

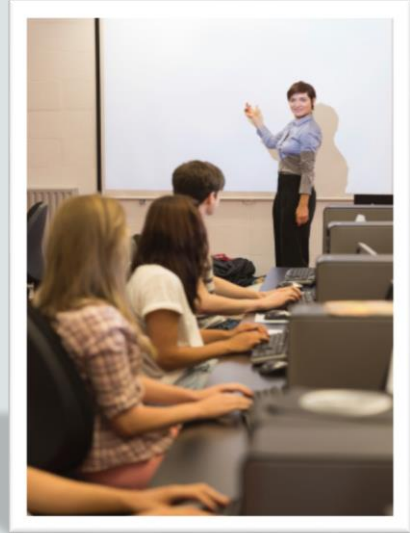


ACADEMY

Programación de Bases de Datos con SQL

9-3

Uso de los Operadores SET



ORACLE ACADEMY

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

Objetivos

En esta lección se abordan los siguientes objetivos:

- Definir y explicar el objetivo de los operadores SET
- Utilizar un operador SET para combinar varias consultas en una sola
- Controlar el orden de las filas devueltas al utilizar los operadores SET

Objetivo

- Los operadores SET se utilizan para combinar los resultados de diferentes sentencias SELECT en una sola salida de resultado.
- A veces desea una sola salida de más de una tabla.
- Si une las tablas, se devuelven las filas que cumplan los criterios de unión, pero ¿qué sucede si una unión devuelve un juego de resultados que no responde a sus necesidades?
- Aquí es donde entran los operadores SET.
- Pueden devolver las filas que se han encontrado en varias sentencias SELECT, las filas que estén en una tabla y no en otra, o bien las filas comunes en ambas sentencias.

Definición de etapa

- Para explicar los operadores SET, se hará referencia a las dos listas siguientes a lo largo de esta lección:

$A = \{1, 2, 3, 4, 5\}$

$B = \{4, 5, 6, 7, 8\}$

- O, en realidad: dos tablas, una llamada A y otra llamada B.

A	A_ID	B	B_ID
	1		4
	2		5
	3		6
	4		7
	5		8

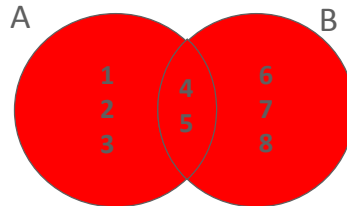
Reglas que Recordar

- Existen algunas reglas que se deben tener en cuenta al utilizar los operadores SET:
 - El número de columnas y los tipos de dato de las columnas debe ser idéntico en todas las sentencias SELECT utilizadas en la consulta.
 - No es necesario que los nombres de las columnas sean idénticos.
 - Los nombres de columna de la salida se toman de los nombres de columnas de la primera sentencia SELECT.
- Por tanto, los alias de columna se deben introducir en la primera sentencia, ya que desearía verlos en el informe terminado.

UNION

- El operador UNION devuelve todas las filas de ambas tablas, después de eliminar los duplicados.

```
SELECT a_id  
FROM   a  
UNION  
SELECT b_id  
FROM   b;
```

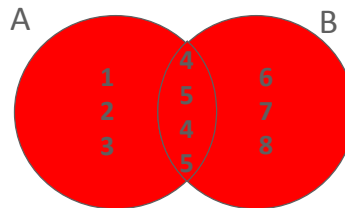


- El resultado de mostrar todos los elementos de A y B eliminando los duplicados es {1, 2, 3, 4, 5, 6, 7, 8}.
- Si ha unido A y B, solo obtendría {4, 5}. Tendría que realizar una unión externa completa para obtener la misma lista que anteriormente.

UNION ALL

- El operador UNION ALL devuelve todas las filas de ambas tablas, sin eliminar los duplicados.

```
SELECT a_id  
FROM   a  
UNION  ALL  
SELECT b_id  
FROM   b;
```

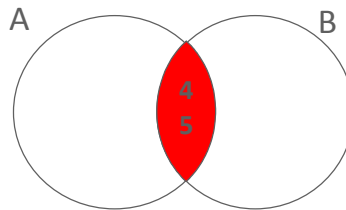


- El resultado de mostrar todos los elementos de A y B sin eliminar los duplicados es {1, 2, 3, 4, 5, 4, 5, 6, 7, 8}.

INTERSECT

- El operador INTERSECT devuelve todas las filas comunes a ambas tablas.

```
SELECT a_id  
FROM   a  
INTERSECT  
SELECT b_id  
FROM   b;
```

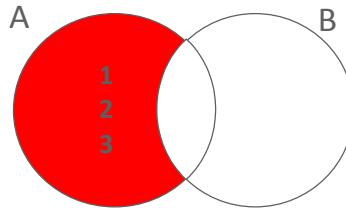


- El resultado de mostrar todos los elementos encontrados tanto en A como en B es {4, 5}.

MINUS

- El operador MINUS devuelve todas las filas encontradas en una tabla, pero no en la otra.

```
SELECT a_id  
FROM   a  
MINUS  
SELECT b_id  
FROM   b;
```



- El resultado de mostrar todos los elementos encontrados en A pero no en B es {1, 2, 3}.
- El resultado de B MINUS A produciría {6, 7, 8}.

Ejemplos de Operadores SET

- A veces, si está seleccionando filas de las tablas que no tengan columnas en común, puede que tenga que crear sus propias columnas en orden para que coincidan con el número de columnas de las consultas.
- La forma más sencilla de hacerlo es incluir uno o más valores NULL en la lista de selección.
- Recuerde que debe asignar a cada uno un alias adecuado y un tipo de dato coincidente.

Ejemplos de Operadores Set

- Por ejemplo:
 - La tabla employees contiene una fecha de contratación, un ID de empleado y un ID de cargo
 - La tabla de historial de cargos contiene el ID de empleado y el ID de cargo, pero no tiene una columna con fecha de contratación.
 - Las dos tablas comparten ID de empleado e ID de cargo, pero el historial de cargos no tiene fecha de inicio.
- Puede utilizar la función TO_CHAR(NULL) para crear columnas coincidentes, como en el siguiente diapositiva.

Ejemplos de Operadores Set

```
SELECT hire_date, employee_id, job_id
FROM employees
UNION
SELECT TO_DATE(NULL), employee_id, job_id
FROM job_history;
```

HIRE_DATE	EMPLOYEE_ID	JOB_ID
17-Jun-1987	100	AD_PRES
17-Sep-1987	200	AD_ASST
21-Sep-1989	101	AD_VP
03-Jan-1990	103	IT_PROG
21-May-1991	104	IT_PROG
13-Jan-1993	102	AD_VP
07-Jun-1994	205	AC_MGR
07-Jun-1994	206	AC_ACCOUNT
17-Oct-1995	141	ST_CLERK
17-Feb-1996	201	MK_MAN
11-May-1996	174	SA_REP
29-Jan-1997	142	ST_CLERK
17-Aug-1997	202	MK_REP
15-Mar-1998	143	ST_CLERK
24-Mar-1998	176	SA_REP
09-Jul-1998	144	ST_CLERK
07-Feb-1999	107	IT_PROG
24-May-1999	178	SA_REP
16-Nov-1999	124	ST_MAN
29-Jan-2000	149	SA_MAN
-	101	AC_ACCOUNT
-	101	AC_MGR
-	102	IT_PROG
-	114	ST_CLERK
-	122	ST_CLERK
-	176	SA_MAN
-	176	SA_REP
-	200	AC_ACCOUNT
-	200	AD_ASST
-	201	MK_REP

La tabla job_history contiene 10 filas, por lo que, cuando se agrega a las 20 filas de la tabla employees, se devuelven 30 filas. Las filas con valores NULL para hire_date proceden de la tabla job_history.

Ejemplos de Operadores Set

- La palabra clave NULL se puede utilizar para que coincida con las columnas de una lista SELECT.
- Se incluye un valor NULL para cada columna que falte.
- Además, se formatea el valor NULL para que coincida con el tipo de dato de la columna a la que representa, por lo que las funciones TO_CHAR, TO_DATE o TO_NUMBER se utilizan para lograr listas SELECT idénticas.

ORDER BY en Operaciones SET

- Si desea controlar el orden de las filas devueltas al utilizar operadores SET en la consulta, la sentencia ORDER BY solo se debe utilizar una vez, en la última sentencia SELECT de la consulta.
- Mediante el ejemplo de la consulta anterior, podríamos utilizar ORDER BY en employee_id para ver los cargos que haya ocupado cada empleado.

```
SELECT hire_date, employee_id, job_id
FROM employees
UNION
SELECT TO_DATE(NULL), employee_id, job_id
FROM job_history
ORDER BY employee_id;
```

ORDER BY en Operaciones SET

```
SELECT hire_date, employee_id, job_id
FROM employees
UNION
SELECT TO_DATE(NULL), employee_id, job_id
FROM job_history
ORDER BY employee_id;
```

HIRE_DATE	EMPLOYEE_ID	JOB_ID
17-Jun-1987	100	AD_PRES
21-Sep-1989	101	AD_VP
-	101	AC_ACCOUNT
-	101	AC_MGR
13-Jan-1993	102	AD_VP
-	102	IT_PROG
03-Jan-1990	103	IT_PROG
21-May-1991	104	IT_PROG
07-Feb-1999	107	IT_PROG
-	114	ST_CLERK
...

Nota: Para facilitar la lectura, solo se muestra parte del resultado de la consulta.

ORDER BY en Operaciones SET

- Podríamos mejorar la capacidad de lectura de la salida, incluyendo las columnas de fecha de inicio y fecha de finalización de la tabla de historial de cargos; para ello, sería necesario hacer coincidir las columnas de ambas consultas, agregando dos columnas TO_DATE(NULL) más a la primera consulta.

```
SELECT hire_date, employee_id, TO_DATE(null) start_date,  
       TO_DATE(null) end_date, job_id, department_id  
FROM   employees  
UNION  
SELECT TO_DATE(null), employee_id, start_date, end_date, job_id,  
       department_id  
FROM   job_history  
ORDER BY employee_id;
```

Como las cabeceras de columna de la salida de la consulta se obtienen de la primera consulta, se les han proporcionado alias con el mismo nombre que las columnas coincidentes de la segunda consulta.

ORDER BY en Operaciones SET

HIRE_DATE	EMPLOYEE_ID	START_DATE	END_DATE	JOB_ID	DEPARTMENT_ID
17-Jun-1987	100	-	-	AD_PRES	90
21-Sep-1989	101	-	-	AD_VP	90
-	101	21-Sep-1989	27-Oct-1993	AC_ACCOUNT	110
-	101	28-Oct-1993	15-Mar-1997	AC_MGR	110
13-Jan-1993	102	-	-	AD_VP	90
-	102	13-Jan-1993	24-Jul-1998	IT_PROG	60
03-Jan-1990	103	-	-	IT_PROG	60
21-May-1991	104	-	-	IT_PROG	60
07-Feb-1999	107	-	-	IT_PROG	60
-	114	24-Mar-1998	31-Dec-1999	ST_CLERK	50
-	122	01-Jan-1999	31-Dec-1999	ST_CLERK	50
16-Nov-1999	124	-	-	ST_MAN	50
17-Oct-1995	141	-	-	ST_CLERK	50
29-Jan-1997	142	-	-	ST_CLERK	50
15-Mar-1998	143	-	-	ST_CLERK	50
...

Nota: Para facilitar la lectura, solo se muestra parte del resultado de la consulta.

Terminología

Entre los términos clave utilizados en esta lección se incluyen:

- INTERSECT
- MINUS
- Operadores SET
- TO_CHAR(null): que coincide con la lista SELECT
- UNION
- UNION ALL

Resumen

En esta lección, debe haber aprendido lo siguiente:

- Definir y explicar el objetivo de los operadores SET
- Utilizar un operador SET para combinar varias consultas en una sola
- Controlar el orden de las filas devueltas al utilizar los operadores SET



 **ACADEMY**