TEMA 9: INSERCIONES, BORRADOS Y MODIFICACIONES

1. Inserciones.

La introducción de datos en una tabla se realiza empleando la sentencia INSERT, que admite varios formatos. El primero de los formatos que vamos a estudiar es el siguiente:

```
INSERT [INTO] Nombre_Tabla [(atributo1,…, atributon)]
VALUES ([DEFAULT|valor11],…, [DEFAULT|valor1n]),
       ([DEFAULT|valor21],…, [DEFAULT|valor2n]),… ;
```

Como se puede observar, se debe indicar obligatoriamente el nombre de la tabla en la que se desean insertar los datos y se pueden especificar a continuación entre paréntesis los nombres de los atributos a los que se va a dar valor. Los valores se especifican después de la palabra VALUES entre paréntesis y cada $valor_{ji}$ se asigna al correspondiente $atributo_i$, es decir, $valor_{11}$ se asigna al $atributo_1$, $valor_{12}$ se asigna al $atributo_2$ y así sucesivamente en la primera fila, $valor_{21}$ se asigna al $atributo_1$, $valor_{22}$ se asigna al $atributo_2$ en la segunda fila, y así sucesivamente. Debe haber obviamente una concordancia de tipos entre los atributos y sus correspondientes valores. Se puede en la misma instrucción añadir más de una fila a la tabla. En vez de indicar un valor concreto para un atributo se puede escribir DEFAULT para indicar que se le asigne el valor por defecto especificado para el mismo en la instrucción de creación de la tabla. En los valores de los atributos se puede hacer referencia a atributos indicados previamente en la sentencia INSERT.

Si no se indican los atributos a los que se va a dar valor después del nombre de la tabla, se sobreentiende que se va a dar valor a todos los atributos de la tabla en el mismo orden en el que aparecen en la definición de la tabla.

Por ejemplo, vamos a añadir dos nuevos artículos a la tabla *Articulo*: uno con código *A0022*, descripción *Cuaderno grande de espiral* y precio 2,80 € y otro con código *A0023*, descripción *Paquete de 500 folios DIN A-4* y precio 4,10 €. Para ello deberemos emplear la siguiente instrucción:

```
mysql> insert into Articulo (CodArt, DesArt, PVPArt)
    -> values ('A0022', 'Cuaderno grande de espiral', 2.80),
    -> ('A0023', 'Paquete de 500 folios DIN A-4', 4.10);
Query OK, 2 rows affected (0.10 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

Como se puede observar, el sistema nos informa de que se han añadido dos nuevas filas a la tabla y que no se han producido duplicados ni hay advertencias. Nos habría valido también la siguiente instrucción:

```
mysql> insert into Articulo
    -> values ('A0022', 'Cuaderno grande de espiral', 2.80),
    -> ('A0023', 'Paquete de 500 folios DIN A-4', 4.10);
```

porque vamos a dar valor a todos los atributos y hemos especificado los valores en el mismo orden en el que están definidos en la tabla *Articulo*.

Ahora queremos añadir un nuevo empleado a la tabla *Emple* con los siguientes datos: número *2244*, apellido *Piñeiro*, oficio *ANALISTA*, fecha de alta *15/12/2013*, salario *2200 €* y número de departamento *10*. Como en este caso no vamos a dar valor a todos los atributos, podemos especificar tras el nombre de la tabla, los nombres de los atributos a los que vamos a dar valor:

```
mysql> insert into emple (emp_no, apellido, oficio, fecha_alt, salario, dept_no)
     -> values (2244, 'PIÑEIRO', 'ANALISTA', '2013/12/15', 2200, 10);
Query OK, 1 row affected (0.08 sec)
```

No obstante, si no queremos indicar los nombres de los atributos a los que se va a dar valor, podemos dar valor a todos los atributos en el orden en el que aparecen en la tabla asignando valores nulos a los atributos con valor desconocido:

```
mysql> insert into emple
     -> values (2244, 'PIÑEIRO', 'ANALISTA', null, '2013/12/15', 2200, null, 10);
```

Existe otra modalidad de sentencia INSERT que solo permite añadir una fila a la tabla y cuyo formato se muestra a continuación:

```
INSERT [INTO] Nombre_Tabla
SET atributo1 = {valor1|DEFAULT}, atributo2 = {valor2|DEFAULT}, …
```

Como se puede observar, se deben indicar por cada uno de los atributos a los que se desea dar valor después de la palabra SET su nombre y a continuación el valor que se le desea asignar o la palabra DEFAULT. Así, con la siguiente orden se añade un nuevo empleado a la tabla *Emple* con número *2245*, apellido *Gómez*, oficio *PROGRAMADOR*, fecha de alta el *07/12/2013*, salario *1500 €*, comisión el 10% del salario y departamento el establecido por defecto:

```
mysql> insert into emple
    -> set emp_no=2245, apellido='GÓMEZ', oficio='PROGRAMADOR',
    -> fecha_alt='2013/12/07', salario=1500, comision = salario*0.1,
    -> dept_no= default;
Query OK, 1 row affected (0.09 sec)
```

Existe una tercera modalidad de sentencia INSERT que permite añadir varias filas a una tabla a partir de los datos de otra tabla obtenidos mediante una sentencia SELECT. Su formato es el siguiente:

```
INSERT [INTO] Nombre_Tabla [(atributo1, atributo2, …, atributon)]
SELECT...
```

Igual que con el otro formato de sentencia INSERT, si no se indican los atributos opcionales, se entiende que se va a dar valor a la totalidad de los atributos de la tabla y en el orden en el que aparecen en su definición.

Para explicar esta modalidad de sentencia INSERT primero vamos a crear una nueva tabla llamada *Directores*, con la misma estructura que la tabla *Emple* con la excepción del campo *oficio*. La creamos con la siguiente sentencia CREATE TABLE:

```
mysql> create table directores
    -> (emp_no int(4) unsigned primary key,
    -> apellido varchar(40) not null,
    -> dir int(4) unsigned,
    -> fecha_alt date not null,
    -> salario float(7,2) unsigned not null,
    -> comision float(7,2) unsigned,
    -> dept_no int(3) unsigned default 10 not null,
    -> foreign key(dir) references emple(emp_no),
    -> foreign key(dept_no) references depart(dept_no) on update cascade);
Query OK, 0 rows affected (0.14 sec)
```

Ahora vamos a añadir a esta tabla todos los datos de la tabla *Emple* correspondientes a los directores. Como vamos a dar valor a todos los atributos de la tabla *Directores*, no los indicamos:

```
mysql> insert into directores
    -> select emp_no, apellido, dir, fecha_alt, salario, comision, dept_no
    -> from emple
    -> where oficio= 'DIRECTOR';
Query OK, 3 rows affected (0.04 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

Tras la ejecución de esta inserción, el contenido de la tabla *Directores* es el siguiente:

```
mysql> select * from directores;
+--------+----------+------+------------+---------+----------+---------+
| emp_no | apellido | dir  | fecha_alt  | salario | comision | dept_no |
+--------+----------+------+------------+---------+----------+---------+
|   7566 | JIMÉNEZ  | 7839 | 2014-02-04 | 2300.00 |     0.00 |      20 |
|   7698 | NEGRO    | 7839 | 2014-05-01 | 2200.00 |     0.00 |      30 |
|   7738 | CEREZO   | 7839 | 2014-09-06 | 2210.00 |     0.00 |      10 |
+--------+----------+------+------------+---------+----------+---------+
3 rows in set (0.00 sec)
```

REPLACE sirve, al igual que INSERT, para añadir filas a una tabla. La diferencia con respecto a INSERT es que, si al introducir una fila, se pretende asignar un valor repetido para

un atributo que es clave primaria o único, no se produce un error, sino que el antiguo registro se borrará antes de insertar el nuevo. Su formato es el mismo que el de la orden INSERT, pero sustituyendo la palabra INSERT por REPLACE.

Para probar esta orden veamos, en primer lugar, cuál es el contenido de la tabla *Artículo*:

```
mysql> select * from Articulo;
+--------+----------------------------+--------+
| CodArt | DesArt                     | PVPArt |
+--------+----------------------------+--------+
| A0012  | Goma de borrar             |   0.16 |
| A0022  | Cuaderno grande de espiral |   2.80 |
| A0043  | Bolígrafo azul             |   0.82 |
| A0075  | Lápiz 2B                   |   0.58 |
| A0078  | Bolígrafo rojo normal      |   1.05 |
| A0089  | Sacapuntas                 |   0.26 |
+--------+----------------------------+--------+
6 rows in set (0.00 sec)
```

Pues bien, en esta situación, si pretendemos añadir un nuevo artículo con código A0022 con la sentencia INSERT se producirá un error, como se puede ver a continuación:

```
mysql> insert into Articulo values ('A0022', 'Paquete de 100 folios', 1.45);
ERROR 1062 (23000): Duplicate entry 'A0022' for key 'PRIMARY'
```

Sin embargo, si ejecutamos esta instrucción con REPLACE, se modifica la fila correspondiente al artículo con código A0022:

```
mysql> replace into Articulo values ('A0022', 'Paquete de 100 folios', 1.45);
Query OK, 2 rows affected (0.04 sec)

mysql> select * from Articulo;
+--------+----------------------+--------+
| CodArt | DesArt               | PVPArt |
+--------+----------------------+--------+
| A0012  | Goma de borrar       |   0.16 |
| A0022  | Paquete de 100 folios |   1.45 |
| A0043  | Bolígrafo azul       |   0.82 |
| A0075  | Lápiz 2B             |   0.58 |
| A0078  | Bolígrafo rojo normal |   1.05 |
| A0089  | Sacapuntas           |   0.26 |
+--------+----------------------+--------+
6 rows in set (0.00 sec)
```

2. Modificaciones.

La modificación de datos de una tabla se puede realizar con la sentencia UPDATE, cuyo formato es el siguiente:

```
UPDATE NomTabla
SET atributo1 = valor1, atributo2 = valor2,…, atributon = valorn
[WHERE condición]
[ORDER BY criterio]
```

Se debe especificar, como se puede observar, el nombre de la tabla en la que se desean modificar los datos tras la palabra UPDATE y tras la palabra SET, por cada uno de los atributos cuyo valor se desea modificar, el nombre del atributo y el nuevo valor que se le desea asignar. Tras la cláusula WHERE se indicará la condición que selecciona las filas que se desean modificar. Si se omite esta cláusula, se actualizarán todas las filas de la tabla. Se puede indicar opcionalmente el orden en el que se desean actualizar los registros mediante la cláusula ORDER BY.

Por ejemplo, supongamos que deseamos incrementar en un 5% los precios de los artículos con precio inferior a 1 €. Para ello, deberemos escribir la siguiente sentencia:

```
mysql> update Articulo
    -> set PVPArt = PVPArt + PVPArt*5/100
    -> where PVPArt < 1;
Query OK, 4 rows affected (0.08 sec)
Rows matched: 4  Changed: 4  Warnings: 0
```

Como se puede observar, el sistema nos informa de que en este caso se han modificado 4 filas de la tabla *Articulo*.

En la sentencia UPDATE se pueden incluir subconsultas tanto en la cláusula SET, para especificar el valor que se desea asignar a uno o varios atributos, como en la cláusula WHERE, para indicar las filas para las que se desea llevar a cabo la modificación.

3. Borrados.

La eliminación de datos de una tabla se puede realizar empleando la sentencia DELETE, cuyo formato es el siguiente:

```
DELETE FROM Nombre_Tabla
[WHERE condición]
[ORDER BY criterio]
```

Se debe especificar, como es obvio, el nombre de la tabla de la que se desean borrar los datos. Si no se incluye cláusula WHERE, se borrarán todas las filas de la tabla. En caso contrario, en la cláusula WHERE se indicará la condición que deben cumplir las filas que se desean eliminar. Se puede incluir en la cláusula ORDER BY el orden en el que se desean borrar las filas de la tabla.

Si deseamos eliminar, por ejemplo, de la tabla *Articulo* los productos con precio inferior a 0,30 €, pondremos:

```
mysql> delete from Articulo
    -> where PVPArt < 0.3;
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key
constraint fails (`pedidos`.`lineapedido`, CONSTRAINT `lineapedido_ibfk_2`
FOREIGN KEY (`CodArt`) REFERENCES `articulo` (`CodArt`) ON UPDATE CASCADE)
```

Como se puede observar, nos sale un mensaje de error porque falla una restricción de clave ajena. Y es que ocurre que hay alguna línea de pedido para algún artículo con precio inferior a 0,30 €, por lo que no se puede llevar a cabo el borrado. Sin embargo, sí que podremos borrar algún artículo con precio superior a 2 €:

```
mysql> delete from Articulo
    -> where PVPArt > 2;
Query OK, 2 rows affected (0.06 sec)
```

En este caso, como se puede observar, se eliminan dos artículos.

Si deseamos eliminar los pedidos para los cuales no se ha creado ninguna línea de pedido, precisaremos de una subconsulta en la cláusula WHERE, como se indica a continuación:

```
mysql> delete from Pedido
    -> where RefPed not in (select RefPed from LineaPedido);
Query OK, 2 rows affected (0.08 sec)
```

Si quisiésemos eliminar todos los pedidos de la tabla *Pedido*, deberíamos emplear la siguiente orden:

```
mysql> delete from Pedido
```