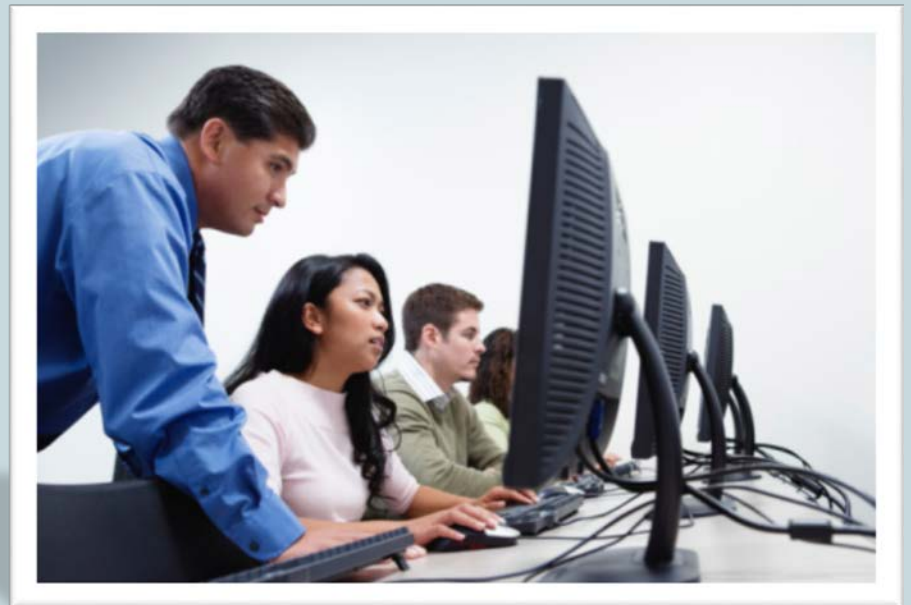




# Programación de Bases de Datos con SQL

## 6-2 Cláusulas Join



# Objetivos

En esta lección se abordan los siguientes objetivos:

- Crear y ejecutar una unión con la cláusula ANSI-99 USING
- Crear y ejecutar una unión con la cláusula ANSI-99 ON
- Crear y ejecutar una consulta ANSI-99 que une tres tablas

# Objetivo

- Al agregar más comandos al vocabulario de la base de datos, estará mejor preparado para diseñar consultas que devuelvan el resultado deseado.
- El objetivo de una unión es enlazar los datos, entre tablas, sin repetir todos los datos en todas las tablas.
- ¿Para qué solicitar más datos de los que realmente necesita?

# Cláusula USING

- En una unión natural, si las tablas tienen columnas con los mismos nombres, pero diferentes tipos de dato, la unión provoca un error.
- Para evitar esta situación, la cláusula de unión se puede modificar con una cláusula USING.
- La cláusula USING especifica las columnas que se deben utilizar para la unión.

# Cláusula USING

- La consulta que se muestra es un ejemplo de la cláusula USING.
- Las columnas a las que se hace referencia en la cláusula USING no deben tener un cualificador (nombre o alias de la tabla) en ninguna ubicación de la sentencia SQL.

```
SELECT first_name, last_name, department_id, department_name  
FROM employees JOIN departments USING (department_id);
```

FIRST_NAME	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Jennifer	Whalen	10	Administration
Michael	Hartstein	20	Marketing
Pat	Fay	20	Marketing
...	...	...	...

# Cláusula USING

- La cláusula USING nos permite utilizar WHERE para limitar las filas de una o de ambas tablas:

```
SELECT first_name, last_name, department_id, department_name  
FROM employees JOIN departments USING (department_id)  
WHERE last_name = 'Higgins';
```

FIRST_NAME	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Shelley	Higgins	110	Accounting

# Cláusula ON

- ¿Qué ocurre si las columnas que se van a unir tienen nombres diferentes, o bien si la unión utiliza operadores de comparación de distinto de, como  $<$ ,  $>$  o BETWEEN?
- No podemos utilizar USING, por lo que en su lugar utilizamos una cláusula ON.
- Esto permite especificar una mayor variedad de condiciones de unión.
- La cláusula ON también nos permite utilizar WHERE para limitar las filas de una o de ambas tablas.



# Ejemplo de Cláusula ON

- En este ejemplo, la cláusula ON se utiliza para unir la tabla employees con la tabla jobs.

```
SELECT last_name, job_title  
FROM employees e JOIN jobs j  
ON (e.job_id = j.job_id);
```

- Se necesita una cláusula ON cuando las columnas comunes tengan nombres diferentes en las dos tablas.

LAST_NAME	JOB_TITLE
King	President
Kochhar	Administration Vice President
De Haan	Administration Vice President
Whalen	Administration Assistant
Higgins	Accounting Manager
Gietz	Public Accountant
Zlotkey	Sales Manager
Abel	Sales Representative
Taylor	Sales Representative
...	

# Ejemplo de Cláusula ON

```
SELECT last_name, job_title
FROM employees e JOIN jobs j
ON (e.job_id = j.job_id);
```

- Al usar una cláusula ON en columnas con el mismo nombre en ambas tablas, debe agregar un calificador (ya sea el alias o el nombre de la tabla), de lo contrario se devolverá un error. En el ejemplo, los alias de la tabla se usan como el calificador e.job\_id = j.job\_id, pero también podrían haberse escrito usando los nombres de la tabla (employees.job\_id = jobs.job\_id).

LAST_NAME	JOB_TITLE
King	President
Kochhar	Administration Vice President
De Haan	Administration Vice President
Whalen	Administration Assistant
Higgins	Accounting Manager
Gietz	Public Accountant
Zlotkey	Sales Manager
Abel	Sales Representative
Taylor	Sales Representative
...	

# Cláusula ON con Cláusula WHERE

- Aquí se muestra la misma consulta con una cláusula WHERE para limitar las filas seleccionadas.

```
SELECT last_name, job_title  
FROM employees e JOIN jobs j  
  ON (e.job_id = j.job_id)  
WHERE last_name LIKE 'H%';
```

LAST_NAME	JOB_TITLE
Higgins	Accounting Manager
Hunold	Programmer
Hartstein	Marketing Manager

# Cláusula ON con Operador Distinto de

- En ocasiones puede que tenga que recuperar los datos de una tabla que no tenga ninguna columna correspondiente en otra tabla.
- Suponga que desea conocer el valor grade\_level para el salario de cada empleado.
- La tabla job\_grades no tiene una columna común con la tabla employees.
- Una cláusula ON nos permite unir las dos tablas

tabla 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

# Cláusula ON con Operador Distinto de

```
SELECT last_name, salary, grade_level, lowest_sal, highest_sal
FROM employees JOIN job_grades
ON(salary BETWEEN lowest_sal AND highest_sal);
```

LAST_NAME	SALARY	GRADE_LEVEL	LOWEST_SAL	HIGHEST_SAL
Vargas	2500	A	1000	2999
Matos	2600	A	1000	2999
Davies	3100	B	3000	5999
Rajs	3500	B	3000	5999
Lorentz	4200	B	3000	5999
Whalen	4400	B	3000	5999
Mourgos	5800	B	3000	5999
Fay	6000	C	6000	9999
...				

# Unión de Tres Tablas

- Tanto USING como ON se pueden utilizar para unir tres o más tablas.
- Supongamos que necesitamos un informe de nuestros empleados, su departamento y la ciudad donde está ubicado el departamento.
- Necesitamos unir tres tablas: employees, departments y locations.



# Ejemplo de la Unión de Tres Tablas

```
SELECT last_name, department_name AS "Department", city
FROM employees JOIN departments USING (department_id)
JOIN locations USING (location_id);
```



LAST_NAME	Department	CITY
Abel	Ventas	Oxford
Davies	Envío	South San Francisco
De Haan	Executive	Seattle
Ernst	IT	Southlake
Fay	Marketing	Toronto
Gietz	Accounting	Seattle
Hartstein	Marketing	Toronto
Higgins	Accounting	Seattle
Hunold	IT	Southlake
...		

# Terminología

Entre los términos clave utilizados en esta lección se incluyen:

- Cláusula ON
- Cláusula USING



# Resumen

En esta lección, debe haber aprendido lo siguiente:

- Crear y ejecutar una unión con la cláusula ANSI-99 USING
- Crear y ejecutar una unión con la cláusula ANSI-99 ON
- Crear y ejecutar una consulta ANSI-99 que une tres tablas

