'. The query is as follows:

```
1 SELECT  
2     c.Nombre AS Nombre_Categoria,  
3     p.Nombre AS Nombre_Producto,  
4     MIN(p.PrecioUnitario) AS Precio_Minimo  
5 FROM  
6     Categorias c  
7 JOIN  
8     Productos p ON c.ID_Categoria = p.ID_Categoria  
9 GROUP BY  
10    c.Nombre, p.Nombre;  
11
```

Below the query, the 'Result Grid' shows the results of the query. The grid has four columns: 'Nombre_Categoria', 'Nombre_Producto', and 'Precio_Minimo'. The results are as follows:

Nombre_Categoria	Nombre_Producto	Precio_Minimo
Beverages	Chai	18.00
Beverages	Chang	19.00
Beverages	Chartreuse verte	18.00
Beverages	Côte de Blaye	263.50
Beverages	Guaraná Fantástica	4.50
Beverages	Ipoh Coffee	46.00
Beverages	Lakkalikööri	18.00
Beverages	Laughing Lumberjack Lager	14.00
Beverages	Outback Lager	15.00
Beverages	Rhönbräu Klosterbier	7.75

5) Seleccionar las categorías que tengan más de 5 productos. Mostrar el nombre de la categoría y el número de productos

SELECT

c.Nombre AS Nombre_Categoria,
COUNT(p.ID_Producto) AS Numero_Productos

FROM

Categorias c

JOIN

Productos p ON c.ID_Categoria = p.ID_Categoria

GROUP BY

c.Nombre

HAVING

COUNT(p.ID_Producto) > 5;

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'SCHEMAS' pane displays the 'northwindnew' database structure, including tables like 'categorias' and 'productos'. The main window shows a SQL query in 'SQL File 20*' that selects category names and product counts, grouped by category name and filtered to show only categories with more than 5 products. Below the query, the 'Result Grid' displays the following data:

Nombre_Categoria	Numero_Productos
Beverages	12
Condiments	12
Confections	13
Dairy Productos	10
Grains/Cereals	7
Meat/Poultry	6
Seafood	12

6) Mostrar el nombre de todos los empleados y el número de órdenes que han hecho.

SELECT

```
e.Nombre AS Nombre_Empleado,  
e.Apellido AS Apellido_Empleado,  
COUNT(pe.ID_Pedido) AS Numero_Ordenes
```

FROM

```
Empleados e
```

LEFT JOIN

```
Pedidos pe ON e.ID_Empleado = pe.ID_Empleado
```

GROUP BY

```
e.ID_Empleado;
```

The screenshot shows a SQL IDE interface. On the left is a 'Navigator' pane with a tree view of schemas. The 'northwindnew' schema is expanded, showing tables like 'categorias', 'clientes', 'detallepedido', and 'empleados'. The 'empleados' table is selected, and its columns are listed: ID_Empleado, Apellido, Nombre, Titulo, Cortesia, FecNacimiento, FecContratacion, Direccion, Ciudad, and Region. Below the tree is a 'No object selected' message.

The main editor pane shows a SQL query:

```
1 • SELECT  
2     e.Nombre AS Nombre_Empleado,  
3     e.Apellido AS Apellido_Empleado,  
4     COUNT(pe.ID_Pedido) AS Numero_Ordenes  
5 FROM  
6     Empleados e  
7 LEFT JOIN  
8     Pedidos pe ON e.ID_Empleado = pe.ID_Empleado  
9 GROUP BY  
10    e.ID_Empleado;  
11
```

Below the query is a 'Result Grid' showing the results of the query. The grid has three columns: 'Nombre_Empleado', 'Apellido_Empleado', and 'Numero_Ordenes'. The results are as follows:

Nombre_Empleado	Apellido_Empleado	Numero_Ordenes
Nancy	Davolio	11
Andrew	Fuller	8
Janet	Leverling	7
Margaret	Peacock	15
Steven	Buchanan	4
Michael	Suyama	8
Robert	King	3
Laura	Callahan	1
Anne	Dodsworth	0

7) Mostrar el nombre y el precio del producto sólo cuando dicho precio sea igual o superior al precio medio de los productos.

SELECT

p.Nombre AS Nombre_Producto,

p.PrecioUnitario AS Precio

FROM

Productos p

WHERE

p.PrecioUnitario >= (SELECT AVG(PrecioUnitario) FROM Productos);

The screenshot shows a SQL IDE interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure, including tables like 'categorias', 'clientes', 'detallepedido', 'empleados', 'expedidores', 'pedidos', 'productos', and 'productos'. The 'productos' table is selected. The main editor displays the following SQL query:

```
1 • SELECT
2     p.Nombre AS Nombre_Producto,
3     p.PrecioUnitario AS Precio
4 FROM
5     Productos p
6 WHERE
7     p.PrecioUnitario >= (SELECT AVG(PrecioUnitario) FROM Productos);
8
```

Below the query editor, the 'Result Grid' shows the results of the query. The grid has two columns: 'Nombre_Producto' and 'Precio'. The results are as follows:

Nombre_Producto	Precio
Unde Bob's Organic Dried Pears	30.00
Northwoods Cranberry Sauce	40.00
Mishi Kobe Niku	97.00
Ikura	31.00
Queso Manchego La Pastora	38.00
Alice Mutton	39.00
Carnarvon Tigers	62.50
Sir Rodney's Marmalade	81.00
Gumbär Gummibärchen	31.23
Schoggi Schokolade	43.90

8) Mostrar el precio medio de los productos agrupados por categorías solo cuando dicho precio medio sea mayor que el precio medio de los productos de la categoría 4. Hazlo usando la cláusula HAVING

SELECT

c.Nombre AS Nombre_Categoria,
AVG(p.PrecioUnitario) AS Precio_Medio

FROM

Categorias c

JOIN

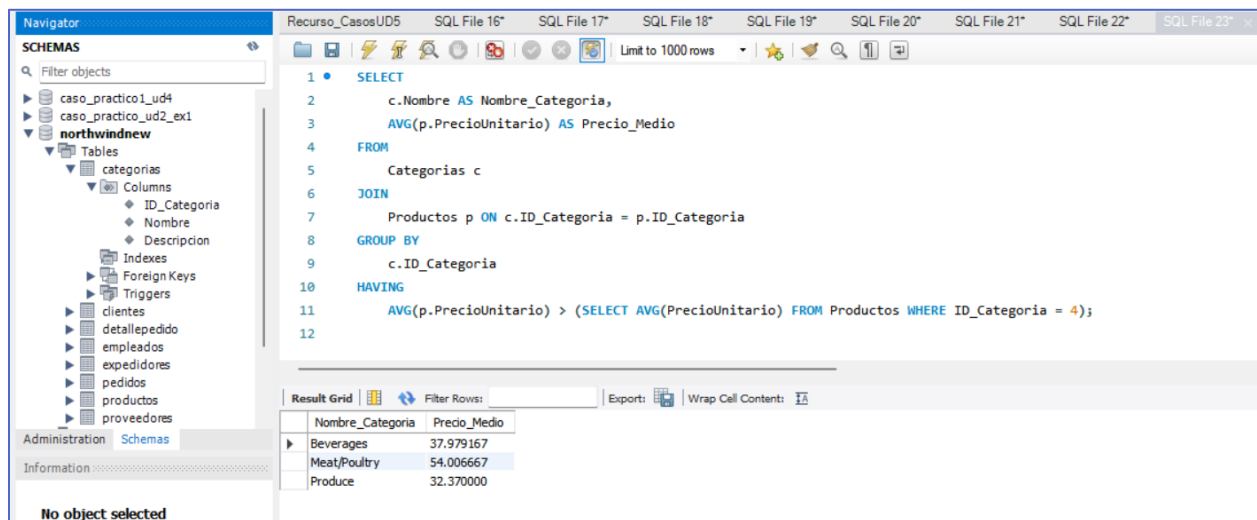
Productos p ON c.ID_Categoria = p.ID_Categoria

GROUP BY

c.ID_Categoria

HAVING

AVG(p.PrecioUnitario) > (SELECT AVG(PrecioUnitario) FROM Productos WHERE
ID_Categoria = 4);



The screenshot shows a SQL IDE interface. On the left is a 'Navigator' pane with a tree view of a database schema. The 'northwindnew' database is expanded, showing tables like 'categorias', 'clientes', 'detallepedido', etc. The 'categorias' table is selected. The main pane displays a SQL query with line numbers 1 through 12. The query is a SELECT statement that joins 'categorias' and 'productos', groups by category, and filters using a HAVING clause to compare the average price of each category to the average price of category 4. Below the query is a 'Result Grid' showing the output of the query. It has two columns: 'Nombre_Categoria' and 'Precio_Medio'. The results are: Beverages (37.979167), Meat/Poultry (54.006667), and Produce (32.370000).

```
1 SELECT
2   c.Nombre AS Nombre_Categoria,
3   AVG(p.PrecioUnitario) AS Precio_Medio
4 FROM
5   Categorias c
6 JOIN
7   Productos p ON c.ID_Categoria = p.ID_Categoria
8 GROUP BY
9   c.ID_Categoria
10 HAVING
11   AVG(p.PrecioUnitario) > (SELECT AVG(PrecioUnitario) FROM Productos WHERE ID_Categoria = 4);
12
```

Nombre_Categoria	Precio_Medio
Beverages	37.979167
Meat/Poultry	54.006667
Produce	32.370000

9) Obtener el Id del empleado, el nombre y el campo ID_Pedido para los empleados cuyo id de empleado sea mayor que 5. Ten en cuenta que en el caso de que algún empleado no haya registrado ningún pedido también queremos que aparezca. Para estos casos, haz que en el campo ORDER_ID aparezca un 0 en lugar de NULL.

```
SELECT
    e.ID_Empleado,
    e.Nombre,
    COALESCE(pe.ID_Pedido, 0) AS ID_Pedido
FROM
    Empleados e
LEFT JOIN
    Pedidos pe ON e.ID_Empleado = pe.ID_Empleado
WHERE
    e.ID_Empleado > 5;
```

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'Schemas' pane shows the 'northwind' database with tables 'categorias', 'clientes', 'detallepedido', and 'empleados'. The 'empleados' table is selected, and its columns are listed: ID_Empleado (int(11) PK), Apellido (varchar(20)), Nombre (varchar(10)), Titulo (varchar(30)), Cortesia (varchar(25)), FecNacimiento (date), FecContratacion (date), Direccion (varchar(60)), Ciudad (varchar(15)), Region (varchar(15)), CodPostal (varchar(10)), and Pais (varchar(15)).

The main pane shows a SQL query in 'SQL File 16*':

```
1 SELECT
2     e.ID_Empleado,
3     e.Nombre,
4     COALESCE(pe.ID_Pedido, 0) AS ID_Pedido
5 FROM
6     Empleados e
7 LEFT JOIN
8     Pedidos pe ON e.ID_Empleado = pe.ID_Empleado
9 WHERE
10    e.ID_Empleado > 5;
11
```

Below the query, the 'Result Grid' shows the results of the query. The columns are 'ID_Empleado', 'Nombre', and 'ID_Pedido'. The results are as follows:

ID_Empleado	Nombre	ID_Pedido
6	Michael	10249
6	Michael	10264
6	Michael	10271
6	Michael	10272
6	Michael	10274
6	Michael	10291
6	Michael	10296
6	Michael	10298
7	Robert	10289
7	Robert	10303

At the bottom, the 'Output' pane shows the execution details: 'Action Output' with a green checkmark, 'Time' 13:11:21, 'Action' SELECT, and 'Message' c.Nombre AS Nombre_Categoria, AVG(p.PrecioUnitario) AS Precio_Medio FROM Categ... 3 row(s) returned.

10) Haz lo mismo que en el apartado anterior, pero para los clientes que no han realizado ningún pedido. Haz que en lugar de mostrar el campo ORDER_ID muestre el valor “SIN PEDIDO”

SELECT

c.ID_Cliente,

c.Nombre,

COALESCE(pe.ID_Pedido, 'SIN PEDIDO') AS Orden_ID

FROM

Clientes c

LEFT JOIN

Pedidos pe ON c.ID_Cliente = pe.ID_Cliente;

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Schemas' pane shows the 'northwindnew' database with tables 'categorias' and 'clientes'. The 'clientes' table is selected, and its columns are listed: ID_Cliente (PK), Nombre, Contacto, TituloContacto, Direccion, Ciudad, Region, CodPostal, Pais, Telefono, and Fax. The main pane displays a SQL query:

```
1 SELECT
2   c.ID_Cliente,
3   c.Nombre,
4   COALESCE(pe.ID_Pedido, 'SIN PEDIDO') AS Orden_ID
5 FROM
6   Clientes c
7 LEFT JOIN
8   Pedidos pe ON c.ID_Cliente = pe.ID_Cliente;
9
```

Below the query, the 'Result Grid' shows the results of the query. The columns are ID_Cliente, Nombre, and Orden_ID. The results are as follows:

ID_Cliente	Nombre	Orden_ID
ALFKI	Alfreds Futterkiste	SIN PEDIDO
ANATR	Ana Trujillo Emparedados y helados	10308
ANTON	Antonio Moreno Taquería	SIN PEDIDO
AROUT	Around the Horn	SIN PEDIDO
BERGS	Berglunds snabbköp	10280
BLAUS	Blauer See Delikatessen	SIN PEDIDO
BLOMP	Blondies père et fils	10265
BLOMP	Blondies père et fils	10297
BOLID	Bólido Comidas preparadas	SIN PEDIDO
BONAP	Bon app'	SIN PEDIDO

At the bottom, the 'Output' pane shows the execution of the query, indicating that 110 row(s) were returned.

11) Crea una transacción en la que, sobre la tabla empleados, incrementes en 100 el salario del empleado con ID = 4 y reduzcas en 100 el salario del empleado con ID = 8. Por último, deshaz los cambios para finalizar la transacción.

```
START TRANSACTION;
```

```
UPDATE Empleados
```

```
SET Salario = Salario + 100
```

```
WHERE ID_Empleado = 4;
```

```
UPDATE Empleados
```

```
SET Salario = Salario - 100
```

```
WHERE ID_Empleado = 8;
```

```
-- Deshacer los cambios
```

```
ROLLBACK;
```

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'SCHEMAS' pane shows the 'northwindnew' database with tables 'categorias', 'clientes', 'detallepedido', and 'empleados'. The 'empleados' table structure is detailed in the 'Table: empleados' section, listing columns like ID_Empleado (PK), Apellido, Nombre, Titulo, Cortesia, FecNacimiento, FecContratacion, Direccion, Ciudad, Region, CodPostal, and Pais. The main window shows a query editor with the following SQL code:

```
1 START TRANSACTION;
2
3 UPDATE Empleados
4 SET Salario = Salario + 100
5 WHERE ID_Empleado = 4;
6
7 UPDATE Empleados
8 SET Salario = Salario - 100
9 WHERE ID_Empleado = 8;
10
11 -- Deshacer los cambios
12 ROLLBACK;
13
```

The 'Output' pane at the bottom shows the execution results:

#	Time	Action	Message
461	13:16:37	UPDATE Empleados SET Salario = Salario + 100 WHERE ID_Empleado = 4	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
462	13:16:37	ROLLBACK	0 row(s) affected

The status bar at the bottom indicates 'Query Completed'.

Conclusiones

Este caso práctico nos ha dado una buena visión de cómo manejar bases de datos mediante varias tareas clave, como crear tablas, agregar datos, hacer consultas y controlar transacciones. A lo largo de los ejercicios, hemos visto cómo diseñar una base de datos efectiva usando claves primarias y foráneas, manejar datos con las instrucciones INSERT, UPDATE y DELETE, y hacer consultas con SELECT para obtener información.

Un punto importante ha sido la aplicación de transacciones, usando las órdenes START TRANSACTION, COMMIT y ROLLBACK. Esto nos ha ayudado a entender cómo gestionar operaciones atómicas, asegurando la integridad de los datos y que los cambios se hagan de manera ordenada y segura. También hemos revisado los resultados y aprendido a verificar y controlar los cambios en la base de datos para evitar errores.

En conclusión, este caso práctico ha sido crucial para afianzar los conceptos básicos de manejo y administración de bases de datos, dándonos una base sobre la que podemos seguir aprendiendo más sobre el manejo de datos a medida que avanzamos. Las habilidades adquiridas aquí nos ayudarán a enfrentar retos más difíciles en la administración de bases de datos.

Referencias

<https://dev.mysql.com/doc/>

https://sqlzoo.net/wiki/SQL_Tutorial

<https://www.w3schools.com/sql/default.asp?PHPSESSID=300ae3404d5fa2612f238abeebb8869c>

https://www.youtube.com/watch?v=U8lgFaLW_Qg

<https://www.youtube.com/watch?v=6TaZtzojx6A>

https://www.apuntesinformaticafp.com/cursos/consultas_avanzadas.html