

INSTRUCCIÓN INSERT

- 3 FORMAS DISTINTAS DE INSERTAR DATOS

INSERT INTO ... VALUES

INSERT INTO ... SET

INSERT INTO ... SELECT

INSERT INTO VALUES



INSERT INTO **clientes**(dni,nombre, apellidos, direccion, localidad,fnac, fcarnet, carnet) VALUES

('96401636R','Manuel','Gutierrez Motos',"Calle Barrio Camino",'Almansa','1992-02-19','2010-08-25','B'),

('1057451R','Pedro','Salas Nieto',"Calle Camarreal",'Zaragoza','1970-12-07','1990-06-13','B'),

('66082349R','Alba','Casaus Rodriguez',"Bajada de San Juan",'Móstoles','1997-02-08','2015-02-21','B');

INSERT INTO VALUES

INSERT INTO tabla (col1, col2, ...)

VALUES

{expr1 | DEFAULT} , {expr2 | DEFAULT}, ...),

{expr1 | DEFAULT} , {expr2 | DEFAULT}, ...),

[ON DUPLICATE KEY UPDATE col_name1=expr[, col_name2=expr] ...]

- Entre paréntesis se escriben los nombres de columnas en las que se asignan valores.
- Tras VALUES y entre paréntesis se especifican los valores que se asignan a las columnas para cada fila insertada.
- En una misma INSERT se pueden añadir o insertar varias filas.
- Se usa DEFAULT para indicar que se asigne a la columna el valor por defecto, si es que esa columna se ha diseñado con valor por defecto.
- Las columnas a las que no se asignan valores en INSERT, las que no están en la lista de columnas, reciben el valor por defecto o nulo. Si no admiten nulos, la instrucción da error.
- La cláusula ON DUPLICATE KEY UPDATE establece que, si al insertar una fila se produce un error por PRIMARY KEY duplicada o repetida, se asignen los valores que se indican tras esa cláusula.

INSERT INTO SET

```
INSERT INTO automoviles SET  
matricula='4751JWV',  
marca='Seat',  
modelo= 'Leon 2.0 TDI',  
color='Negro',  
kilometros= 20,  
extras= 'GPS,SN',  
alquilado= false;
```



INSERT INTO SELECT

```
INSERT INTO contratos(matricula,dnicliente, fini, kini)  
SELECT matricula, '08785691K',curdate(),kilometros  
FROM automoviles  
WHERE alquilado=false AND marca='seat';
```



INSTRUCCIÓN UPDATE

UPDATE [IGNORE] tabla | combinación_de_tablas

SET columna1=expresión1, columna2=expresión2,

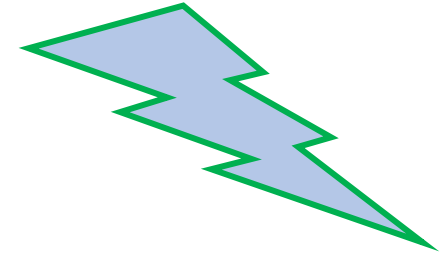
WHERE condición;

- Se modifica una sola tabla, en SET se coloca que valor nuevo tendrá alguno de sus campos y en el WHERE ponemos las condiciones. Si no se pone WHERE, se modifican todas las filas de la tabla.
- La cláusula IGNORE permite modificar las filas que no dan error, aunque haya alguna que si da error.
- Si no ponemos IGNORE, y una fila da error, no modifica ninguna.

INSTRUCCIÓN UPDATE

```
UPDATE automoviles  
SET alquilado=NOT alquilado  
WHERE matricula='7839JDR';
```

```
UPDATE IGNORE contratos  
SET fini=ffin, ffin=fini  
WHERE fini>ffin;
```



```
UPDATE automoviles  
SET matricula='3233JMG'  
WHERE matricula='3273JGH';
```





INSTRUCCIÓN DELETE

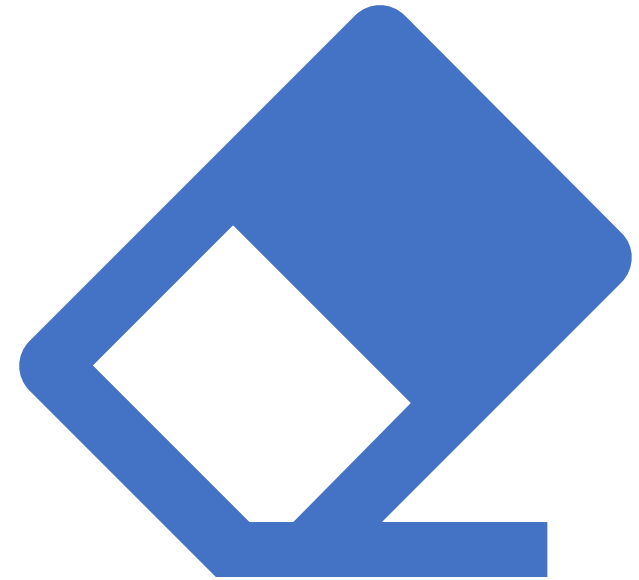
```
DELETE [IGNORE] [tabla1, ...]  
FROM {tabla | combinacion de  
tablas}  
[WHERE condicion]  
[ORDER BY criterio]  
[LIMIT num_filas]
```


INSTRUCCIÓN DELETE

DELETE FROM tabla WHERE

- Instrucción para borrar datos de una tabla y las condiciones de borrado son en la misma tabla.
- Ejemplo:

DELETE FROM contratos
WHERE fini=curdate();



INSTRUCCIÓN DELETE

DELETE FROM **contratos**

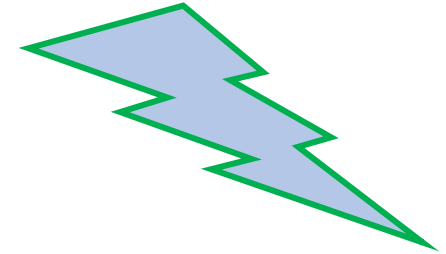
WHERE ffin<date_sub(curdate(),INTERVAL 1 YEAR);

START TRANSACTION;

DELETE FROM contratos

ORDER BY numcontrato DESC LIMIT 1;

ROLLBACK;



INSTRUCCIÓN DELETE

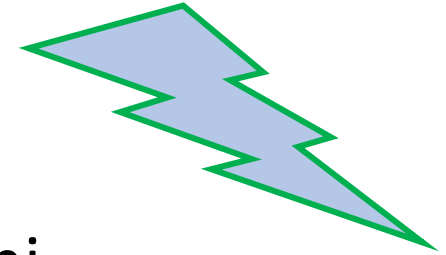
```
DELETE [IGNORE] [tabla1, ...]  
FROM {tabla | combinacion de tablas}  
[WHERE condicion]  
[ORDER BY criterio]  
[LIMIT num_filas]
```

1. Detrás de DELETE se indica la tabla o las tablas en las que se eliminan filas
2. Detrás de FROM o USING se indica la tabla o la combinación de tablas (INNER JOIN, LEFT JOIN, producto cartesiano) sobre las que se va a condicionar el borrado o eliminación.
Necesariamente la tabla de la que se eliminan filas tiene que formar parte de la combinación de tablas.



INSTRUCCIÓN DELETE PARA BORRAR EN VARIAS TABLAS.

```
DELETE contratos
FROM contratos INNER JOIN clientes ON dnicliente=dni
WHERE nombre='carlos javier'
AND apellidos='lopez carvajal';
```



- Borrás de una tabla, pero la condición está en otra tabla.
- En este caso, borras los contratos de Carlos Javier Lopez Carvajal



INSTRUCCIÓN DELETE PARA BORRAR EN VARIAS TABLAS.

- Imagina querer borrar todos los contratos de Mariano Dorado

`DELETE FROM clientes`

`WHERE nombre='mariano'`

`AND apellidos='dorado';`

- Esto funcionaría si tuviéramos borrado en cascada, al borrar los clientes, borras sus contratos.
- Pero si tienes on delete no action, esto no funciona.



INSTRUCCIÓN DELETE PARA BORRAR EN VARIAS TABLAS.

- Una opción para borrar los contratos y al cliente Mariano Dorado podría ser la siguiente:

```
DELETE clientes,contratos
```

```
FROM contratos INNER JOIN clientes ON dnicliente=dni
```

```
WHERE nombre='mariano'
```

```
AND apellidos='dorado';
```

- Pero si tienes on delete no action, esto TAMPOCO FUNCIONA.



INSTRUCCIÓN DELETE PARA BORRAR EN VARIAS TABLAS.

- La opción más adecuada, entonces, es hacer el borrado en dos instrucciones distintas. Primero borras los contratos y luego al cliente:

```
DELETE contratos FROM contratos INNER JOIN clientes ON  
dnicliente=dni
```

```
WHERE nombre='mariano' AND apellidos='dorado';
```

```
DELETE FROM clientes
```

```
WHERE nombre='mariano' AND apellidos='dorado';
```

