1.1

select content\_watch.platform, count (content.paid\_type) as amount

from content\_watch

inner join content on

content\_watch.content\_id = content.content\_id

where (content.paid\_type = 'AVOD' or content.paid\_type = 'SVOD') and (show\_date between '30/09/2018'and '31/10/2018') and (content\_watch.platform = 10 or content\_watch.platform = 11)

group by content\_watch.platform

1.2

select content.compilation\_id, count (content\_watch.user\_id) as amount\_users, MONTH(content\_watch.show\_date) as date\_month, YEAR (content\_watch.show\_date) as date\_year

into #tt1

from content

left join content\_watch on

content\_watch.content\_id = content.content\_id

where content.compilation\_id != NULL

group by YEAR (content\_watch.show\_date), MONTH(content\_watch.show\_date), content.compilation\_id

order by amount\_users desc

select top(5) compilation\_id, date\_month, date\_year

into #tt2

from #tt1

order by amount\_users desc

select content.content\_id, count (content\_watch.user\_id) as amount\_users, MONTH(content\_watch.show\_date) as date\_month, YEAR (content\_watch.show\_date) as date\_year

into #tt3

from content

left join content\_watch on

content\_watch.content\_id = content.content\_id

where content.compilation\_id = NULL

group by YEAR (content\_watch.show\_date), MONTH(content\_watch.show\_date), content.content\_id

order by amount\_users desc

select top(5) content\_id, date\_month, date\_year

into #tt4

from #tt3

order by amount\_users desc

select #tt1.\*, #tt2.\*

from #tt2

left join #tt1

drop table #tt1, #tt2

1.3

select content\_watch.user\_id, content\_watch.show\_date

from content\_watch

where content\_watch.show\_date = (select content\_watch.show\_date from content\_watch where content\_watch.show\_date = DATEADD(day, DATEDIFF(day, +1, GETDATE()), 0) and content\_watch.utm\_medium = 'organic')

< content\_watch.show\_date = (select content\_watch.show\_date from content\_watch where content\_watch.show\_date = DATEADD(day, DATEDIFF(day, +1, GETDATE()), 0) and content\_watch.utm\_medium = 'Referral')

2.1

select content.compilation\_id, count (content\_watch.user\_id) as amount\_users

from content

left join content\_watch on

content\_watch.content\_id = content.content\_id

where content.compilation\_id != NULL and content.episode = 1 and content.episode =

(select max(episode) from content)

group by content.compilation\_id

order by amount\_users desc

2.2

Ретеншн общий

declare @a1 float, @a2 float, @R float

select count (b.watch\_id) as amount\_views, b.user\_id

into #tt8

from [IVI].[dbo].[content] as a

left join [IVI].[dbo].[content\_watch] as b on

b.content\_id = a.content\_id

group by b.user\_id

order by amount\_views desc

set @a1 = (select count(amount\_views) from #tt8)

set @a2 = (select count(amount\_views) from #tt8 where amount\_views > 1)

set @R = @a2/@a1\*100

select @R as 'ретенш, %', @a2 as 'кол-во вернувшихся юзеров', @a1 as 'кол-во юзеров'

drop table #tt8

ретеншн общий и по месяцу октябрь

declare @a11 float, @a12 float, @R1 float, -- общий ретеншн

@a21 float, @a22 float, @R2 float, --ретеншн по дате

@a31 float, @a32 float, @R3 float, --ретеншн по источнику трафика

@a41 float, @a42 float, @R4 float, --ретеншн по монетизации контента

@a51 float, @a52 float, @R5 float-- ретеншн по индентификаторам контента

create table ретеншн (name nvarchar(50), R float, a1 float, a2 float)

select count (b.watch\_id) as amount\_views, b.user\_id

into #ретенш\_общий

from [IVI].[dbo].[content] as a

left join [IVI].[dbo].[content\_watch] as b on

b.content\_id = a.content\_id

group by b.user\_id

order by amount\_views desc

set @a11 = (select count(amount\_views) from #ретенш\_общий)

set @a12 = (select count(amount\_views) from #ретенш\_общий where amount\_views > 1)

set @R1 = @a12/@a11\*100

insert into ретеншн values ('общий', @R1, @a11, @a12)

drop table #ретенш\_общий

select count (b.watch\_id) as amount\_views, b.user\_id

into #ретенш\_месяц

from [IVI].[dbo].[content] as a

left join [IVI].[dbo].[content\_watch] as b on

b.content\_id = a.content\_id

where MONTH(b.show\_date) = 'октябрь'

group by b.user\_id

order by amount\_views desc

set @a21 = (select count(amount\_views) from #ретенш\_месяц)

set @a22 = (select count(amount\_views) from #ретенш\_месяц where amount\_views > 1)

set @R2 = @a22/@a21\*100

insert into ретеншн values ('общий', @R1, @a11, @a12)

drop table #ретенш\_месяц

Ретеншн можно сделать по **дате** (по месяцам или годам, а также вблизи праздников или выходных например). Чтобы увеличить ретенш, нужно, во-первых, сравнить его с ретеншеном этого же периода предыдущего года. Если вырос, значит, что-то делалось правильно, если упал, значит неправильно. Проанализировать, какие действия были успешными (и поменялись ли они вообще с предыдущего года). Далее, для увеличения ретеншена можно предлагать/рекламировать зрителям контент, который выпускается прямо сейчас, а также сезонный контент, например, фильмы про лето или про новый год. Можно делать подборки таких фильмов.

По **источнику трафика**: обычно пользователи, которые остаются, приходят с прямых ссылок. В этом случае пользователям можно предложить сохранить сайт в закладках. Также можно сделать приложение на телефон и desktop приложение, чтобы пользователям было удобнее заходить на сайт (не через браузер).

По **монетизации контента**: посмотреть, в каких случаях пользователи возвращаются. Там, где слабо возвращаются, предложить сменить тип монетизации, например, для avod юзеров предложить подписку. Для svod сменить тариф/тип подписки.

По **индетификаторам контента и сериала**: определить, какие фильмы и сериалы собирают больше всего повторно возвращающихся пользователей. Такие сериалы и фильмы активно продвигать на сайте.

Также ретеншн можно сделать по возрасту, полу, месту проживания, по роду деятельности и увлечениям (школьники, студенты, родители, увлекающиеся искусством). Проанализировать, какой тип монетизации какой группе больше подходит и продвигать этот тип, какие фильмы и сериалы смотрят, какая реклама им нравится и не нравится.

3.

Какие данные понадобятся:

1. Длина рекламных роликов - Lad
2. Частота рекламных роликов - Fad
3. Длина контента - Lc
4. Количество зрителей, начавших смотреть контент - Nstart
5. Количество зрителей, продолживших смотреть контент после первой рекламы – N1ad
6. Количество зрителей, досмотревших контент до конца - Nend

Оценить кол-во рекламы, которое видит пользователь:

Lad \* Fad = L

Узнать, после какой рекламной нагрузки зрители начинают повально отключаться.

Рекламная нагрузка = L/Lc. Придерживаться оптимального значения.

Нужно узнать кол-во пользователей, начавших смотреть контент, кол-во пользователей, продолживших смотреть контент после рекламы.

Доля не отключившихся пользователей = N1ad/Nstart. Посмотреть, после какой максимальной длины первой рекламы начинается спад зрителей. Значит, эта длина – максимально допустимая. Придерживаться оптимальной длины рекламы.

Также узнать, где больше зрителей остаётся: при редких, но длинных рекламах, или при коротких, но частых, и выбрать наиболее подходящий формат.

4.

Может быть неверно выбран регион, в котором будет показываться реклама. Например, в этом регионе намного меньше зрителей и длины просмотров, чем инвентаря. Тогда рекламу просто некому будет показывать. Чтобы уместить большое число инвентаря, нужно будет показывать рекламу слишком часто, что отпугнёт пользователей. А при старой нагрузке реклама не успеет «показаться».

Также слишком короткий период показа инвентаря. Может, стоило его увеличить?

Может быть такое, что по настройкам выбрана какая-то группа людей для показа рекламы. Но в данном регионе может быть слишком мало таких людей. Тогда нужно изменить регион или расширить аудиторию рекламы.