

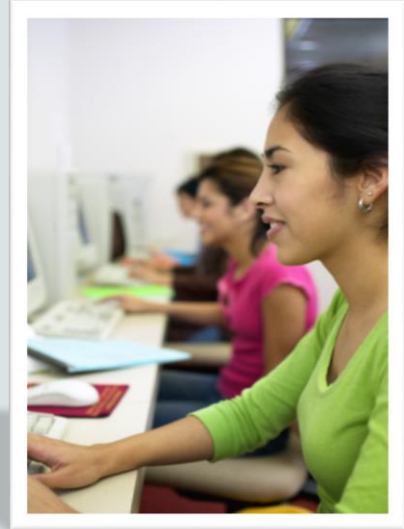


 **ACADEMY**

# Programación de Bases de Datos con SQL

7-1

Unión Igualitaria y  
Producto Cartesiano de Oracle



**ORACLE** ACADEMY

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

# Objetivos

En esta lección, aprenderá a:

- Nombrar las uniones propiedad de Oracle y sus equivalentes en ANSI/ISO SQL: 99
- Crear y ejecutar una sentencia SELECT que da como resultado un producto cartesiano
- Crear y ejecutar sentencias SELECT para acceder a los datos desde más de una tabla utilizando una unión igualitaria
- Crear y ejecutar sentencias SELECT que agregan condiciones de búsqueda usando el operador AND
- Aplicar la regla para utilizar alias de tabla en una sentencia de unión

# Objetivo

- En la sección anterior se ha tratado la consulta y devolución de los datos de más de una tabla en una base de datos relacional utilizando sintaxis ANSI/ISO SQL: 99.
- En las versiones anteriores de las bases de datos Oracle se necesitaba que las uniones utilizaran la sintaxis de unión propiedad de Oracle y muchas de estas bases de datos anteriores aún se utilizan.
- En esta lección se presenta la sintaxis de unión propiedad de Oracle para uniones igualitarias y el producto cartesiano y sus homólogos ANSI/ISO SQL: 99.

En las versiones anteriores a Oracle 9i Database se necesita el uso de sintaxis de unión propiedad de Oracle.

# Comandos de Unión

- Los dos juegos de comandos o sintaxis que se pueden utilizar para realizar conexiones entre las tablas de una base de datos:
  - Uniones propiedad de Oracle
  - Uniones estándar compatibles con ANSI/ISO SQL: 99



# Comparación de Unión

- Comparación de las Uniones Propiedad de Oracle con Uniones ANSI/ISO SQL: 1999

Unión Propiedad de Oracle	Equivalente de ANSI/ISO SQL: 1999
Producto cartesiano	Unión cruzada
Unión igualitaria	UNIÓN NATURAL Cláusula JOIN USING Cláusula JOIN ON (si se utiliza el operador de igualdad)
Unión no igualitaria	Cláusula ON

# Uniones Propiedad de ORACLE

- Para consultar datos de más de una tabla con la sintaxis propiedad de Oracle, utilice una condición de unión en la cláusula WHERE.
- El formato básico de una sentencia de unión es:

```
SELECT table1.column, table2.column  
FROM table1, table2  
WHERE table1.column1 = table2.column2;
```

# Uniones Propiedad de ORACLE

- Imagine el problema que supondría que dos alumnos de la misma clase tuvieran el mismo apellido.
- Cuando sea necesario hablar con "Jackson", el profesor aclara de qué "Jackson" se trata agregando el apellido antes del nombre.
- Para que sea más fácil leer una sentencia Join y acelerar el acceso a la base de datos, es una buena práctica agregar el nombre de la tabla delante del nombre de la columna.

```
SELECT table1.column, table2.column  
FROM table1, table2  
WHERE table1.column1 = table2.column2;
```



# Uniones Propiedad de ORACLE

- A esto se le denomina "cualificar sus columnas".
- La combinación del nombre de tabla y el nombre de columna ayuda a eliminar nombres ambiguos cuando dos tablas contienen una columna con el mismo nombre de columna.
- Si aparece el mismo nombre de columna en ambas tablas, el nombre de columna debe ir precedido del nombre de la tabla.

# Ejemplo de Sintaxis de Unión

- Para cualificar las columnas, utilice la sintaxis nombretabla.nombrecolumna, como se muestra en el siguiente ejemplo.

```
SELECT table1.column, table2.column  
FROM table1, table2  
WHERE table1.column1 = table2.column2;
```

# UNIÓN IGUALITARIA

- Algunas veces denominada unión "simple" o "interna", una unión igualitaria es una unión de tabla que combina filas con los mismos valores para las columnas especificadas.
- Una unión igualitaria es equivalente a ANSI:
  - NATURAL JOIN
  - JOIN USING
  - JOIN ON (cuando la condición de unión utiliza "=").
- En la siguiente diapositiva se muestran el qué, el dónde y el cómo necesarios para unir las tablas.

# UNIÓN IGUALITARIA

- ¿Qué? La cláusula SELECT especifica los nombres de columna que se van a mostrar.
- ¿Dónde? La cláusula FROM especifica las tablas a las que debe acceder la base de datos, separadas por comas.
- ¿Cómo? La cláusula WHERE especifica cómo se van a unir las tablas.
- Una unión igualitaria utiliza el operador Igual que para especificar la condición de unión.

# UNIÓN IGUALITARIA

```
SELECT employees.last_name, employees.job_id, jobs.job_title
```

¿Qué?

```
FROM employees, jobs
```

¿Dónde?

```
WHERE employees.job_id = jobs.job_id;
```

¿Cómo?

LAST_NAME	JOB_ID	JOB_TITLE
King	AD_PRES	President
Kochhar	AD_VP	Administration Vice President
De Haan	AD_VP	Administration Vice President
Whalen	AD_ASST	Administration Assistant
Higgins	AC_MGR	Accounting Manager
Gietz	AC_ACCOUNT	Public Accountant
Zlotkey	SA_MAN	Sales Manager
Abel	SA_REP	Sales Representative
...	...	...

# UNIÓN IGUALITARIA

Otro ejemplo:

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

LAST_NAME	DEPARTMENT_NAME
Whalen	Administration
Hartstein	Marketing
Fay	Marketing
Mourgos	Shipping
Rajs	Shipping
Davies	Shipping
Matos	Shipping
...	...

Nota: La columna utilizada para unir ambas tablas no es necesario que esté en la lista de columnas SELECT.

# Unión de Producto Cartesiano

- Si en la cláusula WHERE de dos tablas de una consulta de unión no se ha especificado ninguna condición de unión o la condición de unión no es válida, Oracle Server devuelve el producto cartesiano de las dos tablas.
- Esta es una combinación de cada fila de una tabla con cada fila de otra.
- Un producto cartesiano es equivalente a un ANSI CROSS JOIN.
- Para evitar un producto cartesiano, incluya siempre una condición de unión válida en una cláusula WHERE.

# Unión de Producto Cartesiano

- En esta consulta, la condición de unión se ha omitido:

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

LAST_NAME	DEPARTMENT_NAME
Abel	Administration
Davies	Administration
De Haan	Administration
Ernst	Administration
Fay	Administration
Gietz	Administration
Grant	Administration
...	...

160 rows returned in 0.01 seconds



# Restricción de las Filas de una Unión

- Al igual que ocurre con las consultas de una sola tabla, la cláusula WHERE se puede utilizar para restringir las filas tenidas en cuenta en una o más tablas de la unión.
- En la consulta mostrada se utiliza el operador AND para limitar las filas devueltas.

```
SELECT employees.last_name, employees.job_id, jobs.job_title
FROM employees, jobs
WHERE employees.job_id = jobs.job_id
AND employees.department_id = 80;
```

LAST_NAME	JOB_ID	JOB_TITLE
Zlotkey	SA_MAN	Sales Manager
Abel	SA_REP	Sales Representative
Taylor	SA_REP	Sales Representative

# Alias

- Trabajar con nombres de tabla y columna largos puede ser complicado.
- Afortunadamente, hay una forma de acortar la sintaxis utilizando alias.
- Para distinguir las columnas que tienen nombres idénticos, pero que residen en tablas diferentes, utilice alias de tabla.
- Un alias de tabla es similar a un alias de columna; cambia el nombre de un objeto dentro de una sentencia.
- Se crea mediante la introducción del nuevo nombre para la tabla, justo después del nombre de tabla en la cláusula from.

# Alias de Tabla

- Los alias de la tabla se utilizan en la consulta siguiente.

```
SELECT last_name, e.job_id, job_title
FROM employees e, jobs j
WHERE e.job_id = j.job_id
AND department_id = 80;
```

LAST_NAME	JOB_ID	JOB_TITLE
Zlotkey	SA_MAN	Sales Manager
Abel	SA_REP	Sales Representative
Taylor	SA_REP	Sales Representative

- Cuando los nombres de columna no están duplicados en dos tablas, no tiene que agregar el alias o nombre de la tabla al nombre de la columna.

# Alias de Tabla

- Si se utiliza un alias de tabla en la cláusula FROM, el alias de tabla se deberá sustituir por el nombre de tabla mediante la sentencia SELECT.
- Si se utiliza el nombre de una tabla en la cláusula SELECT al que se le haya asignado un alias en la cláusula FROM, se producirá un error.

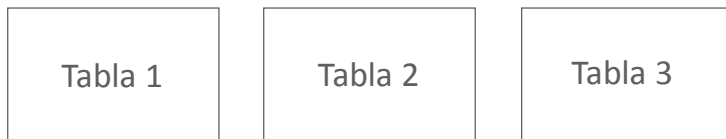
```
SELECT last_name, employees.job_id, job_title
FROM employees e, jobs j
WHERE e.job_id = j.job_id
AND department_id = 80;
```



ORA-00904: "EMPLOYEES"."JOB\_ID": invalid identifier

# Ejemplo de Sintaxis de Unión

- Si deseara unir tres tablas juntas, ¿cuántas uniones necesitaría?
- ¿Cuántos puentes se necesitan para unir tres islas?
- Para unir tres tablas, tendrá que agregar otra condición de unión a la cláusula WHERE utilizando el operador AND



## Ejemplo de Sintaxis de Unión

- Supongamos que necesitamos un informe de nuestros empleados y la ciudad donde está ubicado su departamento.
- Necesitamos unir tres tablas: employees, departments y locations.

```
SELECT last_name, city
FROM employees e, departments d, locations l
WHERE e.department_id = d.department_id
      AND d.location_id = l.location_id;
```

LAST_NAME	CITY
Hartstein	Toronto
Fay	Toronto
Zlotkey	Oxford
Abel	Oxford
...	...

# Terminología

Entre los términos clave utilizados en esta lección se incluyen:

- Alias
- Producto cartesiano
- Unión igualitaria
- Condiciones de unión
- Unión de propiedad

# Resumen

En esta lección, ha aprendido lo siguiente:

- Nombrar las uniones propiedad de Oracle y sus equivalentes en ANSI/ISO SQL: 99
- Crear y ejecutar una sentencia SELECT que da como resultado un producto cartesiano
- Crear y ejecutar sentencias SELECT para acceder a los datos desde más de una tabla utilizando una unión igualitaria
- Crear y ejecutar sentencias SELECT que agregan condiciones de búsqueda usando el operador AND
- Aplicar la regla para utilizar alias de tabla en una sentencia de unión





 **ACADEMY**