

# Grados de aislamiento en SQL

*Hay distintos niveles de aislamiento que podemos parametrizar en SQL.*

SET TRANSACTION

ISOLATION LEVEL [ READ UNCOMMITTED |

READ COMMITTED | REPEATABLE | SERIALIZABLE ]

*Cada opción tiene su costo.*

# READ UNCOMMITTED

*Permite hacer lecturas sucias (dirty reads), donde las consultas dentro de una transacción son afectadas por cambios no confirmados (not committed) de otras transacciones.*

*Esta opción es apenas transaccional, es como no tener transacciones.*

# READ UNCOMMITTED (dirty reads)

T0	T1
<pre>SELECT COUNT(*) AS numerosas FROM personal WHERE hijos &gt; 3;       numerosas -----       15</pre> <pre>SELECT COUNT(*) AS numerosas FROM personal WHERE hijos &gt; 3;       numerosas -----       16</pre>	<pre>UPDATE personal SET hijos=4 WHERE legajo=32;</pre> <pre>ROLLBACK;</pre>

## READ COMMITTED (Non-repeatable reads)

*Los cambios confirmados son visibles dentro de otra transacción, esto significa que dos consultas dentro de una misma transacción pueden retornar diferentes resultados. Generalmente este es el comportamiento por defecto en los SGBD.*

# READ COMMITTED (Non-repeatable reads)

T0	T1
<pre>SELECT COUNT(*) AS numerosas FROM personal WHERE hijos &gt; 3;       numerosas -----       15</pre> <pre>SELECT COUNT(*) AS numerosas FROM personal WHERE hijos &gt; 3;       numerosas -----       15</pre>	<pre>UPDATE personal SET hijos=4 WHERE legajo=32;  COMMIT;</pre>

# READ COMMITTED (Non-repeatable reads)

T0	T1
<pre>SELECT COUNT(*) AS numerosas FROM personal WHERE hijos &gt; 3;       numerosas -----       15</pre> <pre>SELECT COUNT(*) AS numerosas FROM personal WHERE hijos &gt; 3;       numerosas -----       16</pre>	<pre>UPDATE personal SET hijos=4 WHERE legajo=32; COMMIT;</pre>

# REPEATEABLE READS (Phantom reads)

*Dentro de una transacción todas las lecturas son consistentes. En este nivel de aislamiento, el SGBD implementa el control de concurrencia basado en bloqueos, mantiene los bloqueos de lectura y escritura -de los datos seleccionados- hasta el final de la transacción. Sin embargo, no se gestionan los bloqueos de rango, por lo que las **lecturas fantasma** pueden ocurrir*

# REPEATABLE READS (Phantom reads)

T0

```
SELECT COUNT(*) AS numerosas  
FROM personal WHERE hijos > 3;  
numerosas
```

-----  
15

```
SELECT COUNT(*) AS numerosas  
FROM personal WHERE hijos > 3;  
numerosas
```

-----  
15

T1

```
UPDATE personal SET hijos=4  
WHERE legajo=32;  
COMMIT;
```

NO SE PUEDE REALIZAR  
PORQUE LOS DATOS  
ESTÁN BLOQUEADOS  
POR T0



# REPEATABLE READS (Phantom reads)

T0	T1
<pre>SELECT COUNT(*) AS numerosas FROM personal WHERE hijos &gt; 3;       numerosas -----       15</pre> <pre>SELECT COUNT(*) AS numerosas FROM personal WHERE hijos &gt; 3;       numerosas -----       16</pre>	<pre>INSERT INTO personal (legajo, hijos) VALUES (1302,4); COMMIT;</pre>

# SERIALIZABLE

*No se permiten actualizaciones en otras transacciones si una transacción ha realizado una consulta sobre ciertos datos. En este caso las distintas transacciones no se afectan entre sí.*

*Las transacciones están completamente aisladas entre sí, lo que conlleva un costo asociado.*

*Que puede hasta detener todos los procesos!!*

# SERIALIZABLE

T0

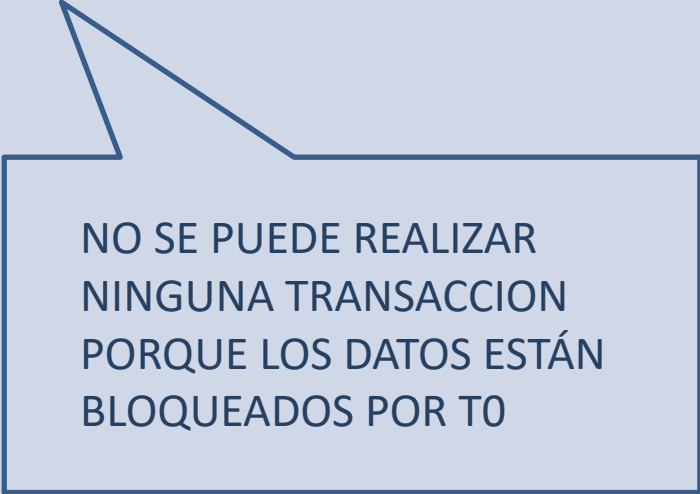
```
SELECT COUNT(*) AS numerosas  
FROM personal WHERE hijos > 3;  
numerosas
```

-----  
15

```
SELECT COUNT(*) AS numerosas  
FROM personal WHERE hijos > 3;  
numerosas
```

-----  
15

T1



NO SE PUEDE REALIZAR  
NINGUNA TRANSACCION  
PORQUE LOS DATOS ESTÁN  
BLOQUEADOS POR T0

# En resumen

Nivel de aislamiento	Lectura sucia	Lectura no repetible	Lectura fantasma
READ UNCOMMITTED	SI	SI	SI
READ COMMITTED	--	SI	SI
REPEATEABLE READ	--	--	SI
SERIALIZABLE	--	--	--