

## BASES DE DATOS - UD4 CASO 1

# CONSULTAS EN SQL

## Introducción

El presente informe detalla el proceso de creación, manipulación y consulta de una base de datos utilizando **MySQL Workbench**, una herramienta ampliamente utilizada para la gestión de bases de datos relacionales. El objetivo de este caso práctico ha sido aplicar conceptos fundamentales de SQL, permitiendo la gestión eficiente de datos mediante operaciones de selección, actualización y eliminación.

Se ha trabajado con una base de datos simulada de una **empresa ficticia**, orientada a la gestión de clientes. La tabla principal, denominada **customer**, almacena información clave como el DNI, nombre, dirección y presupuesto de los clientes. A través de este caso, se busca no sólo dominar la sintaxis de SQL, sino también comprender la lógica detrás del manejo de datos en escenarios empresariales reales.

## Metodología

Para el desarrollo del caso práctico, se siguió una metodología estructurada que permitió organizar de manera eficiente cada etapa del trabajo. El proceso se dividió en las siguientes fases:

- Definición de la Base de Datos:**  
Se creó la base de datos **empresa** y la tabla **customer** definiendo claramente sus campos y tipos de datos para asegurar la integridad de la información.
- Insertión de Datos:**  
Se insertaron registros de prueba con el objetivo de simular un entorno realista, permitiendo realizar consultas SQL significativas.
- Ejecución de Consultas:**  
Se llevaron a cabo diversas consultas SQL para extraer información específica, aplicando filtros, ordenamientos y funciones de agregación.
- Pruebas y Verificación:**  
Cada consulta fue validada mediante pruebas para garantizar la precisión de los resultados. En casos de errores o resultados inesperados, se realizaron análisis detallados para identificar la causa y corregir el problema.
- Resolución de Errores:**  
Se documentaron los errores encontrados durante el proceso (como el problema de la Consulta 9), describiendo el enfoque utilizado para solucionarlos y las lecciones aprendidas.

## BASES DE DATOS - UD4 CASO 1

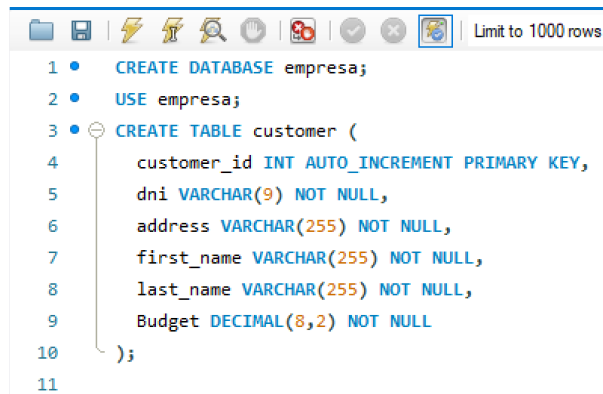
# CONSULTAS EN SQL

## Desarrollo del Caso Práctico

### 1. Creación de la Base de Datos y la Tabla

Se creó la base de datos **empresa** y la tabla **customer** con la siguiente estructura:

```
CREATE DATABASE empresa;  
USE empresa;  
  
CREATE TABLE customer (  
  customer_id INT AUTO_INCREMENT PRIMARY KEY,  
  dni VARCHAR(9) NOT NULL,  
  address VARCHAR(255) NOT NULL,  
  first_name VARCHAR(255) NOT NULL,  
  last_name VARCHAR(255) NOT NULL,  
  Budget DECIMAL(8,2) NOT NULL  
);
```



Se insertaron registros con los datos de varios clientes para realizar las consultas posteriores.

## BASES DE DATOS - UD4 CASO 1

## CONSULTAS EN SQL

Para verificar el funcionamiento correcto de la tabla antes de empezar a editar, hice la siguiente consulta:

The screenshot shows a SQL IDE with a query editor on the left and a result grid on the right. The query editor contains the following SQL code:

```

1 • CREATE DATABASE empresa;
2 • USE empresa;
3 • CREA
4 cu 1 • USE empresa;
5 dn 2 • SELECT * FROM customer;
6 ad 3
7 fi
8 la
9 Bu
10 );
11 • INSE
12 (1,
13 (2,
14 (3,
15 (4,
16 (5,
17 (6,
18 (7,
19 (8,
20 (9,
21 (10,

```

The result grid displays the following data:

customer_id	dni	address	first_name	last_name	Budget
1	45679823F	Calle	Jose	Rodriguez	35600.00
2	12345678N	Calle	Juan	Perez	25000.00
3	34567890G	Avenida	Raul	Ruiz	123900.00
4	12345678H	Calle	Sonia	Diaz	250000.00
5	42345378N	Avenida	Rocio	Perez	15675.00
6	76434457J	Calle	Daniel	Anaya	125900.00
7	34563476L	Calle	Inmaculada	Maldonado	47000.00
8	98654567N	Avenida	Antonio	Ortiz	35750.00
9	42345678T	Calle	Jorge	Pascual	57600.00
10	18674543P	Calle	Enrique	Sanchez	245900.00

## 2. Cuestiones a resolver:

**2.1** Indica la consulta SQL para obtener el nombre y apellidos de todos los clientes cuyo DNI termine con la letra N. Ordena el resultado por el DNI de manera descendente.

```

SELECT first_name, last_name
FROM customer
WHERE dni LIKE '%N'
ORDER BY dni DESC;

```

```

1 • USE empresa;
2 • SELECT * FROM customer;
3 • SELECT first_name, last_name
4 FROM customer
5 WHERE dni LIKE '%N'
6 ORDER BY dni DESC;

```

first_name	last_name
Antonio	Ortiz
Rocio	Perez
Juan	Perez

## BASES DE DATOS - UD4 CASO 1

# CONSULTAS EN SQL

2.2 Indica la consulta SQL para obtener el nombre y DNI de todos los clientes cuyo apellido sea “Pérez”. Ordena el resultado por el campo Nombre.

```
SELECT first_name, dni
FROM customer
WHERE last_name = 'Perez'
ORDER BY first_name;
```

```
7 • SELECT first_name, dni
8     FROM customer
9     WHERE last_name = 'Perez'
10    ORDER BY first_name;
11
```

Result Grid			Filter Rows:
	first_name	dni	
▶	Juan	12345678N	
	Rocio	42345378N	

2.3 Indica la consulta SQL para obtener el DNI de todos los clientes cuya dirección contenga la palabra “Avenida”, en cualquier posición del campo.

```
SELECT dni
FROM customer
WHERE address LIKE '%Avenida%';
```

```
11 • SELECT dni
12     FROM customer
13     WHERE address LIKE '%Avenida%';
14
```

Result Grid		Filter Rows:
	dni	
▶	34567890G	
	42345378N	
	98654567N	

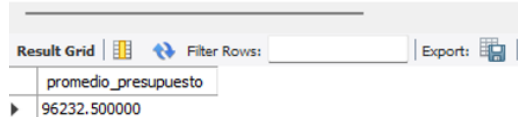
## BASES DE DATOS - UD4 CASO 1

# CONSULTAS EN SQL

2.4 Indica la consulta SQL necesaria para obtener la media de los presupuestos (Budget).

```
SELECT AVG(Budget) AS promedio_presupuesto
FROM customer;
```

```
14 • SELECT AVG(Budget) AS promedio_presupuesto
15 FROM customer;
16
```



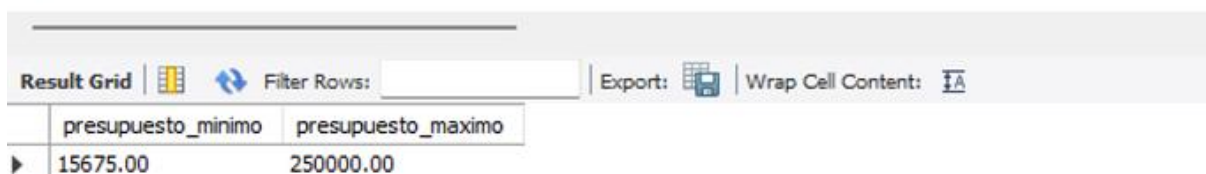
The screenshot shows a SQL query result in a table with one column named 'promedio\_presupuesto' and one row with the value '96232.500000'. The interface includes a 'Result Grid' tab, a 'Filter Rows' input, and an 'Export' button.

promedio_presupuesto
96232.500000

2.5 Indica la consulta SQL que devuelva el presupuesto mínimo y el presupuesto máximo.

```
SELECT MIN(Budget) AS presupuesto_minimo, MAX(Budget) AS presupuesto_maximo
FROM customer;
```

```
16 • SELECT MIN(Budget) AS presupuesto_minimo, MAX(Budget) AS presupuesto_maximo
17 FROM customer;
18
```



The screenshot shows a SQL query result in a table with two columns: 'presupuesto\_minimo' and 'presupuesto\_maximo'. The first row contains the values '15675.00' and '250000.00'. The interface includes a 'Result Grid' tab, a 'Filter Rows' input, an 'Export' button, and a 'Wrap Cell Content' checkbox.

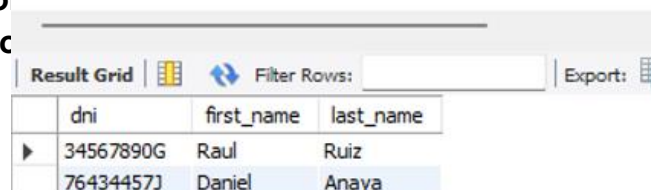
presupuesto_minimo	presupuesto_maximo
15675.00	250000.00

2.6 Indica la consulta SQL que devuelva el DNI, Nombre y Apellido de los clientes con un presupuesto entre 100000 y 200000.

```
SELECT dni, first_name, last_name
FROM customer
WHERE Budget BETWEEN 100000 AND 200000;
```

```
18 • SELECT dni, first_name, last_name
19 FROM customer
20 WHERE Budget BETWEEN 100000 AND 200000;
21
```

2.7 Indica la consulta SQL necesaria para obtener el nombre y Dirección para todos aquellos clientes cuyo presupuesto sea mayor a 100000.



The screenshot shows a SQL query result in a table with three columns: 'dni', 'first\_name', and 'last\_name'. The first two rows contain the values '34567890G', 'Raul', 'Ruiz' and '76434457J', 'Daniel', 'Anaya'. The interface includes a 'Result Grid' tab, a 'Filter Rows' input, and an 'Export' button.

dni	first_name	last_name
34567890G	Raul	Ruiz
76434457J	Daniel	Anaya

## BASES DE DATOS - UD4 CASO 1

# CONSULTAS EN SQL

```
SELECT dni, first_name, address  
FROM customer  
WHERE dni LIKE '4%';
```

```
21 • SELECT dni, first_name, address  
22 FROM customer  
23 WHERE dni LIKE '4%';  
24
```

Result Grid			
Filter Rows: <input type="text"/>			
	dni	first_name	address
▶	45679823F	Jose	Calle
	42345378N	Rocio	Avenida
	42345678T	Jorge	Calle

**2.8 Indica la instrucción SQL necesaria para reducir en 150€ el presupuesto (budget) de los clientes cuyo DNI sea 45679823F o 34567890G.**

Antes de la actualización:

```
SELECT dni, Budget  
FROM customer  
WHERE dni IN ('45679823F', '34567890G');
```

```
24 • SET SQL_SAFE_UPDATES = 0;  
25 • SELECT dni, Budget  
26 FROM customer  
27 WHERE dni IN ('45679823F', '34567890G');  
28
```

Result Grid		
Filter Rows: <input type="text"/>		
Export:		
	dni	Budget
▶	45679823F	35450.00
	34567890G	123750.00

Actualización:

```
UPDATE customer  
SET Budget = Budget - 150  
WHERE dni IN ('45679823F', '34567890G');
```

Se encontró que la actualización no se podía realizar debido a estar MySQL en modo

## BASES DE DATOS - UD4 CASO 1

# CONSULTAS EN SQL

Se realizó una consulta para verificar la consulta adecuada para desactivar esta opción:

```
SET SQL_SAFE_UPDATES = 0;
```

Una vez desactivado el modo seguro, se ejecutó la actualización con éxito.

```
24 • SET SQL_SAFE_UPDATES = 0;
25 • UPDATE customer
26   SET Budget = Budget - 150
27   WHERE dni IN ('45679823F', '34567890G');
```

46 16:07:40 SET SQL\_SAFE\_UPDATES = 0 0 row(s) affected  
47 16:08:53 UPDATE customer SET Budget = Budget - 150 WHERE dni IN ('45679823F', '34567890G') 2 row(s) affected Rows matched: 2 Changed: 2 Warnings: 0

## Después de la actualización:

```
SELECT dni, Budget
FROM customer
WHERE dni IN ('45679823F', '34567890G');
```

```
3 • SELECT dni, Budget
4   FROM customer
5   WHERE dni IN ('45679823F', '34567890G');
6
```

Result Grid | Filter Rows: | Export:

dni	Budget
45679823F	35300.00
34567890G	123600.00

**2.9 Indica la instrucción SQL necesaria para eliminar los clientes cuyo DNI termine por N y su presupuesto sea superior a 50.000.**

## Antes de la eliminación:

```
SELECT *
FROM customer
WHERE dni LIKE '%N' AND Budget > 50000;
```

El resultado de la consulta devuelve valor NULL.

## BASES DE DATOS - UD4 CASO 1

## CONSULTAS EN SQL

```

6 • SELECT *
7 FROM customer
8 WHERE dni LIKE '%N' AND Budget > 50000;
9

```

Durante la realización de e  
simultáneamente con amb  
datos, se concluyó que la a  
datos actuales.

customer_id	dni	address	first_name	last_name	Budget
*	NULL	NULL	NULL	NULL	NULL

en registros que cumplan  
verificaciones y análisis de  
a de coincidencias en los

```

SELECT *
FROM customer
WHERE NOT (dni LIKE '%N' AND Budget > 50000);

```

```

1 • SELECT *
2 FROM customer
3 WHERE NOT (dni LIKE '%N' AND Budget > 50000);
4
5

```

customer_id	dni	address	first_name	last_name	Budget
1	45679823F	Calle	Jose	Rodriguez	35600.00
2	12345678N	Calle	Juan	Perez	25000.00
3	34567890G	Avenida	Raul	Ruiz	123900.00
4	12345678H	Calle	Sonia	Diaz	250000.00
5	42345378N	Avenida	Rocio	Perez	15675.00
6	76434457J	Calle	Daniel	Anaya	125900.00
7	34563476L	Calle	Inmaculada	Maldonado	47000.00
8	98654567N	Avenida	Antonio	Ortiz	35750.00
9	42345678T	Calle	Jorge	Pascual	57600.00
10	18674543P	Calle	Enrique	Sanchez	245900.00
*	NULL	NULL	NULL	NULL	NULL

Se realizaron pruebas adicionales para validar la integridad de los datos, confirmando que la lógica de la consulta es correcta, pero no hay registros que cumplan con los criterios establecidos. Esto se pudo verificar haciendo una consulta opuesta a la inicial, que devolvió la tabla entera.

```

SELECT *
FROM customer
WHERE dni NOT LIKE '%N' AND Budget > 50000;

```



## BASES DE DATOS - UD4 CASO 1

## CONSULTAS EN SQL

```

1 • SELECT *
2 FROM customer
3 WHERE dni NOT LIKE '%N' AND Budget > 50000;
4
5
6

```

customer_id	dni	address	first_name	last_name	Budget
3	34567890G	Avenida	Raul	Ruiz	123900.00
4	12345678H	Calle	Sonia	Diaz	250000.00
6	76434457J	Calle	Daniel	Anaya	125900.00
9	42345678T	Calle	Jorge	Pascual	57600.00
10	18674543P	Calle	Enrique	Sanchez	245900.00
* NULL	NULL	NULL	NULL	NULL	NULL

Eliminación:

Para finalizar el ejercicio haciendo demostración de que se ha comprendido el ejercicio y funcionamiento de la instrucción, se ha decidido hacer la operación contraria. Se identificarán y eliminarán los clientes cuyo DNI NO termine en N y su presupuesto sea superior a 50.000.

```

DELETE FROM customer
WHERE dni NOT LIKE '%N' AND Budget > 50000;

```

46 20:00:58 DELETE FROM customer WHERE dni NOT LIKE '%N' AND Budget > 50000

5 row(s) affected

Después de la eliminación:

```

SELECT *
FROM customer

```

```

1 • SELECT *
2 FROM customer
3
4
5
6

```

customer_id	dni	address	first_name	last_name	Budget
1	45679823F	Calle	Jose	Rodriguez	35600.00
2	12345678N	Calle	Juan	Perez	25000.00
5	42345378N	Avenida	Rocio	Perez	15675.00
7	34563476L	Calle	Inmaculada	Maldonado	47000.00
8	98654567N	Avenida	Antonio	Ortiz	35750.00
11	99999999N	Calle Test	Test	Prueba	60000.00
* NULL	NULL	NULL	NULL	NULL	NULL

## BASES DE DATOS - UD4 CASO 1

# CONSULTAS EN SQL

## 3. Lecciones Aprendidas

A lo largo del desarrollo de este caso práctico, se han adquirido varios aprendizajes significativos relacionados con la gestión de bases de datos en MySQL:

- **Importancia de la Precisión en las Consultas:**  
La correcta formulación de sentencias SQL es esencial para obtener resultados precisos. Pequeños errores en la sintaxis o en las condiciones del **WHERE** pueden generar resultados inesperados.
- **Gestión de Errores:**  
El análisis de errores, como el enfrentado en la Consulta 9, permitió comprender la necesidad de revisar tanto la lógica de las consultas como la integridad de los datos. La capacidad de depurar consultas es fundamental en entornos reales.
- **Validación de Datos:**  
Se destacó la importancia de validar los datos antes de ejecutar operaciones críticas, asegurando que las condiciones establecidas en las consultas se cumplan correctamente.
- **Resiliencia en el Proceso de Aprendizaje:**  
Enfrentar desafíos técnicos fomenta el pensamiento crítico y fortalece la capacidad para resolver problemas complejos en bases de datos.

## 4. Conclusiones

A través de este caso práctico se ha consolidado el conocimiento en la gestión de bases de datos con MySQL, realizando operaciones básicas de consulta, actualización y eliminación de registros de forma eficiente.

Se ha aprendido a:

- Diseñar y estructurar bases de datos relacionales de manera efectiva.
- Aplicar consultas SQL avanzadas para la extracción y manipulación de datos.
- Resolver errores técnicos mediante un enfoque analítico y estructurado.
- Modificar y eliminar registros de forma segura.
- Comprender el impacto de la calidad de los datos en la obtención de resultados precisos.
- Identificar y analizar casos en los que las consultas no devuelven resultados debido a la naturaleza de los datos.

**BASES DE DATOS - UD4 CASO 1**

# CONSULTAS EN SQL

- Reflexionar sobre la importancia de la validación de datos en el ciclo de vida de una base de datos.

Además, el proceso permitió mejorar la capacidad de análisis crítico y la habilidad para documentar procedimientos técnicos de manera clara y concisa.

## 5. Anexos

- Lecciones de la unidad
- [Stack Overflow](#)
- [Youtube](#)