Внедрение партицирования без простоя

Сергей Новиков

ЕДИНЫЙ ЦУПИС

PGConf.Russia
2021

Кто мы

ЕДИНЫЙ ЦУПИС — высоконагруженная fintech-экосистема, одна из крупнейших на рынке.

Собственный электронный кошелёк.

Миллионы платежей ежедневно.

Зачем нам партицирование

Новых данных много, но нужны они не навсегда.

Зачем нам партицирование

Новых данных много, но нужны они не навсегда.

Проще бороться с распуханием.

Зачем нам партицирование

Новых данных много, но нужны они не навсегда.

Проще бороться с распуханием.

Проще управлять архивом.

3... 2... 1...

Партицирование или секционирование?

3... 2... 1...

Партицирование или секционирование?

В основном, говорим про RANGE-партицирование.

3... 2... 1...

Партицирование или секционирование?

В основном, говорим про RANGE-партицирование.

Слайдов много, запоминайте номер, если есть вопросы.

3... 2... 1...

Партицирование или секционирование?

В основном, говорим про RANGE-партицирование.

Слайдов много, запоминайте номер, если есть вопросы.

А практическая часть будет?

Внедрение партиций – общие принципы

В любом случае это будет новая таблица.

Внедрение партиций – общие принципы

В любом случае это будет новая таблица.

Возможно, потребуются изменения в коде приложения.

Внедрение партиций – общие принципы

В любом случае это будет новая таблица.

Возможно, потребуются изменения в коде приложения.

Партицированная таблица не во всём функционирует, как обычно.

Внедрение партиций – базовый пример

Простейшая схема – добавление старой таблицы в качестве дефолтной партиции.

Внедрение партиций – базовый пример

Простейшая схема – добавление старой таблицы в качестве дефолтной партиции.

И это неправильно…

Внедрение партиций – базовый пример

Правильная схема – использование валидируемых ограничений для создания диапазонной партиции.

Внедрение партиций – базовый пример

Правильная схема – использование валидируемых ограничений для создания диапазонной партиции.

Плюс пустая дефолтная партиция.

Внедрение партиций – базовый пример

Всё преобразование делается в одной транзакции.

Внедрение партиций – базовый пример

Всё преобразование делается в одной транзакции.

И эта транзакция должна отрабатывать моментально.

Внедрение партиций – базовый пример

```
-- Таблица, готовая к преобразованию.
CREATE TABLE event (
    event_id    BIGSERIAL   NOT NULL,
    create_date DATE        NOT NULL DEFAULT now(),
    data        JSONB       NOT NULL,

    CONSTRAINT pk_event PRIMARY KEY (event_id, create_date)
);
```

Внедрение партиций – базовый пример

```
-- Добавление валидируемой проверки.
ALTER TABLE event ADD CONSTRAINT ck_event_create_date
    CHECK (create_date < '2021-11-01') NOT VALID;

ALTER TABLE event VALIDATE CONSTRAINT ck_event_create_date;
```

Внедрение партиций – базовый пример

```
-- Подготовка партицированной таблицы.
CREATE TABLE event_partition (
    LIKE event INCLUDING DEFAULTS,
    CONSTRAINT pk_event_partition PRIMARY KEY (event_id, create_date)
) PARTITION BY RANGE (create_date);
```

Внедрение партиций – базовый пример

```
-- Преобразование.
BEGIN;
SET LOCAL statement_timeout = '1s';

ALTER TABLE event RENAME CONSTRAINT pk_event TO pk_event_old;
ALTER TABLE event RENAME TO event_old;

ALTER TABLE event_partition ATTACH PARTITION event_old
    FOR VALUES FROM (MINVALUE) TO ('2021-11-01’);

ALTER TABLE event_partition RENAME TO event;
ALTER TABLE event RENAME CONSTRAINT pk_event_partition TO pk_event;

COMMIT;
```

Внедрение партиций – базовый пример

```
-- Добавление партиции по умолчанию.
CREATE TABLE event_partition_default PARTITION OF event DEFAULT;
```

Внедрение партиций – базовый пример

```
-- Необязательная зачистка мусора.
ALTER TABLE event_old DROP CONSTRAINT ck_event_create_date;
```

Внедрение партиций – базовый пример

```
-- Необязательная зачистка мусора.
ALTER TABLE event_old DROP CONSTRAINT ck_event_create_date;

-- И кое-что мы ещё обсудим далее…
```

Внедрение партиций – борьба с NULL

NULL невозможен, если хочется включить ключ разбиения в состав первичного ключа.

Внедрение партиций – борьба с NULL

NULL невозможен, если хочется включить ключ разбиения в состав первичного ключа.

NULL в случае RANGE-партицирования возможно хранить только в дефолтной партиции.

Внедрение партиций – борьба с NULL

NULL невозможен, если хочется включить ключ разбиения в состав первичного ключа.

NULL в случае RANGE-партицирования возможно хранить только в дефолтной партиции.

https://habr.com/ru/company/haulmont/blog/493954/

Внедрение партиций – борьба с NULL

```
ALTER TABLE event ADD CONSTRAINT ck_event_create_date_not_null
    CHECK (create_date IS NOT NULL) NOT VALID;
ALTER TABLE event VALIDATE CONSTRAINT ck_event_create_date_not_null;

-- До 12 версии.
BEGIN;
LOCK TABLE event IN ACCESS EXCLUSIVE MODE;
UPDATE pg_attribute
SET attnotnull = TRUE
WHERE attrelid = 'event'::REGCLASS::OID AND attname = 'create_date';
COMMIT;

-- Начиная с 12 версии.
ALTER TABLE event ALTER COLUMN create_date SET NOT NULL;
```

Внедрение партиций – первичный ключ

Если приложение задаёт уникальные ключи, всё сложно.

Внедрение партиций – первичный ключ

Если приложение задаёт уникальные ключи, всё сложно:

- Расширение ключа может привести к дубликатам.

Внедрение партиций – первичный ключ

Если приложение задаёт уникальные ключи, всё сложно:

- Расширение ключа может привести к дубликатам.
- ON CONFLICT, возможно, придётся переписывать.

Внедрение партиций – первичный ключ

Если приложение задаёт уникальные ключи, всё сложно:

- Расширение ключа может привести к дубликатам.
- ON CONFLICT, возможно, придётся переписывать.
- ON CONFLICT не позволяет поменять партицию.

Внедрение партиций – первичный ключ

Порядок расширения уникального ключа:

1. Добавить новое ограничение с ключом разбиения.
2. Обеспечить его корректное срабатывание.
3. Поменять в запросе ON CONFLICT.
4. Удалить старое ограничение.

Внедрение партиций – первичный ключ

Чувствительны к смене уникального ключа:

- Внешние ключи.
- Группировка по первичному ключу.

Внедрение партиций – подводные камни

```
-- А что мы здесь забыли?
CREATE TABLE event (
    event_id    BIGSERIAL   NOT NULL,
    create_date DATE        NOT NULL DEFAULT now(),
    data        JSONB       NOT NULL,

    CONSTRAINT pk_event PRIMARY KEY (event_id, create_date)
) PARTITION BY RANGE (create_date);
```

Внедрение партиций – подводные камни

```
-- SERIAL-последовательности требуют перепривязки!
ALTER SEQUENCE event_event_id_seq OWNED BY event.event_id;
```

Внедрение партиций – подводные камни

Права доступа к партициям:

- На чтение/запись – наследуются.
- На изменение схемы – не наследуются.

Внедрение партиций – подводные камни

Длинные имена таблиц – проблема с именами партиций.

Внедрение партиций – подводные камни

Длинные имена таблиц – проблема с именами партиций.

Наш вариант: p*2021_10_hash(schema.table)*

Внедрение партиций – подводные камни

```
-- OID меняется, что может затронуть рекомендательные блокировки.
SELECT pg_advisory_xact_lock('event'::REGCLASS::INT, 12345);
```

Внедрение партиций – работа с кодом

Нужно ли хранить партицирование в коде приложения?

- Путаницы может быть больше, чем пользы.
- Требуется полная автоматизация тестового стенда.

Внедрение партиций – работа с кодом

Общий порядок проведения работ:

1. Архитектурное ревью, проектирование изменений.
2. Подготовка таблицы (индексы, ограничения).
3. Релиз приложения, подготовленного к миграции.
4. Релиз миграции, изменяющей схему таблицы.
5. Преобразование таблицы.

Внедрение партиций – работа с кодом

```
-- Пример миграции для таблицы, требующей подготовки.
CREATE UNIQUE INDEX CONCURRENTLY IF NOT EXISTS ux_event
    ON event (event_id, create_date);

ALTER TABLE event
    ALTER COLUMN create_date SET NOT NULL,
    DROP CONSTRAINT pk_event,
    ADD CONSTRAINT pk_event PRIMARY KEY USING INDEX ux_event;
```

Внедрение партиций – 3 любимых DDL

```
ADD COLUMN … DEFAULT NULL

ADD CONSTRAINT … NOT VALID / VALIDATE CONSTRAINT

CREATE INDEX CONCURRENTLY
```

Внедрение партиций – сложные случаи

Симптомы того, что всё будет непросто:

1. Разбиение по дате, но приложение её не задаёт.
2. ON CONFLICT.
3. Бонус-уровень: дата обновляется (*update_date*).

Внедрение партиций – сложные случаи

Первая идея – использовать триггер, который:

1. Найдёт предыдущую версию строки.
2. Возьмёт из неё старую дату.
3. Подставит эту дату в NEW.
4. Что вызовет срабатывание ON CONFLICT.

Внедрение партиций – сложные случаи

BEFORE ROW триггеры:

1. До 13 версии не работают с партицированием.
2. В 13 версии не позволяют менять партицию.
3. Не обеспечивают защиту от дубликатов.

Внедрение партиций – сложные случаи

Итоговый вариант:

1. Рекомендательная блокировка в транзакции.
2. Ручной UPDATE / INSERT.
3. ON CONFLICT пришлось убрать.

Перенос данных – возможные причины

Очень большая первичная партиция.

Перенос данных – возможные причины

Очень большая первичная партиция.

Неудачное время для преобразования.

Перенос данных – обновление ключа

Значение ключа не должно ни на что влиять.

Перенос данных – обновление ключа

Значение ключа не должно ни на что влиять.

Конкурентные UPDATE будут падать с ошибкой.

Перенос данных – таблицы-копии

Требует запаса места на диске (и не забывайте про WAL).

Перенос данных – таблицы-копии

Требует запаса места на диске (и не забывайте про WAL).

Не применимо при UPDATE.

Перенос данных – таблицы-копии

Требует запаса места на диске (и не забывайте про WAL).

Не применимо при UPDATE.

Можно как нарезать, так и склеивать партиции.

Перенос данных – таблицы-копии

Требует запаса места на диске (и не забывайте про WAL).

Не применимо при UPDATE.

Можно как нарезать, так и склеивать партиции.

Неблокирующая альтернатива VACUUM FULL.

Перенос данных – перенос в транзакции

Алгоритм переноса:

1. Создать новую таблицу с нужными ограничениями.
2. В старой партиции подготовить новое ограничение.
3. В одной транзакции:
   - Перенести строки (ускорит условный индекс).
   - Провалидировать новое ограничение.
   - Перепартицировать таблицу.

Перенос данных – перенос в транзакции

```
-- Подготовка новой партиции.
CREATE TABLE event_new (
    LIKE event INCLUDING DEFAULTS,
    CONSTRAINT pk_event_new PRIMARY KEY (event_id, create_date),
    CONSTRAINT ck_event_new CHECK
        (create_date >= '2021-10-30' AND create_date < '2021-11-01')
);

-- Подготовка переноса данных.
CREATE INDEX CONCURRENTLY ix_event_old_create_date
    ON event_old (create_date) WHERE create_date >= '2021-10-30’;

ALTER TABLE event_old ADD CONSTRAINT ck_event_old_create_date
    CHECK (create_date < '2021-10-30') NOT VALID;
```

Перенос данных – перенос в транзакции

```
BEGIN;
WITH deleted_event AS (
    DELETE FROM event_old WHERE create_date >= '2021-10-30' RETURNING *
)
INSERT INTO event_new TABLE deleted_event;

ALTER TABLE event_old VALIDATE CONSTRAINT ck_event_old_create_date;

ALTER TABLE event DETACH PARTITION event_old;
ALTER TABLE event ATTACH PARTITION event_old
    FOR VALUES FROM (MINVALUE) TO ('2021-10-30');
ALTER TABLE event ATTACH PARTITION event_new
    FOR VALUES FROM ('2021-10-30') TO ('2021-11-01');
COMMIT;
```

Перенос данных – перенос в транзакции

Основная проблема – долгая транзакция.

Перенос данных – перенос в транзакции

Основная проблема – долгая транзакция.

Не требуется много места на диске.

Перенос данных – перенос в транзакции

Основная проблема – долгая транзакция.

Не требуется много места на диске.

Возможно при конкурентной вставке в старую партицию.

Перенос данных – перенос в транзакции

Основная проблема – долгая транзакция.

Не требуется много места на диске.

Возможно при конкурентной вставке в старую партицию.

Применимо при конкурентных UPDATE.

Перенос данных – внешние партиции

Внешние партиции не проверяются на диапазоны.

Перенос данных – внешние партиции

Внешние партиции не проверяются на диапазоны.

Алгоритм переноса (в одной транзакции):

1. Перенести строки на уровне конечных таблиц.
2. Перепартицировать таблицу.

Перенос данных – внешние партиции

Внешние партиции не проверяются на диапазоны.

Алгоритм переноса (в одной транзакции):

1. Перенести строки на уровне конечных таблиц.
2. Перепартицировать таблицу.

Можно делать небольшими порциями много раз.

Перенос данных – внешние партиции

Проблемы:

- Требуется запас *max_connections*.

Перенос данных – внешние партиции

Проблемы:

- Требуется запас *max_connections*.
- Опасно при конкурентных INSERT.

Перенос данных – внешние партиции

Проблемы:

- Требуется запас *max_connections*.
- Опасно при конкурентных INSERT.
- Применимо при UPDATE, но без переноса по ключу.

Перенос данных – внешние партиции

Проблемы:

- Требуется запас *max_connections*.
- Опасно при конкурентных INSERT.
- Применимо при UPDATE, но без переноса по ключу.
- Уникальные ключи не допустимы.

Перенос данных – внешние партиции

Проблемы:

- Требуется запас *max_connections*.
- Опасно при конкурентных INSERT.
- Применимо при UPDATE, но без переноса по ключу.
- Уникальные ключи не допустимы.
- Не работает ON CONFLICT DO UPDATE.

Что ещё можно делать с партициями

Порядок полей таблицы и её партиций может отличаться.

Смена порядка полей – способ сэкономить место.

Что ещё можно делать с партициями

Быстрая смена типов:

1. Архивные партиции копируем с новыми типами.
2. При смене партиции копируем предыдущую.
3. Используем микропартицию для последней копии.
4. Меняем типы и перепартицируем, когда всё готово.

Спасибо за внимание!

Сергей Новиков

ЕДИНЫЙ ЦУПИС