

Lenguaje SQL

Joins

Utilizando cláusulas SQL es posible recuperar datos combinados de varias tablas, uniéndolas entre sí de dos en dos.

En Oracle existen dos formas de hacer una unión entre tablas, la clásica, que está disponible en todas las versiones, pero no cumple con los estándares ANSI y la estándar, que se puede usar a partir de Oracle 9i y cumple con la regulación SQL:1999

Clásica	Estándar
Unión de Igualdad (Equijoin)	Cross Joins
Unión de no igualdad (Non-Equijoin)	Natural Joins
Unión externa (Outer Join)	Cláusula USING
Autounión (Self Join)	Full Outer Joins
	Outer Joins con condiciones arbitrarias

Producto Cartesiano

Cuando se intenta unir dos tablas y la condición no es válida o no existe, se produce un producto cartesiano, que consiste en una combinación de las filas de las dos tablas independientemente de si están relacionadas o no.

Para evitar este tipo de operaciones, que son tremendamente costosas para el servidor y ofrecen informaciones que en muy raras ocasiones son relevantes, no se debe omitir la cláusula WHERE y una condición válida al unir tablas entre sí.

```
SELECT *  
FROM pais, lugar;
```

ID_PAIS	NOMBRE	CONTINENTE	ID_LUGAR	ID_PAIS	NOMBRE	TIPO_LUGAR	HABITANTES	CLIMA
1	España	Europa	18	3	Capri	3	12200	-
1	España	Europa	19	3	Nápoles	5	-	Mediterráneo
1	España	Europa	83	3	Pisa	1	88363	-
1	España	Europa	84	3	Venecia	1	268934	-
1	España	Europa	85	3	Córcega	5	281000	-
1	España	Europa	20	4	París	1	2153600	Semicontinental
1	España	Europa	21	4	Marsella	3	820900	-
1	España	Europa	22	4	Euro Disney	6	-	-
1	España	Europa	23	4	Versalles	1	85726	-

(...)

La consulta devolvería un registro de la tabla lugar por cada registro de la tabla país, aunque no haya relación entre ellos.

I.Unión de tablas con Sintaxis Oracle (Clásica)

```
SELECT tabla1.columna1, tabla2.columna2
FROM tabla1, tabla2
WHERE tabla1.columna1 = tabla2.columna2;
```

Para unir tablas utilizando la sintaxis propia de Oracle, se deben especificar en el FROM las tablas implicadas en la unión y en el WHERE la condición o condiciones de unión entre ellas. En la cláusula SELECT es recomendable especificar el nombre cualificado de las columnas recuperadas, es decir, se debe especificar el nombre de la columna, precedido por el de la tabla propietaria de la misma. En caso de que la columna tenga el mismo nombre en varias tablas, los nombres cualificados son obligatorios.

Si se quiere unir varias tablas, es necesario relacionarlas todas entre sí, es decir, que cada tabla aparezca al menos en alguna de las condiciones de unión.

Unión de Igualdad (Equijoin)

Este tipo de unión implica que los valores de las columnas utilizadas en la condición de unión deben ser iguales.

Es el tipo de unión que se suele producir entre tablas relacionadas por una clave primaria y una clave ajena.

```
SELECT lugar.nombre AS "Nombre del lugar",  
       pais.nombre as "Nombre del País", pais.continente  
FROM pais, lugar  
WHERE pais.id_pais = lugar.id_pais;
```

Nombre Del Lugar	Nombre Del País	CONTINENTE
Capri	Italia	Europa
Nápoles	Italia	Europa
Pisa	Italia	Europa
Venecia	Italia	Europa
Córcega	Italia	Europa
París	Francia	Europa
Marsella	Francia	Europa
Euro Disney	Francia	Europa
Versalles	Francia	Europa
Londres	Reino Unido	Europa

En la cláusula SELECT se especifican las columnas, alias y/o expresiones, cualificadas con el nombre de la tabla correspondiente. En la cláusula FROM se especifican las tablas implicadas, separadas por comas.

En la cláusula WHERE se especifica la condición o condiciones de unión entre las tablas, con los nombres de las columnas implicadas también cualificados.

Es posible especificar tantas cláusulas de restricción en el WHERE como se necesite, utilizando las cláusulas tipo AND y OR.

```
SELECT lugar.nombre AS "Nombre del lugar",
       pais.nombre as "Nombre del País", pais.continente
FROM pais, lugar
WHERE pais.id_pais = lugar.id_pais
AND continente IN ('América del Norte','Oceanía');
```

Nombre Del Lugar	Nombre Del País	CONTINENTE
Los Ángeles	Estados Unidos	América del Norte
Yosemite	Estados Unidos	América del Norte
Yucatán	Méjico	América del Norte
Sídney	Australia	Oceanía
Melbourne	Australia	Oceanía
Caims	Australia	Oceanía
Darwin	Australia	Oceanía
Ayers Rock	Australia	Oceanía
Kings Canyon	Australia	Oceanía
Yellowstone	Estados Unidos	América del Norte

Es posible simplificar la sintaxis de las consultas Join utilizando alias para las tablas implicadas, de forma que no es necesario escribir el nombre de la tabla para cualificar sus columnas, basta con especificar el alias.

```
SELECT l.nombre AS "Nombre del lugar",
       p.nombre as "Nombre del País", p.continente
FROM pais p, lugar l
WHERE p.id_pais = l.id_pais
AND p.continente IN ('América del Norte','Oceanía');
```

Nombre Del Lugar	Nombre Del País	CONTINENTE
Los Ángeles	Estados Unidos	América del Norte
Yosemite	Estados Unidos	América del Norte
Yucatán	Méjico	América del Norte
Sídney	Australia	Oceanía
Melbourne	Australia	Oceanía
Caims	Australia	Oceanía
Darwin	Australia	Oceanía
Ayers Rock	Australia	Oceanía
Kings Canyon	Australia	Oceanía
Yellowstone	Estados Unidos	América del Norte

Una vez que se especifica el alias para la tabla, no puede usarse el nombre original en esa consulta, debe utilizarse siempre el alias asignado.

Los alias pueden tener hasta 30 caracteres de largo, pero se recomienda utilizar nombres lo más cortos y descriptivos posible. Cuando se asigna un alias a una tabla para una SELECT, el alcance de dicho alias se restringe a la cláusula SELECT concreta, si se hace una nueva consulta, el alias no sería válido.

Unión de más de dos tablas.

Para unir un número **n** de tablas, es necesario tener, por lo menos **n-1** condiciones de unión entre ellas.

```
SELECT v.nombre, v.apellidos, vi.fecha, lu.nombre
FROM viajero v, viaje vi, lugar lu
WHERE v.id_viajero = vi.id_viajero
AND vi.id_lugar = lu.id_lugar
ORDER BY fecha;
```

NOMBRE	APELLIDOS	FECHA	NOMBRE
Lorena	Hernández	15/11/01	Kings Canyon
Diana	Menéndez	17/11/01	Florencia
Antonio	Domínguez	12/12/01	Segovia
Beatriz	González	28/12/01	Picos de Europa
Roberto	Fernández	15/01/02	Euro Disney
Damián	Rodríguez	20/01/02	Vladivostok
María	Muñoz	27/01/02	Rabat
Julián	Rodríguez	22/02/02	Boston
Sofía	Martínez	23/02/02	Sídney
Jose Luís	Suárez	08/03/02	Granada

Unión de no igualdad (Non-Equijoin)

Son un tipo de unión no muy utilizado, que une dos tablas por una condición diferente a la de igualdad

```
SELECT v.nombre, v.apellidos, vi.numero_dias, t.nombre
FROM viajero v, viaje vi, tipo_estancia t
WHERE v.id_viajero = vi.id_viajero
AND vi.numero_dias BETWEEN t.dia_desde AND t.dia_hasta
ORDER BY fecha;
```

NOMBRE	APELLIDOS	NUMERO_DIAS	NOMBRE
Lorena	Hernández	12	Vacaciones
Diana	Menéndez	4	Puente
Antonio	Domínguez	6	Viaje
Beatriz	González	3	Puente
Roberto	Fernández	9	Vacaciones
Damián	Rodríguez	6	Viaje
María	Muñoz	6	Viaje
Julián	Rodríguez	13	Vacaciones
Sofía	Martínez	9	Vacaciones
Jose Luís	Suárez	9	Vacaciones

Uniones Externas (Outer Join)

Los registros que no cumplen las condiciones de unión entre las tablas, no aparecen en los resultados de las consultas, sin embargo, es posible mostrar estos registros utilizando las uniones externas.

Estas uniones muestran tanto los registros que cumplen las condiciones como los que no las cumplen, mostrando la información que falta como valores nulos.

```
SELECT tabla1.columna1, tabla2.columna2
FROM tabla1, tabla2
WHERE tabla1.columna1(+) = tabla2.columna2;
```

```
SELECT tabla1.columna1, tabla2.columna2
FROM tabla1, tabla2
WHERE tabla1.columna1 = tabla2.columna2(+);
```

El operador que marca la unión externa en el modo SQL clásico de oracle es el (+), que se coloca junto a la condición de unión que tiene información insuficiente, es decir junto a la columna de la tabla débil.

```
SELECT p.nombre AS "País", l.nombre AS "Lugar"
FROM pais p, lugar l
WHERE p.id_pais = l.id_pais(+)
ORDER BY l.id_pais DESC;
```

País	Lugar
Nueva Zelanda	-
Ecuador	-
Argelia	-
Uruguay	-
Alemania	-
Hungría	Szeged
Hungría	Budapest
República Checa	Brno
República Checa	Praga
Tailandia	Bangkok

El operador de unión externa (+) solamente puede aparecer en un lado de la igualdad y no puede utilizarse con operadores IN ni condiciones OR.

```
SELECT p.nombre AS "País", l.nombre AS "Lugar"
FROM pais p, lugar l
WHERE p.id_pais = l.id_pais(+)
OR p.nombre = 'Ecuador'
ORDER BY l.id_pais DESC;
```

ORA-01719: operador de unión externa (+) no permitido en operando OR ni IN

Autouniones (Self Join)

En ocasiones puede ser necesario unir una tabla consigo misma para obtener un resultado determinado, para ello, se debe instar la tabla dos veces en la cláusula FROM, asignándole un alias diferente.

Por ejemplo, para obtener todos los viajeros del mismo perfil que Beatriz González, deberemos unir la tabla viajeros consigo misma:

```
SELECT v1.nombre, v1.apellidos, v1.perfil_viajero
FROM viajero v1, viajero v2
WHERE v1.perfil_viajero = v2.perfil_viajero
AND v2.nombre = 'Beatriz'
AND v2.apellidos = 'González';
```

NOMBRE	APELLIDOS	PERFIL_VIAJERO
Alberto	Gómez	CLASE_TURISTA
Marcos	Fernández	CLASE_TURISTA
Lorena	Hernández	CLASE_TURISTA
Beatriz	González	CLASE_TURISTA
Margarita	Sánchez	CLASE_TURISTA
Carmen	Ruiz	CLASE_TURISTA
Natalia	Fernández	CLASE_TURISTA
Oriol	Pérez	CLASE_TURISTA
Olga	González	CLASE_TURISTA
Alejandra	Martínez	CLASE_TURISTA

II.Unión de tablas usando sintaxis estándar (SQL: 1999)

```
SELECT tabla1.columna1, tabla2.columna2
FROM tabla1
[CROSS JOIN tabla2] | [NATURAL JOIN tabla2] |
[JOIN tabla2 USING (columna2)] |
[JOIN tabla2
    ON (tabla1.columna1 = tabla2.columna2)] |
[LEFT | RIGHT | FULL OUTER JOIN tabla2
    ON (tabla1.columna1 = tabla2.columna2)];
```

Los resultados obtenidos con ambas sintaxis son equivalentes.

- **CROSS JOIN** devuelve un producto cartesiano de las dos tablas.
- **NATURAL JOIN** une dos tablas basándose en que tengan el mismo nombre de columna.
- **JOIN** realiza una unión entre las dos tablas especificando el nombre de la columna que las une. Se puede expresar como **INNER JOIN**.
- **LEFT / RIGHT / FULL OUTER JOIN** realiza una unión externa entre las dos tablas especificando el nombre de la columna de la unión.

CROSS JOIN

Equivale al producto cartesiano en la unión de dos tablas.

```
SELECT l.nombre, p.nombre
FROM lugar l CROSS JOIN pais p;
```

NOMBRE	NOMBRE
Capri	España
Nápoles	España
Pisa	España
Venecia	España
Córcega	España
París	España
Marsella	España
Euro Disney	España
Versalles	España
Londres	España

NATURAL JOIN

Este tipo de unión se basa en que el nombre de las columnas de las tablas implicadas sea el mismo, además, los tipos de dato de ambas columnas tiene que ser también equivalente, de lo contrario, se producirá un error.

```
SELECT v.nombre, v.apellidos, vi.numero_dias
FROM viajero v NATURAL JOIN viaje vi;
```

NOMBRE	APELLIDOS	NUMERO_DIAS
Carmen	Ruiz	5
Alberto	Gómez	3
Beatriz	González	11
Carlos	Vázquez	8
Susana	Díaz	11
Roberto	Fernández	13
Julián	Rodríguez	12
Sofía	Martínez	6
Jose Luis	Suárez	9
Julián	Rodríguez	8

Es un modo de unión que produce algunos resultados erróneos, sobre todo cuando se añaden o renombran columnas, y no permite realizar uniones de más de dos tablas ya que se corre el riesgo de obtener productos cartesianos. Es mejor evitar esta cláusula en lo posible.

Cláusula USING

Con la cláusula USING, se puede especificar manualmente las columnas de las tablas que están implicadas en una JOIN.

```
SELECT p.nombre, l.nombre  
FROM pais p JOIN lugar l USING(id_pais);
```

NOMBRE	NOMBRE
Italia	Capri
Italia	Nápoles
Italia	Pisa
Italia	Venecia
Italia	Córcega
Francia	París
Francia	Marsella
Francia	Euro Disney
Francia	Versalles
Reino Unido	Londres

Ni la NATURAL JOIN ni la cláusula USING permiten que se cualifique la columna o columnas que relacionan las tablas, ya que se selecciona automáticamente la tabla usada:

```
SELECT p.id_pais, p.nombre, l.nombre  
FROM pais p JOIN lugar l USING(id_pais);
```

ORA-25154: la parte de columna de la cláusula USING no puede tener un cualificador

Si es necesario unir por varias columnas dentro de la cláusula USING, se deben especificar separadas por comas.

Cláusula ON

Con la cláusula ON, además de las columnas implicadas, permite modificar la condición o condiciones de unión

```
SELECT l.nombre, p.nombre  
FROM lugar l JOIN pais p ON (l.id_pais = p.id_pais);
```

NOMBRE	NOMBRE
Capri	Italia
Nápoles	Italia
Pisa	Italia
Venecia	Italia
Córcega	Italia
París	Francia
Marsella	Francia
Euro Disney	Francia
Versalles	Francia
Londres	Reino Unido

Las cláusulas ON se pueden encadenar para unir varias tablas.

```
SELECT v.nombre, v.apellidos, vi. numero_dias, l.nombre
FROM viajero v
JOIN viaje vi ON (v.id_viajero = vi.id_viajero)
JOIN lugar l ON (vi.id_lugar = l.id_lugar);
```

NOMBRE	APELLIDOS	NUMERO_DIAS	NOMBRE
Carmen	Ruiz	5	Praga
Alberto	Gómez	3	Kyoto
Beatriz	González	11	San Diego
Carlos	Vázquez	8	Jersey
Susana	Díaz	11	Coímbra
Roberto	Fernández	13	Córdoba
Julián	Rodríguez	12	Darwin
Sofía	Martínez	6	Lisboa
Jose Luís	Suárez	9	Granada
Julián	Rodríguez	8	Miami

OUTER JOIN

Las uniones externas devuelven, además de los resultados que cumplen las condiciones establecidas, los resultados que no las cumplen, pudiendo especificar el sentido de la condición.

LEFT OUTER JOIN

Recupera todos los resultados de la tabla situada a la izquierda de la condición, aunque no haya coincidencias con la tabla que está a la derecha.

```
SELECT p.nombre AS "País", l.nombre AS "Lugar"
FROM pais p
LEFT OUTER JOIN lugar l ON (p.id_pais = l.id_pais)
ORDER BY id_lugar DESC;
```

País	Lugar
Nueva Zelanda	
Ecuador	
Argelia	
Uruguay	
Alemania	
Estados Unidos	Nueva York
Estados Unidos	Chicago
Estados Unidos	Yosemite
Estados Unidos	Miami
Estados Unidos	Las Vegas
Estados Unidos	Yellowstone
Estados Unidos	Washington

RIGHT OUTER JOIN

Recupera todos los resultados de la tabla situada a la derecha de la condición, aunque no haya coincidencias con la tabla que está a la izquierda. L

```
SELECT p.nombre AS "País", l.nombre AS "Lugar"
FROM pais p
RIGHT OUTER JOIN lugar l ON (p.id_pais = l.id_pais)
ORDER BY p.id_pais DESC;
```

País	Lugar
	Cataratas del Iguazú
	Ciudad del Cabo
	Ayers Rock
	Cataratas del Niágara
	San Diego
	Versalles
	Camino de Santiago
Hungría	Szeged
Hungría	Budapest
República Checa	Praga
República Checa	Brno

FULL OUTER JOIN

Recupera todas las filas de la tabla situada a la derecha de la sentencia aunque no haya coincidencia con la de la izquierda, y también recupera todos los registros de la tabla izquierda aunque no exista ninguna coincidencia con la derecha.

```
SELECT p.nombre AS "País", l.nombre AS "Lugar"  
FROM pais p  
FULL OUTER JOIN lugar l ON (p.id_pais = l.id_pais)  
ORDER BY p.id_pais DESC, id_lugar DESC;
```

País	Lugar
	San Diego
	Camino de Santiago
	Ayers Rock
	Ciudad del Cabo
Hungría	Szeged
Hungría	Budapest
República Checa	Brno
República Checa	Praga
Tailandia	Bangkok
Ecuador	
Nueva Zelanda	
Sudáfrica	
Argelia	