

@author: aar.velasquez@gmail.com

@date: 2020/10/15

## Capítulo 8: Agrupación y agregados

---

El enfoque de este capítulo es cómo se pueden agrupar y agregar los datos para permitir a los usuarios para interactuar con él en un nivel más alto de granularidad que lo que está almacenado en la base de datos.

### ***Conceptos de agrupación***

Por ejemplo, digamos que está a cargo de las operaciones en el banco y le gustaría para saber cuántas cuentas abre cada cajero de banco. Podrías emitir una consulta simple para mirar los datos sin procesar:

```
mysql> SELECT open_emp_id  
-> FROM account;
```

si el banco tiene docenas de empleados y miles de cuentas, este enfoque resultaría tedioso y propenso a errores.

En su lugar, puede pedirle al servidor de la base de datos que agrupe los datos por usted utilizando la cláusula 'group by'. Esta es la misma consulta pero empleando una cláusula 'group by' para agrupar la cuenta datos por ID de empleado:

```
mysql> SELECT open_emp_id  
-> FROM account  
-> GROUP BY open_emp_id;
```

Para ver cuantos cuentas que abrió cada cajero, puede usar una función agregada en la cláusula de selección para cuente el número de filas en cada grupo:

```
mysql> SELECT open_emp_id, COUNT(*) how_many  
-> FROM account  
-> GROUP BY open_emp_id;
```

La función agregada count () cuenta el número de filas en cada grupo, el asterisco le dice al servidor que cuente todo en el grupo.

Al agrupar datos, es posible que deba filtrar los datos no deseados de su conjunto de resultados basado en grupos de datos en lugar de basarse en los datos brutos. Dado que la cláusula **group by** se ejecuta después de que se haya evaluado la cláusula **where** , no puede agregar condiciones de filtro a su cláusula **where** para este propósito.

Incorrecto

```
mysql> SELECT open_emp_id, COUNT(*) how_many
-> FROM account
-> WHERE COUNT(*) > 4
-> GROUP BY open_emp_id;
ERROR 1111 (HY000): Invalid use of group function
```

Correcto

```
mysql> SELECT open_emp_id, COUNT(*) how_many
-> FROM account
-> GROUP BY open_emp_id
-> HAVING COUNT(*) > 4;
```

No puede hacer referencia al recuento de funciones agregadas (\*) en su cláusula where , porque el los grupos aún no se han generado en el momento en que se evalúa la cláusula where .

En lugar, debe poner sus condiciones de filtro de grupo en la cláusula de tener . Esto es lo que hace la consulta se vería como usar tener :

## ***Funciones agregadas***

Las funciones agregadas realizan una operación específica en todas las filas de un grupo. A pesar de que cada servidor de base de datos tiene su propio conjunto de funciones agregadas especializadas, Las funciones agregadas implementadas por todos los servidores principales incluyen:

```
Max()
    Devuelve el valor máximo dentro de un conjunto.
Min()
    Devuelve el valor mínimo dentro de un conjunto
Avg()
    Devuelve el valor medio de un conjunto.
Sum()
    Devuelve la suma de los valores de un conjunto.
Count()
    Devuelve el número de valores de un conjunto.
```

A continuación, se muestra una consulta que utiliza todas las funciones agregadas comunes para analizar los saldos de todas las cuentas corrientes:

```
mysql> SELECT MAX(avail_balance) max_balance,
-> MIN(avail_balance) min_balance,
-> AVG(avail_balance) avg_balance,
-> SUM(avail_balance) tot_balance,
-> COUNT(*) num_accounts
-> FROM account
> WHERE product_cd = 'CHK';
```

## Grupos implícitos versus explícitos

En la mayoría de los casos, sin embargo, querrá recuperar columnas adicionales generados por funciones agregadas.

deberá agregar un Cláusula group by para especificar sobre qué grupo de filas deben ser las funciones agregadas aplicado:

```
SELECT product_cd,  
       MAX(avail_balance) max_balance,  
       MIN(avail_balance) min_balance,  
       AVG(avail_balance) avg_balance,  
       SUM(avail_balance) tot_balance,  
       COUNT(*) num_  
FROM account;  
  
mysql> SELECT product_cd,  
-> MAX(avail_balance) max_balance,  
-> MIN(avail_balance) min_balance,  
-> AVG(avail_balance) avg_balance,  
-> SUM(avail_balance) tot_balance,  
-> COUNT(*) num_accts  
-> FROM account  
-> GROUP BY product_cd;
```

## Contando valores distintos

Al usar la función count () para determinar el número de miembros en cada grupo, tiene la opción de contar a todos los miembros del grupo, o contar solo los valores distintos para una columna en todos los miembros del grupo.

```
SELECT COUNT(DISTINCT open_emp_id)  
  
-> FROM account;
```

## Usando Expresiones

Además de usar columnas como argumentos para agregar funciones, puede construir expresiones siones para usar como argumentos. Por ejemplo, es posible que desee encontrar el valor máximo de depósitos pendientes en todas las cuentas, que se calcula restando la disponibilidad saldo del saldo pendiente. Puede lograr esto a través de la siguiente consulta:

```
mysql> SELECT MAX(pending_balance - avail_balance) max_uncleared  
-> FROM account;
```

## Cómo se manejan los NULL

Al realizar agregaciones o, de hecho, cualquier tipo de cálculo numérico, debe Siempre considere cómo los valores **NULL** pueden afectar el resultado de su cálculo.

## Generación de grupos

La gente rara vez está interesada en ver datos sin procesar; en cambio, las personas que participan en los datos El análisis querrá manipular los datos sin procesar para adaptarse mejor a sus necesidades. Ejemplos de Las manipulaciones de datos comunes incluyen:

- Generar totales para una región geográfica, como el total de ventas europeas
- Encontrar valores atípicos, como el principal vendedor de 2005
- Determinar frecuencias, como el número de cuentas nuevas abiertas para cada rama

La cláusula group by es el mecanismo para agrupar datos dentro de una consulta.

### ***Agrupación de una sola columna***

Tipo de agrupación más simple y más utilizado.

```
mysql> SELECT product_cd, SUM(avail_balance) prod_balance  
-> FROM account  
-> GROUP BY product_cd;
```

### **Agrupación multicolumna**

En algunos casos, es posible que desee generar grupos que abarquen más de una columna.

```
mysql> SELECT product_cd, open_branch_id,  
->SUM(avail_balance) tot_balance  
-> FROM account  
-> GROUP BY product_cd, open_branch_id;
```

Esta versión de la consulta genera 14 grupos, uno por cada combinación de producto y sucursal que se encuentran en la tabla de cuentas . Junto con agregar la columna open\_branch\_id a la cláusula select , también lo agregué a la cláusula group by , ya que open\_branch\_id es recuperado de una tabla y no se genera a través de una función agregada.

### **Agrupación a través de expresiones**

Además de usar columnas para agrupar datos, puede crear grupos basados en los valores generado por expresiones.

```
mysql> SELECT EXTRACT(YEAR FROM start_date) year,  
        ->COUNT(*) how_many  
        -> FROM employee  
        -> GROUP BY EXTRACT(YEAR FROM start_date);
```

### Generando acumulaciones

```
mysql> SELECT product_cd, open_branch_id,  
        ->SUM(avail_balance) tot_balance  
        -> FROM account  
        -> GROUP BY product_cd, open_branch_id WITH ROLLUP;
```

### Condiciones de filtro de grupo

Al agrupar datos, también puede aplicar filtros condiciones a los datos después de que se hayan generado los grupos.