

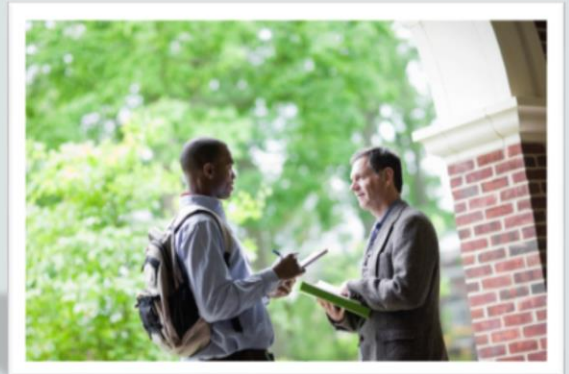


ACADEMY

Fundamentos de bases de datos

6-6

Recuperación de datos mediante SELECT



ORACLE ACADEMY

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

Guía básica

Introducción a
Oracle
Application
Express (APEX)

Lenguaje de
consulta
estructurado
(SQL)

Lenguaje de
definición de
datos (DDL)

Lenguaje de
manipulación
de datos
(DML)

Lenguaje de
control de
transacciones
(TCL)

Se encuentra aquí

Recuperación
de datos
mediante
SELECT

Restricción de
datos
mediante
WHERE

Ordenación de
datos
mediante
ORDER BY

Unión de
tablas
mediante
JOIN



ORACLE ACADEMY

DFo 6-6
Recuperación de datos mediante SELECT

Copyright © 2017, Oracle y/o sus filiales. Todos los derechos reservados. 3

Objetivos

En esta lección se abordan los siguientes objetivos:

- Enumerar las capacidades de las sentencias SQL `SELECT`
- Escribir y ejecutar una sentencia `SELECT` que:
 - Devuelve todas las filas y columnas de una tabla
 - Devuelve columnas específicas de una tabla
 - Utiliza alias de columna para mostrar cabeceras de columna descriptivas



Objetivos

En esta lección se abordan los siguientes objetivos:

- Escribir y ejecutar una sentencia `SELECT` que:
 - Utiliza operadores aritméticos y de concatenación
 - Utiliza cadenas de caracteres literales
 - Elimina filas duplicadas
- Describir la estructura de una tabla



Sentencia SELECT básica

- SELECT identifica las columnas que se van a mostrar.
- FROM identifica la tabla que contiene esas columnas.

```
SELECT {*|[DISTINCT] column|expression [alias],...}  
FROM    table;
```



DFo 6-6
Recuperación de datos mediante SELECT

Copyright © 2017, Oracle y/o sus filiales. Todos los derechos reservados. 6

En la sintaxis:

- SELECT es una lista de una o más columnas.
- * selecciona todas las columnas.
- DISTINCT suprime los duplicados.
- column|expression selecciona la columna o expresión con nombre.
- alias proporciona diferentes cabeceras a las columnas seleccionadas.
- FROM table especifica la tabla que contiene las columnas.

Nota: A lo largo de este curso, las palabras *palabra clave*, *cláusula* y *sentencia* se utilizan de la siguiente forma:

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

Selección de Todas las Columnas

- Para ver todas las columnas de una tabla, coloque un* después de la palabra clave SELECT

```
SELECT *  
FROM departments;
```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting	-	1700

Selección de columnas concretas: proyección

- Puede utilizar la sentencia SELECT para mostrar columnas concretas de la tabla indicando los nombres de columna en el orden que desea verlas, separadas por comas.

```
SELECT department_id, location_id  
FROM departments;
```

DEPARTMENT_ID	LOCATION_ID
10	1700
20	1800
50	1500
60	1400
80	2500
90	1700
110	1700
190	1700

Escritura de Sentencias SQL

Con estas sencillas reglas y directrices que se muestran a continuación, puede construir sentencias válidas que son fáciles de leer y editar.

- Las sentencias SQL no son sensibles a mayúsculas/minúsculas.
- Las sentencias SQL se pueden introducir en una o más líneas.
- Las palabras clave no se pueden abreviar o dividir entre líneas y suelen aparecer en mayúscula.
- Las cláusulas se suelen colocar en líneas independientes.
- El sangrado se utiliza para mejorar la legibilidad
- En Application Express, las sentencias SQL pueden terminar con punto y coma (;), pero no es obligatorio.

Escenario de caso: Recuperación de datos



Profesor

Sean, me gustaría recuperar los datos de las tablas **AUTHORS** y **BOOKS**. ¿Se puede?

Claro. Permítame que recupere los datos y se los muestre



Alumno

Escenario de caso: Recuperación de datos

```
SELECT ID,NAME  
FROM AUTHORS
```

ID	NAME
200	P.G. Wodehouse
300	George Bernard Shaw
100	Leo Tolstoy

```
SELECT *  
FROM BOOKS
```



**Esta es la
información**

ID	TITLE	PUBLISHER_ID	AUTHOR_ID
3	An Unsocial Socialist	30	300
1	War and Peace	10	100
2	The Clicking of Cuthbert	20	200

Ejercicio 1 del proyecto

DFo_6_6_1_Project

Base de datos de la tienda Oracle Baseball League

Escritura y ejecución de sentencias SELECT:

- Recuperar todas las columnas
- Seleccionar columnas concretas

Expresiones Aritméticas

- Crear expresiones con datos de fecha y números mediante operadores aritméticos.
- Los nombres de columna, constantes numéricas y operadores aritméticos se pueden utilizar en expresiones aritméticas.
- Los operadores aritméticos se pueden utilizar en cualquier cláusula de sentencia SQL, excepto FROM.

Operador	Descripción
+	Sumar
-	Restar
*	Multiplicar
/	Dividir

Nota: Con los tipos de dato `DATE` y `TIMESTAMP`, solo puede utilizar operadores de suma y resta.

Uso de Operadores Aritméticos

- En este caso, el operador de suma se utiliza para calcular un aumento de salario de 300 \$ para todos los empleados. `SALARY + 300` se muestra como cabecera de columna.

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

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



DFo 6-6
Recuperación de datos mediante SELECT

Copyright © 2017, Oracle y/o sus filiales. Todos los derechos reservados. 14

La columna calculada resultante, `salary + 300`, no es una nueva columna en la tabla `EMPLOYEES`; es solo para visualización. Por defecto, el nombre de una nueva columna procede del cálculo que lo genera (en este caso, `salary + 300`).

Nota: El servidor de Oracle ignora los espacios en blanco antes y después del operador aritmético.

Reglas de prioridad

- La multiplicación y división se evalúan antes de la suma y la resta.
- Los operadores de la misma prioridad se evalúan de izquierda a derecha.
- Los paréntesis se utilizan para sustituir la prioridad por defecto o para aclarar la sentencia.

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	52800

- Utilice los paréntesis para reforzar el orden estándar de prioridad y mejorar la claridad

```
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	52800

- Puede sustituir las reglas de prioridad utilizando paréntesis para especificar el orden en el que se ejecutan los operadores.

El primer ejemplo de la diapositiva muestra el apellido, el salario y la compensación anual de los empleados. Calcula la compensación anual multiplicando el salario mensual por 12, más un incentivo de 100 \$. La multiplicación se realiza antes que la suma.

El segundo ejemplo de la diapositiva muestra el apellido, el salario y la compensación anual de los empleados. Calcula la compensación anual sumando un incentivo mensual de 100 \$ al salario mensual y, a continuación, multiplicando ese subtotal por 12.

Debido a los paréntesis, la suma tiene prioridad sobre la multiplicación.

Definición de valor nulo

- Un valor nulo es un valor que no está disponible, sin asignar, desconocido o que no es aplicable.
- Un valor nulo no es lo mismo que un cero o un espacio en blanco.

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

LAST_NAME	JOB_ID	SALARY	COMMISSION_PCT	Gietz	AC_ACCOUNT	8300	-
King	AD_PRES	24000	-	Zlotkey	SA_MAN	10500	.2
Kochhar	AD_VP	17000	-	Abel	SA_REP	11000	.3
De Haan	AD_VP	17000	-	Taylor	SA_REP	8600	.2
Whalen	AD_ASST	4400	-	Grant	SA_REP	7000	.15
Higgins	AC_MGR	12000	-	Mourgos	ST_MAN	5800	-
Gietz	AC_ACCOUNT	8300	-	Rajs	ST_CLERK	3500	-
Zlotkey	SA_MAN	10500	.2	Davies	ST_CLERK	3100	-
Abel	SA_REP	11000	.3	Matos	ST_CLERK	2600	-

Las columnas de cualquier tipo de dato pueden contener valores nulos. Sin embargo, algunas restricciones (NOT NULL y PRIMARY KEY) evitan que se utilicen valores nulos en la columna.

Valores Nulos en Expresiones Aritméticas

- Cualquier expresión aritmética que contenga valores nulos se evalúa como nula.

```
SELECT last_name, 12*salary*commission_pct  
FROM employees;
```

LAST_NAME	12*SALARY*COMMISSION_PCT
King	-
Kochhar	-
De Haan	-
Whalen	-
Higgins	-
Gietz	-
Zlotkey	25200
Abel	39600

Gietz	-
Zlotkey	25200
Abel	39600
Taylor	20640
Grant	12600
Mourgos	-
Rajs	-
Davies	-
Matos	-

Definición de alias de columna

Un alias de columna:

- Cambia el nombre de una cabecera de columna
- Es útil para realizar cálculos
- Sigue inmediatamente al nombre de columna (también puede ser la palabra clave opcional `AS` entre el nombre de columna y el alias).
- Necesita comillas dobles si contiene espacios o caracteres especiales o si es sensible a mayúsculas/minúsculas.

Uso de alias de columna

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

- La palabra clave AS es opcional
- Los nombres de columna aparecen en mayúscula por defecto

NAME	COMM
King	-
Kochhar	-
De Haan	-
Whalen	-
Higgins	-

...

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

- Los nombres de columna entre paréntesis aparecerán como se hayan introducido

Name	Annual Salary
King	288000
Kochhar	204000
De Haan	204000
Whalen	52800
Higgins	144000

...



DFo 6-6
Recuperación de datos mediante SELECT

Copyright © 2017, Oracle y/o sus filiales. Todos los derechos reservados. 19

Nota: No se puede hacer referencia a un alias en la lista de columna que contiene la definición del alias. Por ejemplo, al escribir una sentencia SELECT del siguiente modo, se devuelve un error:

```
SELECT last_name "Name" , salary*12 "Annual Salary", Annual Salary +100  
FROM   employees;  
ORA-00904: "ANNUAL SALARY": invalid identifier
```

Operador de concatenación

- Enlaza columnas o cadenas de caracteres a otras columnas
- Se representa con dos barras verticales (||)
- Crea una columna que es una expresión de carácter

```
SELECT last_name||job_id AS "Employees"  
FROM employees;
```

- La concatenación de una sentencia NULL con un carácter da como resultado una cadena de caracteres.

Employees

KingAD_PRES

KochharAD_VP

De HaanAD_VP

WhalenAD_ASST

HigginsAD_MGR

Cadenas de caracteres literales

- Un literal es un carácter, un número o una fecha que se incluye en la sentencia `SELECT`.
- Los valores literales de caracteres y fecha se deben incluir entre comillas simples.
- Cada cadena de caracteres es la salida una vez para cada fila devuelta.

Uso de Cadenas de Caracteres Literales

```
SELECT last_name || ' is a ' || job_id  
       AS "Employee Details"  
FROM   employees;
```

Employee Details

King is a AD_PRES

Kochhar is a AD_VP

De Haan is a AD_VP

Whalen is a AD_ASST

Higgins is a AC_MGR

Gietz is a AC_ACCOUNT

Zlotkey is a SA MAN

- En el ejemplo, el apellido y el job_id de cada empleado se concatenan con un literal para proporcionar a las filas devueltas más significado.

Operador de comillas (q) alternativo

```
SELECT department_name || q'[ Department's Manager Id: ]'  
      || manager_id  
      AS "Department and Manager"  
FROM departments;
```

Department and Manager

Administration Department's Manager Id: 200
Marketing Department's Manager Id: 201
Shipping Department's Manager Id: 124
IT Department's Manager Id: 103
Sales Department's Manager Id: 149
Executive Department's Manager Id: 100
Accounting Department's Manager Id: 205

- Muchas de las sentencias SQL utilizan literales de caracteres en expresiones o condiciones. Si el propio literal contiene una comilla simple, puede utilizar el operador de comillas (q) y seleccionar su propio delimitador de entrecomillado, en este caso, los corchetes []

Puede seleccionar cualquier delimitador adecuado, ya sea de un solo byte o multibyte, o bien cualquiera de los siguientes pares de caracteres:

[], { }, () o < >.

Escenario de caso: Uso del alias de columna



Profesor

Sean, me gustaría ver las distintas ubicaciones en las que se encuentran los miembros.

Puedo crear una consulta simple mediante la sentencia **SELECT** y mostrar esa información.



Alumno

Escenario de caso: Uso de la sentencia SELECT

En este caso, se ha utilizado el operador de concatenación y el alias de columna.



Recuperación correcta de los datos

```
SELECT LAST_NAME || ' ' || FIRST_NAME ||  
       ' IS LOCATED IN ' || CITY AS "MEMBER LOCATION"  
FROM MEMBERS;
```

MEMBER LOCATION
Urguhart Molly IS LOCATED IN Quebec
Biri Ben IS LOCATED IN Columbus
Valasquez Carmen IS LOCATED IN Seattle
Menchu Roberta IS LOCATED IN Brussels

Ejercicio 2 del proyecto

DFo_6_6_2_Project

Base de datos de la tienda Oracle Baseball League

Escritura y ejecución de sentencias SELECT:

- Usar operadores aritméticos
- Usar alias de columna
- Usar cadenas de caracteres literales

Filas Duplicadas

- La visualización por defecto de las consultas incluye todas las filas, también las filas duplicadas.
- Para eliminar filas duplicadas en el resultado, incluya la palabra clave `DISTINCT` en la cláusula `SELECT` inmediatamente después de la palabra clave `SELECT`

```
SELECT department_id  
FROM employees;
```

DEPARTMENT_ID
90
90
90
10
110
110

```
SELECT DISTINCT department_id  
FROM employees;
```

DEPARTMENT_ID
-
90
20
110
80
50



DFo 6-6
Recuperación de datos mediante SELECT

Copyright © 2017, Oracle y/o sus filiales. Todos los derechos reservados. 27

Puede especificar varias columnas después del cualificador `DISTINCT`. Ese cualificador afecta a todas las columnas seleccionadas y el resultado que se muestra es una combinación distinta de las columnas.

```
SELECT DISTINCT department_id, job_id  
FROM employees;
```

Nota: También puede especificar la palabra clave `UNIQUE`, que es sinónimo de la palabra clave `DISTINCT`.

Visualización de la Estructura de Tabla

- Utilice el comando `DESCRIBE` para mostrar la estructura de una tabla, incluidos el nombre de columna, tipo de dato y nulidad.
- O seleccione el explorador de objetos de APEX en el taller de SQL para ver la estructura de la tabla.
- Puede hacer clic en el botón Buscar tablas en los comandos SQL de APEX para ver las estructuras de tablas también.

```
DESC[RIBE] tablename
```

Uso del comando DESCRIBE

DESCRIBE employees

Object Type **TABLE** ⓘ

Object **EMPLOYEES** ⓘ

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMPLOYEES	EMPLOYEE_ID	NUMBER	-	6	0	1	-	-	-
	FIRST_NAME	VARCHAR2	20	-	-	-	✓	-	-
	LAST_NAME	VARCHAR2	25	-	-	-	-	-	-
	EMAIL	VARCHAR2	25	-	-	-	-	-	-
	PHONE_NUMBER	VARCHAR2	20	-	-	-	✓	-	-
	HIRE_DATE	DATE	7	-	-	-	-	-	-
	JOB_ID	VARCHAR2	10	-	-	-	-	-	-
	SALARY	NUMBER	-	8	2	-	✓	-	-
	COMMISSION_PCT	NUMBER	-	2	2	-	✓	-	-
	MANAGER_ID	NUMBER	-	6	0	-	✓	-	-
	DEPARTMENT_ID	NUMBER	-	4	0	-	✓	-	-
	BONUS	VARCHAR2	5	-	-	-	✓	-	-

Resumen

En esta lección, debe haber aprendido lo siguiente:

- Enumerar las capacidades de las sentencias SQL `SELECT`
- Escribir y ejecutar una sentencia `SELECT` que:
 - Devuelve todas las filas y columnas de una tabla
 - Devuelve columnas específicas de una tabla
 - Utiliza alias de columna para mostrar cabeceras de columna descriptivas



Resumen

En esta lección, debe haber aprendido lo siguiente:

- Escribir y ejecutar una sentencia `SELECT` que:
 - Utiliza operadores aritméticos y de concatenación
 - Utiliza cadenas de caracteres literales
 - Elimina filas duplicadas
- Describir la estructura de una tabla





ACADEMY