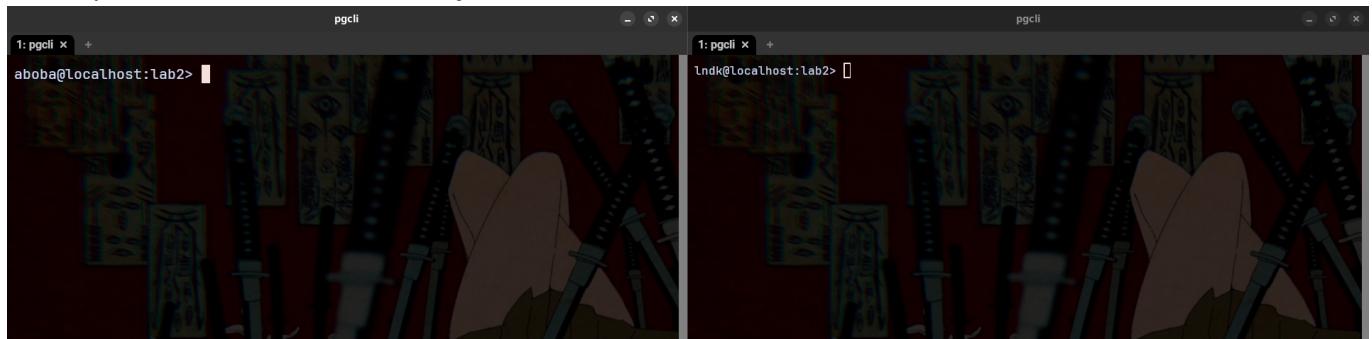


Лабораторная работа №2-2: «Транзакции. Изоляция транзакций»

Выполнил студент группы Б20-505, Ландаков Павел

Создать две сессии в Вашей базе данных. Начать транзакцию на уровне изоляции READ COMMITTED в одной из сессий. Изменить и добавить какие-либо данные. Пронаблюдать за изменениями из обеих сессий. Объяснить полученные результаты. Завершить транзакцию инструкцией ROLLBACK. Пронаблюдать за изменениями из обеих сессий. Объяснить полученные результаты.

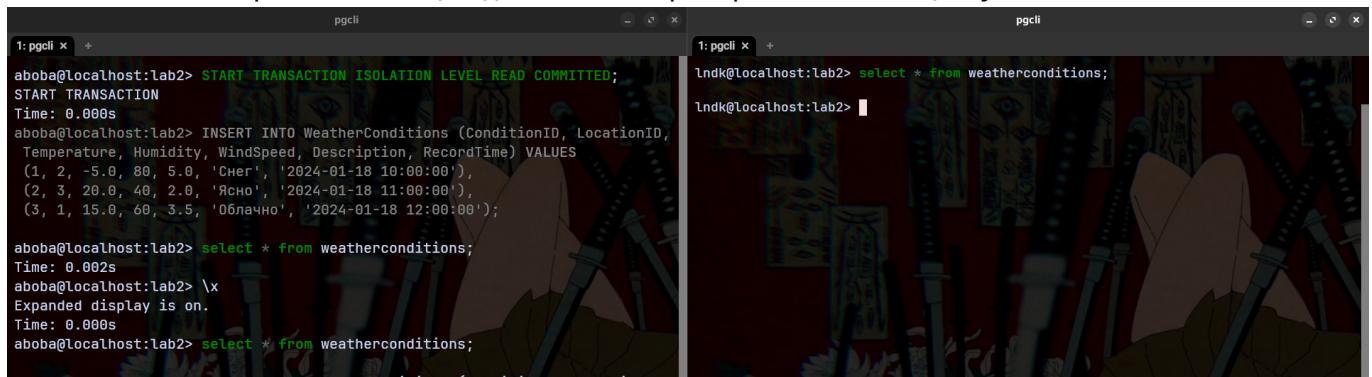
Одновременно подключимся с двух сессий `aboba` и `lndk`:



The screenshot shows two terminal windows side-by-side, both titled "pgcli". The left window is for user "aboba" and the right window is for user "lndk". Both windows show a command-line interface with a dark background and light-colored text. In the aboba session, the prompt is "1: pgcli x + aboba@localhost:lab2>". In the lndk session, the prompt is "1: pgcli x + lndk@localhost:lab2>". There is no visible output or data in either session, indicating that the tables are currently empty.

READ COMMITTED

Сначала включим режим изоляции для `aboba` и проверим что таблицы пустые:



The screenshot shows two terminal windows side-by-side, both titled "pgcli". The left window is for user "aboba" and the right window is for user "lndk". In the aboba session, the user runs a transaction block starting with `START TRANSACTION ISOLATION LEVEL READ COMMITTED;`, inserts three rows into the "WeatherConditions" table, and then runs a `select * from weatherconditions;` query which returns three rows of data. In the lndk session, the user runs a `select * from weatherconditions;` query and also returns three rows of data, showing that the changes made in the aboba session are visible to the lndk session because the transaction was not yet committed. The data inserted in the aboba session is:

| ConditionID | LocationID | Temperature | Humidity | WindSpeed | Description | RecordTime |
|-------------|------------|-------------|----------|-----------|-------------|---------------------|
| 1 | 2 | -5.0 | 80 | 5.0 | 'Снег' | 2024-01-18 10:00:00 |
| 2 | 3 | 20.0 | 40 | 2.0 | 'Ясно' | 2024-01-18 11:00:00 |
| 3 | 1 | 15.0 | 60 | 3.5 | 'Облачно' | 2024-01-18 12:00:00 |

Далее в сессии `aboba` внесем данные в таблицу, но в сессии `lndk` они не отобразятся т.к. транзакция не была сохранена:

The screenshot shows two terminal windows side-by-side. Both are titled 'pgcli'.

Left Terminal:

```
1: pgcli x +
aboba@localhost:lab2> INSERT INTO weatherconditions (ConditionID, LocationID, Temperature, Humidity, WindSpeed, Description, RecordTime) VALUES
(1, 2, -5.0, 80, 5.0, 'Снег', '2024-01-18 10:00:00'),
(2, 3, 20.0, 40, 2.0, 'Ясно', '2024-01-18 11:00:00'),
(3, 1, 15.0, 60, 3.5, 'Облачно', '2024-01-18 12:00:00');

INSERT 0 3
Time: 0.002s
aboba@localhost:lab2> select * from weatherconditions;
-[ RECORD 1 ]-----
conditionid | 1
locationid | 2
temperature | -5.0
humidity | 80
windspeed | 5.0
description | Снег
recordtime | 2024-01-18 10:00:00
-[ RECORD 2 ]-----
conditionid | 2
locationid | 3
temperature | 20.0
humidity | 40
windspeed | 2.0
description | Ясно
recordtime | 2024-01-18 11:00:00
-[ RECORD 3 ]-----
conditionid | 3
locationid | 1
temperature | 15.0
humidity | 60
windspeed | 3.5
description | Облачно
recordtime | 2024-01-18 12:00:00
```

Right Terminal:

```
1: pgcli x +
lndk@localhost:lab2> select * from weatherconditions;
lndk@localhost:lab2> select * from weatherconditions;
lndk@localhost:lab2> [REDACTED]
```

Делаем ROLLBACK:

The screenshot shows two terminal windows side-by-side. Both are titled 'pgcli'.

Left Terminal:

```
1: pgcli x +
aboba@localhost:lab2> ROLLBACK
ROLLBACK
Time: 0.000s
aboba@localhost:lab2> select * from weatherconditions;
aboba@localhost:lab2> [REDACTED]
```

Right Terminal:

```
1: pgcli x +
lndk@localhost:lab2> select * from weatherconditions;
lndk@localhost:lab2> select * from weatherconditions;
lndk@localhost:lab2> select * from weatherconditions;
lndk@localhost:lab2> [REDACTED]
```

| Внесенные в процессе транзакции изменения откатились и база данных вернулась к тому состоянию, которое было до момента начала транзакции

Вносим те же самые изменения, но завершаем транзакцию при помощи **COMMIT**:

The screenshot shows two terminal windows side-by-side. Both are titled 'pgcli'.

Left Terminal:

```
1: pgcli x +
aboba@localhost:lab2> INSERT INTO weatherconditions (ConditionID, LocationID, Temperature, Humidity, WindSpeed, Description, RecordTime) VALUES
(1, 2, -5.0, 80, 5.0, 'Снег', '2024-01-18 10:00:00'),
(2, 3, 20.0, 40, 2.0, 'Ясно', '2024-01-18 11:00:00'),
(3, 1, 15.0, 60, 3.5, 'Облачно', '2024-01-18 12:00:00');

INSERT 0 3
Time: 0.005s
aboba@localhost:lab2> COMMIT;
there is no transaction in progress
None

COMMIT
Time: 0.000s
aboba@localhost:lab2> select * from weatherconditions;
-[ RECORD 1 ]-----
conditionid | 1
locationid | 2
temperature | -5.0
humidity | 80
windspeed | 5.0
description | Снег
recordtime | 2024-01-18 10:00:00
-[ RECORD 2 ]-----
conditionid | 2
locationid | 3
temperature | 20.0
humidity | 40
windspeed | 2.0
description | Ясно
recordtime | 2024-01-18 11:00:00
-[ RECORD 3 ]-----
conditionid | 3
locationid | 1
temperature | 15.0
humidity | 60
windspeed | 3.5
description | Облачно
recordtime | 2024-01-18 12:00:00
```

Right Terminal:

```
1: pgcli x +
lndk@localhost:lab2> select * from weatherconditions;
lndk@localhost:lab2> select * from weatherconditions;
lndk@localhost:lab2> select * from weatherconditions;
lndk@localhost:lab2> [REDACTED]
```

| Транзакция была завершена, изменения внесены в базу данных

REPEATABLE READ

Начало транзакции, таблица пуста:

The screenshot shows two terminal windows side-by-side. Both are titled 'pgcli'.

Left Terminal:

```
1: pgcli x +
aboba@localhost:lab2> START TRANSACTION ISOLATION LEVEL READ COMMITTED;
START TRANSACTION
Time: 0.001s
aboba@localhost:lab2> select * from users;
+-----+-----+-----+-----+
| userid | username | email | password |
+-----+-----+-----+-----+
SELECT 0
Time: 0.002s
aboba@localhost:lab2> [REDACTED]
```

Right Terminal:

```
1: pgcli x +
lndk@localhost:lab2> select * from users;
+-----+-----+-----+-----+
| userid | username | email | password |
+-----+-----+-----+-----+
SELECT 0
Time: 0.002s
lndk@localhost:lab2> [REDACTED]
```

Далее в сессии `lndk` вносятся изменения в таблицу `users`, но в сессии `aboba` они не отображаются:

```
aboba@localhost:lab2> select * from users;
+-----+-----+-----+
| userid | username | email      | password |
+-----+-----+-----+
| 1     | exampleuser | user@example.com | password123 |
+-----+-----+-----+
SELECT 1
Time: 0.004s
aboba@localhost:lab2> select * from users;
+-----+-----+-----+
| userid | username | email      | password |
+-----+-----+-----+
| 1     | exampleuser | user@example.com | password123 |
+-----+-----+-----+
SELECT 1
Time: 0.002s
aboba@localhost:lab2> 
lndk@localhost:lab2> INSERT INTO Users (UserID, Username, Email, Password) VALUES
('1', 'exampleuser', 'user@example.com', 'password123');
INSERT 0 1
Time: 0.004s
lndk@localhost:lab2> select * from users;
+-----+-----+-----+
| userid | username | email      | password |
+-----+-----+-----+
| 1     | exampleuser | user@example.com | password123 |
+-----+-----+-----+
SELECT 1
Time: 0.002s
lndk@localhost:lab2> 
```

| Так как в REPEATABLE READ сессия `aboba` видит данные в том состоянии, в котором они были на момент начала транзакции.

COMMIT и данные появились:

```
aboba@localhost:lab2> select * from users;
+-----+-----+-----+
| userid | username | email      | password |
+-----+-----+-----+
| 1     | exampleuser | user@example.com | password123 |
+-----+-----+-----+
SELECT 1
Time: 0.002s
aboba@localhost:lab2> COMMIT;
COMMIT
Time: 0.001s
aboba@localhost:lab2> select * from users;
+-----+-----+-----+
| userid | username | email      | password |
+-----+-----+-----+
| 1     | exampleuser | user@example.com | password123 |
+-----+-----+-----+
SELECT 1
Time: 0.002s
lndk@localhost:lab2> 
lndk@localhost:lab2> INSERT INTO Users (UserID, Username, Email, Password) VALUES
('1', 'exampleuser', 'user@example.com', 'password123');
INSERT 0 1
Time: 0.004s
lndk@localhost:lab2> select * from users;
+-----+-----+-----+
| userid | username | email      | password |
+-----+-----+-----+
| 1     | exampleuser | user@example.com | password123 |
+-----+-----+-----+
SELECT 1
Time: 0.002s
lndk@localhost:lab2> 
```

SERIALIZABLE

Попытка одновременного изменения одной и той же записи из разных сессий:

```
aboba@localhost:lab2> select * from users;
+-----+-----+-----+
| userid | username | email      | password |
+-----+-----+-----+
| 1     | exampleuser | user@example.com | password123 |
+-----+-----+-----+
SELECT 1
Time: 0.004s
aboba@localhost:lab2> START TRANSACTION ISOLATION LEVEL SERIALIZABLE;
START TRANSACTION
Time: 0.000s
aboba@localhost:lab2> UPDATE Users SET Password = 'new_password123' WHERE Username = 'exampleuser';
You're about to run a destructive command.
Do you want to proceed? [y/N]: y
Your call!
UPDATE 1
Time: 0.001s
aboba@localhost:lab2> select * from users;
+-----+-----+-----+
| userid | username | email      | password |
+-----+-----+-----+
| 1     | exampleuser | user@example.com | new_password123 |
+-----+-----+-----+
SELECT 1
Time: 0.002s
lndk@localhost:lab2> select * from users;
+-----+-----+-----+
| userid | username | email      | password |
+-----+-----+-----+
| 1     | exampleuser | user@example.com | password123 |
+-----+-----+-----+
SELECT 1
Time: 0.004s
lndk@localhost:lab2> START TRANSACTION ISOLATION LEVEL SERIALIZABLE;
START TRANSACTION
Time: 0.000s
lndk@localhost:lab2> UPDATE Users SET Password = 'new_new_password123' WHERE Username = 'exampleuser';
You're about to run a destructive command.
Do you want to proceed? [y/N]: y
Your call!
could not serialize access due to concurrent update
Time: 3.755s (3 seconds), executed in: 3.755s (3 seconds)
lndk@localhost:lab2> 
```

| Уровень изоляции Serializable не дает одновременно изменить одну и ту же строку в разных транзакциях.

Но после того как прописать **COMMIT** в обеих сессиях, то изменение снова станет возможным:

```
aboba@localhost:lab2> select * from users;
+-----+-----+-----+
| userid | username | email      | password |
+-----+-----+-----+
| 1     | exampleuser | user@example.com | new_password123 |
+-----+-----+-----+
SELECT 1
Time: 0.002s
aboba@localhost:lab2> COMMIT;
COMMIT
Time: 0.004s
aboba@localhost:lab2> 
lndk@localhost:lab2> COMMIT;
ROLLBACK
Time: 0.000s
lndk@localhost:lab2> select * from users;
+-----+-----+-----+
| userid | username | email      | password |
+-----+-----+-----+
| 1     | exampleuser | user@example.com | new_new_password123 |
+-----+-----+-----+
SELECT 1
Time: 0.002s
lndk@localhost:lab2> 
```

Заключение

Было исследовано и произведено знакомство с тремя уровнями транзакций в PostgreSQL:

- **READ COMMITTED**: Гарантирует, что каждая транзакция видит только те данные, которые были фиксированы до её начала. Предотвращает "грязное чтение", но не исключает неповторяющиеся чтения или фантомные чтения;
- **REPEATABLE READ**: Обеспечивает, что данные, прочитанные в транзакции, будут одинаковыми при повторном чтении. Предотвращает грязное чтение и неповторяющиеся чтения, но фантомные чтения все еще возможны;
- **SERIALIZABLE**: Наивысший уровень изоляции, который гарантирует полное изолирование транзакции. Предотвращает грязное чтение, неповторяющиеся чтения и фантомные чтения, делая транзакции полностью последовательными.