

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий  
институт

Кафедра «Информатика»  
кафедра

**ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ № 7**

Транзакции  
Тема

Преподаватель		<u>Е. П. Моргунов</u> Инициалы, Фамилия
	Подпись, дата	
Студент	<u>КИ19-17/1Б, №031939174</u> Номер группы, зачетной книжки	<u>А. К. Никитин</u> Инициалы, Фамилия
	Подпись, дата	

Красноярск 2021

## 1 Ход выполнения

### 1.1 Вопрос 1

```
demo=# BEGIN;  
BEGIN  
demo=# SELECT *  
demo=# FROM aircrafts  
demo=# WHERE model ~ '^Аэро'  
demo=# FOR UPDATE;  
 aircraft_code |      model      | range  
-----+-----+-----  
 320           | Аэробус А320-200 | 5700  
 321           | Аэробус А321-200 | 5600  
 319           | Аэробус А319-100 | 6700  
(3 строки)
```

Рисунок 1 – Создание первой параллельной транзакции через BEGIN

```
demo=# SELECT *  
FROM aircrafts  
WHERE model ~ '^Аэро'  
FOR UPDATE;  
█
```

Рисунок 2 – Создание второй параллельной транзакции без использования BEGIN

Как мы видим, вторая операция не завершилась, так как первая транзакция блокирует доступ к этому участку БД, а следовательно, второй запрос также является транзакцией.

### 1.2 Вопрос 3

Потерянное обновление – это когда «одна транзакция перезаписала данные, обновленные и зафиксированные другой транзакцией». В нашем примере данные одной транзакции перезаписывают результат другой транзакции, следовательно происходит потерянное обновление.

Для борьбы с этим можно поменять либо логику изменения, либо уровень изоляции транзакций на REPEATABLE READ или SERIALIZABLE.

```
demo=# BEGIN TRANSACTION ISOLATION LEVEL REPEATABLE READ;  
BEGIN  
demo=# UPDATE aircrafts_tmp  
demo=# SET range = 2500  
demo=# WHERE aircraft_code = 'CR2';  
ОШИБКА: не удалось сериализовать доступ из-за параллельного изменения  
demo=# END;  
ROLLBACK
```

Рисунок 3 – ROLLBACK при попытке изменения поля

### 1.3 Вопрос 5

```
demo=# BEGIN;  
BEGIN  
demo=# SELECT *  
demo=# FROM aircrafts_tmp  
demo=# WHERE model ~ '^Аэро'  
demo=# FOR UPDATE;  
  aircraft_code |          model          | range  
-----+-----+-----  
320             | Аэробус A320-200       | 5700  
321             | Аэробус A321-200       | 5600  
319             | Аэробус A319-100       | 6700  
(3 строки)
```

Рисунок 4 – Первая параллельная транзакция

```
demo=# BEGIN;  
BEGIN  
demo=# SELECT *  
demo=# FROM aircrafts_tmp  
demo=# WHERE model ~ '^Аэробус A320-200'  
demo=# FOR UPDATE;  
█
```

Рисунок 5 – Вторая параллельная транзакция, где множество строк является подмножеством первой транзакции

Как мы видим, операция не завершилась.

```
demo=# BEGIN;
BEGIN
demo=# SELECT *
demo-# FROM aircrafts_tmp
demo-# FOR UPDATE;
█
```

Рисунок 6 - Вторая параллельная транзакция, где множество строк является надмножеством первой транзакции

Очевидно, операция снова не завершилась.

```
demo=# BEGIN;
BEGIN
demo=# SELECT *
demo-# FROM aircrafts_tmp
demo-# WHERE range > 6000
demo-# FOR UPDATE;
█
```

Рисунок 7 - Вторая параллельная транзакция, где множество строк пересекается со множеством строк первой транзакции

И снова операция не завершилась.

```
demo=# BEGIN;
BEGIN
demo=# SELECT *
demo-# FROM aircrafts_tmp
demo-# WHERE model ~ '^Боинг'
demo-# FOR UPDATE;
aircraft_code |      model      | range
-----+-----+-----
773           | Боинг 777-300   | 11100
763           | Боинг 767-300   | 7900
733           | Боинг 737-300   | 4200
(3 строки)
```

Рисунок 8 - Вторая параллельная транзакция, где множество не пересекается со множеством строк первой транзакции

И наконец, операция завершилась успешно.

Таким образом, блокировка происходила тогда, когда вторая транзакция обращалась хотя бы к одной из блокируемых строк.

## 1.4 Вопрос 7

### 1.4.1 Блокировка через FOR UPDATE

```
demo=# BEGIN;  
BEGIN  
demo=# SELECT *  
demo=# FROM aircrafts_tmp  
demo=# WHERE model ~ '^Аэро'  
demo=# FOR UPDATE;  
aircraft_code | model | range  
-----+-----+-----  
320 | Аэробус A320-200 | 5700  
321 | Аэробус A321-200 | 5600  
319 | Аэробус A319-100 | 6700  
(3 строки)
```

Рисунок 9 – Блокировка одной части таблицы

```
demo=# BEGIN;  
BEGIN  
demo=# SELECT *  
demo=# FROM aircrafts_tmp  
demo=# WHERE model ~ '^Боинг'  
demo=# FOR UPDATE;  
aircraft_code | model | range  
-----+-----+-----  
773 | Боинг 777-300 | 11100  
763 | Боинг 767-300 | 7900  
733 | Боинг 737-300 | 4200  
(3 строки)
```

Рисунок 10 – Блокировка другой части таблицы

```
demo=# SELECT *
demo=# FROM aircrafts_tmp
demo=# WHERE model = 'Боинг 777-300'
demo=# FOR UPDATE;
```

Рисунок 11 – Попытка обратиться к заблокированной части  
Попытка привела к блокировке транзакции.

#### 1.4.2 Одновременное изменение данных при уровне изоляции Read Uncommitted

```
demo=# BEGIN ISOLATION LEVEL READ UNCOMMITTED;
BEGIN
demo=# UPDATE aircrafts_tmp
demo=# SET range = range + 500
demo=# WHERE range < 6000
demo=# RETURNING *;
```

aircraft_code	model	range
320	Аэробус A320-200	6200
321	Аэробус A321-200	6100
SU9	Сухой Суперджет-100	3500
733	Боинг 737-300	4700
CN1	Сессна 208 Караван	1700
CR2	Бомбардье CRJ-200	2600

(6 строк)

Рисунок 12 – Первая транзакция

```
demo=# BEGIN ISOLATION LEVEL READ UNCOMMITTED;
BEGIN
demo=# UPDATE aircrafts_tmp
demo=# SET range = range + 400
demo=# WHERE range < 6000
demo=# RETURNING *;
```

Рисунок 13 – Вторая транзакция

```

demo=# BEGIN ISOLATION LEVEL READ UNCOMMITTED;
BEGIN
demo=# UPDATE aircrafts_tmp
demo-# SET range = range + 300
demo-# WHERE range > 6500
demo-# RETURNING *;
  aircraft_code |          model          | range
-----+-----+-----
    773         | Боинг 777-300         | 11400
    763         | Боинг 767-300         |  8200
    319         | Аэробус A319-100      |  7000
(3 строки)

```

Рисунок 14 – Третья транзакция

## 1.5 Вопрос 9

Мы можем заключить, что параллельные сериализованные операции дали тот же результат, который бы дали последовательные операции. При создании индекса не происходит ошибки, так как поиск в этом случае происходит по `tid`, который пользователь не может изменить (не пересоздав индекс) вручную, и следовательно, транзакции не смогут внести изменений в одних и тех же строках.