

Уровни изоляции транзакций





Правила вебинара

<https://aristov.tech>

Задаем вопрос в чат

Можно общаться голосом, для этого поставьте знак ? в чат

Вопросы вижу, отвечу в момент логической паузы

Если остались вопросы, можно написать мне через сайт aristov.tech

<https://aristov.tech>

Маршрут вебинара

<https://aristov.tech>

- ❖ Транзакции
- ❖ Свойства транзакций ACID
- ❖ Уровни изоляции транзакций
- ❖ Коротко о проекте aristov.tech
- ❖ Розыгрыш скидки 30% на курс
- ❖ Ответ на вопросы

<https://aristov.tech>

ACID

Параллельная нагрузка

<https://aristov.tech>

Для работы множества пользователей нам необходимо обеспечить:

- ❖ **параллельную** работу множества сессий
- ❖ **модификацию** данных, но так, чтобы пользователи **не мешали** друг другу ни с точки зрения **чтения** ни с точки зрения **записи**
- ❖ **целостность** данных

<https://aristov.tech>

ACID

ACID

- ❖ **Atomicity** — Атомарность
- ❖ **Consistency** — Согласованность
- ❖ **Isolation** — Изолированность
- ❖ **Durability** — Долговечность

Обеспечивается это все с помощью транзакцией (**transaction**) - это одна или более операций, выполняемое приложением, со следующими свойствами:

- ❖ транзакция выполнена полностью (атомарность)
- ❖ параллельные транзакции не мешают друг другу (изолированность)
- ❖ транзакция переводит базу данных из одного корректного состояния в другое корректное состояние (согласованность)

OLTP - **O**nline **T**ransaction **P**rocessing - Транзакционные системы

Транзакция

BEGIN; – начало транзакции

BEGIN TRANSACTION;

START TRANSACTION;

1 и более команд – при ошибке в одном из SQL, дальнейшие команды выполнены НЕ будут и PostgreSQL будет ждать команды ROLLBACK;

COMMIT; – подтверждение транзакции

COMMIT TRANSACTION;

ROLLBACK; – отмена транзакции

ROLLBACK TRANSACTION;

Уровни изоляции транзакций

<https://aristov.tech>

Стандарт SQL допускает 4 уровня изоляции, которые определяются в терминах аномалий, которые допускаются при конкурентном выполнении транзакций на этом уровне:

- ❖ **dirty read** («грязное» чтение) Транзакция 1 может читать строки измененные, но еще не зафиксированные, транзакцией 2. **ROLLBACK** в транзакции 2 приведет к тому, что транзакция 1 *прочитает данные, которых никогда не существовало.*
- ❖ **non-repeatable read** (неповторяющееся чтение). После того, как транзакция 1 прочитала строку, транзакция 2 изменила или удалила эту строку и выполнила **COMMIT**. *При повторном чтении этой же строки транзакция 1 видит, что строка изменена или удалена.*
- ❖ **phantom read** (фантомное чтение). Транзакция 1 прочитала набор строк по некоторому условию. Затем транзакция 2 добавила строки, также удовлетворяющие этому условию. *Если транзакция 1 повторит запрос, она получит другую выборку строк.*
- ❖ **serialization anomaly** (аномалия сериализации). СУБД пытается выстроить транзакции последовательно во всех возможных комбинациях. При невозможности одного из вариантов происходит данная ошибка.

<https://aristov.tech>

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

Уровни изоляции транзакций

<https://aristov.tech>

Аномалия сериализации. Классический пример из документации:

<https://www.postgresql.org/docs/current/transaction-iso.html#XACT-SERIALIZABLE>

class | value

-----+-----

1 | 10

1 | 20

2 | 100

2 | 200

Suppose that serializable transaction A computes:

```
SELECT SUM(value) FROM mytab WHERE class = 1;
```

and then inserts the result (30) as the **value** in a new row with **class** = 2. Concurrently, serializable transaction B computes:

```
SELECT SUM(value) FROM mytab WHERE class = 2;
```

and obtains the result 300, which it inserts in a new row with **class** = 1. Then both transactions try to commit.

<https://aristov.tech>

Уровни изоляции транзакций

<https://aristov.tech>

If either transaction were running at the Repeatable Read isolation level, both would be allowed to commit; but since there is no serial order of execution consistent with the result, using Serializable transactions will allow one transaction to commit and will roll the other back with this message:

ERROR: could not serialize access due to read/write dependencies among transactions

This is because if A had executed before B, B would have computed the sum 330, not 300, and similarly the other order would have resulted in a different sum computed by A.

<https://aristov.tech>

Уровни изоляции транзакций

<https://aristov.tech>

	dirty read	non-repeatable read	phantom read	serialization anomaly
Read Uncommitted	-	+	+	+
Read Committed	-	+	+	+
Repeatable Read	-	-	-	+
Serializable	-	-	-	-

На всех уровнях не допускается потеря зафиксированных изменений, то есть реализуется буква D - Durability, но это если выставлен synchronous_commit в уровень local или выше

<https://aristov.tech/blog/urovni-izolyaczii-tranzakczij/>

<https://aristov.tech>

Практика

Проект aristov.tech

aristov.tech

Книга по 14 Постгресу (скоро по оптимизации 16 Постгреса)

<https://aristov.tech/#orderbook>

Эксклюзивный курс по Оптимизации Постгреса

<https://aristov.tech/blog/kurs-po-optimizaczii-postgresql-2-0/>

Блог с популярными темами

<https://aristov.tech/blog>

ТГ канал с новостями блога и проекта

https://t.me/aristov_tech

Ютуб канал с интересными видео

<https://www.youtube.com/@aristovtech>

Моя группа ДБА

<https://t.me/+S-twn4foKFAtUNil>

Всегда можно со мной связаться через сайт для консультация, аудита вашего проекта, личного менторинга, обучения и многого другого

PostgreSQL 16 под капотом: архитектура и методы оптимизации

Оглавление

Об авторе	4
1. ПостгресQL 16. Настройка VM, ОС и БД	6
2. Коннектинг к ПостгресQL. Права пользователя	40
3. Настройка файловой системы.....	63
4. Настройка бэкапов и репликации	83
5. Мониторинг, профилирование и логирование	113
6. Тюнинг WAL & shared_buffers	140
7. Vacuum, background writer, work_mem, statistic collector, locks	164
8. Схема данных.....	188
9. Оптимизация запросов	222
10. Обслуживание СУБД. Работа на стенде.....	250
Заключение	274

aristov.tech

Розыгрыш книги

регистрируемся по форме:

<https://forms.gle/9i8eRcSHGLnMXi3B7>

Сам розыгрыш в db-fiddle

<https://www.db-fiddle.com/f/43wbuUzv7YUAcS3aypkvvC/1>

ДЗ

ДЗ

1. Подписаться на канал
2. Подписаться на Ютуб
3. Вступить в мою группу ДБА
4. Посещать открытые уроки
5. Прийти на курс %)

Спасибо за внимание!

Следующее открытый урок 02.03 в 10:00
тему и ссылки для подключения опубликую на сайте в ближайшее время

Аристов Евгений