Transaction Isolation

1. ISOLATION LEVEL

1.1. ISOLATION_DEFAULT

하부 데이터 저장소의 기본 격리수준을 이용한다. 이는 데이터 저장소마다 다르다

1.2. ISOLATION_READ_UNCOMMITTED

커밋되지 않은 데이터 변경사항을 읽을 수 있다. 쓰레기 읽기, 재생 불가능한 읽기, 허상읽기를 허용한다

1.3. ISOLATION_READ_COMMITTED

다른 트랜잭션이 커밋된 변경사항만 읽을 수 있도록 한다. 따라서 쓰레기 읽기는 방지되지만, 재생 불가능한 읽기나 허상 읽기는 여전히 발생한다.

1.4. ISOLATION_REAPEATABLE_READ

같은 데이터 필드는 여러 번 반복해서 읽더라도 동일한 값을 읽도록 한다. 물론 자신이 변경한 필드는 변경된 값을 읽데 된다. 따라서 쓰레기 읽기와 재생 불가능한 읽기는 방지되지만, 허상 읽기는 여전히 발생한다.
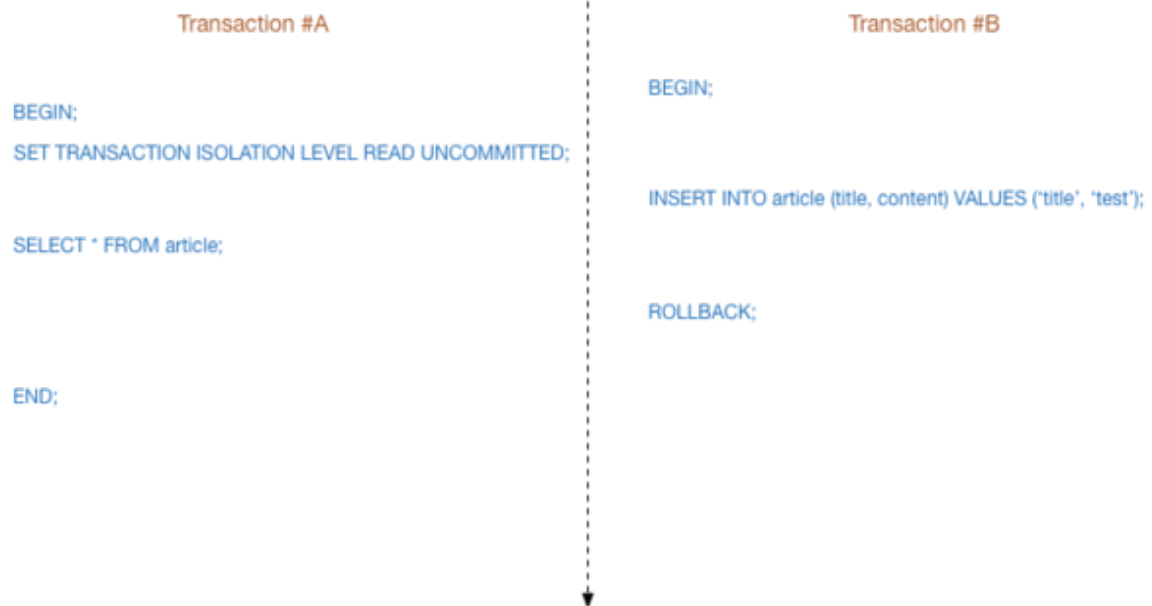
1.5. ISOLATION_SERIALIZABLE

완전한 ACID를 보장하는 격리수준으로 모든 문제점이 방지된다. 그러나 보통 해당 트랜잭션에 관련된 모든 테이블에 잠금을 걸어 버르게 되므로, 가장 성능 비효율적인 격리수준이다.

2. Read phenomena

2.1. DIRTY_READ

A dirty read (aka uncommitted dependency) occurs when a transaction is allowed to read data from a row that has been modified by another running transaction and not yet committed.
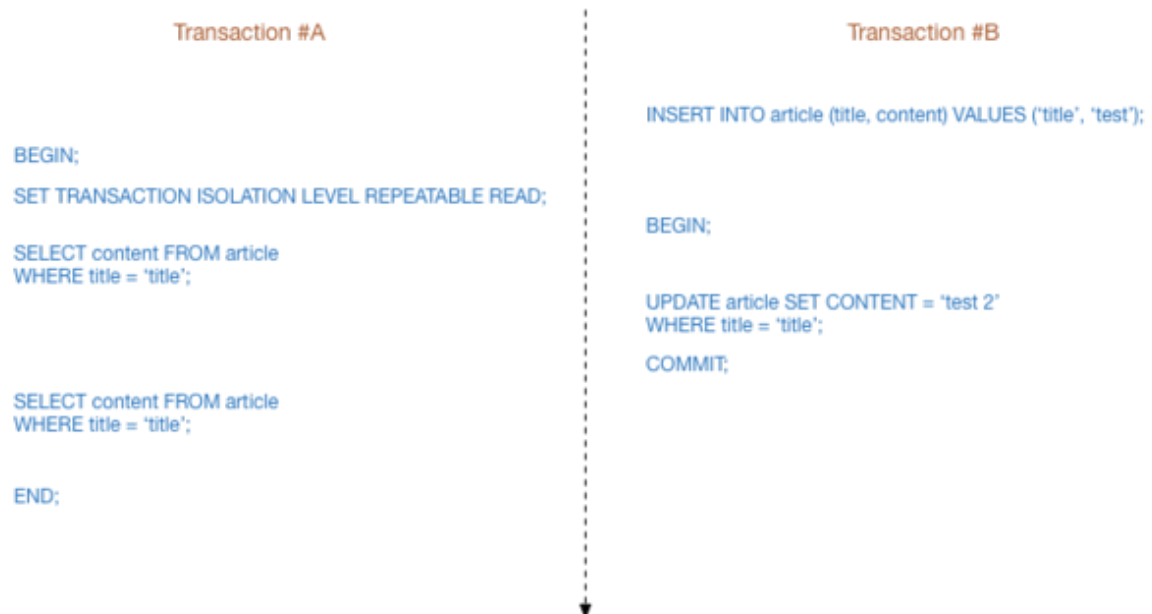
DIRTY READ

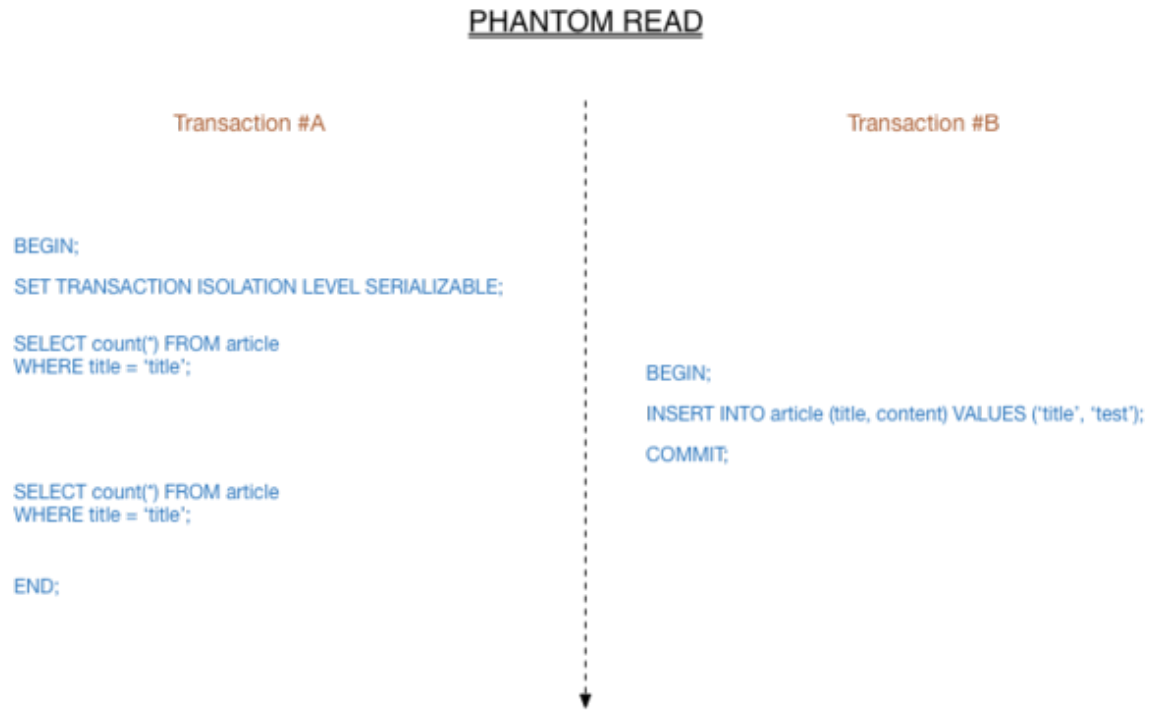| Transaction #A	Transaction #B
	BEGIN;
BEGIN;	
SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;	
	INSERT INTO article (title, content) VALUES ('title', 'test');
SELECT * FROM article;	
	ROLLBACK;
END;	

2.2. NONREPEATABLE_READ

A non-repeatable read occurs, when during the course of a transaction, a row is retrieved twice and the values within the row differ between reads.

NONREPEATABLE READ

Transaction #A	Transaction #B
	INSERT INTO article (title, content) VALUES ('title', 'test');
BEGIN;	
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;	
SELECT content FROM article WHERE title = 'title';	BEGIN;
	UPDATE article SET CONTENT = 'test 2' WHERE title = 'title';
	COMMIT;
SELECT content FROM article WHERE title = 'title';	
END;	

2.3. PHANTOM_READ

A phantom read occurs when, in the course of a transaction, two identical queries are executed, and the collection of rows returned by the second query is different from the first.

PHANTOM READ

Transaction #A | Transaction #B

```
BEGIN;

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;

SELECT count(*) FROM article
WHERE title = 'title';
                                    BEGIN;

                                    INSERT INTO article (title, content) VALUES ('title', 'test');

                                    COMMIT;

SELECT count(*) FROM article
WHERE title = 'title';

END;
```

2.4. Summary

Summary

To summarize, the existing relationship between isolation level and read phenomena may be expressed in the following table:

	dirty reads	non-repeatable reads	phantom reads
READ_UNCOMMITTED	yes	yes	yes
READ_COMMITTED	no	yes	yes
REPEATABLE_READ	no	no	yes
SERIALIZABLE	no	no	no

[PostgreSQL - Transaction](#)

In PostgreSQL, you can request any of the four standard transaction isolation levels. **But internally, there are only three distinct isolation levels, which correspond to the levels Read Committed, Repeatable Read, and Serializable**. When you select the level **Read Uncommitted you really get Read Committed**, and phantom reads are not possible in the PostgreSQL implementation of

Repeatable Read, so the actual isolation level might be stricter than what you select. This is permitted by the SQL standard: the four isolation levels only define which phenomena must not happen, they do not define which phenomena must happen. The reason that PostgreSQL only provides three isolation levels is that this is the only sensible way to map the standard isolation levels to the multiversion concurrency control architecture. The behavior of the available isolation levels is detailed in the following subsections.