Wayne Zhu
Project Part 3 Notes

Techniques to consider for the physical design:
1. Indexing: MySQL already automatically index by primary key using b-tree. This will accelerate the query process speed. See here: https://dev.mysql.com/doc/refman/8.0/en/mysql-indexes.html

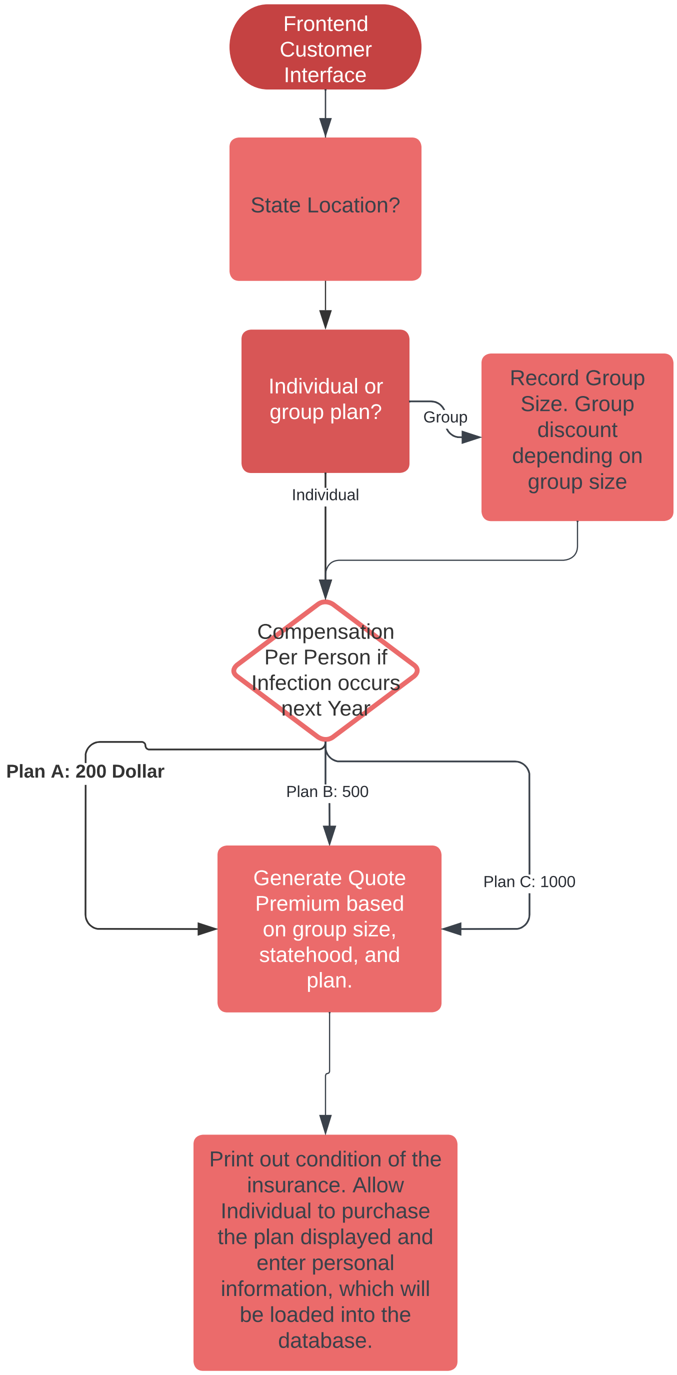
2. Partitioning: I assume here it's not about partition join, but it's actually about splitting table into multiple disk block—aka horizontal partition. I don't see a particular reason to use partition here. I think it will slow down the program not speeding it up. I think we can easily store all the data in the tables without partition because we are using indexing already. Read more about partition in MySQL: https://dev.mysql.com/doc/refman/8.0/en/partitioning-overview.html

3. Clustering Index: I don't see the necessity of having a clustering index in order to accelerate access of the other field. In the future, during the deployment of the product, we can identify the most frequently used search index like Group State that are not key in GroupAccount and make them clustering index. It should be ad hoc.

4. Selective Materialization: If this optimization is turned on, MySQL will store previous subquerys as tables in cache to speed up processing. I checked that the materialization flag=one in my program. https://dev.mysql.com/doc/refman/8.0/en/subquery-materialization.html

5. I didn't use Enterprise Cloud for this project purely because of time constraint but I do see its potential with large unstructured files. I want to introduce some unfamiliar concept but not too many which will overwhelm myself.

## Business Use Case Diagram for Accident Insurance Company for Covid-19 Infection


Frontend Customer Interface
State Location?
Individual or group plan?
Group
Record Group Size. Group discount depending on group size
Individual
Compensation Per Person if Infection occurs next Year
Plan A: 200 Dollar
Plan B: 500
Plan C: 1000
Generate Quote Premium based on group size, statehood, and plan.
Print out condition of the insurance. Allow Individual to purchase the plan displayed and enter personal information, which will be loaded into the database.

This type of accident or injury insurance is quite interesting because it's so real world. Even WSJ talked about it here: https://www.wsj.com/articles/insurers-in-taiwan-pay-the-price-for-misjudging-covid-risks-11655546402

Basically, Taiwan's insurer miscalculated the risk of infection in 2021 because of low case load. Now, Taiwan has so many covid infections and they are losing money—the first in decades. This shed light on the illness, which is hard to predict because of variant.

Our dataset is JHU's cumulative daily US covid cases. We want to use this dataset to simply differentiate different state's customers' risk at covid infection. For example, a New York person will have a much higher chance of getting covid than Alabama person due to regional differences in cases by 100,000 people. I will use a regression machine learning model to get the following equation in R:

IndividualPremium=(StateCovidProbability*Compensation*Multiplier)(1-GroupDiscount)

IndividualPremium: The y in our regression. Price for each individual purchaser. If it's a group plan, then it's the average premium for each individual.

StateCovidProbability: The probability for each individual in each state to get covid next year based on the cumulative annual case probability per 100,000 people in each state. Each state has a different value. We are doing a dummy variable regression including all states as dummy. The chance of being infected by covid in each state is stored in the StateCovidInfo table. E.g. This value is in fact this expression: (Alaska*AlaskaInfectionChance+Alabama*AlabamInfectionChance+…), where Alaska, Alabama are all boolean (dummy) variables.

Compensation: The x in our regression. Payout if the person is infected. Depending on the plan, the payout could be 200, 500, or 1000. Payout that is too high is not encouraged because people would then intentionally get it, and it would be hard to monitor.

Multiplier: Let it be 1.1. We charge 110 percent the expected value to generate profit and cover administration. This is based on the industry standard of 2-3% profit in the insurance industry.

GroupDiscount: Larger group have more bargaining power and therefore is subject to more discount. Right now, the policy is designed to be: 10+, 1% discount. 100+, 3%. 1000+, 5%. 10000+ 7%. Of course, more business research could be done as to what number is the most profitable.

In the long run, we shall also incorporate other countries' covid data in case of new variant and also take into account new treatment, mask mandate, unreported at home test cases and etc. Or we can even scrap tweets to see if people talk about covid more in some area. So many possibilities, but right now we don't want to overcomplicate our initial model. This is a purely business insurance, not a health insurance so calculating risk doesn't mean we are compromising people's wellbeing.

I produce analytics in R, generate data, and populate the StateCovidInfo and ProductPlan tables.

I use Mockaroo to generate mock data for IndividualOrGroupAccount, and then I use SQL query to update the IndividualOrGroupAccount.ActualPremiumPerPerson data based on ProductPlan table.

You should be able to see my database as an sql file. I didn't generate mock data for payment, claim and individual data yet because they are irrelevant to the ML.

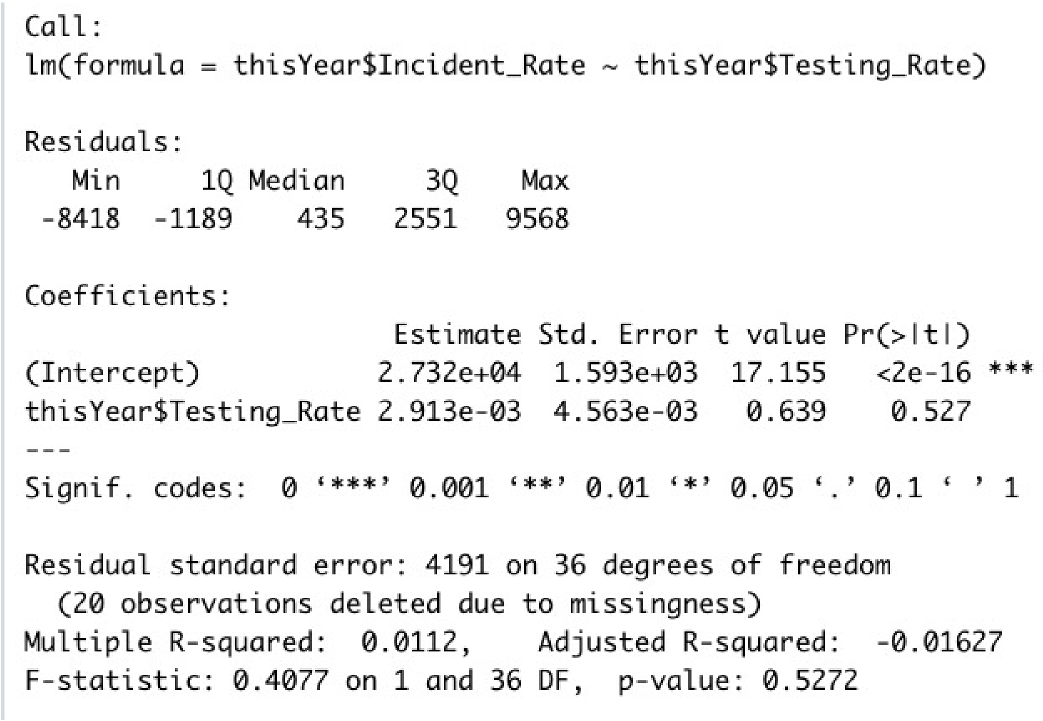
Because of time constraint, I use the simplest ML model: Regression with state dummy variables.

Some ideas for more ambitious project: Instead of looking at state level, it can take user zip code and look at county level covid cases, which is available.

It could also look at testing rate per 100,000 people. It could be that people in states with higher test rate care more about their own and other's health, so the covid cases is closer to the actual value. However, lower test rate could also means that people in those state are taking at-home test instead. Our original model likely vastly underestimate the actual case count due to unreported at home test positive. Our data shows that around 20 percent of people in each state has been infected since last year, which is dramatically lower than the CDC estimate that 30% of the entire population got infected with omicron. Some people even suggests that the actual number is 5-10 times of the reported number. https://www.usnews.com/news/health-news/articles/2022-05-20/latest-covid-19-surge-in-u-s-is-drastically-undercounted

So we can multiple the probability by 3 times for a ballpark number. There is no point of taking different test rate across state into account when at-home test is contributing more to the under-reporting of cases.

Also, I've ran a regression of state test rate against state covid case rate, and there is no significance.

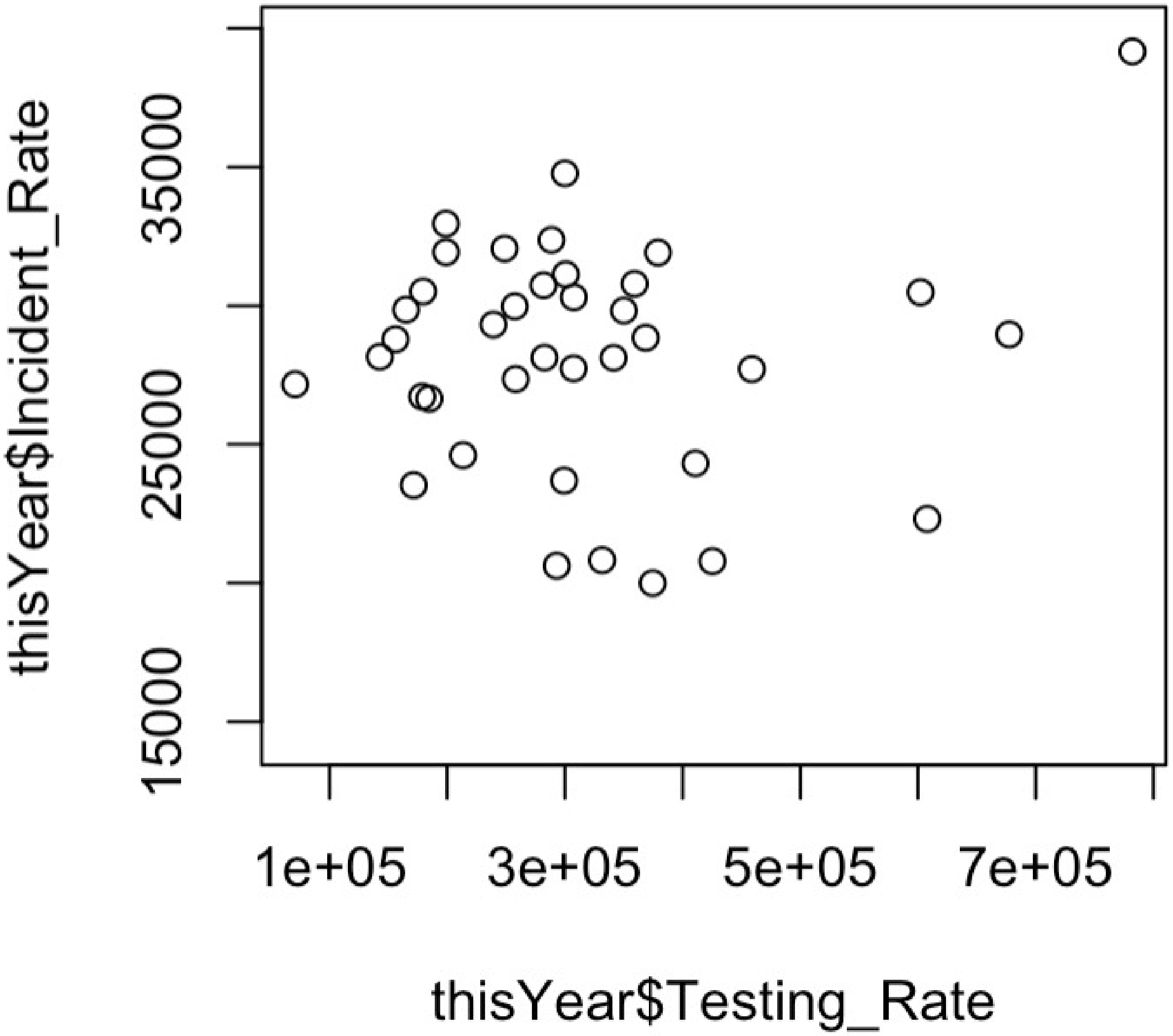


```
Call:
lm(formula = thisYear$Incident_Rate ~ thisYear$Testing_Rate)

Residuals:
   Min     1Q Median     3Q    Max 
 -8418  -1189    435   2551   9568 

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)    
(Intercept)          2.732e+04  1.593e+03  17.155   <2e-16 ***
thisYear$Testing_Rate 2.913e-03  4.563e-03   0.639    0.527    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4191 on 36 degrees of freedom
  (20 observations deleted due to missingness)
Multiple R-squared:  0.0112,	Adjusted R-squared:  -0.01627 
F-statistic: 0.4077 on 1 and 36 DF,  p-value: 0.5272
```


thisYear$Incident_Rate
15000
25000
35000
1e+05
3e+05
5e+05
7e+05
thisYear$Testing_Rate

We, however, do have both datasets in the JHU covid case files.

There are also variant case percentage data available on a national scale. So a new more potent variant will almost certainly lead to a new wave.

Another real world macroscopic change that is slightly harder to track down is the coming of variants. New variants like Beta, Delta, and Omicron, and Omicron's subvariants created huge surges in cases, and it's also the reason why crowd immunity have improved afterward. For example, following the huge increase of cases during Omicron wave this year, new variant—albeit more infectious—led to fewer cases now. Similarly, new vaccines can have the same type of groundbreaking effect—like the period of peace after everyone is vaccinated against delta in the US. Both vaccine and new variant are hard to predict.

However, the likelihood of a new more infectious variant is absolutely correlated with the number of world cases of covid-19. The more people are infected, the more likely the variant is going to mutate. This is the message that the WHO has repeatedly reminded us of. So we can also take into account of the number of global cases right now. I've not taken a time-series analysis class before, so I cannot offer a detailed analytics on the expected value of a new variant and how infectious that would be. There is that tug of war of crowd immunity vs. new variant, which makes it hard to come by a precise number. However, I do remain optimistic that cases are lower now, and the chance of another variant dramatically different from Omicron that is more infectious is low. So I think the insurance company can safely expect fewer cases for next year's plan—I just don't know the precise number with enough confidence to incorporate it into my model.

Reference Architecture

Principles:
IT solutions will align with business strategy.
People first, technologies second (as in Agile).
All IT assets will have identified business and IT owners (ownership is important for motivation)

Framework:

Responsibility of the 6 domains as listed.

Business: Business research into pricing, competition and payout.

People: Offer the best customer service and resolve conflicts between teams.

Application: Incorporating customer feedback into the UI and Java program.

Information: Manage the backend database and debug systemic issues.

Technical: Rerun the Machine Learning analytics regularly based on business research.

Process: Integrate the systems and test run and push to production. Brainstorming ways to bring DevOps into the other domains.

Method:

Plan: We follow Agile methodology, daily standup meeting to corroborate information. We have weekly meeting of all the domain also.

Deliver: We follow Agile methodology and introduce sprints, small patches, revision, and adjustment based on customer's needs.

Operate: We follow DevOps principles to integrate development and production.

Governance:
The RA will be made clear to everyone on the team.
Violaters will be trialed in a kangaroo court. If found guilty, the person will be spanked.