

Национальный исследовательский ядерный университет «МИФИ»
(Московский Инженерно-Физический Институт)
Кафедра №42 «Криптология и кибербезопасность»

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

Тимин Александр Б21-515 (2024г.)

1. Эксперимент с READ COMMITTED

Шаги:

1. Сессия А:

- Устанавливаем уровень изоляции и начинаем транзакцию, изменяем данные в одной из таблиц:

```
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
BEGIN;
UPDATE series.shows SET title = 'Updated Show'
WHERE id = 1;
INSERT INTO series.shows (title) VALUES ('New Show');
```

The screenshot shows the pgAdmin 4 interface. At the top, there are two tabs: 'postgres/postgres@PostgreSQL 17*' and 'postgres/postgres...'. Below the tabs is a toolbar with various icons for file operations, search, and navigation. The main area has two tabs: 'Query' (which is selected) and 'Query History'. In the 'Query' tab, the following SQL code is written:

```
1 -- Устанавливаем уровень транзакции
2 SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
3 -- Начинаем транзакцию
4 BEGIN;
5 -- Вносим изменения
6 UPDATE series.shows SET title = 'Updated Show' WHERE id = 1;
7 INSERT INTO series.shows (title) VALUES ('New Show');
8 SELECT * FROM series.shows;
```

Below the code, there are three tabs: 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is selected and displays a grid of data from the 'series.shows' table. The columns are: id [PK] integer, title text, year integer, budget real, synopsis text. The data rows are:

	id [PK] integer	title text	year integer	budget real	synopsis text
1	2	Breaking Bad	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	6	New Show	[null]	[null]	[null]
6	7	Dev Show	[null]	[null]	[null]
7	1	Updated Show	[null]	[null]	[null]
8	8	New Show	[null]	[null]	[null]

Обратим внимание на индекс новой записи – 8.

2. Сессия Б:

- Выполняем запрос для чтения данных:

```
SELECT * FROM series.shows;
```

postgres/postgres... X postgres/postgres@PostgreSQL 17*

postgres/postgres@PostgreSQL 17

No limit

Query History

```
1 -- Состояние из второй сессии
2 SELECT * FROM series.shows;
```

Data Output Messages Notifications

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thron...	[null]	[null]	[null]
2	2	Breaking Bad	[null]	[null]	[null]
3	3	Stranger Things	[null]	[null]	[null]
4	4	Friends	[null]	[null]	[null]
5	5	The Mandalorian	[null]	[null]	[null]
6	6	New Show	[null]	[null]	[null]
7	7	Dev Show	[null]	[null]	[null]

Как видно из результата запроса, в текущей сессии выводятся данные, актуальные до начала транзакции.

3. Сессия А:

- Выполняем ROLLBACK и проверяем данные:

```
ROLLBACK;
```

The screenshot shows the pgAdmin interface with two tabs: 'postgres/postgres@PostgreSQL 17*' and 'postgres/postgres@PostgreSQL 17'. The second tab is active, displaying a query history with the following SQL code:

```
8  SELECT * FROM series.shows;
9  -- Откатываем изменения
10 ROLLBACK;
11 SELECT * FROM series.shows;
12
```

Below the query history is a 'Data Output' section containing a table with the following data:

	id [PK] integer	title text	year integer	budget real	synopsis text
1		1 Game of Thron...	[null]	[null]	[null]
2		2 Breaking Bad	[null]	[null]	[null]
3		3 Stranger Things	[null]	[null]	[null]
4		4 Friends	[null]	[null]	[null]
5		5 The Mandalorian	[null]	[null]	[null]
6		6 New Show	[null]	[null]	[null]
7		7 Dev Show	[null]	[null]	[null]

Состояние БД откатилось до момента начала транзакции.

4. Сессия Б:

- Снова выполняем запрос на чтение данных.

The screenshot shows the pgAdmin interface with two tabs: 'postgres/postgres... X' and 'postgres/postgres@PostgreSQL 17* X'. The second tab is active, displaying a query history with the following SQL code:

```
1  -- Состояние из второй сессии
2  SELECT * FROM series.shows;
```

Below the query history is a 'Data Output' section containing a table with the following data:

	id [PK] integer	title text	year integer	budget real	synopsis text
1		1 Game of Thron...	[null]	[null]	[null]
2		2 Breaking Bad	[null]	[null]	[null]
3		3 Stranger Things	[null]	[null]	[null]
4		4 Friends	[null]	[null]	[null]
5		5 The Mandalorian	[null]	[null]	[null]
6		6 New Show	[null]	[null]	[null]
7		7 Dev Show	[null]	[null]	[null]

Тут – без сюрпризов.

5. Сессия А:

- о Повторно начинаем транзакцию, но теперь доводим ее до завершения командой COMMIT:

```
BEGIN;  
UPDATE series.shows SET title = 'Updated Show'  
WHERE id = 1;  
INSERT INTO series.shows (title) VALUES  
('Committed Show');  
COMMIT;
```

postgres/postgres@PostgreSQL 17* X postgres/postgres... X

postgres/postgres@PostgreSQL 17

No limit

Query History

```
12 -- Завершаем транзакцию
13 BEGIN;
14 UPDATE series.shows SET title = 'Updated Show' WH
15 INSERT INTO series.shows (title) VALUES ('Committe
16 COMMIT;
17 SELECT * FROM series.shows;
```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	2	Breaking Bad	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	6	New Show	[null]	[null]	[null]
6	7	Dev Show	[null]	[null]	[null]
7	1	Updated Show	[null]	[null]	[null]
8	9	Committed Sh...	[null]	[null]	[null]

6. Сессия Б:

- Проверим, как изменились данные после COMMIT.

postgres/postgres... X postgres/postgres@PostgreSQL 17* X

postgres/postgres@PostgreSQL 17

No limit

Query History

```
1 -- Состояние из второй сессии
2 SELECT * FROM series.shows;
```

Data Output Messages Notifications

	id [PK] integer	title text	year integer	budget real	synopsis text
1	2	Breaking Bad	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	6	New Show	[null]	[null]	[null]
6	7	Dev Show	[null]	[null]	[null]
7	1	Updated Show	[null]	[null]	[null]
8	9	Committed Sh...	[null]	[null]	[null]

Теперь состояние БД идентично в обеих сессиях.

Обратим внимание, что индекс новой записи теперь не 8 (как было раньше), а 9. Предположу, что на последовательность, обеспечивающую автоинкремент, данный уровень изоляции не распространяется. Такое поведение (насколько я понимаю) необходимо для исключения потенциально возможных коллизий при одновременной вставке данных в одну и ту же таблицу в разных транзакциях.

2. Другие уровни изоляции

2.1. Эксперимент с REPEATED READ

Шаги:

1. Сессия А:

- Устанавливаем уровень изоляции и начинаем транзакцию, изменяя данные в одной из таблиц:

```
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
BEGIN;
UPDATE series.shows SET title = 'Another One...
Updated Show' WHERE id = 1;
INSERT INTO series.shows (title) VALUES ('Extra
New Show');
```

The screenshot shows the pgAdmin 4 interface. At the top, there are two tabs: 'postgres/postgres@PostgreSQL 17*' and 'postgres/postgres...'. Below the tabs is a toolbar with various icons for database management. The main area is divided into two sections: 'Query History' and 'Data Output'.

Query History:

```
19 -- Устанавливаем уровень транзакции
20 SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
21 -- Начинаем транзакцию
22 BEGIN;
23 -- Вносим изменения
24 UPDATE series.shows SET title = 'Another One... Updated Show' WHERE id = 1;
25 INSERT INTO series.shows (title) VALUES ('Extra New Show');
26 SELECT * FROM series.shows;
```

Data Output:

	<code>id</code> [PK] integer	<code>title</code> text	<code>year</code> integer	<code>budget</code> real	<code>synopsis</code> text
1	2	Breaking Bad	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	6	New Show	[null]	[null]	[null]
6	7	Dev Show	[null]	[null]	[null]
7	9	Committed Show	[null]	[null]	[null]
8	1	Another One... Updated Show	[null]	[null]	[null]
9	10	Extra New Show	[null]	[null]	[null]

Снова обратим внимание на индекс новой записи – 10.

2. Сессия Б:

- Выполняем запрос для чтения данных:

```
SELECT * FROM series.shows;
```

The screenshot shows the pgAdmin 4 interface with two tabs at the top: 'postgres/postgres@PostgreSQL 17*' and 'postgres/postgres...'. Below the tabs is a toolbar with various icons. The main area is divided into 'Query' and 'Data Output' tabs. The 'Query' tab contains three numbered SQL statements: 27 (comment), 28 (ROLLBACK), and 29 (SELECT query). The 'Data Output' tab displays a table with columns: id [PK] integer, title text, year integer, budget real, and synopsis text. The table rows show data from the 'shows' table.

	id [PK] integer	title text	year integer	budget real	synopsis text
1	2	Breaking Bad	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	6	New Show	[null]	[null]	[null]
6	7	Dev Show	[null]	[null]	[null]
7	1	Updated Show	[null]	[null]	[null]
8	9	Committed Sh...	[null]	[null]	[null]

Как видно из результата запроса, в текущей сессии выводятся данные, актуальные до начала транзакции.

3. Сессия А:

- Выполняем ROLLBACK и проверяем данные:

```
ROLLBACK;
```

The screenshot shows the pgAdmin interface with two tabs: 'postgres/postgres@PostgreSQL 17*' and 'postgres/postgres...'. The first tab is active, showing a query history with the following content:

```
27 -- Откатываем изменения
28 ROLLBACK;
29 SELECT * FROM series.shows;
```

Below the query history, there are tabs for 'Data Output', 'Messages', and 'Notifications'. Under 'Data Output', a table is displayed with the following data:

	id [PK] integer	title text	year integer	budget real	synopsis text
1	2	Breaking Bad	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	6	New Show	[null]	[null]	[null]
6	7	Dev Show	[null]	[null]	[null]
7	1	Updated Show	[null]	[null]	[null]
8	9	Committed Sh...	[null]	[null]	[null]

Состояние БД откатилось до момента начала транзакции.

4. Сессия Б:

- Снова выполняем запрос на чтение данных.

The screenshot shows the pgAdmin interface with two tabs: 'postgres/postgres... X' and 'postgres/postgres@PostgreSQL 17* X'. The second tab is active, showing a query history with the following content:

```
1 -- Состояние из второй сессии
2 SELECT * FROM series.shows;
```

Below the query history, there are tabs for 'Data Output', 'Messages', and 'Notifications'. Under 'Data Output', a table is displayed with the following data:

	id [PK] integer	title text	year integer	budget real	synopsis text
1	2	Breaking Bad	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	6	New Show	[null]	[null]	[null]
6	7	Dev Show	[null]	[null]	[null]
7	1	Updated Show	[null]	[null]	[null]
8	9	Committed Sh...	[null]	[null]	[null]

Тут – без сюрпризов.

5. Сессия А:

- Повторно начинаем транзакцию, но теперь доводим ее до завершения командой COMMIT:

```
BEGIN;  
UPDATE series.shows SET title = 'Another One... Updated Show' WHERE id = 1;  
INSERT INTO series.shows (title) VALUES ('Extra... Committed Show');  
COMMIT;
```

The screenshot shows a pgAdmin 4 interface with two tabs open: 'postgres/postgres@PostgreSQL 17*' and 'postgres/postgres...'. The top tab displays a transaction log with the following content:

```
29  SELECT * FROM series.shows;  
30  -- Завершаем транзакцию  
31  BEGIN;  
32  UPDATE series.shows SET title = 'Another One... Updated S  
33  INSERT INTO series.shows (title) VALUES ('Extra Committed  
34  COMMIT;  
35  SELECT * FROM series.shows;  
36
```

The bottom tab shows a table viewer for the 'series.shows' table with the following data:

	id [PK] integer	title text	year integer	budget real	synopsis text
1	2	Breaking Bad	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	6	New Show	[null]	[null]	[null]
6	7	Dev Show	[null]	[null]	[null]
7	9	Committed Show	[null]	[null]	[null]
8	1	Another One... Updated Show	[null]	[null]	[null]
9	11	Extra Committed Show	[null]	[null]	[null]

6. Сессия Б:

- Проверим, как изменились данные после COMMIT.

postgres/postgres... X postgres/postgres@PostgreSQL 17* X

postgres/postgres@PostgreSQL 17

No limit

Query History

```
1 -- Состояние из второй сессии
2 SELECT * FROM series.shows;
```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	2	Breaking Bad	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	6	New Show	[null]	[null]	[null]
6	7	Dev Show	[null]	[null]	[null]
7	9	Committed Show	[null]	[null]	[null]
8	1	Another One... Updated Show	[null]	[null]	[null]
9	11	Extra Committed Show	[null]	[null]	[null]

Теперь состояние БД идентично в обеих сессиях.

Обратим внимание, что, как и в предыдущем эксперименте, индекс новой записи во время транзакции (которую позже откатили) отличается от индекса новой записи из закоммиченной транзакции.

Результат данного эксперимента внешне идентичен предыдущему. Чтобы наблюдать различие, необходимо было вносить изменения в таблицу из сессии Б в то время, когда в сессии А начата и не завершена транзакция.

2.2. Эксперимент с SERIALIZABLE

Шаги:

1. Сессия А:

- Устанавливаем уровень изоляции и начинаем транзакцию, изменяем данные в одной из таблиц:

```
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
BEGIN;
UPDATE series.shows SET title = 'And Another
One... Updated Show' WHERE id = 1;
INSERT INTO series.shows (title) VALUES ('Mega
Extra New Show');
```

The screenshot shows a pgAdmin 4 interface with a single connection tab labeled 'postgres/postgres@PostgreSQL 17'. The main area displays a query editor with the following SQL code:

```
37 -- Устанавливаем уровень транзакции
38 SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
39 -- Начинаем транзакцию
40 BEGIN;
41 -- Вносим изменения
42 UPDATE series.shows SET title = 'And Another One... Updated Show' WHERE id = 1;
43 INSERT INTO series.shows (title) VALUES ('Mega Extra New Show');
44 SELECT * FROM series.shows;
```

Below the query editor, there are tabs for 'Data Output' and 'Messages'. The 'Data Output' tab is selected and shows a table with the following data:

	id [PK] integer	title text	year integer	budget real	synopsis text
1	2	Breaking Bad	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	6	New Show	[null]	[null]	[null]
6	7	Dev Show	[null]	[null]	[null]
7	9	Committed Show	[null]	[null]	[null]
8	11	Extra Committed Show	[null]	[null]	[null]
9	1	And Another One... Updated Show	[null]	[null]	[null]
10	12	Mega Extra New Show	[null]	[null]	[null]

Снова обратим внимание на индекс новой записи – 12.

2. Сессия Б:

- Выполняем запрос для чтения данных:

```
SELECT * FROM series.shows;
```

-- Состояние из второй сессии

1 SELECT * FROM series.shows;

	id [PK] integer	title text	year integer	budget real	synopsis text
1	2	Breaking Bad	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	6	New Show	[null]	[null]	[null]
6	7	Dev Show	[null]	[null]	[null]
7	9	Committed Show	[null]	[null]	[null]
8	1	Another One... Updated Show	[null]	[null]	[null]
9	11	Extra Committed Show	[null]	[null]	[null]

Как видно из результата запроса, в текущей сессии выводятся данные, актуальные до начала транзакции.

3. Сессия А:

- Выполняем ROLLBACK и проверяем данные:

```
ROLLBACK;
```

postgres/postgres@PostgreSQL 17* × postgres/postgres... ×

postgres/postgres@PostgreSQL 17 No limit E

Query History

```
45 -- Откатываем изменения
46 ROLLBACK;
47 SELECT * FROM series.shows;
```

Data Output Messages Notifications

	id [PK] integer	title text	year integer	budget real	synopsis text
1	2	Breaking Bad	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	6	New Show	[null]	[null]	[null]
6	7	Dev Show	[null]	[null]	[null]
7	9	Committed Show	[null]	[null]	[null]
8	1	Another One... Updated Show	[null]	[null]	[null]
9	11	Extra Committed Show	[null]	[null]	[null]

Состояние БД откатилось до момента начала транзакции.

4. Сессия Б:

- Снова выполняем запрос на чтение данных.

postgres/postgres... × postgres/postgres@PostgreSQL 17* ×

postgres/postgres@PostgreSQL 17 No limit E

Query History

```
1 -- Состояние из второй сессии
2 SELECT * FROM series.shows;
```

Data Output Messages Notifications

	id [PK] integer	title text	year integer	budget real	synopsis text
1	2	Breaking Bad	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	6	New Show	[null]	[null]	[null]
6	7	Dev Show	[null]	[null]	[null]
7	9	Committed Show	[null]	[null]	[null]
8	1	Another One... Updated Show	[null]	[null]	[null]
9	11	Extra Committed Show	[null]	[null]	[null]

Тут – без сюрпризов.

5. Сессия А:

- Повторно начинаем транзакцию, но теперь доводим ее до завершения командой COMMIT:

```
BEGIN;  
UPDATE series.shows SET title = 'And Another  
One... Updated Show' WHERE id = 1;  
INSERT INTO series.shows (title) VALUES ('Mega  
Extra... Committed Show');  
COMMIT;
```

The screenshot shows a pgAdmin 4 interface with two tabs open: 'postgres/postgres@PostgreSQL 17*' and 'postgres/postgres...'. The top bar includes standard database management icons like folder, file, search, filter, and table. Below the tabs is a toolbar with various buttons for navigating and managing data.

The main area displays a query history and a data output table. The query history shows the following sequence of statements:

```
47  SELECT * FROM series.shows;  
48  -- Завершаем транзакцию  
49  BEGIN;  
50  UPDATE series.shows SET title = 'And Another One... Updated  
51  INSERT INTO series.shows (title) VALUES ('Mega Extra Commit'  
52  COMMIT;  
53  SELECT * FROM series.shows;
```

The 'Data Output' tab is active, showing a table with the following data:

	id [PK] integer	title text	year integer	budget real	synopsis text
1	2	Breaking Bad	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	6	New Show	[null]	[null]	[null]
6	7	Dev Show	[null]	[null]	[null]
7	9	Committed Show	[null]	[null]	[null]
8	11	Extra Committed Show	[null]	[null]	[null]
9	1	And Another One... Updated Show	[null]	[null]	[null]
10	13	Mega Extra Committed Show	[null]	[null]	[null]

6. Сессия Б:

- Проверим, как изменились данные после COMMIT.

The screenshot shows the pgAdmin 4 interface. At the top, there are two tabs: 'postgres/postgres...' and 'postgres/postgres@PostgreSQL 17*'. The second tab is active. Below the tabs is a toolbar with various icons for database management. The main area is divided into sections: 'Query' (which is selected), 'Query History', 'Data Output' (selected), 'Messages', and 'Notifications'. The 'Query' section contains the following SQL code:

```
1 -- Состояние из второй сессии
2 SELECT * FROM series.shows;
```

The 'Data Output' section displays the results of the query as a table:

	<code>id</code> [PK] integer	<code>title</code> text	<code>year</code> integer	<code>budget</code> real	<code>synopsis</code> text
1	2	Breaking Bad	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	6	New Show	[null]	[null]	[null]
6	7	Dev Show	[null]	[null]	[null]
7	9	Committed Show	[null]	[null]	[null]
8	11	Extra Committed Show	[null]	[null]	[null]
9	1	And Another One... Updated Show	[null]	[null]	[null]
10	13	Mega Extra Committed Show	[null]	[null]	[null]

Теперь состояние БД идентично в обеих сессиях.

Обратим внимание, что, как и в предыдущем эксперименте, индекс новой записи во время транзакции (которую позже откатили) отличается от индекса новой записи из закоммиченной транзакции.

Результат данного эксперимента внешне идентичен предыдущему. Чтобы наблюдать различие, необходимо было добавлять новые записи в таблицу из сессии Б в то время, когда в сессии А начата и не завершена транзакция, и сразу же проверять наличие фантомов (насколько я понял принцип их возникновения).

3. Видимые различия

Дизайн предыдущих экспериментов не позволял заметить различия в разных уровнях изоляции, изменим его.

Следующие три эксперимента будут проходить по следующему плану:

- 1) Сессия А. Задание уровня изоляции, начало первой транзакции.

Обновление записи с индексом 1, вставка новой записи. Вывод содержимого таблицы.

```
-- Устанавливаем уровень транзакции  
SET TRANSACTION ISOLATION LEVEL ...;
```

```
-- Начинаем транзакцию  
BEGIN;
```

```
-- Вносим изменения  
UPDATE series.shows SET title = 'I Want to Sleep' WHERE  
id = 1;  
INSERT INTO series.shows (title) VALUES ('Abobus  
Show');  
SELECT * FROM series.shows;
```

- 2) Сессия Б. Вывод содержимого таблицы.

```
-- Состояние из второй сессии  
SELECT * FROM series.shows;
```

- 3) Сессия Б. Пытаемся изменить запись с индексом 1.

```
-- Попытка перезаписать запись, изменяемую в 1 сессии  
UPDATE series.shows SET title = 'Lets Try to Overwrite'  
WHERE id = 1;
```

- 4) Сессия Б. Изменяем запись с индексом 2, вставляем новую запись.

Выводим содержимое таблицы.

```
-- Модифицируем данные  
UPDATE series.shows SET title = 'I Want to Sleep too'  
WHERE id = 2;  
INSERT INTO series.shows (title) VALUES ('Mega Abobus  
Show');
```

```
SELECT * FROM series.shows;
```

5) Сессия А. Выводим содержимое таблицы.

```
-- Чо там во второй сессии поменялось?
```

```
SELECT * FROM series.shows;
```

6) Сессия А. Откатываем транзакцию и выводим содержимое таблицы.

```
-- Откатываем изменения
```

```
ROLLBACK;
```

```
SELECT * FROM series.shows;
```

7) Сессия А. Начинаем вторую транзакцию. Обновляем запись с индексом 1, добавляем новую запись в таблицу. Выводим содержимое таблицы.

```
-- Вторая транзакция
```

```
BEGIN;
```

```
UPDATE series.shows SET title = 'I Want to Sleep' WHERE id = 1;
```

```
INSERT INTO series.shows (title) VALUES ('Committed Abobus Show');
```

```
SELECT * FROM series.shows;
```

8) Сессия Б. Выводим содержимое таблицы.

```
-- Состояние из второй сессии
```

```
SELECT * FROM series.shows;
```

9) Сессия Б. Пытаемся изменить запись с индексом 1.

```
-- Попытка перезаписать запись, изменяемую в 1 сессии
```

```
UPDATE series.shows SET title = 'Lets Try to Overwrite' WHERE id = 1;
```

10) Сессия Б. Изменяем записи с индексом 3, вставляем новую запись.

Выводим содержимое таблицы.

```
UPDATE series.shows SET title = 'Lets Try to Overwrite... Again...' WHERE id = 1;
```

```
UPDATE series.shows SET title = 'I Want to Sleep too' WHERE id = 3;
```

```
INSERT INTO series.shows (title) VALUES ('New Mega
Abobus Show');
SELECT * FROM series.shows;
```

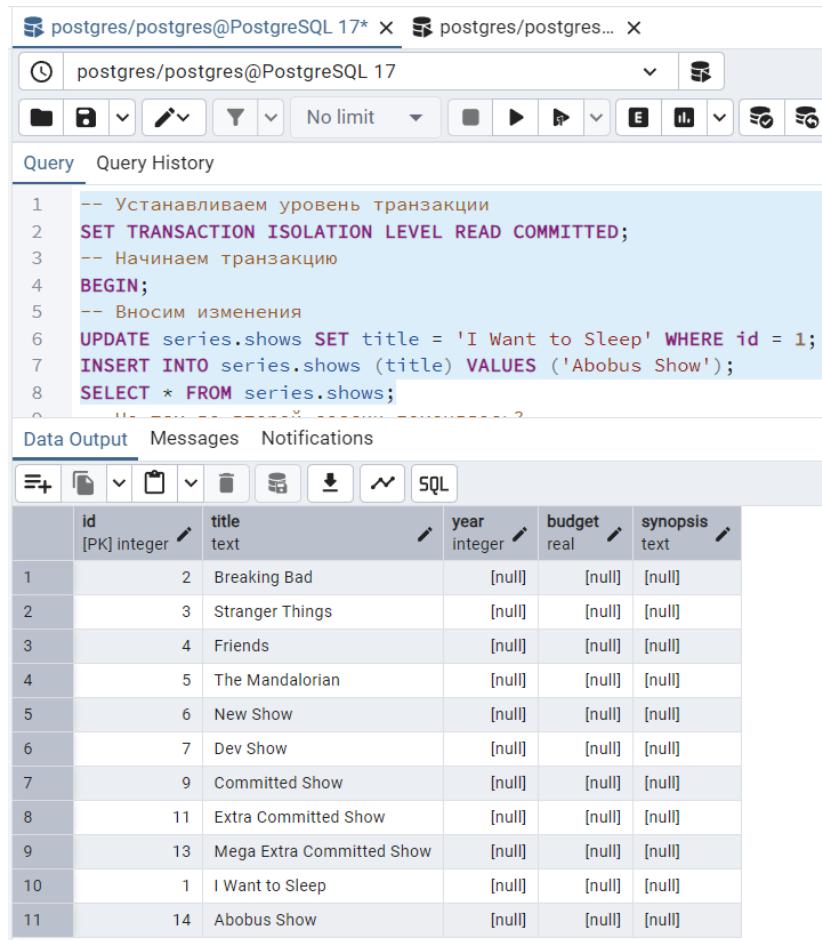
11) Сессия А. Выводим содержимое таблицы.

```
-- Чо там во второй сессии поменялось?
SELECT * FROM series.shows;
```

12) Сессия А. Коммитим транзакцию. Выводим содержимое таблицы.

```
COMMIT;
-- Чо после коммита?
SELECT * FROM series.shows;
```

3.1 READ COMMITTED



The screenshot shows a pgAdmin 4 interface with two tabs open: 'postgres/postgres@PostgreSQL 17*' and 'postgres/postgres...'. The top bar includes standard database management icons. Below the tabs is a toolbar with various buttons. The main area has tabs for 'Query' (selected), 'Query History', 'Data Output', 'Messages', and 'Notifications'. The 'Query' tab contains the following SQL code:

```
1 -- Устанавливаем уровень транзакции
2 SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
3 -- Начинаем транзакцию
4 BEGIN;
5 -- Вносим изменения
6 UPDATE series.shows SET title = 'I Want to Sleep' WHERE id = 1;
7 INSERT INTO series.shows (title) VALUES ('Abobus Show');
8 SELECT * FROM series.shows;
```

Below the code, the 'Data Output' tab is active, displaying a table with the following data:

	id [PK] integer	title text	year integer	budget real	synopsis text
1	2	Breaking Bad	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	6	New Show	[null]	[null]	[null]
6	7	Dev Show	[null]	[null]	[null]
7	9	Committed Show	[null]	[null]	[null]
8	11	Extra Committed Show	[null]	[null]	[null]
9	13	Mega Extra Committed Show	[null]	[null]	[null]
10	1	I Want to Sleep	[null]	[null]	[null]
11	14	Abobus Show	[null]	[null]	[null]

Пока все штатно.

postgres/postgres... X postgres/postgres@PostgreSQL 17* X

postgres/postgres@PostgreSQL 17

No limit

Query History

```
1 -- Состояние из второй сессии
2 SELECT * FROM series.shows;
3 Модифицировано 0 строк
```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	2	Breaking Bad	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	6	New Show	[null]	[null]	[null]
6	7	Dev Show	[null]	[null]	[null]
7	9	Committed Show	[null]	[null]	[null]
8	11	Extra Committed Show	[null]	[null]	[null]
9	1	And Another One... Updated Show	[null]	[null]	[null]
10	13	Mega Extra Committed Show	[null]	[null]	[null]

Из второй сессии не видны изменения, внесенные в транзакции.

The screenshot shows a pgAdmin 4 interface with two tabs: 'postgres/postgres...' and 'postgres/postgres@PostgreSQL 17*'. The second tab is active, displaying a SQL query window. The query is:

```
2 SELECT * FROM series.shows;
3 -- Попытка перезаписать запись, изменяемую в 1 сессии
4 UPDATE series.shows SET title = 'Lets Try to Overwrite' WHERE id = 1;
5 -- Модифицируемая запись
```

Below the query window, there are three tabs: 'Data Output', 'Messages' (which is selected), and 'Notifications'. The 'Messages' tab displays the following error output:

```
ERROR: выполнение оператора отменено по запросу пользователя
CONTEXT: при изменении кортежа (0,18) в отношении "shows"

ОШИБКА: выполнение оператора отменено по запросу пользователя
SQL state: 57014
```

Попытка изменить запись, модифицируемую в активной транзакции, терпит фиаско: СУБД блокирует доступ к записи, ожидая, пока транзакция завершится (запрос был прерван вручную).

postgres/postgres... X postgres/postgres@PostgreSQL 17* X

postgres/postgres@PostgreSQL 17

No limit

Query History

```
-- Модифицируем данные
UPDATE series.shows SET title = 'I Want to Sleep too' WHERE id = 2;
INSERT INTO series.shows (title) VALUES ('Mega Abobus Show');
SELECT * FROM series.shows;
```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	3	Stranger Things	[null]	[null]	[null]
2	4	Friends	[null]	[null]	[null]
3	5	The Mandalorian	[null]	[null]	[null]
4	6	New Show	[null]	[null]	[null]
5	7	Dev Show	[null]	[null]	[null]
6	9	Committed Show	[null]	[null]	[null]
7	11	Extra Committed Show	[null]	[null]	[null]
8	1	And Another One... Updated Show	[null]	[null]	[null]
9	13	Mega Extra Committed Show	[null]	[null]	[null]
10	2	I Want to Sleep too	[null]	[null]	[null]
11	15	Mega Abobus Show	[null]	[null]	[null]

Ничего необычного.

postgres/postgres@PostgreSQL 17* × postgres/postgres... ×

postgres/postgres@PostgreSQL 17

No limit

Query History

```
9 -- Что там во второй сессии поменялось?
10 SELECT * FROM series.shows;
11 Откатили изменения
```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	3	Stranger Things	[null]	[null]	[null]
2	4	Friends	[null]	[null]	[null]
3	5	The Mandalorian	[null]	[null]	[null]
4	6	New Show	[null]	[null]	[null]
5	7	Dev Show	[null]	[null]	[null]
6	9	Committed Show	[null]	[null]	[null]
7	11	Extra Committed Show	[null]	[null]	[null]
8	13	Mega Extra Committed Show	[null]	[null]	[null]
9	1	I Want to Sleep	[null]	[null]	[null]
10	14	Abobus Show	[null]	[null]	[null]
11	2	I Want to Sleep too	[null]	[null]	[null]
12	15	Mega Abobus Show	[null]	[null]	[null]

Хоть мы и находимся в транзакции, мы можем наблюдать изменения из другой сессии.

postgres/postgres@PostgreSQL 17* X postgres/postgres... X

postgres/postgres@PostgreSQL 17 No limit

Query History

```
11 -- Откатываем изменения
12 ROLLBACK;
13 SELECT * FROM series.shows;
```

Data Output Messages Notifications

	id [PK] integer	title text	year integer	budget real	synopsis text
1	3	Stranger Things	[null]	[null]	[null]
2	4	Friends	[null]	[null]	[null]
3	5	The Mandalorian	[null]	[null]	[null]
4	6	New Show	[null]	[null]	[null]
5	7	Dev Show	[null]	[null]	[null]
6	9	Committed Show	[null]	[null]	[null]
7	11	Extra Committed Show	[null]	[null]	[null]
8	1	And Another One... Updated Show	[null]	[null]	[null]
9	13	Mega Extra Committed Show	[null]	[null]	[null]
10	2	I Want to Sleep too	[null]	[null]	[null]
11	15	Mega Abobus Show	[null]	[null]	[null]

Откатились только изменения актуальной транзакции.

postgres/postgres@PostgreSQL 17* × postgres/postgres... ×

postgres/postgres@PostgreSQL 17

No limit

Query History

```
-- Вторая транзакция
BEGIN;
UPDATE series.shows SET title = 'I Want to Sleep' WHERE id = 1;
INSERT INTO series.shows (title) VALUES ('Committed Abobus Show');
SELECT * FROM series.shows;
```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	3	Stranger Things	[null]	[null]	[null]
2	4	Friends	[null]	[null]	[null]
3	5	The Mandalorian	[null]	[null]	[null]
4	6	New Show	[null]	[null]	[null]
5	7	Dev Show	[null]	[null]	[null]
6	9	Committed Show	[null]	[null]	[null]
7	11	Extra Committed Show	[null]	[null]	[null]
8	13	Mega Extra Committed Show	[null]	[null]	[null]
9	2	I Want to Sleep too	[null]	[null]	[null]
10	15	Mega Abobus Show	[null]	[null]	[null]
11	1	I Want to Sleep	[null]	[null]	[null]
12	16	Committed Abobus Show	[null]	[null]	[null]

В Багдаде все спокойно...

postgres/postgres... X postgres/postgres@PostgreSQL 17* X

postgres/postgres@PostgreSQL 17

No limit

Query History

```
9 -- Состояние из второй сессии
10 SELECT * FROM series.shows;
```

Data Output Messages Notifications

	id [PK] integer	title text	year integer	budget real	synopsis text
1	3	Stranger Things	[null]	[null]	[null]
2	4	Friends	[null]	[null]	[null]
3	5	The Mandalorian	[null]	[null]	[null]
4	6	New Show	[null]	[null]	[null]
5	7	Dev Show	[null]	[null]	[null]
6	9	Committed Show	[null]	[null]	[null]
7	11	Extra Committed Show	[null]	[null]	[null]
8	1	And Another One... Updated Show	[null]	[null]	[null]
9	13	Mega Extra Committed Show	[null]	[null]	[null]
10	2	I Want to Sleep too	[null]	[null]	[null]
11	15	Mega Abobus Show	[null]	[null]	[null]

Ожидаемо не видим изменений из транзакции.

The screenshot shows a pgAdmin 4 interface with two tabs at the top: 'postgres/postgres...' and 'postgres/postgres@PostgreSQL 17*'. The second tab is active. Below the tabs is a toolbar with various icons for database management. The main area is divided into sections: 'Query' (selected), 'Query History', 'Data Output' (disabled), 'Messages' (selected), and 'Notifications' (disabled). The 'Messages' section contains the following error message:

```
11 -- Попытка перезаписать запись, изменяемую в 1 сессии
12 UPDATE series.shows SET title = 'Lets Try to Overwrite' WHERE id = 1;
--
```

ERROR: выполнение оператора отменено по запросу пользователя
CONTEXT: при изменении кортежа (0,18) в отношении "shows"

ОШИБКА: выполнение оператора отменено по запросу пользователя
SQL state: 57014

Подвисание при попытке изменить запись, затронутую в транзакции.

postgres/postgres... X postgres/postgres@PostgreSQL 17* X

postgres/postgres@PostgreSQL 17

No limit

Query History

```
12 UPDATE series.shows SET title = 'Lets Try to Overwrite' WHERE id = 1;
-- Модифицируем данные
14 UPDATE series.shows SET title = 'I Want to Sleep too' WHERE id = 3;
15 INSERT INTO series.shows (title) VALUES ('New Mega Abobus Show');
16 SELECT * FROM series.shows;
```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	4	Friends	[null]	[null]	[null]
2	5	The Mandalorian	[null]	[null]	[null]
3	6	New Show	[null]	[null]	[null]
4	7	Dev Show	[null]	[null]	[null]
5	9	Committed Show	[null]	[null]	[null]
6	11	Extra Committed Show	[null]	[null]	[null]
7	1	And Another One... Updated Show	[null]	[null]	[null]
8	13	Mega Extra Committed Show	[null]	[null]	[null]
9	2	I Want to Sleep too	[null]	[null]	[null]
10	15	Mega Abobus Show	[null]	[null]	[null]
11	3	I Want to Sleep too	[null]	[null]	[null]
12	17	New Mega Abobus Show	[null]	[null]	[null]

Все штатно.

postgres/postgres@PostgreSQL 17* × postgres/postgres... ×

postgres/postgres@PostgreSQL 17 No limit E

Query History

```
18 SELECT * FROM series.shows;
19 -- Что там во второй сессии поменялось?
20 SELECT * FROM series.shows;
```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	4	Friends	[null]	[null]	[null]
2	5	The Mandalorian	[null]	[null]	[null]
3	6	New Show	[null]	[null]	[null]
4	7	Dev Show	[null]	[null]	[null]
5	9	Committed Show	[null]	[null]	[null]
6	11	Extra Committed Show	[null]	[null]	[null]
7	13	Mega Extra Committed Show	[null]	[null]	[null]
8	2	I Want to Sleep too	[null]	[null]	[null]
9	15	Mega Abobus Show	[null]	[null]	[null]
10	1	I Want to Sleep	[null]	[null]	[null]
11	16	Committed Abobus Show	[null]	[null]	[null]
12	3	I Want to Sleep too	[null]	[null]	[null]
13	17	New Mega Abobus Show	[null]	[null]	[null]

Как и в прошлый раз, видим изменения из второй сессии.

postgres/postgres@PostgreSQL 17* X postgres/postgres... X

postgres/postgres@PostgreSQL 17

No limit

Query History

```
20 SELECT * FROM series.shows;
21 COMMIT;
22 -- Что после коммита?
23 SELECT * FROM series.shows;
```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	4	Friends	[null]	[null]	[null]
2	5	The Mandalorian	[null]	[null]	[null]
3	6	New Show	[null]	[null]	[null]
4	7	Dev Show	[null]	[null]	[null]
5	9	Committed Show	[null]	[null]	[null]
6	11	Extra Committed Show	[null]	[null]	[null]
7	13	Mega Extra Committed Show	[null]	[null]	[null]
8	2	I Want to Sleep too	[null]	[null]	[null]
9	15	Mega Abobus Show	[null]	[null]	[null]
10	1	I Want to Sleep	[null]	[null]	[null]
11	16	Committed Abobus Show	[null]	[null]	[null]
12	3	I Want to Sleep too	[null]	[null]	[null]
13	17	New Mega Abobus Show	[null]	[null]	[null]

Ни никаких изменений по сравнению с предыдущим скрином.

3.2 REPEATABLE READ

The screenshot shows a pgAdmin 4 interface with two tabs open: 'postgres/postgres@PostgreSQL 17*' and 'postgres/postgres...'. The top bar includes a clock icon, session name, and various toolbar icons. Below the tabs is a toolbar with file, edit, search, and navigation buttons, along with a dropdown set to 'No limit'.

The main area has a 'Query' tab selected, showing the following SQL code:

```
1 -- Устанавливаем уровень транзакции
2 SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
3 -- Начинаем транзакцию
4 BEGIN;
5 -- Вносим изменения
6 UPDATE series.shows SET title = 'I Want to Sleep' WHERE id = 1;
7 INSERT INTO series.shows (title) VALUES ('Abobus Show');
8 SELECT * FROM series.shows;
```

Below the code, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is selected, displaying a table with the following data:

	id [PK] integer	title text	year integer	budget real	synopsis text
1	4	Friends	[null]	[null]	[null]
2	5	The Mandalorian	[null]	[null]	[null]
3	6	New Show	[null]	[null]	[null]
4	7	Dev Show	[null]	[null]	[null]
5	9	Committed Show	[null]	[null]	[null]
6	11	Extra Committed Show	[null]	[null]	[null]
7	13	Mega Extra Committed Show	[null]	[null]	[null]
8	2	I Want to Sleep too	[null]	[null]	[null]
9	15	Mega Abobus Show	[null]	[null]	[null]
10	16	Committed Abobus Show	[null]	[null]	[null]
11	3	I Want to Sleep too	[null]	[null]	[null]
12	17	New Mega Abobus Show	[null]	[null]	[null]
13	1	I Want to Sleep	[null]	[null]	[null]
14	18	Abobus Show	[null]	[null]	[null]

Пока что все по-обычному.

postgres/postgres... X postgres/postgres@PostgreSQL 17* X



postgres/postgres@PostgreSQL 17



No limit



Query Query History

```
1 -- Состояние из второй сессии
2 SELECT * FROM series.shows;
```

Data Output Messages Notifications



SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	4	Friends	[null]	[null]	[null]
2	5	The Mandalorian	[null]	[null]	[null]
3	6	New Show	[null]	[null]	[null]
4	7	Dev Show	[null]	[null]	[null]
5	9	Committed Show	[null]	[null]	[null]
6	11	Extra Committed Show	[null]	[null]	[null]
7	13	Mega Extra Committed Show	[null]	[null]	[null]
8	2	I Want to Sleep too	[null]	[null]	[null]
9	15	Mega Abobus Show	[null]	[null]	[null]
10	1	I Want to Sleep	[null]	[null]	[null]
11	16	Committed Abobus Show	[null]	[null]	[null]
12	3	I Want to Sleep too	[null]	[null]	[null]
13	17	New Mega Abobus Show	[null]	[null]	[null]

Снова.

The screenshot shows the pgAdmin 4 interface with two tabs open: 'postgres/postgres...' and 'postgres/postgres@PostgreSQL 17*'. The second tab is active and displays a SQL query window. The query is:

```
-- Попытка перезаписать запись, изменяемую в 1 сессии
UPDATE series.shows SET title = 'Lets Try to Overwrite' WHERE id = 1;
```

Below the query, the 'Messages' tab is selected, showing the following error messages:

```
ERROR: выполнение оператора отменено по запросу пользователя
CONTEXT: при изменении кортежа (0,24) в отношении "shows"

ОШИБКА: выполнение оператора отменено по запросу пользователя
SQL state: 57014
```

Подвисание (норм).

postgres/postgres... X postgres/postgres@PostgreSQL 17* X

postgres/postgres@PostgreSQL 17 No limit

Query History

```
5 -- Модифицируем данные
6 UPDATE series.shows SET title = 'I Want to Sleep too' WHERE id = 2;
7 INSERT INTO series.shows (title) VALUES ('Mega Abobus Show');
8 SELECT * FROM series.shows;
```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	4	Friends	[null]	[null]	[null]
2	5	The Mandalorian	[null]	[null]	[null]
3	6	New Show	[null]	[null]	[null]
4	7	Dev Show	[null]	[null]	[null]
5	9	Committed Show	[null]	[null]	[null]
6	11	Extra Committed Show	[null]	[null]	[null]
7	13	Mega Extra Committed Show	[null]	[null]	[null]
8	15	Mega Abobus Show	[null]	[null]	[null]
9	1	I Want to Sleep	[null]	[null]	[null]
10	16	Committed Abobus Show	[null]	[null]	[null]
11	3	I Want to Sleep too	[null]	[null]	[null]
12	17	New Mega Abobus Show	[null]	[null]	[null]
13	2	I Want to Sleep too	[null]	[null]	[null]
14	19	Mega Abobus Show	[null]	[null]	[null]

Все по плану.

postgres/postgres@PostgreSQL 17* × postgres/postgres... ×

postgres/postgres@PostgreSQL 17

No limit

Query History

```
9 -- Что там во второй сессии поменялось?
10 SELECT * FROM series.shows;
```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	4	Friends	[null]	[null]	[null]
2	5	The Mandalorian	[null]	[null]	[null]
3	6	New Show	[null]	[null]	[null]
4	7	Dev Show	[null]	[null]	[null]
5	9	Committed Show	[null]	[null]	[null]
6	11	Extra Committed Show	[null]	[null]	[null]
7	13	Mega Extra Committed Show	[null]	[null]	[null]
8	2	I Want to Sleep too	[null]	[null]	[null]
9	15	Mega Abobus Show	[null]	[null]	[null]
10	16	Committed Abobus Show	[null]	[null]	[null]
11	3	I Want to Sleep too	[null]	[null]	[null]
12	17	New Mega Abobus Show	[null]	[null]	[null]
13	1	I Want to Sleep	[null]	[null]	[null]
14	18	Abobus Show	[null]	[null]	[null]

Опа! Не видим изменений из второй сессии.

Признаться, я думал, что фантомы будут, но, видимо, я неправильно понял, когда они возникают(

postgres/postgres@PostgreSQL 17

Query History

```
11 -- Откатываем изменения
12 ROLLBACK;
13 SELECT * FROM series.shows;
```

Data Output Messages Notifications

	id [PK] integer	title text	year integer	budget real	synopsis text
1	4	Friends	[null]	[null]	[null]
2	5	The Mandalorian	[null]	[null]	[null]
3	6	New Show	[null]	[null]	[null]
4	7	Dev Show	[null]	[null]	[null]
5	9	Committed Show	[null]	[null]	[null]
6	11	Extra Committed Show	[null]	[null]	[null]
7	13	Mega Extra Committed Show	[null]	[null]	[null]
8	15	Mega Abobus Show	[null]	[null]	[null]
9	1	I Want to Sleep	[null]	[null]	[null]
10	16	Committed Abobus Show	[null]	[null]	[null]
11	3	I Want to Sleep too	[null]	[null]	[null]
12	17	New Mega Abobus Show	[null]	[null]	[null]
13	2	I Want to Sleep too	[null]	[null]	[null]
14	19	Mega Abobus Show	[null]	[null]	[null]

Теперь видим изменения из второй сессии.

postgres/postgres@PostgreSQL 17* X postgres/postgres... X

postgres/postgres@PostgreSQL 17

No limit

Query History

```
14 -- Вторая транзакция
15 BEGIN;
16 UPDATE series.shows SET title = 'I Want to Sleep' WHERE id = 1;
17 INSERT INTO series.shows (title) VALUES ('Committed Abobus Show');
18 SELECT * FROM series.shows;
```

Но там же второй запрос поменялся?

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	4	Friends	[null]	[null]	[null]
2	5	The Mandalorian	[null]	[null]	[null]
3	6	New Show	[null]	[null]	[null]
4	7	Dev Show	[null]	[null]	[null]
5	9	Committed Show	[null]	[null]	[null]
6	11	Extra Committed Show	[null]	[null]	[null]
7	13	Mega Extra Committed Show	[null]	[null]	[null]
8	15	Mega Abobus Show	[null]	[null]	[null]
9	16	Committed Abobus Show	[null]	[null]	[null]
10	3	I Want to Sleep too	[null]	[null]	[null]
11	17	New Mega Abobus Show	[null]	[null]	[null]
12	2	I Want to Sleep too	[null]	[null]	[null]
13	19	Mega Abobus Show	[null]	[null]	[null]
14	1	I Want to Sleep	[null]	[null]	[null]
15	20	Committed Abobus Show	[null]	[null]	[null]

Штиль... Ветер молчанин...

postgres/postgres... X postgres/postgres@PostgreSQL 17* X

postgres/postgres@PostgreSQL 17

No limit

Query History

```
-- Состояние из второй сессии
SELECT * FROM series.shows;
```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	4	Friends	[null]	[null]	[null]
2	5	The Mandalorian	[null]	[null]	[null]
3	6	New Show	[null]	[null]	[null]
4	7	Dev Show	[null]	[null]	[null]
5	9	Committed Show	[null]	[null]	[null]
6	11	Extra Committed Show	[null]	[null]	[null]
7	13	Mega Extra Committed Show	[null]	[null]	[null]
8	15	Mega Abobus Show	[null]	[null]	[null]
9	1	I Want to Sleep	[null]	[null]	[null]
10	16	Committed Abobus Show	[null]	[null]	[null]
11	3	I Want to Sleep too	[null]	[null]	[null]
12	17	New Mega Abobus Show	[null]	[null]	[null]
13	2	I Want to Sleep too	[null]	[null]	[null]
14	19	Mega Abobus Show	[null]	[null]	[null]

Классика.

The screenshot shows the pgAdmin 4 interface with two tabs open: 'postgres/postgres...' and 'postgres/postgres@PostgreSQL 17*'. The second tab is active and displays a SQL query window. The query is:

```
11 -- Попытка перезаписать запись, изменяемую в 1 сессии
12 UPDATE series.shows SET title = 'Lets Try to Overwrite' WHERE id = 1;
13 -- Модифицируем данные
```

Below the query, the 'Messages' tab is selected, showing the following error messages:

```
ERROR: выполнение оператора отменено по запросу пользователя
CONTEXT: при изменении кортежа (0,24) в отношении "shows"

ОШИБКА: выполнение оператора отменено по запросу пользователя
--
```

Снова подвисание (надо было бы убрать этот пункт из плана...)

postgres/postgres... X postgres/postgres@PostgreSQL 17* X

postgres/postgres@PostgreSQL 17

No limit E

Query History

```
13 -- Модифицируем данные
14 UPDATE series.shows SET title = 'I Want to Sleep too' WHERE id = 3;
15 INSERT INTO series.shows (title) VALUES ('New Mega Abobus Show');
16 SELECT * FROM series.shows;
17
```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	4	Friends	[null]	[null]	[null]
2	5	The Mandalorian	[null]	[null]	[null]
3	6	New Show	[null]	[null]	[null]
4	7	Dev Show	[null]	[null]	[null]
5	9	Committed Show	[null]	[null]	[null]
6	11	Extra Committed Show	[null]	[null]	[null]
7	13	Mega Extra Committed Show	[null]	[null]	[null]
8	15	Mega Abobus Show	[null]	[null]	[null]
9	1	I Want to Sleep	[null]	[null]	[null]
10	16	Committed Abobus Show	[null]	[null]	[null]
11	17	New Mega Abobus Show	[null]	[null]	[null]
12	2	I Want to Sleep too	[null]	[null]	[null]
13	19	Mega Abobus Show	[null]	[null]	[null]
14	3	I Want to Sleep too	[null]	[null]	[null]
15	21	New Mega Abobus Show	[null]	[null]	[null]

Насыпим данных.

postgres/postgres@PostgreSQL 17* × postgres/postgres... ×

postgres/postgres@PostgreSQL 17 No limit

Query History

```
19 -- Чо там во второй сессии поменялось?
20 SELECT * FROM series.shows;
```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	4	Friends	[null]	[null]	[null]
2	5	The Mandalorian	[null]	[null]	[null]
3	6	New Show	[null]	[null]	[null]
4	7	Dev Show	[null]	[null]	[null]
5	9	Committed Show	[null]	[null]	[null]
6	11	Extra Committed Show	[null]	[null]	[null]
7	13	Mega Extra Committed Show	[null]	[null]	[null]
8	15	Mega Abobus Show	[null]	[null]	[null]
9	16	Committed Abobus Show	[null]	[null]	[null]
10	17	New Mega Abobus Show	[null]	[null]	[null]
11	2	I Want to Sleep too	[null]	[null]	[null]
12	19	Mega Abobus Show	[null]	[null]	[null]
13	1	I Want to Sleep	[null]	[null]	[null]
14	20	Committed Abobus Show	[null]	[null]	[null]
15	3	I Want to Sleep too	[null]	[null]	[null]
16	21	New Mega Abobus Show	[null]	[null]	[null]

Все как обы... нет! Фантомы? Чет я запутался... Когда они возникают?

postgres/postgres@PostgreSQL 17* X postgres/postgres... X

postgres/postgres@PostgreSQL 17

No limit

Query History

```
20 SELECT * FROM series.shows;
21 COMMIT;
22 -- Что после коммита?
23 SELECT * FROM series.shows;
24
```

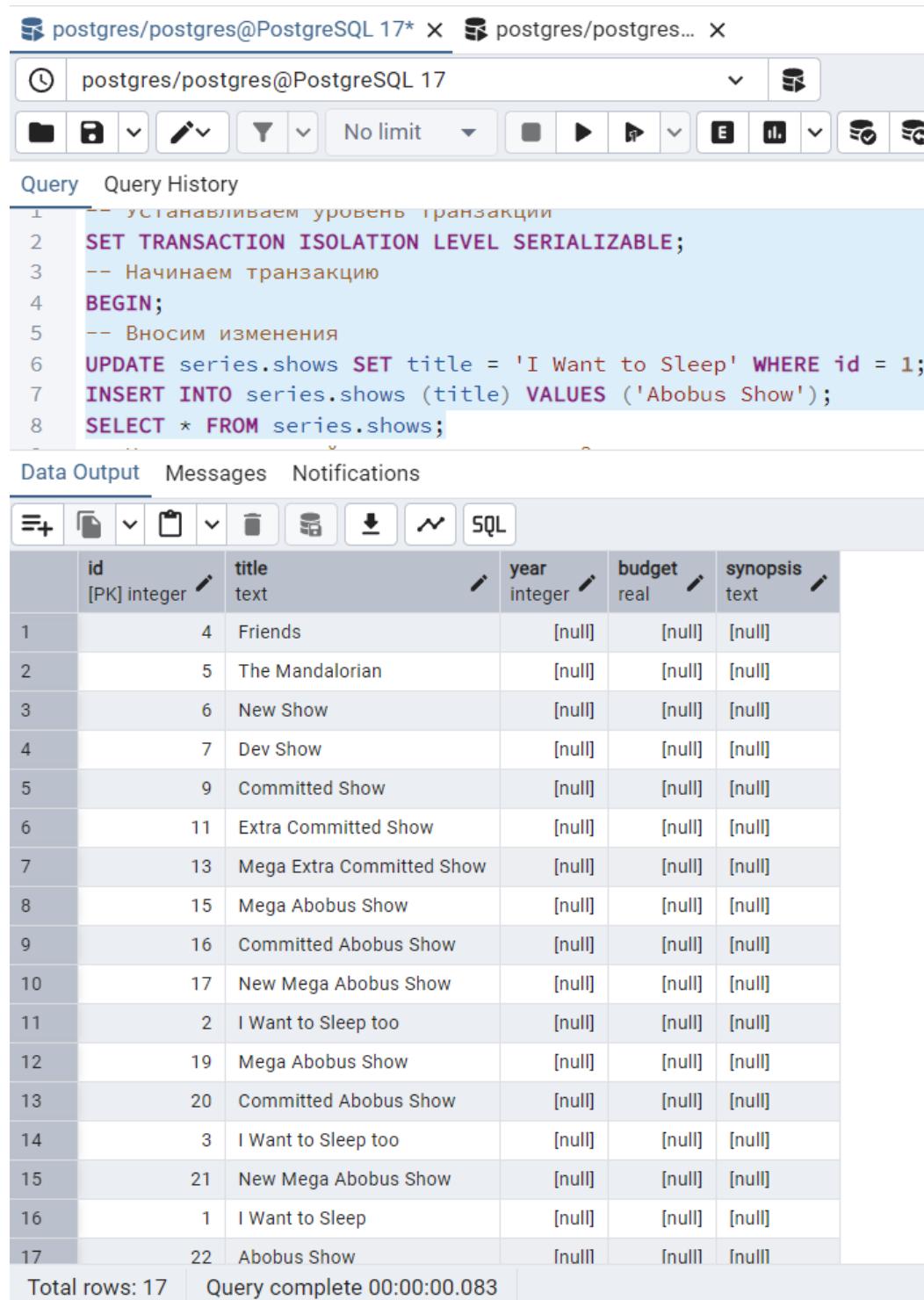
Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	4	Friends	[null]	[null]	[null]
2	5	The Mandalorian	[null]	[null]	[null]
3	6	New Show	[null]	[null]	[null]
4	7	Dev Show	[null]	[null]	[null]
5	9	Committed Show	[null]	[null]	[null]
6	11	Extra Committed Show	[null]	[null]	[null]
7	13	Mega Extra Committed Show	[null]	[null]	[null]
8	15	Mega Abobus Show	[null]	[null]	[null]
9	16	Committed Abobus Show	[null]	[null]	[null]
10	17	New Mega Abobus Show	[null]	[null]	[null]
11	2	I Want to Sleep too	[null]	[null]	[null]
12	19	Mega Abobus Show	[null]	[null]	[null]
13	1	I Want to Sleep	[null]	[null]	[null]
14	20	Committed Abobus Show	[null]	[null]	[null]
15	3	I Want to Sleep too	[null]	[null]	[null]
16	21	New Mega Abobus Show	[null]	[null]	[null]

Ожидаемо.

3.3 SERIALIZABLE



The screenshot shows the pgAdmin 4 interface with two tabs: 'postgres/postgres@PostgreSQL 17*' and 'postgres/postgres...'. The top bar includes a clock icon, a dropdown for session name, and various toolbar icons. The main area has tabs for 'Query' (selected), 'Query History', and 'Data Output'. The 'Query' tab contains the following SQL code:

```
-- устанавливаем уровень транзакции
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
-- Начинаем транзакцию
BEGIN;
-- Вносим изменения
UPDATE series.shows SET title = 'I Want to Sleep' WHERE id = 1;
INSERT INTO series.shows (title) VALUES ('Abobus Show');
SELECT * FROM series.shows;
```

The 'Data Output' tab displays a table with the following data:

	id [PK] integer	title text	year integer	budget real	synopsis text
1	4	Friends	[null]	[null]	[null]
2	5	The Mandalorian	[null]	[null]	[null]
3	6	New Show	[null]	[null]	[null]
4	7	Dev Show	[null]	[null]	[null]
5	9	Committed Show	[null]	[null]	[null]
6	11	Extra Committed Show	[null]	[null]	[null]
7	13	Mega Extra Committed Show	[null]	[null]	[null]
8	15	Mega Abobus Show	[null]	[null]	[null]
9	16	Committed Abobus Show	[null]	[null]	[null]
10	17	New Mega Abobus Show	[null]	[null]	[null]
11	2	I Want to Sleep too	[null]	[null]	[null]
12	19	Mega Abobus Show	[null]	[null]	[null]
13	20	Committed Abobus Show	[null]	[null]	[null]
14	3	I Want to Sleep too	[null]	[null]	[null]
15	21	New Mega Abobus Show	[null]	[null]	[null]
16	1	I Want to Sleep	[null]	[null]	[null]
17	22	Abobus Show	[null]	[null]	[null]

Total rows: 17 Query complete 00:00:00.083

Пока все OK.

postgres/postgres... X postgres/postgres@PostgreSQL 17* X

postgres/postgres@PostgreSQL 17

No limit

Query History

```
1 -- Состояние из второй сессии
2 SELECT * FROM series.shows;
3 Правило: Проверка состояния команды в 1 сессии
```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	4	Friends	[null]	[null]	[null]
2	5	The Mandalorian	[null]	[null]	[null]
3	6	New Show	[null]	[null]	[null]
4	7	Dev Show	[null]	[null]	[null]
5	9	Committed Show	[null]	[null]	[null]
6	11	Extra Committed Show	[null]	[null]	[null]
7	13	Mega Extra Committed Show	[null]	[null]	[null]
8	15	Mega Abobus Show	[null]	[null]	[null]
9	16	Committed Abobus Show	[null]	[null]	[null]
10	17	New Mega Abobus Show	[null]	[null]	[null]
11	2	I Want to Sleep too	[null]	[null]	[null]
12	19	Mega Abobus Show	[null]	[null]	[null]
13	1	I Want to Sleep	[null]	[null]	[null]
14	20	Committed Abobus Show	[null]	[null]	[null]
15	3	I Want to Sleep too	[null]	[null]	[null]
16	21	New Mega Abobus Show	[null]	[null]	[null]

Не видим изменений из транзакции.

postgres/postgres... X postgres/postgres@PostgreSQL 17* X

postgres/postgres@PostgreSQL 17 No limit

Query History

```
3 -- Попытка перезаписать запись, изменяемую в 1 сессии
4 UPDATE series.shows SET title = 'Lets Try to Overwrite' WHERE id = 1;
```

Data Output Messages Notifications

	id [PK] integer	title text	year integer	budget real	synopsis text
1	4	Friends	[null]	[null]	[null]
2	5	The Mandalorian	[null]	[null]	[null]
3	6	New Show	[null]	[null]	[null]
4	7	Dev Show	[null]	[null]	[null]
5	9	Committed Show	[null]	[null]	[null]
6	11	Extra Committed Show	[null]	[null]	[null]
7	13	Mega Extra Committed Show	[null]	[null]	[null]
8	15	Mega Abobus Show	[null]	[null]	[null]
9	16	Committed Abobus Show	[null]	[null]	[null]
10	17	New Mega Abobus Show	[null]	[null]	[null]
11	2	I Want to Sleep too	[null]	[null]	[null]

Waiting for the query to complete...

Подвисание. Для разнообразия заскринил до прерывания.

postgres/postgres... X postgres/postgres@PostgreSQL 17* X

postgres/postgres@PostgreSQL 17

No limit

Query History

```
5 -- Модифицируем данные
6 UPDATE series.shows SET title = 'I Want to Sleep too' WHERE id = 2;
7 INSERT INTO series.shows (title) VALUES ('Mega Abobus Show');
8 SELECT * FROM series.shows;
```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	4	Friends	[null]	[null]	[null]
2	5	The Mandalorian	[null]	[null]	[null]
3	6	New Show	[null]	[null]	[null]
4	7	Dev Show	[null]	[null]	[null]
5	9	Committed Show	[null]	[null]	[null]
6	11	Extra Committed Show	[null]	[null]	[null]
7	13	Mega Extra Committed Show	[null]	[null]	[null]
8	15	Mega Abobus Show	[null]	[null]	[null]
9	16	Committed Abobus Show	[null]	[null]	[null]
10	17	New Mega Abobus Show	[null]	[null]	[null]
11	19	Mega Abobus Show	[null]	[null]	[null]
12	1	I Want to Sleep	[null]	[null]	[null]
13	20	Committed Abobus Show	[null]	[null]	[null]
14	3	I Want to Sleep too	[null]	[null]	[null]
15	21	New Mega Abobus Show	[null]	[null]	[null]
16	2	I Want to Sleep too	[null]	[null]	[null]
17	23	Mega Abobus Show	[null]	[null]	[null]

Добавили данных.

postgres/postgres@PostgreSQL 17* X postgres/postgres... X

(1) postgres/postgres@PostgreSQL 17

No limit

Query History

9 -- Чо там во второй сессии поменялось?

10 SELECT * FROM series.shows;

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	4	Friends	[null]	[null]	[null]
2	5	The Mandalorian	[null]	[null]	[null]
3	6	New Show	[null]	[null]	[null]
4	7	Dev Show	[null]	[null]	[null]
5	9	Committed Show	[null]	[null]	[null]
6	11	Extra Committed Show	[null]	[null]	[null]
7	13	Mega Extra Committed Show	[null]	[null]	[null]
8	15	Mega Abobus Show	[null]	[null]	[null]
9	16	Committed Abobus Show	[null]	[null]	[null]
10	17	New Mega Abobus Show	[null]	[null]	[null]
11	2	I Want to Sleep too	[null]	[null]	[null]
12	19	Mega Abobus Show	[null]	[null]	[null]
13	20	Committed Abobus Show	[null]	[null]	[null]
14	3	I Want to Sleep too	[null]	[null]	[null]
15	21	New Mega Abobus Show	[null]	[null]	[null]
16	1	I Want to Sleep	[null]	[null]	[null]
17	22	Abobus Show	[null]	[null]	[null]

Не видим новых данных из второй сессии.

postgres/postgres@PostgreSQL 17* × postgres/postgres... ×

postgres/postgres@PostgreSQL 17

No limit

Query History

```

11 -- Откатываем изменения
12 ROLLBACK;
13 SELECT * FROM series.shows;

```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	4	Friends	[null]	[null]	[null]
2	5	The Mandalorian	[null]	[null]	[null]
3	6	New Show	[null]	[null]	[null]
4	7	Dev Show	[null]	[null]	[null]
5	9	Committed Show	[null]	[null]	[null]
6	11	Extra Committed Show	[null]	[null]	[null]
7	13	Mega Extra Committed Show	[null]	[null]	[null]
8	15	Mega Abobus Show	[null]	[null]	[null]
9	16	Committed Abobus Show	[null]	[null]	[null]
10	17	New Mega Abobus Show	[null]	[null]	[null]
11	19	Mega Abobus Show	[null]	[null]	[null]
12	1	I Want to Sleep	[null]	[null]	[null]
13	20	Committed Abobus Show	[null]	[null]	[null]
14	3	I Want to Sleep too	[null]	[null]	[null]
15	21	New Mega Abobus Show	[null]	[null]	[null]
16	2	I Want to Sleep too	[null]	[null]	[null]
17	23	Mega Abobus Show	[null]	[null]	[null]

Теперь видим... Но какой ценой? Ценой своих данных 😞

postgres/postgres@PostgreSQL 17* × postgres/postgres... ×

postgres/postgres@PostgreSQL 17 No limit

Query History

```
-- Вторая транзакция
BEGIN;
UPDATE series.shows SET title = 'I Want to Sleep' WHERE id = 1;
INSERT INTO series.shows (title) VALUES ('Committed Abobus Show');
SELECT * FROM series.shows;
```

Data Output Messages Notifications

	id [PK] integer	title text	year integer	budget real	synopsis text
1	4	Friends	[null]	[null]	[null]
2	5	The Mandalorian	[null]	[null]	[null]
3	6	New Show	[null]	[null]	[null]
4	7	Dev Show	[null]	[null]	[null]
5	9	Committed Show	[null]	[null]	[null]
6	11	Extra Committed Show	[null]	[null]	[null]
7	13	Mega Extra Committed Show	[null]	[null]	[null]
8	15	Mega Abobus Show	[null]	[null]	[null]
9	16	Committed Abobus Show	[null]	[null]	[null]
10	17	New Mega Abobus Show	[null]	[null]	[null]
11	19	Mega Abobus Show	[null]	[null]	[null]
12	20	Committed Abobus Show	[null]	[null]	[null]
13	3	I Want to Sleep too	[null]	[null]	[null]
14	21	New Mega Abobus Show	[null]	[null]	[null]
15	2	I Want to Sleep too	[null]	[null]	[null]
16	23	Mega Abobus Show	[null]	[null]	[null]
17	1	I Want to Sleep	[null]	[null]	[null]
18	24	Committed Abobus Show	[null]	[null]	[null]

Новая транзакция.

postgres/postgres... X postgres/postgres@PostgreSQL 17* X

postgres/postgres@PostgreSQL 17 No limit

Query History

```
9 -- Состояние из второй сессии
10 SELECT * FROM series.shows;
```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	4	Friends	[null]	[null]	[null]
2	5	The Mandalorian	[null]	[null]	[null]
3	6	New Show	[null]	[null]	[null]
4	7	Dev Show	[null]	[null]	[null]
5	9	Committed Show	[null]	[null]	[null]
6	11	Extra Committed Show	[null]	[null]	[null]
7	13	Mega Extra Committed Show	[null]	[null]	[null]
8	15	Mega Abobus Show	[null]	[null]	[null]
9	16	Committed Abobus Show	[null]	[null]	[null]
10	17	New Mega Abobus Show	[null]	[null]	[null]
11	19	Mega Abobus Show	[null]	[null]	[null]
12	1	I Want to Sleep	[null]	[null]	[null]
13	20	Committed Abobus Show	[null]	[null]	[null]
14	3	I Want to Sleep too	[null]	[null]	[null]
15	21	New Mega Abobus Show	[null]	[null]	[null]
16	2	I Want to Sleep too	[null]	[null]	[null]
17	23	Mega Abobus Show	[null]	[null]	[null]

Все ожидаемо.

postgres/postgres... X postgres/postgres@PostgreSQL 17* X

postgres/postgres@PostgreSQL 17

No limit ▾

Query History

```
11 -- Попытка перезаписать запись, изменяемую в 1 сессии
12 UPDATE series.shows SET title = 'Lets Try to Overwrite' WHERE id = 1;
```

Data Output Messages Notifications

ERROR: выполнение оператора отменено по запросу пользователя
CONTEXT: при изменении кортежа (0,32) в отношении "shows"

ОШИБКА: выполнение оператора отменено по запросу пользователя
SQL state: 57014

Подвисание.

postgres/postgres... X postgres/postgres@PostgreSQL 17* X

postgres/postgres@PostgreSQL 17

No limit

Query History

```

13 -- Модифицируем данные
14 UPDATE series.shows SET title = 'I Want to Sleep too' WHERE id = 3;
15 INSERT INTO series.shows (title) VALUES ('New Mega Abobus Show');
16 SELECT * FROM series.shows;
17

```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	4	Friends	[null]	[null]	[null]
2	5	The Mandalorian	[null]	[null]	[null]
3	6	New Show	[null]	[null]	[null]
4	7	Dev Show	[null]	[null]	[null]
5	9	Committed Show	[null]	[null]	[null]
6	11	Extra Committed Show	[null]	[null]	[null]
7	13	Mega Extra Committed Show	[null]	[null]	[null]
8	15	Mega Abobus Show	[null]	[null]	[null]
9	16	Committed Abobus Show	[null]	[null]	[null]
10	17	New Mega Abobus Show	[null]	[null]	[null]
11	19	Mega Abobus Show	[null]	[null]	[null]
12	1	I Want to Sleep	[null]	[null]	[null]
13	20	Committed Abobus Show	[null]	[null]	[null]
14	21	New Mega Abobus Show	[null]	[null]	[null]
15	2	I Want to Sleep too	[null]	[null]	[null]
16	23	Mega Abobus Show	[null]	[null]	[null]
17	3	I Want to Sleep too	[null]	[null]	[null]
18	25	New Mega Abobus Show	[null]	[null]	[null]

Докинем данных.

postgres/postgres@PostgreSQL 17* × postgres/postgres... ×

postgres/postgres@PostgreSQL 17 No limit

Query History

```

19 -- Чо там во второй сессии поменялось?
20 SELECT * FROM series.shows;

```

Data Output Messages Notifications

	id [PK] integer	title text	year integer	budget real	synopsis text
1	4	Friends	[null]	[null]	[null]
2	5	The Mandalorian	[null]	[null]	[null]
3	6	New Show	[null]	[null]	[null]
4	7	Dev Show	[null]	[null]	[null]
5	9	Committed Show	[null]	[null]	[null]
6	11	Extra Committed Show	[null]	[null]	[null]
7	13	Mega Extra Committed Show	[null]	[null]	[null]
8	15	Mega Abobus Show	[null]	[null]	[null]
9	16	Committed Abobus Show	[null]	[null]	[null]
10	17	New Mega Abobus Show	[null]	[null]	[null]
11	19	Mega Abobus Show	[null]	[null]	[null]
12	20	Committed Abobus Show	[null]	[null]	[null]
13	21	New Mega Abobus Show	[null]	[null]	[null]
14	2	I Want to Sleep too	[null]	[null]	[null]
15	23	Mega Abobus Show	[null]	[null]	[null]
16	1	I Want to Sleep	[null]	[null]	[null]
17	24	Committed Abobus Show	[null]	[null]	[null]
18	3	I Want to Sleep too	[null]	[null]	[null]
19	25	New Mega Abobus Show	[null]	[null]	[null]

Чо?! Почему они сюда попали? Там же изоляция... O_o

postgres/postgres@PostgreSQL 17* × postgres/postgres... ×

postgres/postgres@PostgreSQL 17

No limit

Query History

```
20 SELECT * FROM series.shows;
21 COMMIT;
22 -- Что после коммита?
23 SELECT * FROM series.shows;
```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	4	Friends	[null]	[null]	[null]
2	5	The Mandalorian	[null]	[null]	[null]
3	6	New Show	[null]	[null]	[null]
4	7	Dev Show	[null]	[null]	[null]
5	9	Committed Show	[null]	[null]	[null]
6	11	Extra Committed Show	[null]	[null]	[null]
7	13	Mega Extra Committed Show	[null]	[null]	[null]
8	15	Mega Abobus Show	[null]	[null]	[null]
9	16	Committed Abobus Show	[null]	[null]	[null]
10	17	New Mega Abobus Show	[null]	[null]	[null]
11	19	Mega Abobus Show	[null]	[null]	[null]
12	20	Committed Abobus Show	[null]	[null]	[null]
13	21	New Mega Abobus Show	[null]	[null]	[null]
14	2	I Want to Sleep too	[null]	[null]	[null]
15	23	Mega Abobus Show	[null]	[null]	[null]
16	1	I Want to Sleep	[null]	[null]	[null]
17	24	Committed Abobus Show	[null]	[null]	[null]
18	3	I Want to Sleep too	[null]	[null]	[null]
19	25	New Mega Abobus Show	[null]	[null]	[null]

Тут все ожидаемо.

Подытоживая результаты трех экспериментов, замечу: различие между экспериментами 3.1 и 3.2 соответствовали ожиданию; ожидал фантомов в обеих транзакциях эксперимента 3.2, но почему-то поймал их только в одной; не ожидал фантомов в эксперименте 3.3, но почему-то поймал в одной из транзакций... Таким образом, эксперименты 3.2 и 3.3 прошли абсолютно одинаково, чего я не ожидал... Причиной может быть случайный пропуск или опечатка в каком-то из запросов (таковых я не нашел...).

4. Транзакция во второй сессии.

Немного изменим дизайн экспериментов 3.1-3.3: теперь во второй сессии будет запущена транзакция на уровне SERIALIZABLE. Также уберем запрос с подвисанием и перед каждым из экспериментов будем сбрасывать данные из таблицы:

```
-- Удалить все добавленные
DELETE FROM series.shows
WHERE id > 5;

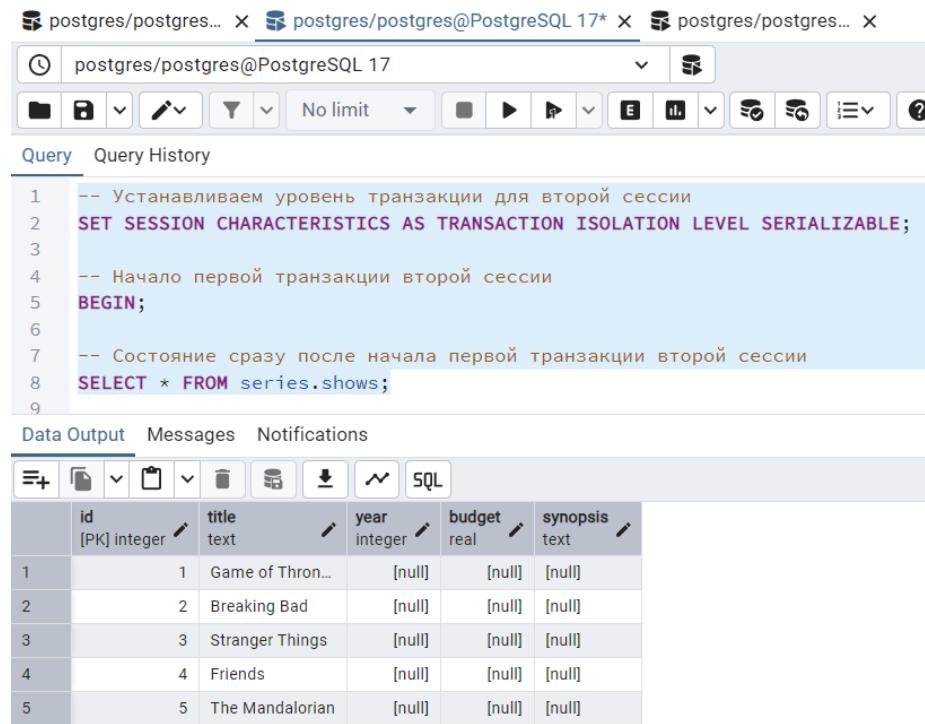
-- Восстановить измененные
UPDATE series.shows SET title = 'Game of Thrones' WHERE
id = 1;
UPDATE series.shows SET title = 'Breaking Bad' WHERE id =
2;
UPDATE series.shows SET title = 'Stranger Things' WHERE
id = 3;
UPDATE series.shows SET title = 'Friends' WHERE id = 4;
UPDATE series.shows SET title = 'The Mandalorian' WHERE
id = 5;

-- Результат
SELECT * FROM series.shows;
```

The screenshot shows the pgAdmin 4 interface. At the top, there are three tabs: 'postgres/postgres...' (active), 'postgres/postgres...' (disabled), and 'postgres/postgres@PostgreSQL 17*' (disabled). Below the tabs is a toolbar with various icons. The main area is divided into two sections: 'Query' (active) and 'Query History'. The 'Query' section contains the provided SQL script. The 'Data Output' section shows a table with the following data:

	id	[PK] integer	title	text	year	integer	budget	real	synopsis	text
1			1	Game of Thron...		[null]		[null]		[null]
2			2	Breaking Bad		[null]		[null]		[null]
3			3	Stranger Things		[null]		[null]		[null]
4			4	Friends		[null]		[null]		[null]
5			5	The Mandalorian		[null]		[null]		[null]

4.1 READ COMMITTED



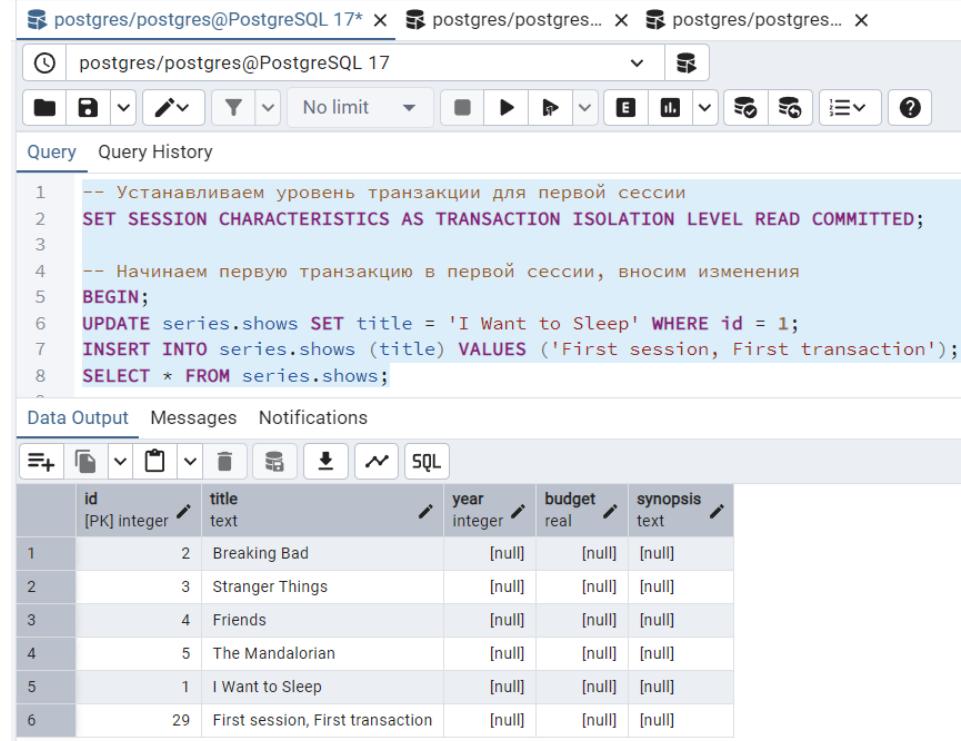
The screenshot shows the pgAdmin 4 interface with three tabs at the top: 'postgres/postgres...' (closed), 'postgres/postgres@PostgreSQL 17*' (active), and 'postgres/postgres...' (closed). The toolbar below has icons for connection, refresh, search, and various operations. The main area is titled 'Query' and contains the following SQL code:

```
1 -- Устанавливаем уровень транзакции для второй сессии
2 SET SESSION CHARACTERISTICS AS TRANSACTION ISOLATION LEVEL SERIALIZABLE;
3
4 -- Начало первой транзакции второй сессии
5 BEGIN;
6
7 -- Состояние сразу после начала первой транзакции второй сессии
8 SELECT * FROM series.shows;
9
```

Below the query results, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The data output table has columns: id [PK] integer, title text, year integer, budget real, synopsis text. The data is:

	id [PK] integer	title	year	budget	synopsis
1	1	Game of Thron...	[null]	[null]	[null]
2	2	Breaking Bad	[null]	[null]	[null]
3	3	Stranger Things	[null]	[null]	[null]
4	4	Friends	[null]	[null]	[null]
5	5	The Mandalorian	[null]	[null]	[null]

Пока что состояние – исходное.



The screenshot shows the pgAdmin 4 interface with three tabs at the top: 'postgres/postgres@PostgreSQL 17*' (active), 'postgres/postgres...' (closed), and 'postgres/postgres...' (closed). The toolbar below has icons for connection, refresh, search, and various operations. The main area is titled 'Query' and contains the following SQL code:

```
1 -- Устанавливаем уровень транзакции для первой сессии
2 SET SESSION CHARACTERISTICS AS TRANSACTION ISOLATION LEVEL READ COMMITTED;
3
4 -- Начинаем первую транзакцию в первой сессии, вносим изменения
5 BEGIN;
6 UPDATE series.shows SET title = 'I Want to Sleep' WHERE id = 1;
7 INSERT INTO series.shows (title) VALUES ('First session, First transaction');
8 SELECT * FROM series.shows;
```

Below the query results, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The data output table has columns: id [PK] integer, title text, year integer, budget real, synopsis text. The data is:

	id [PK] integer	title	year	budget	synopsis
1	2	Breaking Bad	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	1	I Want to Sleep	[null]	[null]	[null]
6	29	First session, First transaction	[null]	[null]	[null]

Ничего необычного.

```

9
10 -- Состояние во второй сессии в первой транзакции после изменений в первой сессии
11 SELECT * FROM series.shows;
12

```

Query History

	<code>id [PK] integer</code>	<code>title</code>	<code>year</code>	<code>budget</code>	<code>synopsis</code>
1		1 Game of Thron...	[null]	[null]	[null]
2		2 Breaking Bad	[null]	[null]	[null]
3		3 Stranger Things	[null]	[null]	[null]
4		4 Friends	[null]	[null]	[null]
5		5 The Mandalorian	[null]	[null]	[null]

Изменений не видно.

```

12
13 -- Модифицируем данные во второй сессии в первой транзакции
14 UPDATE series.shows SET title = 'Updated: Second session, First transaction' WHERE id = 2;
15 INSERT INTO series.shows (title) VALUES ('Inserted: Second session, First transaction');
16 SELECT * FROM series.shows;
17

```

Query History

	<code>id [PK] integer</code>	<code>title</code>	<code>year</code>	<code>budget</code>	<code>synopsis</code>
1		1 Game of Thrones	[null]	[null]	[null]
2		3 Stranger Things	[null]	[null]	[null]
3		4 Friends	[null]	[null]	[null]
4		5 The Mandalorian	[null]	[null]	[null]
5		2 Updated: Second session, First transaction	[null]	[null]	[null]
6		30 Inserted: Second session, First transaction	[null]	[null]	[null]

Ску-ка.

postgres/postgres@PostgreSQL 17* × postgres/postgres... × postgres/postgres... ×

postgres/postgres@PostgreSQL 17 No limit

Query History

```
10 -- Состояние в первой сессии в первой транзакции после изменений во второй сессии
11 SELECT * FROM series.shows;
12
```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	2	Breaking Bad	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	1	I Want to Sleep	[null]	[null]	[null]
6	29	First session, First transaction	[null]	[null]	[null]

В отличие от эксперимента 3.1 тут мы не видим изменений из второй сессии, так как они проводятся в еще активной транзакции.

postgres/postgres@PostgreSQL 17* × postgres/postgres... × postgres/postgres... ×

postgres/postgres@PostgreSQL 17 No limit

Query History

```
13 -- Откатываем изменения первой транзакции первой сессии
14 ROLLBACK;
15 SELECT * FROM series.shows;
```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thron...	[null]	[null]	[null]
2	2	Breaking Bad	[null]	[null]	[null]
3	3	Stranger Things	[null]	[null]	[null]
4	4	Friends	[null]	[null]	[null]
5	5	The Mandalorian	[null]	[null]	[null]

Ожидаемо.

postgres/postgres... X postgres/postgres@PostgreSQL 17* X postgres/postgres... X

postgres/postgres@PostgreSQL 17 No limit

Query History

```
18 -- Состояние во второй сессии в первой транзакции после роллбэка из первой сессии
19 SELECT * FROM series.shows;
```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	2	Updated: Second session, First transacti...	[null]	[null]	[null]
6	30	Inserted: Second session, First transaction	[null]	[null]	[null]

Жоской+мощной изоляции никакие перемены ни по чём.

postgres/postgres... X postgres/postgres@PostgreSQL 17* X postgres/postgres... X

postgres/postgres@PostgreSQL 17 No limit

Query History

```
20
21 -- Состояние во второй сессии после первой транзакции
22 COMMIT;
23 SELECT * FROM series.shows;
```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	2	Updated: Second session, First transacti...	[null]	[null]	[null]
6	30	Inserted: Second session, First transaction	[null]	[null]	[null]

Пока ничего необычного.

postgres/postgres@PostgreSQL 17* | postgres/postgres... | postgres/postgres... |

postgres/postgres@PostgreSQL 17

No limit

Query History

```
17 -- Состояние в первой сессии после первого коммита во второй сессии
18 SELECT * FROM series.shows;
```

Data Output Messages Notifications

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	2	Updated: Second session, First transacti...	[null]	[null]	[null]
6	30	Inserted: Second session, First transaction	[null]	[null]	[null]

Теперь изменения видно и здесь.

postgres/postgres... | postgres/postgres@PostgreSQL 17* | postgres/postgres...

postgres/postgres@PostgreSQL 17

No limit

Query History

```
24
25 -- Начало второй транзакции второй сессии
26 BEGIN;
27
28 -- Состояние сразу после начала второй транзакции второй сессии
29 SELECT * FROM series.shows;
```

Data Output Messages Notifications

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	2	Updated: Second session, First transacti...	[null]	[null]	[null]
6	30	Inserted: Second session, First transaction	[null]	[null]	[null]

Старт новой транзакции.

```

-- Вторая транзакция первой сессии
BEGIN;
UPDATE series.shows SET title = 'I Want to Sleep' WHERE id = 3;
INSERT INTO series.shows (title) VALUES ('First session, Second transaction');
SELECT * FROM series.shows;

```

	<code>id</code> [PK] integer	<code>title</code> text	<code>year</code> integer	<code>budget</code> real	<code>synopsis</code> text
1	1	Game of Thrones	[null]	[null]	[null]
2	4	Friends	[null]	[null]	[null]
3	5	The Mandalorian	[null]	[null]	[null]
4	2	Updated: Second session, First transacti...	[null]	[null]	[null]
5	30	Inserted: Second session, First transaction	[null]	[null]	[null]
6	3	I Want to Sleep	[null]	[null]	[null]
7	31	First session, Second transaction	[null]	[null]	[null]

Старт новой транзакции в первой сессии.

```

-- Состояние во второй сессии во второй транзакции после изменений в первой сессии
SELECT * FROM series.shows;

```

	<code>id</code> [PK] integer	<code>title</code> text	<code>year</code> integer	<code>budget</code> real	<code>synopsis</code> text
1	1	Game of Thrones	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	2	Updated: Second session, First transacti...	[null]	[null]	[null]
6	30	Inserted: Second session, First transaction	[null]	[null]	[null]

И – изоляция.

```
-- Модифицируем данные во второй транзакции второй сессии
UPDATE series.shows SET title = 'Updated: Second session, Second transaction' WHERE id = 4;
INSERT INTO series.shows (title) VALUES ('Inserted: Second session, Second transaction');
SELECT * FROM series.shows;
```

Data Output Messages Notifications

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	5	The Mandalorian	[null]	[null]	[null]
4	2	Updated: Second session, First transaction	[null]	[null]	[null]
5	30	Inserted: Second session, First transaction	[null]	[null]	[null]
6	4	Updated: Second session, Second transacti...	[null]	[null]	[null]
7	32	Inserted: Second session, Second transaction	[null]	[null]	[null]

Ничо необычного.

```
-- Состояние в первой сессии во второй транзакции после изменений во второй сессии
SELECT * FROM series.shows;
```

Data Output Messages Notifications

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	4	Friends	[null]	[null]	[null]
3	5	The Mandalorian	[null]	[null]	[null]
4	2	Updated: Second session, First transacti...	[null]	[null]	[null]
5	30	Inserted: Second session, First transaction	[null]	[null]	[null]
6	3	I Want to Sleep	[null]	[null]	[null]
7	31	First session, Second transaction	[null]	[null]	[null]

Снова все обыденно.

```

28
29 -- Коммит второй транзакции первой сессии
30 COMMIT;
31 SELECT * FROM series.shows;

```

Data Output Messages Notifications

	<code>id [PK] integer</code>	<code>title text</code>	<code>year integer</code>	<code>budget real</code>	<code>synopsis text</code>
1		Game of Thrones	[null]	[null]	[null]
2		Friends	[null]	[null]	[null]
3		The Mandalorian	[null]	[null]	[null]
4		Updated: Second session, First transaction	[null]	[null]	[null]
5		Inserted: Second session, First transaction	[null]	[null]	[null]
6		I Want to Sleep	[null]	[null]	[null]
7		First session, Second transaction	[null]	[null]	[null]

У самурая нет цели – только путь.

```

39 -- Состояние во второй сессии во второй транзакции после коммита из первой сессии
40 SELECT * FROM series.shows;

```

Data Output Messages Notifications

	<code>id [PK] integer</code>	<code>title text</code>	<code>year integer</code>	<code>budget real</code>	<code>synopsis text</code>
1		Game of Thrones	[null]	[null]	[null]
2		Stranger Things	[null]	[null]	[null]
3		The Mandalorian	[null]	[null]	[null]
4		Updated: Second session, First transaction	[null]	[null]	[null]
5		Inserted: Second session, First transaction	[null]	[null]	[null]
6		Updated: Second session, Second transaction	[null]	[null]	[null]
7		Inserted: Second session, Second transaction	[null]	[null]	[null]

А здесь видно отличие выбранной степени изоляции транзакций во второй сессии от READ COMMITTED: мы не видим изменения, произошедшие вне нашей транзакции.

postgres/postgres... X postgres/postgres@PostgreSQL 17* X postgres/postgres...

postgres/postgres@PostgreSQL 17

No limit

Query History

```

42 -- Состояние из второй сессии после второй транзакции
43 COMMIT;
44 SELECT * FROM series.shows;
45

```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	5	The Mandalorian	[null]	[null]	[null]
3	2	Updated: Second session, First transaction	[null]	[null]	[null]
4	30	Inserted: Second session, First transaction	[null]	[null]	[null]
5	3	I Want to Sleep	[null]	[null]	[null]
6	31	First session, Second transaction	[null]	[null]	[null]
7	4	Updated: Second session, Second transaction	[null]	[null]	[null]
8	32	Inserted: Second session, Second transaction	[null]	[null]	[null]

А вот теперь видно изменения из первой сессии.

postgres/postgres@PostgreSQL 17* X postgres/postgres... X postgres/postgres...

postgres/postgres@PostgreSQL 17

No limit

Query History

```

32 -- Состояние в первой сессии после второго коммита во второй сессии
33
34 SELECT * FROM series.shows;

```

Data Output Messages Notifications

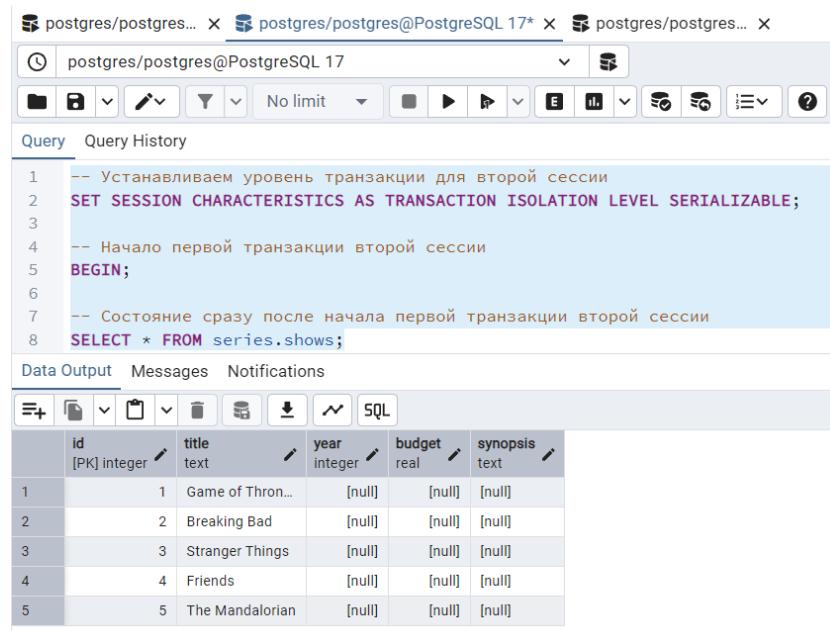
SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	5	The Mandalorian	[null]	[null]	[null]
3	2	Updated: Second session, First transaction	[null]	[null]	[null]
4	30	Inserted: Second session, First transaction	[null]	[null]	[null]
5	3	I Want to Sleep	[null]	[null]	[null]
6	31	First session, Second transaction	[null]	[null]	[null]
7	4	Updated: Second session, Second transaction	[null]	[null]	[null]
8	32	Inserted: Second session, Second transaction	[null]	[null]	[null]

Теперь состояния БД совпадают для обеих сессий.

За кадром сбрасываю состояние таблицы.

4.2 REPEATABLE READ



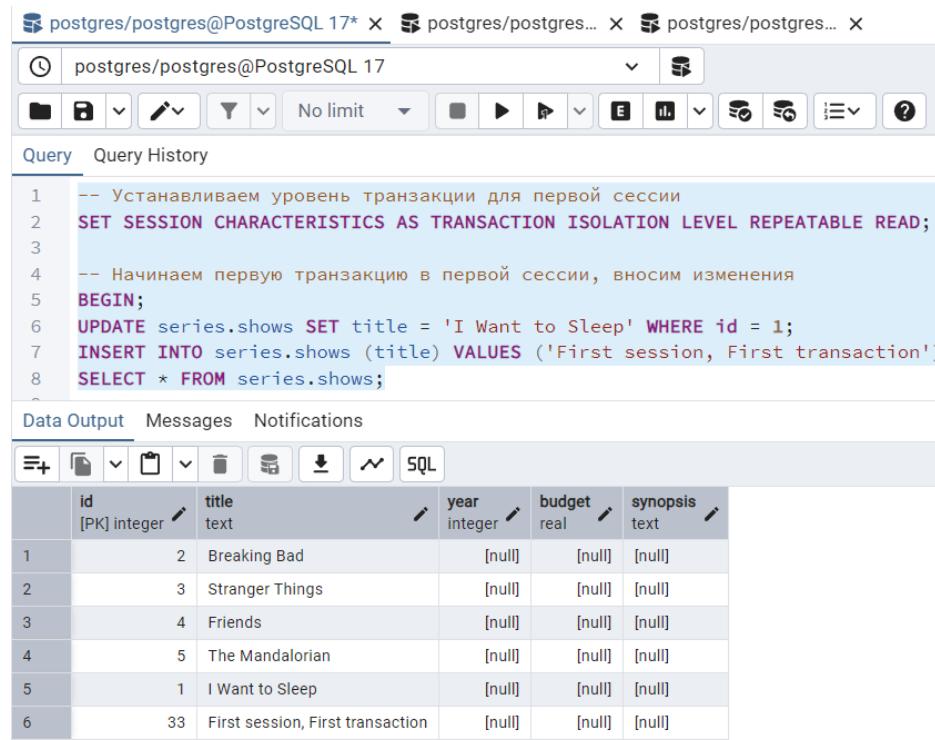
```
-- Устанавливаем уровень транзакции для второй сессии
SET SESSION CHARACTERISTICS AS TRANSACTION ISOLATION LEVEL SERIALIZABLE;

-- Начало первой транзакции второй сессии
BEGIN;

-- Состояние сразу после начала первой транзакции второй сессии
SELECT * FROM series.shows;
```

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thron...	[null]	[null]	[null]
2	2	Breaking Bad	[null]	[null]	[null]
3	3	Stranger Things	[null]	[null]	[null]
4	4	Friends	[null]	[null]	[null]
5	5	The Mandalorian	[null]	[null]	[null]

Все штатно.



```
-- Устанавливаем уровень транзакции для первой сессии
SET SESSION CHARACTERISTICS AS TRANSACTION ISOLATION LEVEL REPEATABLE READ;

-- Начинаем первую транзакцию в первой сессии, вносим изменения
BEGIN;
UPDATE series.shows SET title = 'I Want to Sleep' WHERE id = 1;
INSERT INTO series.shows (title) VALUES ('First session, First transaction');
SELECT * FROM series.shows;
```

	id [PK] integer	title text	year integer	budget real	synopsis text
1	2	Breaking Bad	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	1	I Want to Sleep	[null]	[null]	[null]
6	33	First session, First transaction	[null]	[null]	[null]

Ожидаемо.

```

-- Состояние во второй сессии в первой транзакции после изменений в первой сессии
SELECT * FROM series.shows;

```

	id [PK] integer	title text	year integer	budget real	synopsis text
1		1 Game of Thron...	[null]	[null]	[null]
2		2 Breaking Bad	[null]	[null]	[null]
3		3 Stranger Things	[null]	[null]	[null]
4		4 Friends	[null]	[null]	[null]
5		5 The Mandalorian	[null]	[null]	[null]

Не видим изменений.

```

-- Модифицируем данные во второй сессии в первой транзакции
UPDATE series.shows SET title = 'Updated: Second session, First transaction' WHERE id = 2;
INSERT INTO series.shows (title) VALUES ('Inserted: Second session, First transaction');
SELECT * FROM series.shows;

```

	id [PK] integer	title text	year integer	budget real	synopsis text
1		Game of Thrones	[null]	[null]	[null]
2		Stranger Things	[null]	[null]	[null]
3		Friends	[null]	[null]	[null]
4		The Mandalorian	[null]	[null]	[null]
5		Updated: Second session, First transaction	[null]	[null]	[null]
6		Inserted: Second session, First transaction	[null]	[null]	[null]

Добавим своих.

```

-- Состояние в первой сессии в первой транзакции после изменений во второй сессии
SELECT * FROM series.shows;

```

	id [PK] integer	title text	year integer	budget real	synopsis text
1	2	Breaking Bad	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	1	I Want to Sleep	[null]	[null]	[null]
6	33	First session, First transaction	[null]	[null]	[null]

Не видим изменений второй сессии, потому что транзакция текущей сессии – репитбл, а транзакция второй – сериалайзбл.

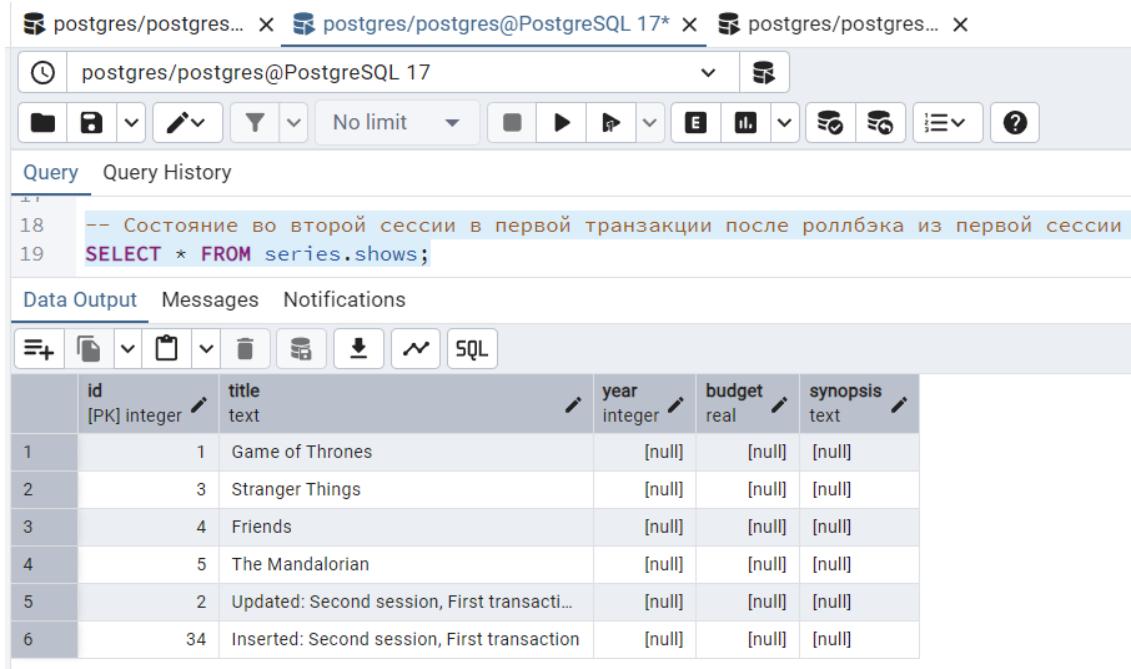
```

-- Откатываем изменения первой транзакции первой сессии
ROLLBACK;
SELECT * FROM series.shows;

```

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	2	Breaking Bad	[null]	[null]	[null]
3	3	Stranger Things	[null]	[null]	[null]
4	4	Friends	[null]	[null]	[null]
5	5	The Mandalorian	[null]	[null]	[null]

После отката не видим изменения второй сессии, так как они – в транзакции.



postgres/postgres... X postgres/postgres@PostgreSQL 17* X postgres/postgres... X

Query History

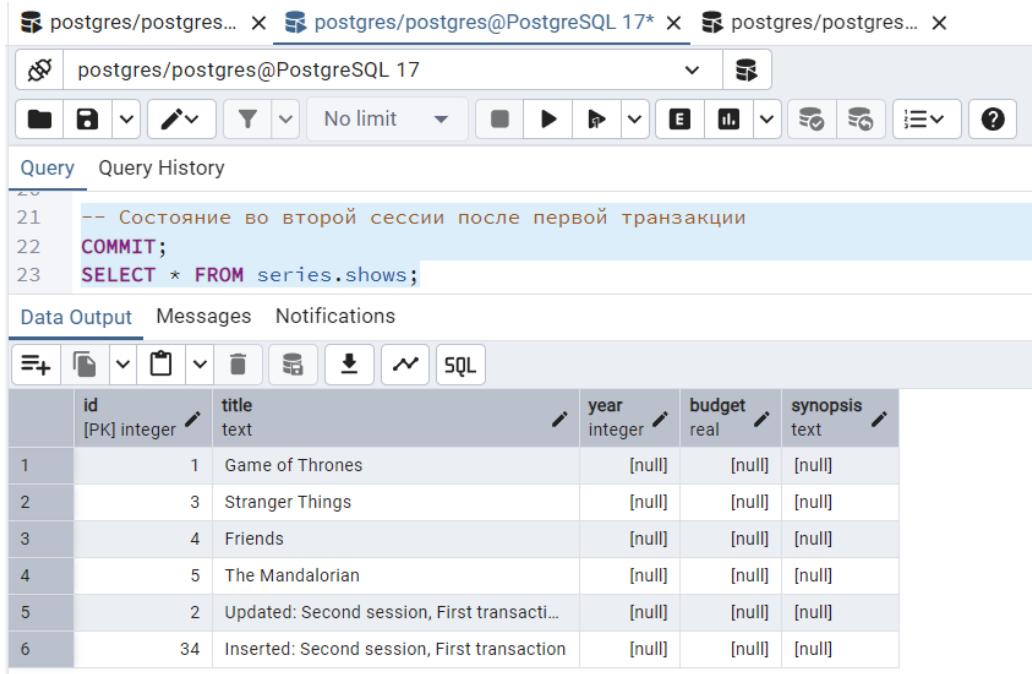
```
18 -- Состояние во второй сессии в первой транзакции после роллбэка из первой сессии
19 SELECT * FROM series.shows;
```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	2	Updated: Second session, First transacti...	[null]	[null]	[null]
6	34	Inserted: Second session, First transaction	[null]	[null]	[null]

Состояние в первой сессии не влияет на состояние во второй, так как мы – в транзакции.



postgres/postgres... X postgres/postgres@PostgreSQL 17* X postgres/postgres... X

Query History

```
20 -- Состояние во второй сессии после первой транзакции
21 COMMIT;
22
23 SELECT * FROM series.shows;
```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	2	Updated: Second session, First transacti...	[null]	[null]	[null]
6	34	Inserted: Second session, First transaction	[null]	[null]	[null]

Так как в первой сессии был откат, даже после коммита во второй сессии не видим изменений из первой.

postgres/postgres@PostgreSQL 17* × postgres/postgres@PostgreSQL 17 × postgres/postgres... ×

postgres/postgres@PostgreSQL 17 No limit

Query History

```
17 -- Состояние в первой сессии после первого коммита во второй сессии
18 SELECT * FROM series.shows;
```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	2	Updated: Second session, First transacti...	[null]	[null]	[null]
6	34	Inserted: Second session, First transaction	[null]	[null]	[null]

Ну а здесь теперь видно изменения из второй сессии.

postgres/postgres... × postgres/postgres@PostgreSQL 17* × postgres/postgres... ×

postgres/postgres@PostgreSQL 17 No limit

Query History

```
25 -- Начало второй транзакции второй сессии
26 BEGIN;
27
28 -- Состояние сразу после начала второй транзакции второй сессии
29 SELECT * FROM series.shows;
```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	2	Updated: Second session, First transacti...	[null]	[null]	[null]
6	34	Inserted: Second session, First transaction	[null]	[null]	[null]

Ничего примечательного.

```
-- Вторая транзакция первой сессии
BEGIN;
UPDATE series.shows SET title = 'I Want to Sleep' WHERE id = 3;
INSERT INTO series.shows (title) VALUES ('First session, Second transaction');
SELECT * FROM series.shows;
```

	<code>id</code> [PK] integer	<code>title</code> text	<code>year</code> integer	<code>budget</code> real	<code>synopsis</code> text
1	1	Game of Thrones	[null]	[null]	[null]
2	4	Friends	[null]	[null]	[null]
3	5	The Mandalorian	[null]	[null]	[null]
4	2	Updated: Second session, First transacti...	[null]	[null]	[null]
5	34	Inserted: Second session, First transaction	[null]	[null]	[null]
6	3	I Want to Sleep	[null]	[null]	[null]
7	35	First session, Second transaction	[null]	[null]	[null]

Просто добавим изменений.

```
-- Состояние во второй сессии во второй транзакции после изменений в первой сессии
SELECT * FROM series.shows;
```

	<code>id</code> [PK] integer	<code>title</code> text	<code>year</code> integer	<code>budget</code> real	<code>synopsis</code> text
1	1	Game of Thrones	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	2	Updated: Second session, First transacti...	[null]	[null]	[null]
6	34	Inserted: Second session, First transaction	[null]	[null]	[null]

Из-за изоляции транзакций не видим изменения из первой сессии.

postgres/postgres... X postgres/postgres@PostgreSQL 17* X postgres/postgres... X

postgres/postgres@PostgreSQL 17

No limit ▾

Query History

```

34 -- Модифицируем данные во второй транзакции второй сессии
35 UPDATE series.shows SET title = 'Updated: Second session, Second transaction' WHERE id = 4;
36 INSERT INTO series.shows (title) VALUES ('Inserted: Second session, Second transaction');
37 SELECT * FROM series.shows;

```

Data Output Messages Notifications

Showing rows 1-7 (of 7)

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	5	The Mandalorian	[null]	[null]	[null]
4	2	Updated: Second session, First transaction	[null]	[null]	[null]
5	34	Inserted: Second session, First transaction	[null]	[null]	[null]
6	4	Updated: Second session, Second transaction	[null]	[null]	[null]
7	36	Inserted: Second session, Second transaction	[null]	[null]	[null]

Добавим своих изменений.

postgres/postgres@PostgreSQL 17* X postgres/postgres... X postgres/postgres... X

postgres/postgres@PostgreSQL 17

No limit ▾

Query History

```

26 -- Состояние в первой сессии во второй транзакции после изменений во второй сессии
27 SELECT * FROM series.shows;

```

Data Output Messages Notifications

Showing rows 1-7 (of 7)

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	4	Friends	[null]	[null]	[null]
3	5	The Mandalorian	[null]	[null]	[null]
4	2	Updated: Second session, First transaction	[null]	[null]	[null]
5	34	Inserted: Second session, First transaction	[null]	[null]	[null]
6	3	I Want to Sleep	[null]	[null]	[null]
7	35	First session, Second transaction	[null]	[null]	[null]

Не видим изменений второй сессии из-за изоляции транзакций.

postgres/postgres@PostgreSQL 17* × postgres/postgres... × postgres/postgres... ×

postgres/postgres@PostgreSQL 17 No limit

Query History

```

29 -- Коммит второй транзакции первой сессии
30 COMMIT;
31 SELECT * FROM series.shows;

```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	4	Friends	[null]	[null]	[null]
3	5	The Mandalorian	[null]	[null]	[null]
4	2	Updated: Second session, First transaction	[null]	[null]	[null]
5	34	Inserted: Second session, First transaction	[null]	[null]	[null]
6	3	I Want to Sleep	[null]	[null]	[null]
7	35	First session, Second transaction	[null]	[null]	[null]

И даже после коммита все равно не видим изменений, ведь во второй сессии транзакция все еще активна.

postgres/postgres... × postgres/postgres@PostgreSQL 17* × postgres/postgres... ×

postgres/postgres@PostgreSQL 17 No limit

Query History

```

38 -- Состояние во второй сессии во второй транзакции после коммита из первой с
39
40 SELECT * FROM series.shows;

```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	5	The Mandalorian	[null]	[null]	[null]
4	2	Updated: Second session, First transaction	[null]	[null]	[null]
5	34	Inserted: Second session, First transaction	[null]	[null]	[null]
6	4	Updated: Second session, Second transaction	[null]	[null]	[null]
7	36	Inserted: Second session, Second transaction	[null]	[null]	[null]

Снова изоляция транзакции не позволяет видеть изменения из первой сессии.

```

41
42 -- Состояние из второй сессии после второй транзакции
43 COMMIT;
44 SELECT * FROM series.shows;
45

```

Data Output Messages Notifications

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	5	The Mandalorian	[null]	[null]	[null]
3	2	Updated: Second session, First transaction	[null]	[null]	[null]
4	34	Inserted: Second session, First transaction	[null]	[null]	[null]
5	3	I Want to Sleep	[null]	[null]	[null]
6	35	First session, Second transaction	[null]	[null]	[null]
7	4	Updated: Second session, Second transacti...	[null]	[null]	[null]
8	36	Inserted: Second session, Second transaction	[null]	[null]	[null]

И только после коммита во второй сессии видим изменения из первой сессии (заметим, что изменения второй сессии – более поздние).

```

32
33 -- Состояние в первой сессии после второго коммита во второй сессии
34 SELECT * FROM series.shows;

```

Data Output Messages Notifications

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	5	The Mandalorian	[null]	[null]	[null]
3	2	Updated: Second session, First transaction	[null]	[null]	[null]
4	34	Inserted: Second session, First transaction	[null]	[null]	[null]
5	3	I Want to Sleep	[null]	[null]	[null]
6	35	First session, Second transaction	[null]	[null]	[null]
7	4	Updated: Second session, Second transacti...	[null]	[null]	[null]
8	36	Inserted: Second session, Second transaction	[null]	[null]	[null]

Идентичное состояние.

Снова сброс состояния за кадром.

4.3 SERIALIZABLE

В данном эксперименте не было замечено отличий в работе по сравнению с экспериментом 4.2, поэтому приведу лишь скриншоты без пояснений.

The screenshot shows the pgAdmin 4 interface with three tabs at the top: 'postgres/postgres...' (active), 'postgres/postgres@PostgreSQL 17*' (selected), and 'postgres/postgres...' (inactive). The main area is a 'Query' tab containing the following SQL code:

```
1 -- Устанавливаем уровень транзакции для второй сессии
2 SET SESSION CHARACTERISTICS AS TRANSACTION ISOLATION LEVEL SERIALIZABLE;
3
4 -- Начало первой транзакции второй сессии
5 BEGIN;
6
7 -- Состояние сразу после начала первой транзакции второй сессии
8 SELECT * FROM series.shows;
9
```

Below the code, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is selected, showing a table with five rows of data from the 'series.shows' table:

	id [PK] integer	title text	year integer	budget real	synopsis text
1		Game of Thron...	[null]	[null]	[null]
2		Breaking Bad	[null]	[null]	[null]
3		Stranger Things	[null]	[null]	[null]
4		Friends	[null]	[null]	[null]
5		The Mandalorian	[null]	[null]	[null]

The screenshot shows the pgAdmin 4 interface with three tabs at the top: 'postgres/postgres@PostgreSQL 17*' (active), 'postgres/postgres...' (selected), and 'postgres/postgres...' (inactive). The main area is a 'Query' tab containing the following SQL code:

```
1 -- Устанавливаем уровень транзакции для первой сессии
2 SET SESSION CHARACTERISTICS AS TRANSACTION ISOLATION LEVEL SERIALIZABLE;
3
4 -- Начинаем первую транзакцию в первой сессии, вносим изменения
5 BEGIN;
6 UPDATE series.shows SET title = 'I Want to Sleep' WHERE id = 1;
7 INSERT INTO series.shows (title) VALUES ('First session, First transaction');
8 SELECT * FROM series.shows;
```

Below the code, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is selected, showing a table with six rows of data from the 'series.shows' table:

	id [PK] integer	title text	year integer	budget real	synopsis text
1		Breaking Bad	[null]	[null]	[null]
2		Stranger Things	[null]	[null]	[null]
3		Friends	[null]	[null]	[null]
4		The Mandalorian	[null]	[null]	[null]
5		I Want to Sleep	[null]	[null]	[null]
6		First session, First transaction	[null]	[null]	[null]

postgres/postgres... X postgres/postgres@PostgreSQL 17* X postgres/postgres... X

postgres/postgres@PostgreSQL 17

No limit

Query History

```
10 -- Состояние во второй сессии в первой транзакции после изменений в первой сессии
11 SELECT * FROM series.shows;
```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thron...	[null]	[null]	[null]
2	2	Breaking Bad	[null]	[null]	[null]
3	3	Stranger Things	[null]	[null]	[null]
4	4	Friends	[null]	[null]	[null]
5	5	The Mandalorian	[null]	[null]	[null]

postgres/postgres... X postgres/postgres@PostgreSQL 17* X postgres/postgres... X

postgres/postgres@PostgreSQL 17

No limit

Query History

```
12
13 -- Модифицируем данные во второй сессии в первой транзакции
14 UPDATE series.shows SET title = 'Updated: Second session, First transaction' WHERE id = 2;
15 INSERT INTO series.shows (title) VALUES ('Inserted: Second session, First transaction');
16 SELECT * FROM series.shows;
```

Data Output Messages Notifications

Showing 1 rows

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	2	Updated: Second session, First transaction...	[null]	[null]	[null]
6	38	Inserted: Second session, First transaction	[null]	[null]	[null]

postgres/postgres@PostgreSQL 17* × postgres/postgres... × postgres/postgres... ×

postgres/postgres@PostgreSQL 17 No limit

Query History

```
10 -- Состояние в первой сессии в первой транзакции после изменений во второй сессии
11 SELECT * FROM series.shows;
```

Data Output Messages Notifications

	<code>id</code> [PK] integer	<code>title</code> text	<code>year</code> integer	<code>budget</code> real	<code>synopsis</code> text
1	2	Breaking Bad	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	1	I Want to Sleep	[null]	[null]	[null]
6	37	First session, First transaction	[null]	[null]	[null]

postgres/postgres@PostgreSQL 17* × postgres/postgres... × postgres/postgres... ×

postgres/postgres@PostgreSQL 17 No limit

Query History

```
12
13 -- Откатываем изменения первой транзакции первой сессии
14 ROLLBACK;
15 SELECT * FROM series.shows;
```

Data Output Messages Notifications

	<code>id</code> [PK] integer	<code>title</code> text	<code>year</code> integer	<code>budget</code> real	<code>synopsis</code> text
1	1	Game of Thron...	[null]	[null]	[null]
2	2	Breaking Bad	[null]	[null]	[null]
3	3	Stranger Things	[null]	[null]	[null]
4	4	Friends	[null]	[null]	[null]
5	5	The Mandalorian	[null]	[null]	[null]

postgres/postgres... X postgres/postgres@PostgreSQL 17* X postgres/postgres... X

postgres/postgres@PostgreSQL 17

No limit

Query History

```

17
18 -- Состояние во второй сессии в первой транзакции после роллбэка из пер
19 SELECT * FROM series.shows;

```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	2	Updated: Second session, First transacti...	[null]	[null]	[null]
6	38	Inserted: Second session, First transaction	[null]	[null]	[null]

postgres/postgres... X postgres/postgres@PostgreSQL 17* X postgres/postgres... X

postgres/postgres@PostgreSQL 17

No limit

Query History

```

20
21 -- Состояние во второй сессии после первой транзакции
22 COMMIT;
23 SELECT * FROM series.shows;
24

```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	2	Updated: Second session, First transacti...	[null]	[null]	[null]
6	38	Inserted: Second session, First transaction	[null]	[null]	[null]

postgres/postgres@PostgreSQL 17* × postgres/postgres... × postgres/postgres... ×

postgres/postgres@PostgreSQL 17

No limit

Query History

```
-- Состояние в первой сессии после первого коммита во второй сессии
SELECT * FROM series.shows;
```

Data Output Messages Notifications

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	2	Updated: Second session, First transacti...	[null]	[null]	[null]
6	38	Inserted: Second session, First transaction	[null]	[null]	[null]

postgres/postgres... × postgres/postgres@PostgreSQL 17* × postgres/postgres... ×

postgres/postgres@PostgreSQL 17

No limit

Query History

```
-- Начало второй транзакции второй сессии
BEGIN;

-- Состояние сразу после начала второй транзакции второй сессии
SELECT * FROM series.shows;
```

Data Output Messages Notifications

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	2	Updated: Second session, First transacti...	[null]	[null]	[null]
6	38	Inserted: Second session, First transaction	[null]	[null]	[null]

postgres/postgres@PostgreSQL 17* × postgres/postgres... × postgres/postgres... ×

postgres/postgres@PostgreSQL 17 No limit

Query History

```

20 -- Вторая транзакция первой сессии
21 BEGIN;
22 UPDATE series.shows SET title = 'I Want to Sleep' WHERE id = 3;
23 INSERT INTO series.shows (title) VALUES ('First session, Second transaction');
24 SELECT * FROM series.shows;

```

Data Output Messages Notifications

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	4	Friends	[null]	[null]	[null]
3	5	The Mandalorian	[null]	[null]	[null]
4	2	Updated: Second session, First transacti...	[null]	[null]	[null]
5	38	Inserted: Second session, First transaction	[null]	[null]	[null]
6	3	I Want to Sleep	[null]	[null]	[null]
7	39	First session, Second transaction	[null]	[null]	[null]

postgres/postgres... × postgres/postgres@PostgreSQL 17* × postgres/postgres... ×

postgres/postgres@PostgreSQL 17 No limit

Query History

```

31 -- Состояние во второй сессии во второй транзакции после изменений в первой сессии
32 SELECT * FROM series.shows;
33

```

Data Output Messages Notifications

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	4	Friends	[null]	[null]	[null]
4	5	The Mandalorian	[null]	[null]	[null]
5	2	Updated: Second session, First transacti...	[null]	[null]	[null]
6	38	Inserted: Second session, First transaction	[null]	[null]	[null]

The screenshot shows the pgAdmin 4 interface with two sessions open:

- Session 1 (Top Tab):** postgres/postgres@PostgreSQL 17*
- Session 2 (Bottom Tab):** postgres/postgres@PostgreSQL 17*

Session 2 Query:

```
-- Модифицируем данные во второй транзакции второй сессии
UPDATE series.shows SET title = 'Updated: Second session, Second transaction' WHERE id = 4;
INSERT INTO series.shows (title) VALUES ('Inserted: Second session, Second transaction');
SELECT * FROM series.shows;
```

Data Output:

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	5	The Mandalorian	[null]	[null]	[null]
4	2	Updated: Second session, First transaction	[null]	[null]	[null]
5	38	Inserted: Second session, First transaction	[null]	[null]	[null]
6	4	Updated: Second session, Second transacti...	[null]	[null]	[null]
7	40	Inserted: Second session, Second transaction	[null]	[null]	[null]

The screenshot shows the pgAdmin interface with three tabs at the top: 'postgres/postgres@PostgreSQL 17*' (selected), 'postgres/postgres...' (disabled), and 'postgres/postgres...' (disabled). The main window displays a query history and a table named 'series.shows'. The table has columns: id [PK] integer, title text, year integer, budget real, and synopsis text. The data shows seven rows, each with an id and a title, but all other columns contain null values.

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	4	Friends	[null]	[null]	[null]
3	5	The Mandalorian	[null]	[null]	[null]
4	2	Updated: Second session, First transaction	[null]	[null]	[null]
5	38	Inserted: Second session, First transaction	[null]	[null]	[null]
6	3	I Want to Sleep	[null]	[null]	[null]
7	39	First session, Second transaction	[null]	[null]	[null]

postgres/postgres@PostgreSQL 17* × postgres/postgres... × postgres/postgres... ×

postgres/postgres@PostgreSQL 17

No limit

Query History

```

29 -- Коммит второй транзакции первой сессии
30 COMMIT;
31 SELECT * FROM series.shows;

```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	4	Friends	[null]	[null]	[null]
3	5	The Mandalorian	[null]	[null]	[null]
4	2	Updated: Second session, First transaction	[null]	[null]	[null]
5	38	Inserted: Second session, First transaction	[null]	[null]	[null]
6	3	I Want to Sleep	[null]	[null]	[null]
7	39	First session, Second transaction	[null]	[null]	[null]

postgres/postgres... × postgres/postgres@PostgreSQL 17* × postgres/postgres... ×

postgres/postgres@PostgreSQL 17

No limit

Query History

```

38
39 -- Состояние во второй сессии во второй транзакции после коммита из первой сессии
40 SELECT * FROM series.shows;

```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	3	Stranger Things	[null]	[null]	[null]
3	5	The Mandalorian	[null]	[null]	[null]
4	2	Updated: Second session, First transaction	[null]	[null]	[null]
5	38	Inserted: Second session, First transaction	[null]	[null]	[null]
6	4	Updated: Second session, Second transaction	[null]	[null]	[null]
7	40	Inserted: Second session, Second transaction	[null]	[null]	[null]

postgres/postgres... X postgres/postgres@PostgreSQL 17* X postgres/postgres... X

postgres/postgres@PostgreSQL 17

No limit ▾ E ⚡

Query History

```
42 -- Состояние из второй сессии после второй транзакции
43 COMMIT;
44 SELECT * FROM series.shows;
```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	5	The Mandalorian	[null]	[null]	[null]
3	2	Updated: Second session, First transaction	[null]	[null]	[null]
4	38	Inserted: Second session, First transaction	[null]	[null]	[null]
5	3	I Want to Sleep	[null]	[null]	[null]
6	39	First session, Second transaction	[null]	[null]	[null]
7	4	Updated: Second session, Second transacti...	[null]	[null]	[null]
8	40	Inserted: Second session, Second transaction	[null]	[null]	[null]

postgres/postgres@PostgreSQL 17* X postgres/postgres... X postgres/postgres... X

postgres/postgres@PostgreSQL 17

No limit ▾ E ⚡

Query History

```
33 -- Состояние в первой сессии после второго коммита во второй сессии
34 SELECT * FROM series.shows;
```

Data Output Messages Notifications

SQL

	id [PK] integer	title text	year integer	budget real	synopsis text
1	1	Game of Thrones	[null]	[null]	[null]
2	5	The Mandalorian	[null]	[null]	[null]
3	2	Updated: Second session, First transaction	[null]	[null]	[null]
4	38	Inserted: Second session, First transaction	[null]	[null]	[null]
5	3	I Want to Sleep	[null]	[null]	[null]
6	39	First session, Second transaction	[null]	[null]	[null]
7	4	Updated: Second session, Second transacti...	[null]	[null]	[null]
8	40	Inserted: Second session, Second transaction	[null]	[null]	[null]

Заключение

В ходе работы были проведены эксперименты по работе с транзакциями различных уровней изоляции и проанализированы их результаты.

Стоит отметить, что эксперименты 3.1-3.3 были поставлены некорректно: в них совершались по 2 транзакции, однако из-за невнимательности уровень изоляции устанавливался только для первых транзакций, в то время как вторые имели его значение по умолчанию. Из-за этого были ошибочно интерпретированы результаты вторых транзакций экспериментов 3.2-3.3 (я там сокрушался по поводу неожиданного появления и поведения фантомных чтений). Данная ошибка была замечена лишь на этапе изменения дизайна эксперимента во время подготовки пункта 4 данного отчета.

Также замечу, что дизайн экспериментов 4.2-4.3 не позволил выявить различия между уровнями изоляции транзакций, которые устанавливались для первых сессий. Вероятно, различия бы проявились, если бы изменения в таблицы вносились поэтапно, то есть:

- 1) стартует транзакция в первой сессии,
- 2) стартует транзакция во второй сессии,
- 3) производятся изменения в первой сессии,
- 4) фиксируются состояния в обеих сессиях,
- 5) производится изменение во второй сессии,
- 6) фиксируются состояния обеих сессий,
- 7) снова изменения в первой сессии и фиксирование состояний,
- 8) снова изменения во второй сессии и фиксирование состояний,
- 9) завершается транзакция первой сессии и фиксируются состояния,
- 10) производятся изменения в обеих сессиях и фиксируются состояния,
- 11) завершается транзакция во второй сессии и финально фиксируется состояние.

При такой схеме эксперимента, предположительно, в finale мы увидим различный порядок внесенных изменений, а также наличие или отсутствие грязных и фантомных чтений – в зависимости от установленных уровней изоляции транзакций.

Приложение

- ERD структуры таблиц;
- Sql-скрипт сброса состояния таблицы для экспериментов 4.1-4.3;
- Sql-скрипт первой сессии для эксперимента 4.1;
- Sql-скрипт второй сессии для экспериментов 4.1-4.3;
- отчет (docx);
- отчет (pdf).