

# Tema 6. DML – DQL. Consultas

---

# Contenido

---

- Introducción. Tabla DUAL. Lenguaje DQL
- SELECT. Sintaxis
- Consultas simples
- Funciones
- Agrupaciones
- Composiciones (JOIN)
- Combinaciones especiales
- Subconsultas
- DQL en DML

# Tabla DUAL

---

La tabla DUAL es una tabla especial comodín de una sola columna predeterminada en todas las de bases de datos.

Su objetivo es poder realizar consultas u obtener datos en los cuales no se realiza una consulta a una tabla.

Ejemplo: `select 1+1 from dual;`

`select sysdate() from dual;`

# DQL. Data Query Language (DQL)

---

- Empezaremos por estudiar la sentencia SELECT, que permite recuperar datos de una o varias tablas.
- La sentencia SELECT es con mucho la más compleja y potente de las sentencias SQL. Empezaremos por ver las consultas más simples, basadas en una sola tabla. Esta sentencia forma parte del DML (lenguaje de manipulación de datos), en esta UF veremos cómo seleccionar columnas de una tabla, cómo seleccionar filas y cómo obtener las filas ordenadas por el criterio que queramos. El resultado de la consulta es una tabla lógica, porque no se guarda en el disco sino que está en memoria y cada vez que ejecutamos la consulta se vuelve a calcular. Cuando ejecutamos la consulta se visualiza el resultado en forma de tabla con columnas y filas, pues en la SELECT tenemos que indicar qué columnas queremos que tenga el resultado y qué filas queremos seleccionar de la tabla origen.

# DQL. Data Query Language (DQL)

---

- Empezaremos por estudiar la sentencia SELECT, que permite recuperar datos de una o varias tablas. La sentencia SELECT es con mucho la más compleja y potente de las sentencias SQL. Empezaremos por ver las consultas más simples, basadas en una sola tabla.
- Esta sentencia forma parte del **DML** (lenguaje de manipulación de datos), en esta UF veremos cómo seleccionar columnas de una tabla, cómo seleccionar filas y cómo obtener las filas ordenadas por el criterio que queramos.
- El resultado de la consulta es una tabla lógica, porque no se guarda en el disco sino que está en memoria y cada vez que ejecutamos la consulta se vuelve a calcular.
- Cuando ejecutamos la consulta se visualiza el resultado en forma de tabla con columnas y filas, pues en la SELECT tenemos que indicar qué columnas queremos que tenga el resultado y qué filas queremos seleccionar de la tabla origen.

# DQL. Data Query Language (DQL)

---

- ❑ Lenguaje de consulta de datos

- ❑ **SELECT** comando de DQL:

- ❑ Conseguir datos de ciertas columnas de tablas (proyectar)

- ❑ Obtener información de filas (registros) de una tabla – Se pueden añadir condiciones

- ❑ Obtener información de múltiples tablas según se relacionen: JOIN

- ❑ Operaciones y cálculos sobre datos

- ❑ Agrupaciones

- ❑ Condiciones y filtros

# SELECT. Sintaxis

---

```
SELECT * | {[DISTINCT] columna | expresión [[AS] alias], ...}  
FROM tabla  
[WHERE condición]  
[GROUP BY <lista_campos>  
          [HAVING <condición_de_agrupación>]]  
[ORDER BY <lista_campos> [DESC|ASC];];
```

# SELECT. Sintaxis

---

- \* El asterisco referencia todas las columnas de tabla. Se puede indicar \* o una lista de selección de columnas. Se puede indicar nombre de columna simple o cualificado (nombreTabla.nombreColumna), en algunos casos que se verán más adelante es obligatorio
- FROM. Tabla a la que se le quiere hacer la consulta
- DISTINCT: Distintos, no muestra información duplicada
- Expresión: expresión válida para SQL
- Alias (AS): nombre que se mostrará la columna en el resultado de la instrucción. También usado para trabajar con consultas multitabla para referenciar a las tablas dentro de la propia consulta
- Tabla. Tabla (s) a consultar
- Where. Condiciones y filtros
- Order By. Agrupaciones. Having condiciones sobre agrupaciones. ASC/DESC: forma de ordenar el resultado (numérico, orden alfabético, etc). ORDER BY col1, col2, etc.



# SELECT. Sintaxis

---

SELECT	[ALL   DISTINCT] [columna1, ... , <u>columnaN</u>   * ]	¿QUÉ MOSTRAR?	ALIAS	FUNCIONES
FROM	[ <u>tabla1</u> , ... , <u>tablaN</u> ]	¿DÓNDE BUSCAR?	TABLAS	
WHERE	condición	¿CONDICIÓN A CUMPLIR?		OPERADORES
ORDER BY	criterios[DESC   ASC];	¿ORDENADO?		

# SELECT. Sintaxis. Ejemplos

---

Muestra el nombre y el código postal de los clientes que viven en México

```
select contactName, postalCode from customer where Country = 'México'
```

Estadísticas 1	Estadísticas 2	customer 3 ×
----------------	----------------	--------------

contactName, postalCode from c *Enter a SQL expression to filter results (use Ctrl+Space)*

ABC contactName	123 postalCode
Ana Trujillo	5.021
Antonio Moreno	5.023

# SELECT. Sintaxis. Ejemplos

---

Muestra todos los valores de la columna país (country)

```
select Country from customer
```

Estadísticas 1 Estadísticas 2 customer 3 ×

Country from customer Enter a SQL expression to filter results (use Ctrl+Space)

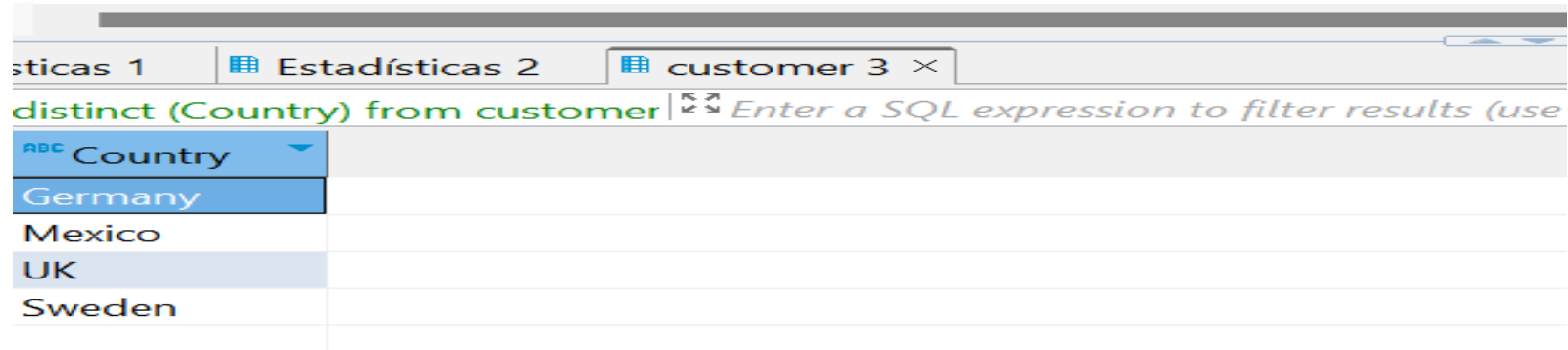
Country  
Germany  
Mexico  
Mexico  
UK  
Sweden

# SELECT. Sintaxis. Ejemplos

---

Muestra los distintos países que hay en la columna country

```
select distinct (Country) from customer
```



# SELECT. Sintaxis. Ejemplos

---

Muestra el nombre y el ID de los clientes cuyo país sea Mexico ordenados por número de ID de mayor a menor

```
select contactName, customerID from customer where country = 'México' order by customerID desc
```

Estadísticas 1	Estadísticas 2	customer 3 ×
contactName, customerID from   Enter a SQL expression to filter results (use Ctrl+Space)		
contactName	customerID	
Antonio Moreno	3	
Ana Trujillo	2	

# SELECT. Sintaxis. Ejemplos

---

Muestra el nombre y el ID de los clientes cuyo país sea Mexico ordenados por número de ID de mayor a menor

```
select contactName, customerID from customer where country = 'México' order by customerID desc
```

Estadísticas 1	Estadísticas 2	customer 3 ×
contactName, customerID from   Enter a SQL expression to filter results (use Ctrl+Space)		
contactName	customerID	
Antonio Moreno	3	
Ana Trujillo	2	

# Consultas. Columnas calculadas. Operadores

---

Además de las columnas que provienen directamente de la tabla origen, una consulta SQL puede incluir columnas calculadas cuyos valores se calculan a partir de los datos almacenados.

Para solicitar una columna calculada, se especifica en la lista de selección una expresión en vez de un nombre de columna

La expresión puede contener sumas, restas, multiplicaciones y divisiones, concatenación, paréntesis y funciones predefinidas: MAX, AVG, SUBSTRING LOWER, UPPER, etc.

# Consultas. Operadores Aritméticos

---

Se pueden utilizar los cuatro operadores aritméticos para hacer cálculos en las consultas. Cuando se utilizan como expresión en una consulta SELECT, no modifican los datos originales.

Precedencia:

1. Más prioridad la multiplicación y división, después la suma y la resta.
2. En caso de igualdad, se realiza primero la operación más a la izquierda.
3. El interior de los paréntesis es lo que se ejecuta primero.
4. Cualquier cálculo sobre valores NULL, el resultado es siempre NULL.



# Consultas. Operadores lógicos y de comparación

---

<b>NOT</b>	TRUE si la condición es falsa
<b>AND</b>	TRUE si las dos condiciones son verdaderas
<b>OR</b>	TRUE si una de las dos condiciones es verdadera

<b>=</b>	Igual
<b>!= &lt;&gt; ^=</b>	Distinto
<b>&lt;</b>	Menor que
<b>&lt;=</b>	Menor o igual que
<b>&gt;</b>	Mayor que
<b>&gt;=</b>	Mayor o igual que

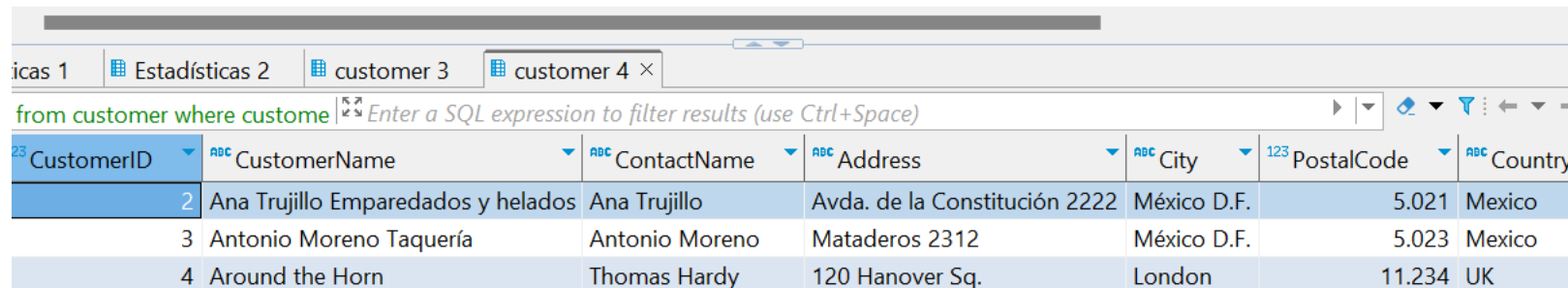
# Consultas. Operadores de comparación: BETWEEN

BETWEEN. Comprueba si un valor está dentro de un rango de valores, desde un valor inicial a un valor final (AMBOS EXTREMOS INCLUIDOS).

Se puede negar la comprobación con NOT. Comprobar si un valor no está dentro del rango NOT BETWEEN

Ejemplo: Muestra los datos de los clientes cuyo ID esté entre el 2 y el 4

```
select * from customer where customerID between 2 and 4
```



The screenshot shows a database query interface with a tab labeled 'customer 4'. The query entered is 'from customer where custome'. Below the query bar, a table displays the results of the query. The table has seven columns: CustomerID, CustomerName, ContactName, Address, City, PostalCode, and Country. The results show three rows of data, corresponding to customer IDs 2, 3, and 4.

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	5.021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	5.023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	11.234	UK

# Consultas. Operadores de comparación: LIKE

---

LIKE. Permite usar caracteres especiales en comparación:

- % representa cualquier cadena de 0 o más caracteres
- \_ representa un solo carácter

**WHERE columna [NOT] LIKE 'plantilla'**

<b>LIKE 'Director'</b>	Cadena 'Director'
<b>LIKE 'D%'</b>	Cualquier cadena que empiece por 'D'
<b>LIKE '%S%'</b>	Cualquier cadena que contenga una 'S'
<b>LIKE '__M'</b>	Cualquier cadena de tres caracteres que termine por 'M'
<b>LIKE 'N_'</b>	Cualquier cadena de dos caracteres que empiece por 'N'
<b>LIKE '_R%'</b>	Cualquier cadena cuyo segundo carácter sea 'R'

# Consultas. Operadores de comparación: IN

---

IN. Comprueba si un valor pertenece a un conjunto.

De la misma forma que en casos anteriores se puede negar con NOT

**<expresión> [NOT] IN (valores separados por comas)**

# Consultas. Operadores de comparación: IS NULL

Las comprobaciones a valores NULL no se pueden hacer con comparadores de igualdad, menor o mayor se deben hacer con IS NULL

De la misma forma que en casos anteriores se puede negar con NOT

**<expresión> IS [NOT] NULL**

# Consultas. Funciones

---

Las funciones permiten realizar cálculos y obtener valores útiles para el usuario

Las funciones se usan dentro de expresiones y actúan con los valores de las columnas

Las funciones producen diferentes resultados:

- Modificación de la información original. Por ejemplo, poner en mayúsculas toda una cadena de texto. OjO: No se modifica el dato almacenado (es persistente) tan sólo al consultarlo y traerlo en la consulta
- Información sobre el contenido. Por ejemplo número de caracteres de una palabra

# Consultas. Funciones

---

- Funciones aritméticas (columnas, filas)
- Funciones de listas
- Funciones sobre caracteres
- Funciones de fechas
- Funciones con valores nulos
- Funciones de conversión
- Funciones condicionales

# Consultas. Funciones de fila: algunos ejemplos

---

<b>ABS(n)</b>	Valor absoluto
<b>CEIL(n)</b>	Entero inmediatamente superior o igual
<b>FLOOR(n)</b>	Entero inmediatamente inferior o igual
<b>MOD(m,n)</b>	Resto de la división de m entre n
<b>POWER(m,exp)</b>	Valor de m elevado a exp
<b>ROUND(n,m)</b>	Valor de n redondeado a m decimales
<b>TRUNCATE(n,m)</b>	Valor de n truncado a m decimales

# Consulta. Funciones. Algunas funciones de fila

```
SELECT ROUND(2.586,1), ROUND(3.45), ROUND(4.1289,2), ROUND(8.1474,3) FROM DUAL;
```

icas 1	Estadísticas 2	customer 3	customer 4	Resultados 5 ×
ROUND(2.586,1), ROUND(3.45), Enter a SQL expression to filter results (use Ctrl+Space)				
ROUND(2.586,1)	ROUND(3.45)	ROUND(4.1289,2)	ROUND(8.1474,3)	
2,6	3	4,13	8,147	

```
SELECT TRUNCATE(2.586,1), TRUNCATE(4.1289,2), TRUNCATE(8.1474, 3) FROM DUAL;
```

cas 1	Estadísticas 2	customer 3	customer 4	Resultados 5 ×
TRUNCATE(2.586,1), TRUNCATE(4.1289,2), Enter a SQL expression to filter results (use Ctrl+Space)				
TRUNCATE(2.586,1)	TRUNCATE(4.1289,2)	TRUNCATE(8.1474, 3)		
2,5	4,12	8,147		



# Consulta. Funciones de columna: algunos ejemplos

---

Son funciones que actúan sobre un grupo de filas para obtener un valor. Los valores nulos son ignorados por las funciones de grupos de valores y los cálculos se realizan sin contar con ellos

<b>AVG(n)</b>	Valor medio sin tomar en cuenta los nulos
<b>COUNT(*   expresión)</b>	Número de datos que cumple la expresión
<b>MAX(expresión)</b>	Valor máximo de la expresión
<b>MIN(expresión)</b>	Valor mínimo de la expresión
<b>SUM(expresión)</b>	Suma los valores de la expresión no nulos
<b>VARIANCE(expresión)</b>	Varianza de los valores de expresión no nulos
<b>STDDEV(expresión)</b>	Desviación estándar de los valores de expresión no nulos

# Consultas. Funciones de columnas.

## Ejemplos

---

- Número de empleados: `select count(*) from emple`

- Fecha de incorporación más antigua: `select min(fecha_alt) from emple`

- Cuantos empleados tienen comisión:

```
select count(comision) from emple
```

```
select count(*) from emple where comision is not null
```

- Cuantos empleados no tienen comisión:

```
select count(*) from emple where comision is null
```

# Consultas. Funciones de listas

---

- Trabajan sobre un grupo de columnas dentro de la misma fila
- Algunos casos:
  - GREATEST (val1, val2,...). Mayor valor NO NULL de la lista. Si hay algún elemento NULL, el resultado es NULL
  - LEAST (val1, val2,...). Menor valor NO NULL de la lista. Si hay algún elemento NULL, el resultado es NULL

# Consultas. Funciones de caracteres y cadenas de texto, algunos ejemplos

---

<b>LENGTH(<u>cad</u>)</b>	Número de caracteres
<b>CONCAT (cad1, cad2)</b>	Cad1 y cad2 concatenadas (equivalente al operador   )
<b>LOWER(<u>cad</u>)</b>	Convierte a minúsculas
<b>UPPER(<u>cad</u>)</b>	Convierte a mayúsculas
<b>LPAD (cad1, n [, cad2])</b>	Añade caracteres a la izquierda de cad1. Si se suprime cad2 se asume relleno con blanco
<b>RPAD (cad1, n [, cad2]).</b>	Añade caracteres a la derecha de cad1. Si se suprime cad2 se asume relleno con blanco

Nota sobre LPAD: si  $n < \text{str}$ , devuelve str recortado a n sin añadir el segundo parámetro. Igual para RPAD

# Consultas. Funciones de caracteres y cadenas de texto, algunos ejemplos

---

- LTRIM (str). Devuelve la cadena de texto con los caracteres en blanco omitidos por la izquierda
- RTRIM (str). Devuelve la cadena de texto con los caracteres en blanco omitidos por la derecha
- TRIM (str). Devuelve la cadena de texto con los caracteres omitidos según el patrón indicado, al principio de la cadena -izquierda- (LEADING), al final -derecha- ( TRAILING) o en ambos extremos (BOTH)
- REPLACE(str, from str, to stre).. 3 parámetros de entrada la cadena con la que trabajar, el valor que se quiere sustituir -reemplazar- el valor sustituto
- SUBSTR(). Admite distintos valores

# Consultas. Funciones de fecha, algunos ejemplos

---

<b>SYSDATE</b>	Obtiene la fecha y hora actuales.
<b>CURRENT_TIMESTAMP</b>	Obtiene la fecha y hora actuales en formato TIMESTAMP.
<b>DATE_ADD(fecha, INTERVAL n unidad)</b>	Devuelve la fecha incrementada en n intervalos.
<b>LAST_DAY(fecha)</b>	Devuelve la fecha del último día del mes que contiene fecha.
<b>NOW()</b>	Hoy
<b>MONTHNAME(fecha)</b>	Devuelve el nombre del mes
<b>EXTRACT(valor FROM fecha)</b>	Extrae un valor de una fecha concreta. El valor puede ser DAY (día), MONTH (mes), YEAR (año), etc.

# Consultas. Funciones de fecha, algunos ejemplos

---

```
select sysdate()
```

```
select current_timestamp()
```

```
select date_add('2023-01-01', interval 3 DAY)
```

```
select LAST_DAY('2023-01-01')
```

```
select NOW()
```

```
select monthname('2023-01-01')
```

```
select EXTRACT(day from '2023-01-01')
```

# Consultas. Funciones con valores nulos.

## Algunos ejemplos:

---

- NULLIF (val1, val2). Devuelve NULL si val1 es igual a val2, sino devuelve val1
- IFNULL(val1, val2)
- COALESCE(listaExpr). Devuelve la primera expresión que no es nula:

```
SELECT SAL, COMM, COALESCE (SAL+COMM, SAL) FROM EMP WHERE DEPTNO=400;
```



# Consultas. Funciones de conversión

---

Convierte datos para que el resultado de una expresión tenga sentido, por ejemplo, se pasa de texto a número o de número a texto, de texto a fecha y viceversa

➤ CAST, admite multitud de valores para convertir de un tipo a otro, algunos ejemplos:

➤ **SELECT CAST('123' AS UNSIGNED) AS MIENTERO**

➤ **select cast('123.456' as decimal(6,3)) as numerodecimal**

# Consultas. Funciones condicionales

➤ CASE: Simula la estructura del tipo Switch que implementa los bucles condicionales anidados:

```
SELECT apellido, oficio,  
CASE OFICIO  
WHEN 'DIRECTOR' THEN 'JEFAZO'  
WHEN 'VENDEDOR' THEN 'JEFECILLO'  
ELSE 'CURRITO'  
END AS CONDICION  
FROM EMPLE
```

ABC	apellido	ABC	oficio	ABC	CONDICION	
	SANCHEZ		EMPLEADO		CURRITO	
	ARROYO		VENDEDOR		JEFECILLO	
	SALA		VENDEDOR		JEFECILLO	
	JIMENEZ		DIRECTOR		JEFAZO	
	MARTIN		VENDEDOR		JEFECILLO	

# Consultas. Anidamientos

---

Las consultas se pueden anidar, por ejemplo en la tabla "EMPLE" todos los apellidos están en mayúsculas y se quiere que al devolverlos esté únicamente en mayúscula la primera letra, para ello:

```
select CONCAT(UPPER(LEFT(apellido,1)),lower(SUBSTR(apellido,2))) from  
emple
```

# Consultas. Web de referencia

---

➤ Funciones: <https://dev.mysql.com/doc/refman/8.0/en/functions.html>

➤ ¡MUY IMPORTANTE!