

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет социальных наук

Магистерская программа

«Социология публичной сферы и цифровая аналитика»

Болсун Д. Ю., Ковалев Д. И., Молокова О. А., Юлдашев М. Д.

Отчет по курсу «Проектирование баз данных»

По теме:

«Анализ YouTube-каналов продакшен-компании Medium Quality Production»

Москва, 2024

Содержание

Содержание	2
Цели проекта и его данные.	3
Концептуальное проектирование	3
Проектирование реляционной модели	6
Развёртывание БД в SQLite	7
Клиентское приложение	12
Заключение	18

Цели проекта и его данные.

Цель проекта: аналитика популярности YouTube-каналов и доходности проектов продакшен-компании Medium Quality Production.

Задачи проекта:

1. Визуализация основных статистик популярности каналов и доходности проектов.
2. Изучение настроения аудитории каналов.
3. Выявление лидирующих по показателям популярности видео на каналах.
4. Выявление лидирующих по показателям доходности проектов на каналах.
5. Анализ связи между характеристиками видео на каналах и показателями их популярности.

Данные: Данные о каналах, видео и их аудиторной активности были собраны с помощью YouTube API (скрипт для сбора данных был реализован на Python). Недоступные для парсинга данные (о финансовых показателях проектов, статьях расходов и доходов) были сгенерированы с помощью Python-библиотек Faker и random.

Концептуальное проектирование

а. Краткое описание предметной области.

База данных предназначена для хранения данных об аудиторной активности, доходах и расходах YouTube-каналов, входящих в них проектов и содержащихся на них видео.

Фокус сделан на трех каналах: “LABELSMART”, “easycorn” и “LABELCOM”, позиционирующих себя как образовательно-развлекательные YouTube-каналы. В проекты (серии видео по определенной тематике) данных каналов входят: “Мир Смеха”, “Торячие минуты”, “Темные дела”, “Аргументы и факты”, “Ночная история”, “Женский голос”, “В эфире”, “Легенды”, “Сцена и юмор”.

б. Описание процесса построения инфологической модели с обоснованием выделения сущностей и связей.

При создании инфологической модели мы стремились выбрать только те сущности, которые непосредственно отвечают целям проекта и будут использованы для аналитики популярности и доходности YouTube-каналов.

Для построения инфологической модели были использованы сущности, которые можно собрать с помощью YouTube API, а также созданы сущности, представляющие основные финансовые аспекты YouTube-каналов. К сущностям относятся:

- *Channel* (канал).
- *Video* (видео).
- *Category* (категория видео).
- *Project* (проект).
- *Comment* (комментарий).
- *Income* (доход).
- *Cost* (расход).

Где возможно, сущностям и их атрибутам были даны оригинальные названия, используемые YouTube.

В инфологической модели представлено 7 сущностей и 36 атрибутов. Атрибут *CommentCount*, принадлежащий сущности *Video* — вычисляемый, поскольку сущность с комментариями представлена в другой части диаграммы. Слабые сущности отсутствуют.

В инфологической модели представлено 5 связей между сущностями. Между сущностями *Video* и *Category* присутствует связь «многие ко многим»: видео может содержать несколько категорий одновременно (например, *Sports* и *Media_and_Entertainment*). Одна категория может относиться ко многим видео. Наследование отсутствует.

с. ER-диаграмма с комментариями.

Диаграмма в хорошем качестве [по ссылке](#).

Сущности и их атрибуты:

1. Channel. Канал, содержимое которого рассматривается в анализе.

- ID — Уникальный Id канала, созданный Youtube.
- Title — Название канала.
- Description — Текстовое описание канала.
- CreationDate — Дата создания канала.
- SubscriberCount — Количество подписчиков на канале.

2. Video. Видеоролики, содержащиеся на рассматриваемых каналах.

- ID — Уникальный Id видео, созданный Youtube.
- Title — Название видео.
- Description — Текстовое описание видео.
- CreationDate — Дата создания видео.
- Duration — Продолжительность видео (минут).
- LikeCount — Количество лайков.
- CommentCount — Количество комментариев.
- ViewCount — Количество просмотров.
- DislikeCount — Количество дизлайков.

3. Category — Тематическая категория, к которой относится видео.

- ID — Созданные Id для 5 категорий.
- Title — Названия 5 категорий (*Music_and_Arts*, *Sports*, *Education_and_Society*, *Politics_and_Religion*, *Media_and_Entertainment*)

4. Project. Проект или серия видео по определенной тематике. Каждый проект содержит доходы и расходы.

- ID — Созданные Id для 9 проектов.
- Title — Название проектов: “Мир Смеха”, “Горячие минуты”, “Темные дела”, “Аргументы и факты”, “Ночная история”, “Женский голос”, “В эфире”, “Легенды”, “Сцена и юмор”
- Description — Текстовое описание проекта.

- Status — *Убыточный* проект или *Прибыльный*. Считается по соотношению расходов и доходов.

5. Comment. Комментарий, представленный под видео.

- ID — Уникальный Id комментария, созданный Youtube.
- Text — Текст комментария.
- LikeCount — Количество лайков.
- DislikeCount — Количество дизлайков.
- CreationDate — Дата создания комментария.
- Sentiment (Bert) — Тональность комментария (негативный, нейтральный, позитивный), выделенная с помощью Bert.

6. Income. Доход, который получает проект.

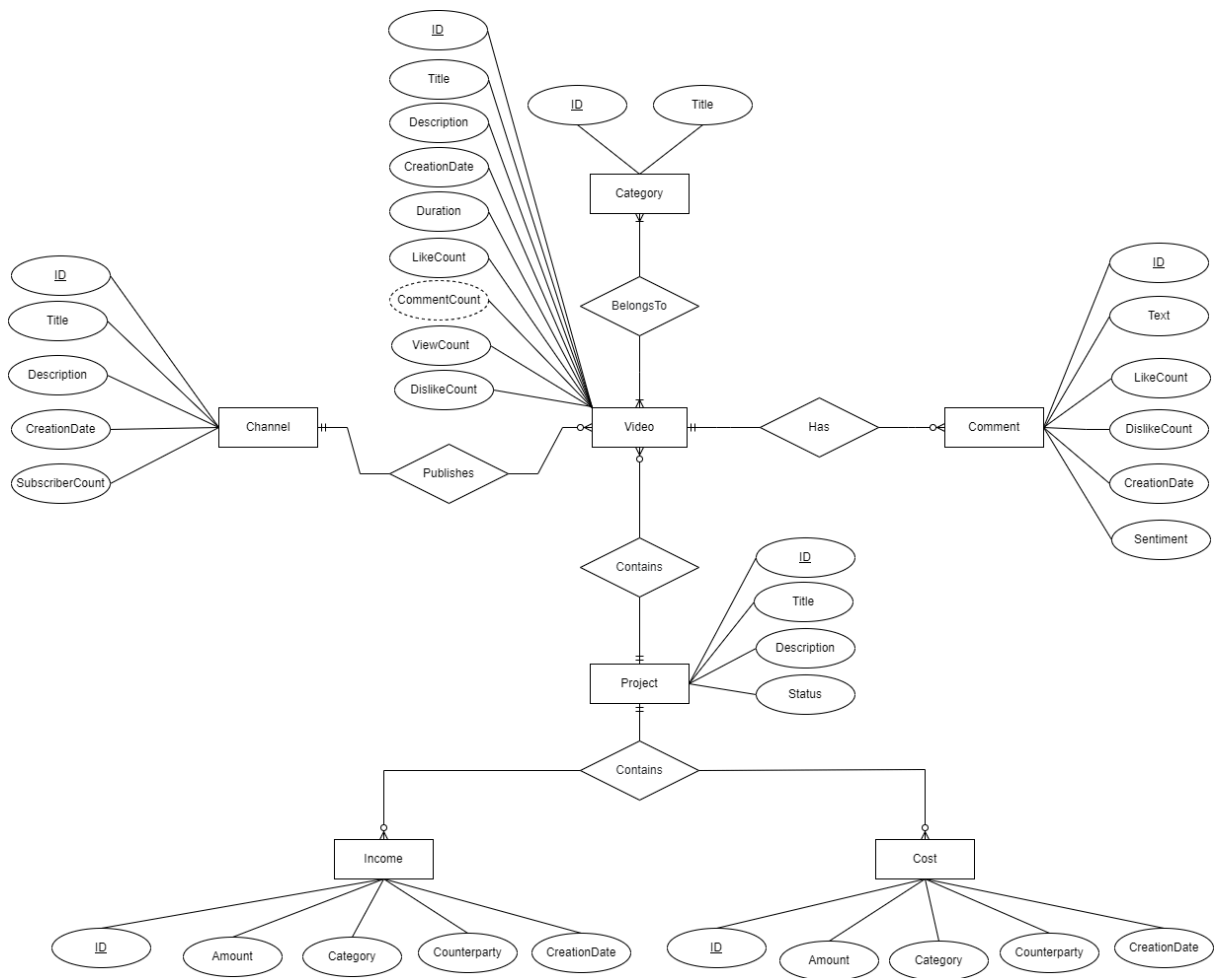
- ID — Созданный Id для 30 статей доходов.
- Amount — Сумма дохода.
- Category — Категория доходов: *Донаты и пожертвования, Премиум-контент* и т. д.

- Counterparty — контрагент (юр. лица): *Щукин Лтд, ИП “Зайцев”* и т. д.

- CreationDate — Дата поступления дохода.

7. Cost. Расход средств, затрачиваемый на проект.

- ID — Созданный Id для 30 статей расходов.
- Amount — Сумма расхода.
- Category — Категория расходов: *Аренда студии, Звукозапись и микширование* и т. д.
- Counterparty — контрагент (юр. лица): *ИП «Ширяева, Туров и Мартынова», ООО «Белова-Евсеев»* и т. д.
- CreationDate — Дата совершения расхода.



Проектирование реляционной модели

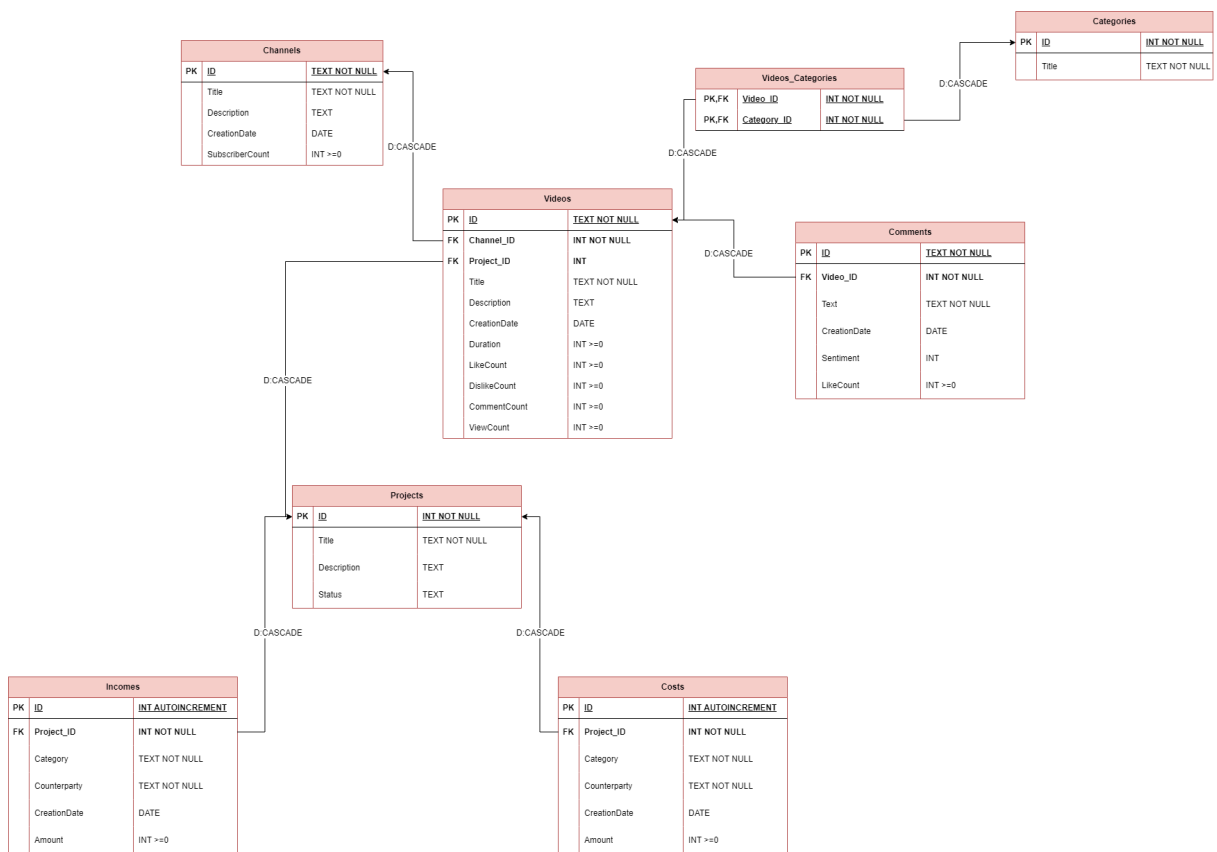
а. Описание процесса перехода к реляционной модели с акцентом на представлении связей «многие ко многим» и наследования.

Переход к реляционной модели производился с учетом следующих аспектов:

- Каждая из 7 сущностей формирует таблицу.
- Каждое отношение «один к одному» или «один ко многим» формирует внешний ключ.
- Отношение «многие ко многим» (между *Video* и *Category*) формирует таблицу *Videos_Catories* + два внешних ключа.
- Типы данных, указанные в модели, выбирались в соответствии с типами данных, полученных при парсинге YouTube.
- В *Costs* и *Incomes* был использован автоинкремент для генерации ID.
- Были заданы ограничения: непустые (NOT NULL) и не-отрицательные (\geq) значения для части атрибутов.
- Особенности ссылочной целостности — каскадное удаление строк из зависимых таблиц при удалении связанных строк в главной таблице.

б. Диаграмма реляционной модели, совмещенная со схемой БД.

Диаграмма в хорошем качестве [по ссылке](#).



Развёртывание БД в SQLite

Для проекта выбрана СУБД SQLite, поскольку она является встроенной, легковесной и не требует настройки сервера. SQLite идеально подходит для небольших приложений и позволяет быстро прототипировать и тестировать базу данных.

Использование в визуализации данных специальных фреймворков Python для построения дашбордов (в частности, Streamlit) снимает необходимость использовать серверный подходы и решения.

Для создания таблиц были использованы DDL-скрипты. В том числе контроль ограничения целостности данных с использованием NOT NULL, CHECK (... >= 0) (проверка значения в столбце) и связи внешнего ключа с использованием REFERENCES ... ON DELETE CASCADE (каскадное удаление значения в других связанных таблицах)

Для сущности Videos:

'''

```

CREATE TABLE IF NOT EXISTS Videos (
    ID TEXT PRIMARY KEY NOT NULL,
    Channel_ID INTEGER NOT NULL,
    Project_ID INTEGER,
    Title TEXT NOT NULL,

```

```
Description TEXT,  
CreationDate DATE,  
Duration INTEGER CHECK (Duration >= 0),  
LikeCount INTEGER CHECK (LikeCount >= 0),  
DislikeCount INTEGER CHECK (DislikeCount >= 0),  
CommentCount INTEGER CHECK (CommentCount >= 0),  
ViewCount INTEGER CHECK (ViewCount >= 0),  
FOREIGN KEY (Channel_ID) REFERENCES Channels (ID) ON DELETE CASCADE  
FOREIGN KEY (Project_ID) REFERENCES Projects (ID) ON DELETE CASCADE  
)  
'''
```

Для сущности Channels:

```
'''  
CREATE TABLE IF NOT EXISTS Channels (  
    ID TEXT PRIMARY KEY NOT NULL,  
    Title TEXT NOT NULL,  
    Description TEXT,  
    CreationDate DATE,  
    SubscriberCount INTEGER CHECK (SubscriberCount >= 0)  
)  
'''
```

Для сущности Comments:

```
'''  
CREATE TABLE IF NOT EXISTS Comments (  
    ID TEXT PRIMARY KEY NOT NULL,  
    Video_ID INTEGER NOT NULL,  
    Text TEXT NOT NULL,  
    CreationDate DATE,  
    LikeCount INTEGER CHECK (LikeCount >= 0),  
    Sentiment INTEGER,  
    FOREIGN KEY (Video_ID) REFERENCES Videos (ID)  
        ON DELETE CASCADE  
)  
'''
```

Для сущности Videos_Categories

'''

```
CREATE TABLE IF NOT EXISTS Videos_Categories (  
    Video_ID INTEGER NOT NULL,  
    Category_ID INTEGER NOT NULL,  
    PRIMARY KEY (Video_ID, Category_ID),  
    FOREIGN KEY (Video_ID) REFERENCES Videos (ID) ON DELETE CASCADE  
    FOREIGN KEY (Category_ID) REFERENCES Categories (ID) ON DELETE CASCADE  
)
```

'''

Для сущности Categories

'''

```
CREATE TABLE IF NOT EXISTS Categories (  
    ID INTEGER PRIMARY KEY NOT NULL,  
    Title TEXT NOT NULL  
)
```

''')

Для сущности Projects

'''

```
CREATE TABLE IF NOT EXISTS Projects (  
    ID INTEGER PRIMARY KEY NOT NULL,  
    Title TEXT NOT NULL,  
    Description TEXT,  
)
```

''')

Для сущности Costs

'''

```
CREATE TABLE IF NOT EXISTS Costs (  
    ID INTEGER PRIMARY KEY AUTOINCREMENT,  
    Project_ID INTEGER NOT NULL,  
    Category TEXT NOT NULL,  
    Counterparty TEXT NOT NULL,
```

```
CreationDate DATE,  
Amount INTEGER CHECK (Amount >= 0),  
FOREIGN KEY (Project_ID) REFERENCES Projects (ID) ON DELETE CASCADE  
)  
""
```

Для сущности Incomes
""

```
CREATE TABLE IF NOT EXISTS Incomes (  
ID INTEGER PRIMARY KEY AUTOINCREMENT,  
Project_ID INTEGER NOT NULL,  
Category TEXT NOT NULL,  
Counterparty TEXT NOT NULL,  
CreationDate DATE,  
Amount INTEGER CHECK (Amount >= 0),  
FOREIGN KEY (Project_ID) REFERENCES Projects (ID) ON DELETE CASCADE  
)  
""
```

В ходе реализации базы данных было решено добавить атрибут «Status» к сущности Projects. Для этого был использован следующий DDL-оператор "ALTER TABLE Projects ADD COLUMN Status TEXT;"

Для дополнительного улучшения качества данных и проверки их целостности использовано два триггера, позволяющие менять статус проекта с убыточного на прибыльный и обратно
""

```
CREATE TRIGGER AfterInsertIncomes  
AFTER INSERT ON Incomes  
FOR EACH ROW  
BEGIN  
UPDATE Projects  
SET Status = CASE  
WHEN (SELECT SUM(Amount) FROM Incomes WHERE Project_ID =  
NEW.Project_ID) >  
(SELECT SUM(Amount) FROM Costs WHERE Project_ID = NEW.Project_ID)  
THEN 'Прибыльный'  
ELSE 'Убыточный'  
END
```

```

WHERE ID = NEW.Project_ID;
END;
'''

-- 2. Триггер для расходов на вставку
CREATE TRIGGER AfterInsertCosts
AFTER INSERT ON Costs
FOR EACH ROW
BEGIN
    UPDATE Projects
    SET Status = CASE
        WHEN (SELECT SUM(Amount) FROM Incomes WHERE Project_ID =
NEW.Project_ID) >
        (SELECT SUM(Amount) FROM Costs WHERE Project_ID = NEW.Project_ID)
        THEN 'Прибыльный'
        ELSE 'Убыточный'
    END
    WHERE ID = NEW.Project_ID;
END;
'''

```

Запросы к БД

Введите запрос:

```
SELECT name, sql FROM sqlite_master WHERE type='trigger';
```

Выполнить запрос

	name	sql
0	AfterInsertIncomes	CREATE TRIGGER AfterInsertIncomes AFTER INSERT ON Incomes FOR EACH ROW BEGIN
1	AfterInsertCosts	CREATE TRIGGER AfterInsertCosts AFTER INSERT ON Costs FOR EACH ROW BEGIN UP

Для проверки работы триггеров был использован DML-оператор INSERT

```

'''
INSERT INTO Incomes (ID, Project_ID, Category, Counterparty, CreationDate, Amount)
VALUES (31, 3, 'Продажа цифрового контента', 'Андреев Лимитед', '2019-11-09', 1527410);
'''

```

Примеры DML-оператора SELECT

```

'''
SELECT Projects.Title, COUNT (Videos.ID) FROM Projects JOIN Videos on
Videos.Project_ID = Projects.ID GROUP BY Projects.Title;
'''

```

Запросы к БД

Введите запрос:

```
SELECT Projects.Title, COUNT (Videos.ID) FROM Projects JOIN Videos on Videos.Project_ID = Projects.ID GROUP BY Projects.Title;
```

Выполнить запрос

	Title	COUNT (Videos.ID)
0	Аргументы и факты	332
1	В эфире	114
2	Горячие минуты	86
3	Женский голос	333
4	Легенды	74
5	Мир Смеха	76
6	Ночная история	103
7	Сцена и юмор	99
8	Темные дела	122

Клиентское приложение

В рамках данного этапа проекта нашей задачей было создание интерактивного приложения, которое предоставляет статистику по видео, каналам, и доходам авторов на YouTube. Приложение ориентировано на использование базы данных для анализа и визуализации ключевых метрик: лайки, комментарии, доходы и расходы.

Клиентское приложение реализовано с помощью Streamlit, поскольку с его помощью можно превратить скрипты на Python в интерактивные веб-приложения.

Преимущества Streamlit для нашего проекта:

- Streamlit обеспечивает простую интеграцию с Plotly и Pandas и позволяет встраивать интерактивные графики, диаграммы и таблицы без сложной настройки.
- Streamlit идеально подходит для проектов, где основное внимание уделяется визуализации данных.
- Возможность легко интегрироваться с SQLite для извлечения и манипулирования данными.
- Streamlit автоматически обновляет приложение при изменении кода, что упрощает процесс тестирования и внедрения новых функций.

СТРУКТУРА ПРИЛОЖЕНИЯ

Разделы:

- Введение: Описание целей и задач проекта.
- Анализ: Выполнение запросов к базе данных и визуализация данных.
- Результаты: Краткое резюме с ключевыми выводами.

ФУНКЦИОНАЛЬНОСТЬ

Подключение к базе данных:

- Пользователь может выбрать файл базы данных (.db) и подключиться к нему через интерфейс.
- Реализована проверка и обработка ошибок подключения.

Выполнение SQL-запросов:

- Поддерживается выполнение произвольных SQL-запросов (DDL/DML) через текстовое поле.
- Отображение результатов запросов в виде таблиц.

ВИЗУАЛИЗАЦИЯ ДАННЫХ:

Графики и метрики:

- Линейные графики для временной динамики.
- Столбчатые графики для топов.
- Диаграммы рассеяния для анализа зависимостей.
- Круговые диаграммы для распределения доходов и расходов по категориям.
- Интерфейс позволяет интерактивно фильтровать данные (по каналам, годам, категориям).

Для анализа доступны четыре сущности:

- Каналы: Средние метрики по просмотрам, лайкам, дизлайкам и комментариям; динамика показателей; корреляция продолжительности видео с лайками комментариями.
- Видео: Топ-10 видео по ключевым метрикам (просмотры, лайки, комментарии, дизлайки) в рамках выбранного канала и года.
- Комментарии: Топ-10 позитивных и негативных комментариев с наибольшим числом лайков.
- Проекты: Топ-10 прибыльных и убыточных проектов, визуализация доходов и расходов по категориям.

Данные по всем сущностям можно смотреть за 2022-2024 годы. Выбор временного диапазона осуществляется с помощью слайдера.

СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ ПРОЕКТА В STREAMLIT:

Анализ метрик канала YouTube:

- Пользователь выбирает конкретный канал.
- Приложение отображает средние показатели (просмотры, лайки, дизлайки, комментарии).
- Пользователь изучает временную динамику метрик для выбранного канала через интерактивные графики.

Оценка эффективности видео:

- Выбирается канал и годовой диапазон через слайдер.
- Приложение показывает Топ-10 видео по просмотрам, лайкам, комментариям и дизлайкам.
- Пользователь может сравнить успех разных видео по ключевым метрикам.

Изучение настроения аудитории:

- Анализируется динамика тональности комментариев (позитивные, негативные, нейтральные) для определенного канала.

- Пользователь видит графики, отображающие изменения настроения аудитории со временем.

Корреляционный анализ:

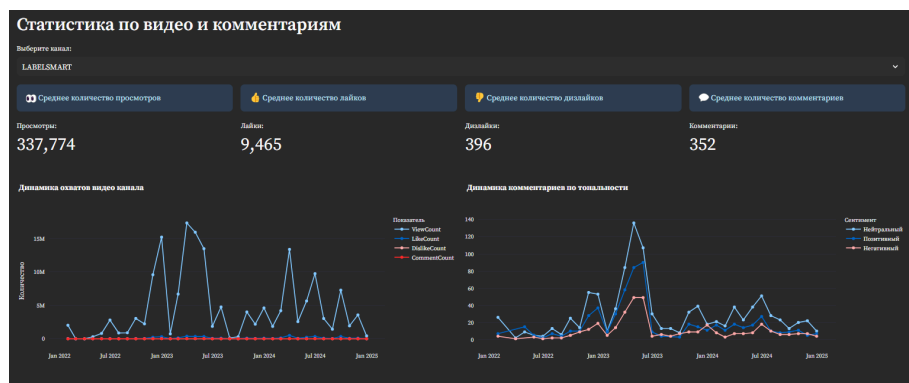
- Пользователь исследует зависимость между продолжительностью видео и количеством комментариев или лайков.
- Приложение предоставляет таблицу с коэффициентами корреляции для разных проектов.

Оценка доходности проектов:

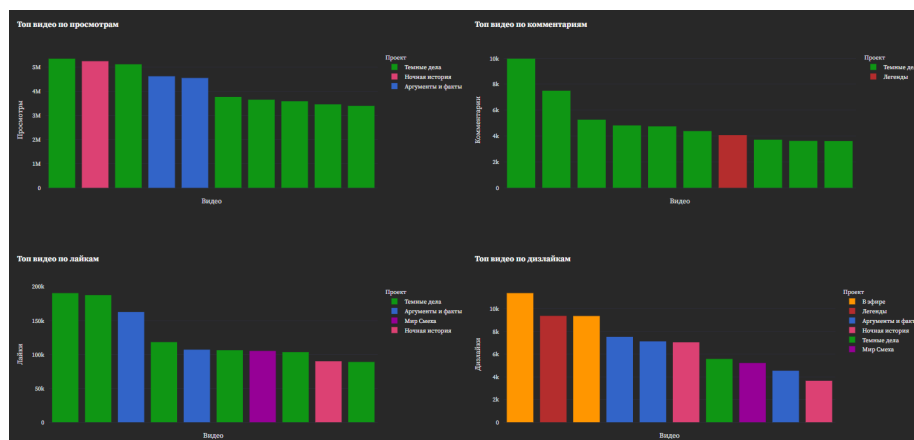
- Выбирается временной диапазон.
- Приложение визуализирует Топ-10 прибыльных и убыточных проектов.
- Пользователь видит распределение доходов и расходов по категориям с помощью круговых диаграмм.
- Эти сценарии позволяют применять проект для анализа данных, стратегического планирования и оценки успеха контента на YouTube.

ПРИМЕРЫ КОМПЛЕКСНОЙ ИНФОГРАФИКИ:

1. Динамика охватов видео каналов и динамика комментариев по тональности:



2. Гистограммы с топ-10 по видео ключевым метрикам:

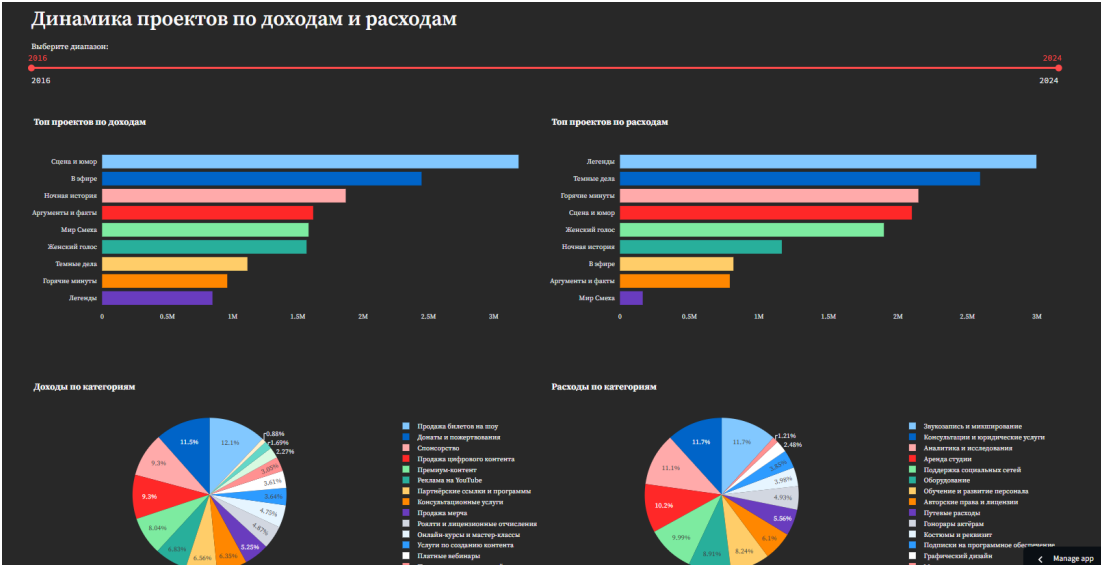


3. Таблицы с топ 10 самыми позитивными и самыми негативными комментариями:

Топ-10 позитивных комментариев		
Text	LikeCount	VideoTitle
1,497 Надпись этот ролик станет катализатором для продвижения дела. Продать огласки уже избыточно и зае	5,800	ЕСТЬ ВЕРСИИ №1 ИСЧЕЗНОВЕНИЕ ДЕТЕЙ Дмитрий Попов, Фест Твой сфин
1,799 Я говорю: ВИ Вы говорите: КА 🧡🧡🧡 Классный выпуск! Спасибо!	890	НАДЕЖДА #18. ЧУВСТВО ВИННЫ. СКАДЧИНОВА & ИРОВИЦЫНА & КАЧМАЛОВ & ДЖАБРАИЛОВ & НИ
1,744 Комментарий и поддержка Элины! Очень интересно и в тему комментарики, дополнения и вопросы. Мое	602	ДЕЛА №9 / СЕМЬЯ КАНИНАЛОВ / СПЕКСИВЕРЫ / Саша Сунин, Эмир Капинов, Рашат1 centaur
402 Хочу увидеть подкаст с Димой / Эмиром и конечно Саша Сунин 🧡 557 комментариев 🧡 / Аня / какие счас	641	ДЕЛА №11 / КЛОУН УБИИЦА / Саша Сунин, Дмитрий Попов, Алексей Стаханов
1,040 Это ролик В тему ещё! Но чтобы Зур был настоящим гостем 🧡🧡🧡	580	ДЕЛА СТОМ-КА СТУДИА ИМА МАМАНИ Айданов, Чабаров, Косов, Косов, Косов, Косов
774 Как здорово, что вы снова не пытаетесь переложить свой череп впер. Фигуры и планы не	499	ДЕЛА №2 / ГОМЕЛСКИЙ МАМАК / Саша Сунин, Павел Давыдов, Рашат1 centaur
450 Мне понравилось, что у Саши была-то совесть! А Лео, почему под его обаяние 🧡 Не выдала ни в ка	389	ДЕЛА №11 / КЛОУН УБИИЦА / Саша Сунин, Дмитрий Попов, Алексей Стаханов
1,142 Так и рад не Юриша! Очень приятно видеть его в обложке ютуба! Одно из любимых видео! А от ку	301	ФАКТЫ #6. Аманат Мусаталиев, Викторина Саламатова, Игорь Дюбраилов, Илья Шенков, Александр И
1,024 Очень классный выпуск, насладились от души 🧡 Дима и Аманат гармонично, а Эмир как всегда свали	246	ФАКТЫ #8: Аманат Мусаталиев & Эмир Капинов & Дима Цалая & Александр Букин & Илья Шенков
1,300 Эх, Кориса явно не знал куда шла, бедолага 🧡 Обычно не задумывался об этом, но на фоне таких работ	174	ДЕЛА №19 / КАННИВАЛ ДЮМАГАЛИНОВ / Фест, Вил, Милу

Топ-10 негативных комментариев		
Text	LikeCount	VideoTitle
226 Как человек который работал в такси волегалась/недала не садиться в машину с другим номером итд!Н	833	Шоу о мистике #2 Ировицына & Арванов & Сергеевич & Ларов & Мойвазидов
1,229 История от Димы конечно, вообще муть. Вот по-на твоя "равнодуш" реальные жертвы бытия в поли	408	ДЕЛА №4 / ДОРОГА СМЕРТИ. БАНДА ГТА / Саша Сунин, Денис Доронин, Стас Кругляцкий
711 При всей моей любви, присутствия Алены в этом выпуске - это абсурд	372	НАДЕЖДА #27. АЛЬБОЛ. ВОДОНАЕВА & ИРОВИЦЫНА & ДЖАБРАИЛОВ & КАЧМАЛОВ
51 15 минут ещё тылаи, а в твое боксе, что выпуск закончился. Прощу Вас, милая!!!!!! Приглашайте ещё раз	297	НАДЕЖДА #25. ЭТОИМ. СМЕРТЯКОВ & НИ & КАЧМАЛОВ & ИРОВИЦЫНА & ДЖАБРАИЛОВ
954 Я смотрю от много видео, подобный разбор действительно видео с Вадим на плане. Уаа, его убил	201	ЕСТЬ ВЕРСИИ №1 Смерть Вадим Савельев Дмитрий Попов Фест Твой сфин
489 Я в 11 классе посмотрела историю! Довольствуйся от истории своего учителя, были очень обидно, не	137	ДЕЛА №10 / ПОВЕЛЕНИЕ АНДРЕЕВЫХ МАНИКА / Саша Сунин, Рашат1 centaur, Олег Парфенов
1,048 Странно представить состояние ролевый и близки ему мерз. А особенно последний диалог	130	ДЕЛА №9 / СЕМЬЯ КАНИНАЛОВ / СПЕКСИВЕРЫ / Саша Сунин, Эмир Капинов, Рашат1 centaur
1,306 Взяли зана Илья. Постепенно стал появляться чужой и незнаком	114	Шоу о мистике #4 Дюбраилов & Арванов & Сергеевич & Ларов & Мойвазидов
650 Я уже персонажи, думаю Зоя не расскажет про бодые детство 🧡🧡	112	НАДЕЖДА #26. СМЫСЛ КИВИН. МУСАТАЛИЕВ & ИРОВИЦЫНА & ДЖАБРАИЛОВ & КАЧМАЛОВ
82 Периодика под "Коллекционные сказы" и подкасты 🧡	106	ФАКТЫ #6. Алексей Савельев, Валентин Сидоров, Илья Братин

4. Топ прибыльных и убыточных проектов, включающий столбчатые и круговые диаграммы:



ПРИМЕРЫ SQL ЗАПРОСОВ:

1. SQL-запрос для вывода Топ-5 видео с наибольшим количеством просмотров:

```
SELECT Title AS "Название видео",
       ViewCount AS "Количество просмотров"
FROM Videos
ORDER BY ViewCount DESC
LIMIT 5;
```

Отчет:

Введите запрос:

```
SELECT Title AS "Название видео",
ViewCount AS "Количество просмотров"
FROM Videos
ORDER BY ViewCount DESC
```

Выполнить запрос

	Название видео	Количество просмотров
0	СКОЛЬКО ЯЗЫКОВ ЗНАЕТ ЧЕБАТКОВ? #натальякарта	13,825,763
1	ДОРОХ ПОПЫТАЛСЯ ПОЦЕЛОВАТЬ ЯНУ КОШКИНУ 🍷 #ивн #кошкина #дорох #иванченко #жс	12,195,473
2	Наталья карта #1 Антон Шастун Шастун, Журавлев, Иванченко	12,116,895
3	ОЛЕСЯ НЕМНОЖКО ПЕРЕПУТАЛА 🍷 #натальякарта #иванченко #тимати #юмор #журавле	12,014,856
4	У ТЕБЯ ОСТАЛОСЬ 4 ГОДА 🍷 #натальякарта #мигель #иванченко #журавлев #астропрогно	11,200,352

Проверка ограничений целостности ☐ Описание триггеров ☐

- SQL-запроса, который анализирует данные из нескольких таблиц и показывает среднюю вовлеченность (отношение лайков, комментариев и просмотров) по каналам и проектам:

SELECT

Channels.Title AS "Канал",

Projects.Title AS "Проект",

COUNT(Videos.ID) AS "Количество видео",

AVG(Videos.ViewCount) AS "Среднее число просмотров",

AVG(Videos.LikeCount) AS "Среднее число лайков",

AVG(Videos.CommentCount) AS "Среднее число комментариев",

*(AVG(Videos.LikeCount) + AVG(Videos.CommentCount)) /
AVG(Videos.ViewCount) * 100 AS "Средняя вовлеченность (%)"*

FROM Videos

JOIN Channels ON Videos.Channel_ID = Channels.ID

LEFT JOIN Projects ON Videos.Project_ID = Projects.ID

GROUP BY Channels.Title, Projects.Title

ORDER BY "Средняя вовлеченность (%)" DESC

LIMIT 10;

Отчет:

Введите запрос:

```
SELECT
Channels.Title AS "Канал",
Projects.Title AS "Проект",
COUNT(Videos.ID) AS "Количество видео".
```

Выполнить запрос

	Канал	Проект	Количество видео	Среднее число просмотров	Среднее число лайков	Среднее число комментариев	C
0	easycom	В эфире	13	1,402,531	67,648.8462	321.0769	
1	easycom	Темные дела	10	509,227	23,406.8	185.6	
2	easycom	Ночная история	15	322,211.3333	14,039.6667	158.6667	
3	easycom	Мир Смеха	9	1,004,954	39,057.6667	165.6667	
4	easycom	Горячие минуты	10	1,619,876.5	62,012.4	204.3	
5	easycom	Сцена и юмор	5	763,808.2	28,780.2	225.6	
6	easycom	Аргументы и факты	172	1,779,097.186	65,357.1337	590.4593	
7	LABELCOM	Темные дела	34	704,128.3529	25,255.8824	105.7059	
8	LABELSMART	Сцена и юмор	57	169,679.3684	5,585.6491	192.6667	
9	easycom	Легенды	8	713,067	23,911.875	221.375	

3. SQL-запрос, который анализирует данные по просмотрам, лайкам и комментариям видео, сгруппированные по каналам, с подсчетом средней вовлеченности (отношение суммы лайков и комментариев к просмотрам):

SELECT

Channels.Title AS "Название канала",

COUNT(Videos.ID) AS "Количество видео",

AVG(Videos.ViewCount) AS "Среднее количество просмотров",

AVG(Videos.LikeCount) AS "Среднее количество лайков",

AVG(Videos.CommentCount) AS "Среднее количество комментариев",

ROUND(

*(SUM(Videos.LikeCount) + SUM(Videos.CommentCount)) * 1.0 /*
SUM(Videos.ViewCount),

3

) AS "Средняя вовлеченность"

FROM Videos

JOIN Channels ON Videos.Channel_ID = Channels.ID

GROUP BY Channels.Title

HAVING SUM(Videos.ViewCount) > 100000

ORDER BY "Средняя вовлеченность" DESC

LIMIT 10;

Отчет:

Введите запрос:						
SELECT Channels.Title AS "Название канала", COUNT(Videos.ID) AS "Количество видео", AVG(Videos.ViewCount) AS "Среднее количество просмотров".						
Выполнить запрос						
	Название канала	Количество видео	Среднее количество просмотров	Среднее количество лайков	Среднее количество комментариев	Сред
0	easycam	500	1,061,726.978	38,217.864	381.42	
1	LABELCOM	350	975,659.2686	30,224.3914	218.88	
2	LABELSMART	489	337,773.6667	9,464.544	352.4315	

Заключение

Проект представляет собой интерактивное приложение, разработанное на базе Streamlit, для анализа и визуализации данных о YouTube-каналах, видео и связанных финансовых показателях. Оно позволяет пользователям исследовать ключевые метрики (просмотры, лайки, комментарии, доходы и расходы), через интерфейс с использованием SQL-запросов и интерактивных графиков.

Данный проект может быть полезен пользователям YouTube, создающим контент. Наше клиентское приложение позволит ему:

- Анализировать эффективность контента на YouTube
- Исследовать вовлеченность аудитории
- Планировать доходы и расходы
- Презентовать результаты анализа для бизнеса или образовательных целей

Таким образом, проект демонстрирует возможности применения технологий визуализации данных и работы с базами данных для решения реальных задач. Особенность нашего проекта - простой интерфейс с возможностью фильтрации данных, выполнения SQL запросов и визуализации метрик.