ArgentinaPrograma
YoProgramo

Agrupación en SQL

por Leonardo Blautzik, Federico Gasior y Lucas Videla

Julio / Diciembre 2021

Ministerio de
Desarrollo Productivo
Argentina

Agrupación

A veces los datos que obtenemos no son suficientes para determinar la información por separado

Para ello debemos operarlos en conjunto para obtener tuplas de información útil

El estandar SQL define una forma de juntar las tuplas agrupandolas a través de una sentencia GROUP BY

GROUP BY

El GROUP BY devuelve una tupla por cada grupo o conjunto formado

Cada grupo puede ser declarado como una columna dentro del GROUP BY o a través de una función de agregación

Al declararse algún dato de agrupación, cada columna de una consulta deberá aplicarlo

SQLite en particular permite tener columnas que no esten agrupadas, pero obtiene el primer valor del grupo

Funciones de agregación

Las funciones de agregación más comunes son:

- avg(X)
- count(*) / count(X)
- group_concat(X) / group_concat(X,Y)
- max(X)
- min(X)
- sum(X)

Ejemplo

La siguiente consulta devolverá el precio promedio de los productos por categoría

```
SELECT avg(precio) AS 'promedio', categoria
FROM productos
WHERE precio IS NOT NULL
GROUP BY categoria
```

Condiciones de filtro

No puedo utilizar promedio dentro del WHERE ya que forma parte de la agrupación

```
SELECT avg(precio) AS 'promedio', categoria
FROM productos
WHERE precio IS NOT NULL
GROUP BY categoria
```

HAVING

Utilizando HAVING puedo agregar condiciones de los datos ya agrupados

```
SELECT avg(precio) AS 'promedio', categoria
FROM productos
WHERE precio IS NOT NULL
GROUP BY categoria
HAVING promedio > 100
```

Trabajo en clase

Base de datos de ventas

Consultas

Total de precio del stock completo de productos

```
SELECT sum(precio_actual * stock)
  AS 'precio total productos en stock'
FROM productos
```

Consultas

Cantidad de vendedores que ingresaron por año

```
SELECT count(*) AS 'cantidad',
  strftime('%Y', fecha_ingreso) AS 'año_ingreso'
FROM vendedores
GROUP BY "año_ingreso"
```

Consultas

Cantidad de vendedores por sucursal (un id no es una sucursal)

```
SELECT sucursales.nombre AS 'nombre sucursal',
  count(*) AS 'cantidad vendedores'
FROM vendedores
JOIN sucursales ON sucursales.id = sucursal
GROUP BY sucursal
```

Consultas

Cantidad de ventas por sucursal

```
SELECT sucursales.nombre, count(*) AS 'cantidad ventas'
FROM ventas
JOIN vendedores ON vendedores.id = ventas.id_vendedor
JOIN sucursales ON sucursales.id = vendedores.sucursal
GROUP BY sucursales.id
```

Consultas

Cantidad de ventas por cada vendedor por cada sucursal

```
SELECT sucursales.nombre, vendedores.nombre,
  count(*) AS 'cantidad ventas'
FROM ventas
JOIN vendedores ON vendedores.id = ventas.id_vendedor
JOIN sucursales ON sucursales.id = vendedores.sucursal
GROUP BY vendedores.id, sucursales.id
ORDER BY sucursales.id
```

Consultas

Stock de productos vendidos por nombre de producto

```
SELECT productos.nombre, sum(cantidad) AS 'cantidad'
FROM ventas_productos
JOIN productos ON productos.id = ventas_productos.id_producto
GROUP BY id_producto
```

Consultas

Calcular costos totales

```
SELECT ventas.id,
  sum(productos.precio_actual * ventas_productos.cantidad) AS 'total'
FROM ventas_productos
JOIN productos ON productos.id = ventas_productos.id_producto
JOIN ventas ON ventas.id = ventas_productos.id_venta
GROUP BY ventas.id
```

Consultas

Actualizar costos totales

```
UPDATE ventas
SET total = total_venta
FROM (
  SELECT ventas_productos.id_venta,
    sum(productos.precio_actual * ventas_productos.cantidad) AS 'tota
  FROM ventas_productos
  JOIN productos ON productos.id = ventas_productos.id_producto
  JOIN ventas ON ventas.id = ventas_productos.id_venta
  GROUP BY ventas_productos.id_venta
) WHERE ventas.id = id_venta
```

Consultas

Precio de ventas por mes donde la cantidad de ventas sea menor a 10

```
SELECT strftime('%Y%m', fecha) AS 'periodo', sum(total) AS 'total'
FROM ventas
GROUP BY periodo
HAVING count(*) < 10
ORDER BY periodo
```

Consultas

Mejor y peor mes de ventas en precio

```
SELECT periodo, max(total) AS 'total'
FROM (
  SELECT strftime('%Y%m', fecha) AS 'periodo', sum(total) AS 'total'
  FROM ventas  GROUP BY periodo
)
UNION
SELECT periodo, min(total) AS 'total'
FROM (
  SELECT strftime('%Y%m', fecha) AS 'periodo', sum(total) AS 'total'
  FROM ventas  GROUP BY periodo
)
ORDER BY total
```

¡Muchas Gracias!

continuará...

Ministerio de
Desarrollo Productivo
Argentina