

# Universidad Nacional Autónoma de México

Facultad de Ciencias

---

## Fundamentos de Bases de Datos

### Práctica 06. Mantenimiento de llaves foráneas

---

Equipo : eSosQLones

Estrada Garcia Luis Gerardo	319013832
Jiménez Hernández Allan	420003478
Mancera Quiroz Javier Alejandro	319274831
Mora Hernández Dulce Julieta	319236448
Peña Nuñez Axel Yael	318279754

## Actividades

Deberán realizar un documento PDF con el nombre Práctica06. Donde contesten las siguientes preguntas.

Recuerden que la investigación debe ser desde el punto de vista de PostgreSQL.

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

Una política de mantenimiento de llaves foráneas es un conjunto de reglas y procedimientos que permiten tener un control de las operaciones entre las tablas que se relacionan a través de llaves foráneas, con el fin de garantizar la integridad de las relaciones entre tablas durante la actualización o eliminación de datos.

Entre las políticas suelen incluir lo que sigue

- Creación y definición de llaves foráneas
- Actualización y eliminación de registros relacionados
- Control de referencias huérfanas
- Mantenimiento y auditoría
- Documentación y capacitación

2. Para cada política que investigaron, ¿cómo se indica en SQL?

- Creación y definición de llaves foráneas `alter table NOMBRETABLA1 add constraint NOMBREERESTRICCION foreign key (CAMPOCLAVEFORANEA) references NOMBRETABLA2 (CAMPOCLAVEPRIMARIA);`

Además de lo anterior, contamos con algunas restricciones para las FK, por ejemplo, no podemos insertar un elemento en una tabla que contiene una FK a menos que el valor al que se haga referencia exista en la tabla a la que se hace referencia(dicha restricción es automática).

Para eliminar utilizamos `ON DELETE` y para actualizar `ON UPDATE`, seguido de alguno de los modificadores:

(a) `RESTRICT`

`RESTRICT` no nos permite eliminar o modificar filas en la tabla referenciada si existen filas con el mismo valor de clave foránea.

`FOREIGN KEY (columna) REFERENCES tablareferenciada (columnareferenciada) ON DELETE RESTRICT;`

(b) `CASCADE`

Se utiliza para borrar o modificar una clave en una fila en la tabla referenciada con un valor determinado de clave, `CASCADE` implica borrar

las filas con el mismo valor de FK o modificar los valores de esas FK.  
FOREIGN KEY (columna) REFERENCES tablareferenciada (columnaref-  
erenciada) ON DELETE CASCADE;

(c) SET NULL

Borra o modifica una clave en una fila en la tabla referenciada con un valor determinado de clave.

FOREIGN KEY (columna) REFERENCES tablareferenciada (columnaref-  
erenciada) ON DELETE SET NULL;

(d) NO ACTION

NO ACTION permite que las FK no se modifiquen ni se eliminen filas en la tabla que las contiene.

FOREIGN KEY (columna) REFERENCES tablareferenciada (columnaref-  
erenciada) ON DELETE NO ACTION;

(e) SET DEFAULT

Este comando permite borrar o modificar una clave en una fila en la tabla referenciada con un valor determinado.

FOREIGN KEY (columna) REFERENCES tablareferenciada (columnaref-  
erenciada) ON DELETE SET DEFAULT;

- Actualización y eliminación de registros relacionados

Para eliminar los registros de una tabla usamos el comando:

ALTER TABLE nombretabla DROP FOREIGN KEY nombrestreccion

- Control de referencias huérfanas

En este caso, podemos definir restricciones por medio de MATCH FULL, MATCH SIMPLE o MATCH PARTIAL.

Por ejemplo

FOREIGN KEY (a,b) REFERENCES foo MATCH FULL

- Mantenimiento y auditoría

El mantenimiento y la auditoría de políticas de FK generalmente se realizan mediante la realización de consultas de manera periódicas para verificar que no existan referencias incorrectas.

- Documentación y capacitación

Para agregar comentarios podemos realizarlo de dos maneras

(a) (–) al comienzo de la línea para comentarios de una línea

```
select * from libros – mostramos los registros de libros;
```

- (b) Para comentarios de varias líneas colocamos `/*` al comienzo del bloque de comentario y al finalizarlo `*/`.

```
select titulo, autor  
/*mostramos títulos y  
nombres de los autores*/  
from libros;
```

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

- Creación y definición de llaves foráneas

Esta política nos permite establecer una relación entre dos tablas en una base de datos donde la FK permite hacer referencia a la PK permitiendo la relación entre registros de las tablas.

La creación y definición de FK primero debemos definirla en la tabla donde queremos establecer una relación especificando la columna que será nuestra FK así como la tabla a la que hace referencia.

Una vez realizado lo anterior, cuando se inserte o actualice un registro en la tabla, PostgreSQL verifica si existe un registro correspondiente en la tabla.

- Actualización y eliminación de registros relacionados

Este punto nos permite saber como tratar los registros relacionados cuando se actualiza o elimina un registro en la tabla principal. Además, nos garantiza la consistencia en nuestros datos.

Para utilizarlas podemos especificar el comportamiento deseado por medio de `ON UPDATE` y `ON DELETE` cuando definimos a la FK.

- Control de referencias huérfanas

Esta política busca evitar tener referencias huérfanas (registros en tablas secundarias con referencias a registros inexistentes en la principal).

Para controlarlas, podemos utilizar `MATCH FULL` (que garantiza que todas las columnas de la FK coincidan con las de la PK), `MATCH SIMPLE` o `MATCH PARTIAL` al definir la FK.

- Mantenimiento y auditoría

Esta política garantiza la integridad referencial a lo largo del tiempo, permitiéndonos detectar problemas y así, corregirlos.

El mantenimiento implica realizar comprobaciones periódicas de integridad referencial mediante consultas SQL con el fin de detectar problemas, para que en caso de encontrarlos, solucionarlos, mientras que la auditoría implica revisar regularmente las políticas y procedimientos para asegurarse de que sigan siendo efectivos y en caso de no serlo, documentar los cambios.

- Documentación y capacitación

La documentación y la capacitación nos ayudan a comprender las políticas de las FK.

La documentación registra las políticas y procedimientos en un lugar accesible para que los usuarios y administradores de la base de datos los consulten. Por su parte, la capacitación nos sirve para informar sobre cómo aplicar y seguir estas políticas para mantener la integridad de los datos.

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

- RESTRICT

- Ventajas: RESTRICT se verifica al final de la declaración, en lugar de al final de la transacción.
- Desventajas: Previene la eliminación de una fila en la tabla referenciada.

- CASCADE

- Ventajas: Mantiene la integridad de los datos, ya que ayuda a prevenir situaciones donde un registro está referenciado a algo que ya no existe, es decir, que fue eliminado.
- Desventajas: Suele ser muy lento realizar eliminaciones o actualizaciones en cascada.

- SET NULL

- Ventajas: Las bases de datos automáticamente excluyen los valores NULL cuando usan funciones internas, lo que ahorra un poco de procesamiento.
- Desventajas: Puede que ocupe más espacio, al hacer las llaves foráneas null en la tabla referenciada, en lugar de eliminar las filas.

- NO ACTION

- Ventajas: Para eliminar una fila referenciada en la tabla principal, luego insertar una fila de reemplazo con la misma llave, antes de terminar la transacción. Usar NO ACTION permite hacer esto, sin usar UPDATE.

- Desventajas: Si cualquier fila en la tabla referenciada aún existe cuando se verifica la restricción, se lanza un error; lo que evita la eliminación o actualización en la tabla principal.

- SET DEFAULT

- Ventajas: No deja como null el atributo de la llave foránea en la tabla referenciada, sino que le pone el valor definido por default, esto ayuda a que si se tiene una restricción de NOT NULL, no se verifiquen esos casos, ya que no van a ser NULL, lo que ayuda a evitar errores.
- Desventajas: No se pueden asignar valores NULL, y si se elimina una fila de la tabla principal, la fila no se elimina en la tabla referenciada.

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

Usamos CASCADE, porque mantiene una mejor integridad en los datos al momento de hacer actualizaciones o eliminaciones, ya que lo hace para todas las tablas que están referenciadas y no solo a la tabla principal.