

Лабораторная работа №2-2: «Транзакции. Изоляция транзакций»

Используется та же схема базы данных, что и в лабораторной работе #2.1

Подключимся с двух разных сессий:

```

pgcli -h localhost -p 5432 -d lab21 -U db_admin
> wg
pgcli -h localhost -p 5432 -d lab21 -U db_admin
Server: PostgreSQL 14.8
Version: 4.0.1
Home: http://pgcli.com
db_admin@localhost:lab21> \d
+-----+-----+-----+-----+
| Schema | Name  | Type  | Owner |
+-----+-----+-----+-----+
| public | roles | table | ksv   |
| public | tickets | table | ksv   |
| public | users | table | ksv   |
+-----+-----+-----+-----+
SELECT 3
Time: 0.008s
db_admin@localhost:lab21>

pgcli -h localhost -p 5432 -d lab21 -U ksv
> pgcli -h localhost -p 5432 -d lab21 -U ksv
Server: PostgreSQL 14.8
Version: 4.0.1
Home: http://pgcli.com
ksv@localhost:lab21> \d
+-----+-----+-----+-----+
| Schema | Name  | Type  | Owner |
+-----+-----+-----+-----+
| public | roles | table | ksv   |
| public | tickets | table | ksv   |
| public | users | table | ksv   |
+-----+-----+-----+-----+
SELECT 3
Time: 0.007s
ksv@localhost:lab21>

```

| Здесь и далее сессия слева = сессия1, сессия справа = сессия2

Создать две сессии в Вашей базе данных. Начать транзакцию на уровне изоляции READ COMMITTED в одной из сессий. Изменить и добавить какие-либо данные.

Выставим в сессии 1 уровень изоляции READ COMMITTED, изначально все таблицы пустые:

```

pgcli -h localhost -p 5432 -d lab21 -U ksv
ksv@localhost:lab21> SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
SET TRANSACTION can only be used in transaction blocks
None
SET
Time: 0.000s
ksv@localhost:lab21>

pgcli -h localhost -p 5432 -d lab21 -U ksv
ksv@localhost:lab21> select * from theatre_manager_schema performances;
+-----+-----+-----+
| performanceid | spectacleid | datetime |
+-----+-----+-----+
|               |             |          |
+-----+-----+-----+
SELECT 0
Time: 0.003s
ksv@localhost:lab21>

```

Далее вставим данные через сессию 1:

```

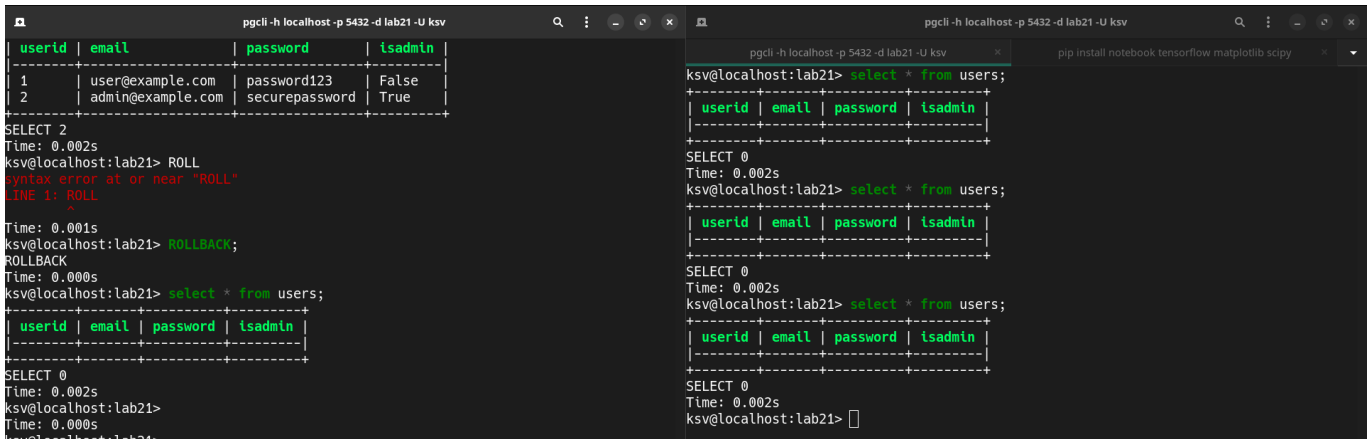
pgcli -h localhost -p 5432 -d lab21 -U ksv
ksv@localhost:lab21> START TRANSACTION ISOLATION LEVEL READ COMMITTED;
START TRANSACTION
Time: 0.000s
ksv@localhost:lab21> select * from users;
+-----+-----+-----+-----+
| userid | email | password | isadmin |
+-----+-----+-----+-----+
|        |      |          |         |
+-----+-----+-----+-----+
SELECT 0
Time: 0.002s
ksv@localhost:lab21> INSERT INTO users (UserID, Email, Password, IsAdmin) VALUES
(1, 'user@example.com', 'password123', FALSE),
(2, 'admin@example.com', 'securepassword', TRUE);
INSERT 0 2
Time: 0.001s
ksv@localhost:lab21> select * from users;
+-----+-----+-----+-----+
| userid | email | password | isadmin |
+-----+-----+-----+-----+
| 1      | user@example.com | password123 | False |
| 2      | admin@example.com | securepassword | True |
+-----+-----+-----+-----+
SELECT 2
Time: 0.002s
ksv@localhost:lab21>

pgcli -h localhost -p 5432 -d lab21 -U ksv
ksv@localhost:lab21> select * from users;
+-----+-----+-----+-----+
| userid | email | password | isadmin |
+-----+-----+-----+-----+
|        |      |          |         |
+-----+-----+-----+-----+
SELECT 0
Time: 0.002s
ksv@localhost:lab21> select * from users;
+-----+-----+-----+-----+
| userid | email | password | isadmin |
+-----+-----+-----+-----+
|        |      |          |         |
+-----+-----+-----+-----+
SELECT 0
Time: 0.002s
ksv@localhost:lab21>

```

Во второй сессии данные не отобразились так как не был совершен COMMIT на вставленные данные. В первой сессии из-за выставленного уровня изоляции мы видим незакоммиченные данные

Делаем ROLLBACK

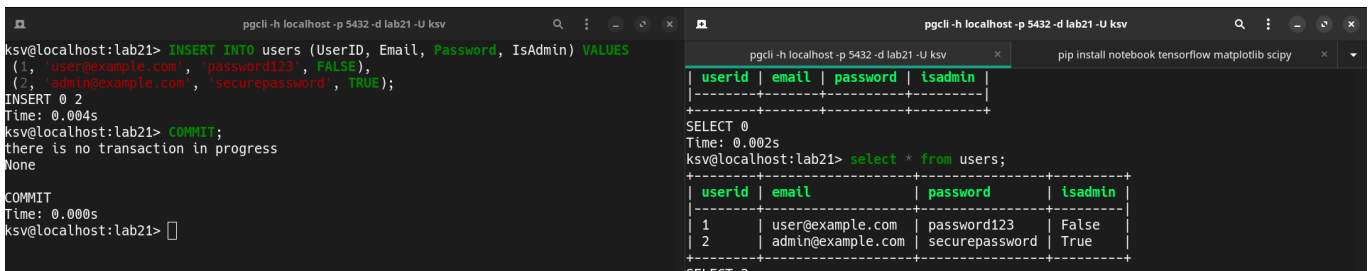


```
pgcli -h localhost -p 5432 -d lab21 -U ksv
+----+-----+-----+-----+
| userid | email | password | isadmin |
+----+-----+-----+-----+
| 1 | user@example.com | password123 | False |
| 2 | admin@example.com | securepassword | True |
+----+-----+-----+-----+
SELECT 2
Time: 0.002s
ksv@localhost:lab21> ROLL
syntax error at or near "ROLL"
LINE 1: ROLL
^
Time: 0.001s
ksv@localhost:lab21> ROLLBACK;
ROLLBACK
Time: 0.000s
ksv@localhost:lab21> select * from users;
+----+-----+-----+-----+
| userid | email | password | isadmin |
+----+-----+-----+-----+
| 1 | user@example.com | password123 | False |
| 2 | admin@example.com | securepassword | True |
+----+-----+-----+-----+
SELECT 0
Time: 0.002s
ksv@localhost:lab21>
ksv@localhost:lab21>

pgcli -h localhost -p 5432 -d lab21 -U ksv
ksv@localhost:lab21> select * from users;
+----+-----+-----+-----+
| userid | email | password | isadmin |
+----+-----+-----+-----+
| 1 | user@example.com | password123 | False |
| 2 | admin@example.com | securepassword | True |
+----+-----+-----+-----+
SELECT 0
Time: 0.002s
ksv@localhost:lab21> select * from users;
+----+-----+-----+-----+
| userid | email | password | isadmin |
+----+-----+-----+-----+
| 1 | user@example.com | password123 | False |
| 2 | admin@example.com | securepassword | True |
+----+-----+-----+-----+
SELECT 0
Time: 0.002s
ksv@localhost:lab21> select * from users;
+----+-----+-----+-----+
| userid | email | password | isadmin |
+----+-----+-----+-----+
| 1 | user@example.com | password123 | False |
| 2 | admin@example.com | securepassword | True |
+----+-----+-----+-----+
SELECT 0
Time: 0.002s
ksv@localhost:lab21> 
```

Мы отменили транзакцию => откатили изменения в базе данных

Всё-таки COMMIT'им изменения



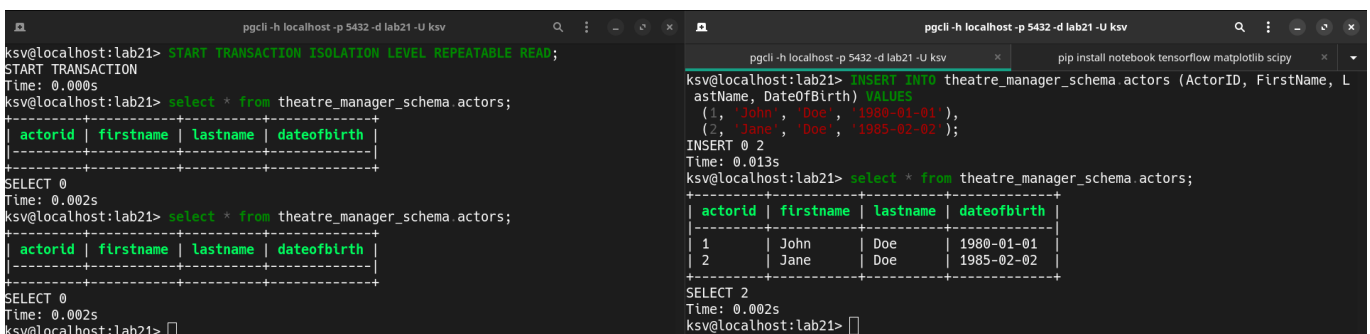
```
pgcli -h localhost -p 5432 -d lab21 -U ksv
ksv@localhost:lab21> INSERT INTO users (UserID, Email, Password, IsAdmin) VALUES
(1, 'user@example.com', 'password123', FALSE),
(2, 'admin@example.com', 'securepassword', TRUE);
INSERT 0 2
Time: 0.004s
ksv@localhost:lab21> COMMIT;
there is no transaction in progress
None
COMMIT
Time: 0.000s
ksv@localhost:lab21>

pgcli -h localhost -p 5432 -d lab21 -U ksv
+----+-----+-----+-----+
| userid | email | password | isadmin |
+----+-----+-----+-----+
| 1 | user@example.com | password123 | False |
| 2 | admin@example.com | securepassword | True |
+----+-----+-----+-----+
SELECT 0
Time: 0.002s
ksv@localhost:lab21> select * from users;
+----+-----+-----+-----+
| userid | email | password | isadmin |
+----+-----+-----+-----+
| 1 | user@example.com | password123 | False |
| 2 | admin@example.com | securepassword | True |
+----+-----+-----+-----+
SELECT 2
Time: 0.002s
ksv@localhost:lab21> 
```

Мы сохранили изменения в базе, поэтому изменения стали общедоступны и их видно из любой сессии

Выполнить те же операции для уровней изоляции REPEATABLE READ и SERIALIZABLE. Объяснить различия

REPEATABLE READ



```
pgcli -h localhost -p 5432 -d lab21 -U ksv
ksv@localhost:lab21> START TRANSACTION ISOLATION LEVEL REPEATABLE READ;
START TRANSACTION
Time: 0.000s
ksv@localhost:lab21> select * from theatre_manager_schema actors;
+----+-----+-----+-----+
| actorid | firstname | lastname | dateofbirth |
+----+-----+-----+-----+
| 1 | John | Doe | 1980-01-01 |
| 2 | Jane | Doe | 1985-02-02 |
+----+-----+-----+-----+
SELECT 0
Time: 0.002s
ksv@localhost:lab21> select * from theatre_manager_schema actors;
+----+-----+-----+-----+
| actorid | firstname | lastname | dateofbirth |
+----+-----+-----+-----+
| 1 | John | Doe | 1980-01-01 |
| 2 | Jane | Doe | 1985-02-02 |
+----+-----+-----+-----+
SELECT 0
Time: 0.002s
ksv@localhost:lab21>

pgcli -h localhost -p 5432 -d lab21 -U ksv
ksv@localhost:lab21> INSERT INTO theatre_manager_schema actors (ActorID, FirstName, L
astName, DateOfBirth) VALUES
(1, 'John', 'Doe', '1980-01-01'),
(2, 'Jane', 'Doe', '1985-02-02');
INSERT 0 2
Time: 0.013s
ksv@localhost:lab21> select * from theatre_manager_schema actors;
+----+-----+-----+-----+
| actorid | firstname | lastname | dateofbirth |
+----+-----+-----+-----+
| 1 | John | Doe | 1980-01-01 |
| 2 | Jane | Doe | 1985-02-02 |
+----+-----+-----+-----+
SELECT 2
Time: 0.002s
ksv@localhost:lab21> 
```

Из-за того что мы выставили уровень изоляции REPEATABLE READ сессия 1 видит только те изменения, которые были внесены на момент начала транзакции. Чтобы увидеть изменения,

внесенные из другой сессии необходимо совершить COMMIT:

```
ksv@localhost:lab21> select * from theatre_manager_schema.actors;
 actorid | firstname | lastname | dateofbirth
-----+-----+-----+-----
 1       | John     | Doe      | 1980-01-01
 2       | Jane     | Doe      | 1985-02-02
(2 rows)

SELECT 0
Time: 0.002s
ksv@localhost:lab21> COMMIT;
COMMIT
Time: 0.000s
ksv@localhost:lab21> select * from theatre_manager_schema.actors;
 actorid | firstname | lastname | dateofbirth
-----+-----+-----+-----
 1       | John     | Doe      | 1980-01-01
 2       | Jane     | Doe      | 1985-02-02
(2 rows)

SELECT 2
Time: 0.002s
ksv@localhost:lab21>
```

Здесь можно увидеть, что данные появились

SERIALIZABLE

```
pgcli -h localhost -p 5432 -d lab21 -U ksv
ksv@localhost:lab21> START TRANSACTION ISOLATION LEVEL SERIALIZABLE;
START TRANSACTION
Time: 0.000s
ksv@localhost:lab21> UPDATE theatre_manager_schema.actors SET lastname = 'Ivanova' WHERE
  firstname = 'Jane';
You're about to run a destructive command.
Do you want to proceed? [y/N]: y
Your call!
UPDATE 1
Time: 0.001s
ksv@localhost:lab21> select * from theatre_manager_schema.actors;
 actorid | firstname | lastname | dateofbirth
-----+-----+-----+-----
 1       | John     | Doe      | 1980-01-01
 2       | Jane     | Ivanova  | 1985-02-02
(2 rows)

SELECT 2
Time: 0.002s
ksv@localhost:lab21>
```

```
pgcli -h localhost -p 5432 -d lab21 -U ksv
> pgcli -h localhost -p 5432 -d lab21 -U ksv
Server: PostgreSQL 14.8
Version: 4.0.1
Home: http://pgcli.com
ksv@localhost:lab21> select * from theatre_manager_schema.actors;
 actorid | firstname | lastname | dateofbirth
-----+-----+-----+-----
 1       | John     | Doe      | 1980-01-01
 2       | Jane     | Doe      | 1985-02-02
(2 rows)

SELECT 2
Time: 0.004s
ksv@localhost:lab21> START TRANSACTION ISOLATION LEVEL SERIALIZABLE;
START TRANSACTION
Time: 0.000s
ksv@localhost:lab21> select * from theatre_manager_schema.actors;
 actorid | firstname | lastname | dateofbirth
-----+-----+-----+-----
 1       | John     | Doe      | 1980-01-01
 2       | Jane     | Doe      | 1985-02-02
(2 rows)

SELECT 2
Time: 0.002s
ksv@localhost:lab21> UPDATE theatre_manager_schema.actors SET lastname = 'Smirnova' WHERE
  firstname = 'Jane';
You're about to run a destructive command.
Do you want to proceed? [y/N]: y
Your call!
^Cancelled query
ksv@localhost:lab21>
```

Уровень изоляции **SERIALIZABLE** запрещает одновременное изменение данных в разных сессиях, для того чтобы произвести изменения в сессии 2 необходимо сначала совершить COMMIT в обеих сессиях:

```
ksv@localhost:lab21> UPDATE theatre_manager_schema.actors SET lastname = 'Ivanova' WHERE
  firstname = 'Jane';
You're about to run a destructive command.
Do you want to proceed? [y/N]: y
Your call!
UPDATE 1
Time: 0.001s
ksv@localhost:lab21> select * from theatre_manager_schema.actors;
 actorid | firstname | lastname | dateofbirth
-----+-----+-----+-----
 1       | John     | Doe      | 1980-01-01
 2       | Jane     | Ivanova  | 1985-02-02
(2 rows)

SELECT 2
Time: 0.002s
ksv@localhost:lab21> COMMIT;
COMMIT
Time: 0.004s
ksv@localhost:lab21> COMMIT;
there is no transaction in progress
None

COMMIT
Time: 0.000s
ksv@localhost:lab21> select * from theatre_manager_schema.actors;
 actorid | firstname | lastname | dateofbirth
-----+-----+-----+-----
 1       | John     | Doe      | 1980-01-01
 2       | Jane     | Smirnova | 1985-02-02
(2 rows)

SELECT 2
Time: 0.002s
ksv@localhost:lab21>
```

```
ksv@localhost:lab21> select * from theatre_manager_schema.actors;
 actorid | firstname | lastname | dateofbirth
-----+-----+-----+-----
 1       | John     | Doe      | 1980-01-01
 2       | Jane     | Ivanova  | 1985-02-02
(2 rows)

SELECT 2
Time: 0.002s
ksv@localhost:lab21> UPDATE theatre_manager_schema.actors SET lastname = 'Smirnova' WHERE
  firstname = 'Jane';
You're about to run a destructive command.
Do you want to proceed? [y/N]: y
Your call!
UPDATE 1
Time: 0.004s
ksv@localhost:lab21> select * from theatre_manager_schema.actors;
 actorid | firstname | lastname | dateofbirth
-----+-----+-----+-----
 1       | John     | Doe      | 1980-01-01
 2       | Jane     | Smirnova | 1985-02-02
(2 rows)

SELECT 2
syntax error at or near "\"
LINE 1: \
Time: 0.003s
ksv@localhost:lab21> COMMIT;
there is no transaction in progress
None

COMMIT
Time: 0.000s
ksv@localhost:lab21>
```

Заключение:

Были изучены три уровня изоляции в PostgreSQL:

1. READ COMMITTED:

- Транзакция видит только те данные, которые были зафиксированы до её начала.
- Минимизация блокировок, подходит для многих обычных операций.

2. REPEATABLE READ:

- Гарантирует, что любые данные, прочитанные в транзакции, не изменятся в течение всей транзакции.
- Предотвращает феномен неповторяемого чтения, но может привести к более длительным блокировкам.

3. SERIALIZABLE:

- Самый строгий уровень, обеспечивает полную изоляцию транзакции, как если бы они выполнялись последовательно.
- Используется для обеспечения полной последовательности и целостности транзакций, но может сильно снижать параллельность и производительность.