

@author aar.velasquez@gmail.com @date 2020/10/14

Capítulo 4: Filtración

Evaluación de condición

Una cláusula **where** **puede** tener una o más condiciones separado por los operadores **AND** u **OR**, entonces todas las condiciones debe evaluarse como verdadero para que la fila se incluya en el conjunto de resultados.

- si están separados por el operador **AND** , todos deben evaluar a verdadero para que la fila se incluya en el conjunto de resultados.
- si todas las condiciones de la cláusula **where** están separadas por el operador **or** , solo una de las condiciones debe evaluarse como verdadera para que la fila se incluya en el conjunto de resultados.

Usando paréntesis

Si su cláusula **WHERE** incluye tres o más condiciones que utilizan tanto el operador **AND** como el operador **OR** debe usar paréntesis para dejar clara su intención.

Usando el operador not

Si bien es fácil de manejar para el servidor de base de datos, generalmente es difícil para una persona evaluar una cláusula **where** que incluya el operador **not** , razón por la cual no contrarrestarlo muy a menudo.

Construyendo una condición

Una condición se compone de uno o más expresiones junto con uno o más operadores. Una expresión puede ser cualquiera de los siguientes: - Un número - Una columna en una tabla o vista - Una cadena literal, como 'Teller' - Una función incorporada, como concat ('Aprendizaje', ' ', 'SQL') - Una subconsulta - Una lista de expresiones, como ('Teller', 'Head Teller', 'Operations Manager')

- Los operadores utilizados dentro de las condiciones incluyen:
- los operadores de comparación, como = , != , < , > , <> , LIKE , IN , y BETWEEN
- Operadores aritméticos, como + , - , * AND /

Tipos de condición

- Condiciones de igualdad
- Condiciones de rango
 - El operador **between**

Cuando tenga un límite superior e inferior para su rango, puede optar por utilizar una condición única que utiliza el operador **between** en lugar de utilizar dos condiciones.

```
-mysql> SELECT emp_id, fname, lname, start_date
-> FROM employee
-> WHERE start_date BETWEEN '2005-01-01' AND '2007-01-01';
```

hay un par de cosas a tener en cuenta al utilizar el operador between:

- siempre debe especificar el límite inferior del rango primero (después 'between') y el límite superior del rango (después 'AND').
- Para trabajar con rangos de cadenas, necesita conocer el orden de los caracteres dentro de su juego de caracteres (el orden en el que se ordenan los caracteres dentro de un juego de caracteres se llama una colación).

- Condiciones de afiliación

- Con el **operador in**, puede escribir una sola condición sin importar cuántas expresiones están en el set.

```
SELECT account_id, product_cd, cust_id, avail_balance
FROM account
WHERE product_cd IN ('CHK', 'SAV', 'CD', 'MM');
```

- Usando subconsultas

puede utilizar una subconsulta para generar un conjunto sobre la marcha

```
mysql> SELECT account_id, product_cd, cust_id,
avail_balance
-> FROM account
-> WHERE product_cd IN (SELECT product_cd FROM
product
->WHERE product_type_cd = 'ACCOUNT');
```

- Usando **not in**

- Condiciones coincidentes

- Si bien la función incorporada left () hace el trabajo, no le brinda mucha flexibilidad.
- En su lugar, puede utilizar caracteres comodín para crear expresiones de búsqueda, como se demuestra en la siguiente sección.

- Usando comodines

Al buscar coincidencias de cadenas parciales, es posible que le interese:

- Cadenas que comienzan / terminan con un determinado carácter
- Cadenas que comienzan / terminan con una subcadena
- Cadenas que contienen cierto carácter en cualquier lugar dentro de la cadena
- Cadenas que contienen una subcadena en cualquier lugar dentro de la cadena
- Cadenas con un formato específico, independientemente de los caracteres individuales

-mysql>SELECT lname

```
-> FROM employee
-> WHERE lname LIKE '_a%e%';
```

- La expresión de búsqueda en el ejemplo anterior especifica cadenas que contienen una a en el segunda posición y seguida de una e en cualquier otra posición de la cadena (incluida la Ultima posicion). La Tabla 4-5 muestra algunas expresiones de búsqueda más y sus interpretaciones.

-Usando expresiones regulares

Si encuentra que los caracteres comodín no proporcionan suficiente flexibilidad, puede usar expresiones regulares para construir expresiones de búsqueda.

```
- mysql> SELECT emp_id, fname, lname
-> FROM employee
-> WHERE lname REGEXP '^[FG]';
```

NULL:

-Es la ausencia de un valor

-Null es un poco resbaladizo, ya que hay varios sabores de null :

*No aplica:

Como la columna de identificación de empleado para una transacción que tuvo lugar en un cajero automático máquina.

*Valor aún no conocido:

Por ejemplo, cuando no se conoce la identificación federal en el momento en que se crea una fila de cliente Valor indefinido.

Por ejemplo, cuando se crea una cuenta para un producto que aún no se ha agregado a la base de datos.

- Cuando trabaje con null , debe recordar:

- Una expresión puede ser nula, pero nunca puede ser igual a nula.
- Dos nulos nunca son iguales entre sí.

...

Para probar si una expresión es nula , debe usar el operador ****IS NULL**** , como se demuestra:

```
mysql> SELECT emp_id, fname, lname, superior_emp_id  
-> FROM employee  
   > WHERE superior_emp_id IS NULL;
```