

Транзакции, SQL и PostgreSQL

Александр Неганов

Selectel

29 апреля 2013 г.

Транзакции

Что такое транзакция?

Транзакции

Что такое транзакция?

- Версия состояния базы данных

Транзакции

Что такое транзакция?

- Версия состояния базы данных
- Все или ничего

Транзакции

Что такое транзакция?

- Версия состояния базы данных
- Все или ничего
- Изоляция

Изоляция транзакций

``Феномены`` (проблемы) которые могут возникнуть:

Изоляция транзакций

``Феномены`` (проблемы) которые могут возникнуть:

- Потерянное обновление (lost update)

Изоляция транзакций

``Феномены`` (проблемы) которые могут возникнуть:

- Потерянное обновление (lost update) - в SQL стандартах такого не бывает

Изоляция транзакций

``Феномены`` (проблемы) которые могут возникнуть:

- Потерянное обновление (lost update) - в SQL стандартах такого не бывает
- Грязное чтение (dirty read)

Изоляция транзакций

``Феномены`` (проблемы) которые могут возникнуть:

- Потерянное обновление (lost update) - в SQL стандартах такого не бывает
- Грязное чтение (dirty read)
- Неповторяющееся чтение (non-repeatable read)

Изоляция транзакций

``Феномены`` (проблемы) которые могут возникнуть:

- Потерянное обновление (lost update) - в SQL стандартах такого не бывает
- Грязное чтение (dirty read)
- Неповторяющееся чтение (non-repeatable read)
- Фантомное чтение/фантомная вставка (phantom read)

Уровни изоляции транзакций

Стандарт SQL определяет четыре уровня изоляции транзакций:

Уровни изоляции транзакций

Стандарт SQL определяет четыре уровня изоляции транзакций:

- READ UNCOMMITTED (dirty read, non-repeatable read, phantom read)

Уровни изоляции транзакций

Стандарт SQL определяет четыре уровня изоляции транзакций:

- READ UNCOMMITTED (dirty read, non-repeatable read, phantom read)
- READ COMMITTED (non-repeatable read, phantom read)

Уровни изоляции транзакций

Стандарт SQL определяет четыре уровня изоляции транзакций:

- READ UNCOMMITTED (dirty read, non-repeatable read, phantom read)
- READ COMMITTED (non-repeatable read, phantom read)
- REPEATABLE READ (phantom read)

Уровни изоляции транзакций

Стандарт SQL определяет четыре уровня изоляции транзакций:

- READ UNCOMMITTED (dirty read, non-repeatable read, phantom read)
- READ COMMITTED (non-repeatable read, phantom read)
- REPEATABLE READ (phantom read)
- SERIALIZABLE - никакие эффекты невозможны, параллельное выполнение - то же самое что и последовательное.

Транзакции, как их понимает PostgreSQL 9.1

Транзакции, как их понимает PostgreSQL 9.1

- Транзакции могут видеть данные только подтвержденных (committed) транзакций ...

Транзакции, как их понимает PostgreSQL 9.1

- Транзакции могут видеть данные только подтвержденных (committed) транзакций ...
- ... `READ UNCOMMITTED` то же что и `READ COMMITTED`

Транзакции, как их понимает PostgreSQL 9.1

- Транзакции могут видеть данные только подтвержденных (committed) транзакций ...
- ... `READ UNCOMMITTED` то же что и `READ COMMITTED`
- ... Фантомное чтение само по себе невозможно

Транзакции, как их понимает PostgreSQL 9.1

- Транзакции могут видеть данные только подтвержденных (committed) транзакций ...
- ... `READ UNCOMMITTED` то же что и `READ COMMITTED`
- ... Фантомное чтение само по себе невозможно
- `UPDATE / DELETE` лочит изменяемые строки до окончания транзакции

Транзакции, как их понимает PostgreSQL 9.1

- Транзакции могут видеть данные только подтвержденных (committed) транзакций ...
- ... `READ UNCOMMITTED` то же что и `READ COMMITTED`
- ... Фантомное чтение само по себе невозможно
- `UPDATE / DELETE` лочит изменяемые строки до окончания транзакции
- ... что может привести к dead lock'ам

Транзакции, как их понимает PostgreSQL 9.1

- Транзакции могут видеть данные только подтвержденных (committed) транзакций ...
- ... `READ UNCOMMITTED` то же что и `READ COMMITTED`
- ... Фантомное чтение само по себе невозможно
- `UPDATE / DELETE` лочит изменяемые строки до окончания транзакции
- ... что может привести к dead lock'ам
- ... которые хорошо определяются

Транзакции, как их понимает PostgreSQL 9.1

- Транзакции могут видеть данные только подтвержденных (committed) транзакций ...
- ... `READ UNCOMMITTED` то же что и `READ COMMITTED`
- ... Фантомное чтение само по себе невозможно
- `UPDATE / DELETE` лочит изменяемые строки до окончания транзакции
- ... что может привести к dead lock'ам
- ... которые хорошо определяются
- Ошибка сериализации (40001)

Транзакции, как их понимает PostgreSQL 9.1

- Транзакции могут видеть данные только подтвержденных (committed) транзакций ...
- ... `READ UNCOMMITTED` то же что и `READ COMMITTED`
- ... Фантомное чтение само по себе невозможно
- `UPDATE / DELETE` лочит изменяемые строки до окончания транзакции
- ... что может привести к dead lock'ам
- ... которые хорошо определяются
- Ошибка сериализации (40001)
- `REPEATABLE READ` - ``слепок`` состояния БД на начало транзакции + ошибки сериализации

Транзакции, как их понимает PostgreSQL 9.1

- Транзакции могут видеть данные только подтвержденных (committed) транзакций ...
- ... READ UNCOMMITTED то же что и READ COMMITTED
- ... Фантомное чтение само по себе невозможно
- UPDATE / DELETE лочит изменяемые строки до окончания транзакции
- ... что может привести к dead lock'ам
- ... которые хорошо определяются
- Ошибка сериализации (40001)
- REPEATABLE READ - ``слепок`` состояния БД на начало транзакции + ошибки сериализации
- SERIALIZABLE - проверка зависимостей данных, ошибки сериализации

READ COMMITTED / READ UNCOMMITTED

Самый слабый уровень изоляции транзакций

READ COMMITTED / READ UNCOMMITTED

Самый слабый уровень изоляции транзакций

- Уровень изоляции по-умолчанию

READ COMMITTED / READ UNCOMMITTED

Самый слабый уровень изоляции транзакций

- Уровень изоляции по-умолчанию
- Каждый запрос видит состояние БД на момент начала запроса.

READ COMMITTED / READ UNCOMMITTED

Самый слабый уровень изоляции транзакций

- Уровень изоляции по-умолчанию
- Каждый запрос видит состояние БД на момент начала запроса.
- UPDATE / DELETE лочит строки, удовлетворяющие WHERE, перепроверяет и изменяет.

READ COMMITTED / READ UNCOMMITTED

Самый слабый уровень изоляции транзакций

- Уровень изоляции по-умолчанию
- Каждый запрос видит состояние БД на момент начала запроса.
- UPDATE / DELETE лочит строки, удовлетворяющие WHERE, перепроверяет и изменяет.
- Могут возникать странные, очень странные эффекты.

REPEATABLE READ

REPEATABLE READ

- ``Слепок`` состояния БД на начало транзакции

REPEATABLE READ

- ``Слепок`` состояния БД на начало транзакции
- UPDATE / DELETE пытается лочить строки, удовлетворяющие WHERE, и, возможно, выкидывает ошибку сериализации

REPEATABLE READ

- ``Слепок`` состояния БД на начало транзакции
- UPDATE / DELETE пытается лочить строки, удовлетворяющие WHERE, и, возможно, выкидывает ошибку сериализации
- ... к которой нужно быть готовым

REPEATABLE READ

- ``Слепок`` состояния БД на начало транзакции
- UPDATE / DELETE пытается лочить строки, удовлетворяющие WHERE, и, возможно, выкидывает ошибку сериализации
- ... к которой нужно быть готовым
- До 9.1 выбирался при указании SERIALIZABLE

SERIALIZABLE

Самый строгий уровень изоляции транзакций

SERIALIZABLE

Самый строгий уровень изоляции транзакций

- ``Слепок`` состояния БД на начало транзакции

SERIALIZABLE

Самый строгий уровень изоляции транзакций

- ``Слепок`` состояния БД на начало транзакции
- Проверка зависимостей по данным между транзакциями, ошибка сериализации

SERIALIZABLE

Самый строгий уровень изоляции транзакций

- ``Слепок`` состояния БД на начало транзакции
- Проверка зависимостей по данным между транзакциями, ошибка сериализации
- Не добавляет больше локов чем REPEATABLE READ

SERIALIZABLE

Самый строгий уровень изоляции транзакций

- ``Слепок`` состояния БД на начало транзакции
- Проверка зависимостей по данным между транзакциями, ошибка сериализации
- Не добавляет больше локов чем REPEATABLE READ
- ... но разумеется, есть overhead

SERIALIZABLE

Самый строгий уровень изоляции транзакций

- ``Слепок`` состояния БД на начало транзакции
- Проверка зависимостей по данным между транзакциями, ошибка сериализации
- Не добавляет больше локов чем REPEATABLE READ
- ... но разумеется, есть overhead
- Sequential scan затрагивает все строки -> больше ошибок сериализации

SERIALIZABLE

Самый строгий уровень изоляции транзакций

- ``Слепок`` состояния БД на начало транзакции
- Проверка зависимостей по данным между транзакциями, ошибка сериализации
- Не добавляет больше локов чем REPEATABLE READ
- ... но разумеется, есть overhead
- Sequential scan затрагивает все строки -> больше ошибок сериализации
- ... но зато строго повторяемый результат

SERIALIZABLE

Самый строгий уровень изоляции транзакций

- ``Слепок`` состояния БД на начало транзакции
- Проверка зависимостей по данным между транзакциями, ошибка сериализации
- Не добавляет больше локов чем REPEATABLE READ
- ... но разумеется, есть overhead
- Sequential scan затрагивает все строки -> больше ошибок сериализации
- ... но зато строго повторяемый результат
- ... но пока нет поддержки для Hot Standby

Транзакции: выводы

Транзакции: выводы

- Простые транзакции - READ COMMITTED, но с осторожностью

Транзакции: выводы

- Простые транзакции - READ COMMITTED, но с осторожностью
- Отсутствие зависимостей по данным - REPEATABLE READ

Транзакции: выводы

- Простые транзакции - READ COMMITTED, но с осторожностью
- Отсутствие зависимостей по данным - REPEATABLE READ
- Надежно, но без Hot standby и можно медленно - SERIALIZABLE

SQL

Создан в 1974, последняя ревизия стандарта - 2011 год.

SQL

Создан в 1974, последняя ревизия стандарта - 2011 год.

- Стандарт не является полностью свободно/бесплатно доступным.

SQL

Создан в 1974, последняя ревизия стандарта - 2011 год.

- Стандарт не является полностью свободно/бесплатно доступным.
- Стандарт не покрывает всех нужд, существуют свои расширения

SQL

Создан в 1974, последняя ревизия стандарта - 2011 год.

- Стандарт не является полностью свободно/бесплатно доступным.
- Стандарт не покрывает всех нужд, существуют свои расширения
- ... тем не менее, поддержка XML там есть.

SQL

Создан в 1974, последняя ревизия стандарта - 2011 год.

- Стандарт не является полностью свободно/бесплатно доступным.
- Стандарт не покрывает всех нужд, существуют свои расширения
- ... тем не менее, поддержка XML там есть.
- Отличающиеся реализации стандарта разными СУБД

SQL

Создан в 1974, последняя ревизия стандарта - 2011 год.

- Стандарт не является полностью свободно/бесплатно доступным.
- Стандарт не покрывает всех нужд, существуют свои расширения
- ... тем не менее, поддержка XML там есть.
- Отличающиеся реализации стандарта разными СУБД
- NULL, IS NULL, constraints, аргумент функций

SQL

Создан в 1974, последняя ревизия стандарта - 2011 год.

- Стандарт не является полностью свободно/бесплатно доступным.
- Стандарт не покрывает всех нужд, существуют свои расширения
- ... тем не менее, поддержка XML там есть.
- Отличающиеся реализации стандарта разными СУБД
- NULL, IS NULL, constraints, аргумент функций
- Один из самых понятных DSL, по мнению независимых экспертов

Вопросы?