

## Preguntas

1. ¿Qué es una política de mantenimiento de llaves foráneas?
2. Para cada política que investigaron, ¿Cómo se indica en SQL?
3. Para cada política que investigaron, ¿Cuál es su objeto y su funcionamiento?
4. Para cada política que investigaron, ¿Cuáles son sus ventajas y desventajas?
5. Con base a lo anterior, ¿Cuál política utilizarán para su esquema, y por qué motivo?

## Respuestas

1. Son acciones que se realizan sobre las entidades hijas de un padre este se elimina o actualiza la llave primaria. Existen diferentes tipos de políticas de mantenimiento que pueden eliminar, prohibir, actualizar a un valor por omisión o no hacer nada dependiendo de como sean configuradas. Las políticas de mantenimiento se configuran por cada constraint en una tabla. En la practica rara vez se actualiza la llave primaria de una entidad (es preferible eliminar y crear de nuevo con la llave correcta) por lo que las políticas usualmente son aplicadas al momento de eliminar.
2. Se indican con `ON DELETE` y `ON UPDATE` seguido de esto se debe especificar la acción a realizar que puede ser cualquiera de las siguientes: `SET NULL`, `SET DEFAULT`, `RESTRICT`, `NO ACTION`, `CASCADE`
3.
  - (a) `SET NULL`: Pone la referencia a la tabla padre con el valor `NULL`
  - (b) `SET DEFAULT`: Pone la referencia a la tabla padre con el valor por omisión que se definió durante la creación de la tabla hija.
  - (c) `RESTRICT`: Impide que se elimine la tabla padre sin antes eliminar o cambiar las referencias en los hijos afectados.
  - (d) `NO ACTION`: No realiza ninguna acción.
  - (e) `CASCADE`: Elimina los valores de la tabla hija que tengan la referencia al padre.
4.
  - (a) `SET NULL`: **Ventajas**: Conserva toda la información, no altera registros. **Desventajas**: Imposible de implementar si la restricción de `NOT NULL` esta en efecto.
  - (b) `SET DEFAULT`: **Ventajas**: Conserva toda la información, los hijos se asignan a un valor conocido. **Desventajas**: Puede ocasionar efectos secundarios no deseados, se debe de definir un valor por omisión al momento de crear la tabla hija
  - (c) `RESTRICT`: **Ventajas**: Seguridad de que no se eliminan datos de manera accidental. **Desventajas**: Es necesario actualizar o eliminar todos los hijos a mano antes de hacer un cambio en el padre.
  - (d) `NO ACTION`: **Ventajas**: No hay que pensar en la implementación **Desventajas**: Produce efectos secundarios no deseados, restricciones en las columnas puede hacer que falle
  - (e) `CASCADE`: **Ventajas**: Es la implementación más común, el comportamiento es natural si se elimina un dato sus dependientes también deben de eliminarse **Desventajas**: Riesgo de accidentalmente eliminar multiples entradas a traves de distintas tablas, operación más costosa
5. Elegimos la implementación con `CASCADE` para todas las restricciones de llave foranea en las tablas. Esto es debido a que tenemos restricciones del tipo `NOT NULL`, no existen valores por omision definidos y `RESTRICT` solo causaria problemas al tratar de borrar algún dato en tablas como *Estetica* o *Dueño* ya que tienen demasiadas dependencias.

## Fuentes

- <https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-foreign-key/>