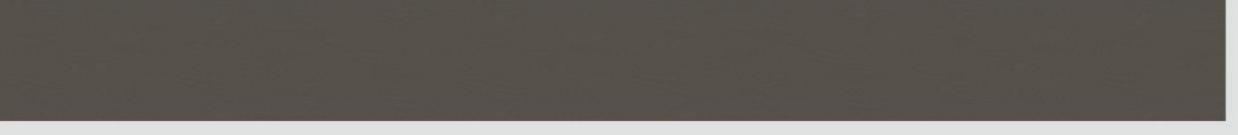




ORACLE

Academy



Database Foundations

6-9

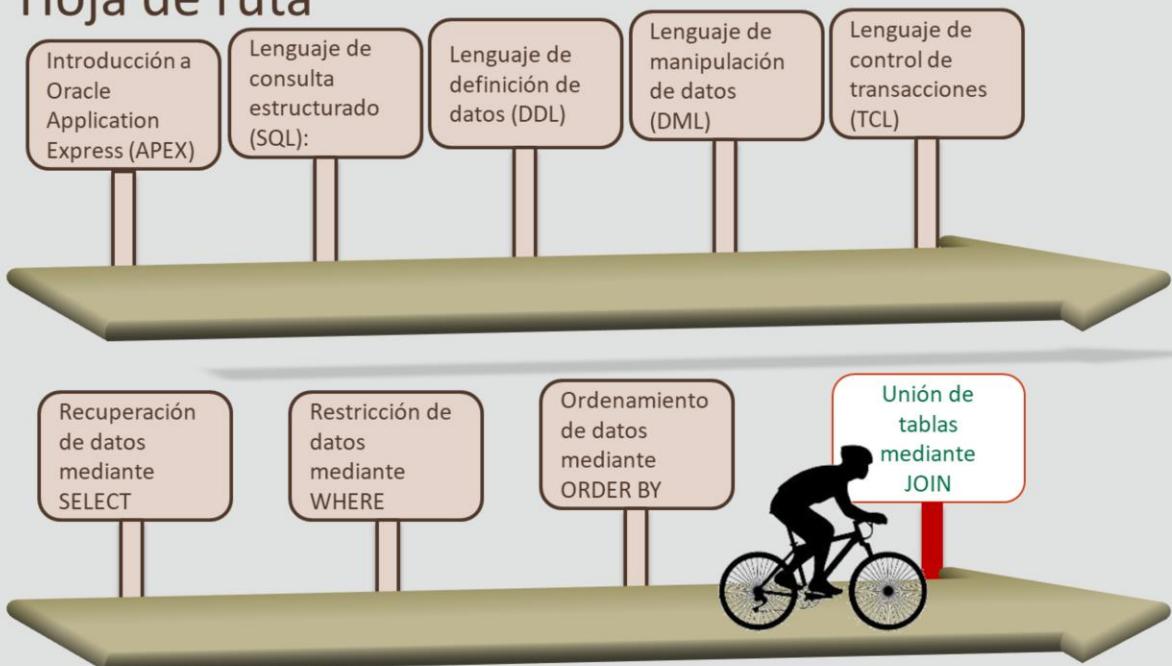
Unión de tablas mediante JOIN

ORACLE
Academy



Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Hoja de ruta



ORACLE
Academy

DFo 6-9
Unión de tablas mediante JOIN

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados. 3

Objetivos

- En esta lección se abordan los siguientes objetivos:
 - Escribir sentencias SELECT para acceder a datos de más de una tabla mediante uniones igualitarias y no igualitarias
 - Utilizar una unión automática para unir una tabla a sí misma
 - Utilizar uniones OUTER para visualizar datos que normalmente no cumplen una condición de unión
 - Generar un producto cartesiano (unión cruzada) de todas las filas de dos o más tablas



Obtención de datos de varias tablas

- A veces necesita utilizar datos de más de una tabla
- Para producir el informe, necesita enlazar las tablas EMPLOYEES y DEPARTMENTS, y acceder a los datos de ambas:

EMPLOYEES

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
100	King	90
149	Zlotkey	80
103	Ernst	60

...

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID
60	IT	1400
80	Sales	2500
90	Executive	1700

...

EMPLOYEE_ID	DEPARTMENT_ID	DEPARTMENT_NAME
100	90	Executive
149	80	Sales
102	60	IT

Tipos de uniones

- Uniones compatibles con el estándar SQL:1999:
 - Uniones naturales con la cláusula NATURAL JOIN
 - Uniones con la cláusula USING
 - Uniones con la cláusula ON
 - Uniones OUTER:
 - LEFT OUTER JOIN
 - RIGHT OUTER JOIN
 - FULL OUTER JOIN
 - CROSS JOIN

Unión de tablas mediante la sintaxis SQL:1999

- Utilizar una unión para consultar datos de más de una tabla:

```
SELECT    table1.column, table2.column
FROM      table1
[NATURAL JOIN table2] |
[JOIN table2 USING (column_name)] |
[JOIN table2
    ON (table1.column_name = table2.column_name)] |
[LEFT|RIGHT|FULL OUTER JOIN table2
    ON (table1.column_name = table2.column_name)] |
[CROSS JOIN table2];
```



En la sintaxis:

- table1.column indica la tabla y la columna desde las que se recuperan los datos.
- NATURAL JOIN une dos tablas basándose en el mismo nombre de la columna.
- JOIN table2 USING column_name realiza una unión igualitaria basándose en el nombre de la columna.
- JOIN table2 ON table1.column_name = table2.column_name realiza una unión igualitaria basándose en la condición de cláusula ON.
- LEFT/RIGHT/FULL OUTER se utiliza para realizar uniones OUTER.
- CROSS JOIN devuelve un producto cartesiano de las dos tablas.

Cualificación de nombres de columna ambiguos

- Utilizar prefijos de tabla para cualificar los nombres de columna que están en varias tablas y evitar que se produzcan ambigüedades
- Utilizar prefijos de tabla para aumentar la velocidad de análisis de la sentencia
- En lugar de prefijos de nombre de tabla completos, utilizar alias de tabla
- El alias de tabla proporciona a una tabla un nombre más corto, mantiene el código SQL más pequeño y utiliza menos memoria

Cualificación de nombres de columna ambiguos

- Utilizar alias de tabla para distinguir columnas que tienen nombres idénticos, pero que residen en diferentes tablas

```
SELECT e.first_name, d.department_name, d.manager_id  
FROM employees e JOIN departments d  
USING (department_id);
```

- Nota: Consulte las directrices de los alias de tabla en las notas de la diapositiva

FIRST_NAME	DEPARTMENT_NAME	MANAGER_ID
Jennifer	Administration	200
Michael	Marketing	201



Academy

DFo 6-9
Unión de tablas mediante JOIN

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados. 9

Directrices

- El nombre de tabla se especifica por completo, seguido de un espacio y del alias de tabla. Por ejemplo, a la tabla EMPLOYEES se le puede proporcionar el alias e, y a la tabla DEPARTMENTS el alias d.
- Los alias de tabla pueden tener hasta 30 caracteres de longitud, pero los alias más cortos son mejores que los largos.
- Si se utiliza un alias de tabla para un nombre de tabla determinado en la cláusula FROM, el alias de tabla se deberá sustituir por el nombre de tabla mediante la sentencia SELECT (pero no en la cláusula USING; esto se analizará más adelante).
- Los alias de tabla deben ser significativos.
- El alias de tabla es válido solo para la sentencia SELECT actual.

Creación de uniones naturales

- La cláusula NATURAL JOIN está basada en todas las columnas de las dos tablas que tienen el mismo nombre y el mismo tipo de dato
- Selecciona filas de las dos tablas que tienen valores iguales en todas las columnas coincidentes
- Si las columnas con los mismos nombres tienen tipos de datos diferentes, se devolverá un error

Recuperación de registros con uniones naturales

- Utiliza el único campo que es común a las dos tablas (DEPARTMENT_ID) para lleva a cabo la unión

```
SELECT department_id, department_name, location_id, city
FROM departments NATURAL JOIN locations;
```

DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID	CITY
20	Marketing	1800	Toronto
80	Sales	2500	Oxford
60	IT	1400	Southlake
50	Shipping	1500	South San Francisco
10	Administration	1700	Seattle
90	Executive	1700	Seattle

ORACLE ...

Academy

DFo 6-9
Unión de tablas mediante JOIN

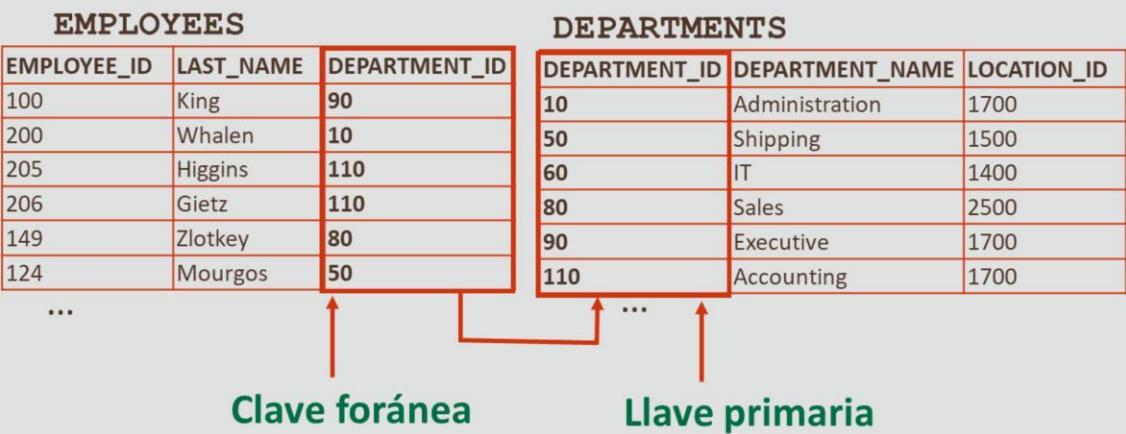
Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados. 11

Creación de uniones con la cláusula USING

- Si las tablas que se unen comparten varias columnas, se utilizarán todos los campos comunes en la unión
- Utilizar USING para especificar una única columna para una cláusula JOIN en lugar de una cláusula NATURAL JOIN
- La cláusula USING también se puede utilizar para hacer coincidir columnas con el mismo nombre, pero diferentes tipos de datos
- Las cláusulas NATURAL JOIN y USING se excluyen mutuamente

Unión de nombres de columna

- Los valores de la columna DEPARTMENT_ID deben ser iguales en ambas tablas. Esto se denomina una unión igualitaria (también denominada unión simple o interna)



Recuperación de registros con la cláusula USING

- La cláusula USING especifica que la unión se realiza con la columna DEPARTMENT_ID y no con MANAGER_ID, que también es una columna común

```
SELECT employee_id, last_name, location_id,  
       department_id  
  FROM employees JOIN departments  
USING (department_id);
```

EMPLOYEE_ID	LAST_NAME	LOCATION_ID	DEPARTMENT_ID
200	Whalen	1700	10
201	Hartstein	1800	20
202	Fay	1800	20
124	Mourgos	1500	50

...

ORACLE

Academy

DFo 6-9
Unión de tablas mediante JOIN

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados. 14

Uso de alias de tabla con la cláusula USING

- No utilizar un nombre o alias de la tabla en la cláusula USING
- Si la misma columna se utiliza en otro lugar de la sentencia SQL, no se le puede agregar un alias

```
SELECT l.city, d.department_name  
FROM locations l JOIN departments d  
USING (location_id)  
WHERE d.location_id = 1400;
```



ORA-25154: column part of USING clause cannot have qualifier

ORACLE

Academy

DFo 6-9

Unión de tablas mediante JOIN

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados. 15

Por ejemplo, la siguiente sentencia es válida:

```
SELECT l.city, d.department_name  
FROM locations l JOIN departments d  
USING (location_id)  
WHERE location_id = 1400;
```

Las columnas comunes a ambas tablas, pero que no se utilizan en la cláusula USING, deben tener como prefijo un alias de tabla, de lo contrario se devolverá el error "column ambiguously defined".

En la siguiente sentencia, manager_id está presente tanto en la tabla EMPLOYEES como en la tabla DEPARTMENTS; si manager_id no tiene un alias de tabla como prefijo, se devuelve el error "column ambiguously defined".

La siguiente sentencia es válida:

```
SELECT first_name, d.department_name, d.manager_id  
FROM employees e JOIN departments d USING (department_id)  
WHERE department_id = 50;
```

Creación de uniones con la cláusula ON

- Con la cláusula NATURAL JOIN se crea una unión igualitaria de todas las columnas con el mismo nombre y tipo de dato
- Utilizar la cláusula ON para especificar las condiciones arbitrarias o las columnas que se van a unir
- La condición de unión está separada de otras condiciones de búsqueda

Creación de uniones con la cláusula ON

- La cláusula ON facilita la comprensión del código
- La cláusula USING crea una unión igualitaria entre dos tablas utilizando una columna con el mismo nombre, independientemente del tipo de dato
- La cláusula ON crea una unión igualitaria entre dos tablas utilizando una columna de cada tabla independientemente del nombre o el tipo de dato

Recuperación de registros con la cláusula ON

- También puede utilizar la cláusula ON para unir columnas que tienen nombres o tipos de datos diferentes

```
SELECT e.employee_id, e.last_name, e.department_id,
       d.department_id, d.location_id
  FROM employees e JOIN departments d
 WHERE (e.department_id = d.department_id);
```

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_ID	LOCATION_ID
200	Whalen	10	10	1700
201	Hartstein	20	20	1800
202	Fay	20	20	1800
124	Mourgos	50	50	1500
141	Rajs	50	50	1500
142	Davies	50	50	1500



Academy

DFo 6-9
Unión de tablas mediante JOIN

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados. 18

Creación de uniones en 3 direcciones con la cláusula ON

- Debe haber 2 sentencias de unión al unir 3 tablas como se muestra a continuación:

```
SELECT employee_id, city, department_name
FROM employees e JOIN departments d
ON d.department_id = e.department_id
JOIN locations l
ON d.location_id = l.location_id;
```

EMPLOYEE_ID	CITY	DEPARTMENT_NAME
201	Toronto	Marketing
202	Toronto	Marketing
149	Oxford	Sales
174	Oxford	Sales
176	Oxford	Sales
103	Southlake	IT

ORACLE ...

Academy

DFo 6-9
Unión de tablas mediante JOIN

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados. 19

Una unión en tres direcciones es una unión de tres tablas. El optimizador decide la ejecución de la unión, así como el orden. Aquí, la primera unión que se realiza es EMPLOYEES JOIN DEPARTMENTS . La primera condición de unión puede hacer referencia a las columnas de EMPLOYEES y DEPARTMENTS , pero no puede hacer referencia a las columnas de LOCATIONS . La segunda condición de unión puede hacer referencia a las columnas de las tres tablas.

El ejemplo de código de la diapositiva también se puede realizar con la cláusula USING:

```
SELECT e.employee_id, l.city, d.department_name
FROM employees e
JOIN departments d
USING (department_id)
JOIN locations l
USING (location_id);
```

Escenario de caso: Cláusula ON



Recuperación de
datos de tres
tablas

```
SELECT b.title as "BOOK TITLE",
       a.name as "AUTHOR",
       t.id as "BOOK TRANSACTION"
    FROM authors a JOIN books b
      ON a.id = b.author_id
     JOIN book_transactions t
      ON b.id = t.book_id;
```

Recuperación correcta de datos
con la cláusula ON

BOOK_TITLE	AUTHOR	BOOK_TRANSACTION
The Clicking of Cuthbert	P.G. Wodehouse	OD0002
War and Peace	Leo Tolstoy	OD0001
An Unsocial Socialist	George Bernard Shaw	OD0003

Aplicación de condiciones adicionales a una unión

- Uso de la cláusula AND o WHERE para aplicar condiciones adicionales:

```
SELECT e.employee_id, e.last_name, e.department_id,
       d.department_id, d.location_id
  FROM employees e JOIN departments d
 ON      (e.department_id = d.department_id)
AND      e.manager_id = 149 ;
```

Or

```
SELECT e.employee_id, e.last_name, e.department_id,
       d.department_id, d.location_id
  FROM employees e JOIN departments d
 ON      (e.department_id = d.department_id)
WHERE    e.manager_id = 149 ;
```



Academy

DFo 6-9
Unión de tablas mediante JOIN

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados. 21

Ejercicio 1 del proyecto

- DFo_6_9_1_Project
 - Base de datos de la tienda Oracle Baseball League
 - Escriba sentencias SELECT con datos de varias tablas mediante uniones igualitarias y no igualitarias
 - Uniones naturales, cláusula USING y ON, uniones en 3 direcciones



ORACLE
Academy

DFo 6-9
Unión de tablas mediante JOIN

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados. 22

Unión de una tabla consigo misma

EMPLOYEES (WORKER)

EMPLOYEE_ID	LAST_NAME	MANAGER_ID
100	King	-
101	Kochhar	100
102	De Haan	100
200	Whalen	101
205	Higgins	101
206	Gietz	205
149	Zlotkey	100
174	Abel	149
176	Taylor	149
201	Hartstein	100
202	Fay	201

...

EMPLOYEES (MANAGER)

EMPLOYEE_ID	LAST_NAME
100	King
101	Kochhar
102	De Haan
200	Whalen
205	Higgins
206	Gietz
149	Zlotkey
174	Abel
176	Taylor
201	Hartstein
202	Fay

...

MANAGER_ID en la tabla WORKER es igual a
EMPLOYEE_ID en la tabla MANAGER



DFo 6-9
Unión de tablas mediante JOIN

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados. 23

Puede que a veces necesite unir una tabla consigo misma. Para buscar el nombre de cada supervisor del empleado, necesita unir la tabla EMPLOYEES consigo misma o realizar una unión automática. Por ejemplo, para buscar el nombre del supervisor de Kochar:

- Busque a Kochar en la tabla EMPLOYEES consultando la columna LAST_NAME .
- Busque el número del supervisor de Kochar consultando la columna MANAGER_ID . El número del supervisor de Kochar es 100.
- Busque el nombre del supervisor con un EMPLOYEE_ID de 100 consultando la columna LAST_NAME . El número de empleado de King es 100, por lo que King es el supervisor de Kochar.

Durante este proceso, busca dos veces en la tabla. La primera vez, cuando consulta la tabla para buscar a Kochar en la columna LAST_NAME y el valor MANAGER_ID de 100. La segunda vez, cuando consulta la columna EMPLOYEE_ID para buscar 100 y la columna LAST_NAME para buscar King.

Uniones automáticas con la cláusula ON

- La cláusula ON también se puede utilizar para unir columnas con nombres diferentes, ya sea en la misma tabla o en otra diferente

```
SELECT worker.last_name emp, manager.last_name mgr
FROM employees worker JOIN employees manager
ON (worker.manager_id = manager.employee_id);
```

EMP	MGR
Kochhar	King
De Haan	King
Zlotkey	King
Mourgos	King
Hartstein	King

...

ORACLE

Academy

DFo 6-9
Unión de tablas mediante JOIN

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados. 24

Uniones no igualitarias

- La tabla JOB_GRADES define el rango de valores de LOWEST_SAL y HIGHEST_SAL de cada GRADE_LEVEL

```
SELECT  *
FROM    job_grades;
```

GRADE_LEVEL	LOWEST_SAL	HIGHEST_SAL
A	1000	2999
B	3000	5999
C	6000	9999
D	10000	14999
E	15000	24999
F	25000	40000



Academy

DFo 6-9
Unión de tablas mediante JOIN

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados. 25

Una unión no igualitaria es una condición de unión que contiene algún operador diferente del operador de igualdad.

La relación entre la tabla EMPLOYEES y JOB_GRADES es un ejemplo de unión no igualitaria.

Los rangos de la columna SALARY en la tabla EMPLOYEES oscilan entre los valores en las columnas LOWEST_SAL y HIGHEST_SAL de la tabla JOB_GRADES. Por lo tanto, se pueden agregar grados a los empleados según su salario. La relación se obtiene mediante un operador distinto del operador de igualdad (=).

Uniones no igualitarias

- Por lo tanto, la columna GRADE_LEVEL se puede utilizar para asignar grados a cada empleado en función de su salario

EMPLOYEES

EMPLOYEE_ID	SALARY
100	24000
101	17000
102	17000
200	4400
205	12000
206	8300
149	10500
174	11000
176	8600
178	7000

JOB_GRADES

GRADE_LEVEL	LOWEST_SAL	HIGHEST_SAL
A	1000	2999
B	3000	5999
C	6000	9999
D	10000	14999
E	15000	24999
F	25000	40000

ORACLE ...

Academy

DFo 6-9
Unión de tablas mediante JOIN

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados. 26

Una unión no igualitaria es una condición de unión que contiene algún operador diferente del operador de igualdad.

La relación entre la tabla EMPLOYEES y JOB_GRADES es un ejemplo de unión no igualitaria.

Los rangos de la columna SALARY en la tabla EMPLOYEES oscilan entre los valores en las columnas LOWEST_SAL y HIGHEST_SAL de la tabla JOB_GRADES. Por lo tanto, se pueden agregar grados a los empleados según su salario. La relación se obtiene mediante un operador distinto del operador de igualdad (=).

Recuperación de registros con uniones no igualitarias

- Este ejemplo crea una unión no igualitaria para evaluar el grado de salario de un empleado. El salario debe estar entre cualquier par de los rangos de salario bajos y altos

```
SELECT e.last_name, e.salary, j.grade_level
FROM employees e JOIN job_grades j
ON e.salary BETWEEN j.lowest_sal AND j.highest_sal;
```

LAST_NAME	SALARY	GRADE_LEVEL
Vargas	2500	A
Matos	2600	A
Davies	3100	B
Rajs	3500	B
Lorentz	4200	B
Whalen	4400	B

ORACLE ...

Academy

DFo 6-9
Unión de tablas mediante JOIN

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados. 27

Es importante tener en cuenta que todos los empleados aparecerán solo una vez al ejecutar esta consulta. Ningún empleado se repite en la lista por los siguientes motivos:

- Ninguna de las filas de la tabla JOB_GRADES contiene grados que se solapen. Es decir, el valor de salario de un empleado solo puede oscilar entre los valores de salario bajo y alto de una de las filas de la tabla de grados de salario.
- Todos los salarios de los empleados oscilan entre los límites proporcionados por la tabla de grados de cargo. Es decir, ningún empleado gana menos que el valor más bajo de la columna LOWEST_SAL o más que el valor más alto de la columna HIGHEST_SAL.

En el ejemplo de la diapositiva se han especificado los alias de tabla por motivos de rendimiento, no por una posible ambigüedad.

Devolución de registros sin coincidencia directa con uniones OUTER

DEPARTMENTS

DEPARTMENT_NAME	DEPARTMENT_ID
Administration	10
Marketing	20
Shipping	50
IT	60
Sales	80
Executive	90
Accounting	110
Contracting	190

No hay ningún empleado en el departamento 190

Unión igualitaria con EMPLOYEES

DEPARTMENT_ID	LAST_NAME
90	King
90	Kochhar
90	De Haan
10	Whalen
80	Taylor
-	Grant
50	Mourgos
20	Fay

...

Al empleado "Grant" no se le ha asignado un ID de departamento



Academy

DFo 6-9
Unión de tablas mediante JOIN

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados. 28

Si una fila no cumple una condición de unión, la fila no aparece en el resultado de consultas.

En el ejemplo de la diapositiva, se utiliza una condición de unión igualitaria simple en las tablas EMPLOYEES y DEPARTMENTS para devolver el resultado a la derecha.

```
SELECT employees.department_id, department_name, last_name
from employees, departments
where employees.department_id = departments.department_id;
```

El juego de resultados no contiene:

- El ID de departamento 190, porque no hay ningún empleado con dicho ID registrado en la tabla EMPLOYEES
- El empleado con apellido Grant, porque a este empleado no se le ha asignado un ID de departamento

Para devolver el registro de departamento que no tiene ningún empleado, o los empleados que no tienen un departamento asignado, puede utilizar la unión OUTER.

Uniones INNER frente a uniones OUTER

- En SQL:1999, la unión de dos tablas que devuelven solo filas coincidentes se denomina unión INNER.
(Cláusulas NATURAL JOIN, USING, ON)
- Una unión entre dos tablas que devuelve los resultados de la unión INNER así como las filas no coincidentes de la tabla de la izquierda (o de la derecha) se denomina una unión OUTER izquierda (o derecha)
- Una unión entre dos tablas que devuelve los resultados de una unión INNER así como los resultados de una unión OUTER izquierda y derecha es una unión OUTER completa

ORACLE

Academy

DFo 6-9

Unión de tablas mediante JOIN

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados. 29

LEFT OUTER JOIN

- En este caso queremos ver los registros de todos los empleados (tabla de la izquierda) incluso si no están asignados a un departamento

```
SELECT e.last_name, e.department_id, d.department_name  
FROM employees e LEFT OUTER JOIN departments d  
ON (e.department_id = d.department_id);
```

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Whalen	10	Administration
Fay	20	Marketing
Hartstein	20	Marketing
Vargas	50	Shipping
Matos	50	Shipping
Higgins	110	Accounting
Grant	-	-

ORACLE

...

Academy

DFo 6-9
Unión de tablas mediante JOIN

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

30

RIGHT OUTER JOIN

- En este caso queremos ver todos los registros de departamento (tabla de la derecha) incluso si no contienen empleados

```
SELECT e.last_name, e.department_id, d.department_name
FROM employees e RIGHT OUTER JOIN departments d
ON (e.department_id = d.department_id);
```

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Whalen	10	Administration
Hartstein	20	Marketing
Fay	20	Marketing
Mourgos	50	Shipping
Rajs	50	Shipping
Davies	50	Shipping
-	-	Contracting



...

Academy

DFo 6-9
Unión de tablas mediante JOIN

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados. 31

FULL OUTER JOIN

- En este caso queremos ver todos los registros de empleado y todos los registros de departamento

```
SELECT e.last_name, e.department_id, d.department_name  
FROM employees e FULL OUTER JOIN departments d  
ON (e.department_id = d.department_id);
```

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
King	90	Executive
Kochhar	90	Executive
Taylor	80	Sales
Grant	-	-
Mourgos	50	Shipping
Fay	20	Marketing
-	-	Contracting

ORACLE

...

Academy

DFo 6-9

Unión de tablas mediante JOIN

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

32

Productos cartesianos

- Un producto cartesiano se genera cuando se muestran todas las combinaciones de filas. Todas las filas de la primera tabla se unen a todas las filas de la segunda tabla

```
SELECT last_name, department_name  
FROM employees, departments;
```

- Se genera un producto cartesiano cuando se omite una condición de unión o esta no es válida
- Se incluye siempre una condición de unión válida si desea evitar un producto cartesiano

```
SELECT last_name, department_name  
FROM employees e, departments d  
WHERE e.department_id = d.department_id;
```

ORACLE

Academy

DFo 6-9
Unión de tablas mediante JOIN

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados. 33

Un producto cartesiano suele generar un gran número de filas y el resultado es poco útil, salvo para determinadas pruebas en las que necesite generar un gran número de filas para simular una cantidad razonable de datos.

Generación de un producto cartesiano

EMPLOYEES (40 filas)

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
100	King	90
149	Zlotkey	80
103	Ernst	60

...

DEPARTMENTS (9 filas)

DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID
60	IT	1400
80	Sales	2500
90	Executive	1700

...

Producto cartesiano:
 $40 \times 9 = 360$ filas

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID
100	King	90	Administration	1700
101	Kochhar	90	Administration	1700
102	De Haan	90	Administration	1700
200	Whalen	10	Administration	1700
205	Higgins	110	Administration	1700
206	Gietz	110	Administration	1700
149**	Zlotkey	80	Administration	1700

Creación de uniones cruzadas

- La cláusula CROSS JOIN produce el producto combinado de dos tablas
- Esto también se denomina un producto cartesiano entre dos tablas

```
SELECT last_name, department_name  
FROM employees  
CROSS JOIN departments ;
```



DFo 6-9
Unión de tablas mediante JOIN

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados. 35

La técnica CROSS JOIN se puede aplicar de forma útil en algunas situaciones. Por ejemplo, para devolver el coste de mano de obra total por oficina por mes, incluso si el mes X no tiene ningún coste de mano de obra, puede realizar una unión cruzada de oficinas con la tabla de todos los meses.

Es una práctica aconsejable declarar de forma explícita CROSS JOIN en SELECT si desea crear un producto cartesiano. Por lo tanto, queda muy claro que desea que esto ocurra y que no se trata del resultado de la falta de uniones.

Ejercicio 2 del proyecto

- DFo_6_9_2_Project
 - Base de datos de la tienda Oracle Baseball League
 - Escriba sentencias SELECT con datos de varias tablas mediante uniones igualitarias y no igualitarias
 - Uniones automáticas, OUTER JOIN, productos cartesianos



ORACLE
Academy

DFo 6-9
Unión de tablas mediante JOIN

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados. 36

Resumen

- En esta lección, debe haber aprendido a hacer lo siguiente:
 - Escribir sentencias SELECT para acceder a datos de más de una tabla mediante uniones igualitarias y no igualitarias
 - Utilizar una unión automática para unir una tabla a sí misma
 - Utilizar uniones OUTER para visualizar datos que normalmente no cumplen una condición de unión
 - Generar un producto cartesiano (unión cruzada) de todas las filas de dos o más tablas



ORACLE

Academy