

Módulo 2

Código: 484

Bases de datos

Técnico Superior en Desarrollo de
Aplicaciones Multiplataforma



UNIDAD 6

DQL

Guía didáctica profesor



Sesiones 5 a 9. Contenidos teóricos

FUNCIONES DE GRUPO

Funciones que aplicadas en columnas devuelven una única línea con un solo valor para cada grupo.

FUNCIÓN	SIGNIFICADO
COUNT(*)	Cuenta los elementos de un grupo. Se utiliza el asterisco para no tener que indicar un nombre de columna concreto, el resultado es el mismo para cualquier columna.
SUM (expresión)	Suma los valores de la expresión.
AVG (expresión)	Calcula la media aritmética sobre la expresión indicada.
MIN (expresión)	Mínimo valor que toma la expresión indicada.
MAX (expresión)	Máximo valor que toma la expresión indicada.

Sesiones 5 a 9. Contenidos teóricos

FUNCIONES DE GRUPO

Ejemplo: Mostrar los precios más caro y más barato de todos los artículos.

```
SELECT MAX (precio), MIN(precio) FROM articulo;
```

Ejemplo: Mostrar el precio medio de aquellos artículos cuyo stock es inferior a las 25 unidades

```
SELECT AVG(precio) FROM articulo WHERE stock < 25;
```

Ejemplo: Escribir una sentencia que obtenga el stock más alto para aquellos artículos cuyo precio es inferior a 1.5 euros

```
SELECT MAX(stock) FROM articulo WHERE precio < 1.5;
```

Ejemplo: Mostrar el número de artículos.

```
SELECT COUNT(*) FROM articulo;
```

Sesiones 5 a 9. Contenidos teóricos

FUNCIONES DE GRUPO

Ejemplo: Mostrar la cantidad de artículos que tienen stock.

```
SELECT COUNT (stock) FROM articulo;
```

Ejemplo: Mostrar el precio medio de los artículos.

```
SELECT AVG (precio) FROM articulo;
```

Sesiones 5 a 9. Contenidos teóricos

ERRORES HABITUALES

Caso 1: ¿Cuál es la diferencia entre las siguientes dos sentencias?

```
SELECT COUNT (DISTINCT precio) FROM articulo;
```

```
SELECT DISTINCT COUNT (precio) FROM articulo;
```

- La primera sentencia calcula el número de precios distintos de los artículos.
- Mientras que la segunda calcula el número de artículos cuyos precios no son nulos y, una vez calculado este único valor, elimina los valores repetidos del resultado anterior obtenido. Como la función count devuelve un único valor, obviamente no puede haber repetidos y el modificador distinct no hace nada en este último caso.

Sesiones 5 a 9. Contenidos teóricos

ERRORES HABITUALES

Caso 2: ¿Cuál es la diferencia entre las siguientes dos sentencias?

```
SELECT COUNT (codigo) FROM articulo;
```

```
SELECT COUNT (*) FROM articulo;
```

- La primera sentencia cuenta el número de códigos de artículos no nulos en la tabla de artículos. Ahora bien, ¿pueden existir códigos nulos? En este caso no ya que *codigo* es la clave primaria. Así pues, lo que hace es contar el número de filas.
- La segunda sentencia realiza la misma labor, cuenta número de filas de la tabla de artículos. La única diferencia entre ambas suele ser el tiempo de ejecución. La mayor parte de los sistemas de gestión de bases de datos ejecutan mucho más eficientemente la segunda que la primera.

Sesiones 5 a 9. Contenidos teóricos

ERRORES HABITUALES

Caso 3: ¿Qué calcula la siguiente sentencia?

```
SELECT precio FROM articulo WHERE MAX(precio)=3 ;
```

No calcula nada pues es errónea. La restricción en el where se aplica siempre a cada fila. Ahora bien, la expresión existente (max(precio)) no puede aplicarse a cada fila pues es una operación que se aplica a toda una columna. De este caso se puede extraer una regla general muy importante:

Las funciones de grupo (agregación) no pueden estar nunca en la cláusula where.

Sesiones 5 a 9. Contenidos teóricos

AGRUPACIONES

Es muy común utilizar consultas en las que se desee agrupar los datos a fin de realizar cálculos sobre los grupos. Para ello se utiliza la cláusula GROUP BY.

```
SELECT listaDeExpresiones  
FROM listaDeTablas  
[WHERE condiciones]  
[GROUP BY grupos]  
[HAVING condicionesDeGrupo]  
[ORDER BY columnas];
```

Sesiones 5 a 9. Contenidos teóricos

AGRUPACIONES

En el apartado GROUP BY, se indican las columnas por las que se agrupa.

```
SELECT municipio FROM cliente GROUP BY municipio;
```

Esta consulta produciría un error:

```
SELECT municipio, cp FROM cliente GROUP BY municipio;
```

ERROR la columna cp debe aparecer en la expresión GROUP BY. cp no está disponible directamente ya que tiene valores distintos para los registros del mismo grupo.

Si utilizamos un GROUP BY, en la SELECT solo pueden aparecer los campos del GROUP BY y funciones de grupo.

Sesiones 5 a 9. Contenidos teóricos

AGRUPACIONES

Dada la siguiente tabla *Sales*:

Country	Region	Sales
Canadá	Alberta	100
Canadá	Columbia Británica	200
Canadá	Columbia Británica	300
Estados Unidos	Montana	100

Si ejecutamos la siguiente sentencia SQL:

```
SELECT Country, SUM(sales) AS TotalSales
```

```
FROM Sales
```

```
GROUP BY Country;
```

Se formarán 2 grupos: Canadá y Estados Unidos

Sesiones 5 a 9. Contenidos teóricos

AGRUPACIONES

Dada la siguiente tabla *Sales*:

Country	Region	Sales
Canadá	Alberta	100
Canadá	Columbia Británica	200
Canadá	Columbia Británica	300
Estados Unidos	Montana	100

Si ejecutamos la siguiente sentencia SQL:

```
SELECT Country, SUM(sales) AS TotalSales
```

```
FROM Sales
```

```
GROUP BY Country;
```

Se formarán 2 grupos: Canadá y Estados Unidos

Sesiones 5 a 9. Contenidos teóricos

AGRUPACIONES

```
SELECT Country, SUM(sales) AS TotalSales FROM Sales GROUP BY Country;
```

Country	Region	Sales
Canadá	Alberta	100
Canadá	Columbia Británica	200
Canadá	Columbia Británica	300
Estados Unidos	Montana	100

El resultado que se obtendrá estará compuesto de 2 filas en las que la primera columna será el país y la segunda la suma de ventas de las filas agrupadas.

Sesiones 5 a 9. Contenidos teóricos

AGRUPACIONES

```
SELECT Country, SUM(sales) AS TotalSales FROM Sales GROUP BY Country;
```

Country	TotalSales
Canadá	600
Estados Unidos	100

El resultado que se obtendrá estará compuesto de 2 filas en las que la primera columna será el país y la segunda la suma de ventas de las filas agrupadas.

Sesiones 5 a 9. Contenidos teóricos

AGRUPACIONES

Dada la siguiente tabla *Sales*:

Country	Region	Sales
Canadá	Alberta	100
Canadá	Columbia Británica	200
Canadá	Columbia Británica	300
Estados Unidos	Montana	100

Si ejecutamos la siguiente sentencia SQL:

```
SELECT Country, Region, SUM(sales) AS TotalSales  
FROM Sales  
GROUP BY Country, Region;
```

Se formarán 3 grupos según se indica en la tabla.

Sesiones 5 a 9. Contenidos teóricos

AGRUPACIONES

Dada la siguiente tabla *Sales*:

Country	Region	Sales
Canadá	Alberta	100
Canadá	Columbia Británica	200
Canadá	Columbia Británica	300
Estados Unidos	Montana	100

Si ejecutamos la siguiente sentencia SQL:

```
SELECT Country, Region, SUM(sales) AS TotalSales  
FROM Sales  
GROUP BY Country, Region;
```

Se formarán 3 grupos según se indica en la tabla.

Sesiones 5 a 9. Contenidos teóricos

AGRUPACIONES

```
SELECT Country, Region, SUM(sales) AS TotalSales FROM Sales GROUP BY Country, Region;
```

Country	Region	TotalSales
Canadá	Alberta	100
Canadá	Columbia Británica	500
Estados Unidos	Montana	100

Sesiones 5 a 9. Contenidos teóricos

AGRUPACIONES

FUNCIONES DE CÁLCULO: Lo interesante de la creación de grupos es las posibilidades de cálculo que ofrece utilizando las siguientes funciones.

```
SELECT articulo, SUM(precio*cant) FROM linea_ticket GROUP BY articulo;
```

FUNCIÓN	SIGNIFICADO
COUNT (*)	Cuenta los elementos de un grupo. Se utiliza el asterisco para no tener que indicar un nombre de columna concreto, el resultado es el mismo para cualquier columna.
SUM (expresión)	Suma los valores de la expresión.
AVG (expresión)	Calcula la media aritmética sobre la expresión indicada.
MIN (expresión)	Mínimo valor que toma la expresión indicada.
MAX (expresión)	Máximo valor que toma la expresión indicada.
STDDEV (expresión)	Calcula la desviación estándar.
VARIANCE (expresión)	Calcula la varianza.

Sesiones 5 a 9. Contenidos teóricos

AGRUPACIONES

Ejemplo: Mostrar el número de municipios para cada provincia.

```
SELECT provincia, COUNT(codigo) FROM municipio GROUP BY provincia;
```

Ejemplo: Mostrar el menor código de municipio para cada provincia.

```
SELECT provincia, MIN(codigo) FROM municipio GROUP BY provincia;
```

Sesiones 5 a 9. Contenidos teóricos

AGRUPACIONES POR VARIAS COLUMNAS

También se puede agrupar la información por varias columnas de modo que cada grupo se formará por aquellas filas que tienen el mismo valor en todos los campos que construyen el GROUP BY.

Ejemplo. Obtener la cantidad de tickets de cada cliente para cada tipo de IVA distinto.

```
SELECT cliente, iva, COUNT(*) FROM ticket GROUP BY cliente, iva;
```

Sesiones 5 a 9. Contenidos teóricos

AGRUPACIONES POR VARIAS COLUMNAS

Consulta Incorrecta: SELECT cliente, AVG(dto), iva FROM ticket GROUP BY cliente;

La sentencia es incorrecta dado que para cada cliente no existe un único iva sino varios los cuales no caben en una única línea. Este es un ejemplo de consulta incorrecta que devuelve muchos datos para cada grupo.

Sesiones 5 a 9. Contenidos teóricos

AGRUPACIONES POR VARIAS COLUMNAS

Consulta Incorrecta: Si se utilizan funciones de grupos y grupos, en la select solo pueden incluirse los campos del group by y las funciones de grupo (max, min, avg ...)

```
SELECT cliente, AVG(dto), iva FROM ticket GROUP BY cliente;
```

Sesiones 5 a 9. Contenidos teóricos

AGRUPACIONES - HAVING

Una consulta con agrupación admite dos tipos de restricciones:

- Las restricciones de fila van en la cláusula where y se aplican a cada fila
- Las restricciones de grupo van en la cláusula having y se aplican al grupo.

El having va después del group by.

Ejemplo. Calcula el descuento medio aplicado en los tickets cuyo código es mayor que 25 para cada cliente pero sólo si dicho descuento supera el 4 %.

```
SELECT cliente, AVG(dto)
FROM ticket
WHERE codigo > 25
GROUP BY cliente
HAVING AVG(dto) > 4;
```

Sesiones 5 a 9. Contenidos teóricos

AGRUPACIONES

Restricción de fila → WHERE

Restricción de grupo → HAVING

- Obtener tickets del año 2019.
- Obtener clientes con más de 2 tickets.
- Obtener tickets con código sea menor que 20.
- Obtener clientes cuyo IVA máximo sea 21.
- Obtener tickets con IVA 21.

Sesiones 5 a 9. Contenidos teóricos

AGRUPACIONES

Restricción de fila → WHERE

Restricción de grupo → HAVING

- Obtener tickets del año 2019: Restricción de fila.
- Obtener clientes con más de 2 tickets: Restricción de grupo.
- Obtener tickets con código sea menor que 20: Restricción de fila.
- Obtener clientes cuyo IVA máximo sea 21: Restricción de grupo.
- Obtener tickets con IVA 21: Restricción de fila.

Sesiones 5 a 9. Contenidos teóricos

AGRUPACIONES

- Si una condición puede ir en el WHERE y en el HAVING , es aconsejable ponerla en el WHERE porque el procesamiento de la consulta es más rápida.
- Si la restricción se aplica en el WHERE el sistema rápidamente elimina las filas. En caso contrario, debe procesar todas las filas, crear todos los grupos y sólo finalmente descartar algunos grupos.

Sesiones 5 a 9. Contenidos teóricos

AGRUPACIONES

Ejemplo. Para cada artículo, mostrar el descuento máximo aplicado en sus tickets y el número de unidades vendidas. Considérense sólo las líneas de los primeros 30 tickets. Sólo se deben mostrar aquellos artículos cuyo número de unidades vendidas supera 5 y cuyo código comienza por 'TOR'.

```
SELECT articulo, MAX(dto), SUM(cant)
  FROM linea_ticket
 WHERE ticket BETWEEN 1 AND 30 AND articulo LIKE 'TOR%'
 GROUP BY articulo HAVING SUM(cant) > 5;
```

Sesiones 5 a 9. Contenidos teóricos

AGRUPACIONES – Pasos a ejecutar en consultas con having

- Se toma la tabla linea_ticket. (**FROM**)
- Se dejan sólo las filas cuyo ticket se encuentra entre 1 y 30; y cuyo artículo comienza por 'TOR'. (**WHERE**)
- Se agrupan las filas en grupos según el artículo. (**GROUP BY**)
- Se aplican las restricciones de grupo, es decir, se dejan aquellos artículos cuyas ventas superan las 5 unidades. (**HAVING**)
- Para cada grupo se muestra el código del artículo, el descuento máximo y el total de unidades vendidas. (**SELECT**)

REGLAS A TENER EN CUENTA

- Todo lo que aparece en el SELECT y en el HAVING, o son funciones de grupo o están en el GROUP BY.
- Las funciones de grupo no pueden aparecer en el WHERE.
- Normalmente no hace falta agrupar si después (en las cláusulas HAVING o SELECT) no se utilizan funciones de grupo.