



 **ACADEMY**

Programación de bases de datos con SQL

1-3

Anatomía de una Sentencia SQL



ORACLE ACADEMY

Copyright © 2017, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivos

En esta lección se abordan los siguientes objetivos:

- Emparejar proyección y selección con sus capacidades correctas
- Crear una sentencia SELECT básica
- Utilizar la sintaxis correcta para mostrar todas las filas de una tabla
- Utilizar la sintaxis correcta para seleccionar columnas específicas de una tabla, modificar la forma en que se muestran los datos y realizar cálculos utilizando expresiones aritméticas y operadores

Objetivos

En esta lección se abordan los siguientes objetivos:

- Formular consultas mediante la prioridad de operador correcta para mostrar los resultados deseados
- Definir un valor nulo
- Demostrar el efecto que los valores nulos crean en las expresiones aritméticas
- Construir una consulta con un alias de columna

Palabra clave SELECT

- SELECT es una de las palabras clave más importantes, si no la más importante, en SQL.
- SELECT se utiliza para recuperar información de la base de datos. Cuando aprenda cómo utilizar SELECT, habrá abierto la puerta de la base de datos.
- Imagine una base de datos que contiene información sobre películas como, por ejemplo, el título, el género, el estudio, el productor, la fecha de estreno, las series, el país, el idioma, la puntuación, la duración y demás.
- ¿Qué pasaría si solo necesitan los títulos de las películas creadas en India?
- La sentencia SELECT le permite buscar datos específicos.

Sentencia SELECT

- La sentencia SELECT recupera información de la base de datos.
- La sintaxis de la sentencia SELECT es la siguiente:

```
SELECT <column_name(s)>  
FROM <table_name>;
```

- En su formato más simple, una sentencia SELECT debe incluir lo siguiente:
 - Una cláusula SELECT, que especifica las columnas que se van a mostrar.
 - Una cláusula FROM, que especifica la tabla que contiene las columnas enumeradas en la cláusula SELECT

Cláusula SELECT: especifica las columnas que se van a mostrar (o utilice el símbolo * para mostrar todas las columnas).

Una cláusula FROM: especifica la tabla que contiene las columnas enumeradas en la cláusula SELECT.

Columna: una implantación de un atributo o relación en una tabla.

Convenciones

- En este curso, se utilizará lo siguiente:

```
SELECT last_name  
FROM employees
```

- Una palabra clave hace referencia a un comando SQL individual.
- Por ejemplo, SELECT y FROM son palabras claves.
- Una cláusula es una parte de una sentencia SQL.
- Por ejemplo, SELECT employee_id,last_name, es una cláusula.
- Una sentencia es una combinación de dos o más cláusulas.
- Por ejemplo, SELECT last_name FROM employees es una sentencia.



ACADEMY

DPS1L3
Anatomía de una Sentencia SQL

Copyright © 2017, Oracle y/o sus filiales. Todos los derechos reservados.

7

En este curso se utilizan las siguientes convenciones de estilo SQL:

Palabras clave SQL en MAYÚSCULAS: por ejemplo, SELECT, FROM, WHERE.

Los nombres de la tabla y la columna en minúsculas: por ejemplo, first_name, employees.

Cada cláusula en una nueva línea: por ejemplo,

```
SELECT last_name
```

```
FROM employees
```

```
WHERE employee_id = 101;
```

Capacidades de las Sentencias SELECT

- Proyección: se utiliza para seleccionar columnas de una tabla
- Selección: se utiliza para seleccionar filas de una tabla

Tabla 2: proyección

Tabla 2: selección

Proyección: puede utilizar la capacidad de proyección en SQL para elegir las columnas en una tabla que desee que le devuelva su consulta. Puede seleccionar el número de columnas de la tabla que desee.

Selección: puede utilizar la capacidad de selección en SQL para elegir las filas en una tabla que desee que le devuelva una consulta. Puede utilizar varios criterios para restringir las filas que ve

Proyección y Selección

ID	FIRST_NAME	LAST_NAME	SALARY
10	John	Doe	4000
20	Jane	Jones	3000
30	Sylvia	Smith	5000
40	Hai	Nguyen	6000

Proyección

Selección

```
SELECT salary
FROM employees
WHERE last_name like 'Smith';
```

La cláusula SELECT determina la proyección.

La cláusula WHERE determina la selección.

La ejecución de esta consulta daría como resultado la visualización de la columna de salario de cualquier empleado con el last_name (apellido) Smith.

Selección de Todas las Columnas

- Puede mostrar todas las columnas de datos de una tabla mediante el uso de un símbolo de asterisco (*) en lugar de un nombre de columna en la cláusula SELECT.
- En el ejemplo, están seleccionadas todas las columnas de la tabla de países.

```
SELECT *  
FROM countries;
```

COUNTRY_ID	COUNTRY_NAME	REGION_ID
Esquema de	Canadá	2
DE	Germany	1
UK	United Kingdom	1
US	United States of America	2

A veces, realizar una tabla `SELECT * FROM` puede tardar bastante tiempo en devolver los datos. Todo depende del número de filas almacenadas en esa tabla concreta. Recuerde que las tablas de Oracle pueden almacenar millones de filas de datos.

Selección de Todas las Columnas

- También puede mostrar todas las columnas en una tabla enumerándolas individualmente.

```
SELECT country_id, country_name, region_id  
FROM countries;
```

COUNTRY_ID	COUNTRY_NAME	REGION_ID
Esquema de	Canadá	2
DE	Germany	1
UK	United Kingdom	1
US	United States of America	2

Todos los ejemplos de las diapositivas tienen un punto y coma después de cada sentencia. Oracle Application Express no necesita esta sintaxis, pero las demás interfaces de Oracle SQL sí, por lo que se han incluido en los cursos.

Proyección de Columnas Concretas

- Si desea PROYECTAR para que solo se muestren las columnas específicas de una tabla, simplemente enumere cada uno de los nombres de las columnas que desee y separe cada nombre con una coma en la cláusula SELECT.

```
SELECT location_id, city, state_province  
FROM locations;
```

LOCATION_ID	CITY	STATE_PROVINCE
1800	Toronto	Ontario
2500	Oxford	Oxford
1400	Southlake	Texas
1500	South San Francisco	California
1700	Seattle	Washington

Uso de Operadores Aritméticos

- Con unas sencillas reglas y directrices, puede construir sentencias SQL que sean fáciles de leer y de editar.
- Conocer las reglas hará que el aprendizaje de SQL sea más fácil.
- Puede que necesite modificar la forma en la que se muestran los datos, realizar cálculos o consultar supuestos de posibilidades.
- Por ejemplo, "¿qué pasaría si a cada empleado se le elevara el sueldo un 5 %?"
- ¿Cómo afectaría eso a sus cifras de beneficio anuales?"

Uso de Operadores Aritméticos

- Estos tipos de cálculos son posibles mediante las expresiones aritméticas.
- Ya está familiarizado con las expresiones aritméticas de matemáticas:
 - sumar (+), restar (-) , multiplicar (*) y dividir (/)
- Tenga en cuenta que este ejemplo no crea nuevas columnas en las tablas ni cambia los valores reales de los datos.
- Los resultados de los cálculos solo aparecerán en la salida.

Expresión aritmética: una expresión que da como resultado un valor numérico.

Uso de Operadores Aritméticos

- El ejemplo que se muestra utiliza el operador de suma para calcular un aumento de sueldo de 300 para todos los empleados y muestra una nueva columna, SALARY+300, en la salida.

```
SELECT last_name, salary, salary + 300  
FROM employees;
```

- Colocar espacios en blanco antes y después de un operador aritmético no afectará a la salida.

LAST_NAME	SALARY	SALARY+300
King	24000	24300
Kochhar	17000	17300
De Haan	17000	17300
Whalen	4400	4700
Higgins	12000	12300
Gietz	8300	8600
Zlotkey	10500	10800
Abel	11000	11300
Taylor	8600	8900
Grant	7000	7300

Operador aritmético: símbolo que se utiliza para realizar una operación en algunos valores.

Prioridad de los Operadores Aritméticos

- La prioridad es el orden en el que Oracle evalúa los diferentes operadores en la misma expresión.
- Al evaluar una expresión con varios operadores, Oracle evalúa los operadores con mayor prioridad antes de evaluar los de menor prioridad.
- Oracle evalúa los operadores con la misma prioridad de izquierda a derecha dentro de una expresión.

Prioridad de los Operadores Aritméticos

- Los operadores aritméticos realizan las operaciones matemáticas de multiplicación, división, suma y resta.
- Si estos operadores aparecen juntos en una expresión, la multiplicación y la división se evaluarán primero.
- Por lo que el orden es: * / + -.
- Una forma fácil recordar la prioridad de los operadores es la fórmula mnemotécnica: Mi Deuda Se Reduce

Prioridad de los Operadores Aritméticos

- Si los operadores en una expresión tienen la misma prioridad, la evaluación se realiza de izquierda a derecha.
- Siempre puede utilizar los paréntesis para forzar que la expresión entre paréntesis se evalúe primero.
- En las tablas de ejemplo que aparecen en la siguiente diapositiva, tenga en cuenta las diferencias en la salida entre la consulta que utilizó los paréntesis y la que no.

Prioridad de los Operadores Aritméticos

Prioridad de Operadores

```
SELECT last_name, salary, 12*salary +100
FROM employees;
```

LAST_NAME	SALARY	12*SALARY+100
King	24000	288100
Kochhar	17000	204100
De Haan	17000	204100
Whalen	4400	52900
Higgins	12000	144100
Gietz	8300	99700

Uso de paréntesis

```
SELECT last_name, salary, 12*(salary +100)
FROM employees;
```

LAST_NAME	SALARY	12*(SALARY+100)
King	24000	289200
Kochhar	17000	205200
De Haan	17000	205200
Whalen	4400	54000
Higgins	12000	145200
Gietz	8300	100800

El ejemplo de la izquierda utiliza la prioridad de los operadores para determinar el orden en el que se realizan las operaciones. Como * (multiplicación) tiene mayor prioridad, el salario se multiplica primero por 12 y, a continuación, se agrega 100 al resultado de la multiplicación. Por lo tanto, para el empleado King (salario, 24 000), se calcula $12 * 24\,000 = 288\,000$, después se añade 100 para dar la respuesta 288 100.

El ejemplo de la derecha utiliza paréntesis para forzar la adición que se llevará a cabo en primer lugar, el resultado de la adición, entonces, se multiplica por 12. Por lo tanto, King tiene $24\,000 + 100 = 24\,100$, después se multiplica por 12 para proporcionar la respuesta 289 200.

Valores NULL

- En SQL, NULL es una palabra interesante.
- Para comprender el concepto NULL, debe saber qué es NULL y qué no es NULL.
- NULL es un valor que no está disponible, sin asignar, desconocido o que no es aplicable.
- Un valor NULL no es lo mismo que un cero o un espacio.
- En SQL, un cero es un número y un espacio es un carácter.

NULL: un valor que no está disponible, sin asignar, desconocido o que no es aplicable.

Valores NULL

- A veces, no conoce el valor de una columna.
- En una base de datos, puede almacenar valores desconocidos en sus bases de datos.
- Las bases de datos relacionales utilizan un marcador llamado NULL o null para representar dichos valores desconocidos.

NULL: un valor que no está disponible, sin asignar, desconocido o que no es aplicable.

Valores NULL

- Si cualquier valor de columna en una expresión aritmética es null, el resultado es nulo o desconocido.
- Si intenta dividir un número entre un valor null, el resultado será nulo o desconocido.
- Sin embargo, si se intenta una división entre cero, aparecerá un mensaje de error.

Salarios y Comisiones

LAST_NAME	JOB_ID	SALARY	COMMISSION_PCT
King	AD_PRES	24000	-
Kochhar	AD_VP	17000	-
De Haan	AD_VP	17000	-
Whalen	AD_ASST	4400	-
Higgins	AC_MGR	12000	-
Gietz	AC_ACCOUNT	8300	-
Zlotkey	SA_MAN	10500	2
Abel	SA_REP	11000	3

Valores NULL

```
SELECT last_name, job_id, salary,  
commission_pct, salary*commission_pct  
FROM employees;
```

Salaries and Commissions

LAST_NAME	JOB_ID	SALARY	COMMISSION_PCT	SALARY*COMMISSION_PCT
King	AD_PRES	24000	-	-
Kochhar	AD_VP	17000	-	-
De Haan	AD_VP	17000	-	-
Whalen	AD_ASST	4400	-	-
Higgins	AC_MGR	12000	-	-
Gietz	AC_ACCOUNT	8300	-	-
Zlotkey	SA_MAN	10500	2	2100
Abel	SA_REP	11000	3	3300
Taylor	SA_REP	8600	2	1720



ACADEMY

DPS1L3
Anatomía de una Sentencia SQL

Copyright © 2017, Oracle y/o sus filiales. Todos los derechos reservados.

23

En el ejemplo, una fila con un valor null en la columna commission_pct devolverá un valor nulo para la columna salary*commission_pct en esta consulta.

Los valores NULL aparecen en APEX utilizando el símbolo – (guión).

Alias

- Un alias es una forma de cambiar el nombre de una cabecera de columna en la salida.
- Sin alias, cuando se muestra el resultado de una sentencia SQL, el nombre de las columnas que se muestran será el mismo que los nombres de columna de la tabla o un nombre que muestre una operación aritmética como $12 * (\text{SALARY} + 100)$.
- Es probable que desee que la salida le muestre un nombre que sea más fácil de comprender, un nombre más "descriptivo".
- Los alias de columna le permiten cambiar el nombre de las columnas en la salida.

Alias de columna: cambia el nombre de una cabecera de columna para fines de visualización. El nombre de la columna original en la tabla permanece sin cambios.

Alias

- Hay varias reglas al utilizar los alias de columna para dar formato a la salida.
- Un alias de columna:
 - Cambia el nombre de una cabecera de columna
 - Es útil para realizar cálculos
 - Sigue de forma inmediata al nombre de columna
 - Puede tener la palabra clave opcional AS entre el nombre de columna y el alias
 - Necesita comillas dobles si el alias contiene espacios o caracteres especiales o es sensible a mayúsculas/minúsculas

Uso de Alias de Columna

- La sintaxis de los alias es la siguiente:

```
SELECT * |column|expr [ AS alias], .....  
FROM table;
```

- Ejemplos:

```
SELECT last_name AS name,  
commission_pct AS comm  
FROM employees;
```

NAME	COMM
King	-
Kochhar	-
De Haan	-

```
SELECT last_name "Name",  
salary*12 "Annual Salary"  
FROM employees;
```

Name	Annual Salary
King	288000
Kochhar	204000
De Haan	204000



ACADEMY

DPS1L3
Anatomía de una Sentencia SQL

Copyright © 2017, Oracle y/o sus filiales. Todos los derechos reservados.

26

Las comillas dobles son necesarias para alias con más de una palabra o cuando el formato del alias es distinto del nombre en mayúsculas predeterminadas (p. ej., Nombre del Departamento, departamento o dept.)

El primer ejemplo se utiliza la palabra clave AS, pero no tiene "comillas dobles", por lo que los alias se muestran con las MAYÚSCULAS predeterminadas.

El segundo ejemplo omite AS, pero como el alias es de más de una palabra, utiliza las comillas.

El uso de la palabra clave opcional AS facilita la lectura de las sentencias SQL.

Terminología

Entre los términos clave utilizados en esta lección se incluyen:

- Expresión aritmética
- Operador aritmético
- Cláusula
- Columna
- Alias de columna
- Cláusula FROM
- NULL

Terminología

Entre los términos clave utilizados en esta lección se incluyen:

- Proyección
- Cláusula SELECT
- Selección
- Sentencia SELECT
- Sentencia
- Cláusula WHERE
- * (Asterisco)

Resumen

En esta lección, debe haber aprendido lo siguiente:

- Emparejar proyección y selección con sus capacidades correctas
- Crear una sentencia SELECT básica
- Utilizar la sintaxis correcta para mostrar todas las filas de una tabla
- Utilizar la sintaxis correcta para seleccionar columnas específicas de una tabla, modificar la forma en que se muestran los datos y realizar cálculos utilizando expresiones aritméticas y operadores

Resumen

En esta lección, debe haber aprendido lo siguiente:

- Formular consultas mediante la prioridad de operador correcta para mostrar los resultados deseados
- Definir un valor nulo
- Demostrar el efecto que los valores nulos crean en las expresiones aritméticas
- Construir una consulta con un alias de columna



 **ACADEMY**