Введение в реляционные базы данных

Лекция 11: Транзакции

Артем Толканев

December 14, 2024

Транзакции

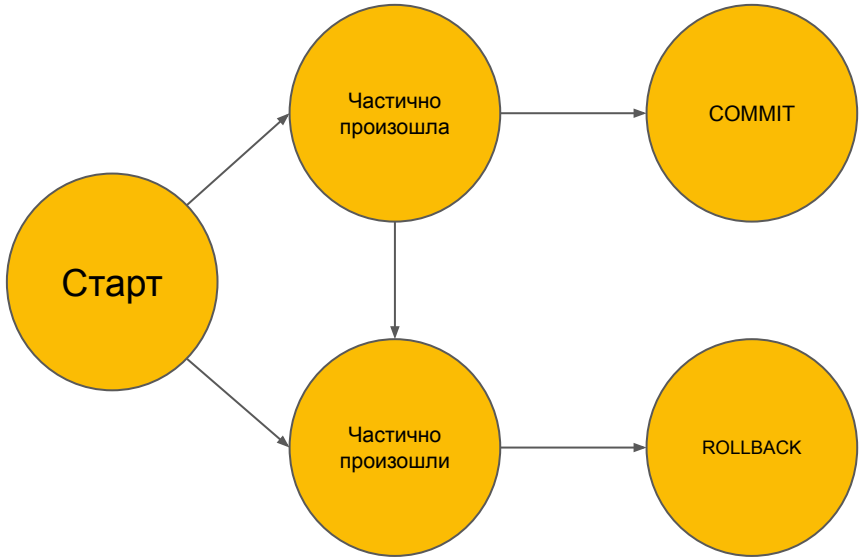
Набор операций чтения и записи данных, которые можно считать в конкретном случае логически одной, то есть полностью произойдут (`COMMIT`) или полностью отменятся(`ROLLBACK`), называют **транзакцией**.

Atomicity (**Атомарность**)

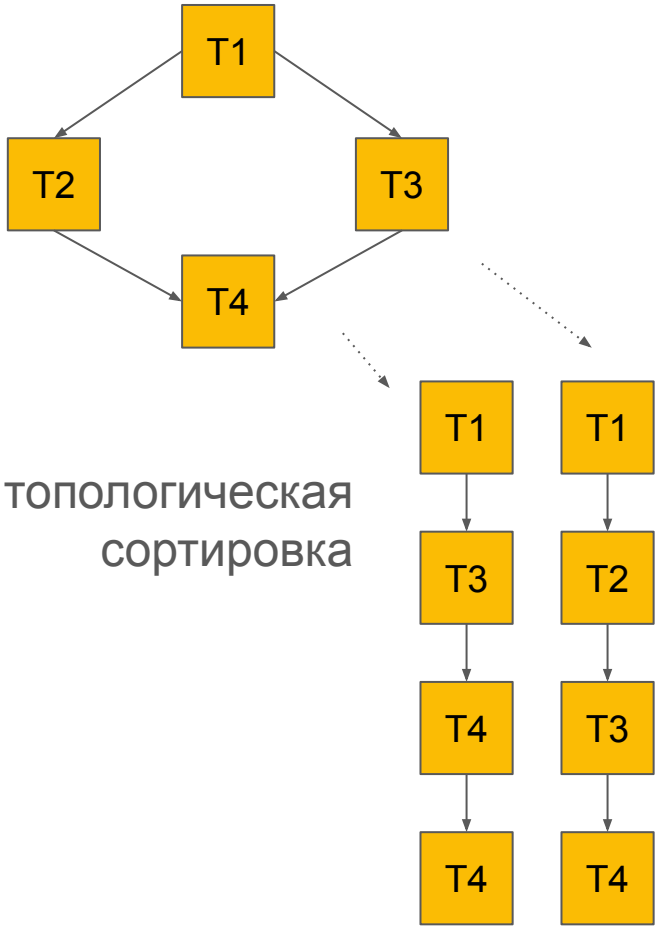
Consistency (**Согласованность**)

Isolation (**Изоляционность**)

Durability (**Устойчивость**)



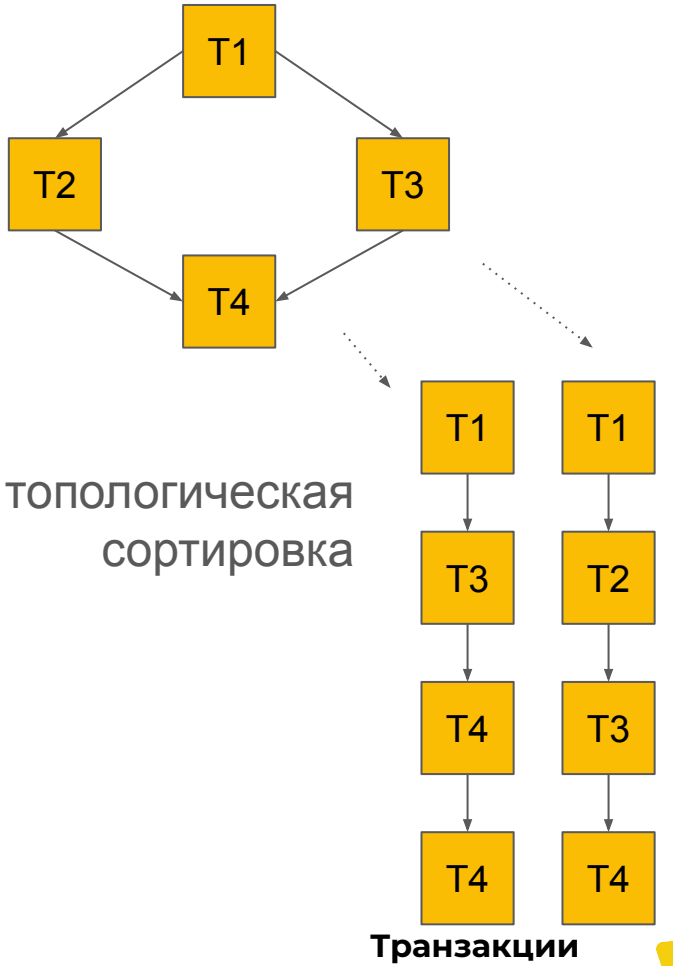
T1	T2	T3	T4
BEGIN	**BEGIN**	**BEGIN**	**BEGIN**
R(A)			
W(A)	R(A)	R(A)	
COMMIT	R(B)	R(C)	R(B)
	W(B)	W(C)	R(C)
	COMMIT	**COMMIT**	W(B)
			W(C)
			COMMIT



girafe ai

Мы можем обеспечить согласованность базы данных при параллельном выполнении, убедившись, что любое выполняемое расписание имеет тот же эффект, что и расписание, которое могло бы произошли без какого-либо параллельного выполнения.

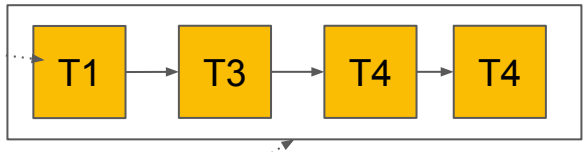
Такие расписания называются **сериализуемыми расписаниями**.



Согласованность базы данных обеспечивается:

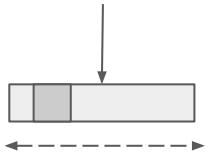
Уровень параллелизма

транзакциями



сериализуемыми расписаниями

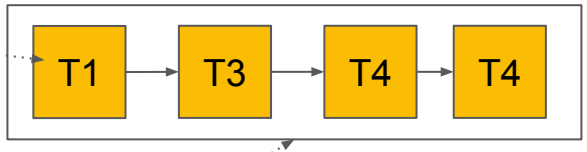
Стандарт SQL: уровни изоляции



Согласованность базы данных обеспечивается:

Уровень параллелизма

транзакциями



сериализуемыми расписаниями

Стандарт SQL: уровни изоляции

Serializable
(упорядочиваемость)

Repeatable read
(повторяющееся чтение)

Read committed
(чтение фиксированных данных)

Read uncommitted
(чтение незафиксированных данных)

Стандарт SQL: уровни изоляции

Serializable
(упорядочиваемость)

Repeatable read
(повторяющееся чтение)

Read committed
(чтение фиксированных данных)

Read uncommitted
(чтение незафиксированных данных)

```
CREATE TABLE colors( id int, col text);
INSERT INTO colors VALUES (1,'black'),(2,'white');

 BEGIN TRANSACTION ISOLATION LEVEL SERIALIZABLE;
 SELECT pg_sleep(2);
 UPDATE colors SET col = 'black'
 WHERE col = 'white';
 COMMIT;

 BEGIN TRANSACTION ISOLATION LEVEL SERIALIZABLE;
 SELECT pg_sleep(2);
 UPDATE colors SET col = 'white'
 WHERE col = 'black';
 COMMIT;
```

Стандарт SQL: уровни изоляции

Serializable
(упорядочиваемость)

Serialization Anomaly

Repeatable read
(повторяющееся чтение)

Read committed
(чтение фиксированных данных)

Read uncommitted
(чтение незафиксированных данных)

```
CREATE TABLE colors( id int, col text);
INSERT INTO colors VALUES (1,'black'),(2,'white');

BEGIN TRANSACTION ISOLATION LEVEL REPEATABLE  READ;
SELECT pg_sleep(2);
UPDATE colors SET col = 'black'
WHERE col = 'white';
COMMIT;

BEGIN TRANSACTION ISOLATION LEVEL REPEATABLE  READ;
SELECT pg_sleep(2);
UPDATE colors SET col = 'white'
WHERE col = 'black';
COMMIT;
```


Стандарт SQL: уровни изоляции

Serializable
(упорядочиваемость)

Repeatable read
(повторяющееся чтение)

Read committed
(чтение фиксированных данных)

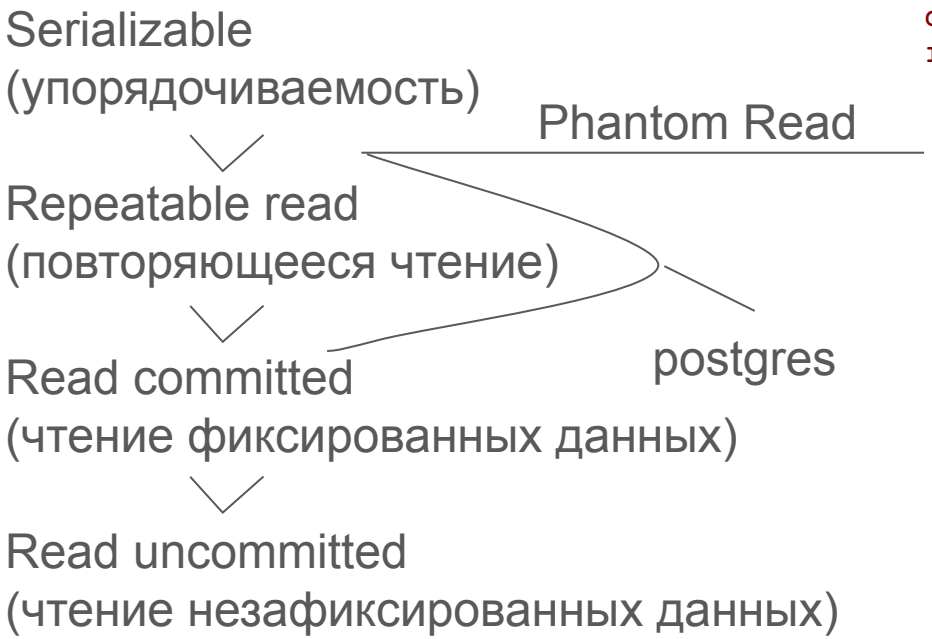
Read uncommitted
(чтение незафиксированных данных)

```
CREATE TABLE test(id serial NOT NULL,value int NULL);
INSERT INTO test(value) VALUES(1);

  BEGIN TRANSACTION ISOLATION LEVEL SERIALIZABLE;
  SELECT pg_sleep(2);
  (SELECT count(id) FROM test);
  SELECT pg_sleep(6);
  (SELECT count(id) FROM test);
  COMMIT;

  BEGIN TRANSACTION;
  SELECT pg_sleep(4);
  INSERT INTO test(value) VALUES(4);
  COMMIT;
```


Стандарт SQL: уровни изоляции




```
CREATE TABLE test(id serial NOT NULL,value int NULL);
INSERT INTO test(value) VALUES(1);

  BEGIN TRANSACTION ISOLATION LEVEL Read committed;
  SELECT pg_sleep(2);
  (SELECT count(id) FROM test);
  SELECT pg_sleep(6);
  (SELECT count(id) FROM test);
  COMMIT;

  BEGIN TRANSACTION;
  SELECT pg_sleep(4);
  INSERT INTO test(value) VALUES(4);
  COMMIT;
```

Стандарт SQL: уровни изоляции

Serializable
(упорядочиваемость)

Repeatable read
(повторяющееся чтение)

Read committed
(чтение фиксированных данных)

Read uncommitted
(чтение незафиксированных данных)

```
CREATE TABLE test(id serial NOT NULL,value int NULL);
INSERT INTO test(value) VALUES(1),(4),(4),(4);

  BEGIN TRANSACTION ISOLATION LEVEL REPEATABLE  READ;
  SELECT pg_sleep(2);
  (SELECT * FROM test WHERE value = 4);
  SELECT pg_sleep(6);
  (SELECT * FROM test WHERE value = 4);
  COMMIT;

  BEGIN TRANSACTION;
  SELECT pg_sleep(4);
  UPDATE test SET value = 2 WHERE id = 2;
  COMMIT;
```


Стандарт SQL: уровни изоляции

Serializable
(упорядочиваемость)

Repeatable read
(повторяющееся чтение)

Nonrepeatable Read

Read committed
(чтение фиксированных данных)

Read uncommitted
(чтение незафиксированных данных)

```
CREATE TABLE test(id serial NOT NULL,value int NULL);
INSERT INTO test(value) VALUES(1),(4),(4),(4);

  BEGIN TRANSACTION ISOLATION LEVEL Read committed;
  SELECT pg_sleep(2);
  (SELECT * FROM test WHERE value = 4);
  SELECT pg_sleep(6);
  (SELECT * FROM test WHERE value = 4);
  COMMIT;

  BEGIN TRANSACTION;
  SELECT pg_sleep(4);
  UPDATE test SET value = 2 WHERE id = 2;
  COMMIT;
```


Стандарт SQL: уровни изоляции

Serializable
(упорядочиваемость)

Repeatable read
(повторяющееся чтение)

Read committed
(чтение фиксированных данных)

Read uncommitted
(чтение незафиксированных данных)

Dirty Read

не в postgres

```
CREATE TABLE test(id serial NOT NULL,value int NULL);
INSERT INTO test(value) VALUES(1),(4),(4),(4);

  BEGIN TRANSACTION ISOLATION LEVEL Read committed;
  SELECT pg_sleep(2);
  (SELECT * FROM test WHERE value = 4);
  SELECT pg_sleep(6);
  (SELECT * FROM test WHERE value = 4);
  COMMIT;

  BEGIN TRANSACTION;
  SELECT pg_sleep(4);
  UPDATE test SET value = 2 WHERE id = 2;
  SELECT pg_sleep(6);
  COMMIT;
```


Стандарт SQL: уровни изоляции

Serializable
(упорядочиваемость)

Repeatable read
(повторяющееся чтение)

Read committed
(чтение фиксированных данных)

Dirty Read

Read uncommitted
(чтение незафиксированных данных)

не в postgres -> посмотрим в MS SQL

```
CREATE TABLE test(id serial NOT NULL,value int NULL);
INSERT INTO test(value) VALUES(1),(4),(4),(4);

  BEGIN TRANSACTION;
  SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
  WAITFOR DELAY '00:00:02';
  (SELECT * FROM test WHERE value = 4);
  WAITFOR DELAY '00:00:06'
  (SELECT * FROM test WHERE value = 4);
  COMMIT;


  BEGIN TRANSACTION;
  WAITFOR DELAY '00:00:04'
  UPDATE test SET value = 2 WHERE id = 2;
  WAITFOR DELAY '00:00:06'
  ROLLBACK;
```


Стандарт SQL: уровни изоляции

Serializable
(упорядочиваемость)

Repeatable read
(повторяющееся чтение)

Read committed
(чтение фиксированных данных)

Read uncommitted
(чтение незафиксированных данных)

Вопрос:

Nonrepeatable Read

vs

Dirty Read

Стандарт SQL: уровни изоляции

Serializable
(упорядочиваемость)

Repeatable read
(повторяющееся чтение)

Read committed
(чтение фиксированных данных)

Read uncommitted
(чтение незафиксированных данных)

Вопрос:

Nonrepeatable Read

vs

Dirty Read

```
SELECT pg_sleep(6)(WAITFOR DELAY '00:00:06');
ROLLBACK;
```


О чем мы не будем говорить

Способы реализации параллельности транзакций:

Two-phase locking

Optimistic Concurrency Control

Multi-Version Concurrency Control