
**LENGUAJE
MANIPULACIÓN DE
DATOS SQL. INSERT,
DELETE, UPDATE.**

+

o

•

1. INTRODUCCIÓN

- Lenguaje de manipulación de datos: permite realizar altas, bajas, modificaciones y consultas de datos.
- Vamos a estudiar la inserción de datos, la modificación de los datos existentes y borrado de los datos.

2. INSERCIÓN DE REGISTROS

- **INSERT:** permite insertar nuevas filas o registros en una tabla existente.

`INSERT INTO nombre_tabla (lista_campos) VALUES (lista_valores);`

nombre_tabla: nombre de la tabla a la que quieres añadir los registros

lista_campos: campos de la tabla a la que quieres añadir los nuevos valores. Se puede omitir si se indican todos los valores en lista_valores y en el mismo orden. (ejemplo más adelante).

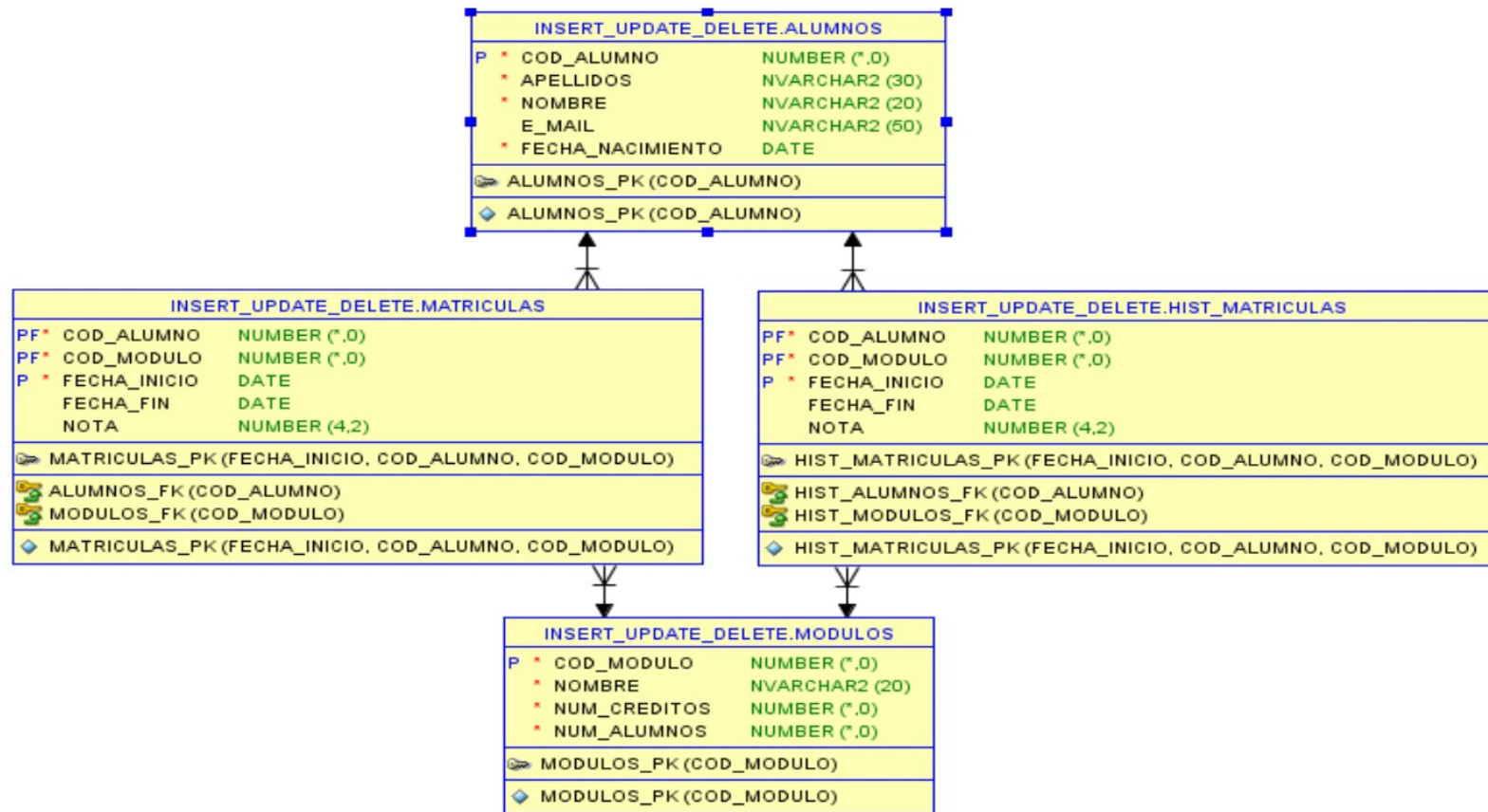
lista_valores: valores nuevos que queremos añadir a la tabla.

2. INSERCIÓN DE REGISTROS

INSERT INTO nombre_tabla (lista_campos) VALUES (lista_valores);

- **Para insertar los datos correctamente:**
- Los elementos de lista_campos y de lista_valores se separan por comas.
- Cada campo de lista_campos debe tener un valor válido en la posición correspondiente de la lista_valores.
- Hay que utilizar los identificadores correspondientes (nombres) de la tabla de la que procedan los campos de la lista_campos y los tipos de datos de la lista_valores tienen que corresponder con los datos de la tabla de la que proceden los datos.
- Cualquier columna que no se encuentre en la lista de columnas recibirá el valor NULL, siempre y cuando no esté definida como NOT NULL, en cuyo caso INSERT fallará.
- Si no se escribe la lista de columnas, se han de introducir valores en todas las columnas.

2. INSERCIÓN DE REGISTROS



2. INSERCIÓN DE REGISTROS

Añade un nuevo alumno a la tabla de alumnos:

```
INSERT INTO ALUMNOS (COD_ALUMNO, APELLIDOS, NOMBRE, E_MAIL, FECHA_NACIMIENTO)  
VALUES (21, 'HUERTAS', 'MIGUEL', 'HUESTAS@GMAIL.COM', '12/10/1955');  
  
INSERT INTO ALUMNOS VALUES (21, 'HUERTAS', 'MIGUEL', 'HUESTAS@GMAIL.COM',  
'12/10/1955');
```

Añade un nuevo módulo a la tabla de módulos:

```
INSERT INTO MODULOS (COD_MODULO, NOMBRE, NUM_CREDITOS, NUM_ALUMNOS)  
VALUES (10, 'DISEÑO GRAFICO', 5, 15);  
  
INSERT INTO MODULOS VALUES (10, 'DISEÑO GRAFICO', 5, 15);
```

Así, se insertan los registros uno a uno. Si queremos realizar un alta masiva de datos en una tabla utilizaremos **SELECT**.

2.1. INSERCIÓN CON SELECT

- Hay que disponer de todos los datos antes de realizar la inserción. Podemos obtener los datos realizando una consulta (Estructura similar a la subconsulta).
- Síntaxis:

```
INSERT INTO NOMBRETABLA1 [(COLUMNA [, COLUMNA] ...)]  
  SELECT {COLUMNA [, COLUMNA] ... | *}  
FROM NOMBRETABLA2 [CLÁUSULAS DE SELECT];
```

2.1. INSERCIÓN CON SELECT

Añade la tabla llamada Histórico de matrículas las matrículas del módulo 1:

```
INSERT INTO HIST_MATRICULAS  
  SELECT COD_ALUMNO, COD_MODULO, FECHA_INICIO, FECHA_FIN, NOTA  
  FROM MATRICULAS  
  WHERE COD_MODULO = 1;
```

No añadimos la cláusula VALUES. De esta forma es más rápido insertar los datos a una tabla.

Los campos que obtengamos como resultado de la consultan deben ser igual en tipo y número que los campos de la tabla en los que los vamos a insertar y, además, seguir el mismo orden.

2.1. INSERCIÓN CON SELECT

Vamos a añadir información que obtenemos de una consulta con información que no podemos obtener de una consulta.

Ejemplo: La fecha de finalización y la nota de la tabla Histórico matrículas toman valores NULL. Añadir los registros del módulo 8 de la tabla de matriculas a la de histórico matrículas poniendo como fecha de finalización la fecha de hoy y la nota un 5.

```
INSERT INTO HIST_MATRICULAS (COD_ALUMNO, COD_MODULO,  
FECHA_INICIO, FECHA_FIN, NOTA)
```

```
SELECT COD_ALUMNO, COD_MODULO, FECHA_INICIO, SYSDATE, 5  
FROM MATRICULAS
```

```
WHERE COD_MODULO = 8;
```

```
INSERT INTO HIST_MATRICULAS  
SELECT COD_ALUMNO, COD_MODULO, FECHA_INICIO, SYSDATE, 5  
FROM MATRICULAS  
WHERE COD_MODULO = 8;
```

2.1. INSERCIÓN CON SELECT

- También podemos realizar el volcado de toda la información de una tabla en otra.
- Ejemplo: Si la tabla histórico matrículas estuviera totalmente vacía, se puede pasar toda la información de la tabla de matrículas de la siguiente forma:

```
INSERT INTO HIST_MATRICULAS SELECT * FROM MATRICULAS;
```

3. MODIFICACIÓN DE REGISTROS

- UPDATE: Permite modificar los valores de determinados registros de las tablas de la base de datos.
- Síntaxis:

```
UPDATE NOMBRE_TABLA  
SET NOMBRE_CAMPO = VALOR [, NOMBRE_CAMPO = VALOR]...  
[WHERE CONDICIÓN];
```

NOMBRE_TABLA: Nombre de la tabla que queremos modificar.

NOMBRE_CAMPO: Nombre de los campos que queremos modificar.

VALOR: Valor nuevo que se le asocia (usando el signo =) al campo especificado.

WHERE: Opcional. Si no se especifica la condición, los cambios afectaran a todos los registros.

3. EJEMPLOS DE MODIFICACIÓN DE REGISTROS

Modificar la nota, incrementándola en 1, de aquellos módulos que hayan finalizado:

UPDATE HIST_MATRICULAS SET NOTA=NOTA+1; ¿Qué pasa si hay alumnos que tengan un 10?

Modificar la nota de los alumnos matriculados para que tengan un 4.9:

UPDATE MATRICULAS SET NOTA = 4.9;

Modificar la nota de los alumnos matriculados para que tengan un 5 y la fecha de finalización sea hoy:

UPDATE MATRICULAS SET NOTA = 5, FECHA_FIN = SYSDATE;

Cambia la nota a 7 de los alumnos matriculados que sean del módulo 1:

UPDATE MATRICULAS SET NOTA = 7 WHERE COD_MODULO=1;

3.1 MODIFICACIÓN CON SELECT

Sintaxis:

```
UPDATE NOMBRE_TABLA  
SET NOMBRE_CAMPO = VALOR [, NOMBRE_CAMPO = VALOR]...  
WHERE CAMPO = SELECT {COLUMNA [, COLUMNA] ... | *}  
FROM NOMBRE_TABLA2 [CLÁUSULAS DE SELECT];
```

Modifica la fecha de finalización para que sea dentro de tres días de aquellos módulos que tengan más de tres alumnos.

```
UPDATE MATRICULAS SET FECHA_FIN= SYSDATE + 3  
WHERE COD_MODULO IN (SELECT COD_MODULO  
FROM MODULOS  
WHERE NUM_ALUMNOS > 3);
```

.

3. MODIFICACIÓN CON SELECT

- Vamos a modificar la consulta anterior para que la fecha de finalización sea igual a la fecha de hoy, pero solo vamos a modificar la fecha de finalización de los cursos que, a los que anteriormente les hayamos dado fecha de finalización.

```
UPDATE MATRICULAS SET FECHA_FIN= SYSDATE  
WHERE FECHA_FIN IS NOT NULL AND  
COD_MODULO IN (SELECT COD_MODULO  
                FROM MODULOS  
                WHERE NUM_ALUMNOS > 3);
```

3.2 MODIFICACIÓN DE CAMPOS CON SELECT

```
UPDATE NOMBRE_TABLA  
SET NOMBRE_CAMPO1 = (SELECT NOMBRE_CAMPO1 ...),  
    NOMBRE_CAMPO2 = (SELECT NOMBRE_CAMPO2 ...)  
WHERE condición;
```

Modificar el módulo de sistemas para que tenga el mismo número de alumnos que el módulo que más alumnos tenga y la media de créditos de todos los módulos:

```
UPDATE MODULOS SET  
NUM_ALUMNOS = (SELECT(MAX(NUM_ALUMNOS)) FROM MODULOS),  
NUM_CREDITOS = (SELECT AVG(NUM_CREDITOS) FROM MODULOS)  
WHERE NOMBRE LIKE 'SISTEMAS';
```

3.2 MODIFICACIÓN DE CAMPOS CON SELECT

Se pueden modificar varios campos realizando una sola consulta. El resultado de la consulta y los campos deben coincidir en el número de campos, orden y tipo.

```
UPDATE NOMBRE_TABLA1  
SET (NOMBRE_CAMPO1, NOMBRE_CAMPO2, NOMBRE_CAMPO2, ...)  
= (NOMBRE_CAMPO1, NOMBRE_CAMPO2, NOMBRE_CAMPO2, FROM TABLA1  
[clausulas SELECT])  
[WHERE condición];
```

Vamos a actualizar los créditos y el número de alumnos del módulo 6 con los datos del módulo 7:

```
UPDATE MODULOS  
SET (NUM_CREDITOS, NUM_ALUMNOS)  
= (SELECT NUM_CREDITOS, NUM_ALUMNOS FROM MODULOS WHERE COD_MODULO = 7 )  
WHERE COD_MODULO=6; LA SUBCONSULTA SOLO DEBE DEVOLVER UN VALOR PARA CADA  
CAMPO.
```


4. BORRADO DE REGISTROS

DELETE: Permite borrar o eliminar registros de una tabla.

DELETE FROM nombre_tabla [WHERE condición];

Si no se indica la condición en where se eliminará todo el contenido de la tabla.

DELETE FROM MATRICULAS; Esta sentencia elimina toda la información de la tabla MATRÍCULAS

DELETE FROM MATRICULAS WHERE COD_MODULO=1; Esta sentencia elimina toda la información de aquellos alumnos del módulo 1 de la tabla de MATRICULAS.

TAMBIÉN SE PUEDEN BORRAR DATOS UTILIZANDO SELECT: Por ejemplo, borrar las matrículas de los alumnos que hayan nacido a partir de los años 90.

```
DELETE FROM MATRICULAS  
WHERE COD_ALUMNO IN  
(SELECT COD_ALUMNO  
FROM ALUMNOS  
WHERE ALUMNOS.FECHA_NACIMIENTO > '01/01/1990');
```