



PROFESSIONAL  
Posigres

ИЗОЛЯЦИЯ



Стандарт SQL и уровни изоляции

Особенности реализации в PostgreSQL

Управление изоляцией транзакций

Последовательность операций,  
составляющая логическую единицу работы

**Атомарность** — все или ничего

при фиксации выполняются все операции,  
при откате не выполняется ни одна

**Согласованность** — целостность данных

система переходит из одного согласованного состояния в другое

**Изоляция** — от других транзакций

на результат не должны оказывать влияние  
другие параллельно работающие транзакции

**Долговечность** — даже после сбоя

зафиксированные изменения никогда не теряются

## Полная изоляция

результат параллельного выполнения транзакций совпадает с результатом их последовательного выполнения

## Четыре уровня изоляции

READ UNCOMMITTED

READ COMMITTED

REPEATABLE READ

SERIALIZABLE

## «Грязное» чтение

транзакция может «видеть» незафиксированные изменения в других транзакциях

## Неповторяющееся чтение

повторный запрос может показывать измененные строки

## Фантомное чтение

повторный запрос может показывать новые строки

# Стандарт SQL

Уровень изоляции	«Грязное» чтение	Неповторяющееся чтение	Фантомное чтение
READ UNCOMMITTED	да	да	да
READ COMMITTED	—	да	да
REPEATABLE READ	—	—	да
SERIALIZABLE	—	—	—

На всех уровнях не допускается потеря зафиксированных изменений

Отсутствие всех трех феноменов не гарантирует полной изоляции

# Реализация в PostgreSQL

Уровень изоляции	«Грязное» чтение	Неповторяемое чтение	Фантомное чтение	Аномалии сериализации
READ UNCOMMITTED	—	да	да	да
READ COMMITTED	—	да	да	да
REPEATABLE READ	—	—	—	да
SERIALIZABLE	—	—	—	—

Определяется моментом создания снимка

## READ COMMITTED

создание снимка при старте *каждой* команды транзакции  
каждая команда транзакции «видит» изменения,  
зафиксированные на момент старта команды  
возможен несогласованный снимок данных

## REPEATABLE READ, SERIALIZABLE

создание снимка при старте *первой* команды транзакции  
все команды транзакции «видят» изменения,  
зафиксированные на момент старта первой команды транзакции

## READ UNCOMMITTED

«Грязное» чтение не допускается

READ UNCOMMITTED = READ COMMITTED



## Уровень изоляции для новых транзакций

параметр *default\_transaction\_isolation*

## Установка

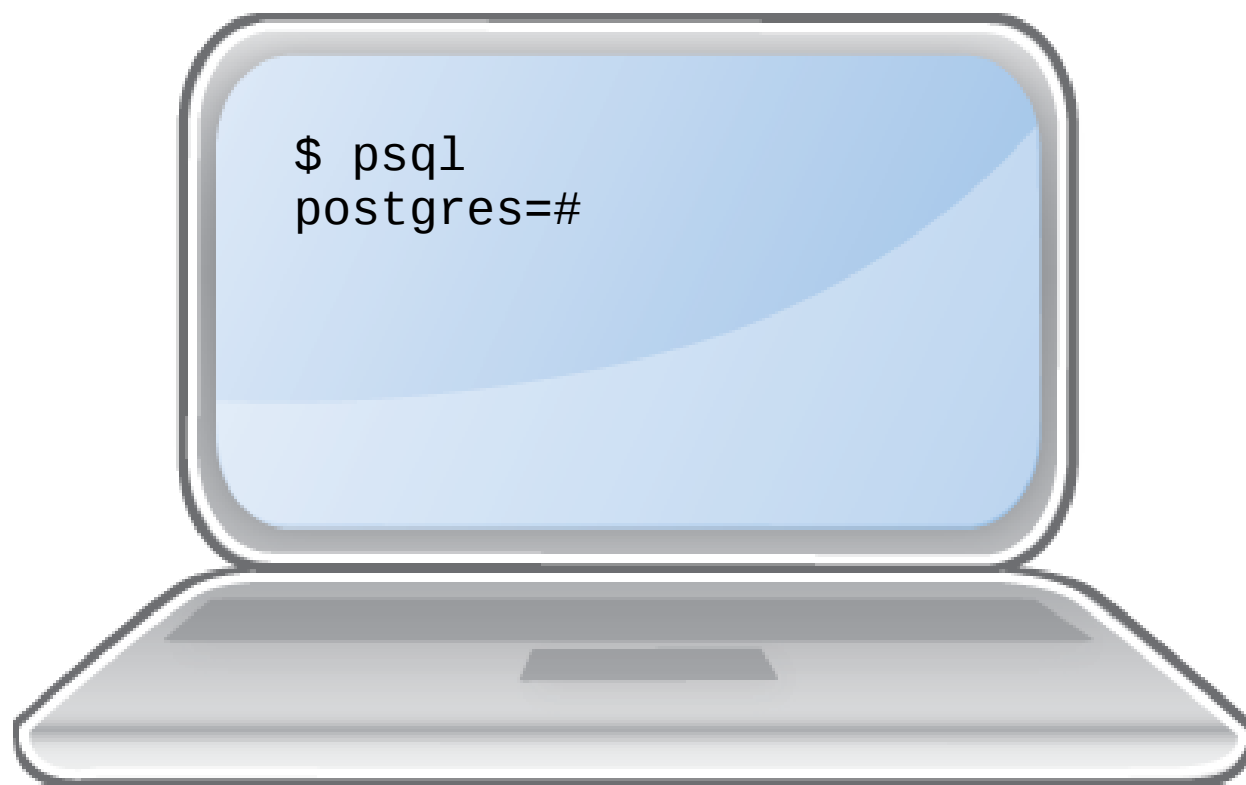
команда `BEGIN ISOLATION LEVEL ...`

команда `SET TRANSACTION ISOLATION LEVEL ...`

## Проверка для текущей транзакции

параметр *transaction\_isolation*

# Демонстрация



«Грязное» чтение не допускается, уровень изоляции READ UNCOMMITTED не нужен

READ COMMITTED используется по умолчанию. Важно учитывать ряд особенностей

Реализация REPEATABLE READ более строгая, чем в стандарте SQL и не допускает фантомного чтения

Для транзакций с уровнем изоляции SERIALIZABLE гарантируется, что результат их одновременной работы будет такой же, как если бы они исполнялись последовательно, одна за другой

Для всех уровней изоляции гарантируется, что читатели не блокируют читателей и писателей, а писатели не блокируют читателей

Выполняется в базе данных DB2.

## 1. Изоляция и системный каталог.

Создайте простую функцию, возвращающую 1.

Во второй сеансе откройте транзакцию (уровень изоляции не важен) и измените определение функции так, чтобы она возвращала 2.

Не закрывая транзакцию, выполните функцию как в первом, так и втором сеансах.

Зафиксируйте изменения во втором сеансе и вызовите функцию еще раз в первом сеансе.

Объясните полученные результаты.

## 2. Момент создания снимка.

Создайте любую таблицу и откройте REPEATABLE READ транзакцию и не выполняйте других команд.

Во втором сеансе (уровень изоляции не важен) добавьте строку в таблицу и зафиксируйте изменения.

Видна ли новая строка в REPEATABLE READ транзакции? Проверьте и объясните результат.



### **Авторские права**

Курс «Администрирование PostgreSQL 9.5. Расширенный курс»  
© Postgres Professional, 2016 год.  
Авторы: Егор Рогов, Павел Лузанов

### **Использование материалов курса**

Некоммерческое использование материалов курса (презентации, демонстрации) разрешается без ограничений. Коммерческое использование возможно только с письменного разрешения компании Postgres Professional. Запрещается внесение изменений в материалы курса.

### **Обратная связь**

Отзывы, замечания и предложения направляйте по адресу:  
[edu@postgrespro.ru](mailto:edu@postgrespro.ru)

### **Отказ от ответственности**

Компания Postgres Professional не несет никакой ответственности за любые повреждения и убытки, включая потерю дохода, нанесенные прямым или косвенным, специальным или случайным использованием материалов курса. Компания Postgres Professional не предоставляет каких-либо гарантий на материалы курса. Материалы курса предоставляются на основе принципа «как есть» и компания Postgres Professional не обязана предоставлять сопровождение, поддержку, обновления, расширения и изменения.

Стандарт SQL и уровни изоляции  
Особенности реализации в PostgreSQL  
Управление изоляцией транзакций

Последовательность операций,  
составляющая логическую единицу работы

**Атомарность** — все или ничего

при фиксации выполняются все операции,  
при откате не выполняется ни одна

**Согласованность** — целостность данных

система переходит из одного согласованного состояния в другое

**Изоляция** — от других транзакций

на результат не должны оказывать влияние  
другие параллельно работающие транзакции

**Долговечность** — даже после сбоя

зафиксированные изменения никогда не теряются

Изоляция является одним из четырех основных свойств транзакций, скрывающихся за аббревиатурой АСИД (ACID). Полностью изолированные транзакции должны работать таким образом, чтобы не оказывать друг на друга никакого влияния.

## Полная изоляция

результат параллельного выполнения транзакций совпадает с результатом их последовательного выполнения

## Четыре уровня изоляции

READ UNCOMMITTED

READ COMMITTED

REPEATABLE READ

SERIALIZABLE

Реализация полной изоляции одновременно работающих транзакций является сложной задачей, поэтому стандарт SQL допускает некоторые послабления в реализации изоляции. С этой целью в нем определены 4 уровня изоляции транзакций, различающиеся тем, в какой степени одновременно работающие транзакции могут воздействовать на результаты друг друга.

К этим четырем уровням относятся: READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ, SERIALIZABLE.

Самый строгий уровень, обеспечивающий полную изоляцию, называется SERIALIZABLE. Он гарантирует что команды, выполняемые в одновременно работающих транзакциях, приведут к такому же результату, какой получился бы в случае последовательного (одна завершилась, следующая началась) выполнения этих транзакций.



## «Грязное» чтение

транзакция может «видеть» незафиксированные изменения в других транзакциях

## Неповторяющееся чтение

повторный запрос может показывать измененные строки

## Фантомное чтение

повторный запрос может показывать новые строки

Послабления для остальных трех уровней изоляции определяются тем, какие ситуации (аномалии) допускаются при одновременном выполнении транзакций.

Стандартом определены три аномалии:

- «Грязное» чтение (dirty read). Транзакция T1 может читать строки измененные, но еще не зафиксированные транзакцией T2. В этом случае, отмена (ROLLBACK) изменений в T2 приведет к тому, что T1 прочитает данные, которых никогда не существовало.
- Неповторяющееся чтение (non-repeatable read). После того, как транзакция T1 прочитала строку, транзакция T2 изменила или удалила эту строку и зафиксировала изменения (COMMIT). При повторном чтении этой же строки, транзакция T1 видит, что строка изменена или удалена. Т. е. результаты повторного запроса не повторились.
- Фантомное чтение (phantom read). Транзакция T1 прочитала набор строк по некоторому критерию. Затем транзакция T2 добавила строки, также удовлетворяющие этому критерию. Если транзакция T1 повторит запрос, она увидит новые строки (фантомы). Как и в случае с неповторяющимся чтением, повторный запрос может возвращать другой результат. Только при фантомном чтении могут появиться новые строки, которых не было при первом запросе, в то время как при неповторяющемся чтении изменяются или пропадают строки, которые существовали после первого запроса.

Уровень изоляции	«Грязное» чтение	Неповторяющееся чтение	Фантомное чтение
READ UNCOMMITTED	да	да	да
READ COMMITTED	—	да	да
REPEATABLE READ	—	—	да
SERIALIZABLE	—	—	—

На всех уровнях не допускается потеря зафиксированных изменений  
Отсутствие всех трех феноменов не гарантирует полной изоляции

Стандартом SQL определено соответствие между уровнями изоляции и теми аномалиями, которые они могут допускать.

Данная таблица не должна вводить в заблуждение о том, что уровень изоляции SERIALIZABLE определяется только запретом на появление трех феноменов. Реализация уровня изоляции SERIALIZABLE (гарантирующего что результат работы одновременных транзакций будет такой же, как и при их последовательном исполнении) требует большего. Подробнее об этом ниже.

Также нужно отметить, что для каждого из четырех уровней изоляции не допускается потеря уже сделанных и зафиксированных изменений в других транзакциях (lost updates).

Уровень изоляции	«Грязное» чтение	Неповторяемое чтение	Фантомное чтение	Аномалии сериализации
READ UNCOMMITTED	—	да	да	да
READ COMMITTED	—	да	да	да
REPEATABLE READ	—	—	—	да
SERIALIZABLE	—	—	—	—

Реализация уровней изоляции в PostgreSQL строже, чем определено стандартом:

- благодаря многоверсионности (MVCC), «грязное» чтение не допускается вообще
- Уровень REPEATABLE READ не допускает не только неповторяемое чтение, но и фантомное чтение

В таблицу добавлена колонка «Аномалии сериализации», для явного подчеркивания того факта, что уровень изоляции SERIALIZABLE не ограничивается отсутствием трех описанных в стандарте аномалий.

Подробнее об уровнях изоляции в PostgreSQL:

<http://www.postgresql.org/docs/current/static/transaction-iso.html>

Определяется моментом создания снимка

## READ COMMITTED

создание снимка при старте *каждой* команды транзакции  
каждая команда транзакции «видит» изменения,  
зафиксированные на момент старта команды  
возможен несогласованный снимок данных

## REPEATABLE READ, SERIALIZABLE

создание снимка при старте *первой* команды транзакции  
все команды транзакции «видят» изменения,  
зафиксированные на момент старта первой команды транзакции

## READ UNCOMMITTED

«Грязное» чтение не допускается  
READ UNCOMMITTED = READ COMMITTED

Предыдущие таблицы говорят о том, какие аномалии не допускаются на том или ином уровне изоляции. Не менее важно понимать какие именно данные будут видны в транзакции для каждого уровня изоляции.

Видимость данных в PostgreSQL определяется моментом создания снимка данных.

Для уровня изоляции READ COMMITTED снимок создается при старте каждой команды транзакции. Таким образом, каждая команда транзакции будет видеть только те изменения, которые были зафиксированы на момент старта команды, а также изменения сделанные самой транзакцией. Это означает, что одна и та же команда, выполненная несколько раз, может видеть разные (уже измененные) данные. В некоторых случаях, после ожидания блокировки, команда будет повторно перечитывать данные, что может привести к получению несогласованных данных.

Для уровней изоляции REPEATABLE READ и SERIALIZABLE снимок создается при старте первой команды транзакции и остается неизменным до ее завершения. Таким образом, все команды транзакции видят одинаковые данные.

Поскольку «грязное» чтение в PostgreSQL невозможно, то уровень изоляции READ UNCOMMITTED эквивалентен уровню изоляции READ COMMITTED.

## Уровень изоляции для новых транзакций

параметр *default\_transaction\_isolation*

## Установка

команда `BEGIN ISOLATION LEVEL ...`

команда `SET TRANSACTION ISOLATION LEVEL ...`

## Проверка для текущей транзакции

параметр *transaction\_isolation*

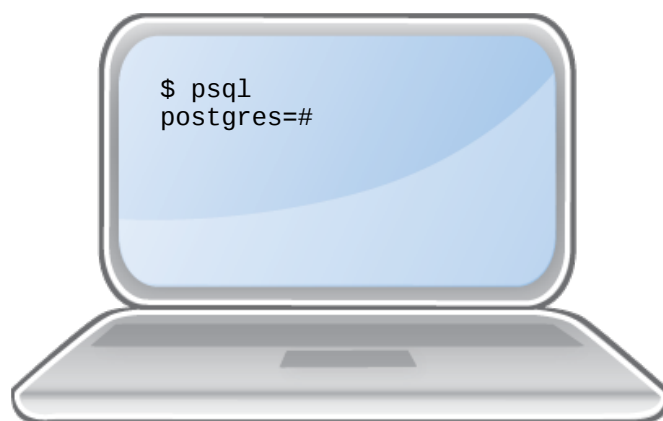
Многие приложения по умолчанию работают в режиме `autocommit`. Это означает, что любая команда выполняется в отдельной транзакции.

Для того, чтобы явно начать транзакцию необходимо выполнить команду `BEGIN`. При этом для транзакции будет установлен уровень изоляции заданный параметром *default\_transaction\_isolation*. По умолчанию, значение этого параметра `READ COMMITTED`.

Явно указать уровень изоляции при открытии транзакции можно двумя способами:

- В самой команде `BEGIN`, после ключевых слов `ISOLATION LEVEL`
- Командой `SET TRANSACTION`, которая должна быть первой после `BEGIN`

Уровень изоляции для текущей транзакции можно узнать из значения параметра *transaction\_isolation*.



«Грязное» чтение не допускается, уровень изоляции READ UNCOMMITTED не нужен

READ COMMITTED используется по умолчанию. Важно учитывать ряд особенностей

Реализация REPEATABLE READ более строгая, чем в стандарте SQL и не допускает фантомного чтения

Для транзакций с уровнем изоляции SERIALIZABLE гарантируется, что результат их одновременной работы будет такой же, как если бы они исполнялись последовательно, одна за другой

Для всех уровней изоляции гарантируется, что читатели не блокируют читателей и писателей, а писатели не блокируют читателей

Выполняется в базе данных DB2.

1. Изоляция и системный каталог.

Создайте простую функцию, возвращающую 1.

Во второй сеансе откройте транзакцию (уровень изоляции не важен) и измените определение функции так, чтобы она возвращала 2.

Не закрывая транзакцию, выполните функцию как в первом, так и втором сеансах.

Зафиксируйте изменения во втором сеансе и вызовите функцию еще раз в первом сеансе.

Объясните полученные результаты.

2. Момент создания снимка.

Создайте любую таблицу и откройте REPEATABLE READ транзакцию и не выполняйте других команд.

Во втором сеансе (уровень изоляции не важен) добавьте строку в таблицу и зафиксируйте изменения.

Видна ли новая строка в REPEATABLE READ транзакции? Проверьте и объясните результат.

1. Для создания функции можно воспользоваться следующим шаблоном:

```
create or replace function f() returns int
as $$ select 1; $$ language sql;
```