

Curso de Base de Datos

Sesión 6: SQL DML - Agrupaciones

2022-2

Semana 6 SQL DML



Profesor del curso:
César Aguilera
Luis Ríos



Elaborado por:
César Aguilera
Luis Ríos



Revisado por:
César Aguilera
Rony Cueva
Luis Ríos

Saberes previos

- SQL DML, lenguaje de manipulación de datos

SQL - DML

SELECT

GROUP BY

Agrupar filas seleccionadas en una sola con información agregada, basándose en la(s) expresión(es).

HAVING

Restringe qué filas obtenidas por el GROUP BY serán devueltas por la consulta.

UNION

UNION ALL

INTERSECT

MINUS

Combinan (de acuerdo a la correspondiente operación relacional) el resultado de dos consultas en una sola. La cantidad y tipos de datos de las columnas de las consultas deben ser iguales (aunque se permiten diferentes longitudes)

ORDER BY

Ordena las filas seleccionadas de la consulta.

FOR UPDATE

Bloquea (“lock”) las filas seleccionadas por la consulta hasta que finalice la transacción.

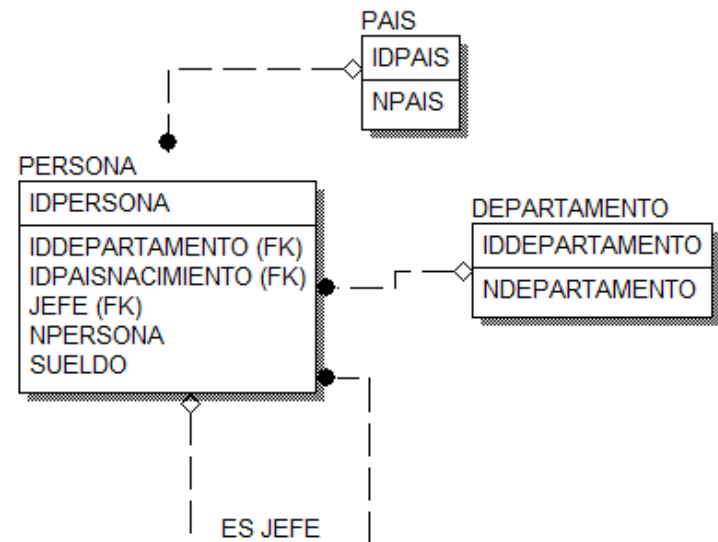
GROUP BY - FUNCIONES

9. Obtener el sueldo mínimo, promedio y máximo de cada departamento

```
SELECT      IdDepartamento, MIN(Sueldo), AVG(Sueldo),  
            MAX(Sueldo)
```

```
FROM  Persona
```

```
GROUP BY    IdDepartamento
```



Consultas sumariadas

Consultas agrupadas (cláusula GROUP BY)

Al igual que los subtotales son útiles en informes impresos, con frecuencia es conveniente sumarizar los resultados de la consulta a un nivel <<subtotal>>.

Cuál es el tamaño medio de un pedido?

```
SELECT AVG(IMPORTE) FROM PEDIDOS
```

AVG(IMPORTE)
\$8,256.37

\$8,256.37

¿Cuál es el tamaño medio de pedido para cada vendedor?

```
SELECT REP, AVG(IMPORTE) FROM PEDIDOS
```

REP	AVG(IMPORTE)
101	\$8,876.00
102	\$5,694.00
103	\$11,350.00

```
GROUP BY REP
```

Consultas sumarias

Consultas agrupadas (cláusula **GROUP BY**)

La primera consulta es una consulta sumaria simple como la de los ejemplos anteriores. La segunda consulta produce varias filas sumarias – una fila por cada grupo, sumalizando los pedidos aceptados por un solo vendedor.

SQL lleva a cabo la consulta del modo siguiente:

1. SQL divide los pedidos en grupos de pedidos, un grupo por cada vendedor. Dentro de cada grupo, todos los pedidos tienen el mismo valor en la columna REP.
2. Por cada grupo, SQL calcula el valor medio de la columna IMPORTE para todas las filas del grupo, y genera una única fila sumario de resultados. La fila contiene el valor de la columna REP del grupo y el tamaño de pedido medio calculado.

Consultas sumariadas

Consultas agrupadas (cláusula GROUP BY)

He aquí algunos ejemplos adicionales de consultas agrupadas.
¿Cuál es el rango de cuotas asignadas en cada oficina?

```
SELECT OFICINA_REP, MIN(CUOTA), MAX(CUOTA)
FROM REPVENTAS
GROUP BY OFICINA_REP
```

OFICINA_REP	MIN(CUOTA)	MAX(CUOTA)
-----	-----	-----
NULL	NULL	NULL
11	275,000.00	300,000.00
12	200,000.00	300,000.00
13	350,000.00	300,000.00
14	350,000.00	350,000.00

Consultas sumariadas

¿Cuántos clientes diferentes son atendidos por cada vendedor?

```
SELECT REP_CLIE, 'Clientes por rep. de venta' , COUNT(DISTINCT NUM_CLIE)  
FROM CLIENTES  
GROUP BY REP_CLIE
```

REP_CLIE	Clientes por rep. de venta	COUNT(DISTINCT NUM_CLIE)
-----	-----	-----
101	Clientes por rep. de venta	3
102	Clientes por rep. de venta	4
103	Clientes por rep. de venta	3
104	Clientes por rep. de venta	1
105	Clientes por rep. de venta	2

Cuando la cláusula **GROUP BY** está presente, informa a SQL que debe dividir los resultados detallados en grupos y aplicar la función de columna separadamente a cada grupo, produciendo un único resultado por cada grupo.

Consultas resumizadas

Múltiples columnas de agrupación

Se pueden agrupar resultados de consulta en base a contenidos de dos o más columnas.

Calcula los pedidos totales por cada vendedor (REP) y por cada cliente (CUST).

```
SELECT REP, CUST, SUM(IMPORTE) FROM PEDIDOS  
GROUP BY REP, CUST
```

REP	CUST	SUM(IMPORTE)
-----	-----	-----
101	2102	300,150.00
101	2108	20,130.00
101	2113	22,000.00
102	2106	350,000.00
102	2120	3,750.00
103	2111	4,589.00

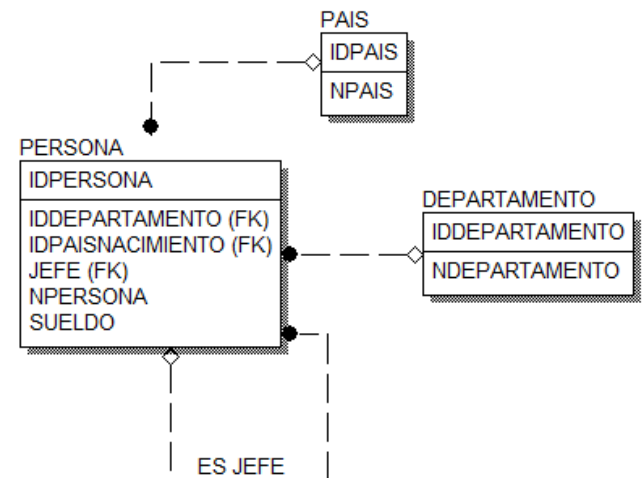
SQL - DML

SELECT - Usando Expresiones

Se desea hallar el impuesto total que se debe pagar por los sueldos, agrupados por departamento. El impuesto es el 10% del sueldo.

```
SELECT a.IdDepartamento||'-'||b.Ndepartamento Departamento, 0.1*(SUM(a.Sueldo))  
Impuesto  
FROM Persona a, Departamento b  
WHERE a.IdDepartamento = b.IdDepartamento  
GROUP BY a.IdDepartamento||'-'||b.Ndepartamento;
```

<u>Departamento</u>	<u>Impuesto</u>
001-Finanzas	1500
002-Marketing	2000
...	...



Consultas sumarias

Múltiples columnas de agrupación

Incluso con múltiples columnas de agrupación, SQL sólo proporciona un único nivel de agrupación. También puede ordenar los datos con la cláusula ORDER BY.

Calcula los pedidos totales para cada cliente de cada vendedor (REP), ordenados por cliente y dentro de cada cliente por vendedor.

```
SELECT CLIE, REP, SUM(IMPORTE) FROM PEDIDOS  
GROUP BY CLIE, REP  
ORDER BY CLIE, REP
```

CLIE	REP	SUM(IMPORTE)
-----	-----	-----
2101	106	10,150.00
2102	101	23,830.00
2103	105	35,754.00
2106	102	50,134.00

Consultas sumariadas

Condiciones de búsqueda de grupos(cláusula HAVING)

Al igual que la cláusula WHERE puede ser utilizada para seleccionar y rechazar filas individuales que participan en una consulta, la cláusula HAVING puede ser utilizada para seleccionar y rechazar grupos de filas.

La cláusula HAVING especifica por tanto una condición de búsqueda para grupos.

¿Cuál es el tamaño de pedido promedio para cada vendedor (REP) cuyos pedidos totalizan más de \$30,000?.

```
SELECT REP, AVG(IMPORTE)
FROM PEDIDOS
GROUP BY REP
HAVING SUM(IMPORTE) > 30,000
```

Consultas sumariadas

Condiciones de búsqueda de grupos(cláusula HAVING)

REP	AVG(IMPORTE)
-----	-----
105	8,150.00
101	16,430.00
107	11,754.00
102	50,134.00

La cláusula GROUP BY dispone primero de los pedidos en grupos por vendedor. La consulta HAVING elimina entonces los grupos en donde el total de los pedidos no excede a \$30,000.00.

Finalmente, la cláusula SELECT calcula el tamaño de pedido medio para cada uno de los grupos restantes y genera los resultados de la consulta.

Consultas sumarias

Condiciones de búsqueda de grupos(cláusula HAVING)

Por cada oficina con dos o más personas (vendedores), calcular la cuota total y las ventas totales para todos los vendedores (REP) que trabajan en la oficina.

```
SELECT CIUDAD, SUM(CUOTA), SUM(REPVENTAS.VENTAS)
FROM OFICINAS, REPVENTAS
WHERE OFICINAS.OFICINA = REPVENTAS.OFICINA_REP
GROUP BY CIUDAD
HAVING COUNT(*) > 2
```

CIUDAD	SUM(CUOTA)	SUM(REPVENTAS.VENTAS)
-----	-----	-----
Lima	\$800,150.00	\$735,451.00
Chiclayo	\$116,430.00	\$384,750.00
Trujillo	\$311,754.00	\$692,455.00

Consultas sumariadas

Condiciones de búsqueda de grupos (cláusula HAVING)

1. Compone las tablas OFICINAS y REPVENTAS para hallar la ciudad en donde trabaja cada vendedor.
2. Agrupa las filas resultantes por oficinas.
3. Elimina los grupos con dos o menos filas, éstas representan oficinas que no satisfacen el criterio de la cláusula HAVING.
4. Calcula la cuota total y las ventas totales para cada grupo.

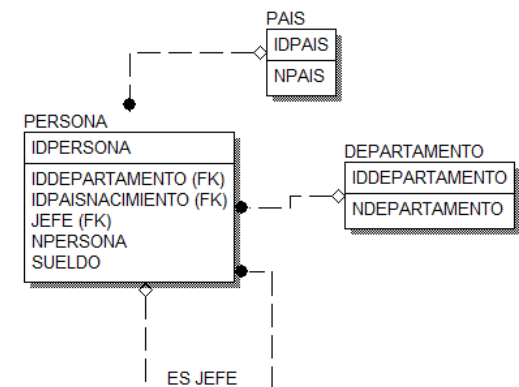
SELECT

- Obtener el sueldo promedio y total de sueldos de cada departamento (aunque no tenga empleados)

```
SELECT D.IdDepartamento, D.NDepartamento,
       AVG(P.Sueldo), SUM(P.Sueldo)
FROM   Departamento D, Persona P
WHERE  D.IdDepartamento (+) = P.IdDepartamento
GROUP BY      D.IdDepartamento, NDepartamento
ORDER BY      D.IdDepartamento
```

Función de columna

Outer join



SELECT

- Obtener el sueldo promedio y total de sueldos de cada departamento (aunque no tenga empleados)

```
SELECT D.IdDepartamento, D.NDepartamento,  
       AVG(P.Sueldo), SUM(P.Sueldo)  
FROM   Departamento D, Persona P  
WHERE  D.IdDepartamento (+) = P.IdDepartamento  
GROUP BY      D.IdDepartamento, NDepartamento  
ORDER BY      D.IdDepartamento
```

OUTER JOIN, fuerza a que aparezcan valores y completa con valores nulos.

El (+) se debe poner a continuación de la columna cuya tabla va a tener añadida la fila NULL imaginaria.

Consultas

Condición lateral (Outer Join)

Tabla Chicas

Nombre	Ciudad
María	Boston
Nancy	NULL
Susan	Chicago
Beatriz	Chicago
Ana	Denver

Tabla Chicos

Nombre	Ciudad
Juan	Boston
Henry	Boston
Jorge	NULL
Sam	Chicago
James	Dallas

- Lista las chicas y chicos de la misma ciudad y las chicas desparejadas.

Consultas

Condición lateral (Outer Join)

```
SELECT * FROM CHICAS A, CHICOS B  
WHERE A.CIUDAD = B.CIUDAD (+)
```

A.Nombre	A.Ciudad	B.Nombre	Ciudad
María	Boston	Juan	Boston
María	Boston	Henry	Boston
Susan	Chicago	Sam	Chicago
Beatriz	Chicago	Sam	Chicago
Ana	Denver	NULL	NULL
Nancy	NULL	NULL	NULL

La consulta produce seis filas de resultado, mostrando los pares chico/chica emparejados y las chicas desparejadas. Los chicos desparejados faltan en los resultados.

Consultas

Condición lateral (Outer Join)

Lista las chicas y chicos de la misma ciudad y los chicos desparejados.

```
SELECT * FROM CHICAS A, CHICOS B WHERE A.CIUDAD (+) = B.CIUDAD
```

A.Nombre	A.Ciudad	B.Nombre	Ciudad
María	Boston	Juan	Boston
María	Boston	Henry	Boston
Susan	Chicago	Sam	Chicago
Beatriz	Chicago	Sam	Chicago
NULL	NULL	Jorge	Dallas
NULL	NULL	James	NULL

La consulta también produce seis filas de resultado, mostrando los pares chico/chica coincidentes y los chicos no coincidentes. Esta vez las chicas desparejadas faltan en los resultados.

Resumen

En esta sesión, debe haber comprendido lo siguiente:

- SQL DQL es un comando que se utiliza para consultar los datos de la base de datos
- SQL DML son comandos que se utilizan para modificar los datos de la base de datos