
ORACLE
Academy

Programación de Bases de Datos con SQL

2-3
Operadores de Comparación

ORACLE
Academy

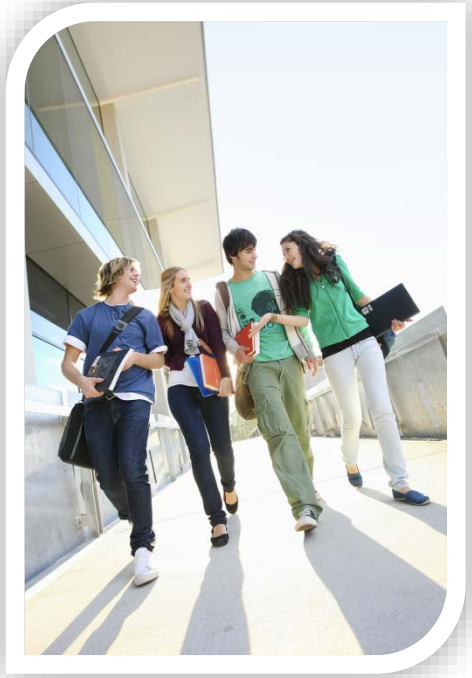
Objetivos

En esta lección se abordan los siguientes objetivos:

- Aplicar el operador de comparación adecuado para devolver un resultado deseado
- Demostrar un uso adecuado de las condiciones BETWEEN, IN y LIKE para devolver un resultado deseado
- Distinguir entre cero y NULL, el cual significa que no está disponible, que está sin asignar, que es desconocido o que no es aplicable
- Explicar el uso de las condiciones de comparación y NULL

Objetivo

- Las comparaciones se utilizan en las conversaciones diarias sin realmente pensar en ello.
  - Las comparaciones se utilizan en las conversaciones diarias sin realmente pensar en ello.
  - Las comparaciones se utilizan en las conversaciones diarias sin realmente pensar en ello.
  - Las comparaciones se utilizan en las conversaciones diarias sin realmente pensar en ello.



Objetivo

- La necesidad de expresar estos tipos de comparaciones también existe en SQL.
- Las condiciones de comparación se utilizan para buscar datos en una tabla que cumplan determinadas condiciones.
- Poder formular una cláusula SELECT para devolver datos específicos es una función potente de SQL.

Operadores de Comparación

- Ya está familiarizado con los operadores de comparación, como igual que (=), menor que (<) y mayor que (>).
- SQL tiene otros operadores que agregan funcionalidad para recuperar juegos específicos de los datos.
- Son los siguientes:
  - BETWEEN…AND
  - IN
  - LIKE

BETWEEN...AND

- El operador BETWEEN...AND se utiliza para seleccionar y mostrar las filas según un rango de valores.
- Cuando se utiliza con la cláusula WHERE, la condición BETWEEN...AND devolverá un rango de valores entre los límites inferior y superior, incluyendo estos.

BETWEEN…AND

- Tenga en cuenta que en el ejemplo de la base de datos Employees, los valores devueltos incluyen el valor del límite inferior y el valor del límite superior.
- Los valores especificados con la condición BETWEEN se incluyen.
- Tenga en cuenta también que el valor de límite inferior se debe enumerar en primer lugar.

```
SELECT last_name, salary
FROM employees
WHERE salary BETWEEN 9000 AND 11000;
```

LAST_NAME	SALARY
Zlotkey	10500
Abel	11000
Hunold	9000

- Tenga en cuenta también que el valor de límite inferior se debe enumerar en primer lugar.

BETWEEN…AND

- Usar BETWEEN...AND es lo mismo que utilizar la siguiente expresión:

```
WHERE salary >= 9000 AND salary <=11000;
```

- De hecho, no hay ningún beneficio en el rendimiento al usar una expresión u otra.
- Para mayor simplicidad al leer código, utilizamos BETWEEN...AND.

IN

- La condición IN también se conoce como la "condición de miembro".
- Se utiliza para probar si un valor está en un juego de valores especificado.
- Por ejemplo, IN se podría utilizar para identificar a los alumnos cuyos números de identificación sean 2349, 7354 o 4333, o a las personas cuyos código de llamada por teléfono internacional sean 1735, 82 o 10.

```
SELECT city, state_province, country_id
FROM locations
WHERE country_id IN('UK', 'CA');
```

CITY	STATE_PROVINCE	COUNTRY_ID
Toronto	Ontario	CA
Oxford	Oxford	UK

IN

- En este ejemplo, la cláusula WHERE también se puede escribir como un juego de condiciones OR:

```
SELECT city, state_province, country_id
FROM locations
WHERE country_id IN('UK', 'CA');
…
WHERE country_id = 'UK' OR country_id = 'CA';
```

- Al igual que con BETWEEN...AND, la condición IN se puede escribir con sintaxis igual de eficaz.

LIKE

- ¿Alguna vez ha ido de compras buscando algo que ha visto en una revista o en televisión, pero no estaba seguro del artículo exacto?
- Con las búsquedas en las bases de datos pasa más o menos lo mismo.
- Un administrador puede saber que el apellido de un empleado empieza por "S", pero no saber el nombre completo del empleado.
- Afortunadamente, en SQL, la condición LIKE permite seleccionar las filas que coincidan con caracteres, fechas o patrones de números.

LIKE

- Dos símbolos, el (%) y el guión bajo (_), llamados caracteres comodín, se pueden utilizar para crear una cadena de búsqueda.
- El símbolo del porcentaje (%) se utiliza para representar cualquier secuencia de cero o más caracteres.

```
SELECT last_name
FROM employees
WHERE last_name LIKE '_o%';
```

LAST_NAME
Kochhar
Lorentz
Mourgos

LIKE

- El símbolo del carácter de subrayado (_) se utiliza para representar un único carácter.
- En el ejemplo que se muestra a continuación, se devolverán todos los empleados cuyos apellidos empiecen por cualquier letra seguida de una "o" y, a continuación, seguidos de cualquier otro número de letras.

```
SELECT last_name
FROM employees
WHERE last_name LIKE '_o%';
```

LAST_NAME
Kochhar
Lorentz
Mourgos

LIKE

```
SELECT last_name
FROM employees
WHERE last_name LIKE '_o%';
```

- ¿Cuáles de los siguientes apellidos se podría haber devuelto con la consulta anterior?
  - 1. Sommersmith
  - 2. Oog
  - 3. Fong
  - 4. Mo

LIKE

- Una opción adicional que es importante:
  - Cuando necesite tener una coincidencia exacta para una cadena que tenga un carácter % o _, deberá indicar que dichos caracteres no son comodines sino parte del elemento que está buscando.

LIKE

- La opción ESCAPE se puede utilizar para indicar que los caracteres _ o % son parte del nombre, no valores comodín.
- Por ejemplo, si deseamos recuperar el JOB_ID de un empleado de la tabla employees que contenga el patrón _R, tendríamos que utilizar un carácter de escape para indicar que estamos buscando un guión bajo, y no solo cualquier carácter.

```
SELECT last_name, job_id
FROM EMPLOYEES
WHERE job_id LIKE '%\_R%' ESCAPE '\';
```

- En este ejemplo, se utiliza la barra invertida '\' como carácter de escape, pero se puede utilizar cualquier carácter.

LIKE

- Sin la opción ESCAPE, se devolverían todos los empleados que tuvieran una R en sus JOB_ID.

```
SELECT last_name, job_id
FROM EMPLOYEES
WHERE job_id LIKE '%_R%'
```

LAST_NAME	JOB_ID
Abel	SA_REP
Davies	ST_CLERK
Ernst	IT_PROG
Fay	MK_REP
Fay	MK_REP
Grant	SA_REP
Higgins	AC_MGR
Hunold	IT_PROG
King	AD_PRES
Lorentz	IT_PROG
Matos	ST_CLERK
Rajs	ST_CLERK
Taylor	SA_REP
Vargas	ST_CLERK

IS NULL, IS NOT NULL

- ¿Se acuerda de NULL?
- Es el valor que no está disponible, sin asignar, desconocido o que no es aplicable.
- Poder probar el valor NULL es a menudo aconsejable.
- Es posible que desee saber todas las fechas en junio en las que, ahora mismo, no tenga programado un concierto.
- Es posible que desee saber todos los clientes que no tengan números de teléfono registrados en su base de datos.

IS NULL, IS NOT NULL

- La condición IS NULL prueba los datos que no están disponibles, sin asignar, o datos desconocidos.
- IS NOT NULL prueba los datos disponibles en la base de datos.
- En el ejemplo de la siguiente diapositiva, la cláusula WHERE se escribe para recuperar todos los apellidos de aquellos empleados que no tengan un jefe.

IS NULL, IS NOT NULL

```
SELECT last_name, manager_id
FROM employees
WHERE manager_id IS NULL;
```

LAST_NAME
King

- El empleado King es el Presidente de la compañía, por lo que no tiene jefe.

```
SELECT last_name, commission_pct
FROM employees
WHERE commission_pct IS NOT NULL;
```

LAST_NAME	COMMISSION_PCT
Zlotkey	2
Abel	3
Taylor	2
Grant	15

- IS NOT NULL devuelve las filas que tienen un valor en la columna commission_pct.

Terminología

Entre los términos clave utilizados en esta lección se incluyen:

- BETWEEN…AND
- IN
- LIKE
- IS NULL
- IS NOT NULL

Resumen

En esta lección, debe haber aprendido lo siguiente:

- Aplicar el operador de comparación adecuado para devolver un resultado deseado
- Demostrar un uso adecuado de las condiciones BETWEEN, IN y LIKE para devolver un resultado deseado
- Distinguir entre cero y NULL, este último significa que no está disponible, sin asignar, desconocido o que no es aplicable.
- Explicar el uso de las condiciones de comparación y NULL


ORACLE
Academy