**MANIPULACIÓN DE REGISTROS: INSERT, DELETE Y UPDATE**

**I.- INSERT**

**EJERCICIO 1**: Crear una tabla AMIGOS dentro de la BD Empleados2 (Cargar el fichero Empleados2.sql) que posea un ID que sea entero not null y los campos para almacenar el nombre, el teléfono y el trabajo que serán cadenas de longitud 30, 10 Y 15 respectivamente:

Las consultas de acción son aquellas que no devuelven ningún registro, son las encargadas de acciones como añadir, borrar y modificar registros.

**INSERTAR REGISTROS: INSERT INTO**

Agrega un registro en una tabla. Esta consulta puede ser de dos tipos:

* Insertar un único registro
* Insertar en una tabla los registros contenidos en otra tabla.

Debe especificarse cada uno de los campos del registro al que se le va a asignar un valor así como el valor para dicho campo. Cuando no se especifica dicho campo, se inserta el valor predeterminado o Null. Los registros se agregan al final de la tabla.

* ***Para insertar un único Registro:***

En este caso la sintaxis es la siguiente:

**INSERT INTO Tabla (campo1, campo2, ..., campoN) VALUES (valor1, valor2, ..., valorN)**Esta consulta graba en el campo1 el valor1, en el campo2 y valor2 y así sucesivamente.

**EJERCICIO 2**: Realice esta inserción: 1,'GOMEZ, JUAN','JUEZ'

Al hacer un SELECT \* veremos que el resto de campos estas vacío, es decir a null.

* ***Si se van a insertar todos los campos el formato es:***

**INSERT INTO Tabla VALUES (valor1, valor2, ..., valorN)**

**EJERCICIO 3**: Realice esta inserción: 2, 'SANCHEZ, LUIS','607564321','ESTUDIANTE'

**EJERCICIO 4**: Pruebe a insertar un registro sin incluir el ID ¿Qué ocurre?

**EJERCICIO 5**: Inserte 4 registros más CON EL ID CORRELATIVO (3,4,5,6...10) , en los que algunos tengan el teléfono a null. Si no lo ha hecho modifique el campo IdAmigo para que sea auto\_increment

Si ya hay registros introducidos y estos tienen un idAmigo repetido será necesario borrarlos para poder crear la clave primaria.

Haga un SELECT \* de los amigos cuyo teléfono no es null.

Modifica la tabla amigos, añadiendo la restricción que el nombre sea único y obligatorio.

Elimina la restricción que acabas de introducir:

***Para insertar Registros de otra Tabla:***

Para insertar todos los campos:

**INSERT INTO TablaDestino SELECT \* FROM TablaOrigen [WHERE...]**

De esta forma los campos de Tabla Origen se grabarán en Tabla, para realizar esta operación es necesario que todos los campos de Tabla Origen estén contenidos en TablaDestino, es decir, que TablaDestino posea todos los campos de TablaOrigen (coincidan en orden y tipo, no es necesario que el nombre de los campos sea igual).

En este tipo de inserción de datos hay que tener especial cuidado con los campos contadores o auto\_incrementados puesto que al insertar un valor en un campo de este tipo se escribe el valor que contenga su campo homólogo en la tabla origen, no incrementándose como le corresponde.

**EJERCICIO 6a**: Cree la tabla amigos2 que sea una copia de amigos, pero sin registros:

**EJERCICIO 6b**: Inserte todos los registros de amigos a amigos2:

Para insertar sólo algunos campos

Se puede también especificar los campos que deseo copiar, y de esta forma que tabla destino sólo albergue los campos que me interesan.

En este caso la sintaxis es:

**INSERT INTO TablaDestino [(campo1, campo2, , campoN) ]   SELECT [campo1, campo2, , campoN] FROM TablaOrigen [WHERE...]**

**EJERCICIO 6c**: Inserte todos los registros de amigos a amigos2, pero sólo los campos nombre, teléfono y trabajo:

En este caso se seleccionarán los campos 1,2,..., n de la tabla destino y se grabarán en los campos 1,2,.., n de la Tabla origen. La condición SELECT puede incluir la cláusula WHERE para filtrar los registros a copiar. Si Tabla destino y Tabla Origen poseen la misma estructura podemos simplificar la sintaxis.

**EJERCICIO 7**: Fijándonos en la descripción de AMIGOS insertemos desde la tabla EMPLE solo los campos nombre (nombre y apellido de emple), trabajo (oficio de emple) en amigos2.

Compruebe que se ha producido la copia.

También podemos insertar registros nuevos en una tabla especificando alguno de los campos con un select y otros como valores fijos.

**EJERCICIO 8:** Hagamos la misma inserción anterior, pero en teléfono insertemos el valor ‘9111111’ en todos los registros.

**EJERCICIO 9**: Fijándonos en la descripción de AMIGOS insertemos desde la tabla AGENDA y EMPLE. Los campos nombre (nombre y apellido de emple), trabajo (oficio de emple) y teléfono (de agenda), pero sólo de los que pertenecen al departamento 20.

Compruebe que se ha producido la copia

También podemos insertar registros nuevos en una tabla especificando alguno de los campos con un select y otros como valores fijos.

**INSERT INTO TablaDestino SELECT DISTINCT (VALOR FIJO1, CAMPO1,....) FROM TablaOrigen [WHERE]**

**EJERCICIO 10:** Insertemos al Presidente de la empresa en nuestra agenda con el select anterior, pero en oficio pongamos el valor fijo “JEFAZO”

**EJERCICIO 11:** ¿QUÉ HACE ESTA SENTENCIA?

INSERT INTO AMIGO2

SELECT null, ‘PEREZ, RAQUEL’, TELEFONO, TRABAJO

FROM AMIGOS WHERE TELEFONO=(SELECT MAX(TELEFONO) FROM AGENDA);

Modificar las tablas anteriores, de forma que en la agenda sólo tendremos registros de empleados y en amigos también. Establecer las claves primarias y ajenas para que si se elimina un registro de empleado se elimine en cascada todos los registros correspondientes en las tablas amigos y agenda.

**EJERCICIO 12:** Vuelve a realizar la inserción del ejercicio anterior, pero teniendo en cuenta que el idAmigo ya no es un campo auto\_incrementado y las restricciones que se acaban de añadir.

**EJERCICIO 13**: Inserta en EMPLE al empleado 1111, ‘GARCIA’ de oficio ANALISTA, con fecha de alta la actual (SYSDATE), en el departamento 30, con salario y comisión iguales al que tiene el sueldo mínimo del departamento 30. El resto de valores a nulo.

**EJERCICIO 14**: Inserta en EMPLE al empleado 1221, ‘PARADA’ de DIRECTOR, del departamento 40, con fecha de alta la actual(SYSDATE), con salario igual al máximo sueldo de la empresa que no sea el del presidente. El resto de valores a nulo.

**EJERCICIO 15**: Inserta en EMPLE al empleado 9999, ‘CLEMENTE’ de ‘VENDEDOR’, en el departamento con menos empleados, con fecha de alta la actual(SYSDATE), y salario y comisión a 0. El resto de valores a nulo.

* **CREAR UNA TABLA NUEVA (Sólo la estructura)**

SINTAXIS 1

**CREATE TABLE nueva\_tabla LIKE TablaOrigen;**

Crea la tabla con la misma estructura, sin registros, pero con las restricciones.

* **CREAR UNA TABLA NUEVA INSERTANDO UNA SELECCIÓN DE REGISTROS**

SINTAXIS 2

**CREATE TABLE nueva\_tabla AS SELECT campos FROM tabla\_origen [WHERE];**

Crea la tabla con la misma estructura y copia todos los registros pero sin las restricciones. Habría que añadirle las restricciones posteriormente.

**EJERCICIO 16**: Vamos a crear una tabla TRABAJADORES sólo con los amigos cuya profesión no es nula. Añadir el código y comprobar el resultado.

**EJERCICIO 17**: Vamos a crear una tabla MOVILES sólo con los campos nombre y Teléfono de los amigos cuyo teléfono comience por 6

**EJERCICIO 18**: Crear una tabla DIRECTORES sólo con los campos apellido y departamento de los directores de emple. Añadir el código y comprobar el resultado.

**II.- DELETE**

Una consulta de eliminación suprime los registros de una o más de las tablas listadas en la cláusula FROM que satisfagan la cláusula WHERE.

Esta consulta elimina los registros completos, no es posible eliminar el contenido de algún campo en concreto. Su sintaxis es:

**DELETE FROM Tabla WHERE condición**

Una vez que se han eliminado los registros utilizando una consulta de borrado, no puede deshacer la operación, a no ser que use rollback. Si desea saber qué registros se eliminarán, primero examine los resultados de una consulta de selección que utilice el mismo criterio y después ejecute la consulta de borrado. Mantenga copias de seguridad de sus datos en todo momento.

Si elimina los registros equivocados podrá recuperarlos desde las copias de seguridad.   
  
**EJERCICIO 1:**

Cree una tabla EMPLE2 que sea una copia de EMPLE con los comandos aprendidos anteriormente

Cree una tabla DEPART2 que sea una copia de DEPART con los comandos aprendidos anteriormente

**EJERCICIO 2:**

Antes de ejecutar, compruebe los registros a borrar con una sentencia SELECT \* de la condición dada ¿QUÉ REGISTO BORRA ESTA SENTENCIA?.

DELETE FROM  EMPLE2 WHERE OFICIO=’DIRECTOR’ AND DEPT\_NO=40;

**EJERCICIO 3:**

Borre de EMPLE2 los empleados que sean ANALISTAS y no del departamento 20

**EJERCICIO 4:**

Borre de EMPLE2 los empleados que sean VENDEDORES y del departamento 30

**~~EJERCICIO 5~~(\*NOTA):**

Borre de EMPLE2 los empleados cuyo salario sea igual al sueldo menor de su departamento

1/ Visualizar el sueldo menor de cada departamento

select dept\_no,salario from emple2 e where salario = (select min(salario) from EMPLE2 where e.dept\_no=dept\_no group by dept\_no);

mysql> delete from emple2 where salario = (select min(salario) from EMPLE2 e where e.dept\_no=dept\_no group by dept\_no);

ERROR 1093 (HY000): You can't specify target table 'emple2' for update in FROM clause

**EJERCICIO 5(\*):**

Borre de EMPLE2 los empleados cuyo salario sea menor al sueldo medio de su departamento

1/ Visualizar dentro de cada departamento los salarios menores a la media

**EJERCICIO 6(\*):**

Borre de EMPLE2 el empleado con la máxima comisión

**EJERCICIO 7(\*):**

Borre de EMPLE2 los empleados que pertenecen al departamento con más número de personas

1/ Departamento con mayor número de personas

**EJERCICIO 8:**

Borre de DEPART2 los departamentos con menos de 4 empleados

1/ Visualizar los departamentos con menos de 4 empleados:

**EJERCICIO 9:**

Borre de DEPART2 el departamento 40 sólo si no tiene empleados

**EJERCICIO 10:**

Borre de AGENDA los empleados del departamento 20 cuyo teléfono es nulo

**EJERCICIO 11(\*):**

Borre los empleados que pertenecen a la Sede Central

**Ejercicio 12(\*):**

El anterior pero solo si existen empleados en Madrid

**Ejercicio 13(\*):**

Borre los centros que tengan empleados viviendo en Madrid

No se puede introducir esa consulta en una variable porque devuelve más de un resultado.

**III.- UPDATE**

Crea una consulta de actualización que cambia los valores de los campos de una tabla especificada basándose en un criterio específico. Su sintaxis es:

**UPDATE Tabla SET Campo1=Valor1, Campo2=Valor2, CampoN=ValorN WHERE Criterio**

UPDATE es especialmente útil cuando se desea cambiar un gran número de registros o cuando éstos se encuentran en múltiples tablas. Puede cambiar varios campos a la vez.

UPDATE no genera ningún resultado. Para saber qué registros se van a cambiar, hay que examinar primero el resultado de una consulta de selección que utilice el mismo criterio y después ejecutar la consulta de actualización.

Si en una consulta de actualización suprimimos la cláusula WHERE todos los registros de la tabla señalada serán actualizados.

**EJERCICIO 1:**

Cree una tabla EMPLE3 que sea una copia de EMPLE con los comandos aprendidos anteriormente

Cree una tabla DEPART3 que sea una copia de DEPART con los comandos aprendidos anteriormente

**EJERCICIO 2:**

Antes de ejecutar, compruebe los registros a actualizar con una sentencia SELECT \* de la condición dada ¿QUÉ CAMBIOS HACE ESTA SENTENCIA?.

UPDATE Emple3 SET APELLIDO='Ruiz' WHERE APELLIDO='SALA';

¿Y ESTA?

UPDATE Emple2 SET APELLIDO=’RUIZ’;

Ejecute ROLLBACK para recuperar la situación anterior. Comente los resultados obtenidos.

**EJERCICIO 3:**

Aumente el salario a todos los empleados un 10%

**EJERCICIO 4:**

Cambie las comisiones nulas por 0

**EJERCICIO 5:**

Cambie los oficios a mayúsculas y la primera letra de todos los apellidos a mayúscula y las siguientes a minúsculas

**EJERCICIO 6:**

Cambie la fecha de alta a la actual a todos los DIRECTORES

**EJERCICIO 7:**

Cambiar el centro del departamento de INVESTIGACION a la sede que está en Sevilla.

**LOS VALORES PARA LOS CAMPOS TAMBIEN PUEDEN OBTENERSE DE UN SUBSELECT**

**UPDATE TABLA SET CAMPO= (SELECT CAMPO …………..) [WHERE]**

**EJERCICIO 8(\*):**

Cambie la fecha de alta del empleado ‘GIL’ a la fecha de alta de ‘NEGRO’

**EJERCICIO 9(\*):**

Cambie el departamento de ‘GIL’ al departamento con menos empleados.

**EJERCICIO 10:**

Doblar el salario de los que trabajan en CONTABILIDAD o en VENTAS

**Nota(\*)** Los ejercicios marcados con \* pueden no ejecutarse en mysql, se escribirá la sentencia en el fichero, aunque para probarla en vez de realizar un delete/update se realizará una select.

<http://www.mundoracle.com/funciones-sql.html?Pg=sql_plsql_3.htm>