



[Inicio](#) > [SQL](#) > [Lecciones SQL](#) >

## P03 Modificar, claves ajenas y filtros adicionales

### CONTENIDOS

- 1 Más manipulación de datos: Update, modificando valores
- 2 Clave ajena
- 3 Integridad referencial
  - 3.1 IR y delete
  - 3.2 IR y update
- 4 Modificando una definición de clave ajena
- 5 Expresiones de selección de filas
  - 5.1 Rangos
  - 5.2 Subcadenas de caracteres
- 6 Información adicional

### Más manipulación de datos: Update, modificando valores

Update forma parte de la órdenes clásicas de manipulación de datos: select, insert, delete y update. Simplemente, por cuestiones de equilibrio de contenidos la hemos dejado para esta sesión.

Antes de comenzar, reconstruyamos la base de datos:

```
delete from imparte;  
delete from profesores;  
delete from asignaturas;
```

```
insert into asignaturas (codigo, descripcion, creditos, creditosp)  
values ('HI', 'HISTORIA DE LA INFORMATICA', 4.5, NULL);  
insert into asignaturas (codigo, descripcion, creditos, creditosp)  
values ('FBD', 'FUNDAMENTOS DE LAS BASES DE DATOS', 6.0, 1.5);  
insert into asignaturas (codigo, descripcion, creditos, creditosp)  
values ('DGBD', 'DISEÑO Y GESTION DE BASES DE DATOS', 6.0, 3.0);  
insert into asignaturas (codigo, descripcion, creditos, creditosp)  
values ('PC', 'PROGRAMACION CONCURRENTES', 6.0, 1.5);
```

```
insert into asignaturas (codigo, descripcion, creditos, creditosp)
values ('FP', 'FUNDAMENTOS DE LA PROGRAMACION', 9.0, 4.5);
```

```
insert into profesores (dni, nombre, categoria, ingreso) values
('21111222', 'EVA GOMEZ', 'TEU', '1993-10-01');
```

```
insert into profesores (dni, nombre, categoria, ingreso) values
('21222333', 'MANUEL PALOMAR', 'TEU', '1989-06-16');
```

```
insert into profesores (dni, nombre, categoria, ingreso) values
('21333444', 'RAFAEL ROMERO', 'ASO6', '1992-06-16');
```

```
insert into imparte (dni, asignatura) values ('21111222', 'FBD');
```

```
insert into imparte (dni, asignatura) values ('21111222', 'DGBD');
```

```
insert into imparte (dni, asignatura) values ('21333444', 'PC');
```

La sentencia UPDATE nos permite modificar la información contenida en una tabla.

```
UPDATE nombreTabla [aliasTabla]
SET { {columna=expresion | columna=subconsulta} |
listaColumnas=subconsulta}
[ WHERE condición]
```

No se pueden modificar varias tablas a la vez en una misma sentencia. Para modificar los valores de varias tablas varias habrá que ejecutar tantas sentencias UPDATE como tablas queramos modificar. (NOTA: al igual que con *delete* MySQL sí proporciona sintaxis específica para poder hacerlo pero otros SGBD no).

```
select codigo,creditos,creditosp from asignaturas;
```

codigo	creditos	creditosp
DGBD	6.0	3.0
FBD	6.0	1.5
FP	9.0	4.5
HI	4.5	null
PC	6.0	1.5

```
update asignaturas set creditos = 0;
```

```
select codigo,creditos from asignaturas;
```

codigo	creditos
DGBD	0.0
FBD	0.0
FP	0.0
HI	0.0

PC	0.0
----	-----

Cuando se desea modificar más de una columna se indicará la lista de columnas y valores separadas por comas:

Modifica los créditos de las asignaturas a 4, y los créditos prácticos a 2.

update asignaturas set **creditos=4, creditosp=2;**

codigo	creditos	creditosp
DGBD	4.0	2.0
FBD	4.0	2.0
FP	4.0	2.0
HI	4.0	2.0
PC	4.0	2.0

En el caso de que se indique alguna condición, se modificarán sólo aquellas filas de la tabla que cumplan la condición o condiciones impuestas:

```
select * from profesores;
```

dni	nombre	categoria	ingreso
21111222	EVA GOMEZ	TEU	1993-10-01
21222333	MANUEL PALOMAR	TEU	1989-06-16
21333444	RAFAEL ROMERO	ASO6	1992-06-16

Modifique la fecha de ingreso a 1 de enero de 2003 sólo a aquellos profesores cuya categoría sea TEU.

```
update profesores
set ingreso='2003-01-01'
where categoria = 'TEU';
```

```
select * from profesores;
```

dni	nombre	categoria	ingreso
21111222	EVA GOMEZ	TEU	2003-01-01
21222333	MANUEL PALOMAR	TEU	2003-01-01
21333444	RAFAEL ROMERO	ASO6	1992-06-16

## Clave ajena

Una clave ajena es una columna o columnas en una tabla que almacena valores que hacen referencia a datos en otras tablas. Una clave ajena es una referencia, no sirve para

identificar, por lo tanto admite duplicados y nulos. Por otro lado, la clave ajena siempre tiene la misma estructura y tipos de datos que la clave primaria de la tabla a la que está apuntando.

PROFESORES ( DNI varchar(10), nombre varchar(40), categoria char(4), ingreso date )  
Clave primaria (DNI)

ASIGNATURAS ( codigo char(5), descripcion varchar(35), creditos decimal(3,1), creditosp decimal(3,1) )  
Clave primaria (codigo)

IMPARTE ( dni : varchar(10), asignatura : char(5) )  
Clave primaria: (dni, asignatura)

**Clave ajena: dni → PROFESORES**

**Clave ajena: asignatura → ASIGNATURAS**

Pero lo más importante de las restricciones de clave ajena es que permiten garantizar la **integridad referencial** de los datos almacenados en nuestra base de datos. Más adelante explicaremos qué es esto pero antes debemos crear la base de datos de ejemplo.

## Motores de MySQL

Lo visto hasta ahora constituye el estándar de SQL y es la sintaxis admitida en los SGBD de mayor penetración comercial.

Sin embargo, MySQL es un producto peculiar ya que ofrece varias opciones de almacenamiento y gestión de las tablas en aras de ofrecer alternativas que mejoren el rendimiento o la integridad de datos. En particular podemos elegir entre tablas **MyISAM** e **InnoDB**.

Si queremos mantener integridad referencial entre nuestras tablas es imprescindible que especifiquemos que queremos usar el motor **InnoDB**. En una instalación estándar de MySQL, **si no decimos nada, la creación de una tabla se hace por defecto en MyISAM y tiene un efecto importantísimo: si definimos claves ajenas, el sistema las ignora y no revisa la integridad referencial.**

**Todas las tablas implicadas en una clave ajena han de estar definidas en este motor**, tanto la que aloja a la clave ajena como a la que hace referencia, y **es obligatorio indicar la columna o columnas que constituyen la clave primaria** en esta segunda. Aunque todavía no hemos definido clave ajena ninguna a continuación sí lo vamos a hacer, y puesto que van a hacer referencia a PROFESORES y ASIGNATURAS estas tablas también deben almacenarse como **INNODB**.

Por tanto, y puesto que queremos garantizar las restricciones de clave ajena, hemos de completar las definiciones manejadas hasta ahora. Antes de comenzar, reconstruyamos la base de datos:

```
drop table if exists imparte;  
// "if exists" comprueba primero si existe tal tabla antes de  
eliminarla
```

```
// se utiliza para evitar el aviso o error de "tabla no
encontrada"
drop table if exists profesores;
drop table if exists asignaturas;

create table profesores (
DNI varchar(10),
nombre varchar(40),
categoria char(4),
ingreso date,
primary key (DNI)) engine=innodb;

create table asignaturas (
codigo char(5),
descripcion varchar(35),
creditos decimal(3,1),
creditosp decimal(3,1),
coordina varchar(10),
primary key (codigo),
foreign key (coordina) references profesores(DNI) )
engine=innodb;

insert into profesores (dni, nombre, categoria, ingreso) values
('21111222', 'EVA GOMEZ', 'TEU', '1993-10-01');
insert into profesores (dni, nombre, categoria, ingreso) values
('33222333', 'MANUEL PALOMAR', 'TEU', '1989-06-16');
insert into profesores (dni, nombre, categoria, ingreso) values
('44333444', 'RAFAEL ROMERO', 'ASO6', '1992-06-16');

insert into asignaturas (codigo, descripcion, creditos, creditosp,
coordina)
values ('HI', 'HISTORIA DE LA INFORMATICA', 4.5, NULL, NULL);
insert into asignaturas (codigo, descripcion, creditos, creditosp,
coordina)
values ('FBD', 'FUNDAMENTOS DE LAS BASES DE DATOS', 6.0, 1.5,
'21111222');
insert into asignaturas (codigo, descripcion, creditos, creditosp,
coordina)
values ('DGBD', 'DISEÑO Y GESTION DE BASES DE DATOS', 6.0, 3.0,
'21111222');
insert into asignaturas (codigo, descripcion, creditos, creditosp,
coordina)
values ('PC', 'PROGRAMACION CONCURRENTES', 6.0, 1.5, NULL);
insert into asignaturas (codigo, descripcion, creditos, creditosp,
coordina)
values ('FP', 'FUNDAMENTOS DE LA PROGRAMACION', 9.0, 4.5, NULL);
```

## Integridad referencial

La integridad referencial es la propiedad de las claves ajenas que nos asegura que todas las referencias desde una hacia otra tabla son consistentes. Más o menos se formula como que "una clave ajena o es totalmente nula o contiene valores almacenados previamente en la clave primaria a la que apunta".

Lo que se consigue con esto es que, por ejemplo, no digamos que una asignatura está siendo coordinada por un profesor que no existe. Por ejemplo, el sistema no nos va a permitir hacer la siguiente inserción:

```
insert into asignaturas (codigo, descripcion, creditos, creditosp,
coordina)
values ('FP2', 'FUNDAMENTOS DE LA PROGRAMACIÓN 2', 6.0, 3.0,
'55555666');
```

Puesto que en PROFESORES no hay ninguna fila con ese DNI, la orden no tiene otro efecto que mostrar un mensaje de error.

```
Cannot add or update a child row: a foreign key constraint fails
(`xxx`.`asignatura`, CONSTRAINT ... FOREIGN KEY (`coordina`)
REFERENCES `profesores` (`DNI`))
```

Ahora vamos a crear IMPARTE. La tabla IMPARTE relaciona profesores y asignaturas informando de qué profesores imparten qué asignatura:

```
IMPARTE ( dni : varchar(10), asignatura : char(5) )
Clave primaria: (dni, asignatura)
Clave ajena: dni → PROFESORES
Clave ajena: asignatura → ASIGNATURAS
```

```
create table imparte (
dni varchar(10),
asignatura char(5),
primary key (dni,asignatura),
foreign key (dni) references profesores (dni),
foreign key (asignatura) references asignaturas (codigo)
engine=innodb;
```

La integridad referencial exige que el valor que tenga la columna dni exista en la tabla PROFESORES; de la misma manera cada código de asignatura deberá existir en la tabla ASIGNATURAS.

En este momento, los DNI almacenados en PROFESORES son:

dni
21111222
21111222
21333444

y los códigos de ASIGNATURAS:

codigo
DGBD
FBD
FP
HI
PC

Inserta la información de que el profesor identificado como 21333444 imparte la asignatura identificada como DGBD

```
insert into imparte (dni, asignatura) values ('21333444','DGBD');
select * from imparte; XXXX
```

dni	asignatura
21111222	DGBD

## IR y delete

La integridad referencial también evita que se eliminen datos que están siendo usados en alguna clave ajena. Sólo se podrán borrar aquellas filas que no estén siendo referenciadas, a través de ninguna clave ajena, desde otra tabla. Por ejemplo, la asignatura DGBD es impartida por el profesor con dni 21333444: no se podrá borrar la asignatura si antes no se eliminan las filas correspondientes a esa asignatura en la tabla imparte.

Desde otro punto de vista, si el sistema nos permitiera eliminar la fila de ASIGNATURAS ('DGBD', 'DISEÑO Y GESTION DE BASES DE DATOS', 6.0, 3.0, '21111222'), en IMPARTE estaríamos diciendo que Eva Gómez está impartiendo una asignatura que no sabemos cuál es. Esto es lo que evita la integridad referencial.

```
delete from asignaturas where codigo='DGBD';
```

```
Cannot delete or update a parent row: a foreign key constraint
fails (`xxx`.`imparte`, CONSTRAINT `imparte_ibfk_2` FOREIGN KEY
(`asignatura`) REFERENCES `asignaturas` (`codigo`))
```

Lo que sí podemos hacer es borrar primero esa fila de IMPARTE y después la asignatura:

Eliminando las referencias que nos impiden el borrado:

```
-- eliminamos las referencias en imparte
delete from imparte where asignatura='DGBD';

-- ahora sí puedo borrar la asignatura DGBD
-- porque ya no hay ninguna clave ajena que le esté haciendo
referencia
delete from asignaturas where codigo='DGBD';
```

## IR y update

UPDATE sólo generará problemas de integridad referencial si el dato a modificar es un valor de clave primaria que está siendo referenciada por alguna clave ajena.

Vamos primero a poblar la tabla IMPARTE:

```
insert into imparte (dni, asignatura) values ('21111222', 'FBD');
insert into imparte (dni, asignatura) values ('21111222', 'DGBD');
insert into imparte (dni, asignatura) values ('21333444', 'PC');

select * from imparte;
```

dni	asignatura
21111222	DGBD
21111222	FBD
21333444	PC

La siguiente orden **generaría un problema de integridad referencial**.

```
update asignaturas set codigo = 'BD1' where codigo = 'FBD';

-- ¡ERROR!
```

Si pudiéramos cambiar el identificador 'FBD' por 'BD1', en IMPARTE seguiría la información de que '21111222' imparte 'FBD', asignatura que ya no existiría. Por eso el sistema no nos deja, en principio, hacer directamente esta operación. Para poder realizarla necesitamos 3 pasos consistentes en

1. en ASIGNATURAS, insertar una nueva fila en asignaturas con el identificador BD1 y copiando el resto de los valores
2. en IMPARTE, cambiar las referencias a FBD por BD1
3. en ASIGNATURAS, eliminar la fila de FBD



**-- 1. nueva asignatura con los datos de la antigua**

```
insert
into asignaturas (codigo,descripcion,creditos,creditosp,coordin
values ('BD1', 'FUNDAMENTOS DE LAS BASES DE DATOS', 6.0, 1.5,
'21111222');
```

```
select * from asignaturas;
```

codigo	descripcion	creditos	creditosp	coordina
DGBD	DISEÑO Y GESTION DE BASES DE DATOS	6.0	3.0	21111222
FBD	FUNDAMENTOS DE LAS BASES DE DATOS	6.0	1.5	21111222
FP	FUNDAMENTOS DE LA PROGRAMACION	9.0	4.5	null
HI	HISTORIA DE LA INFORMATICA	4.5	null	null
PC	PROGRAMACION CONCURRENTES	6.0	1.5	null
<b>BD1</b>	<b>FUNDAMENTOS DE LAS BASES DE DATOS</b>	<b>6.0</b>	<b>1.5</b>	<b>21111222</b>

**-- 2. modificación de imparte**

```
update imparte set asignatura = 'BD1' where asignatura =
'FBD';
```

```
select * from imparte;
```

dni	asignatura
21111222	DGBD
21111222	<b>BD1</b>
21333444	PC

**-- 3. borrado de la asignatura antigua**

```
delete from asignaturas where codigo = 'FBD';
```

```
select * from asignaturas;
```

codigo	descripcion	creditos	creditosp	coordina
DGBD	DISEÑO Y GESTION DE BASES DE DATOS	6.0	3.0	21111222
FP	FUNDAMENTOS DE LA PROGRAMACION	9.0	4.5	null
HI	HISTORIA DE LA INFORMATICA	4.5	null	null
PC	PROGRAMACION CONCURRENTES	6.0	1.5	null
<b>BD1</b>	<b>FUNDAMENTOS DE LAS BASES DE DATOS</b>	<b>6.0</b>	<b>1.5</b>	<b>21111222</b>

## Modificando una definición de clave ajena

alter table.....

## Expresiones de selección de filas

### Rangos

Expresiones del tipo  $10 \leq x \leq 100$  se pueden construir utilizando el operador de construcción de rangos BETWEEN. La sintaxis de tal subexpresión de la cláusula where es la siguiente:

expresión **[NOT] BETWEEN** expresión **AND** expresión

Créditos y descripción de las asignaturas cuyo número de créditos está entre 5 y 8.

```
select credits, descripcion
from asignaturas
where credits between 5 and 8
```

credits	descripcion
6.0	DISEÑO Y GESTION DE BASES DE DATOS
6.0	FUNDAMENTOS DE LAS BASES DE DATOS
6.0	PROGRAMACION CONCURRENTES

## Subcadenas de caracteres

Podemos preguntar por subcadenas dentro de columnas de tipo carácter. Para ello utilizaremos los operadores LIKE (o MATCHES), que soportan la siguiente sintaxis:

expresión **[NOT] LIKE** 'cadena'

La cadena de caracteres cadena admite los comodines % (tanto por ciento) y \_ (subrayado):

- % indica una cadena de caracteres de cualquier longitud (Ali% = Alicante, Aligerar, Ali, ...)
- \_ un carácter cualquiera (Ali\_ = Alic, Alig, Ali, ...)

Profesores que atiendan al nombre de 'RAFA'.

```
select * from profesores where nombre like 'RAFA%'
```

dni	nombre	categoria	ingreso
21333444	RAFAEL ROMERO	ASO6	1992-06-16

Código de las asignaturas de 'Bases de Datos'

```
select codigo from asignaturas where descripcion like '%BASES DE DATOS%'
```

codigo
DGBD
FBD

Código de las asignaturas, siendo tal código de 2 caracteres

```
select codigo from asignaturas where codigo like '__'
```

codigo
FP
HI
PC

Descripción de las asignaturas cuya última palabra contiene 'INFORMATICA' y un carácter adicional.

```
select descripcion from asignaturas where descripcion like '%INFORMATICA_'
```

descripcion
HISTORIA DE LA INFORMATICA

## Información adicional

En esta [página](#) puedes encontrar contenidos adicionales que ampliarán algunos de los conceptos mostrados en esta lección.



FBDdocs por [BDgite](#) se encuentra bajo una [Licencia Creative Commons Atribución-CompartirIgual 3.0 Unported](#). Basada en una obra en <http://fbddocs.dlsi.ua.es>. Permisos que vayan más allá de lo cubierto por esta licencia pueden encontrarse en <http://fbddocs.dlsi.ua.es/autores>.

[BDgite \(GITE-11014-UA\)](#),  
Departamento de Lenguajes y Sistemas Informáticos,  
Universidad de Alicante

[Iniciar sesión](#) | [Informar de uso inadecuado](#) | [Imprimir página](#) | Con la tecnología de [Google Sites](#)