



# Programación de Bases de Datos con SQL

2-2

Limitación de Filas Seleccionadas



# Objetivos

En esta lección se abordan los siguientes objetivos:

- Aplicar sintaxis SQL para restringir las filas devueltas de una consulta
- Demostrar la aplicación de la sintaxis de la cláusula WHERE
- Explicar el motivo por el que es importante, desde una perspectiva de negocio, poder limitar fácilmente los datos recuperados de una tabla
- Crear y generar salidas mediante una consulta SQL que contenga cadenas de caracteres y valores de fecha
- ¿Ha tenido alguna vez "sobrecarga de información"?

**ORACLE®**

Academy

DPS2L2  
Limitación de Filas Seleccionadas

Copyright © 2019, Oracle y/o sus filiales. Todos los derechos reservados.

3

# Objetivo

En esta lección se abordan los siguientes objetivos:

- La televisión está encendida, su madre le pregunta cómo le ha ido en clase, suena el teléfono y el perro está ladrando.
- ¿No sería fantástico poder limitar la cantidad de información que tiene que procesar a la vez?
- En SQL, este es el trabajo de la cláusula WHERE.
- Es importante poder seleccionar la información que necesita ver de una tabla.
- Las tablas pueden tener millones de filas de datos, y es un desperdicio de recursos buscar y que devuelvan datos que no necesite o desee.

# Sentencia SELECT

- SELECT se utiliza para recuperar información de la base de datos.
- Una sentencia SELECT debe incluir como mínimo una cláusula SELECT y una cláusula FROM.
- La cláusula WHERE es opcional.

```
SELECT*|{[DISTINCT] column | expression alias}..  
FROM table  
[WHERE condition(s)];
```

# Cláusula WHERE

- Al recuperar datos de la base de datos, puede que necesite limitar las filas de datos que se muestran.
- Esto se consigue con la cláusula WHERE.
- Una cláusula WHERE contiene una condición que se debe cumplir e, inmediatamente después, le sigue la cláusula FROM en una sentencia SQL.
- La sintaxis de la cláusula WHERE es:

```
WHERE column_name comparison_condition comparison_value
```

- Nota: No se puede utilizar un alias en la cláusula WHERE.

# Cláusula WHERE

- Examine la siguiente sentencia SQL de la base de datos Employees:

```
SELECT employee_id, first_name, last_name  
FROM employees;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME
100	Steven	King
101	Neena	Kochhar
102	Lex	De Haan

- Mediante la adición de una cláusula WHERE, las filas se limitan a aquellas filas en las que el valor de employee\_id es 101.

```
SELECT employee_id, first_name, last_name  
FROM employees  
WHERE employee_id = 101;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME
101	Neena	Kochhar

# Operadores de Comparación en la Cláusula WHERE

- Como ha visto en la diapositiva anterior, el signo = se puede utilizar en la cláusula WHERE.
- Además del operador "igual que" ( = ), se pueden utilizar otros operadores de comparación para comparar una expresión con otra:
  - = igual que
  - > mayor que
  - >= mayor o igual que
  - < menor que
  - <= menor o igual que
  - <> no es igual que (o != o ^=)



# Operadores de Comparación en la Cláusula WHERE

- En el siguiente ejemplo, la columna `department_id` se utiliza en la cláusula `WHERE`, con el operador de comparación `=`.
- Se devuelven todos los empleados con un `department_id` de 90.

```
SELECT employee_id, last_name, department_id
FROM employees
WHERE department_id = 90;
```

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
100	King	90
101	Kochhar	90
102	De Haan	90

# Cadenas de Caracteres y de Fecha en la Cláusula WHERE

- Las fechas y cadenas de caracteres de la cláusula WHERE se deben incluir entre comillas simples (' ').
- Sin embargo, los números no se deben incluir entre comillas simples.

# Cadenas de Caracteres y de Fecha en la Cláusula WHERE

- Observe el siguiente ejemplo de la base de datos Employees.
- La cláusula WHERE contiene una cadena y se incluye entre comillas simples.

```
SELECT first_name, last_name  
FROM employees  
WHERE last_name = 'Taylor';
```

# Cadenas de Caracteres y de Fecha en la Cláusula WHERE

- Qué cree que pasará si la cláusula WHERE se escribiera así:

```
WHERE last_name = 'jones';
```

- Todas las búsquedas de caracteres son sensibles a mayúsculas/minúsculas.
- Debido a que la tabla Employees almacena todos los apellidos con el uso de mayúsculas adecuado, no se devolverán filas en este ejemplo.

# Cadenas de Caracteres y de Fecha en la Cláusula WHERE

- Es importante recordar este aspecto.
- En otra lección, aprenderá a utilizar otras palabras clave SQL como UPPER, LOWER e INITCAP que facilitarán evitar los errores de la sensibilidad a mayúsculas/minúsculas.

# Operadores de Comparación en la Cláusula WHERE

- Los operadores de comparación se pueden utilizar en todas las formas siguientes en la cláusula WHERE:

```
WHERE hire_date < '01-Jan-2000'
```

```
WHERE salary >= 6000
```

```
WHERE job_id = 'IT_PROG'
```

- En el siguiente ejemplo de la base de datos Employees, ¿qué filas se seleccionarán?
- ¿Se incluirán los salarios de 3000 en el juego de resultados?

```
SELECT last_name, salary  
FROM employees  
WHERE salary <= 3000;
```

# Terminología

Entre los términos clave utilizados en esta lección se incluyen:

- Cláusula WHERE
- Operadores de comparación

# Resumen

En esta lección, debe haber aprendido lo siguiente:

- Aplicar sintaxis SQL para restringir las filas devueltas de una consulta
- Demostrar la aplicación de la sintaxis de la cláusula WHERE
- Explicar el motivo por el que es importante, desde una perspectiva de negocio, poder limitar fácilmente los datos recuperados de una tabla
- Crear y generar salidas mediante una consulta SQL que contenga cadenas de caracteres y valores de fecha



