UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD

Formato para prácticas de laboratorio

CARRERA	PLAN DE ESTUDIO	CLAVE ASIGNATURA	NOMBRE DE LA ASIGNATURA
IC	2003-1	5046	Bases de Datos

PRÁCTICA No.	LABORATORIO DE	Bases de Datos	DURACIÓN (HORA)
8	NOMBRE DE LA PRÁCTICA	MySQL Uniones	2

1. INTRODUCCIÓN

Los sistemas de software organizan sus datos en bases de datos. Éstas a su vez están conformadas por múltiples tablas. En ocasiones, recuperar la información deseada requiere buscar en múltiples tablas simultáneamente. Para facilitar la selección de información en dos tablas, SQL provee una función conocida como join o en español unión. El uso de la instrucción join reduce el número de querys de SQL que se tienen que escribir para acceder la información deseada.

2. OBJETIVO (COMPETENCIA)

Recuperar información de múltiples tablas empleando la instrucción join de una manera creativa y lógica.

Formuló	Revisó	Aprobó	Autorizó
Cecilia M. Curlango Rosas	Gloria Etelbina Chávez Valenzuela		Miguel Ángel Martínez Romero
Maestro	Coordinador de la Carrera	Gestión de la Calidad	Director de la Facultad

3. FUNDAMENTO

Las tablas de una base de datos se pueden relacionar entre si por medio de campos llave. Una llave primaria es una columna que contiene un valor único que no se repite en ninguno de los

renglones. El propósito es unir datos entre tablas de tal manera que no se tengan que estar repitiendo los mismos datos en todas las tablas.

La Tabla 1 Empleados tiene como llave primaria el campo `empleadoID`, esto significa que no cada clave corresponde a un solo empleado. Este identificador es independiente del nombre de los empleados. La Tabla 2 Ordenes tiene como llave primaria a la columna `productoID`. En esta misma tabla, el campo `empleadoID` es una referencia a la tabla Empleados.

empleadoID	***Nombre***
101	Carmona, Juan
102	Alvarez, Daniela
103	Sanchez, Julieta
456	Torres, Esperanza

Tabla 1Empleados

productoID	***Producto***	***empleadoID***
340	copiadora	102
854	sumadora	103
971	lampara	103

Tabla 2Ordenes

Para seleccionar datos de ambas tablas se puede usar el siguiente query (quien ordenó un producto y qué fue ese producto) el cual nos arroja como resultado los registros mostrados en la Tabla 3.

```
SELECT Empleados.Nombre, Ordenes.Producto
FROM Empleados, Ordenes
WHERE
Empleados.empleadoID=Ordenes.empleadoID
```

Query 1

Nombre	Producto
Alvarez,Daniela	copiadora
Sanchez,Julieta	sumadora
Sanchez,Julieta	lampara

Tabla 3Resultado de query 1.

Si quisiéramos saber quien ordenó una lampara, el query sería:

```
SELECT Empleados.Nombre
FROM Empleados, Ordenes
WHERE
Empleados.empleadoID=Ordenes.empleadoID
AND Ordenes.Producto='lampara'
```

Query 2

Nombre
Sanchez,Julieta

Tabla 4Resultado de query 2.

Otra forma de hacer el Query 1 es utilizando un join específicamente un Inner Join. La sintaxis de un inner join genérico sería:

```
SELECT campo1, campo2, campo3
FROM primera_tabla
INNER JOIN segunda_tabla
ON primera_tabla.campo_llave = segunda_tabla.llave_foranea
```

Específicamente para obtener la misma información que en el Query 1 el nuevo query sería como el que se muestra en el Query 3.

```
SELECT Empleados.nombre,
Ordenes.Producto
FROM Empleados
INNER JOIN Ordenes
ON
Empleados.empleadoID=Ordenes.EmpleadoID
```

Query 3

Un inner join regresa todos los renglones en los cuales hay concordancia. Si hay renglones en Empleados que no concuerdan con Ordenes, entonces estos renglones no se mostrarán.

Otro tipo de join que existe es el Left Join. Este join regresa todos los renglones de la primer tabla aun cuando no concuerde con la segunda tabla. En este caso en el que los reglones no concuerden, tambien se muestran. La sintaxis de este join es:

```
SELECT campo1, campo2, campo3
FROM primera_tabla
LEFT JOIN segunda_tabla
ON primera_tabla.campo_llave = segunda_tabla.llave_foranea
```

El Query 4 se utiliza para obtener un listado de todos los empleados y sus todas las ordenes que puedan tener. El resultado de este query puede verse en Tabla 5.

```
SELECT Empleados.Nombre, Ordenes.Producto
FROM Empleados
LEFT JOIN Ordenes
ON Empleados.empleadoID=Ordenes.empleadoID
```

Query 4

Nombre	Producto
Carmona,Juan	NULL
Alvarez,Daniela	copiadora
Sanchez,Julieta	sumadora
Sanchez,Julieta	lampara
Torres,Esperanza	NULL

Tabla 5Resultado del query 4

Otro tipo de join es el Right Join. Un Right Join regresa todos los renglones de la segunda tabla, aun cuando no exista concordancia con la primer tabla. Si existen renglones en la segunda tabla que no concuerdan con algo de la primer tabla, entonces estos renglones también se muestra. A continuación se muestra la sintaxis de este join:

```
SELECT campo1, campo2, campo3
FROM primera_tabla
RIGHT JOIN segunda_tabla
ON primera_tabla.campo_llave = segunda_tabla.llave_foranea
```

El siguiente query lista todas las ordenes y los nombres que los que hicieron esas ordenes:

```
SELECT Empleados.Nombre, Ordenes.Producto
FROM Empleados
RIGHT JOIN Ordenes
ON Empleados.empleadoID=Ordenes.EmpleadoID
```

Query 5

Nombre	***Producto***
Alvarez,Daniela	copiadora
Sanchez,Julieta	sumadora
Sanchez,Julieta	lampara

Tabla 6Resultado del Query 5

Para saber quien ordenó una lampara, el query se escribiría como sigue:

```
SELECT Empleados.Nombre
FROM Empleados
INNER JOIN Ordenes
ON Empleados.empleadoID=Ordenes.empleadoID
WHERE Ordenes.Producto='lampara'
```

Query 6

4. PROCEDIMIENTO (DESCRIPCIÓN)

A) EQUIPO NECESARIO	MATERIAL DE APOYO
Computadoras con el sistema operativo Linux Acceso y permisos para trabajar con el manejador de bases de datos MySQL en un servidor	Manual de MySQL

B) DESARROLLO DE LA PRÁCTICA

Utilice las tablas de la base de datos descritas en la practica 6 para realizar y probar los siguiente queries empleando joins.

1. Muestre los nombres de todos los jefes de departamento.
2. Muestre los nombres de los participantes en proyectos.
3. Muestre los nombres de las personas que trabajan en el departamento de investigación.
4. Muestre el nombre de cada persona que ha participado en proyectos y el numero total de horas que ha dedicado a proyectos.
5. Muestre la ubicación de los nombres de cada departamento y las cuidades en las que se encuentran.
6. Muestre los nombres de los dependientes de aquellos empleados cuyo ingreso es mayor a 300.
7. Ejercicios adicionales planteados por el maestro.

C) CÁLCULOS Y REPORTE

5. RESULTADOS Y CONCLUSIONES

6. ANEXOS

7. REFERENCIAS

Manual de Referencia MySQL Sintaxis `join`: http://dev.mysql.com/doc/refman/5.0/en/join.html