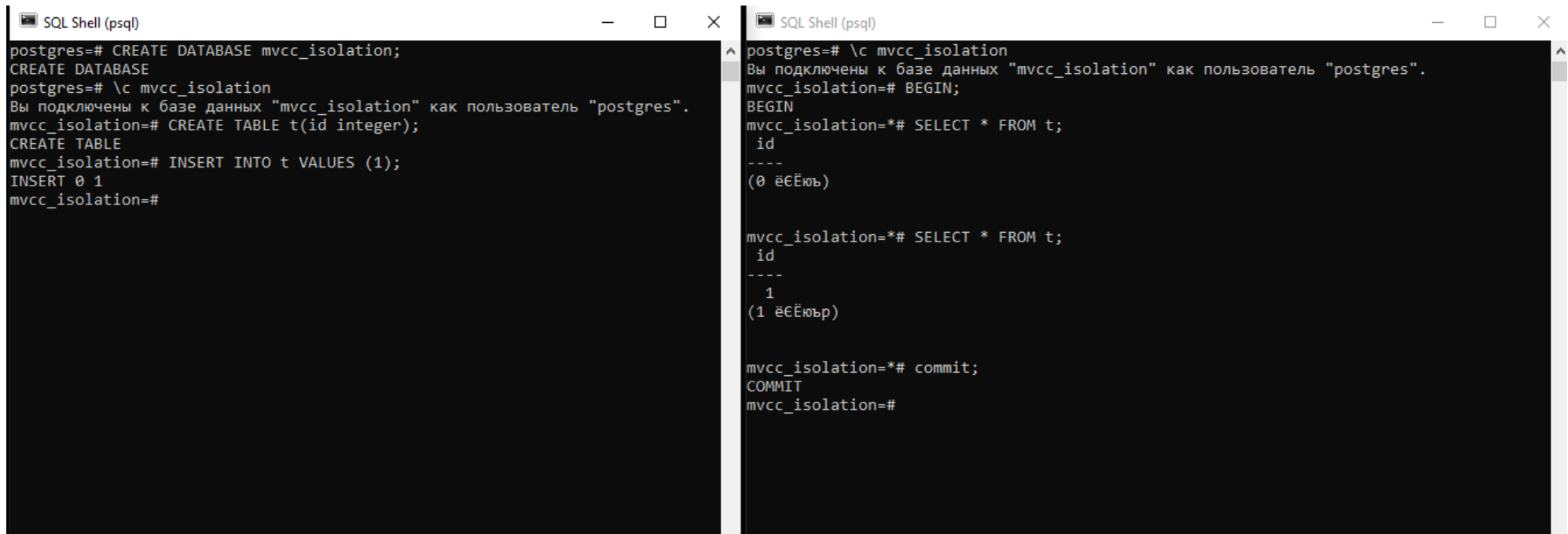


## Уровни изоляции

Проверьте, что на уровне изоляции Read Committed не предотвращается аномалия фантомного чтения.



```
SQL Shell (psql)
postgres=# CREATE DATABASE mvcc_isolation;
CREATE DATABASE
postgres=# \c mvcc_isolation
Вы подключены к базе данных "mvcc_isolation" как пользователь "postgres".
mvcc_isolation=# CREATE TABLE t(id integer);
CREATE TABLE
mvcc_isolation=# INSERT INTO t VALUES (1);
INSERT 0 1
mvcc_isolation=#

SQL Shell (psql)
postgres=# \c mvcc_isolation
Вы подключены к базе данных "mvcc_isolation" как пользователь "postgres".
mvcc_isolation=# BEGIN;
BEGIN
mvcc_isolation=# SELECT * FROM t;
 id
----
(0 ёЁЁюъ)

mvcc_isolation=# SELECT * FROM t;
 id
----
  1
(1 ёЁЁюър)

mvcc_isolation=# commit;
COMMIT
mvcc_isolation=#
```

Транзакция Read Committed видит добавленную строку.

В транзакции с уровнем изоляции Read Committed несколько раз вызывается функция, определение которой изменяется в это время в другой транзакции. Что произойдет? Проверьте и объясните результат.

```
SQL Shell (psql)
mvcc_isolation=#
mvcc_isolation=# CREATE FUNCTION f() RETURNS integer AS $$
mvcc_isolation$$ SELECT 1;
mvcc_isolation$$ $$ LANGUAGE sql;
CREATE FUNCTION
mvcc_isolation=# CREATE OR REPLACE FUNCTION f() RETURNS integer AS $$
mvcc_isolation$$ SELECT 2;
mvcc_isolation$$ $$ LANGUAGE sql;
CREATE FUNCTION
mvcc_isolation=# SELECT f();
 f
---
 2
(1 row)

mvcc_isolation=#

SQL Shell (psql)
mvcc_isolation=#
mvcc_isolation=# BEGIN ISOLATION LEVEL READ COMMITTED;
BEGIN
mvcc_isolation=*# SELECT f();
 f
---
 1
(1 row)

mvcc_isolation=*# SELECT f();
 f
---
 1
(1 row)

mvcc_isolation=*#
```

Многоверсионность в PostgreSQL распространяется и на таблицы системного каталога. В том числе и на `pg_proc`, где хранится исходный код функций. Поэтому разные транзакции могут видеть и вызывать разные экземпляры одной и той же функции. Но при этом даже на уровне `Read Committed` не возникает неповторяемого чтения — этим таблицы системного каталога отличаются от обычных.

Создайте пустую таблицу. Откройте транзакцию с уровнем изоляции Repeatable Read(и пока не выполняйте в ней никаких команд).В другом сеансе добавьте в таблицу строку. Видна ли добавленная строка в открытой транзакции?

```
SQL Shell (psql)
mvcc_isolation=#
mvcc_isolation=# INSERT INTO t VALUES (2);
INSERT 0 1
mvcc_isolation=# INSERT INTO t VALUES (3);
INSERT 0 1
mvcc_isolation=#

SQL Shell (psql)
mvcc_isolation=#
mvcc_isolation=# BEGIN ISOLATION LEVEL REPEATABLE READ;
BEGIN
mvcc_isolation=*# SELECT * FROM t;
 id
----
  1
  2
(2 rows)

mvcc_isolation=*# SELECT * FROM t;
 id
----
  1
  2
(2 rows)

mvcc_isolation=*# commit;
COMMIT
mvcc_isolation=#
```

Транзакции Repeatable Read (и Serializable) видят данные на момент начала первой команды, а не на момент выполнения BEGIN. Если сейчас добавить в таблицу еще строк, результат повторной выборки уже не изменится.