# **Отчётность**

# **Содержание**

# Цели проекта и его данные

* Концептуальное проектирование
* Проектирование реляционной модели
* Развёртывание БД в выбранной СУБД
* Разработка клиентского приложения
* Заключение

**Цели проекта и его данные**  
Цель проекта — разработка базы данных для анализа взаимодействий пользователей с ботами в Telegram, отслеживания рекламы через YouTube, а также учета подписок на каналы.  
Данные включают: информацию о пользователях ботов, видео на YouTube, каналах, подписках и рекламных ссылках.

**Концептуальное проектирование**

**Краткое описание предметной области**  
Система предназначена для управления рекламой, пользователями и взаимодействиями в Telegram и YouTube. Она позволяет отслеживать просмотры, клики, подписки и анализировать эффективность рекламных кампаний.

**Описание процесса построения инфологической модели с обоснованием выделения сущностей и связей**

1. **Выделение сущностей**:
   * **BotUser**: представляет пользователей Telegram, взаимодействующих с ботами.
   * **Bot**: хранит данные о ботах.
   * **AdAnchors**: используется для отслеживания рекламы, связанной с ботами и видео.
   * **Video**: хранит информацию о видеороликах на YouTube.
   * **YoutubeChannel**: представляет YouTube-каналы.
   * **TelegramChannel**: содержит данные о каналах в Telegram.
   * **TelegramSubscription**: учитывает подписки пользователей на Telegram-каналы.
   * **BotMessage**: хранит статистику сообщений от ботов.
2. **Выделение связей**:
   * **BotUser ↔ Bot**: связь через bot\_id, отражает принадлежность пользователя к боту.
   * **BotUser ↔ AdAnchors**: связь через anchor\_id, показывает, через какую рекламу пользователь был привлечен.
   * **AdAnchors ↔ Video**: связь через video\_id, связывает рекламу с видео.
   * **Video ↔ YoutubeChannel**: связь через channel\_id, указывает, к какому каналу принадлежит видео.
   * **TelegramSubscription ↔ BotUser**: связь через bot\_user\_id, связывает подписку с пользователем.
   * **TelegramSubscription ↔ TelegramChannel**: связь через telegram\_channel\_id, отражает, на какой канал оформлена подписка.
   * **BotMessage ↔ Bot**: связь через bot\_id, учитывает статистику сообщений, отправленных ботом.

Каждая сущность и связь выделена на основе требований к анализу данных и управлению взаимодействиями.

**Проектирование реляционной модели**

**Описание процесса перехода к реляционной модели**

**Нормализация данных**:

- Каждая сущность из концептуальной модели преобразована в отдельную таблицу.

**Связи «многие ко многим»**:

- Например, связь между BotUser и TelegramChannel реализована через таблицу TelegramSubscription, содержащую внешние ключи bot\_user\_id и telegram\_channel\_id.

**Наследование**:

- Прямого наследования нет, но **сущности имеют общие атрибуты**: например, id и created\_at есть во многих таблицах. Это можно рассматривать как "неявное наследование" общей структуры.

**5.Развёртывание БД в выбранной СУБД**

См. github

**6. Разработка клиентского приложения**

**Запросы и отчёты**

*SQL-запрос для подсчёта подписчиков канала:*

SELECT COUNT(\*) AS subscriber\_count

FROM TelegramSubscription

WHERE telegram\_channel\_id = 'channel-id';

*Отчёт по эффективности рекламы:*

SELECT AdAnchors.id, AdAnchors.clicks\_count, Video.views\_count

FROM AdAnchors

JOIN Video ON AdAnchors.video\_id = Video.id;

**7. Заключение**  
Проект успешно реализован:

* Разработана база данных для управления рекламой, пользователями и подписками.
* Обеспечена целостность данных.
* Создано тестовое приложение для управления системой.  
  Рекомендации: добавить мониторинг и автоматизацию обновления данных.