Facebook products

Ads

Mobile (use facebook without downloading an app)

Timeline

News Feed

Product data scientist

Technical, type of work way more analytical

Quick analysis change direction of product

Choose whatever toolset

First part is always SQL, 20-30% doing SQL

I notebook

Internship project self-contained thing

Quick analysis, not incredibly complicated things

* **Messaging**
* Watch videos together with friends
* Express through animated effects
* Replies and forwarding
* Payments
* **Workplace**
* Set up and ran separately on FB accounts, makes use of groups, instant messaging, as well as a news feed
* Companies use it internally for collaboration and integration

What types of Friends, discover new friends, negative friending experience

**Facebook metrics**

**Quantity**

1. **Reach** (both paid and organic)

* The number of people content is seen by on FB
* **Referral Traffic**
* Amount of Visitors the Facebook Page gets

1. **Impressions**

* The number of times posts are seen in general (marketing rule of 7: it is necessary to get your customers to see your content multiple times before they take action)

**Quality**

1. **Engagement**

* A sign that people like the content that you are sharing
* Post-clicks, Reactions, Comments and shares
* That includes the number of visitors clicking links on your posts
* **Page Likes and follows**

1. Number of people that follow brand on FB (unlikes, organic likes, paid likes)
2. Video Retention

How to measure feature success (eg. Instagram shopping)

* + Funnels: Visualize people’s progress to conversion through their actions across your Page and website, mobile and desktop, and more.
  + Retention: perhaps it may be useful to measure how many people would continue to use the feature having been first exposed to it
    - Group audience across any number of dimensions
    - Demographics, behaviours, languages
  + Product specific metrics: DAU (daily active users), time spent, level of engagement (click rates)

First, you need to answer the most basic question: are people using the feature? There are a few key metrics you should look at to get a complete answer to that question:

* total number of times people are using the feature (click rate)
* the number of unique users who are using the feature (logins)
* the percentage of your total active users who are using the feature
* the average number of times per day users are using the feature

Dig deeper into the event:

* Instagram tags
* Recommendations
* Instagram shop, Collections
* Shopping from Creators
* Product launches
* Live Shopping

**Deeper Dives into geographic dispersion**

 Get data on iOS and Android user in terms of:  
  
    - number of users in each age  
    - Geo locationns  
    - Income  
    - Mostly accessed apps  
    - Time spent on phone  
    - Number of users on iOS and Android  
  
The age, mostly accessed apps, geolocations, number of users and time spent on phone could tell us a few things:  
  
    - Is there a significant age discrepancies?  
    - Are users on iOS and Android focus on different apps?  
    - Does iOS users travel more frequently than Android users on average in terms of distances and time?  
    - Even the percentage tells iOS has higher %, what about number of users?  
    - Time spent on phone: if the 65% Android users and 30% iOS users spend much less time on the phone, that just means these groups of people may not be aware of different apps that are available on their phones  
  
- How the users get the app  
  
    - In what ways do most users on iOS and Android get the app? Through recommendations? Direct download from App store? Or the phone they get come with the apps?  
    - If the iOS phones come with the app, or put instagram on top of the list in recommendations, but the Android does not, then that could be the problem.  
    - Test recommending apps on both iOS and Android and see response rate.  
  
- Investigate the user feedbacks  
  
    - Is it possible that the user experience for using Instagram on iOS and Android is so different?  
    - Or does it simply because running on Instagram on Android devices is a pain in the ass?

第三个问题：we are recently thinking about publishing a DO NOT DISTURB function, what are you going to do to determine if you are going to publish this feature. 回答： 这是一个典型的AB testing的问题，在做决定前肯定是要做一下AB testing的，这一点我是抓住了，但是我犯错误的是AB testing用什么了做metric, 我提了一些乱七八糟的做survey啊这类，其实我最应该用的是第二个问题的customer adoption，看那个average number of onboarded在有那个DO NOT DISTURB function下是不是更多。我在面试的时候完全没有想到第二道题就是第三道题可以用的内容，把问题和问题之间孤立了，没有想到问题和问题之间也是有联系的，同时呢这个也让我想第二道题面试官想要的答案是不是因为这个product缺少一些feature,所以adoption 减少了。

新人看不到product题是什么，根据评论区猜测一下是：总Comment/DAU 同比去年同一时间增长50%。Why?以下是我自己的思路

首先确定好前提。套用知乎的套路，先问是不是再问为什么。同比去年增长50%是一个普遍的trend还是一时兴起。把时间线拉长，看是突然增加的还是慢慢的增长的以一个trend。

* 突然增加：可能是有重大变化发生。这道题就变成了，有个metrics突然增长50%，why？ => sudden economic reasons (general election), eruption of some unexpected news (natural disasters), quite easy to diagnose
* 慢慢增长：说明我们产品做得越来越好。这道题就变成了，什么features影响了我们的长期增长。=> gradual increase means that we have successfully improved our products

**METRICS**

无论是哪一个都可以以下面的方式展开。对这个metrics本身也要问是什么。这是一个分子分母的metrics。结合aarrr模型看：

* 总用户基数是不是增长？看注册用户增长，referral rate。
* 还是用户的comment和active比率有变化？看retention rate，不同种类activity的metrics。

接下来确定了哪个metrics具体改变之后分内外两个角度看问题：  
内部问题：

* 大化小，simpson paradox：看具体哪个部分有变化：哪个country？哪个market？哪一类demographic用户？哪一类内容的comment？新出现的retention是哪一类内容？哪一类产品？等等，都可以给我们不同的答案。
* 大化小，长期vs短期：可以建立一个模型，Y是DAU或者comment，X是我们能收集的所有的features，train到我们历史数据，看看哪个feature大大影响了我们的Y。这些feature就是短期的metrics，咱们的Y就是一个长期的metrics比如一年的retention。Features有：新产品新功能Binary，用户profile，用户browsing behavior，channel，用户network属性，时间因素等等。
* NLP模型可以用于feature engineering：用户的说话风格，topic 的热度，用户的sentiment变化等等
* 产品本身的变化：比如新功能。看看他们有没有给我们带来显著增长：AA或者AB testing。看使用前使用后短期内变化；看同类型用户，使用和不使用的区别

外部问题

* 市场变化，大事件（covid），竞争对手倒闭。
* **Evaluate the use of introducing a banner FB marketplace** 
  + Want to test if we can successfully influence the users who reach the marketplace to click on sell something and eventually post adds to sell based on criterias
* We can access the effectiveness of the product
  + A/B, (time spent on market place before making an actual transaction)
  + Surveys of whether users find it easier to shop/ asking about whether they have been positively influenced by the introduction…
* Visualization: Histogram to bucket the average time taken between user entering the market place and clicking on the sell something button
* Metrics: average number of unique sellers who click on the sell something button
* Conversion rate
* Randomized control trail

Ads, Mobile, Timeline, News Feed, Messaging

Facebook stories

* The overall goal of the Stories feature is to increase engagement and retention of Facebook users. Similar to Snapchat Stories, the new Facebook feature allows users to share a disappearing story. The hope is that people will find this new format of sharing more fun and engaging, to the extent that people create and share more user-generated content and open the app more frequently. An added bonus for Facebook Stories would be to attract Snapchat’s segment of 30-year-olds and younger and the 40+ segment that Snapchat does not focus on.
* My understanding is that Facebook Stories only works on mobile devices and can be shared in three ways: publicly as a touch-enabled circle above the Facebook News Feed; as a Direct Message to a friend; and as a regular post to the user’s feed

Artist talk with PowerPoint, followed by discussion/ q and a session

* How to measure feature success (eg. Instagram shopping)
  + Funnels: Visualize people’s progress to conversion through their actions across your Page and website, mobile and desktop, and more.
  + Retention: perhaps it may be useful to measure how many people would continue to use the feature having been first exposed to it
    - Group audience across any number of dimensions
    - Demographics, behaviours, languages
  + Product specific metrics: DAU (daily active users), time spent, level of engagement (click rates)

First, you need to answer the most basic question: are people using the feature? There are a few key metrics you should look at to get a complete answer to that question:

* total number of times people are using the feature (click rate)
* the number of unique users who are using the feature (logins)
* the percentage of your total active users who are using the feature
* the average number of times per day users are using the feature

Dig deeper into the event:

* Instagram tags
* Recommendations
* Instagram shop, Collections
* Shopping from Creators
* Product launches
* Live Shopping

Product:  
facebook for business 的 survey/feedback 有客户反映或投诉 facebook for business tools navigation menu 很难用，having difficulties in finding the tools they want to use from the navigation menu. 作为数据分析员，具体讲讲怎么用数据找出客户 difficulties 的原因。怎么解决这个问题？

2 product generalist interview 问了同样的一个问题： 怎么用location 数据？ 从Facebook products (existing products or new product) and advertisting 两方面谈。

* Country level: language
* City level: characteristics, culture…
* Location: activity (work, studying, shopping, movies)=> Recommendations: common recommendations to all potential visitors, others devised from …
* Surrounding advertisement, advertisement related to the activites…
* Friend recommendations: flows, movement of people, recommend people to partake in events in places that they recently visited, provided they didn’t move to a new city or country

问了怎么identify the location is at home or at work or at shopping?

* At home: sleep, weekends, user has any posts that are being tagged as at home, or mentions the phrase “at home”.

Evaluate一个new feature - merchandise shopping：. From 1point 3acres bbs  
这是常见的evaluate feature，让你想一个feature类型的题。  
首先搞清楚Goal & Metrics. FB 的目标是connect people，所以用户engagement很重要，可以用session time，或者retention。另外因为是marketplace，revenue也很重要要考量。  
然后这种买卖的东西要考虑对买方的影响和卖方的影响，就是内部因素vs外部因素。  
首先，我们在没有任何ab testing的情况要通过已有data 判断有没有这个必要：

1. Painkiller：有没有这个pain，比如用户抱怨需要这个（通过NLP）；用户因为没有这个花很多时间；Churn Analysis，用户退出是因为没有这类服务
2. Vitamin：有没有很多人在用这个领域需要我们优化；Cluster找到heavy usage pattern；NLP看看用户谈论的是不是很多；根据goal metrics建supervised模型，看有没有这相关的东西影响我们goal。
3. Me too：竞争对手有没有搞这个，并且提高他们的流量，revenue。
4. 卖方的影响：卖方有没有人提出想做这个，需求大不大
5. 买方的影响：买的人有没有提出想要这个。

然后有了想法就要test。可以ab，但是要考虑social effect，ab比较难以分开。可以找某一类别的产品先做merchandise shopping，然后看他们的反应。算是一个不严谨的ab testing。

1. 内部：merchandise本身销量，浏览量对比同类非merchandise - 对卖家的影响
2. 外部：这个merchandise本身带来的此类产品的销量和人们总体的engagement。可以看view，人们在此类产品浏览总时间，人们在此类产品的总revenue - 对买家和环境的影响

如果卖家赚了更多钱，用户engage增加，我们revenue增加，可以继续扩大test慢慢释放到整个market。

How do you measure the health of Mentions, Facebook’s app for celebrities? How can FB determine if its worth to keep using it?

Hypothesis Testing

The usual line of reasoning is as follows:

1. There is an **initial research hypothesis** of which the truth is unknown.
2. The first step is to state the relevant **null** and **alternative hypotheses**. This is important, as mis-stating the hypotheses will muddy the rest of the process.
3. The second step is to consider the [statistical assumptions](https://en.wikipedia.org/wiki/Statistical_assumption) being made about the sample in doing the test; for example, assumptions about the [statistical independence](https://en.wikipedia.org/wiki/Statistical_independence) or about the form of the distributions of the observations. This is equally important as invalid assumptions will mean that the results of the test are invalid.
4. Decide which test is appropriate, and state the relevant [**test statistic**](https://en.wikipedia.org/wiki/Test_statistic) T.
5. Derive the distribution of the test statistic under the null hypothesis from the assumptions. In standard cases this will be a well-known result. For example, the test statistic might follow a [Student's t distribution](https://en.wikipedia.org/wiki/Student%27s_t_distribution) with known degrees of freedom, or a [normal distribution](https://en.wikipedia.org/wiki/Normal_distribution) with known mean and variance. If the distribution of the test statistic is completely fixed by the null hypothesis we call the hypothesis simple, otherwise it is called composite.
6. Select a significance level (*α*), a probability threshold below which the null hypothesis will be rejected. Common values are 5% and 1%.
7. The distribution of the test statistic under the null hypothesis partitions the possible values of T into those for which the null hypothesis is rejected -- the so-called *critical region* -- and those for which it is not. The probability of the critical region is *α*. In the case of a composite null hypothesis, the maximal probability of the critical region is *α*.
8. Compute from the observations the observed value tobs of the test statistic T.
9. Decide to either reject the null hypothesis in favor of the alternative or not reject it. The decision rule is to reject the null hypothesis H0 if the observed value tobs is in the critical region, and to accept or "fail to reject" the hypothesis otherwise.

**Be able to explain CLM and LLN clearly**

**LLM**

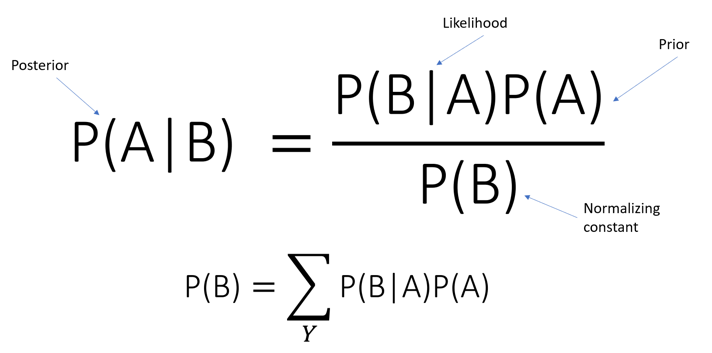
* Convergence of lim (n-> inf) |xn-x|> e=0
* The law of large numbers, in probability and statistics, states that as a sample size grows, its mean gets closer to the average of the whole population.
* The average result obtained from a large number of trails is should be close to the expected value as more trials are performed

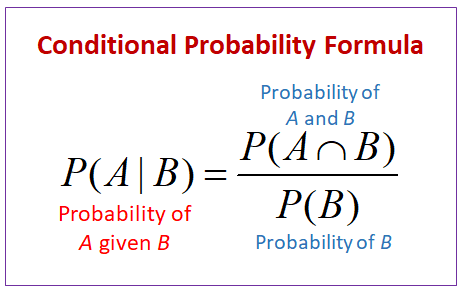
**CLM**

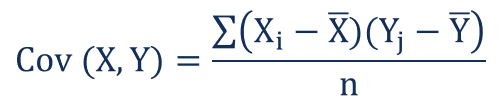
* If you have a population with mean x and standard deviation y
* If you were to take random samples from the population, then the sample mean will be approximately normally distributed with mean as the population mean and estimated standard deviation s/root(n)

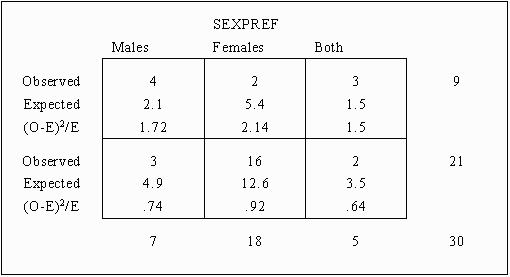
**How to explain confidence interval to business person**

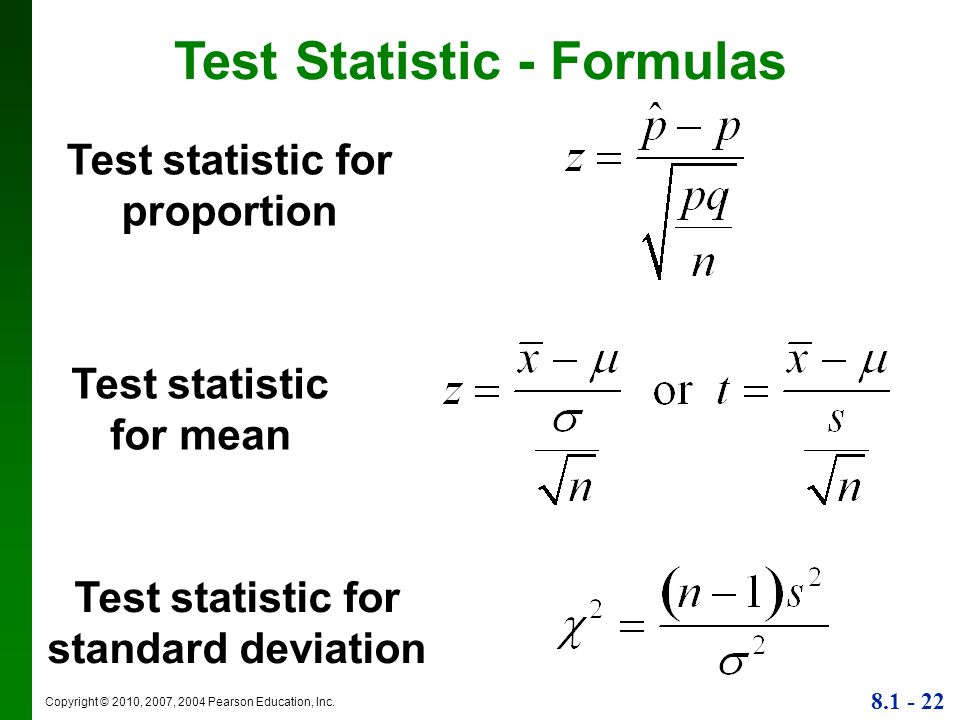
* Confidence interval gives the % probability that an estimated range of possible values in fact includes the actual value being estimated.

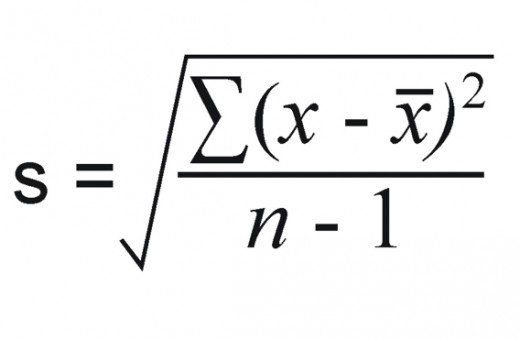








 2-2.16\*2/((0.1)(0.9))



Number of posts/engagements

Test group

Control

In [statistics](https://en.wikipedia.org/wiki/Statistics), the **standard deviation** is a measure of the amount of variation or [dispersion](https://en.wikipedia.org/wiki/Statistical_dispersion) of a set of values.[[1]](https://en.wikipedia.org/wiki/Standard_deviation#cite_note-StatNotes-1) A low standard deviation indicates that the values tend to be close to the [mean](https://en.wikipedia.org/wiki/Mean) (also called the [expected value](https://en.wikipedia.org/wiki/Expected_value)) of the set, while a high standard deviation indicates that the values are spread out over a wider range.

**Useful metrics**

* Impression count
* Click through rate
* Button hover time
* Time spent on page
* Bounce rate on the button’s click through link
* Reactions
* Likes, Comments, Shares
* Post engagement Rate
* Account mentions (organic mentions)
* Reach (potential unique viewers a post could have)
* How would you measure the health of Mentions, Facebook's app for celebrities? How can FB determine if it's worth it to keep using it?  
    
  If a celebrity starts to use Mentions and begins interacting with their fans more, what part of the increase can be attributed to a celebrity using Mentions, and what part is just a celebrity wanting to get more involved in fan engagement?   [6 Answers](https://www.glassdoor.com.hk/Interview/How-would-you-measure-the-health-of-Mentions-Facebook-s-app-for-celebrities-How-can-FB-determine-if-it-s-worth-it-to-keep-QTN_1975752.htm)
* There is a table that tracks every time a user turns a feature on or off, with columns user\_id, action ("on" or "off), date, and time.  
    
  How many users turned the feature on today?  
  How many users have ever turned the feature on?  
  In a table that tracks the status of every user every day, how would you add today's data to it?   [12 Answers](https://www.glassdoor.com.hk/Interview/There-is-a-table-that-tracks-every-time-a-user-turns-a-feature-on-or-off-with-columns-user-id-action-on-or-off-date-QTN_1975753.htm)
* If 70% of Facebook users on iOS use Instagram, but only 35% of Facebook users on Android use Instagram, how would you investigate the discrepancy?   [9 Answers](https://www.glassdoor.com.hk/Interview/If-70-of-Facebook-users-on-iOS-use-Instagram-but-only-35-of-Facebook-users-on-Android-use-Instagram-how-would-you-inves-QTN_1975754.htm)
* How do you measure newsfeed health?   [3 Answers](https://www.glassdoor.com.hk/Interview/How-do-you-measure-newsfeed-health-QTN_1975755.htm)
* If a PM says that they want to double the number of ads in Newsfeed, how would you figure out if this is a good idea or not?   [6 Answers](https://www.glassdoor.com.hk/Interview/If-a-PM-says-that-they-want-to-double-the-number-of-ads-in-Newsfeed-how-would-you-figure-out-if-this-is-a-good-idea-or-not-QTN_1975756.htm)
* We have two options for serving ads within Newsfeed:  
  1 - out of every 25 stories, one will be an ad  
  2 - every story has a 4% chance of being an ad  
    
  For each option, what is the expected number of ads shown in 100 news stories?  
  If we go with option 2, what is the chance a user will be shown only a single ad in 100 stories? What about no ads at all?   [17 Answers](https://www.glassdoor.com.hk/Interview/We-have-two-options-for-serving-ads-within-Newsfeed-1-out-of-every-25-stories-one-will-be-an-ad-2-every-story-has-a-QTN_1975757.htm)Show Less
* How do you map nicknames (Pete, Andy, Nick, Rob, etc) to real names?   [7 Answers](https://www.glassdoor.com.hk/Interview/How-do-you-map-nicknames-Pete-Andy-Nick-Rob-etc-to-real-names-QTN_1975758.htm)
* Facebook sees that likes are up 10% year over year, why could this be?   [7 Answers](https://www.glassdoor.com.hk/Interview/Facebook-sees-that-likes-are-up-10-year-over-year-why-could-this-be-QTN_1975759.htm)
* How many high schools that people have listed on their profiles are real? How do we find out, and deploy at scale, a way of finding invalid schools?

On this date: show feature you have made x years ago

* + Ignore it
  + Reshare it
  + Hide it
* Positive view of sentiment
* Benefits and risk
  + DAU
* Post and share more
* Overall shares not up
* In the test group: people use the feature, despite this…
* Many people use … posts increase overall, not commenting and liking
* More comments and more likes, compared to control group
* Likes, shares, in total
* 50% control=> no feature to share
* =>
* Total likes, total comments
* Eligible to be show

# Write your MySQL query statement below

"""

# you want the final table to have 2 cols, customer id and count no trans

"""

select customer\_id, count(\*) as count\_no\_trans

from visits

left join transactions on visits.visit\_id = transactions.visit\_id

where transaction\_id is null

group by customer\_id

"""

# Write your MySQL query statement below

"""

select name, sum(transactions.amount) as balance

from users

left join transactions on transactions.account=users.account

group by transactions.account

having sum(transactions.amount)>10000

select name, sum(transactions.amount) as balance

from users

left join transactions on users.account = transactions.account

group by transactions.account

having sum(transactions.amount) > 10000

select seller\_id

from sales

where sum(sales.price)

WITH cte AS # must store temporary result in table called cte

(SELECT seller\_id, SUM(price) total\_price

FROM Sales

GROUP BY seller\_id)

SELECT seller\_id

FROM cte

WHERE total\_price IN (SELECT MAX(total\_price) FROM cte)

"""

# Write your MySQL query statement below

"""

WITH cte AS

(Select seller\_id, SUM(price) tprice

FROM sales

group by seller\_id)

Select seller\_id

from cte

WHERE tprice in (select max(tprice) From cte)

"""

# Write your MySQL query statement below

"""

SELECT id, name

FROM students

join departments on departments.id=students.department\_id

where departments.id is null

Select a.id, a.name from students a

left join departments b

on a.department\_id = b.id

where b.id is null

https://leetcode.com/problems/students-with-invalid-departments/submissions/

select id, name from students where department\_id not in (select id from Departments)

https://leetcode.com/problems/students-with-invalid-departments/submissions/

SELECT a.id, a.name FROM students a

left join departments on a.department\_id=departments.id

where departments.id is null

https://leetcode.com/problems/warehouse-manager/

SELECT name AS WAREHOUSE\_NAME, SUM(calc) AS VOLUME

"""

rename warehouse name as name

"""

FROM

(

SELECT a.name, a.product\_id, units \* widthheightlength AS calc

FROM Warehouse a

INNER JOIN products b

ON a.product\_id = b.product\_id

)a

GROUP BY name

"""

need to group by name

"""

"""

A subquery is a query that is nested inside a SELECT, INSERT, UPDATE, or DELETE statement, or inside another subquery.

"""

From

(

select product, id,width\*length\*height as vol from products

)a

select w.name as warehouse\_name, SUM(p.width\*p.length\*p.height) as vol

from Warehouse w Products p

where p.product\_id=w.product\_id

group by warehouse\_name

select w.name as warehouse\_name, SUM(p.width\*p.length\*p.height) as vol

from Warehouse w, Products p

where p.product\_id=w.product\_id

group by warehouse\_name

SELECT w.name AS Warehouse\_Name, SUM(w.units\* p.width \* p.length \* p.height) AS volume

FROM warehouse w, products p

WHERE p.product\_id = w.product\_id

GROUP BY w.name

"""

https://leetcode.com/problems/replace-employee-id-with-the-unique-identifier/

left join

"""

# Write your MySQL query statement below

select em.unique\_id, employees.name

from EmployeeUNI em

right join employees on employees.id=em.id

"""

https://leetcode.com/problems/find-the-team-size/

Key: you can create

"""

SELECT d1.employee\_id, d2.team\_size

FROM Employee d1 LEFT JOIN

(SELECT team\_id, COUNT(team\_id) AS team\_size

FROM Employee

GROUP BY team\_id) d2

ON d1.team\_id = d2.team\_id

"""

https://leetcode.com/problems/unpopular-books/

"""

select t2.book\_id, t2.quantity, t2.dispatch\_date

from orders t2 left join

(select t1.book\_id, t1.name

from books)t3

group by book\_id

on t2.book\_id=t3.book\_id

WHERE t2.dispatch\_date netween (select max(tprice) From cte)

"""

SQL regex like vocabulary

"""

Select patient\_id, patient\_name, conditions from

Patients where regexp\_like(conditions,' \*DIAB1')

"""

SQL delete

"""

delete from customers where customername="somename"

"""

SQL select top n

"""

SELECT TOP 3 \* FROM Customers

WHERE Country='Germany';

"""

https://leetcode.com/problems/friend-requests-ii-who-has-the-most-friends/

"""

from request\_accepted

select requester\_id, accepter\_id

group by requester\_id

# Write your MySQL query statement below

select t1.requester\_id id, SUM(t1.id1+t2.id2)

from(select requester\_id, COUNT(requester\_id) id1

from request\_accepted

group by requester\_id)t1

,(select accepter\_id, COUNT(accepter\_id) id2

from accepter\_id

group by accepter\_id)t2

on t1.requester\_id=t2.accepter\_id

select t1.requester\_id id, SUM(t1.id1+t2.id2)

from(select requester\_id, COUNT(requester\_id) id1

from request\_accepted

group by requester\_id)t1

,(select accepter\_id, COUNT(accepter\_id) id2

from accepter\_id

group by accepter\_id)t2

on t1.id1=t2.id2

select t1.requester\_id id, t1.id1+t2.id2

from(select requester\_id, COUNT(requester\_id) id1 from request\_accepted group by requester\_id)t1

,(select accepter\_id, COUNT(accepter\_id) id2

from accepter\_id

group by accepter\_id)t2

on t1.id1=t2.id2

select friend,max(cnt)

from (

select friend,count(friend) cnt

From (

select requester\_id as friend

from request\_accepted

union all

select accepter\_id as friend

from request\_accepted)t

group by friend

)

select friend,cnt

from (

select friend,count(friend) cnt

From (

select requester\_id as friend

from request\_accepted

union all

select accepter\_id as friend

from request\_accepted)t

group by friend

)t1

order by cnt desc

select friend id,cnt num

from (

select friend,count(friend) cnt

From (

select requester\_id as friend

from request\_accepted

union all

select accepter\_id as friend

from request\_accepted)t

group by friend

)t1

order by cnt desc

limit 1

"""

https://leetcode.com/problems/capital-gainloss/

1393. Capital Gain/Loss

"""

select stock\_name, capital\_gain\_loss

SELECT s.stock\_name, s.operation, if(s.operation="Buy",-1\*s.price,s.price)

from stocks s

"""

https://leetcode.com/problems/second-highest-salary/

"""

"""

Write your MySQL query statement below

rank descending

Choose secondmax

"""

select max(salary) as secondhighestsalary

from Employee

where salary not in (select max(salary) from employee)

"""

https://leetcode.com/problems/apples-oranges/submissions/

"""

# Write your MySQL query statement below

select sale\_date, sum(sold) as diff

from(

select sale\_date, fruit, if(s.fruit="oranges",-1\*s.sold\_num,s.sold\_num) as sold

from sales s) t

group by sale\_date

"""

https://leetcode.com/problems/department-highest-salary/

"""

# Write your MySQL query statement below

select name department, name employee, max(salary)

from(

select t1.salary, t1.departmentid, t1.Name

from employee t1

left join department t2 on t1.departmentid=t2.id

order by salary desc) t3

group by department

select employee.salary, employee.departmentid , employee.name as n

from employee

left join department t2 on employee.departmentid=t2.id

order by salary desc

select name department, n employee, max(salary)

from(

select employee.salary, employee.departmentid , employee.name as n, t2.name

from employee

left join department t2 on employee.departmentid=t2.id

order by salary desc) t3

group by department

# Write your MySQL query statement below

select name department, n employee, max(salary)

from(

select employee.salary, employee.departmentid , employee.name as n, t2.name

from employee

left join department t2 on employee.departmentid=t2.id

order by salary desc) t3

group by department

select name department, n employee, max(salary)

from(

select employee.salary, employee.departmentid , employee.name as n, t2.name

from employee

left join department t2 on employee.departmentid=t2.id

order by salary desc) t3

on max(salary) is salary

group by department

select name department, n employee, max(salary)

from(

select employee.salary, employee.departmentid , employee.name as n, t2.name

from employee

left join department t2 on employee.departmentid=t2.id

order by salary desc) t3

group by department

"""

Note: max is simply used to return the max of a sequence, not references

"""

select departmentid, name, salary

from employee join department

on department.id=employee.departmentid

where (department.id,salary) in (

select departmentid , max(salary) from employee group by departmentid)

"""

rectified

"""

select department.name department, employee.name employee, salary

from employee join department

on department.id=employee.departmentid

where (department.id,salary) in (

select departmentid , max(salary) from employee group by departmentid)

# Write your MySQL query statement below

select en.student\_id, en.course\_id, en.grade from enrollments en

right join (

select student\_id , max(grade) grade from enrollments group by student\_id order by course\_id asc) s

on en.student\_id=s.student\_id and en.grade=s.grade

group by student\_id

order by course\_id asc, student\_id asc

# Write your MySQL query statement below

WITH temporaryTable (averageValue) as

(select en.student\_id, en.course\_id, en.grade from enrollments en

order by course\_id asc)

select en.student\_id, en.course\_id, en.grade

from Table

right join (

select student\_id , max(grade) grade from enrollments group by student\_id ) s

on en.student\_id=s.student\_id and en.grade=s.grade

group by student\_id

select table1.student\_id, min(table1.course\_id) course\_id, table1.grade

from (select en.student\_id, en.course\_id, en.grade from enrollments en

order by course\_id asc) table1

right join (

select en.student\_id , max(en.grade) grade from enrollments en group by student\_id ) s

on table1.student\_id=s.student\_id and table1.grade=s.grade

group by table1.student\_id

order by table1. student\_id asc, table1. student\_id asc

"""

https://leetcode.com/problems/nth-highest-salary/

"""

CREATE FUNCTION getNthHighestSalary(N IN NUMBER) RETURN NUMBER IS

result NUMBER;

BEGIN

/\* Write your PL/SQL query statement below \*/

select distinct salary INTO result from(

select salary , dense\_rank() over(order by salary desc ) AS RNK

from employee)

where rnk=N;

RETURN result;

END;

"""

1) How manyusers turned the feature on today?

USER\_ID || ACTION||DATE||TIME

"""

select count(distinct user\_id) from t1

where curdate()=date and action="on"

"""

How manyusers have ever turned the feature on?

"""

"""

WUT

"""

SELECT A.DATE, B.USER\_ID, B.STATUS

(SELECT GENERATE\_SERIES('2018-01-01'::DATE,'2018-09-01'::DATE, '1D')::DATE) TABLEA(DATE)

LEFT JOIN

(SELECT \* FROM

TABLE

QUALIFY ROW\_NUMBER() OVER (PARTITION BY USER\_ID, DATEORDER BY TIME DESC) =1

) B

ON TABLEA.DATE >= B.DATE

QUALIFY ROW\_NUMBER() OVER(PARTITION BY A.DATE, B.USER\_IDORDER BY B.DATE DESC) =1

"""

Data:

table1

| date | business\_id | action | tool |

table2

| date | business\_id | country | industry |

Find top 3 tools most used by business in each industry.

"""

select industry, Count(tool) tools from(

select t1.business\_id, t1.date, t1.tool, t2.industry from table1 t1

left join table2 t2

on t1.business\_id=t2.business\_id and t1.date=t2.date) t3

group by industry

order by tool desc

limit 3

"""

https://leetcode.com/problems/friend-requests-i-overall-acceptance-rate/

"""

select count(friend\_request.sender\_id)/count(request\_accepted.requester\_id)

from friend\_request, request\_accepted

cast(your\_float\_column as decimal(10,2))

# Write your MySQL query statement below

select cast(count(distinct friend\_request.\*)/count(distinct request\_accepted.\*) as decimal(10,2) )accept\_rate

from friend\_request, request\_accepted

# Write your MySQL query statement below

select isnull(count((distinct request\_accepted.requester\_id,accepter\_id )/cast(count(distinct friend\_request.sender\_id, send\_to\_id ) as decimal(10,2)),0)accept\_rate

from friend\_request, request\_accepted

# Write your MySQL query statement below

select ifnull(count((distinct request\_accepted.requester\_id,accepter\_id )/cast(count(distinct friend\_request.sender\_id, send\_to\_id ) as decimal(10,2)),0)accept\_rate

from friend\_request, request\_accepted

# Write your MySQL query statement below

select ifnull(round(count(distinct request\_accepted.requester\_id,accepter\_id )/count( distinct friend\_request.sender\_id, send\_to\_id ),2),0) accept\_rate

from friend\_request, request\_accepted

"""IMPORTANT SQL TIME FUNCTION

"""

SELECT DATEADD(year, 1, '2017/08/25') AS DateAdd;

SELECT DATEDIFF(year, '2017/08/25', '2011/08/25') AS DateDiff;

SELECT DATEFROMPARTS(2018, 10, 31) AS DateFromParts;

SELECT DATENAME(year, '2017/08/25') AS DatePartString;

SELECT GETDATE();

"""

https://stackoverflow.com/questions/249301/simple-random-samples-from-a-sql-database

"""

# Write your MySQL query statement below

select s2.player\_name, s2.day, (if(s2.gender="Female",1\*s2.score\_points,0)+if(s2.gender="Male",1\*s2.score\_points,0), female\_total female\_total, gender

from scores s2 join (

select player\_name, day, if(gender="Male",1\*score\_points,0) male\_total, gender

from scores

) s1

on s1.player\_name=s2.player\_name and s1.day=s2.day

# Write your MySQL query statement below

select s2.gender, s2.day, sum(score\_points) over (partition by gender order by day) total

from scores s2

# Write your MySQL query statement below

# Write your MySQL query statement below

select customer\_id, product\_id, count(\*) as count\_trans

from orders

group by customer\_id, product\_id

order by count\_trans desc

select o.customer\_id, o.product\_id, count(\*) cnt, p.product\_name

from orders o right join products p

group by o.customer\_id, o.product\_id

order by cnt desc

# Write your MySQL query statement below

select p.product\_name, p.product\_id, c.customer\_id from products p right join (

select customer\_id, product\_id, count(\*) as count\_trans

from orders

group by customer\_id, product\_id

order by count\_trans desc)c

on p.product\_id=c.product\_id and p.customer\_id=c.customer\_id

select c.custo,p.product\_id,p.product\_name from products p right join (

select customer\_id custo, product\_id produ, count(\*) as count\_trans

from orders

group by customer\_id, product\_id

order by count\_trans desc)c

on product\_id=c.produ

select customer\_id custo, product\_id produ, count(\*) as count\_trans

from orders

group by customer\_id, product\_id

order by count\_trans desc

# Write your MySQL query statement below

with cte as (

select customer\_id,

product\_id,

count(order\_date) as t

from orders

group by customer\_id, product\_id),

cte2 as (select customer\_id,

product\_id,

dense\_rank() over (partition by customer\_id order by t desc) as rk

from cte)

select c.customer\_id as customer\_id,

c.product\_id as product\_id,

p.product\_name as product\_name

from cte2 as c

left join products as p

on c.product\_id = p.product\_id

where c.rk = 1

order by c.customer\_id, c.product\_id

;

# Write your MySQL query statement below

with cte as (

select customer\_id,

product\_id,

count(order\_date) as t

from orders

group by customer\_id, product\_id),

cte2 as (select customer\_id,

product\_id,

dense\_rank() over (partition by customer\_id order by t desc) as rk

from cte)

select c.customer\_id,p.product\_id,p.product\_name from cte2 c left join products p on p.product\_id=c.product\_id where c.rk=1

with s as (select user\_id from activity where activity\_date between "2019-07-27" and "2019-08-27")

select count(\*)/count(distinct user\_id) average\_sessions\_per\_user

from s

# Write your MySQL query statement below

with s as (select user\_id, count(distinct user\_id, session\_id)st from activity where activity\_date between "2019-06-27" and "2019-07-27" )

select count(\*)/st average\_sessions\_per\_user

from s

# Write your MySQL query statement below

with s as (select user\_id, count(distinct user\_id, session\_id)st from activity where activity\_date between "2019-06-27" and "2019-07-27" group by user\_id)

select count(\*)/st average\_sessions\_per\_user

from s

# Write your MySQL query statement below

with s as (select user\_id, count(distinct user\_id, session\_id)st from activity where activity\_date between "2019-06-28" and "2019-07-27" group by user\_id)

select ifnull(round(sum(st)/count(\*),2),0)average\_sessions\_per\_user

from s

""'

select b.name, b.book\_id from books b

right join(

select book\_id, sum(quantity) s

from orders

where dispatch\_date between "2018-06-23" and "2019-06-23"

group by book\_id

having s<10)t1

on b.book\_id=t1.book\_id

'''

# Write your MySQL query statement below

select b.name, b.book\_id from books b where b.available\_from < "2019-05-23" and b.book\_id not in

(select book\_id from orders

where dispatch\_date between "2018-06-23" and "2019-06-23"

group by book\_id

having sum(quantity)>=10)

#having b.available\_from < "2019-05-23"

select m.movie\_id, m.title from movies m right join(

select user\_id, count(rating) cnt

from Movie\_Rating

order by cnt desc

limit 1 )t on m.movie\_id=t.movie\_id

# Write your MySQL query statement below

with cte1 as(

select u.name from users u right join(

select user\_id, count(rating) cnt

from Movie\_Rating

order by cnt desc

limit 1 )t on u.user\_id=t.user\_id)

with cte2 as(

select mo.title from movies mo right join(

select movie\_id, sum(rating)/count(rating) su

from movie\_rating

where created\_at>= '2020-02-01' and created\_at <= '2020-02-29'

group by movie\_id

order by su desc, movie\_id

limit 1)f on mo.movie\_id=f.movie\_id)

select name from cte1

union all

select title from ct2

(select name results from(

select u.name from users u right join(

select user\_id, count(rating) cnt

from Movie\_Rating

order by cnt desc

limit 1 )t on u.user\_id=t.user\_id order by mo.title asc)ct1)

union

(select title from(

select mo.title from movies mo right join(

select movie\_id, sum(rating)/count(rating) su

from movie\_rating

where created\_at>= '2020-02-01' and created\_at <= '2020-02-29'

group by movie\_id

order by su desc

limit 1)f on mo.movie\_id=f.movie\_id order by mo.title asc)ct3)

# Write your MySQL query statement below

select name from(

select u.name, count(rating) cnt

from Movie\_Rating t

left join (select u.name, u.user\_id from users u order by u.name asc)u

on u.user\_id=t.user\_id

order by cnt desc

limit 1)l

union

select title from(

select m.title, avg(rating) average

from Movie\_Rating t

left join (select m.movie\_id, m.title from movies m order by m.title asc)m

on t.movie\_id=m.movie\_id

order by average desc

limit 1)x

(

SELECT a.name as results

FROM Users a LEFT JOIN Movie\_Rating b

ON a.user\_id = b.user\_id

GROUP BY a.name

ORDER BY COUNT(a.user\_id) DESC, a.name ASC

LIMIT 1

)

UNION

(

SELECT a.title as results

FROM Movies a LEFT JOIN Movie\_Rating b

ON a.movie\_id = b.movie\_id

WHERE EXTRACT(YEAR\_MONTH FROM created\_at) = '202002'

GROUP BY a.movie\_id

ORDER BY AVG(b.rating) DESC, a.title ASC

LIMIT 1

)

# Write your MySQL query statement below

select name as results from(

select u.name, count(rating) cnt

from Movie\_Rating t

left join (select u.name, u.user\_id from users u)u

on u.user\_id=t.user\_id

group by u.name

order by cnt desc, u.name asc

limit 1)l

union

select title from(

select m.title, avg(rating) average

from Movie\_Rating t

left join (select m.movie\_id, m.title from movies m )m

on t.movie\_id=m.movie\_id

where created\_at>= '2020-02-01' and created\_at <= '2020-02-29'

group by m.title

order by average desc, m.title asc

limit 1)x

# more intuitive solution

SELECT DISTINCT a.id

, (SELECT name FROM accounts WHERE id=a.id) AS name

FROM logins a, logins b

WHERE a.id = b.id AND DATEDIFF(a.login\_date, b.login\_date) BETWEEN 1 AND 4

GROUP BY a.id, a.login\_date

HAVING COUNT(DISTINCT b.login\_date) = 4

# Write your MySQL query statement below

select distinct b.id, a.name

FROM Accounts a

right JOIN

(select id, login\_date,

dense\_rank() over(order by login\_date) rn1,

dense\_rank() over(partition by id order by login\_date) rn2

FROM Logins) b

ON a.id = b.id

GROUP BY b.id,abs(rn1-rn2)

HAVING COUNT(distinct rn2) >= 5

ORDER BY b.id

select distinct recommended\_page from(

select page\_id as recommended\_page from likes

where user\_id in (( select user1\_id from friendship

where user2\_id=1)

union

(select user2\_id from friendship

where user1\_id=1)

)

)ct2

—————————————————————————————

select distinct page\_id as recommended\_page from likes

where user\_id in (( select user1\_id from friendship

where user2\_id=1)

union

(select user2\_id from friendship

where user1\_id=1)

)

and page\_id not in (select page\_id from likes where user\_id=1)

Table Name: Product

id (int) [Primary Key]

sku (varchar 32)

product\_name (varchar 128)

product\_description (text)

current\_price (decimal(8,2))

quantity\_in\_stock (int)

Table Name: Invoice

id (int) [Primary Key]

invoice\_number (varchar(255))

customer\_id (int)

user\_account\_id (int)

total\_price (decimal(8,2))

time\_issued (varchar)

time\_due (varchar)

time\_paid (varchar)

time\_canceled (varchar)

time\_refunded (varchar)

Table Name: invoice\_item

id (int) [Primary Key]

invoice\_id (int)

product\_id (int)

quantity (int)

price(decimal(8,2))

line\_total\_price(decimal(8,2))

WITH

user\_streaks AS

(

SELECT user\_id, level as lvl, rank() over (order by level desc) rnk

FROM table

WHERE connect\_by\_isleaf = 1

START WITH date = trunc(sysdate) - 1

CONNECT BY PRIOR date = date + 1 AND PRIOR user\_id = user\_id

)

SELECT user\_id, lvl, rnk

FROM user\_streaks

WHERE rnk &lt;= 100

select count(\*) as volume, timespent\_sec as time\_bucket

from session\_details

group by timespent\_sec

order by 2