Quillis
explore your business

Аналитическое хранилище на GREENPLUM

Докладчики:

Гладких Борис

Кузьмичева Надежда

Хандадашев Рустам

Как мы строили Хранилище на GreenPlum или тонкости ELT в условиях Postgres 9.4

Quillis

explore your business

Компания - разработчик собственных продуктов Quillis Ecosystem с многолетним опытом в аналитическом консалтинге и визуализации данных

https://quillis.ru

Почта: contact@quillis.ru

Quillis

Обзорная информация

Greenplum

Quillis
explore your business

- GreenPlum — аналитическая колоночная массивно-параллельная СУБД
- Несколько взаимосвязанных экземпляров базы данных PostgreSQL
- Архитектурная концепция MPP, без разделения ресурсов (Shared Nothing)

MPP — massively parallel processing (массово-параллельная обработка)

- Независимые узлы, особенность MPP (процессор, оперативная память)
- Автоматическое распределение данных по узлам (шардинг)
- Параллельные запросы на узлах

Quillis
explore your business

Vanilla vs Arenadata DB (субъективно)

Vanilla v 6.19

- ✓ Greenplum Command Center (GPCC)
- ✓ Platform Extension Framework (PXF)
- ✗ Поддержка

Arenadata DB v 6.11.2 EE

- ✓ Поддержка
- ✓ Arenadata Command Center (ADCC)
- ✗ Platform Extension Framework (PXF)

PXF (Platform Extension Framework)

PXF - программная платформа, обеспечивающая параллельный высокопроизводительный доступ к данным

Quillis
explore your business

01 **API**

Поддержка SQL запросов к внешним источникам данных

PXF (Platform Extension Framework)

PXF - программная платформа, обеспечивающая параллельный высокопроизводительный доступ к данным

02

SQL запросы

Простой и стандартизированный API

PXF (Platform Extension Framework)

PXF - программная платформа, обеспечивающая параллельный высокопроизводительный доступ к данным

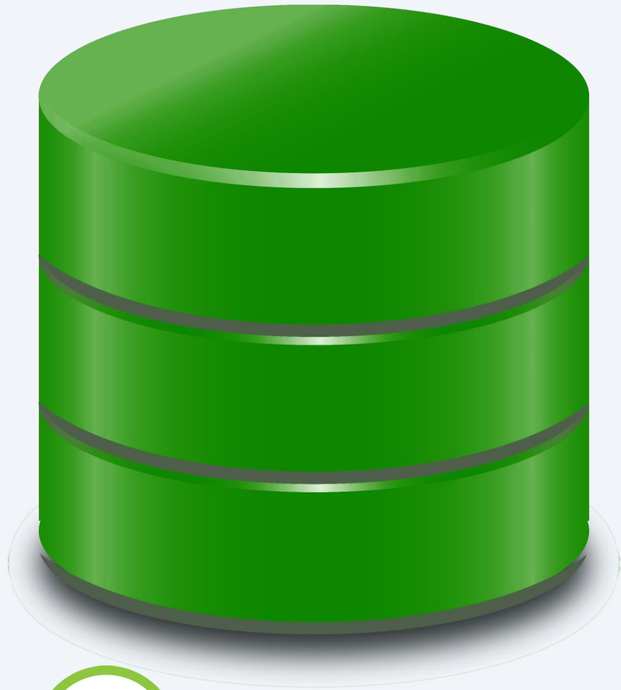
Quillis
explore your business

03 **Форматы данных**

Вариативность форматов данных

MySQL®

ORACLE



Greenplum

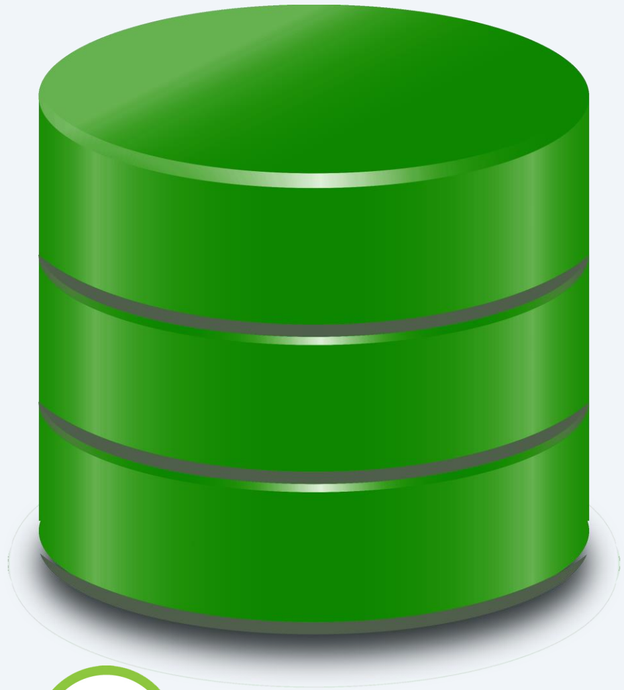
PXF (Platform Extension Framework)

PXF - программная платформа, обеспечивающая параллельный высокопроизводительный доступ к данным

Quillis
explore your business

04 **Независимость от платформы**

Возможность запрашивать данные из различных облачных хранилищ



PXF (Platform Extension Framework)

PXF - программная платформа, обеспечивающая параллельный высокопроизводительный доступ к данным

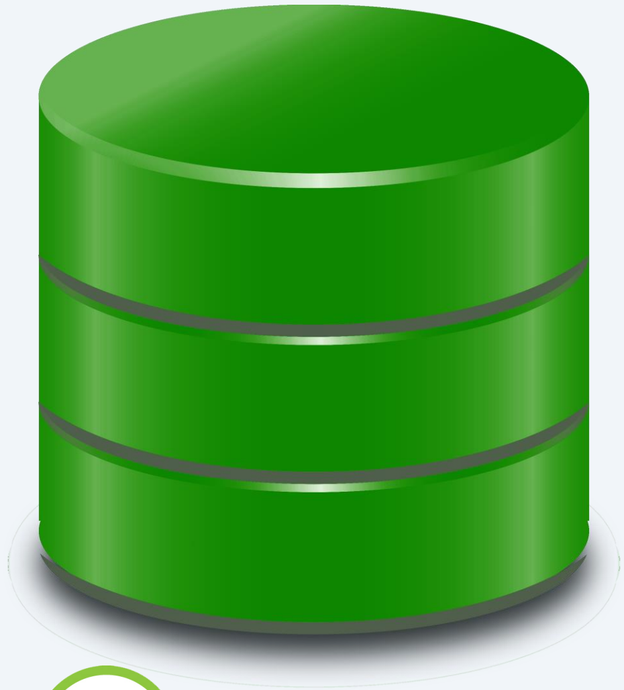
Quillis
explore your business

05 **Масштабируемость**

Многопоточность, способность разбивать и распределять задачи по выбору данных из различных внешних источников

HIVE

APACHE HBASE



Greenplum

Архитектура DWH

Quillis
explore your business

01 Локальный слой

Слой первичных данных, на котором выполняется загрузка информации из систем-источников в исходном качестве и полном объеме

02 Слой справочных и консолидированных данных

Слой, на котором выполняется консолидация данных из разных источников

03 Слой аналитических витрин

Данные преобразуются к структурам, удобным для анализа и использования в аналитических отчетах

04 Витрины в ClickHouse

Аналитические витрины выгруженные в ClickHouse для BI платформ

ELT (Extract. Load. Transform.)

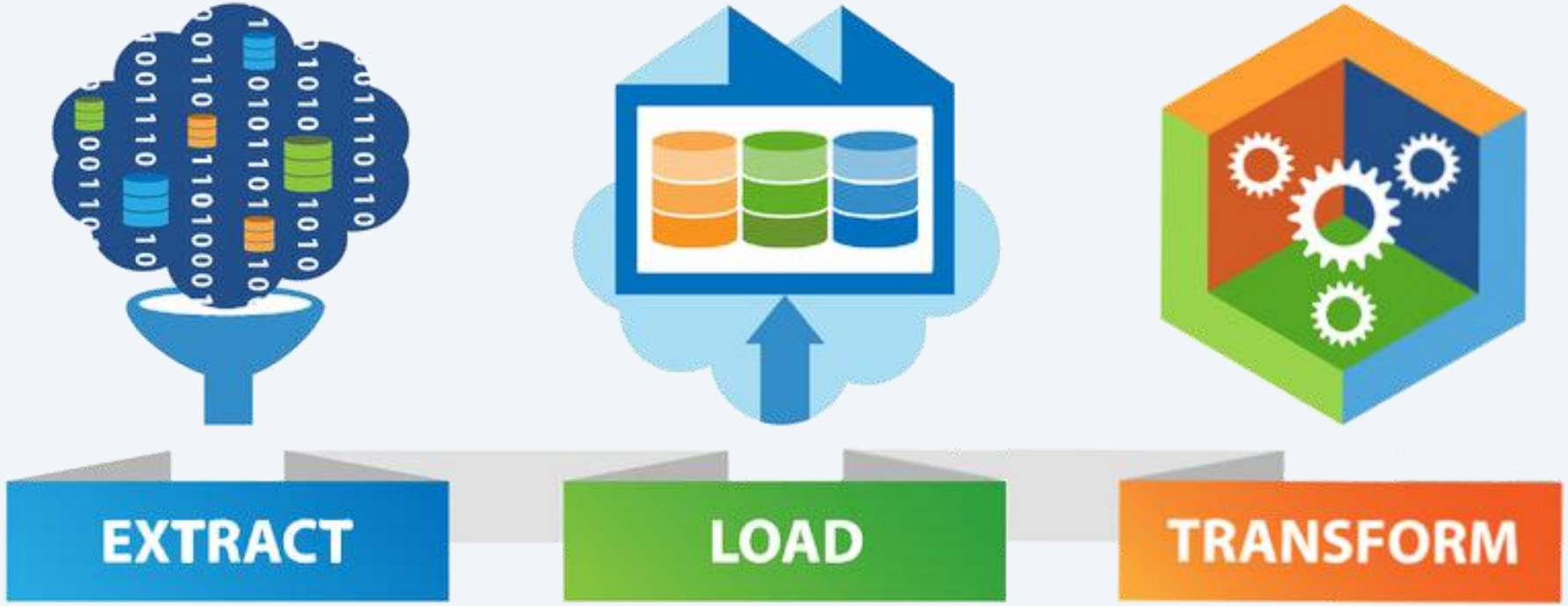
Quillis
explore your business

Подход ELT позволяет извлекать все данные сразу и формировать некое условное «озеро данных» в локальном слое DWH

Варианты загрузки:

- Полная загрузка
- Инкрементная загрузка

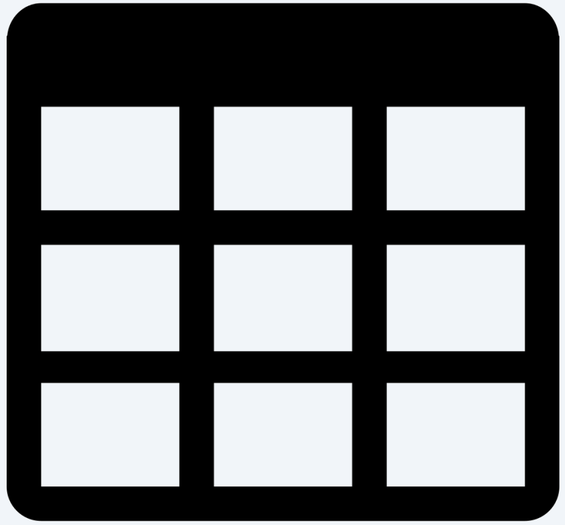
Данный этап включает в себя применение к данным различных преобразований для удобного анализа и отчётности



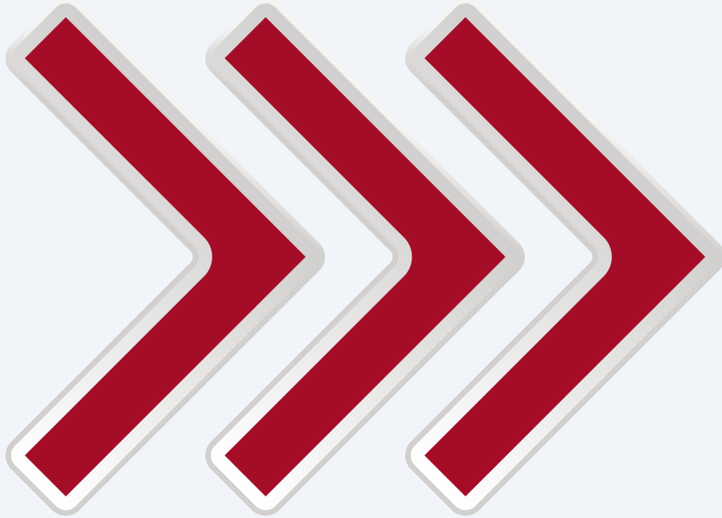
Автоматизация разработки программного кода

Quillis
explore your business

Jinja2



ТАБЛИЦА



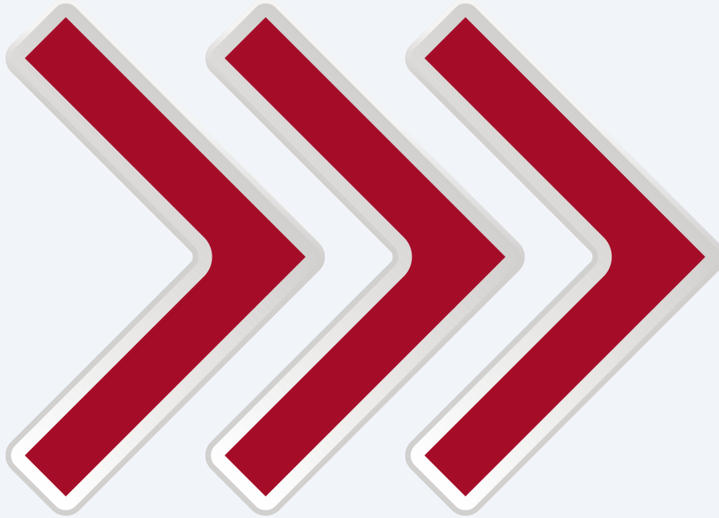
pg_class

Автоматизация разработки программного кода

Quillis
explore your business

Jinja2

CONTEXT

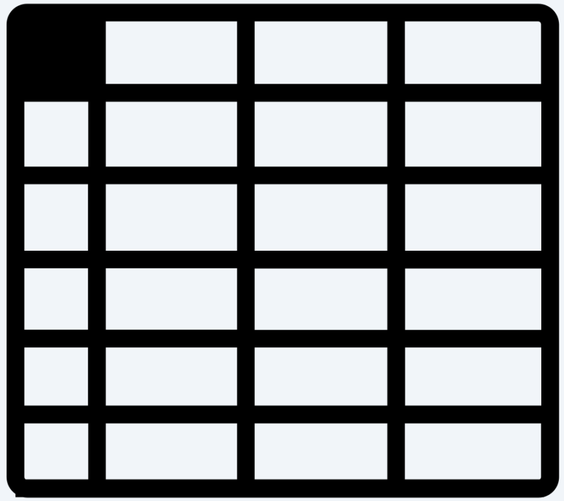


Python script

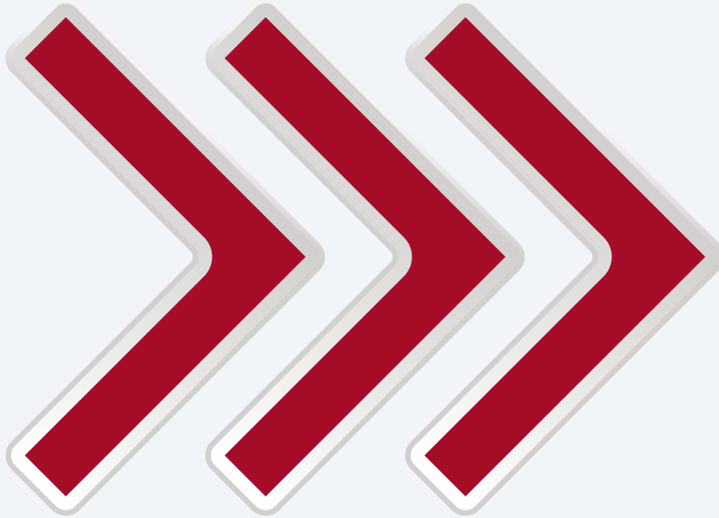
Автоматизация разработки программного кода

Quillis
explore your business

Jinja2



ШАБЛОН

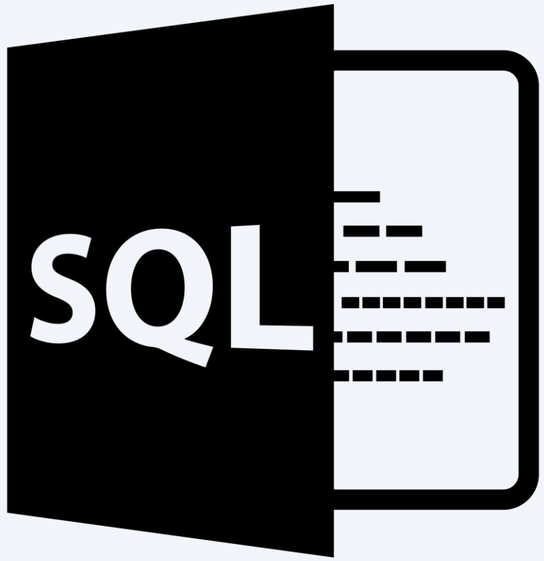


Jinja-render

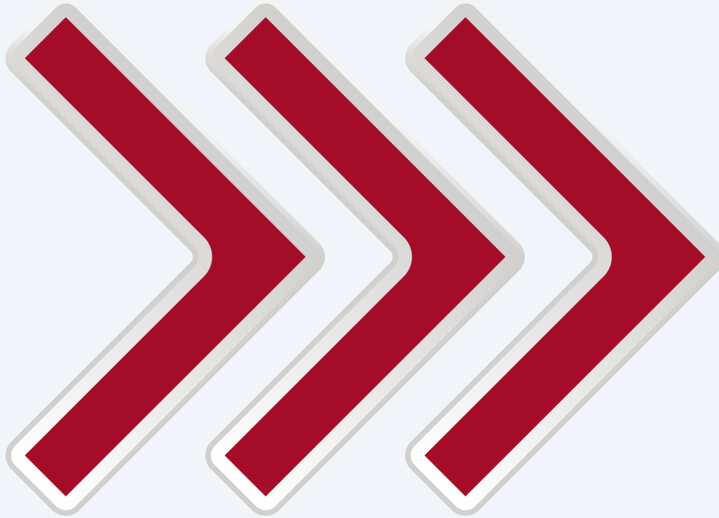
Автоматизация разработки программного кода

Quillis
explore your business

Jinja2



SQL КОД



- Таблица
- Процедура

Quillis
explore your business

Пример контекста

```
CONTEXTS = {
    "group_1": {
        "changeset_number": "PIvanov:20240409-XXXX-XX-ZZ",
        "schema_name": "lcl_test1",
        "entities": [
            {
                "function_name": "lcl_test1.test_function1",
                "src_table": "src_test1.src_test1",
                "dest_table": {
                    "name": "lcl_test1.dest_test1",
                    "distribution_by": "id",
                    "fields": [
                        {"name": "id", "type": "int", "value": "id"},
                        {"name": "name3", "type": "varchar(256)", "value": "name33"},
                        {"name": "ldate2", "type":"time", "value": "v_ldate2"}
                    ]
                }
            }
        ]
    }
}
```

Quillis
explore your business

Шаблон и результат

```
--liquibase formatted sql
{% for entity in entities %}
{% if entity.dest_table.fields is defined %}
--changeset {{ changeset_number }}
CREATE TABLE {{entity.dest_table.name}} (
{% for dst_field in entity.dest_table.fields %}
  {{ dst_field.name }} {{ dst_field.type }}{% if loop.index < loop.length %},
  {% endif %}
{% endfor %}
)
WITH (appendonly = true, compresstype=ZSTD, compresslevel=10)
 DISTRIBUTED BY ({{entity.dest_table.distribution_by}})
;
{% endif %}
{% endfor %}
```

```
--liquibase formatted sql
--changeset PIvanov:20240409-XXXX-XX-ZZ
CREATE TABLE lcl_test1.dest_test1 (
  id int,
  name3 varchar(256),
  ldate2 time)
WITH (appendonly = true, compresstype=ZSTD, compresslevel=10)
 DISTRIBUTED BY (id)
;
```

Quillis
explore your business

Неожиданные проблемы

Quillis
explore your business

Логирование результатов

Автономные транзакции - есть в Oracle, PostgresPro Enterprise 9.6

В Greenplum, под капотом которого Potgres 9.4 - НЕТ

DBLINK - замена автономным транзакциям

Quillis
explore your business

PXF filter pushdown

Таблица фактов:

```
;
CREATE TABLE fact_pxf (
lyear SMALLINT,
id INTEGER,
lmonth SMALLINT,
ldate TIMESTAMP
)
WITH (appendoptimized=true)
DISTRIBUTED RANDOMLY
PARTITION BY RANGE(lyear)
SUBPARTITION BY RANGE(lmonth) SUBPARTITION TEMPLATE
(START (1::SMALLINT) END (12::SMALLINT) EVERY (1))
(
START (2017::SMALLINT) END (2025::SMALLINT) EVERY (1)
);
```

Конфигурационная таблица:

```
CREATE TABLE config_tbl (
dwh_config_code VARCHAR NULL,
dwh_config_value VARCHAR NULL
)
DISTRIBUTED REPLICATED;
```

Quillis
explore your business

PXF filter pushdown

Пример 1:

```
SELECT *
FROM src_fact_pxf
WHERE make_date(lyear, lmonth, 1) > '2017-01-01'::DATE
```

Пример 2:

```
SELECT *
FROM src_fact_pxf
WHERE lyear = (SELECT dwh_config_value::SMALLINT
                 FROM config_tbl
                WHERE dwh_config_code = 'lyear')
```

Пример 3:

```
SELECT *
FROM src_fact_pxf
WHERE ldate > '1900-01-01'::DATE
```

```
test_4pgconf=# SELECT datname, waiting, state, query  FROM pg_stat_activity WHERE usename = 'idvp_test';
   datname    | waiting |        state        |                                query
--------------+---------+---------------------+-----------------------------------------------------------
 test_4pgconf | f       | idle in transaction | SELECT lyear, id, lmonth FROM public.fact_subpart_by_lyear
(1 row)
```

Особенности фильтрации данных по параметру

Quillis
explore your business

Таблица фактов:

```
;
CREATE TABLE fact_subpart_by_lyear (
lyear SMALLINT,
id INTEGER,
lmonth SMALLINT
)
WITH (appendoptimized=true)
DISTRIBUTED RANDOMLY
PARTITION BY RANGE(lyear)
SUBPARTITION BY RANGE(lmonth) SUBPARTITION TEMPLATE
(START (1::SMALLINT) END (12::SMALLINT) EVERY (1))
(
START (2017::SMALLINT) END (2025::SMALLINT) EVERY (1)
);
```

Конфигурационная таблица:

```
CREATE TABLE config_tbl(
dwh_config_code VARCHAR NULL,
dwh_config_value VARCHAR NULL
)
DISTRIBUTED REPLICATED;
```

```
SELECT COUNT(*)
FROM fact_subpart_by_lyear
WHERE lyear >=
(SELECT dwh_config_value::SMALLINT
FROM config_tbl WHERE dwh_config_code = 'lyear')
AND lmonth = 10::SMALLINT;
```

Optimizer: Pivotal Optimizer (GPORCA)

```
-> Partition Selector for fact_subpart_by_lyear
(dynamic scan id: 1)
Partitions selected: 8 (out of 88)
-> Dynamic Seq Scan on fact_subpart_by_lyear (dynamic
scan id: 1) Filter: (lmonth = 10::smallint)
```

Optimizer: Postgres query optimizer

```
...
-> Seq Scan on
fact_subpart_by_lyear_1_prt_3_2_prt_10
Filter: ((lyear >= $0) AND (lmonth = 10::smallint))
...
-> Seq Scan on
fact_subpart_by_lyear_1_prt_8_2_prt_10
Filter: ((lyear >= $0) AND (lmonth = 10::smallint))
```

```
SELECT COUNT(*)
FROM fact_subpart_by_lyear
WHERE lyear >= 2023::SMALLINT
AND lmonth = 10::SMALLINT;
```

Optimizer: Pivotal Optimizer (GPORCA)

```
-> Partition Selector for fact_subpart_by_lyear
(dynamic scan id: 1)
Partitions selected: 2 (out of 88)
-> Dynamic Seq Scan on fact_subpart_by_lyear (dynamic
scan id: 1)
Filter: ((lyear >= 2023::smallint) AND (lmonth =
10::smallint))
```

Optimizer: Postgres query optimizer

```
-> Seq Scan on fact_subpart_by_lyear_1_prt_7_2_prt_10
Filter: ((lyear >= 2023::smallint) AND (lmonth =
10::smallint))
-> Seq Scan on fact_subpart_by_lyear_1_prt_8_2_prt_10
Filter: ((lyear >= 2023::smallint) AND (lmonth =
10::smallint))
```

Quillis
explore your business

Очень медленный INSERT

```
CREATE SEQUENCE fact_by_lyear_fct_is_seq
 INCREMENT BY 1 MINVALUE 1 CACHE 1;

CREATE TABLE fact_by_lyear (
 fct_id BIGINT
  DEFAULT NEXTVAL('fact_by_lyear_fct_id_seq'::regclass),
 lyear SMALLINT,
 fact_by_lyear_id INTEGER
)
WITH (
 appendonly = true, compresstype=ZSTD, compresslevel=10
)
DISTRIBUTED BY (fct_id);
```

Sequence Cache ~ 1 секунда

```
INSERT INTO fact_by_lyear
(lyear, fact_by_lyear_id)
SELECT 2023::SMALLINT, *
  FROM pg_catalog.generate_series
(1, 1 000 000)
```

Время выполнения 20 секунд!!!

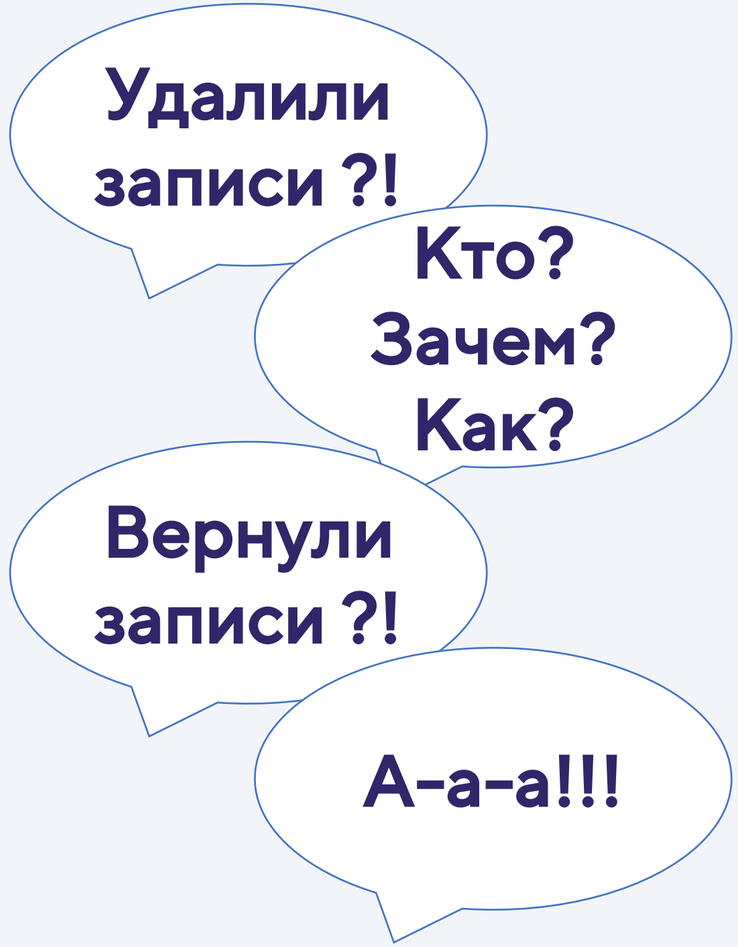
Quillis
explore your business

Фантомы при чтении

```
DROP TABLE IF EXISTS fact_table;
CREATE TABLE fact_table (
  fct_id INTEGER,
  fact_table_id INTEGER,
  is_bool BOOLEAN
)
WITH (appendoptimized=true, orientation=column)
DISTRIBUTED BY (fct_id);

CREATE INDEX fact_table_idx ON fact_table
(fact_table_id);

INSERT INTO fact_table (fct_id, fact_table_id, is_bool)
SELECT generate_series, generate_series, true
FROM generate_series(1, 30 000 000);
```



```
test_4pgconf=> SELECT COUNT(*) FROM fact_table WHERE fact_table_id < 15000;
 count
-------
 14999
(1 row)

test_4pgconf=> SELECT COUNT(*) FROM fact_table WHERE fct_id < 15000;
 count
-------
 14999
(1 row)

test_4pgconf=> UPDATE fact_table SET is_bool = false WHERE fact_table_id < 5000 OR fact_table_id >15000000;
UPDATE 15004999
test_4pgconf=> VACUUM ANALYZE fact_table;
VACUUM
test_4pgconf=> SELECT COUNT(*) FROM fact_table WHERE fact_table_id < 15000;
 count
-------
     0
(1 row)

test_4pgconf=> SELECT COUNT(*) FROM fact_table WHERE fct_id < 15000;
 count
-------
 14999
(1 row)

test_4pgconf=> REINDEX INDEX fact_table_idx;
REINDEX
test_4pgconf=> SELECT COUNT(*) FROM fact_table WHERE fact_table_id < 15000;
 count
-------
 14999
(1 row)
```

Quillis
explore your business

Исчезли записи ?!

HASH JOIN

```
SELECT fct.*, dct.dct_name
FROM fact_part_by_lyear fct
JOIN dct_small dct ON dct.dct_id = fct.dct_id
WHERE fct.lyear = 2021::SMALLINT
AND fct.id > 10018852
```

~ 300 млн. строк в факте
(из них ~4 тыс. ссылаются на справочник)
~ 500 строк в справочнике
Hash-таблица строится по справочнику

```
Hash Join  (rows=151)
  Hash Cond: (fact_part_by_lyear.dct_id = dct_small.dct_id)
  ->  Sequence  (rows=18129840)
  ...
  ->  Hash  (rows=500)
        ->  Broadcast Motion 16:16  (rows=500)
              ->  Seq Scan on dct_small
```

```
SELECT fct.*, dct.dct_name
FROM fact_part_by_lyear fct
LEFT JOIN dct_small dct ON dct.dct_id = fct.dct_id
WHERE fct.lyear = 2021::SMALLINT
AND fct.id > 10018852 AND fct.dct_id < 1000
```

~ 300 млн. строк в факте
(из них ~4 тыс. ссылаются на справочник)
~ 500 строк в справочнике
Hash-таблица строится по факту

```
Hash Right Join  (rows=333)
   Hash Cond: (dct_small.dct_id = fact_part_by_lyear.dct_id)
   ->  Seq Scan on dct_small  (rows=39)
         Filter: (dct_id < 1000)
   ->  Hash  (rows=333)
         ->  Redistribute Motion 16:16  (rows=333)
               Hash Key: fact_part_by_lyear.dct_id
               ->  Sequence  (rows=327)
```

Текущие показатели DWH

Quillis
explore your business

01

Источники и таблицы

Кол-во источников: 27
Кол-во таблиц: 415
Кол-во веб-сервисов: 2
Суммарный объемы данных во всех ис точниках: до 15 ТВ

02

Объекты хранилища

Локальные таблицы - 415
Справочники - 116
Кроссы - 28
Факты - 68

Витрины - 19 (таблицы) 54 (представления)

03

Локальный слой

Время загрузки данных - 1ч 40мин
Ежедневный объем загрузки - 100Гб (сжатие ZSTD, 10)

04

Время расчета Объектов DWH

Расчет справочников - 22мин
Расчет фактов - 30мин
Расчет витрин - 1ч 12мин

05

Занимаемое место

Общий размер 5 Тб (сжатие ZSTD, 10)
Локальный слой - 1,5 Тб
Справочники - 720 Гб
Кросс таблицы - 595 Гб
Факты - 860 Гб
Витрины - 1,4 Тб

06

ClickHouse

Кол-во объектов: 86
Сумарный объем: 137,54 Гб
Время загрузка данных: 1ч 09мин.
Кол-во загружаемых строк: 1 999 595 905

Quillis

explore your business

СПАСИБО ЗА ВНИМАНИЕ