Мануал по SQL для маркетинговой аналитики и перфоманса

# 1. Введение

SQL (Structured Query Language) — язык для управления и анализа данных в реляционных базах. В маркетинговой аналитике и перфомансе он используется для извлечения, трансформации и анализа данных о пользователях, рекламных кампаниях и бизнес-показателях.

# 2. Основы SQL для аналитика

## 2.1. Базовые конструкции

* SELECT — выборка данных
* FROM — таблица
* WHERE — фильтрация строк
* GROUP BY — группировка
* HAVING — фильтрация групп
* ORDER BY — сортировка
* JOIN — объединение таблиц (INNER, LEFT, RIGHT, FULL)

## 2.2. Типы данных

* Числовые (INT, FLOAT, DECIMAL) — для расчётов
* Строковые (VARCHAR, TEXT) — для категорий, тегов
* Даты и время (DATE, TIMESTAMP) — для анализа по периодам
* JSON — для хранения сложных структур (например, событий или свойств пользователя)

# 3. Маркетинговая аналитика: ключевые метрики и SQL-примеры

## 3.1. Метрики

|  |  |  |
| --- | --- | --- |
| **Метрика** | **Описание** | **Формула (SQL-подход)** |
| CPC | Cost Per Click — стоимость за клик | SUM(cost) / SUM(clicks) |
| CPM | Cost Per Mille — стоимость за 1000 показов | (SUM(cost) / SUM(impressions)) \* 1000 |
| CTR | Click Through Rate — кликабельность | SUM(clicks) / SUM(impressions) |
| CPA | Cost Per Action — стоимость за действие | SUM(cost) / SUM(actions) |
| ROI | Return on Investment — рентабельность | (SUM(revenue) - SUM(cost)) / SUM(cost) |
| ROAS | Return on Ad Spend — возврат на рекламу | SUM(revenue) / SUM(cost) |
| ROMI | Return on Marketing Investment — учитывает маркетинговые бюджеты и доходы | Аналог ROI |
| LTV | Lifetime Value — пожизненная ценность клиента | Суммирование дохода от клиента за весь период |
| Churn Rate | Процент оттока клиентов | COUNT(churned\_customers) / COUNT(total\_customers) |

## 3.2. Пример подсчёта CTR и CPC

SELECT  
 campaign\_id,  
 SUM(impressions) AS impressions,  
 SUM(clicks) AS clicks,  
 SUM(cost) AS cost,  
 ROUND(SUM(clicks)::decimal / NULLIF(SUM(impressions), 0), 4) AS ctr,  
 ROUND(SUM(cost)::decimal / NULLIF(SUM(clicks), 0), 2) AS cpc  
FROM ad\_stats  
GROUP BY campaign\_id;

## 3.3. Когортный анализ (Retention Curves)

Идея: Группировать пользователей по дате их первого действия (регистрации, покупки) и отслеживать удержание по дням/неделям.

WITH cohorts AS (  
 SELECT  
 user\_id,  
 MIN(DATE(event\_date)) AS cohort\_date  
 FROM user\_events  
 GROUP BY user\_id  
),  
events AS (  
 SELECT  
 u.user\_id,  
 c.cohort\_date,  
 DATE(e.event\_date) AS event\_day,  
 DATE\_PART('day', DATE(e.event\_date) - c.cohort\_date) AS days\_after\_cohort  
 FROM user\_events e  
 JOIN cohorts c ON e.user\_id = c.user\_id  
)  
SELECT  
 cohort\_date,  
 days\_after\_cohort,  
 COUNT(DISTINCT user\_id) AS active\_users  
FROM events  
WHERE days\_after\_cohort BETWEEN 0 AND 30  
GROUP BY cohort\_date, days\_after\_cohort  
ORDER BY cohort\_date, days\_after\_cohort;

## 3.4. Атрибуция

Основные модели атрибуции:  
- Last Click — последний клик получает 100% конверсии  
- First Click — первый клик получает 100% конверсии  
- Position-based — распределение между первым и последним кликом, остальные — доля  
- Data-driven — модели на основе ML (в сложных системах)

* SQL пример для last-click атрибуции:

WITH conversions AS (  
 SELECT  
 user\_id,  
 conversion\_id,  
 MAX(click\_timestamp) AS last\_click\_time  
 FROM clicks  
 GROUP BY user\_id, conversion\_id  
),  
last\_click\_source AS (  
 SELECT  
 c.conversion\_id,  
 cl.source  
 FROM conversions c  
 JOIN clicks cl ON cl.user\_id = c.user\_id AND cl.click\_timestamp = c.last\_click\_time  
)  
SELECT  
 source,  
 COUNT(conversion\_id) AS conversions  
FROM last\_click\_source  
GROUP BY source;

# 4. Перфоманс в SQL: оптимизация и продвинутые техники

## 4.1. Оптимизация запросов

* Используйте EXPLAIN для анализа плана выполнения запросов
* Добавляйте индексы на колонки, используемые в WHERE, JOIN, GROUP BY
* Фильтруйте данные до агрегаций
* Избегайте SELECT \* — выбирайте только нужные поля
* Используйте CTE (WITH) для читаемости и повторного использования промежуточных данных

## 4.2. Работа с временными таблицами

CREATE TEMP TABLE temp\_active\_users AS  
SELECT user\_id  
FROM user\_events  
WHERE event\_date >= CURRENT\_DATE - INTERVAL '7 days';  
  
SELECT  
 temp\_active\_users.user\_id,  
 purchases.amount  
FROM temp\_active\_users  
JOIN purchases ON purchases.user\_id = temp\_active\_users.user\_id;

## 4.3. Регулярные выражения в SQL

SELECT  
 user\_id,  
 email\_text  
FROM users  
WHERE email\_text ~ '^[A-Za-z0-9.\_%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$';

## 4.4. Парсинг JSON-полей

SELECT  
 user\_id,  
 json\_data ->> 'utm\_source' AS utm\_source,  
 json\_data -> 'metrics' ->> 'clicks' AS clicks  
FROM events  
WHERE (json\_data ->> 'event\_type') = 'click';

# 5. Рекомендации по изучению

* Начните с основ SQL: SELECT, JOIN, GROUP BY
* Изучите функции агрегирования и оконные функции
* Практикуйтесь на реальных данных: рекламных кампаниях, user events
* Изучите инструменты EXPLAIN и индексы для оптимизации
* Разберитесь с JSON и регулярными выражениями для сложных данных
* Освойте основные маркетинговые метрики и их вычисление через SQL
* Познакомьтесь с атрибуционными моделями и когортным анализом

# 6. Полезные ресурсы

* [SQLZoo](https://sqlzoo.net) — интерактивное обучение SQL
* [Mode Analytics SQL Tutorial](https://mode.com/sql-tutorial/) — аналитический подход
* [PostgreSQL Documentation](https://www.postgresql.org/docs/current/functions-json.html) — работа с JSON
* Книги: «SQL for Data Analysis» — аналитический SQL
* Онлайн курсы на Coursera, Udemy по маркетинговой аналитике и перфомансу

---  
  
 С любовью, Дуплей Максим Игоревич ...  
 Самый ахуенный преподаватель у человечества ...