



REGLAS DE INTEGRIDAD DE DATOS

Jesús Reyes Carvajal

La integridad de los datos pueden ser de dos tipos:

- 1) Las restricciones de integridad de usuario son condiciones específicas de una base de datos concreta; es decir, son las que se deben cumplir en una base de datos particular con unos usuarios concretos, pero que no son necesariamente relevantes en otra base de datos. (Ej. validar que los sueldos de una relación EMPLEADOS no pueden ser negativos y por tanto se debe definir una condición)
- 2) Las reglas de integridad de modelo, en cambio, son condiciones más generales, propias de un modelo de datos, y se deben cumplir en toda base de datos que siga dicho modelo.

Regla de integridad de unicidad de la clave primaria

La regla de integridad de unicidad está relacionada con la definición de clave primaria. Concretamente, establece que toda clave primaria que se elija para una relación no debe tener valores repetidos.

DESPACHOS		
<u>edificio</u>	<u>número</u>	<i>superficie</i>
Marina	120	10
Marina	122	15
Marina	230	20
Diagonal	120	10

En esta relación, dado que la clave primaria está formada por *edificio* y *número*, *no hay ningún* despacho que repita tanto *edificio* como *número* de otro despacho.

Regla de integridad de entidad de la clave primaria

La regla de integridad de entidad de la clave primaria dispone que los atributos de la clave primaria de una relación no pueden tener valores nulos.

DESPACHOS		
<u>edificio</u>	<u>número</u>	superficie
Marina	120	10
Marina	122	15
Marina	230	20
Diagonal	120	10

En esta relación, puesto que la clave primaria está formada por *edificio* y *número*, *no hay ningún* despacho que tenga un valor nulo para *edificio*, *ni tampoco para número*.

Regla de integridad referencial

La regla de integridad referencial está relacionada con el concepto de *clave foránea*. Concretamente, determina que todos los valores que toma una clave foránea deben ser valores nulos o valores que existen en la clave primaria que referencia.

- Relación *DESPACHOS*:

DESPACHOS		
<u>edificio</u>	<u>número</u>	<i>superficie</i>
Marina	120	10
Marina	122	15
Marina	230	20
Diagonal	120	10

- Relación *EMPLEADOS*:

Donde *edificiodesp* y *númerodesp* de la relación *EMPLEADOS* forman una clave foránea que referencia la relación *DESPACHOS*.


EMPLEADOS				
<u>DNI</u>	<i>nombre</i>	<i>apellido</i>	<i>edificiodesp</i>	<i>númerodesp</i>
40.444.255	Juan	García	Marina	120
33.567.711	Marta	Roca	Marina	120
55.898.425	Carlos	Buendía	Diagonal	120
77.232.144	Elena	Pla	NULO	NULO

REFERENCIAS INCORRECTAS

Supongamos que en el ejemplo anterior hubiese un empleado con los valores <56.666.789, Pedro, López, Valencia, 325>.

No hay un despacho con los valores **Valencia y 325** para *edificio y número*, *la tupla de este empleado hace una referencia incorrecta; es decir, indica un despacho para el empleado que, de hecho, no existe.*

La regla de integridad referencial **establece** que *los valores de las llaves foráneas pueden ser valores nulos, o bien valores que coinciden con los valores de las llaves primarias de las entidades referenciadas.*



Un SGBD relacional tendrá que hacer cumplir esta regla de integridad y por tanto deberá efectuar comprobaciones cuando se produzcan las siguientes operaciones:

- a) Inserciones en una relación que tenga una clave foránea.
- b) Modificaciones que afecten a atributos que pertenecen a la clave foránea de una relación.
- c) Borrados en relaciones **referenciadas** por otras relaciones.
- d) Modificaciones que afecten a atributos que pertenecen a la clave primaria de una relación **referenciada** por otra relación.

En el ejemplo anterior, donde *edificiodesp* y *númerodesp* de la relación *EMPLEADOS* forman una clave foránea que referencia la relación *DESPACHOS*:

Las siguientes operaciones provocarían el incumplimiento de la regla de integridad referencial:

- Inserción de <12.764.411, Jorge, Puig, **Diagonal, 220**> en *EMPLEADOS*.
- Modificación de <40.444.255, Juan, García, Marina, 120> de *EMPLEADOS* por <40.444.255, Juan, García, **Marina, 400**>.
- **Borrado** de <**Marina, 120**, 10> de *DESPACHOS*.
- **Modificación** de <Diagonal, 120, 10> de *DESPACHOS* por <**París, 120**, 10>.

Una forma habitual de mantener estas reglas consiste en rechazar toda operación de actualización que deje la base de datos en un estado en el que alguna regla no se cumpla. En algunos casos, sin embargo, el SGBD tiene la posibilidad de aceptar la operación y efectuar acciones adicionales compensatorias, de modo que el estado que se obtenga satisfaga las reglas de integridad, a pesar de haber ejecutado la operación.



Esta última política se puede aplicar a las siguientes operaciones de actualización que violarían la regla de integridad:

- a) **Borrado** de una tupla que tiene una clave primaria referenciada.
- b) **Modificación** de los valores de los atributos de la clave primaria de una tupla que tiene una clave primaria referenciada.

En los casos anteriores, algunas de las políticas que se pueden aplicar serán las siguientes:

Restricción, actualización en cascada y anulación.

Regla de integridad referencial

Restricción

La política de restricción consiste en no aceptar la operación de actualización (**eliminación y modificación**).

Consiste en no permitir borrar una tupla si tiene una clave primaria referenciada por alguna clave foránea.

- Relación *CLIENTES*:

CLIENTES	
<u>numcliente</u>	...
10	-
15	-
18	-

- Relación *PEDIDOS_PENDIENTES*

PEDIDOS_PENDIENTES		
<u>numped</u>	...	numcliente*
1.234	-	10
1.235	-	10
1.236	-	15

* {numcliente} referencia *CLIENTES*.

Si aplicamos la restricción en caso de borrado y, por ejemplo, queremos borrar al cliente número 10, no podremos hacerlo porque tiene pedidos pendientes que lo referencian.

Regla de integridad referencial

Actualización en cascada

La política de actualización en cascada consiste en permitir la operación de actualización de la tupla, y en efectuar operaciones compensatorias que propaguen en cascada la actualización a las tuplas que la referenciaban; se actúa de este modo para mantener la integridad referencial.

- Relación *EDIFICIOS*:

EDIFICIOS	
<u>nombreedificio</u>	...
Marina	—
Diagonal	—

- Relación *DESPACHOS*:

DESPACHOS		
<u>edificio*</u>	<u>número</u>	<u>superficie</u>
Marina	120	10
Marina	122	15
Marina	230	20
Diagonal	120	10

* {edificio} referencia *EDIFICIOS*.

Si aplicamos la actualización en cascada en caso de **borrado** y, por ejemplo, queremos borrar el edificio Diagonal, se borrará también en Despachos Diagonal 120.

- Relación *EDIFICIOS*:

EDIFICIOS	
<u>nombreedificio</u>	...
Marina	—

- Relación *DESPACHOS*:

DESPACHOS		
<u>edificio*</u>	<u>número</u>	<u>superficie</u>
Marina	120	10
Marina	122	15
Marina	230	20

* {edificio} referencia *EDIFICIOS*.

b) Si aplicamos la actualización en cascada en caso de **modificación**, y queremos modificar el nombre del edificio Marina por Mar, también se cambiará Marina por Mar en los despachos Marina 120, Marina 122 y Marina 230, nos quedaría así:

- Relación *EDIFICIOS*:

EDIFICIOS	
<u>nombreedificio</u>	...
Mar	–

- Relación *DESPACHOS*:

DESPACHOS		
<u>edificio*</u>	<u>número</u>	<i>superficie</i>
Mar	120	10
Mar	122	15
Mar	230	20

* {edificio} referencia *EDIFICIOS*.

Regla de integridad referencial

Anulación

Esta política consiste en permitir la operación de actualización de la tupla y en efectuar operaciones compensatorias que pongan valores nulos a los atributos de la clave foránea de las tuplas que la referencian; esta acción se lleva a cabo para mantener la integridad referencial.

- Relación *VENDEDORES*:

VENDEDORES	
<u>numvendedor</u>	...
1	-
2	-
3	-

- Relación *CLIENTES*:

CLIENTES		
<u>numcliente</u>	...	vendedorasig*
23	-	1
35	-	1
38	-	2
42	-	2
50	-	3

* {vendedorasig} referencia *VENDEDORES*.

a) Si aplicamos la anulación en caso de **borrado** y, por ejemplo, queremos borrar al vendedor número 1, se modificarán todos los clientes que lo tenían asignado, y pasarán a tener un valor nulo en *vendedorasig*.

- Relación *VENDEDORES*:

VENDEDORES	
<u>numvendedor</u>	...
2	–
3	–

- Relación *CLIENTES*:

CLIENTES		
<u>numcliente</u>	...	<i>vendedorasig</i> *
23	–	NULO
35	–	NULO
38	–	2
42	–	2
50	–	3

* {*vendedorasig*} referencia *VENDEDORES*.

b) Si aplicamos la anulación en caso de **modificación**, y ahora queremos cambiar el número del vendedor 2 por 5, se modificarán todos los clientes que lo tenían asignado y pasarán a tener un valor nulo en *vendedorasig*. Nos quedaría:

- Relación *VENDEDORES*:

VENDEDORES	
<u>numvendedor</u>	...
5	–
3	–

- Relación *CLIENTES*:

CLIENTES		
<u>numcliente</u>	...	<i>vendedorasig</i> *
23	–	NULO
35	–	NULO
38	–	NULO
42	–	NULO
50	–	3

* {*vendedorasig*} referencia *VENDEDORES*.



Selección de la política de mantenimiento de la integridad referencial

Hemos visto que en caso de borrado o modificación de una clave primaria referenciada por alguna clave foránea hay varias políticas de mantenimiento de la regla de integridad referencial.

El diseñador puede elegir para cada clave foránea qué política se aplicará en caso de borrado de la clave primaria referenciada, y cuál en caso de modificación de ésta.

El diseñador deberá tener en cuenta el significado de cada clave foránea concreta para poder elegir adecuadamente.

SINTAXIS DE REGLAS DE INTEGRIDAD

```
FOREIGN KEY ( column [, ... ] )  
REFERENCES reftable [ ( refcolumn [, ... ] ) ]  
[ MATCH matchtype ]  
[ ON DELETE action ]  
[ ON UPDATE action ]
```

Hay tres parámetros cuando definimos una clave foránea que son muy importantes y que definen como la base de datos se va a comportar para salvaguardar la integridad de nuestros datos. Estos parámetros son:

MATCH *tipo*
ON DELETE *acción*
ON UPDATE *acción*

En donde *tipo* puede tener estos valores:

FULL: No permite que una columna tenga el valor NULL en una clave foránea compuesta por varias columnas

SIMPLE: Permite que una columna tenga el valor NULL en una clave foránea compuesta por varias columnas



acción puede tener estos valores:

NO ACTION: Produce un error indicando que un DELETE ó UPDATE creará una violación de la clave foránea definida.

RESTRICT: Produce un error indicando que un DELETE ó UPDATE creará una violación de la clave foránea definida.

CASCADE: Borra ó actualiza automáticamente todas las referencias activas

SET NULL: Define las referencias activas como NULL

SET DEFAULT: Define las referencias activas como el valor por defecto (si está definido) de las mismas

Ejemplo

FACTURAS	
FAC.NUM	CLIENTE
00001	1
00002	1
00003	2
00004	3

CLIENTES	
ID	NOMBRE
1	Nombre1
2	Nombre2

```
ALTER TABLE facturas ADD CONSTRAINT clientes_id_fk  
FOREIGN KEY (cliente) REFERENCES clientes(id)  
MATCH FULL ON DELETE RESTRICT ON UPDATE CASCADE;
```



```
postgres=# INSERT INTO facturas (facnum,cliente) VALUES  
( '0004',3);
```

```
postgres=# DELETE FROM clientes WHERE id = 1;
```

```
postgres=# UPDATE clientes SET id = 3 where id = 1;
```



CUADRO RESUMEN DE REGLAS

ON UPDATE { NO ACTION | CASCADE | SET NULL | SET DEFAULT }

ON DELETE { NO ACTION | CASCADE | SET NULL | SET DEFAULT }