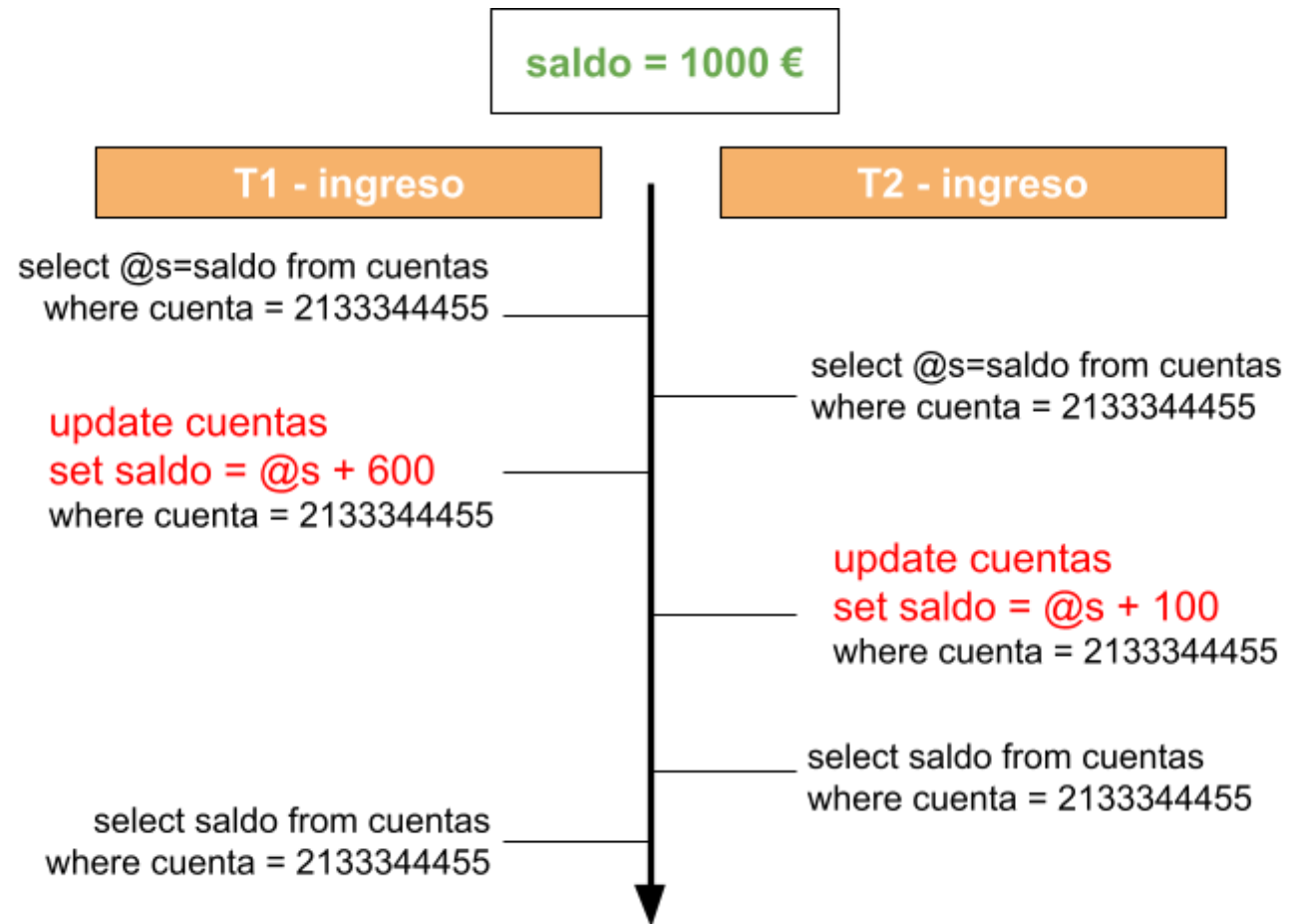


Sesión 02 - Transacciones

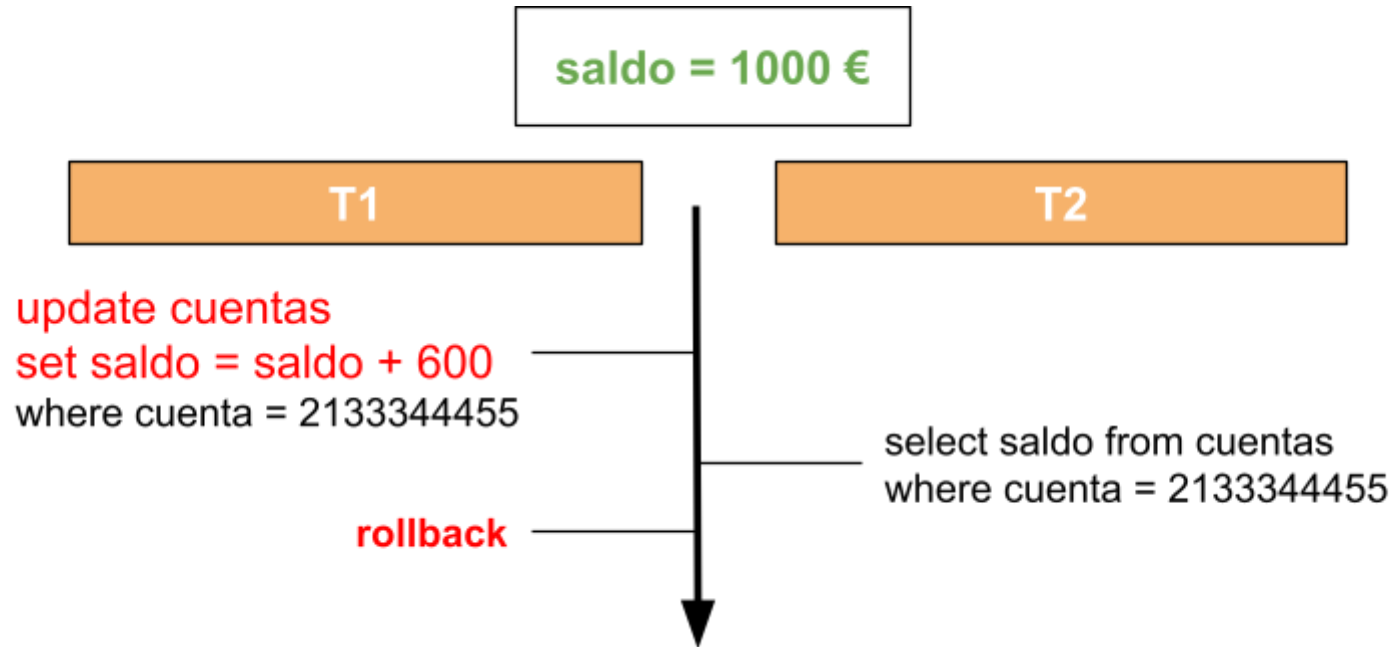
Qué es una transacción



Concurrencia

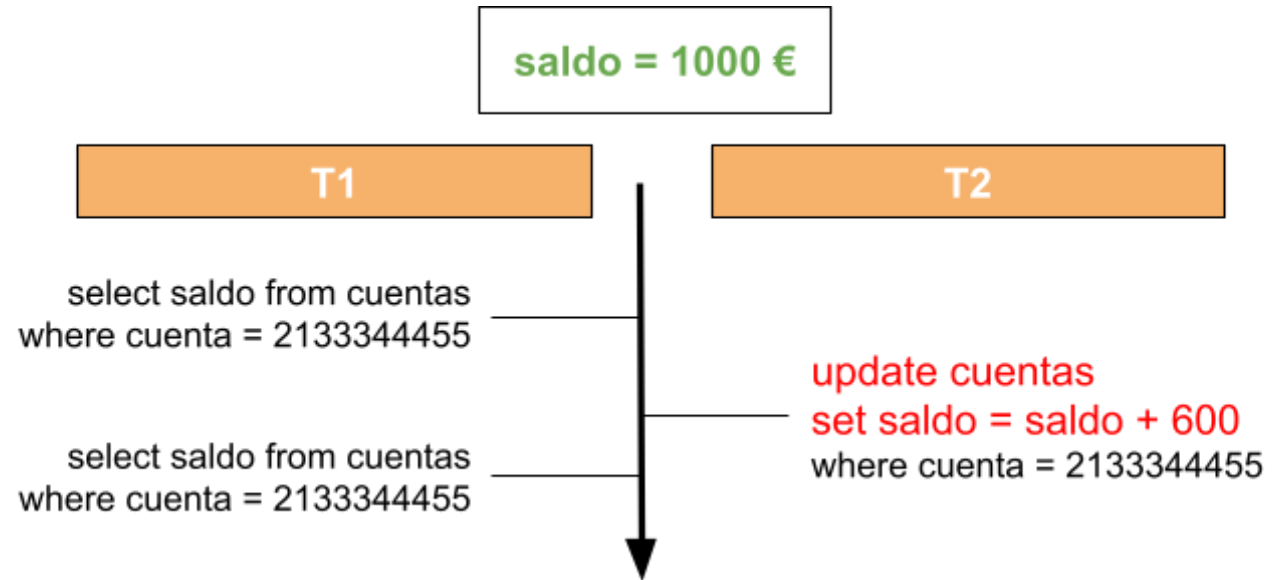
Lecturas sucias

Una transacción lee datos no confirmados por otra transacción.



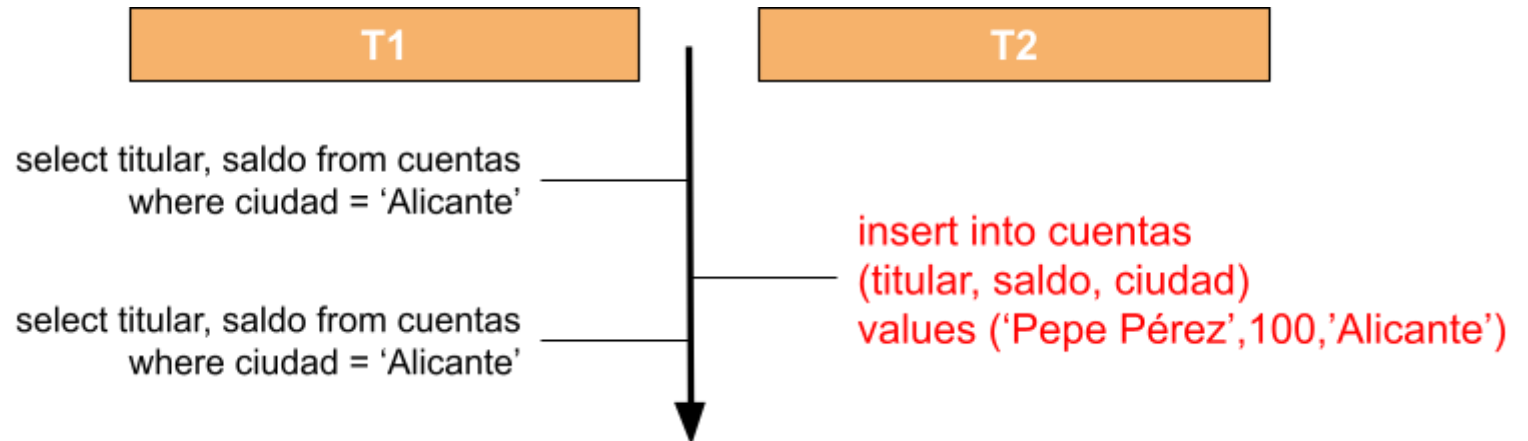
Lecturas no repetibles

Una transacción modifica datos leídos por otra.



Lecturas fantasma

Una transacción podría obtener diferentes conjuntos de datos con la misma consulta.



ACID: Atomicity Consistency Isolation Durability

Atomicidad Consistencia Aislamiento Durabilidad

Atomicidad

Fin de la transacción

- **COMMIT**: confirma las operaciones realizadas y se hacen persistentes
- **ROLLBACK**: "deshace" todas las operaciones, volvemos al estado anterior a la ejecución de la transacción

Transacciones de confirmación automática	<pre>insert into trantest2 values (1,10); insert into trantest2 values (20,20); insert into trantest2 values (3,30);</pre>
Transacciones explícitas	<pre>SET XACT_ABORT OFF BEGIN TRAN insert into trantest2 values (1,10); insert into trantest2 values (20,20); insert into trantest2 values (3,30); COMMIT</pre>
Transacciones implícitas	<pre>SET IMPLICIT_TRANSACTIONS ON insert into trantest2 values (1,10); insert into trantest2 values (20,20); insert into trantest2 values (3,30); COMMIT</pre>

Aislamiento

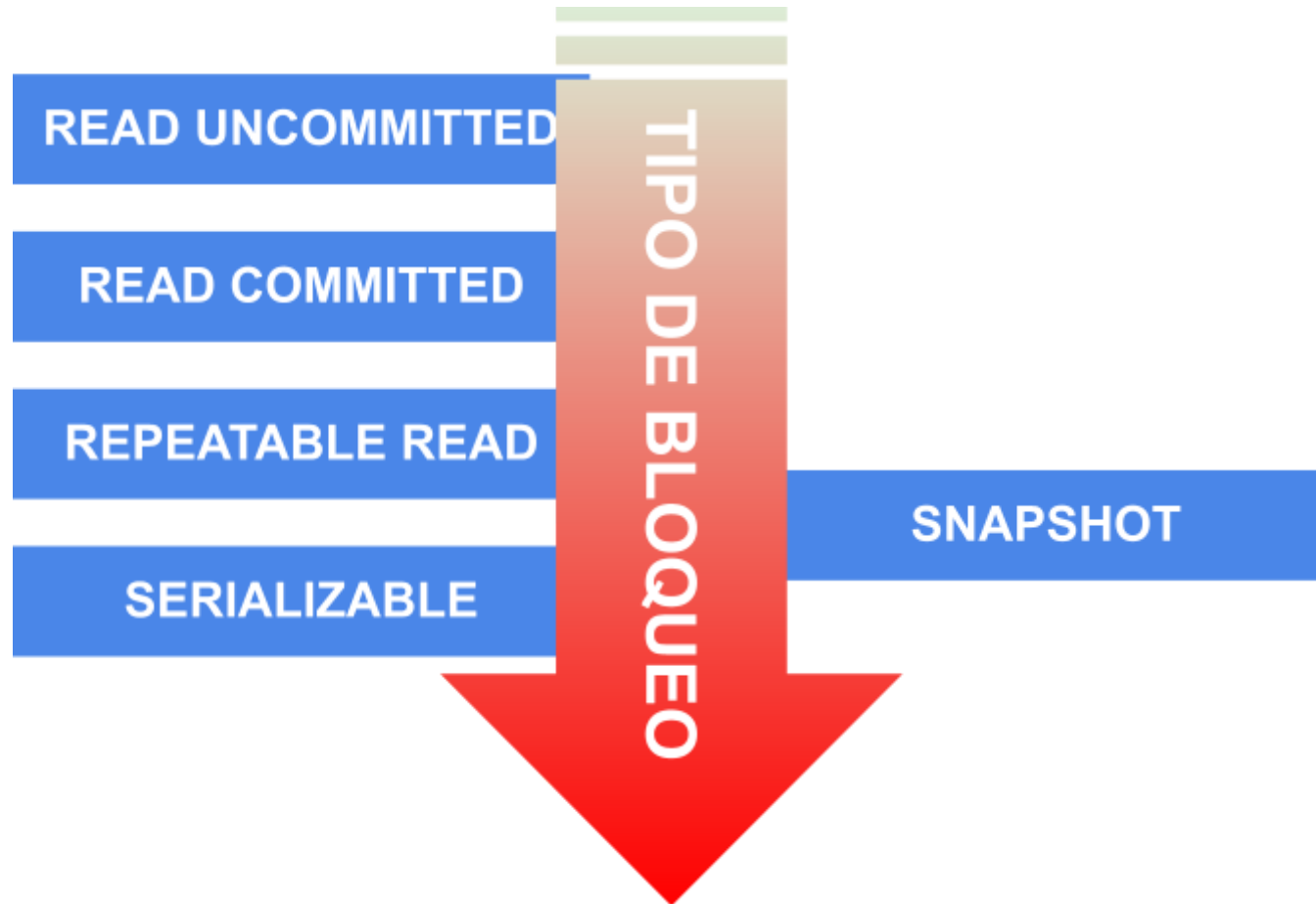
READ UNCOMMITTED	Permite lecturas no confirmadas
READ COMMITTED	Todas las lecturas que haga han de estar confirmadas
REPEATABLE READ	Ha de obtener las mismas lecturas en todo momento
SNAPSHOT	Los cambios se producen en una instantánea
SERIALIZABLE	Nadie puede cambiar los datos a actualizar

Nivel de aislamiento	Lectura sucia	Lectura no repetibles	Lectura fantasma
Read Uncommitted	puede ocurrir	puede ocurrir	puede ocurrir
Read Committed	-	puede ocurrir	puede ocurrir
Repeatable Read	-	-	puede ocurrir
Serializable	-	-	-

Nivel de aislamiento	Bloqueo de escritura	Bloqueo de lectura	Bloqueo de rango
Read Uncommitted	-	-	-
Read Committed	V	-	-
Repeatable Read	V	V	-
Serializable	V	V	V

[https://es.wikipedia.org/wiki/Aislamiento_\(ACID\)](https://es.wikipedia.org/wiki/Aislamiento_(ACID))

¿Por qué no siempre SERIALIZABLE?

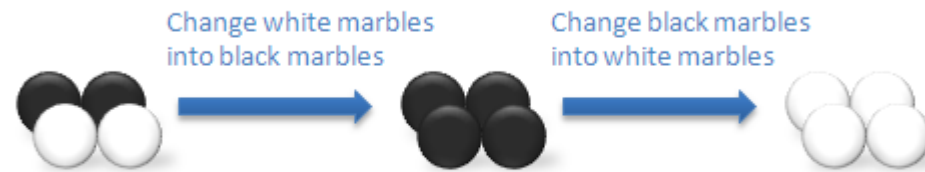


Serializable \approx bloqueo \rightarrow 1 tran detrás de otra (pesimista)

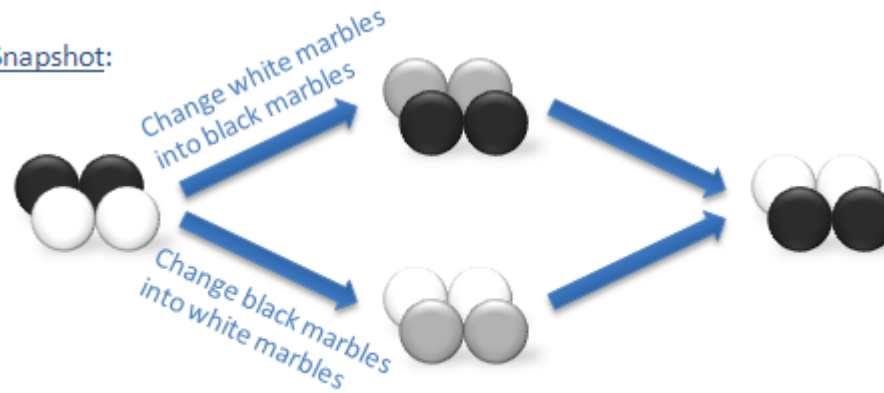
Snapshot (versiones de fila) \approx no bloqueo (excepto entre escrituras) \rightarrow copia de datos \rightarrow ¿inconsistencias? (optimista)

Tiene más sentido en BD multiversión u operaciones masivas similares a copia y respaldo (evita backup y restauración)

Serializable:



Snapshot:



Variables útiles

@@TRANCOUNT	if @@TRANCOUNT >0 rollback go
@@ERROR	delete from... IF @@ERROR <> 0 BEGIN -- Return 99 to the calling program to indicate failure. PRINT N'An error occurred deleting the candidate information.'; RETURN 99; END
@@ROWCOUNT	update ... SELECT @ErrorVar = @@ERROR ,@RowCountVar = @@ROWCOUNT ; IF @ErrorVar <> 0 BEGIN...END IF @RowCountVar <> 0 BEGIN...END

TRY..CATCH

BEGIN TRY

{ sql_statement | statement_block }

END TRY

BEGIN CATCH

[{ sql_statement | statement_block }]

END CATCH

```
CREATE PROCEDURE usp_GetErrorInfo
AS
SELECT
    ERROR_NUMBER() AS ErrorNumber
    ,ERROR_SEVERITY() AS ErrorSeverity
    ,ERROR_STATE() AS ErrorState
    ,ERROR_PROCEDURE() AS ErrorProcedure
    ,ERROR_LINE() AS ErrorLine
    ,ERROR_MESSAGE() AS ErrorMessage;
GO
```

```
BEGIN TRY
    SELECT 1/0;
END TRY
BEGIN CATCH
    EXECUTE usp_GetErrorInfo;
END CATCH
```

iNo lo captura todo!

```
BEGIN TRY
    -- Table does not exist; object name
    resolution
    -- error not caught.
    SELECT * FROM NonexistentTable;
END TRY
BEGIN CATCH
    SELECT
        ERROR_NUMBER() AS ErrorNumber
        ,ERROR_MESSAGE() AS ErrorMessage;
END CATCH
```

Messages

Msg 208, Level 16, State 1, Line 4
El nombre de objeto 'NonexistentTable' no es válido.

```
IF OBJECT_ID ( N'usp_ExampleProc', N'P' ) IS NOT
NULL
    DROP PROCEDURE usp_ExampleProc;
GO
-----
CREATE PROCEDURE usp_ExampleProc
AS
    SELECT * FROM NonexistentTable;
GO
-----
BEGIN TRY
    EXECUTE usp_ExampleProc;
END TRY
BEGIN CATCH
    SELECT
        ERROR_NUMBER() AS ErrorNumber
        ,ERROR_MESSAGE() AS ErrorMessage;
END CATCH;
```

Results

Messages

	ErrorNumber	ErrorMessage
1	208	El nombre de objeto 'NonexistentTable' no es válido.

En transacciones

BEGIN TRANSACTION;

BEGIN TRY

```
-- Generate a constraint violation error.  
DELETE FROM Production.Product  
WHERE ProductID = 980;
```

END TRY

BEGIN CATCH

```
SELECT  
    ERROR_NUMBER() AS ErrorNumber  
    ,ERROR_SEVERITY() AS ErrorSeverity  
    ,ERROR_STATE() AS ErrorState  
    ,ERROR_PROCEDURE() AS ErrorProcedure  
    ,ERROR_LINE() AS ErrorLine  
    ,ERROR_MESSAGE() AS ErrorMessage;
```

```
IF @@TRANCOUNT > 0
```

```
    ROLLBACK TRANSACTION;
```

END CATCH;

```
IF @@TRANCOUNT > 0
```

```
    COMMIT TRANSACTION;
```

```
GO
```