

# *Система продажи билетов на этапы чемпионата мира «Формула 1»*



*Выполнил: студент 2 семестра проекта “Технотрек”*

*Стасьев Денис*

*2019*

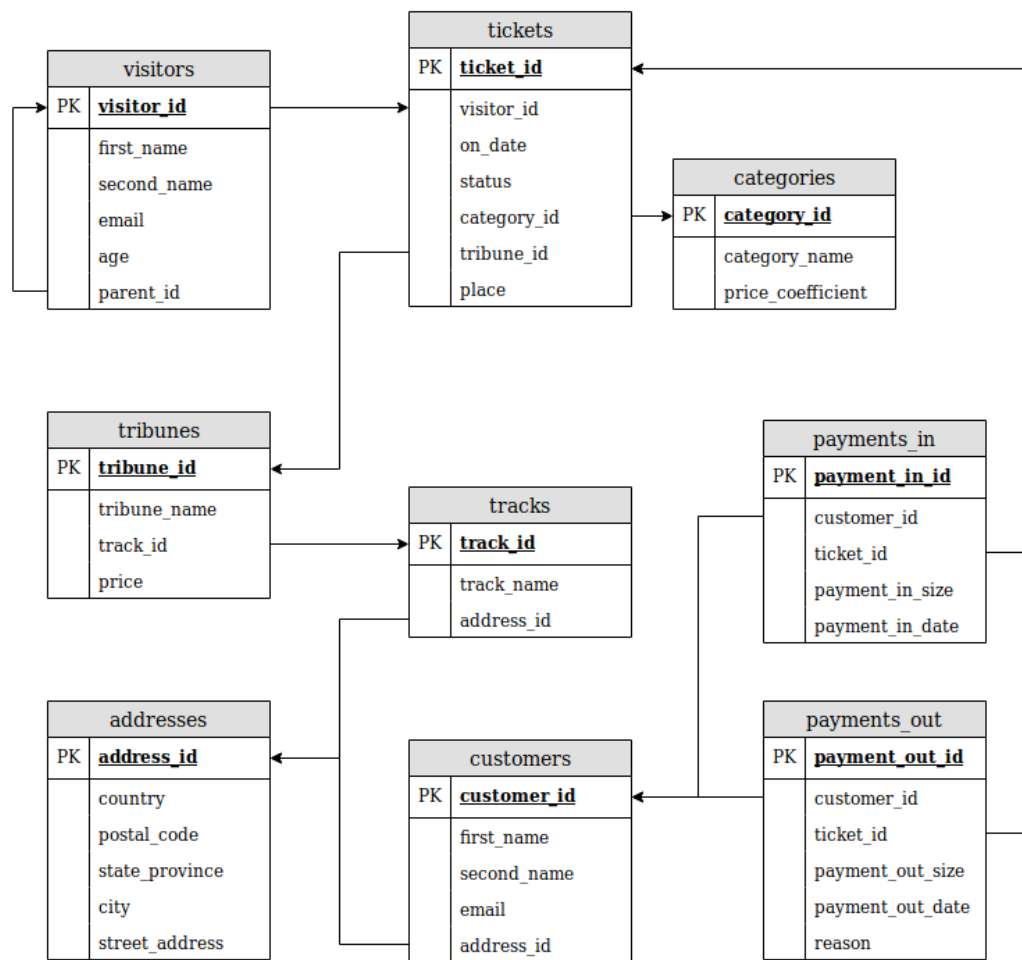
# Идея



- *Чемпионат мира «Формула 1» – не только гонки*
- *Большой объём данных*
- *Пример: система продажи билетов*

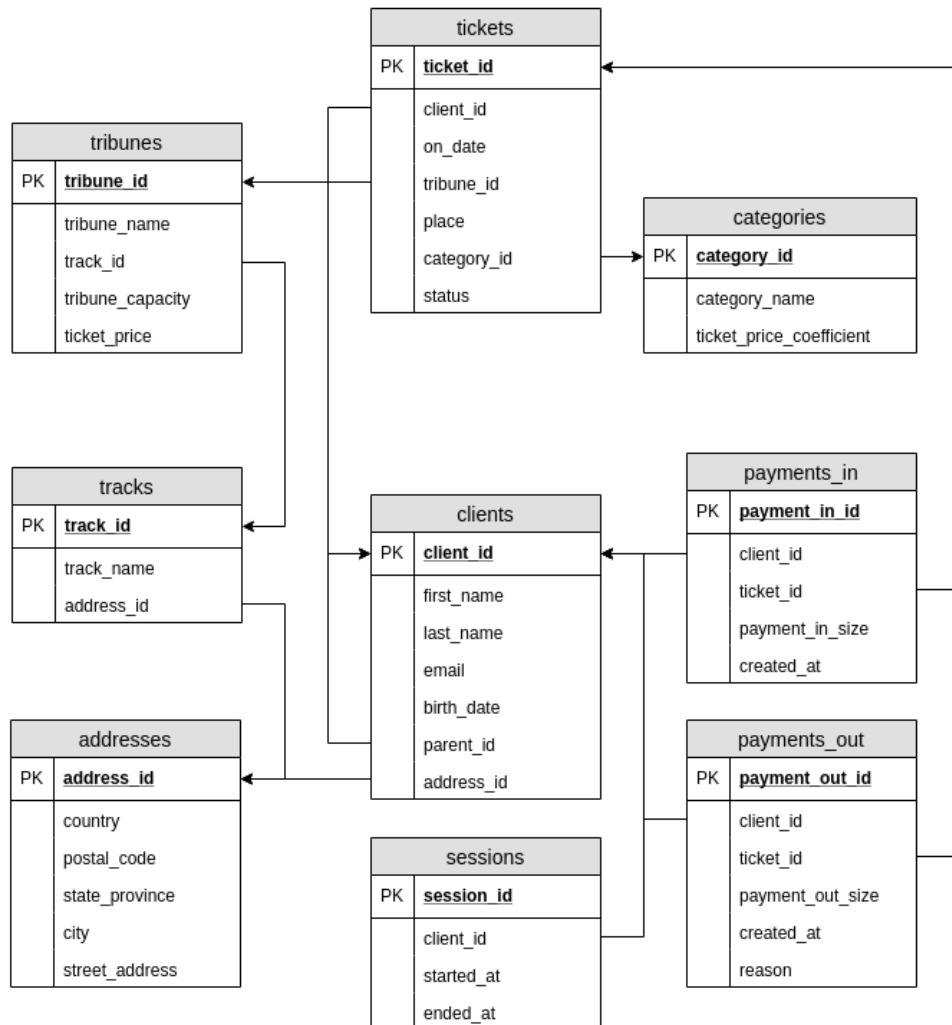


# Схема данных



- *Как определить, что места на конкретной трибуне на конкретную дату закончились?*
- *Может ли visitor быть customer'ом?*
- *Что означает поле parent\_id в таблице visitors?*
- *Можно ли одним платежом закрывать несколько билетов?*

# Новая схема данных



- *Как определить, что места на конкретной трибуне на конкретную дату закончились?* ●
- *Может ли visitor быть customer'ом?* ●
- *Можно ли одним платежом закрывать несколько билетов?* ●
- *Sessions*

# DAU – Daily Active Users



TEXHOTPEK

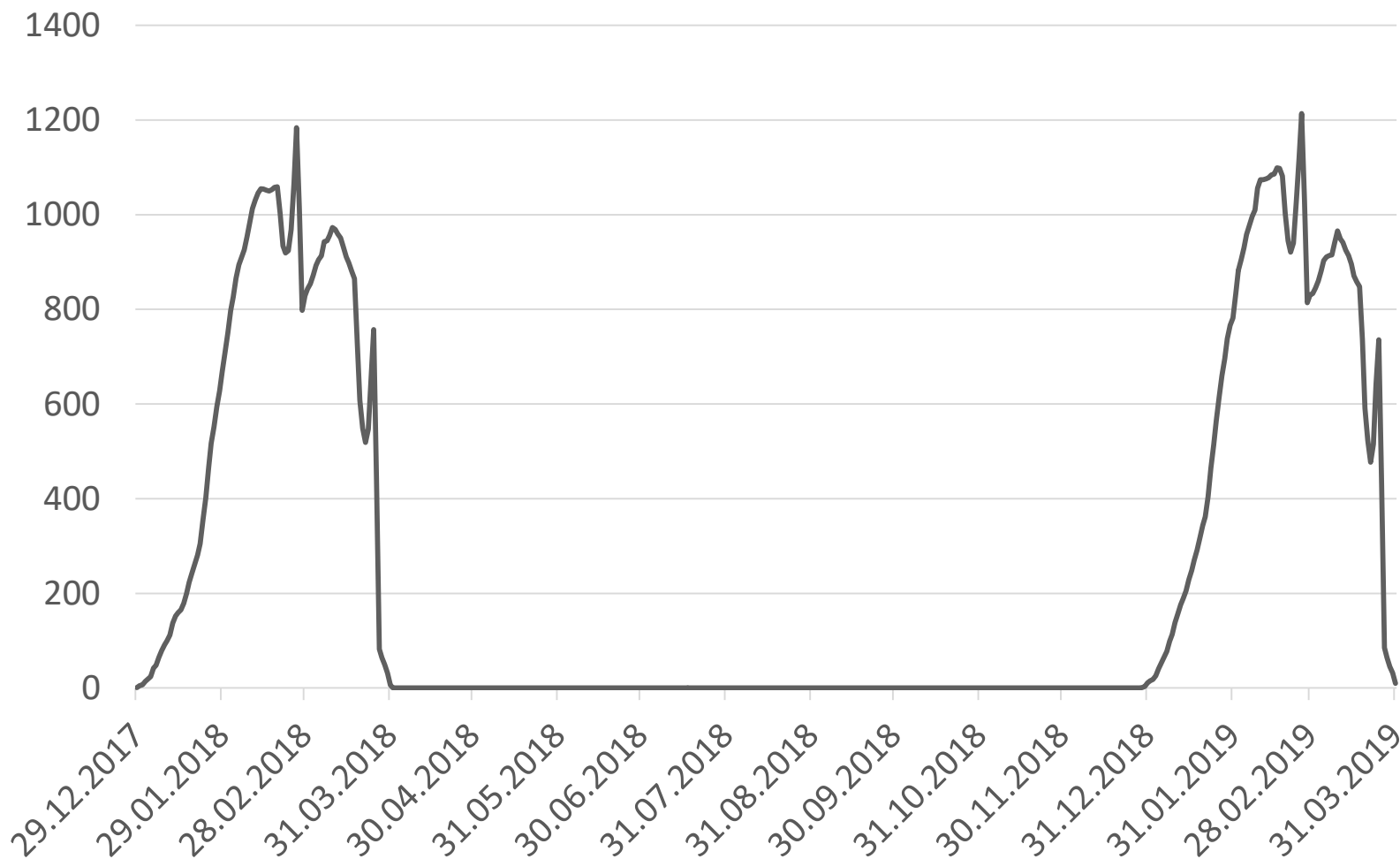
```
1 • USE technotrack_formula_1;
2
3 -- DAU - Daily Active Users
4
5 • EXPLAIN WITH RECURSIVE cte AS
6 (
7     SELECT MIN(CAST(started_at AS DATE)) AS dt FROM sessions
8     UNION ALL
9     SELECT dt + INTERVAL 1 DAY
10    FROM cte
11   WHERE dt + INTERVAL 1 DAY <= (SELECT MAX(CAST(ended_at AS DATE)) FROM sessions)
12 )
13 SELECT cte.dt AS `Date`, COUNT(DISTINCT sessions.client_id) AS `DAU`
14 FROM sessions RIGHT JOIN cte
15 ON CAST(sessions.started_at AS DATE) <= cte.dt AND cte.dt <= CAST(sessions.ended_at AS DATE)
16 GROUP BY cte.dt
17 ORDER BY cte.dt;
```

| # | id | select_type | table      | partitions | type | possible_keys | key  | key_len | ref  | rows  | filtered | Extra                              |
|---|----|-------------|------------|------------|------|---------------|------|---------|------|-------|----------|------------------------------------|
| 1 | 1  | PRIMARY     | <derived2> | NULL       | ALL  | NULL          | NULL | NULL    | NULL | 3     | 100.00   | Using temporary; Using filesort    |
| 2 | 1  | PRIMARY     | sessions   | NULL       | ALL  | NULL          | NULL | NULL    | NULL | 12... | 100.00   | Using where; Using join buffer ... |
| 3 | 2  | DERIVED     | sessions   | NULL       | ALL  | NULL          | NULL | NULL    | NULL | 12... | 100.00   | NULL                               |
| 4 | 3  | UNION       | cte        | NULL       | ALL  | NULL          | NULL | NULL    | NULL | 2     | 100.00   | Recursive; Using where             |
| 5 | 5  | SUBQUERY    | sessions   | NULL       | ALL  | NULL          | NULL | NULL    | NULL | 12... | 100.00   | NULL                               |

# *DAU – Daily Active Users*



TEXHOTPEK



# Revenue

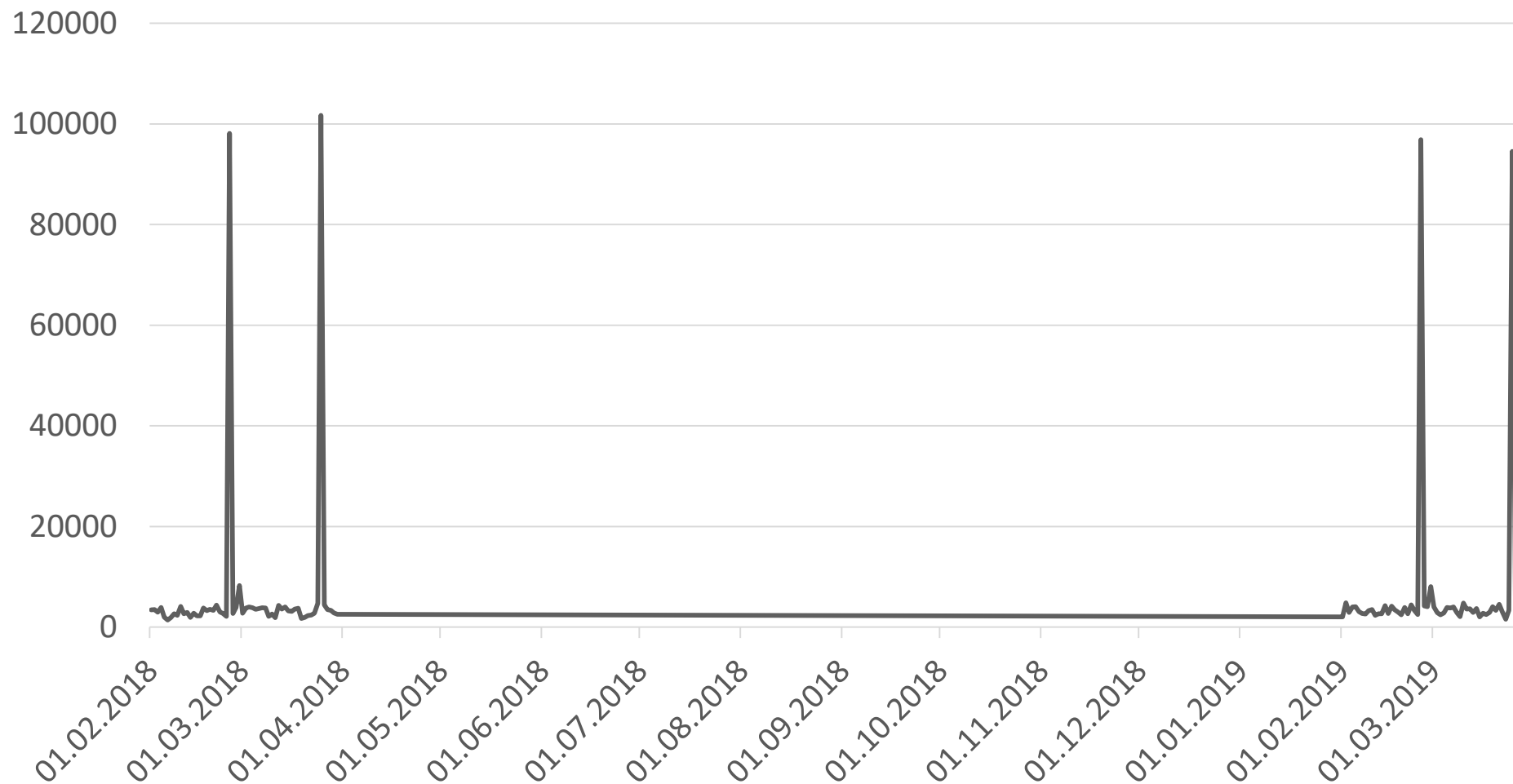


TEXHOTPEK

```
1 • USE technotrack_formula_1;
2
3 EXPLAIN SELECT CAST(created_at AS DATE) AS dt, SUM(payments_in.payment_in_size)
4 FROM payments_in
5 GROUP BY dt;
```

| # | id | select_type | table       | partitions | type | possible_keys | key  | key_len | ref  | rows | filtered | Extra           |
|---|----|-------------|-------------|------------|------|---------------|------|---------|------|------|----------|-----------------|
| 1 | 1  | SIMPLE      | payments_in | NULL       | ALL  | NULL          | NULL | NULL    | NULL | 6000 | 100.00   | Using temporary |

# Revenue





# PPU – Percentage of Paying Users



TEXHOTPEK

```
1 • USE technotrack_formula_1;
2
3 -- PPU – Percentage of Paying Users – доля платящей аудитории относительно DAU
4
5 • EXPLAIN WITH RECURSIVE cte AS (
6     SELECT MIN(CAST(created_at AS DATE)) AS dt FROM payments_in
7     UNION ALL
8     SELECT dt + INTERVAL 1 DAY
9     FROM cte
10    WHERE dt + INTERVAL 1 DAY <= (SELECT MAX(CAST(created_at AS DATE)) FROM payments_in)
11 ),
12 pu_per_day AS (
13     SELECT CAST(p.created_at AS DATE) AS dt,
14            COUNT(DISTINCT p.client_id) as users_paying
15     FROM payments_in AS p
16     INNER JOIN sessions AS s
17     ON s.client_id = p.client_id
18     WHERE s.started_at <= p.created_at AND p.created_at <= s.ended_at
19     GROUP BY dt
20 ),
```

# PPU – Percentage of Paying Users



```

36  SELECT cte.dt AS `Date`,
37         COALESCE(COALESCE(pu.users_paying, 0) / COALESCE(dau.users_all, 1), 0) AS `PPU – Percentage of Paying Users`
38  FROM cte
39  LEFT JOIN pu_per_day AS pu ON cte.dt = pu.dt
40  LEFT JOIN dau_per_day AS dau ON cte.dt = dau.dt;

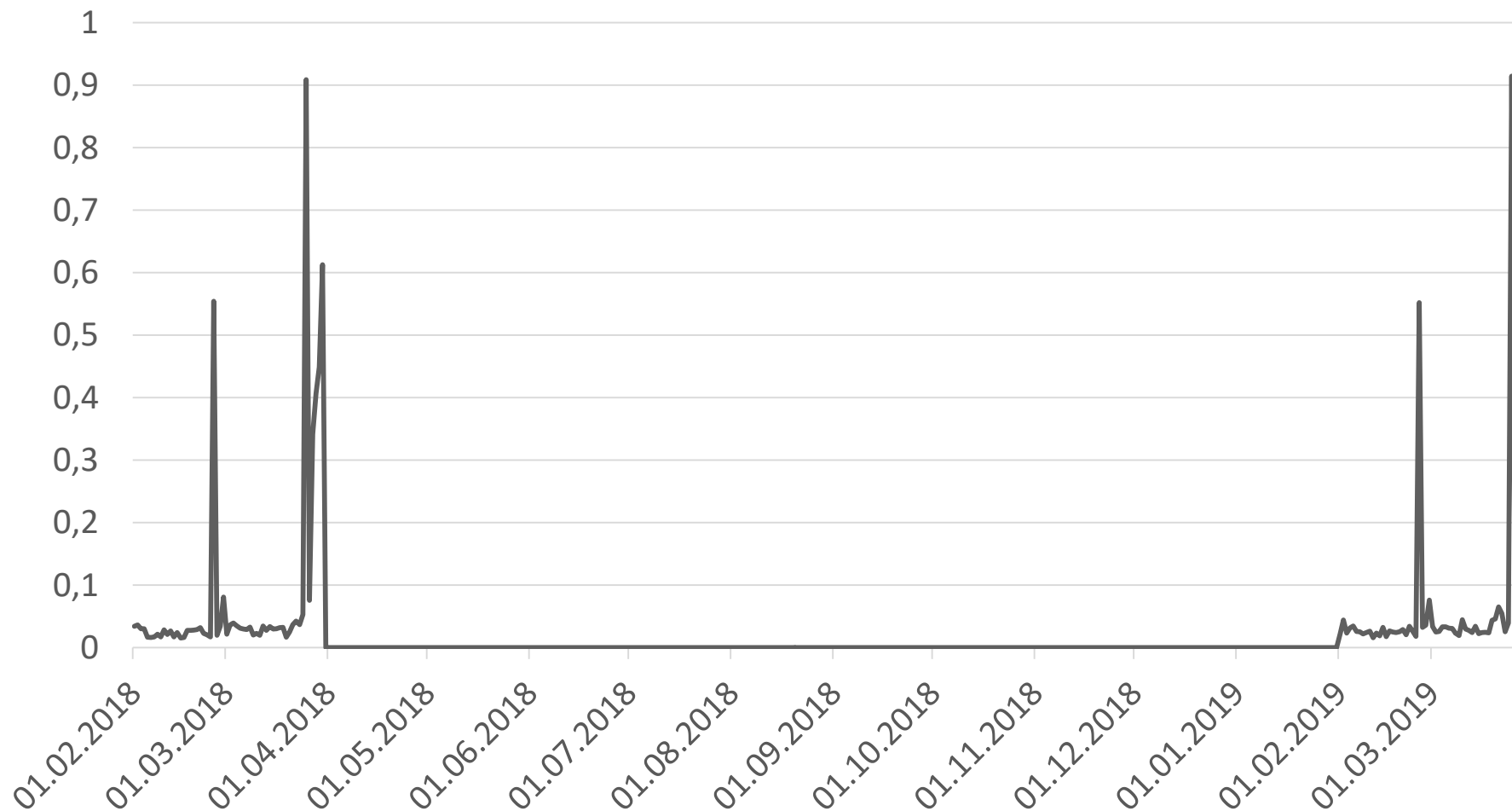
```

| #  | id | select_type | table       | partitions | type | possible_keys  | key      | key_len | ref    | rows  | filtered | Extra                              |
|----|----|-------------|-------------|------------|------|----------------|----------|---------|--------|-------|----------|------------------------------------|
| 1  | 1  | PRIMARY     | <derived2>  | NULL       | ALL  | NULL           | NULL     | NULL    | NULL   | 3     | 100.00   | NULL                               |
| 2  | 1  | PRIMARY     | <derived6>  | NULL       | ref  | <auto_key0>    | <a... 4  |         | cte... | 35    | 100.00   | NULL                               |
| 3  | 1  | PRIMARY     | <derived7>  | NULL       | ref  | <auto_key0>    | <a... 4  |         | cte... | 364   | 100.00   | NULL                               |
| 4  | 7  | DERIVED     | <derived8>  | NULL       | ALL  | NULL           | NULL     | NULL    | NULL   | 3     | 100.00   | Using temporary; Using filesort    |
| 5  | 7  | DERIVED     | sessions    | NULL       | ALL  | NULL           | NULL     | NULL    | NULL   | 12... | 100.00   | Using where; Using join buffer ... |
| 6  | 8  | DERIVED     | sessions    | NULL       | ALL  | NULL           | NULL     | NULL    | NULL   | 12... | 100.00   | NULL                               |
| 7  | 9  | UNION       | cte         | NULL       | ALL  | NULL           | NULL     | NULL    | NULL   | 2     | 100.00   | Recursive; Using where             |
| 8  | 11 | SUBQUERY    | sessions    | NULL       | ALL  | NULL           | NULL     | NULL    | NULL   | 12... | 100.00   | NULL                               |
| 9  | 6  | DERIVED     | p           | NULL       | ALL  | fk_payment...  | NULL     | NULL    | NULL   | 6000  | 100.00   | Using filesort                     |
| 10 | 6  | DERIVED     | s           | NULL       | ref  | fk_sessions... | fk_... 4 |         | tec... | 5     | 11.11    | Using where                        |
| 11 | 2  | DERIVED     | payments_in | NULL       | ALL  | NULL           | NULL     | NULL    | NULL   | 6000  | 100.00   | NULL                               |
| 12 | 3  | UNION       | cte         | NULL       | ALL  | NULL           | NULL     | NULL    | NULL   | 2     | 100.00   | Recursive; Using where             |
| 13 | 5  | SUBQUERY    | payments_in | NULL       | ALL  | NULL           | NULL     | NULL    | NULL   | 6000  | 100.00   | NULL                               |

# *PPU – Percentage of Paying Users*



TEXHOTPEK



# *План масштабирования*



## *Вертикальное масштабирование*

## *Горизонтальное масштабирование*

- *Функциональное секционирование: разделить addresses трасс и клиентов*
- *Репликация платежей и билетов*
- *Шардирование таблиц tickets, clients, addresses, payments\_in*

# План масштабирования



## *Обратное масштабирование*

- *После прохождения этапа чемпионата данные о нём удаляются из `tickets`, `clients` (и связанные `addresses`), `payments_in` и `payments_out`*
- *Данные о трассе в `tribunes` архивируем*

# *Система продажи билетов на этапы чемпионата мира «Формула 1»*

---



ТЕХНОТРЕК

*Спасибо за внимание!*