МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ
МГПУ

# Практическое задание 10. Аналитика с использованием сложных типов данных. Поиск и анализ продаж.

Практическое задание 10.
Аналитика с использованием сложных типов данных. Поиск и анализ продаж.

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

Руководитель отдела продаж выявил проблему: у отдела продаж нет простого способа найти клиента. К счастью, вы вызвались создать проверенную внутреннюю поисковую систему, которая сделает всех клиентов доступными для поиска по их контактной информации и продуктам, которые они приобрели в прошлом:

1. Используя таблицу **customer_sales**, создайте доступное для поиска представление с одной записью для каждого клиента. Это представление должно быть отключено от столбца **customer_id** и доступно для поиска по всей базе данных, что связано с этим клиентом:

- **имя,**
- **адрес электронной почты,**
- **телефон,**
- **приобретенные продукты.**

Можно также включить и другие поля.

Практическое задание 10.
Аналитика с использованием сложных типов данных. Поиск и анализ продаж

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

**2.** Создайте доступный для поиска индекс, созданного вами ранее представления.

**3.** У кулера с водой продавец спрашивает, можете ли вы использовать свой новый поисковый прототип, чтобы найти покупателя по имени Дэнни, купившего скутер Bat. Запросите новое представление с возможностью поиска, используя ключевые слова «Danny Bat». Какое количество строк вы получили?

**4.** Отдел продаж хочет знать, насколько часто люди покупают скутер и автомобиль. Выполните перекрестное соединение таблицы продуктов с самой собой, чтобы получить все отдельные пары продуктов и удалить одинаковые пары (например, если название продукта совпадает). Для каждой пары выполните поиск в представлении, чтобы узнать, сколько клиентов соответствует обоим продуктам в паре. Можно предположить, что выпуски ограниченной серии можно сгруппировать вместе с их аналогом стандартной модели (например, Bat и Bat Limited Edition можно считать одним и тем же скутером).

Практическое задание 10. **Решение**
Аналитика с использованием сложных типов данных. Поиск и анализ продаж

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

1. Создать материализованное представление для таблицы **customer_sales:**

```
CREATE MATERIALIZED VIEW customer_search AS (
        SELECT
                customer_json -> 'customer_id' AS customer_id, customer_json,
                to_tsvector('english', customer_json) AS search_vector
        FROM customer_sales
);
```

2. Создать индекс **GIN** в представлении:

```
CREATE INDEX customer_search_gin_idx ON customer_search USING GIN(search_vector);
```

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

**3.** Выполнить запрос, используя новую базу данных с возможностью поиска:

```
SELECT
      customer_id,
      customer_json
FROM customer_search
WHERE search_vector @@ plainto_tsquery('english', 'Danny Bat');
```

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

**4.** Вывести уникальный список скутеров и автомобилей (и удаление ограниченных выпусков) с помощью **DISTINCT**:

```
SELECT DISTINCT
      p1.model,
      p2.model
FROM products p1
      LEFT JOIN products p2 ON TRUE
WHERE p1.product_type = 'scooter'
      AND p2.product_type = 'automobile'
      AND p1.model NOT ILIKE '%Limited
Edition%';
```

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

**5.** Преобразование вывода в запрос:

```
SELECT DISTINCT
plainto_tsquery('english', p1.model) &&
plainto_tsquery('english', p2.model)
FROM products p1
LEFT JOIN products p2 ON TRUE
WHERE p1.product_type = 'scooter'
AND p2.product_type = 'automobile'
AND p1.model NOT ILIKE '%Limited Edition%';
```

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

**6.** Запрос базы данных, используя каждый из объектов **tsquery**, и подсчитать вхождения для каждого объекта:

```
SELECT
        sub.query,
        (
                SELECT COUNT(1)
                FROM customer_search
                WHERE customer_search.search_vector @@ sub.query)
FROM    (
        SELECT DISTINCT
                plainto_tsquery('english', p1.model) &&
                plainto_tsquery('english', p2.model) AS query
        FROM products p1
        LEFT JOIN products p2 ON TRUE
        WHERE p1.product_type = 'scooter’
        AND p2.product_type = 'automobile’
        AND p1.model NOT ILIKE '%Limited Edition%’
        ) sub
ORDER BY 2 DESC;
```