

Темпоральные таблицы & путешествия в прошлое в PostgreSQL

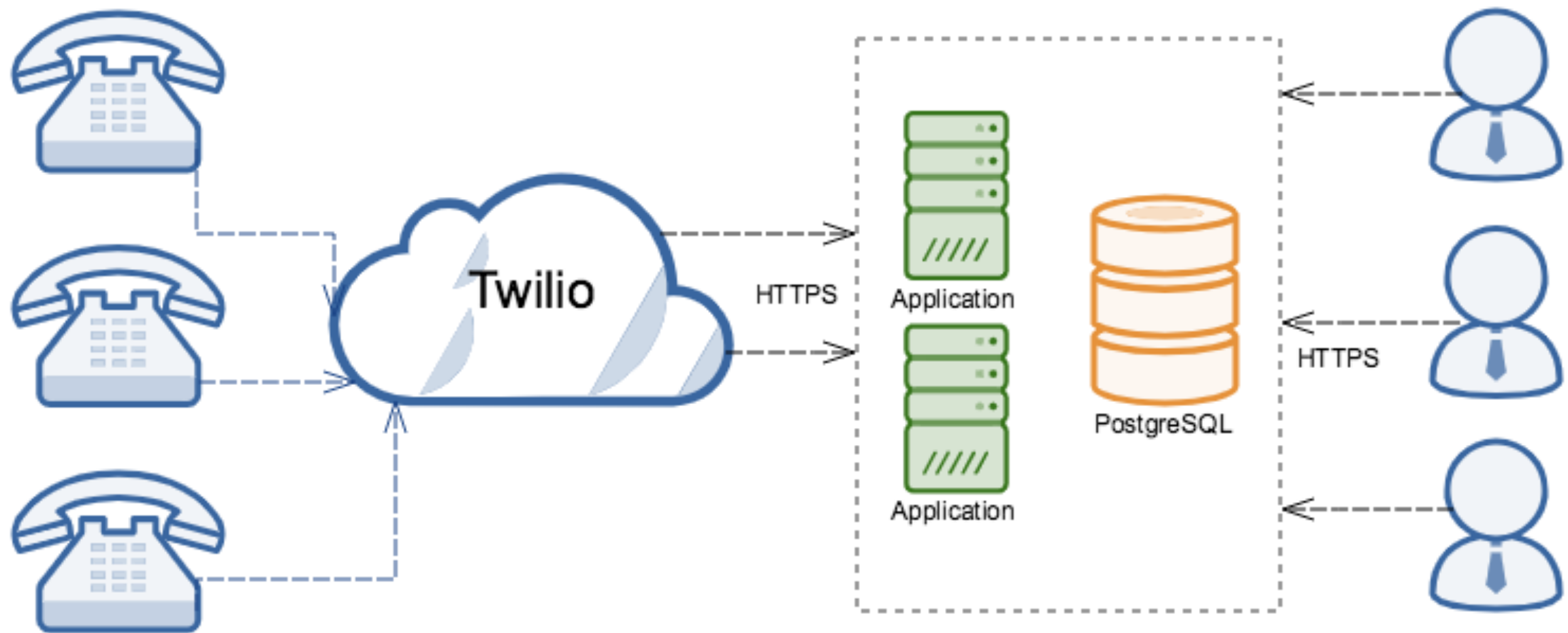
О себе

- Техлид @ www.glia.com
- Бэкэнд-разработчик с 2004
- Работаю с PostgreSQL с 2009
- Основные языки: Elixir, Ruby
- Любимый язык: SQL

План

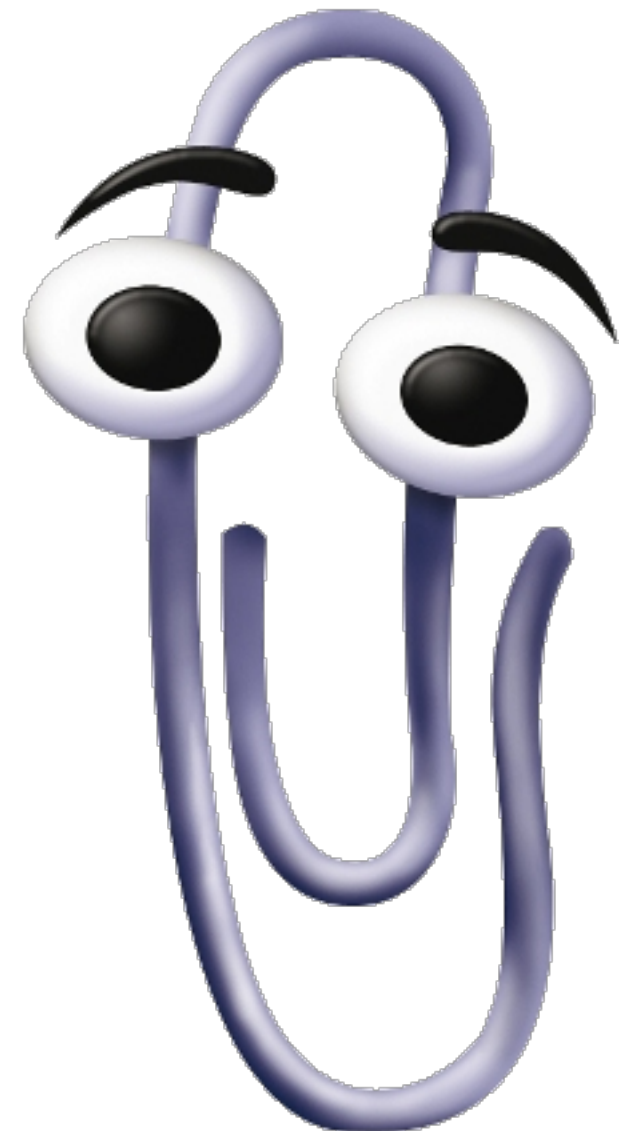
- Баги на живых системах
- Темпоральные таблицы
- Проблемы
- Альтернативы
- Демо

Колл-центр



Жалобы от клиентов

- Почему звонок был отклонен?
- Почему звонок обслужил Вася, а не Петя?
- Почему вчера у Пети было в три раза больше звонков, чем у Васи?



Логи

No results found 😊

Unfortunately I could not find any results matching your search. I tried really hard. I looked all over

Expand your time range

I see you are looking at an index with a date field. It is possible your query does not match anything time picker. For future reference, you can use the time picker to select a range of dates. But

Refine your query

The search bar at the top of the page can be used to refine your query.



Логи

13:48:41.219	info	Operator updated	a925e5a5-52bd-43a4-8009-55fdf65b6a24	temporarily-unavailable
13:48:51.542	info	Operator removed	a925e5a5-52bd-43a4-8009-55fdf65b6a24	temporarily-unavailable
13:48:59.280	info	Operator added	b29beeaf-2863-4d27-984b-b6b8b3ac5465	available
13:48:59.298	info	Queue availability statuses are updated	-	-
13:49:12.172	info	Operator updated	b29beeaf-2863-4d27-984b-b6b8b3ac5465	temporarily-unavailable

Логи

2019-03-10 11:11 Operator updated, id=007

- **Что обновилось?**
- **Что было до этого?**
- **Почему обновилось?**
- **Персональная информация**

Логи

- **"Реактивная" природа**

Нельзя вернуться в прошлое и добавить логов

- **Сложно пользоваться**

> 2019-03-10 23:00 Operator is ready, id=007

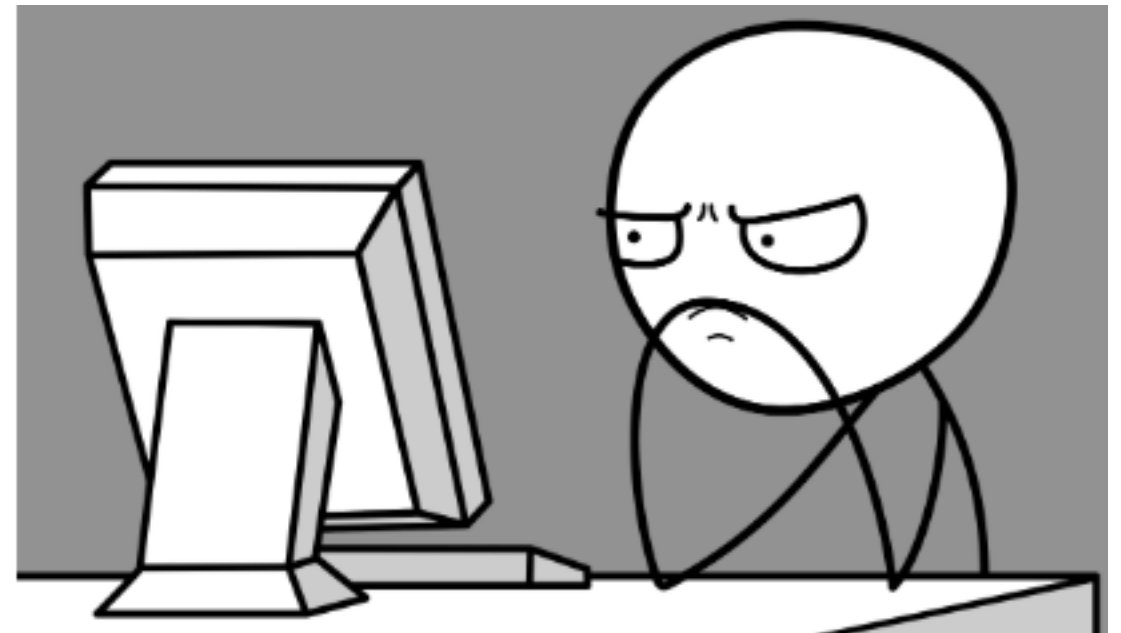
> 2019-03-12 01:00 Operator is offline, id=007

Если искать только по дате 2019-03-11, то оператор 007 будет невидим

- **Недолговечность ***

Пример

- 5 вопросов типа "Почему ответил Вася, а не Петя?"
- 4 часа времени разработчика
- **Как быть дальше?**
 - Приложение на Elixir
 - Служба поддержки пользуется Redash (т.е. ходит в базу в обход приложения), хочется SQL
 - Нужно сделать быстро



Между чем выбирали

- Больше логов
 - Выяснилось, что больше уже некуда

Между чем выбирали

- Больше логов
 - Выяснилось, что больше уже некуда
- Плагин к ORM
 - Для Ecto не нашлось, в Hibernate есть **Envers**
 - Дополнительный запрос к БД
 - Часто неудобный формат для запросов SQL

Темпоральные таблицы

- **Отладка и аудит**
 - Какой был статус оператора, когда нам позвонил клиент?
 - Какой был адрес у клиента, когда мы отправили ему посылку?
 - Пересчитайте налоги на состояние 31-го декабря
- **Откат на предыдущую версию**
- **Исторические данные**

Как это работает

- **Дополнительная колонка(-и) в основной таблице**

- `valid_from + valid_to`
- `system_period (tstzrange)`

- **В Oracle, IBM DB2, MS SQL 2016**

```
ALTER TABLE table ADD SYSTEM VERSIONING;  
SELECT * FROM table AS OF SYSTEM TIME '<timestamp>'
```

- **В PostgreSQL**

- Триггер / расширение `temporal_tables`
- Ручные SQL-запросы

Пример

https://github.com/arkhipov/temporal_tables

```
$ pgxnclient install temporal_tables
```

```
CREATE EXTENSION temporal_tables;
```

Нужен суперпользователь!

Пример

```
CREATE TABLE agents (  
    id integer NOT NULL,  
    status character varying NOT NULL,  
    system_period tstzrange NOT NULL  
);  
  
CREATE TABLE agents_history (LIKE agents);  
  
CREATE TRIGGER versioning_trigger  
BEFORE INSERT OR UPDATE OR DELETE ON agents  
FOR EACH ROW EXECUTE PROCEDURE  
versioning ('system_period', 'agents_history');
```


Пример

```
INSERT INTO agents (id, status) VALUES (1, 'ready');
```

"agents"

id	status	system_period
1	ready	["18:59:17",)

(1 row)

"agents_history"

id	status	system_period
----	--------	---------------

(0 rows)

Пример

```
UPDATE agents SET status = 'away' WHERE id = 1;
```

"agents"

id	status	system_period
1	away	["19:10:47",)

(1 row)

"agents_history"

id	status	system_period
1	ready	["18:59:17","19:10:47")

(1 row)

Пример

```
UPDATE agents SET status = 'offline' WHERE id = 1;
```

"agents"

id	status	system_period
1	offline	["19:15:06",)

(1 row)

"agents_history"

id	status	system_period
1	ready	["18:59:17", "19:10:47")
1	away	["19:10:47", "19:15:06")

(2 rows)

Пример

```
DELETE FROM agents WHERE id = 1;
```

"agents"

id	status	system_period
(0 rows)		

"agents_history"

id	status	system_period
1	ready	["18:59:17", "19:10:47"]
1	away	["19:10:47", "19:15:06"]
1	offline	["19:15:06", "19:17:47"]

(3 rows)

Как пользоваться

```
CREATE VIEW agents_with_history AS  
SELECT * FROM agents  
UNION ALL  
SELECT * FROM agents_history
```

Как пользоваться

Как выглядели мои данные в 19:00?

```
SELECT * FROM agents_with_history
WHERE id = 1
      AND system_period @> '19:12'::timestamptz
```

id	status	system_period
1	away	["19:10:47","19:15:06")

(1 row)

```
SELECT * FROM agents_with_history
WHERE id = 1
      AND system_period @> '19:09'::timestamptz
```

id	status	system_period
1	ready	["18:59:17","19:10:47")

(1 row)

Как пользоваться

Как менялись мои данные с 18:55 до 19:12?

```
SELECT * FROM agents_with_history
WHERE id = 1
      AND system_period <@ tstzrange(' [18:55,19:12] ')
```

id	status	system_period
1	ready	["18:59:17","19:10:47")
1	away	["19:10:47","19:15:06")

(2 rows)

```
SELECT * FROM agents_with_history
WHERE id = 1
      AND system_period <@ tstzrange(' [19:05,) ')
```

id	status	system_period
1	away	["19:10:47","19:15:06")
1	offline	["19:15:06","19:15:47")

(2 rows)

Как пользоваться

Откатиться на версию строки на состояние 19:00

```
UPDATE agents SET status = (  
    SELECT status FROM agents_with_history  
    WHERE id = 1  
        AND system_period @> '19:00'::timestamptz  
) WHERE id = 1
```


Как пользоваться

Сколько времени были доступны операторы в прошлом месяце?

```
SELECT
    SUM(upper(system_period) - lower(system_period))
FROM
    agents_with_history
WHERE
    status = 'online' AND
    system_period <@ '[2019-02-01, 2019-03-01)::tstzrange
```

sum

```
52 days 477:28:19.726338
(1 row)
```

Как пользоваться

Какой оператор был занят больше всех в прошлом месяце?

```
SELECT
  id,
  SUM(upper(system_period) - lower(system_period))
FROM
  agents_with_history
WHERE
  status = 'busy' AND
  system_period <@ '[2019-02-01, 2019-03-01)::tstzrange
GROUP BY id
ORDER BY 2 DESC
LIMIT 1;
```

id	sum
10211	15:10:28.921703

(1 row)

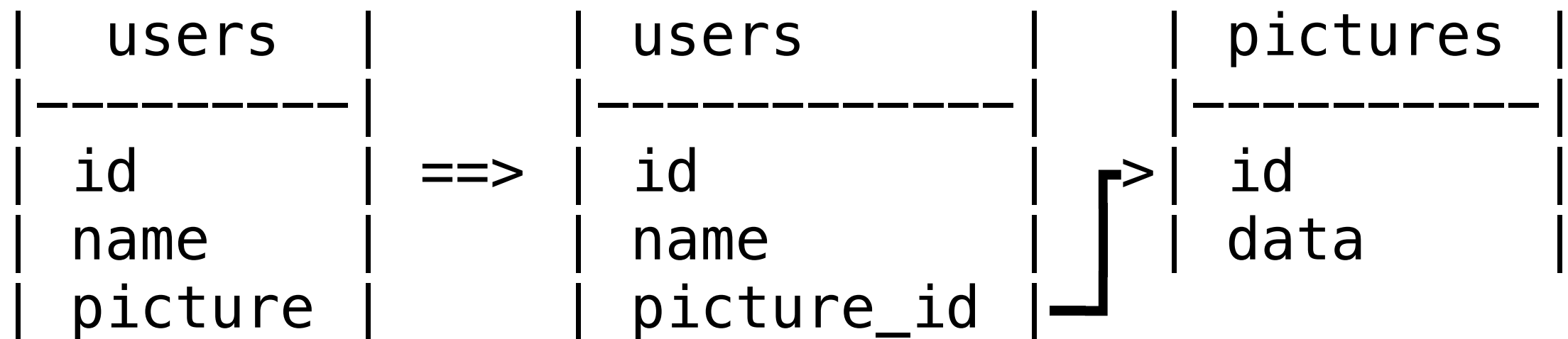
Преимущества

- **ACID**
- **Бэкапы**
- **Легко встроить в существующую БД**
- **Простой интерфейс - SQL-запросы**
- **Мало влияет на производительность**

Дублирование данных

- **Нормализация**

Можно вынести тяжелые колонки в отдельную таблицу



- **Не версионировать некоторые колонки**
- **Периодически чистить исторические данные**

Синхронизация структуры

- Синхронизация структуры

- ALTER TABLE agents ADD COLUMN foo text;
ALTER TABLE agents_history ADD COLUMN foo text;
- ALTER TABLE agents DROP COLUMN foo;
ALTER TABLE agents_history DROP COLUMN foo text;

- Наследование

- CREATE TABLE agents (PRIMARY KEY(id)) INHERITS
(agents_history)
- SELECT FROM ONLY agents_history

- Event triggers

Подводные камни

- Первичные ключи, внешние ключи
- Обязательный тип `tstzrange` для колонки `system_period`
 - Поддерживается только индексом типа "gist"
 - Нельзя сделать составной индекс `gist + gin`, но можно `gist + btree`
- Нестандартное расширение
 - Есть SQL-версия
 - https://github.com/nearform/temporal_tables

Альтернативы

- **Коммерческие БД**
Oracle, IBM DB2, MS SQL

- **Плагин к ORM**

- **Неизменяемые данные**

agents	statuses
+-----+	+-----+
id	agent_id
name	status
	timestamp

Демо

Site ID(s)

21855327-18f7-4b89-82a4-fbdc5716179a ✕

Queue ID(s)

Point in time

dd/mm/yyyy, --:--:--

(GMT+02:00) Tallinn



Show

Queue Name	ID	Status	Enqueued visitors	Staffing operators
------------	----	--------	-------------------	--------------------

Итого

- **Логи помогают, но не всегда**
- **Темпоральные таблицы - мощный инструмент**
 - Но есть свои особенности
 - Но можно и без них

Вопросы?

Слайды:

<https://github.com/take-five/temporal-tables-presentation>

