

Технологии обработки транзакций клиент-серверных приложений

ФИО преподавателя: Матчин Василий Тимофеевич

e-mail: matchin@mirea.ru

[Online-edu.mirea.ru](https://online-edu.mirea.ru)

online.mirea.ru

Условия обучения

- По итогам изучения дисциплины проводится экзамен
- В течение семестра необходимо выполнить все задания по календарному плану, которые опубликованы на Учебном портале
- Баллы за активность до 25 баллов

ТЕМА

Аномалии уровня изоляции Read Committed

Понимание термина транзакции

Транзакцией называется множество операций, выполняемое приложением, которое переводит базу данных из одного корректного состояния в другое корректное состояние (условие согласованное — consistent) при условии, что транзакция выполнена полностью (условие атомарности — atomicity) и без помех со стороны других транзакций (условие изоляции — isolation).

При этом система подтверждает (условие надёжности — durability) пользователю, что транзакция выполнена.

Т.о. выполняются требования ACID

Несо согласованное чтение

Из определения транзакции следует, что согласованность наступает тогда, когда транзакция переводит базу данных из одного корректного состояния в другое корректное состояние, т.е. состояние базы данных до выполнения транзакции будет абсолютно точно соответствовать состоянию базы данных после выполнения транзакции.

Несо согласованное чтение наступает тогда, когда транзакция переводит базу данных из одного корректного состояния в любое некорректное состояние, т.е. будут нарушены логические связи между данными.

Известные уровни изоляции

Изоляция \ Аномалия	Потерянные изменения	Грязное чтение	Неповторяющееся чтение	Фантомное чтение	Другие аномалии
Read Uncommitted	—	да	да	да	да
Read Committed	—	—	да	да	да
Repeatable Read	—	—	—	—	да
Serializable	—	—	—	—	—

Исходные данные

Возьмем уже использовавшийся пример данных

```
=> CREATE TABLE accounts(  
    id integer PRIMARY KEY GENERATED BY DEFAULT AS  
    IDENTITY,  
    number text UNIQUE,  
    client text,  
    amount numeric  
);
```

```
=> INSERT INTO accounts VALUES  
    (1, '1001', 'alex', 1000.00),  
    (2, '2001', 'petr', 100.00),  
    (3, '2002', 'petr', 900.00);
```

Проверка значений в БД

=> `SELECT * FROM accounts WHERE client = 'petr';`

id	number	client	amount
2	2001	petr	100.00
3	2002	petr	900.00

(2 rows)

Транзакция Т1: уменьшаем значение

Начинаем транзакцию, которая уменьшает баланс petr:

```
=> BEGIN;
```

```
=> UPDATE accounts SET amount = amount - 100  
WHERE id = 3;
```

```
|  amount  
|  -----  
|  800.00  
|  (1 row)
```

Параллельная транзакция T2

В это же время транзакция T2 выполняется одновременно с транзакцией T1 и начисляет проценты на все счета клиентов с общим балансом, равным или превышающим 1000:

```
| => UPDATE accounts SET amount = amount * 1.01  
| WHERE client IN (  
|   SELECT client  
|   FROM accounts  
|   GROUP BY client  
|   HAVING sum(amount) >= 1000  
| );
```

Транзакция T1 завершается

=> COMMIT;

Смотрим результат

=> SELECT * FROM accounts WHERE client = 'petr';

Id	number	client	amount
2	2001	petr	101.0000
3	2002	petr	808.0000

(2 rows)

Итоги работы транзакций T1 и T2

Транзакция T2 получает некорректные данные: часть строк видна на один момент времени, часть — на другой.

ТЕМА

Аномалии уровня изоляции Repeatable Read

Отсутствие неповторяющегося и фантомного чтений

Название уровня изоляции говорит о том, что чтение является повторяемым.

Проверим такое поведение и убедимся и в отсутствии фантомных чтений. Для этого в первой транзакции вернем счета petr в прежнее состояние и создадим новый счет для victor

Известные уровни изоляции

Изоляция \ Аномалия	Потерянные изменения	Грязное чтение	Неповторяющееся чтение	Фантомное чтение	Другие аномалии
Read Uncommitted	—	да	да	да	да
Read Committed	—	—	да	да	да
Repeatable Read	—	—	—	—	да
Serializable	—	—	—	—	—

Транзакция T1

=> BEGIN;

=> UPDATE accounts SET amount = 100.00 WHERE id = 2;

=> UPDATE accounts SET amount = 900.00 WHERE id = 3;

=> INSERT INTO accounts VALUES

(4, '3001', 'victor', 100.00);

=> SELECT * FROM accounts ORDER BY id;

id	number	client	amount
1	1001	alex	1000.00
2	2001	petr	100.00
3	2002	petr	900.00
4	3001	victor	100.00

Транзакция T2

| => BEGIN ISOLATION LEVEL REPEATABLE READ;

| => SELECT * FROM accounts ORDER BY id;

id	number	client	amount
1	1001	alice	1000.00
2	2001	bob	101.0000
3	2002	bob	808.0000

(3 rows)

Продолжение работы T1 и T2

T1

=> COMMIT;

T2

| => SELECT * FROM accounts ORDER BY id;

id	number	client	amount
1	1001	alex	1000.00
2	2001	petr	101.0000
3	2002	petr	808.0000

(3 rows)

| => COMMIT; //T2

Вопросы



Список литературы

1. Практическое применение команд PostgreSQL
<http://snakeproject.ru/python/Postgres.html>

Спасибо за внимание!