

# **Bases de Datos**

## **Unidad 7: Edición avanzada de los datos. INSERT**

## Inserción de filas. INSERT

Hasta ahora hemos visto inserciones con INSERT usando tres sintaxis:

**INSERT ..... VALUES**

**INSERT ..... SET**

**INSERT ..... SELECT**

Con las dos primeras sintaxis hemos insertado filas en una tabla con valores constantes, obtenidos de una función o resultado de una expresión calculada.

Pero hay más posibilidades:

**Podemos cargar valores obtenidos de una subconsulta.**

***Ejemplo 1: queremos insertar mediante una sola instrucción dos nuevos contratos realizados en la fecha actual por el cliente de DNI 11223344M para los automóviles de matrícula 5031JHL y 4738JBJ. En kilómetros iniciales del contrato, se deben cargar los kilómetros que hay registrados en los automóviles de esas matrículas.***

Una solución, no adecuada para realizar esto, sería mirar los kilómetros que hay en la tabla automóviles para esos automóviles y cargar en INSERT esos valores. Esto no es adecuado ya que cualquier ejercicio, salvo que se diga lo contrario, se debe resolver con una instrucción. No se podrían consultar primero los kilómetros de los automóviles con SELECT. **Pero, si se pudiera hacer, la solución sería:**

```
INSERT INTO contratos (matricula, dnicliente, fini, kini)  
VALUES ('5031JHL', '11223344M', curdate(), 24796),  
('4738JBJ', '11223344M', curdate(), 8008);
```

## Inserción de filas. INSERT

*Una solución, que sería perfectamente válida para resolver el ejemplo anterior, es hacer uso de la sintaxis INSERT ... SELECT:*

```
INSERT INTO contratos (matricula, dncliente, fini, kini)
SELECT matricula, '11223344M', curdate(), kilometros
FROM automoviles
WHERE matricula='5031JHL' OR matricula='4738JBJ';
```

## Inserción de filas. INSERT

*Pero hay otra solución que usando la sintaxis INSERT ... VALUES permite resolver el ejercicio. Pero en la instrucción tendremos que usar SUBCONSULTAS para obtener los kilómetros de cada automóvil:*

```
INSERT INTO contratos (matricula, dnicliente, fini, kini)
VALUES
('5031JHL' , '11223344M', curdate(), (SELECT kilometros FROM automoviles
WHERE matricula='5031JHL')),
('4738JBJ', '11223344M', curdate(), (SELECT kilometros FROM automoviles WHERE
matricula='4738JBJ'));
```

## Inserción de filas. INSERT

***Ejemplo 2: Añadir un nuevo contrato con fecha de hoy realizado por Sandra Flores Jorge sobre el automóvil de matrícula '2058JGF' poniendo los kilómetros iniciales a los kilómetros del automóvil.***

```
INSERT INTO contratos (matricula,dnicliente,fini,kini)
VALUES ('2058JGF' ,
(SELECT dni FROM clientes WHERE nombre='Sandra' AND apellidos='flores jorge'),
curdate(),
(SELECT kilometros FROM automoviles WHERE matricula='2058JGF'));
```

***También se podría solucionar sin subconsultas, usando la sintaxis INSERT ... SELECT, pero habría que hacer la SELECT sobre un producto cartesiano:***

```
INSERT INTO contratos (matricula,dnicliente,fini,kini)
SELECT '2058JGF' , dni, curdate(), kilometros
FROM clientes, automoviles
WHERE matricula='2058JGF' AND nombre='Sandra' AND apellidos='flores jorge';
```

**¿Podemos sustituir el producto cartesiano por un INNER JOIN ON...?**

***Ejemplo 3: Añadir un nuevo contrato con fecha de hoy realizado por Anais Rodríguez sobre el automóvil más barato de los que no tienen un contrato sin finalizar actualmente. En kilómetros iniciales pondremos el valor cero.***

***Hay que tener en cuenta que para sacar el automóvil más barato de los que no están contratados actualmente (según la fecha final de contratos), haríamos:***

***SELECT matricula FROM automoviles WHERE alquilado=false ORDER BY precio LIMIT 1;***

Por lo que la instrucción para insertar el contrato sería:

***Ejemplo 3: Añadir un nuevo contrato con fecha de hoy realizado por Anais Rodríguez sobre el automóvil más barato de los que no tienen un contrato sin finalizar actualmente. En kilómetros iniciales pondremos el valor cero.***

***Hay que tener en cuenta que para sacar el automóvil más barato de los que no están contratados actualmente (según la fecha final de contratos), haríamos:***

***SELECT matricula FROM automoviles WHERE alquilado=false ORDER BY precio LIMIT 1;***

Por lo que la instrucción para insertar el contrato sería:

**INSERT INTO contratos (matricula,dnicliente,fini,kini)  
VALUES (  
(SELECT matricula FROM automoviles WHERE alquilado=false ORDER BY precio  
LIMIT 1),  
(SELECT dni FROM clientes WHERE nombre='Anais' AND apellidos='Rodriguez'),  
curdate(),  
0);**



***Ejemplo 4: Añadir un nuevo contrato con fecha de hoy realizado por la cliente de dni ' 11223344M ' sobre los 3 automóviles más baratos. En kilómetros iniciales pondremos el valor cero.***

***Esto no lo podemos intentar hacer con una subconsulta de esta forma (estaríamos tratando de insertar 3 matrículas en un mismo VALUES, por tanto, en un mismo contrato:***

```
INSERT INTO contratos (matricula,dnicliente,fini,kini)  
VALUES (  
(SELECT matricula FROM automoviles ORDER BY precio LIMIT 3),  
' 11223344M ',  
curdate(),  
0);
```

Pero no podemos hacerlo así, ya que estaríamos tratando de asignar tres matrículas distintas a un único registro . Hay que crear tres registros distintos, es decir, tres contratos, uno por cada coche.

***Ejemplo 4:*** *Añadir nuevos contratos con fecha de hoy realizados por la cliente de dni '11223344M' sobre los 3 automóviles más baratos. La única forma de hacerlo con una sola instrucción es mediante la sintaxis INSERT ... SELECT*

```
INSERT INTO contratos (matricula,dnicliente,fini,kini)
SELECT matricula, '11223344M', curdate(), kilometros
FROM automoviles
WHERE matricula NOT IN (SELECT matricula FROM contratos WHERE ffin IS NULL)
ORDER BY precio LIMIT 3;
```

La condición establecida con where es necesaria para no incluir contratos de automóviles actualmente en alquiler.

## Inserción de filas. INSERT

*Ejemplo 5: Añadir un nuevo contrato con fecha de hoy realizado por Anais Rodríguez sobre el automóvil más barato de los que no tienen un contrato sin finalizar actualmente. En kilómetros iniciales pondremos el valor cero.*

*Hay que tener en cuenta que para sacar el automóvil más barato de los que no están contratados actualmente (según la fecha final de contratos), haríamos:*

*SELECT matricula FROM automoviles WHERE matricula NOT IN (SELECT matricula FROM contratos WHERE ffin IS NULL) ORDER BY precio LIMIT 1;*

Por lo que la instrucción para insertar el contrato sería, en principio:

```
INSERT INTO contratos (matricula,dnicliente,fini,kini)
VALUES (
(SELECT matricula FROM automoviles WHERE matricula NOT IN (SELECT matricula FROM
contratos WHERE ffin IS NULL) ORDER BY precio LIMIT 1),
(SELECT dni FROM clientes WHERE nombre= 'Anais' AND apellidos= 'Rodriguez'),
curdate(), 0);
```

**Pero esto da error, NO SE PUEDE TENER EN UNA SUBCONSULTA LA TABLA EN LA QUE SE ESTÁN INSERTANDO DATOS. En definitiva, no se puede usar contratos en la subconsulta para insertar datos en esa misma tabla contratos.**

Error Code: 1093. You can't specify target table 'contratos' for update in FROM clause

## Inserción de filas. INSERT

*Ejemplo 5: Añadir un nuevo contrato con fecha de hoy realizado por Anais Rodríguez sobre el automóvil más barato de los que no tienen un contrato sin finalizar actualmente. En kilómetros iniciales pondremos el valor cero.*

**LA SOLUCIÓN ES HACER UN RENOMBRADO A TABLA DE UNA CONSULTA SOBRE CONTRATOS.**

**La SELECT sería así:**

*SELECT matricula FROM automoviles WHERE matricula NOT IN (SELECT a.matricula FROM (SELECT matricula FROM contratos WHERE ffin IS NULL ) AS a) ORDER BY precio LIMIT 1;*

Por lo que la instrucción para insertar el contrato sería, en principio:

**INSERT INTO contratos (matricula,dnicliente,fini,kini)**

**VALUES (**

**(SELECT matricula FROM automoviles WHERE matricula NOT IN (SELECT a.matricula FROM (SELECT matricula FROM contratos WHERE ffin IS NULL ) AS a) ORDER BY precio LIMIT 1),  
(SELECT dni FROM clientes WHERE nombre= 'Anais' AND apellidos= 'Rodriguez'),  
curdate(), 0);**

# **Bases de Datos**

## **Unidad 7: Edición avanzada de los datos. UPDATE**

## Modificación de contenidos de filas. UPDATE

Hemos visto que la sintaxis de **UPDATE** es:

**UPDATE** [IGNORE] tabla | **combinación\_de\_tablas** SET columna1=expresión, columna2=expresión, ..... **WHERE** condicion;

En la instrucción **UPDATE** podemos usar una combinación de tablas para:

- Indicar las tablas en las que se modifican datos.
- Establecer las condiciones que cumplen las filas en las que se modifican datos.

***Ejemplo 1: Modificar la columna alquilado de la tabla automóviles para que todos los automóviles que están actualmente contratados queden marcados como no disponibles y que los que no están contratados figuren como disponibles.***

**UPDATE** automoviles SET alquilado=false;

**UPDATE** automoviles INNER JOIN contratos  
**ON** contratos.matricula=automoviles.matricula  
**SET** alquilado=true **WHERE** ffin IS NULL;

## Modificación de contenidos de filas. UPDATE

No es normal, aunque se puede, modificar datos de columnas de dos o más tablas en UPDATE.

***Ejemplo 2: Modificar la columna alquilado de la tabla automóviles para que todos los automóviles que están actualmente contratados queden marcados como no disponibles y que la fecha inicial de los contratos actuales de esos automóviles, la fecha de inicio sea un día superior a la que tienen actualmente.***

```
UPDATE automoviles INNER JOIN contratos
ON contratos.matricula=automoviles.matricula
SET alquilado=true,fini=adddate(fini, interval 1 day)
WHERE ffin IS NULL;
```

## Modificación de contenidos de filas. UPDATE

*Ejemplo 3: Suponiendo que añadimos una columna importe a la tabla contratos, establecer el importe de cada contrato finalizado como el precio por los días alquilado.*

**ALTER TABLE contratos ADD COLUMN importe decimal(6,2);**



## Modificación de contenidos de filas. UPDATE

*Ejemplo 3: Suponiendo que añadimos una columna importe a la tabla contratos, establecer el importe de cada contrato finalizado como el precio por los días alquilado.*

```
ALTER TABLE contratos ADD COLUMN importe decimal(6,2);
```

```
UPDATE automoviles INNER JOIN contratos  
ON contratos.matricula=automoviles.matricula  
SET importe=precio*datediff(ffin,fini)  
WHERE ffin IS NOT NULL;
```

## Modificación de contenidos de filas. UPDATE

Dentro de UPDATE, podemos usar subconsultas, tanto para obtener los valores que se asignan con SET como para establecer las condiciones WHERE.

***Ejemplo 4: Modificar las fechas de inicio y de finalización de los contratos realizados por Mariano Dorado para que tengan una fecha un día superior a la que tienen.***

## Modificación de contenidos de filas. UPDATE

Dentro de UPDATE, podemos usar subconsultas, tanto para obtener los valores que se asignan con SET como para establecer las condiciones WHERE.

***Ejemplo 4: Modificar las fechas de inicio y de finalización de los contratos realizados por Mariano Dorado para que tengan una fecha un día superior a la que tienen.***

**UPDATE contratos**

**SET fini=adddate(fini,INTERVAL 1 DAY),ffin=adddate(ffin,INTERVAL 1 DAY)**

**WHERE dnicliente=**

**(SELECT dni FROM clientes WHERE nombre='Mariano' AND apellidos='Dorado');**

Este ejemplo también se podría haber resuelto fácilmente con UPDATE usando un INNER JOIN entre clientes y contratos.

## Modificación de contenidos de filas. UPDATE

*Ejemplo 5: Establecer que el contrato 25 fue realizado por Antonio Diaz Vera.*

UPDATE contratos

SET dnicliente=

(SELECT dni FROM clientes WHERE nombre='Antonio' AND apellidos='Diaz vera')

WHERE numcontrato=25;

*Ejemplo 6: Establecer que en los kilómetros de los automóviles se carguen los kilómetros finales del último contrato finalizado de cada automóvil.*

```
UPDATE automoviles SET kilometros=  
(SELECT max(kfin) FROM contratos WHERE kfin IS NOT NULL AND  
contratos.matricula=automoviles.matricula);
```

Como puede verse, en la subconsulta, se compara la matricula de cada contrato con la matricula del automóvil que se está modificando. La ejecución de esta instrucción supone que por cada automóvil, se ejecuta la subconsulta para obtener el máximo valor de kilómetros finales para ese automóvil.

Si un automóvil no tuviera contratos, se cargaría el valor NULL en kilómetros, lo cual no sería muy adecuado.

## Modificación de contenidos de filas. UPDATE

*Ejemplo 7: Para que no ocurra lo anterior, modifica los kilómetros sólo de automóviles con contratos finalizados para que contengan los kilómetros finales de cada uno de los últimos contratos finalizados.*

Una posible solución es:

```
UPDATE automoviles SET kilometros=
(SELECT max(kfin) FROM contratos WHERE kfin IS NOT NULL AND
contratos.matricula=automoviles.matricula)
WHERE matricula IN
(SELECT matricula FROM contratos WHERE ffin IS NOT NULL);
```

Otra posible solución, quizás más complicada, es usar una subconsulta dentro de la referencia de tablas renombrada a tabla C.

```
UPDATE automoviles INNER JOIN (SELECT matricula,max(kfin) AS m FROM
contratos WHERE ffin IS NOT NULL GROUP BY matricula) AS c ON
c.matricula=automoviles.matricula SET kilometros=m;
```

## Modificación de contenidos de filas. UPDATE

*Ejemplo 8: Establecer que el contrato número 26 ha sido realizado por el mismo cliente del contrato número 4.*

Si hacemos, como puede suponerse:

```
UPDATE contratos
SET dnicliente=
(SELECT dnicliente FROM contratos WHERE numcontrato=4)
WHERE numcontrato=26;
```

Nos da un error en la instrucción, bastante común, debido a que no se puede usar en una subconsulta la misma tabla sobre la que se están modificando datos.

Error Code: 1093. You can't specify target table 'contratos' for update in FROM clause

Para solucionarlo hay que usar un renombrado de subconsulta a tabla:

```
UPDATE contratos
SET dnicliente=
(SELECT a.dnicliente FROM (SELECT * FROM contratos WHERE numcontrato=4) AS a)
WHERE numcontrato=26;
```

## Modificación de contenidos de filas. UPDATE

*Ejemplo 9: Suponiendo que en la tabla clientes hay una columna NUMCONTRATOS para contener cuantos contratos ha realizado cada cliente, establecer en la columna NUMCONTRATOS el número de contratos realizados por Mariano Dorado.*

### UPDATE clientes

```
SET numcontratos= (SELECT count(*) FROM contratos WHERE dni=dnicliente)  
WHERE nombre='Mariano' AND apellidos='Dorado';
```

- En la instrucción UPDATE establecemos que se modifique sólo el cliente llamado Mariano Dorado.
- Para ese cliente, en la subconsulta se obtiene cuantos contratos ha realizado y se asigna ese valor a numcontratos.
- En la subconsulta se cuentan sólo contratos en los que **dnicliente** coincide con el valor de **dni** (el DNI del cliente que se está modificando, es decir, de Mariano Dorado).

¿Es coherente la subconsulta propuesta atendiendo al enunciado del ejemplo?

¿De qué manera podemos mejorar la condición de la cláusula WHERE?



## Modificación de contenidos de filas. UPDATE

### UPDATE clientes

SET numcontratos= (SELECT count(\*) FROM contratos WHERE dni=dnicliente)  
WHERE nombre='Mariano' AND apellidos='Dorado';

- Dicho de otra forma, cuando tenemos una subconsulta dentro de un update tenemos que tener en cuenta que esa subconsulta se va a volver a evaluar por cada línea de la tabla que estamos modificando.
- En este ejemplo, por cada cliente en clientes se evaluará la subconsulta: **SELECT count(\*) FROM contratos WHERE dni=dnicliente.**
- Por eso no tenemos que poner group by, tenemos que usar un where para filtrar todos los contratos y quedarnos solo los que corresponden al dni del cliente que se esté evaluando en ese momento (cuando se modifica la primera línea el dni de la primera, en la segunda el de la segunda, etc.)

## Modificación de contenidos de filas. UPDATE

***Ejemplo 10: Suponiendo que en la tabla clientes hay una columna NUMCONTRATOS para contener cuantos contratos ha realizado cada cliente, establecer en la columna NUMCONTRATOS el número de contratos realizados por cada cliente.***

### UPDATE clientes

**SET numcontratos= (SELECT count(\*) FROM contratos WHERE dni=dnicliente);**

- En la instrucción UPDATE establecemos que se modifiquen todos, no hay WHERE aplicado.
- Al ejecutar una UPDATE que afecta a varias filas y que tiene subconsultas asignadas a valores con SET, por cada fila, se realiza la subconsulta correspondiente. Por tanto, en nuestro caso, por cada cliente (por cada dni), se obtienen cuantos contratos tienen un dnicliente igual al dni del cliente que se está modificando.
- Así a cada cliente se le asigna el número de contratos obtenidos en la subconsulta (los contratos del cliente). A los que no tienen contratos se les asigna de forma correcta el valor cero.