

**Universidad Nacional Autónoma de México**  
**Facultad de Ciencias, 2023-I**  
**Fundamentos de Bases de Datos**

Práctica 07:  
Mantenimiento de Llaves Foráneas.

Alumnos:

**Galeana Vidaurri Rodrigo**  
**Garcia Arce Marco Antonio**  
**Pérez Romero Natalia Abigail**  
**Rosales Jaimes Victor**  
**Rojas Jarillo Mauricio**

Profesor:

**Gerardo Avilés Rosas**

Ayudantes de teoría:

**Gerardo Uriel Soto Miranda**  
**Tania Naomi Barajas Pulido**

Ayudantes de laboratorio:

**Ricardo Badillo Macías**  
**Carlos Augusto Escalona Navarro**

# 1. Preguntas de investigación

## 1. ¿Qué es una política de mantenimiento de llaves foráneas?

Una llave foránea (FK) es una columna o combinación de columnas que se usa para establecer y aplicar un vínculo entre los datos de dos tablas a fin de controlar los datos que puede almacenar una tabla de llave foránea. Se crea un vínculo entre dos tablas cuando las columnas de una de ellas hacen referencia a las columnas de la otra que contienen el valor de llave principal. Esta columna se convierte en una llave foránea para la segunda tabla.

Para ejemplificar el concepto vamos a considerar las siguientes dos tablas:

Libro	
PK	<u>Código</u>
FK	codigoEditorial
	Título
	Autor

Editorial	
PK	<u>Código</u>
	Nombre

Podemos notar que la llave foránea de la tabla Libro hace referencia a la tabla Editorial por medio de codigoEditorial. Ahora supongamos que queremos actualizar el código de una editorial porque en un inicio se escribió mal, pero hay un inconveniente, la editorial con el código erróneo ya se relacionaba con varias instancias de Libro, pero ¿qué pasa cuando actualizamos el código?, pues todas las instancias de Libro que se relacionaban con esta editorial pero con el código erróneo quedarán huérfanas y romperán la integridad referencial, lo mismo sucedería si quisiéramos eliminar una editorial. Nuestras bases de datos siempre deberán cumplir con la integridad referencial, para que no nos suceda entran en juego las políticas de mantenimiento de las llaves foráneas que son las acciones que tomamos para preservar siempre la integridad referencial. A continuación mostramos 3 políticas de mantenimiento de llaves foráneas:

### ■ Política por omisión

Esta política nos permite mantener la integridad referencial al actualizar y eliminar pero para ello debemos seguir ciertos pasos:

#### ● Eliminar:

Supongamos que tenemos dos tablas A y B, la llave de A tiene como llave primaria a  $PK_A$  y la tabla B tiene como llave primaria a  $PK_B$ , además  $PK_A$  es llave foránea de B. Si deseamos eliminar un renglón de B se permite hacerlo y funciona de manera correcta, pero si queremos eliminar un elemento de A (digamos que dicho elemento se identifica por  $\alpha$ ) tenemos que hacer lo siguiente:

1. Borrar los renglones de la tabla B donde  $\alpha$  figure como llave foránea
2. Borrar el elemento de la tabla A

Si no se hace de ese modo postgresql no dejará que el eliminado se efectúe.

#### ● Actualización:

No es posible hacerlo bajo esta política.

### ■ Política de establecimiento de nulos

La idea fundamental es establecer como nulos los valores que se eliminan o actualizan en donde figuran en otra tabla como llaves foráneas.

### ■ Política en Cascada

Supongamos que tenemos dos tablas A y B, la llave de A tiene como llave primaria a  $PK_A$  y la tabla B tiene como llave primaria a  $PK_B$ , además  $PK_A$  es llave foránea de B. Si deseamos eliminar un renglón de B se permite hacerlo y funciona de manera correcta, pero si queremos operar un elemento de A (digamos que dicho elemento se identifica por  $\alpha$ ) hacemos:

- Si lo queremos eliminar, este se elimina y elimina también a los elementos en la tabla B en los que  $\alpha$  figure como llave foránea.
- Si lo queremos actualizar, este se elimina y actualiza también a los elementos en la tabla B en los que  $\alpha$  figure como llave foránea.

## 2. Para cada política que investigaron, ¿cómo se indica en SQL? Supongamos que tenemos dos tablas A y B, la llave de A tiene como llave primaria a $PK_A$ y la tabla B tiene como llave primaria a $PK_B$ , además $PK_A$ es llave foránea de B.

- **Política por omisión:** queremos eliminar el elemento identificado por **ident** de la tabla A

```

1 -- Eliminamos primero los renglones de la tabla B donde indent sea el valor
2 -- de su llave foranea
3 delete from B where FK_B = indent;
4
5 -- Eliminamos el renglon de la tabla A donde indent sea la PK_A
6 delete from A where PK_A = indent;

```

- **Política por establecimiento de nulos:** queremos eliminar el elemento identificado por **indent** de la tabla A

```

1 -- Eliminar
2 alter table B add constraint FK_B foreign key(FK_A) references A(FK_A)
3 on delete set null;
4 -- Actualizar
5 alter table B add constraint FK_B foreign key(FK_A) references A(FK_A)
6 on update set null;

```

- **Política de Cascada**

```

1 -- Eliminar
2 alter table B add constraint FK_B foreign key(FK_A) references A(PK_A)
3 on delete cascade;
4 -- Actualizar
5 alter table B add constraint FK_B foreign key(FK_A) references A(PK_A)
6 on update cascade;

```

3. Para cada política que investigaron, ¿cuál es su objeto y su funcionamiento?

Supongamos que tenemos dos tablas A y B, la llave de A tiene como llave primaria a  $PK_A$  y la tabla B tiene como llave primaria a  $PK_B$ , además  $PK_A$  es llave foránea de B. Supongamos además que queremos operar un elemento de A (digamos que dicho elemento se identifica por  $\alpha$ ):

- **Política por omisión:**

Esta política solo puede eliminar, para esto borrar un renglón en A primero todos los renglones en la tabla B que tengan como llave foránea a  $\alpha$  y luego eliminamos el renglón de A identificado por  $\alpha$ . Si queremos eliminar un elemento de B lo hacemos directamente.

- **Política por establecimiento de nulos**

El procedimiento para eliminar y actualizar es técnicamente el mismo, lo que hace es cambiar el valor de llave foránea en B que sea igual a  $\alpha$  por null y luego elimina/actualiza el renglón identificado por  $\alpha$ . Esto en una sola instrucción.

- **Política por cascada**

El procedimiento para eliminar y actualizar es técnicamente el mismo, lo que hace es actualizar/eliminar los renglones en la tabla B donde  $\alpha$  figura como llave foránea y luego elimina/actualiza el renglón de A identificado por  $\alpha$ .

4. Para cada política que investigaron, ¿cuáles son sus ventajas y desventajas?

- **Política por omisión:**

Esta política cumple con ser consistente una vez terminado todo el proceso, además de que elimina lo que le indiquemos que elimine, el inconveniente más grande es que esta política no nos permite actualizar de ninguna forma.

- **Política por establecimiento de nulos**

Esta política cumple con ser consistente y permite tanto eliminar como actualizar, pero genera valores nulos los cuales en baja escala son manejables pero en una escala lo suficientemente grande genera basura, aunque el tener valores nulos no es malo, esto depende de los intereses o intenciones que se tengan.

- **Política por cascada**

Cumple con ser consistente, en algunos casos, dependiendo de los intereses y manejo de la tabla puede que se pierda información pero en general es la política más usada.

5. Con base a lo anterior, ¿cuál política utilizarán para su esquema, y por qué motivo?

Usaremos la política cascada debido a que consideramos que es la mejor manera de mantener la consistencia en nuestra base de datos, además de que pretendemos usar en mayor parte la actualización y la manera en la que actualiza esta política es la más eficiente y que no genera ruido en las demás tablas, además evitamos los valores nulos que en nuestro caso no son deseados.