

Bases de Datos

5.- Lenguaje de Manipulación de Datos

IES La Encantá



Índice



- DML
 - Inserción – **Create**
 - Leer - **Read**
 - Actualización - **Update**
 - Borrado - **Delete**
- DML a partir de consultas
- Comprobando las FK
- Exportando e Importando datos



Inserción

- <https://mariadb.com/kb/en/insert/>
- **INSERT [INTO] tabla VALUES**
(valores);
- **INSERT INTO** tabla (campos) **VALUES**
(valores);
- INSERT INTO CLIENTE VALUES ('11111111',
'Pedro', 'Gutiérrez', 'Crespo', 33,
NULL);
- INSERT INTO CLIENTE (dni, nombre,
apellido1, apellido2) VALUES ('11111111',
'Pedro', 'Gutiérrez', 'Crespo');

Restricciones en la inserción



- Los valores insertados deben ajustarse a las columnas de tres formas:
 - 1) Deben ser del mismo tipo de datos.
 - 2) Deben entrar en los límites del tamaño de sus columnas.
 - 3) Debe haber una correspondencia posicional uno a uno entre valores y columnas.
- Si no se especifica la lista de campos, la lista de valores debe seguir el orden de las columnas según fueron creados (comando `DESCRIBE`)



Inserción mediante SET

- **INSERT [INTO] tabla SET col=valor**
[{, col=valor}];
- INSERT INTO CLIENTE SET dni='12345678',
nombre='Juan', apellido1='Casas';



Insertar múltiples filas

- ```
INSERT [INTO] nombreTabla(col1, col2, ...)
VALUES (val1, val2, ...),
 (... , ... , ...);
```
- ```
INSERT INTO CLIENTE VALUES
('11111111', 'Pedro', 'Gutiérrez',
'Crespo', DEFAULT, NULL),
('22222222', 'Pablo', 'García', 'Martínez', D
EFAULT, 'C/ASTURIAL 25'),
('33333333', 'Ana', 'López', 'Sánchez', 'Elch
e', 'Avda. Constitución 1');
```



Inserción con claves ajenas

- Si queremos insertar una clave ajena respecto a un campo PK autoincrementable:
 - 1º insertar tabla con PK
 - 2º insertar tabla con FK y referenciar campo con `LAST_INSERT_ID()`
- `INSERT INTO CURSO (nombre, duracion)
values ('BBDD', 300);`
- `INSERT INTO AULA (nombre, planta, curso)
values ('Info 1', 1, LAST_INSERT_ID());`

Inserción de datos a partir de una consulta



- **INSERT [INTO] tabla [(campos)]
SENTENCIA-SELECT**
 - INSERT INTO CLIENTE (dni, nombre, apellido1) SELECT dni, nombre, apellido1 FROM EMPLEADO WHERE tienda=333;
- Las consultas mediante SELECT las estudiaremos en la siguiente unidad.



Inserción rápida

- Deshabilitar las claves ajenas y crear una transacción:
- `ALTER TABLE tabla DISABLE KEYS;`
- `BEGIN`
- `-- insertamos los datos`
- `COMMIT;`
- `ALTER TABLE tabla ENABLE KEYS;`

Cargando los datos desde un archivo



- <https://mariadb.com/kb/en/load-data-infile/>
- Por defecto, los datos se almacenan en un archivo con las columnas separadas mediante tabuladores (\t) y las filas con saltos de línea (\n).
- ```
LOAD DATA INFILE archivo INTO TABLE
tabla;
```

  - ```
LOAD DATA INFILE '/temp/datos.txt' INTO  
TABLE EMPLEADO;
```

Otros separadores / columnas



- Normalmente, las columnas también se separan con comas (formato CSV).

```
LOAD DATA INFILE archivo INTO TABLE  
tabla FIELDS TERMINATED BY ',';
```

- Si la primera columna es autoincrementable, es conveniente indicar a mano el resto de columnas:

```
LOAD DATA INFILE archivo INTO TABLE  
tabla (col12, col13, ...);
```

Gestión de conflictos con LOAD



- `LOAD DATA [LOCAL] INFILE archivo
[REPLACE | IGNORE] INTO TABLE tabla
[IGNORE num LINES];`
- Si el archivo de carga desde el cliente → LOCAL
- Si ya existen datos que provocan algún conflicto:
→ indicar REPLACE o IGNORE.
- Para que ignore la primera línea del fichero de
datos (cabecera con el nombre de las columnas)
→ al final → `IGNORE 1 LINES`
 - `LOAD DATA INFILE '/temp/datos.txt' REPLACE
INTO TABLE EMPLEADO IGNORE 1 LINES;`



Actualización de datos

- <https://mariadb.com/kb/en/update/>
- **UPDATE** tabla **SET** col=valor
[{,col=valor}] [**WHERE** condicion]
 - UPDATE CLIENTE SET nombre='Juan' WHERE ciudad='Elche'
 - UPDATE CLIENTE SET nombre='Juan', apellido1='Casas' WHERE ciudad='Elche'
 - UPDATE CLIENTE SET ciudad='Elche' WHERE ciudad is null;
 - UPDATE CLIENTE SET ciudad='Elche'
- Las condiciones WHERE las estudiaremos en la siguiente unidad.

Cálculos en SET



- Al modificar un campo, podemos basar su nuevo valor en el valor previo.
 - `UPDATE EMPLEADO SET sueldo = sueldo * 1.05;`
 - `UPDATE EMPLEADO SET sueldo = sueldo + 50 WHERE ciudad='Elche';`



Borrado de filas

- <https://mariadb.com/kb/en/insert/>
- **DELETE FROM** tabla [**WHERE** condicion]
 - DELETE FROM CLIENTE WHERE dni='11111111';
 - DELETE FROM CLIENTE;
- *¡No te olvides poner el WHERE en el DELETE FROM!*
 - https://youtu.be/i_cVJglz_Cs

Vaciado de tablas



- Para vaciar una tabla (más eficiente y reinicia los autonuméricos):
 - **TRUNCATE TABLE** tabla;
 - Realmente hace un `drop table` seguido de `create table`
- Si la tabla tiene referencias de alguna FK, es necesario deshabilitarlas.

```
SET FOREIGN_KEY_CHECKS = 0;  
TRUNCATE TABLE tabla;  
SET FOREIGN_KEY_CHECKS = 1;
```


Actualización o borrado con claves ajenas



- Si queremos borrar/actualizar un dato (PK o AK) que es referenciado por una FK ¿qué sucede?
- Dependerá de cómo se creó la FK:
 - ON DELETE, ON UPDATE
 - NO ACTION: se impide la operación (por defecto)
 - CASCADE: la operación se propaga
 - SET NULL: la clave ajena se pone a NULL



¿Alguna pregunta?