Использовал диалект PostgreSQL

Задание 1

1.

```
SELECT
        DATE(show_date) AS days,
   SUM(CASE
                WHEN paid_type = 'SVOD' THEN 1
        ELSE 0
        END) AS views_svod,
   SUM(CASE
                WHEN paid_type = 'AVOD' THEN 1
                ELSE 0
        END) AS views_avod
FROM content_watch JOIN content
        ON content_watch.content_id = content.content_id
WHERE
        show_date > (NOW() - INTERVAL '30 DAY')
        AND
        platform IN (10, 11)
GROUP BY days
ORDER BY days
```

2.

Топ 5 единичного контента за август 2021 года:

```
SELECT
        content_watch.content_id,
        COUNT(DISTINCT user_id) AS unique_views
FROM content_watch JOIN content
        ON content_watch.content_id = content.content_id
WHERE
        compilation_id IS NULL
        AND
        date_part('year', show_date) = '2021'          -- указать год
        AND
        date_part('month', show_date) = '8'            -- указать месяц
GROUP BY
        content_watch.content_id
ORDER BY
        unique_views DESC
LIMIT 5
```

Топ 5 сериалов за август 2021 года:

```
SELECT
	compilation_id,
	COUNT(DISTINCT user_id) AS unique_views
FROM content_watch JOIN content
	ON content_watch.content_id = content.content_id
WHERE
	compilation_id IS NOT NULL
	AND
	date_part('year', show_date) = '2021'		-- указать год
	 AND
	date_part('month', show_date) = '8'		-- указать месяц
GROUP BY
	compilation_id
ORDER BY
	unique_views DESC
LIMIT 5
```

3.

```
SELECT DISTINCT user_id
FROM (
	SELECT
		user_id,
		utm_medium,
		lead(utm_medium, 1) OVER w AS next_utm_medium
	FROM content_watch
	WHERE
		show_date >= current_date - INTEGER '1'
		AND
		show_date < current_date
	WINDOW w AS
		 (
		PARTITION BY user_id
		ORDER BY show_date ASC
		 )
	  ) AS t_1
WHERE
	utm_medium ILIKE 'organic'
	AND
	next_utm_medium ILIKE 'referral'
```

Задание 2

1.

Для определения “цепляемости”/”крутости” сериала я бы опирался на следующие метрики:

- **views -** количество просмотров сериала

```
SELECT
        compilation_id,
        COUNT(watch_id) AS views
FROM content_watch JOIN content
        ON content_watch.content_id = content.content_id
WHERE
        compilation_id IS NOT NULL
GROUP BY
   compilation_id
ORDER BY
        views DESC
```

- **number_of_users -** количество смотрящих людей

```
SELECT
        compilation_id,
        COUNT(DISTINCT user_id) AS number_of_users
FROM content_watch JOIN content
        ON content_watch.content_id = content.content_id
WHERE
        compilation_id IS NOT NULL
GROUP BY
        compilation_id
ORDER BY
        number_of_users DESC
```

- **mean_duration -** средняя продолжительность просмотра в процентах. Для расчета необходимы данные о длительности каждой серии (***duration***). Пример расчета для каждой серии:

```
SELECT
        compilation_id,
        content.content_id,
        ROUND(AVG(show_duration::NUMERIC/duration * 100), 2) AS
mean_duration_percentages
FROM content_watch JOIN content
        ON content_watch.content_id = content.content_id
WHERE compilation_id IS NOT NULL
GROUP BY
```

```
        compilation_id,
        content.content_id
```

- **views_above_50_percent** - процент просмотров с длительностью, превышающую половину серии. Для расчета необходимы данные о длительности каждой серии (***duration***). Пример расчета для каждой серии:

```
SELECT
        compilation_id,
        content.content_id,
        COUNT(watch_id) AS views_above_50_percent
FROM content_watch JOIN content
        ON content_watch.content_id = content.content_id
WHERE
        compilation_id IS NOT NULL
        AND
        ROUND((show_duration::NUMERIC/duration * 100), 2) > 50
GROUP BY
        compilation_id,
        content.content_id
```

Также стоит обратить внимание на:

- процент пользователей, посмотревших две и более серий
- процент пользователей, смотревших 2 и более серий подряд
- частоту запросов в поисковой форме
- рейтинги внутри сервиса, imdb и подобные
- упоминания в соцсетях и медиа

В результате я получил бы полную картину, присвоил каждой метрике вес и выставил итоговый рейтинг.

2.

Исходя из имеющихся данных можно выделить сегменты по критериям:

- источники трафика
- платформа
- тип монетизации
- регулярность входов
- единичный контент/сериал

Для выделения сегментов по другим критериям необходимы дополнительные данные:

- география (страна, язык)
- дата регистрации
- соц-дем (пол, возраст)

- регулярность/количество платежей
- характеристика контента (жанр, длительность)

Пример расчета retention:

- за 2018 год
- разбиение по месяцам
- по источнику ‘organic’

```
SELECT
        date_part('year', show_date) AS year,
        date_part('month', show_date) as month,
         /* количество вернувшихся клиентов */
        COUNT(DISTINCT user_id) AS number,
        /* расчет retention */
        ROUND(100 * (COUNT(DISTINCT user_id)::NUMERIC / (
                        /* количество клиентов в начале периода */
                        SELECT COUNT(user_id)::NUMERIC
                        FROM content_watch
                        WHERE
                                /* указываем месяц начала периода */
                                date_part('month', show_date) = '1'
                                AND
                                /* указываем год начала периода */
                                date_part('year', show_date) = '2018'
                                AND
                                /* указываем сегмент */
                                utm_medium ILIKE 'organic'
                        )
                )
         , 1) AS retention_percentages
FROM content_watch
WHERE
        /* выбираем год */
        date_part('year', show_date) = '2018'
        AND
        user_id IN (
        /* список клиентов на начало периода */
                SELECT DISTINCT user_id
                FROM content_watch
                WHERE
                        /* указываем месяц начала периода */
                        date_part('month', show_date) = '1'
                        AND
                        /* указываем год начала периода */
                        date_part('year', show_date) = '2018'
                        AND
                        /* указываем сегмент */
```

```
                utm_medium ILIKE 'organic'
        )
GROUP BY
    year,
    month
```

Задание 3.

Для интерпретации результатов я мог бы использовать t-тест, бутстрап и критерий mann-whitney, а потом сравните p-value. По ссылке ниже мой учебный проект с похожим кейсом.

https://github.com/amilchakov/Data_Analyst/blob/main/Projects/%D0%A1%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B3%D1%80%D1%83%D0%BF%D0%BF%20%D0%B1%D1%83%D1%82%D1%81%D1%82%D1%80%D0%B0%D0%BF%D0%BE%D0%BC%20%D0%B8%20u-%D1%82%D0%B5%D1%81%D1%82%D0%BE%D0%BC.ipynb