



Autor: **Nicola Strappazon**
Web: <http://nicola.strappazon.me>
Revisión: 17/11/11

Sección 7

Usando los operadores de conjunto

Este material se encuentra basado en el curso de Fundamentos a SQL de ORACLE, el cual es adaptado para el producto PostgreSQL, todos los ejemplos, códigos fuentes y la Base de Datos HR es propiedad de ORACLE.

Objetivos

Al completar esta lección usted podrá entender los siguientes puntos:

- Describir cuales son los diversos tipos de operadores de conjunto.
- Como usar los operadores de conjunto en múltiples consultas.
- Controlar el orden de como son retornados los registros

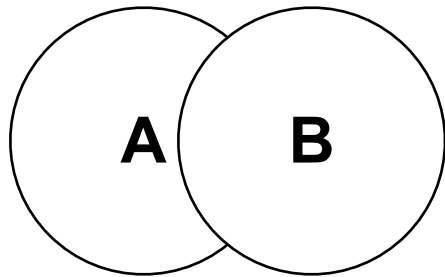
Consideraciones de los operadores

- La lógica de estos tipos de operadores nace de la teoría de conjuntos.
- Permite unir dos o mas consultas en una sola.
- Cada consulta debe retornar la misma cantidad de columnas y del mismo tipo de dato.
- No importa si el nombre de las columna son diferentes, siempre se muestran los nombres de la primera consulta..
- No importa la cantidad de registros retornados.
- Puede utilizar la cláusula **ORDER BY** o **LIMIT**, al final de la ultima consulta, esto afecta a todo el resultado.

¿Cuándo se usan los operadores?

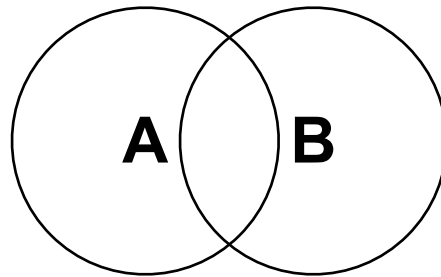
- Reportes donde se requiere complementar los datos mediante totales, estos pueden estar divididos por grupos.
- Construir un reporte que consolide un solo resultado desde diversas fuentes.

Tipos de operadores de conjuntos



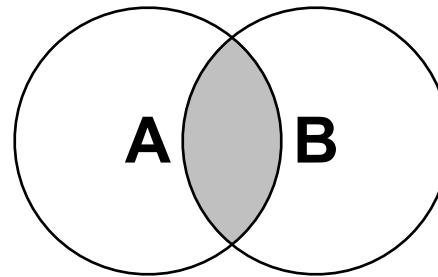
UNION

Una distintas consultas, no muestra los registros duplicados.



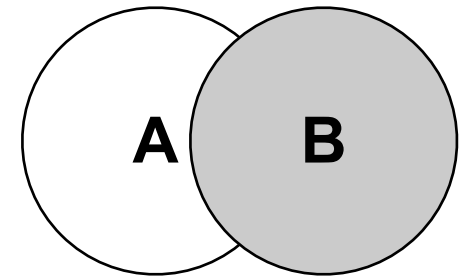
UNION ALL

Una distintas consultas, si muestra los registros duplicados.



INTERSECT

Una distintas consultas, retorna únicamente los registros que son duplicados.



EXCEPT

Una distintas consultas, y solo muestra los registros de A y no los de B.

Usando el operador UNION

Lista los códigos de cada empleado, con su cargo actual y su historio.

```
SELECT employee_id, job_id
FROM employees
UNION
SELECT employee_id, job_id
FROM job_history;
```

employee_id	job_id
100	AD_PRES
101	AC_ACCOUNT
101	AC_MGR
101	AD_VP
102	AD_VP
...	

(105 rows)

Usando el operador UNION ALL

Lista los códigos de cada empleado, con su cargo actual y su historio, sin importar que se repitan.

```
SELECT employee_id, job_id, department_id
FROM   employees
UNION ALL
SELECT employee_id, job_id, department_id
FROM   job_history
ORDER BY employee_id;
```

employee_id	job_id	department_id
200	AC_ACCOUNT	90
200	AD_ASST	10
200	AD_ASST	90
201	MK_REP	20

(117 rows)

Usando el operador INTERSECT

Lista los códigos de cada empleado, con su cargo actual y su historio, siempre y cuando se repitan.

```
SELECT employee_id, job_id
FROM   employees
INTERSECT
SELECT employee_id, job_id
FROM   job_history
```

employee_id	job_id
176	SA_REP
200	AD_ASST

(2 rows)

Usando el operador EXCEPT

Lista los códigos de cada empleado, con su cargo actual y su historio, siempre y cuando no existan en la segunda consulta.

```
SELECT employee_id, job_id
FROM   employees
EXCEPT
SELECT employee_id, job_id
FROM   job_history;
```

employee_id	job_id
199	SH_CLERK
201	MK_MAN
202	MK_REP
203	HR_REP

(105 rows)

Manejo de columnas no existentes

- Puede remplazar el nombre de una columna por un valor por defecto.
- Los valores serán colocados en la consulta que no tiene la columna a mostrar.
- Estos valores deben ser del tipo de dato a la columna que hacemos referencia, como por ejemplo:
 - Utilizamos `null` como un valor genérico, el valor 0 u otro tipo de literales.

Utilizando el valor null

```
SELECT department_id, hire_date, null
FROM   employees
UNION
SELECT department_id, null, location_id
FROM   departments
LIMIT 100;
```

department_id	hire_date	?column?
10	1987-09-17	
10		1700
20	1996-02-17	
20	1997-08-17	
20		1800
30	1994-12-07	
...		

(100 rows)

Utilizando valores por defecto

```
SELECT employee_id, job_id, salary
FROM   employees
UNION
SELECT employee_id, job_id, 0
FROM   job_history;
```

employee_id	job_id	salary
100	AD_PRES	24000
101	AC_ACCOUNT	0
101	AC_MGR	0
101	AD_VP	17000
102	AD_VP	17000
102	IT_PROG	0
103	IT_PROG	9000
...		

(130 rows)

Resumen

En esta lección, usted debió entender como:

- Usar el operador **UNION** para retornar diferentes registros.
- Usar el operador **UNION ALL** para retornar todos los registros incluyendo los duplicados.
- Usar el operador **INTERSECT** para retornar todos los registros comunes.
- Usar el operador **EXCEPT** para retornar todos los registros que son diferentes en la primera consulta, y estos no se encuentran en la segunda consulta.
- Ordenar los registros con la cláusula **ORDER BY** o definir un máximo con la cláusula **LIMIT**.