UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)

Formato para prácticas de laboratorio

CARRERA	PLAN DE ESTUDIO	CLAVE DE UNIDAD DE APRENDIZAJE	NOMBRE DE LA UNIDAD DE APRENDIZAJE
Ingeniero en Computación	2009-2	12111	Base de Datos

PRÁCTICA No.	LABORATORIO DE	Base de Datos	DURACIÓN (HORAS)
8	**NOMBRE DE LA PRÁCTICA**	Modificación y Eliminación de datos UPDATE Y DELETE	2

1. INTRODUCCIÓN

Al inicio o arranque de un sistema de información construido alrededor de una base de datos, es muy común que la primera vez que se almacenan datos contenga mucha información errónea y por lo tanto ocupe de mecanismos o instrucciones que permitan corregir, modificar, editar o eliminar esta información.

La modificación y eliminación son dos de los procesos más comunes que se realizan con tablas en bases de datos, a estos procesos, se les conoce como actualización, edición o modificación de los datos, registros, tuplas o renglones contenidos en una tabla.

Las instrucciones **UPDATE** y **DELETE** son las instrucciones en SQL especializadas en esta área de procesos comunes con tablas, y son estas instrucciones las que se desarrollarán en esta práctica.

2. OBJETIVO (COMPETENCIA)

Realizar la actualización y eliminación de datos sobre las tablas de una base de datos, utilizando las instrucciones del DBMS de MySQL para la correcta manipulación de la información, con actitud positiva, propositiva y analítica.

3. FUNDAMENTO

Modificación de Información dentro de una tabla (UPDATE)

Hasta ahora ya conocemos la forma de introducir datos en las tablas que componen nuestra base de datos, pero que pasa cuando echamos un vistazo, y nos damos cuenta de que hemos cometido algún error en alguno de los datos, ¿qué hacemos para arreglar este tipo de problemas?.

La instrucción **UPDATE**, nos permite realizar cambios en los datos que ya tenemos dentro de una tabla de nuestra base de datos. La sintaxis de esta orden es:

```
UPDATE nom_Tabla SET atributo1=expr1, atributo2=expr2, WHERE condicion ;
```

La instrucción **UPDATE** actualiza o modifica los renglones de una tabla, **SET** le indica a MYSQL cuales son las columnas a modificar y **WHERE** se usa para seleccionar el renglón o renglones que serán modificados. Si la clausula **WHERE** se omite, todos los renglones serán modificados. A continuación se muestras las formas más comunes en que la instrucción **UPDATE** es utilizada.

1.- Actualizar una columna o varias columnas a todos los renglones de una tabla.

Ejemplo 1: Poner el valor de “Domicilio Conocido” en la columna Direccion de la tabla Persona.

```
UPDATE Persona SET Direccion = “Domicilio Conocido”;
```

A continuación se muestra la pantalla que aparecerá al ejecutarse la instrucción SQL anterior:

P_id	Apellido	Nombre	Dirección	Ciudad
2	Chávez	Gloria	Domicilio Conocido	Mexicali
10	Curlango	Cecilia	Domicilio Conocido	Mexicali
11	Ibarra	Jorge	Domicilio Conocido	Mexicali
3	López	Alicia	Domicilio Conocido	Tijuana
5	Martínez	Laura	Domicilio Conocido	NULL
1	Morgan	Kenneth	Domicilio Conocido	Caléxico
4	Navarro	Pablo	Domicilio Conocido	Mexicali
14	Rodríguez	Marcela	Domicilio Conocido	Mexicali

También se pueden utilizar expresiones algebraicas.

Ejemplo 2: En la siguiente instrucción se incrementara en 10% el valor actual de la columna sueldos de todas las filas de la tabla Empleado.

```
UPDATE Empleado SET Sueldo = Sueldo * 1.10 ;
```

Se afecta a la tabla Empleado quedando como se muestra a continuación:

E_id	Apellido	Nombre	Dirección	Ciudad	Sueldo
10	Curlango	Cecilia	Independencia S/N	Mexicali	1650.00
11	Ibarra	Jorge	Colon S/N	Mexicali	2970.00
12	Flores	Brenda	Revolución S/N	Tijuana	550.00
13	Herrera	Eva	Windows S/N	Caléxico	2081.20
14	Rodríguez	Marcela	Arista S/N	Mexicali	2860.00
20	NULL	NULL	YA EXISTE	NULL	0.00

2.- El caso más común es solo actualizar un renglón o registro, para esta situación solo usar la claúsula **WHERE**.

Ejemplo 3: En la siguiente instrucción, se pone el valor de 500 a la columna Sueldo de la tabla Empleado pero solo donde el valor de la columna Dirección sea igual a “YA EXISTE”.

```
UPDATE Empleado set Sueldo = 500 where Direccion = “YA EXISTE”;
```

A continuación se muestra la pantalla que aparecerá al ejecutarse la instrucción SQL anterior:

E_id	Apellido	Nombre	Dirección	Ciudad	Sueldo
10	Curlango	Cecilia	Independencia S/N	Mexicali	1650.00
11	Ibarra	Jorge	Colon S/N	Mexicali	2970.00
12	Flores	Brenda	Revolución S/N	Tijuana	550.00
13	Herrera	Eva	Windows S/N	Caléxico	2081.20
14	Rodríguez	Marcela	Arista S/N	Mexicali	2860.00
20	NULL	NULL	YA EXISTE	NULL	500.00

Ejemplo 4: Con la siguiente instrucción, actualizaremos la información de Jorge Ibarra poniendo el valor de “López Mateos” en Dirección y “Ensenada” en Ciudad.

```
UPDATE Empleado SET Direccion=”Lopez Mateos”, City=”Ensenada”
WHERE Apellido=”Ibarra” AND Nombre=”Jorge”;
```

A continuación se muestra la pantalla que aparecerá al ejecutarse la instrucción SQL anterior:

E_id	Apellido	Nombre	Dirección	Ciudad	Sueldo
10	Curlango	Cecilia	Independencia S/N	Mexicali	1650.00
11	Ibarra	Jorge	López Mateos	Ensenada	2970.00
12	Flores	Brenda	Revolución S/N	Tijuana	550.00
13	Herrera	Eva	Windows S/N	Caléxico	2081.20
14	Rodríguez	Marcela	Arista S/N	Mexicali	2860.00
20	NULL	NULL	YA EXISTE	NULL	500.00

3.- **UPDATE** también puede actualizar un subconjunto de renglones de una tabla.

Ejemplo 5: Con la siguiente instrucción, se asigna el valor de 3500.00 a la columna Sueldo de la tabla Empleado, pero solo en aquel o aquellos renglones en los que el valor de la columna Ciudad sea igual a “Mexicali”.

```
UPDATE Empleado set Sueldo = 3500.00 WHERE Ciudad = “Mexicali”;
```

A continuación se muestra como quedara la tabla Empleado después de ejecutarse la instrucción SQL anterior:

E_id	Apellido	Nombre	Dirección	Ciudad	Sueldo
10	Curlango	Cecilia	Independencia S/N	Mexicali	3500.00
11	Ibarra	Jorge	Colon S/N	Mexicali	2970.00
12	Flores	Brenda	Revolución S/N	Tijuana	550.00
13	Herrera	Eva	Windows S/N	Caléxico	2081.20
14	Rodríguez	Marcela	Arista S/N	Mexicali	3500.00
20	NULL	NULL	YA EXISTE	NULL	500.00

El operador **LIKE** es usado en forma conjunta con la clausula **WHERE** para realizar la búsqueda de un patrón específico en una columna en particular y actualizar los registros que coincidan con ese patrón.

Ejemplo 6: Supongamos que queremos bajarle 15% el sueldo a los registros de la tabla Empleado que vivan en una ciudad que en su nombre contenga “xi”, hacemos lo siguiente:

```
UPDATE Empleado SET Sueldo=Sueldo*.85 WHERE Ciudad LIKE "%xi%";
```

Al ejecutarse la instrucción anterior sobre la tabla Empleado, obtenemos lo siguiente como resultado.

E_id	Apellido	Nombre	Dirección	Ciudad	Sueldo
10	Curlango	Cecilia	Independencia S/N	Mexicali	2975.00
11	Ibarra	Jorge	Colon S/N	Mexicali	2524.50
12	Flores	Brenda	Revolución S/N	Tijuana	550.00
13	Herrera	Eva	Windows S/N	Caléxico	1769.02
14	Rodríguez	Marcela	Arista S/N	Mexicali	2975.00
20	NULL	NULL	YA EXISTE	NULL	500.00

UPDATE Y SELECT en el mismo query.

Se pueden realizar subconsultas de manera que el resultado que se genere de la subconsulta sirva para realizar la modificación sobre una tabla distinta a la que se hizo la subconsulta o sobre la misma tabla.

Para el primer caso, si observas la tabla Persona todas las filas tienen en la campo Dirección el valor de **“Domicilio Conocido”** sin embargo, en la tabla Empleado cada fila tienen la dirección correcta para cada empleado. En este caso vamos a modificar una tabla distinta a la tabla donde se hará la subconsulta. Para este tipo de actualización no existe ningún problema, ya que la consulta y la actualización se hacen sobre tablas diferentes.

Ejemplo 7: Seleccionar de la tabla Empleado la dirección del empleado 10 y actualizar la dirección del mismo empleado sobre la tabla Persona, para hacer la actualización hacemos lo siguiente:

```
UPDATE Persona SET Direccion=(SELECT Direccion FROM Empleado
WHERE E_id=10) WHERE P_id=10;
```

Para el segundo caso, en el que se necesita hacer una modificación sobre la misma tabla que se hace la subconsulta, **“no existe solución”**, es un problema de lógica, ya que no puedes consultar y modificar al mismo tiempo una tabla porque puede darse que la modificación del registro que estás haciendo afecte el resultado de la consulta que estás haciendo para la modificación. Si este fuera el caso, entonces tendríamos las siguientes preguntas:

- ¿En qué estado está el registro: modificado o no?
- ¿Debe la consulta incluirlo, o no?
- Si lo estás modificando, entonces no cumple con el predicado por lo tanto no lo debe mostrar, pero si no lo muestra, ¿cómo lo modificas?

Como SQL utiliza algoritmos que son usados de forma general para TODO TIPO de consultas, y no sólo para esta consulta especifica que se desea, entonces, SQL simplemente NO TE DEJARÁ HACERLO JAMÁS.

¿Cómo se hace entonces?, Bueno, para esto se deben construir tablas transaccionales o temporales, donde se almacenen las modificaciones, y una vez confirmadas, se utilicen para actualizar la tabla original.

Ejemplo 8: Queremos incrementar en 1000 pesos, el sueldo del empleado de la tabla Empleado que tiene el salario mayor, para hacer la actualización hacemos lo siguiente:

```
UPDATE Empleado SET Sueldo=Sueldo+1000 WHERE Sueldo =
(SELECT maximo FROM
(SELECT  max(Sueldo) as maximo from Empleado)
as sueldo_max);
```

Eliminación de Información dentro de una tabla (DELETE)

Una de las operaciones de proceso o manipulación de tablas en MySQL es **DELETE** que permite borrar o eliminar algún registro o renglón de la tabla, un subconjunto de renglones de la tabla o si es necesario eliminar todos los renglones de la tabla. La sintaxis de esta orden es:

```
DELETE FROM nombre_Tabla WHERE condicion;
```

La instrucción DELETE borra o elimina los renglones de una tabla, FROM le indica a MYSQL la tabla sobre la cual se hará la eliminación, y WHERE cuál o cuáles son los renglones que se desean eliminar. Si la clausula WHERE se omite, todos los renglones serán eliminados. Los casos más comunes son:

1.- EL caso más simple es eliminar un renglón cualquiera.

Ejemplo 9: Con la siguiente instrucción, se borra el registro cuya columna Direccion sea igual a “YA EXISTE” de la tabla Empleado.

```
DELETE from Empleado where Direccion = “YA EXISTE”;
```

Al ejecutarse la instrucción anterior sobre la tabla Empleado, obtenemos lo siguiente como resultado.

E_id	Apellido	Nombre	Dirección	Ciudad	Sueldo
10	Curlango	Cecilia	Independencia S/N	Mexicali	2975.00
11	Ibarra	Jorge	Colon S/N	Mexicali	2524.50
12	Flores	Brenda	Revolución S/N	Tijuana	550.00
13	Herrera	Eva	Windows S/N	Caléxico	1769.02
14	Rodríguez	Marcela	Arista S/N	Mexicali	2975.00

2.- También pueden eliminarse un subconjunto de renglones usando la clausula WHERE con un filtro o condición.

Ejemplo 10: Con la siguiente instrucción se borra el registro o los registros de la tabla Empleado en los cuales su columna Sueldo sea menor o igual a 950.

```
DELETE from Empleado where Sueldo <= 950.00”;
```

Al ejecutarse la instrucción anterior sobre la tabla Empleado, obtenemos lo siguiente como resultado.

E_id	Apellido	Nombre	Dirección	Ciudad	Sueldo
10	Curlango	Cecilia	Independencia S/N	Mexicali	2975.00
11	Ibarra	Jorge	Colon S/N	Mexicali	2524.50
13	Herrera	Eva	Windows S/N	Caléxico	1769.02
14	Rodríguez	Marcela	Arista S/N	Mexicali	2975.00

El operador **LIKE** es usado en forma conjunta con la clausula **WHERE** para realizar la búsqueda de un patrón específico en una columna en particular y eliminar los registros que coincidan con ese patrón.

Ejemplo 11: Supongamos que queremos eliminar los registros de la tabla Empleado que en su apellido contengan “rr”, hacemos lo siguiente:

```
DELETE FROM Empleado WHERE Apellido LIKE "%rr%";
```

Al ejecutarse la instrucción anterior sobre la tabla Empleado, obtenemos lo siguiente como resultado.

E_id	Apellido	Nombre	Dirección	Ciudad	Sueldo
10	Curlango	Cecilia	Independencia S/N	Mexicali	2975.00
14	Rodríguez	Marcela	Arista S/N	Mexicali	2975.00

3.- Se puede eliminar todos los registros o renglones de una tabla, como se muestra en el siguiente ejemplo.

```
DELETE FROM Empleado;
```

Ejemplo 12: Supongamos que queremos eliminar todos los registros de la tabla Empleado, hacemos lo siguiente:

```
DELETE FROM Empleado;
o
DELETE * FROM Empleado;
```

Al ejecutarse la instrucción anterior sobre la tabla Empleado eliminaremos todos los registros de la tabla.

No es recomendable utilizar la instrucción anterior porque se eliminan todos los registros de la tabla Empleado. Si lo hacen y necesitan los datos de la tabla, pueden volver a insertarlos como lo vieron en las prácticas anteriores.

Un consejo, cuando se quiera borrar registros de una tabla, se puede probar primero con un SELECT, si el SELECT funciona correctamente y te selecciona exactamente los registros que quieras borrar; entonces cambias SELECT por DELETE.

DELETE Y SELECT en el mismo query.

Se pueden realizar subconsultas de manera que el resultado que se genere de la subconsulta sirva para realizar la eliminación sobre una tabla distinta a la que se hizo la subconsulta o sobre la misma tabla.

La sintaxis que se utiliza para lo anterior, es la misma que utilizamos en UPDATE Y SELECT, obviamente cambiando la instrucción UPDATE por la instrucción DELETE.

4. PROCEDIMIENTO (DESCRIPCIÓN)

A) EQUIPO NECESARIO	MATERIAL

Computadora con MySQL instalado.

B) DESARROLLO DE LA PRÁCTICA

1. Utiliza la base de datos creada en la practica 4 llamada **practica4_XXXXX**, donde XXXXX representan el numero de tu matricula. En el caso de haber borrado la base de datos o modificado la información o alterado la estructura, hacerla de nuevo como marca la practica 4.

2. Realiza el **Ejemplo 1** de la fundamentación y comprueba su funcionalidad. Posteriormente, consulta las tablas afectadas para que compares los resultados que obtienes con los que se muestran en los ejemplos de esta práctica

3. Altera la estructura de la tabla Empleado y agrega una columna con el nombre de Sueldo, utilizando la característica que se describe a continuación. Después de hacerlo, verifica los valores de la columna Sueldo de la tabla Empleado.

Atributo	Tipo	Observación
Sueldo	Flotante	

4. Utilizando los ejemplos de la fundamentación, encuentra la forma de modificar la columna Sueldo de la tabla Empleado que agregaste en el paso 3, de manera que se vea la columna de la siguiente forma para cada uno de los empleados de la tabla:

Apellido	Nombre	Sueldo
Curlango	Cecilia	1500.00
Ibarra	Jorge	2700.00
Flores	Brenda	500.00
Herrera	Eva	1892.00
Rodríguez	Marcela	2600.00

5. Realiza del **Ejemplos 2** hasta el **Ejemplo 6** de la fundamentación y comprueba su funcionalidad. En cada ejemplo que vayas haciendo, consulta las tablas afectadas para que compares los resultados que obtienes con los que se muestran en los ejemplos de esta práctica.

6. Realiza el **Ejemplos 7** de la fundamentación, comprueba su funcionalidad. ¿Cómo queda la tabla afectada después de ejecutar esta actualización?

7. Actualiza cada una de las direcciones de la tabla Persona con la información de la tabla Empleado como se hizo en el **Ejemplo 7**, esto es, utilizando UPDATE Y SELECT simultáneamente.

8. Realiza el **Ejemplos 8** de la fundamentación, comprueba su funcionalidad. ¿Cómo queda la tabla afectada después de ejecutar esta actualización?

9. Realiza del **Ejemplos 9** hasta el **Ejemplo 12** de la fundamentación y comprueba su funcionalidad. En cada ejemplo que vayas haciendo, consulta las tablas afectadas para que compares los resultados que obtienes con los que se muestran en los ejemplos de esta práctica.

10. Eliminar de la tabla Persona toda la información del empleado que tenga el menor sueldo de la tabla Empleado, utiliza para llevar a cabo la eliminación las instrucciones **DELETE** Y **SELECT** simultáneamente. ¿Cómo queda la tabla afectada después de ejecutar esta eliminación?

11. Eliminar de la tabla Empleado toda la información del empleado que tenga el mayor sueldo, utiliza para llevar a cabo la eliminación las instrucciones **DELETE** Y **SELECT** simultáneamente. ¿Cómo queda la tabla afectada después de ejecutar esta eliminación?

C) CÁLCULOS Y REPORTE

El alumno entregará al maestro los resultados obtenidos, de forma que, el maestro pueda verificar y validar las soluciones dadas por el alumno a cada uno de los puntos descritos en la práctica.

5. RESULTADOS Y CONCLUSIONES

Al finalizar la práctica, el alumno será capaz de realizar modificaciones de información sobre los registros de una tabla en una base de datos, así como la eliminación de registros, a través de instrucciones SQL como UPDATE y DELETE utilizando el DBMS de MySQL

6. ANEXOS

7. REFERENCIAS

[1] Manual de Referencia MySQL: http://dev.mysql.com/doc/refman/5.5/en/

[2] Sintaxis de Instrucciones SQL: http://dev.mysql.com/doc/refman/5.5/en/sql-syntax.html

[3] Tutorial de MySQL: http://dev.mysql.com/doc/refman/5.5/en/tutorial.html

M.C. Pablo M. Navarro			
Nombre y Firma del Maestro	Nombre y Firma del Responsable de Programa Educativo	Nombre y Firma del Responsable de Gestión de Calidad	Nombre y Firma del Director de la Facultad