

Case para vaga de Data Scientist / BI

Teste de SQL

Para as perguntas abaixo, escreva um SQL que resulte em uma tabela com duas colunas, uma para cada eixo do gráfico.

Pode ser um SQL padrão compatível com Sql Server, MySql ou Redshift.

1 - Curva de retenção

Dado que temos uma tabela com as características abaixo:

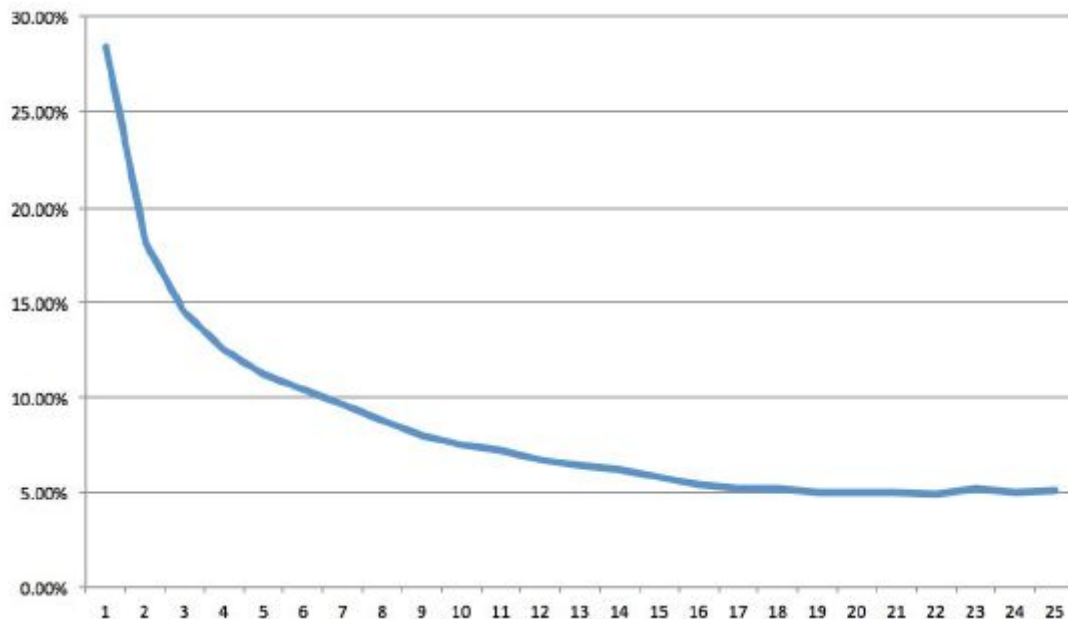
● users

○ user_id - int

○ signed_in_at - datetime

Esta tabela contém todos os logins do usuário na plataforma, podendo haver mais de um login no dia.

Faça um SQL para obter a curva de retenção, que é a quantidade de usuários que ficam após X semanas, conforme exemplo abaixo:



Onde no eixo X temos semanas, e no eixo Y quantidade ou % de usuários.

Este gráfico representa quantas semanas cada usuário permanece na plataforma.

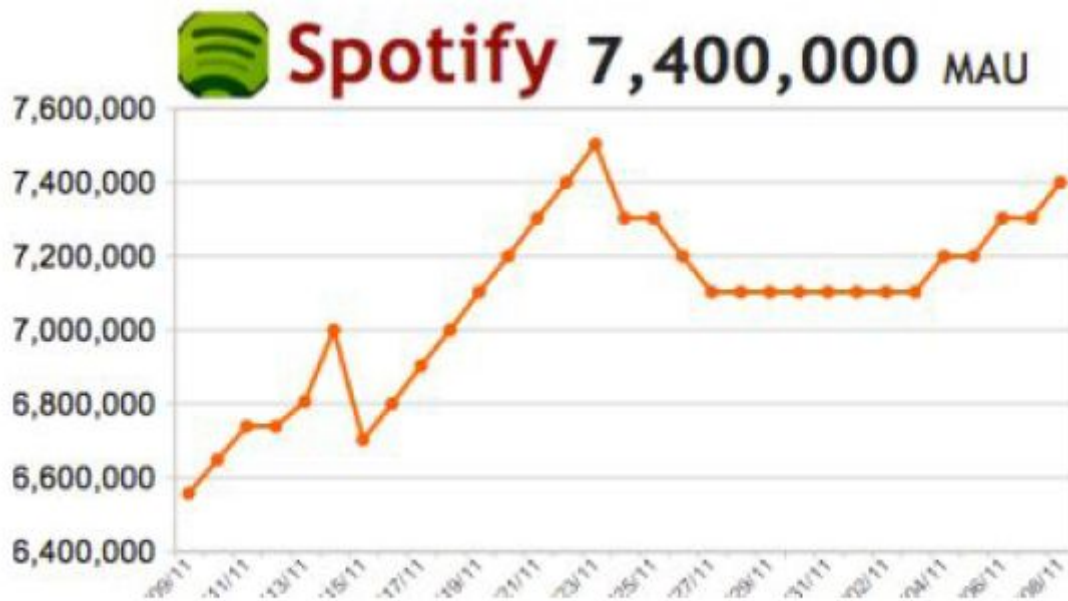
```
WITH tab1 AS (SELECT signed_in_at, DATE_TRUNC('week', signed_in_at) AS wk, user_id
FROM users
ORDER BY 1),
tab2 AS (SELECT tab1.wk, COUNT(DISTINCT tab1.user_id) AS ct
FROM tab1
GROUP BY 1)
```

```
SELECT ROW_NUMBER() OVER (ORDER BY tab2.wk) AS Row, 100*tab2.ct/SUM(tab2.ct) OVER
(ORDER BY tab2.wk)
FROM tab2
ORDER BY 1
```

2 - MAU

Dada a mesma tabela acima, calcule o MAU, ou Monthly Active Users mês a mês.

O gráfico representa quantos usuários usaram a plataforma em cada mês, pelo menos uma vez naquele mês.



Onde o eixo Y o número de usuários e no eixo X o mês.

Onde o eixo Y o número de usuários e no eixo X o mês.

```
WITH tab1 AS (SELECT signed_in_at, DATE_TRUNC('month', signed_in_at) AS month1, user_id
FROM users
ORDER BY 1),
tab2 AS (SELECT tab1.month1, COUNT(DISTINCT tab1.user_id) AS ct
FROM tab1
GROUP BY 1)
```

```
SELECT ROW_NUMBER() OVER (ORDER BY tab2.month1) AS Row, tab2.ct
FROM tab2
ORDER BY 1
```

3 - MAU Delta

Agora, com os dados do gráfico acima, prepare uma outra alternativa que mostra o delta do MAU, ou seja apenas a variação em relação ao mês anterior.

Se em um mês houve aumento de mau, a barra do gráfico naquele mês deve ser positiva e contar quantos usuários a mais tivemos, se houve decréscimo, deve ser negativa da mesma forma.

```
WITH tab1 AS (SELECT signed_in_at, DATE_TRUNC('month', signed_in_at) AS month1, user_id
FROM users
ORDER BY 1),
tab2 AS (SELECT tab1.month1, COUNT(DISTINCT tab1.user_id) AS ct
FROM tab1
GROUP BY 1),
tab3 AS (SELECT tab2.month1 AS month1, tab2.ct AS ct2
FROM tab2
ORDER BY 1)
SELECT tab3.month1, LEAD(tab3.ct2 OVER ORDER BY tab3.month1))-tab3.ct2
FROM tab3
ORDER BY 1
```

Teste aberto

Agora apenas elabore uma resposta com base em suas experiências não precisando ser exata, imaginando como você iria tentar fazer a tarefa.

Não precisa se aprofundar a nível de SQL ou ferramentas, apenas o racional que usaria para chegar ao resultado.

Digamos que alguém da equipe chegou a você com esta pergunta: Me ajuda?

4 - Oferta e Demanda

Temos um Marketplace, o que significa que temos clientes de dois lados, um é o trabalhador, o outro é o usuário. Se tivermos usuários enviando serviço demais a partir de uma origem e não houverem trabalhadores próximos, não conseguiremos fechar estes serviços, por outro lado se tivermos trabalhadores demais em um local e não houver serviço ali, eles podem deixar a plataforma. Neste cenário, como você imagina uma análise de oferta e demanda, onde a empresa gostaria de saber como está o balanceamento de seu marketplace.

Analisar uma tabela onde houvesse a localização do trabalhador e a do serviço. Seria preciso agrupar por localidade e contar quantos trabalhadores existem por cidade e quanto serviço existe por cidade. Seria preciso fazer este acompanhamento semanalmente para verificar o balanço de serviços e trabalhadores por cidade.

5 - Funil de Cadastro

Em nossa plataforma o usuário instala um app em seu celular onde faz o cadastro e fica disponível para pegar trabalho. O ideal é que ele ache nosso aplicativo organicamente ou através de anúncios e o instale e queira fazer o cadastro e utilizar o app, sem a necessidade de alguém o instruindo.

Neste cenário, como você ajudaria o pessoal de Produto a medir como está o funil de cadastro, saber se a experiência está boa, se está melhorando ou piorando a cada nova feature que é lançada e quem sabe propor mudanças em alguma etapa desta jornada do nosso usuário?

Verificar a variação semanal da quantidade de acessos por usuário, a partir do momento em que uma nova feature foi adicionada no aplicativo e observar se houve aumento ou declínio. É preciso ter cuidado na análise, já que correlação não implica em causalidade, então mudanças nesta variável podem ter várias causas (sazonalidade por exemplo). O ideal é antes de lançar novas funções fazer um teste AB para se certificar que aquela mudança realmente melhora o número de acessos do usuário. A mesma análise pode ser feita medindo o tempo total em que o usuário usa o aplicativo por semana e a variação semanal.