



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA

Tarea: Niveles de aislamiento

GRUPO:

01

ALUMNO :

Genis Cruz Lourdes Victoria

PROFESOR(A):

**Fernando Arreola
Franco**



Ciudad Universitaria, Cd. Mx., 2022

1. Transaction Isolation El SQL estándar define cuatro niveles de aislamiento de transacciones, dependiendo de si se permite o no que aparezcan los siguientes fenómenos: Lectura sucia (dirty read): una transacción lee datos modificados por otra transacción aunque ésta no haya realizado un COMMIT. Lectura No Repetible (nonrepeatable read): una transacción relee datos que ya leyó antes y encuentra que han sido modificados por otra transacción que hizo COMMIT después de la primera lectura. Lectura fantasma (phantom read): una transacción vuelve a ejecutar un select y encuentra que el resultado ha cambiado, debido a que otra transacción realizó modificaciones e hizo un COMMIT. Los distintos niveles con sus restricciones asociadas se presentan en la siguiente tabla:

Nivel de aislamiento	Lectura sucia	Lectura No Repetible	Fantasma
Read uncommitted	Possible	Possible	Possible
Read committed	Not possible	Possible	Possible
Repeatable read	Not possible	Not possible	Possible
Serializable	Not possible	Not possible	Not possible

Cuando ejecutamos una orden SQL, Postgres realiza un autocommit. Es decir, la orden es una transacción en sí misma. Si lo que queremos es englobar un conjunto de órdenes dentro de una transacción debemos iniciar la transacción tanto con BEGIN TRANSACTION como con START TRANSACTION, cuya funcionalidad es la misma y cuya sintaxis viene dada por:

```
BEGIN [ WORK | TRANSACTION ] [ transaction_mode [, ...] ]

START TRANSACTION [ transaction_mode [, ...] ]

where transaction_mode is one of:

ISOLATION LEVEL { SERIALIZABLE | REPEATABLE READ |
  READ COMMITTED | READ UNCOMMITTED }
READ WRITE | READ ONLY
```

También es posible utilizar la orden SET TRANSACTION para establecer el funcionamiento de la siguiente transacción, pero no afectará a las subsiguientes. Cuando se inicia una transacción definimos el tipo de acceso que vamos a realizar (READ WRITE o READ ONLY), así como el nivel de aislamiento que queremos. En el caso de Postgres los cuatro niveles se reducen a dos, para mantener coherencia con el sistema de concurrencia de tipo multiversión. Uno es Read committed (valor nivel por defecto) y el otro es Serializable. Cuando especificamos el nivel Read uncommitted asocia el Read committed; y cuando especificamos el nivel Repeatable read asocia el Serializable. Para finalizar una transacción podemos utilizar la orden COMMIT si queremos que quede reflejo

en la base de datos de nuestras operaciones. En caso de fracaso de la transacción o si no queremos que quede huella de las modificaciones realizadas por la transacción ejecutaremos un ROLLBACK.

En resumen...

El estándar SQL define **Cuatro niveles de aislamiento**. La más estricta es la serialización, que se define en el estándar con un párrafo completo. Se dice que la ejecución concurrente arbitraria de un grupo de transacciones serializarles está garantizada para ser efectiva como si fueran ejecutadas una a una en un orden determinado. Los otros tres niveles están definidos por los fenómenos generados por la interacción entre transacciones concurrentes, cada nivel requiere que no ocurra ningún fenómeno. Tenga en cuenta que debido a la definición de serializarle, ninguno de estos fenómenos puede suceder en este nivel (esto no es sorprendente, si el efecto de la transacción es el mismo que ejecutar solo uno a la vez, ¿cómo puede ver el fenómeno debido a la interacción?

Los fenómenos que están prohibidos en cada nivel son:

Lectura sucia

Una transacción lee los datos escritos por otra transacción paralela no confirmada.

No se puede leer repetidamente

Una transacción vuelve a leer los datos leídos anteriormente y encuentra que los datos han sido modificados por otra transacción (enviada después de la lectura inicial).

Lectura fantasma

Una transacción vuelve a ejecutar una consulta que devuelve un conjunto de filas que cumple una condición de búsqueda y descubre que el conjunto de filas que cumple la condición ha cambiado debido a otra transacción enviada recientemente.

Excepción de serialización

El resultado de confirmar con éxito un grupo de transacciones es inconsistente con todos los posibles resultados de ejecución en serie de estas transacciones.

Bibliografía:

1. *Niveles de aislamiento en Postgres [Online]*
<http://di002.edv.uniovi.es/~fcano/bdatos/practicas/isolationLevel/isolationPostgres.pdf>

2. Nivel de aislamiento de transacciones de PostgreSQL
<https://programmerclick.com/article/53931662974/>