First in order to allow to SQL database to handle hashtags, we need to modify the 'statuses' table in SQL. In the 'statuses' table, first we create a column or a field and name it as hashtag. In this column, we capture the hashtags’ value from each single post from the table. Since the hashtags’ value appear in the text column in the 'statuses' table, we store the string value starting behind the symbol “#” and ending before space since the format of hashtag is an unbroken word or phrase. If there is no hashtag from the post, we will have a null value in the hashtag field. If there are more than one hashtag value capture on a single, we can store them in the same cell of the hashtag field separated by commas.

The modification that we have created would be benefic for searching the hashtags and finding the relationship between hashtags and their posts. It is also an efficient way to search since we do not need to create any additional tables that would causing more space for the database to store things. Regarding to the hashtag searching, we can conduct queries include the ‘like’ condition to search the desire the hashtag value that we are looking for. After executing the query, we will have all the posts’ information that contain the hashtag. And further by inner joining the ‘users’ table, we can also find the users’ information regarding the original post that has the hashtag.

If twitter adds the ability of customizing tweet’s font and color, I would modify the ‘statuses’ table in SQL in order to help supporting this function. In the ‘status’ table, we can create a column or a field and name it as ‘font\_style’. In this field, it captures the font style of each single post from the ‘status’ table. And likewise, for the colors we can create another column or field with name ‘color’. And it would be interesting and more meaningful for the perspective of analytics, if twitter can set up rules like different colors represent different mood or status of the users when posting the tweets. So, the ‘color’ field stores the color of each tweets in the ‘statuses’ table.

For Mongo database, I would modify the ‘twitter\_statuses’ table, ‘twitter\_retweets’ table and ‘twitter\_favorites’ table since all of these three tables contain the information of the posts. For the ‘twitter\_statuses’ table, I would add two fields with the name of ‘font\_style’ and ‘color’ underneath the field ‘text’. For ‘twitter\_retweets’ table, I would add these new fields in the group of ‘retweeted\_status’ since the color and font style belong to the original post’s information.

I would say the implement on SQL will be a little bit easier than implementing on Mongo database. Since there is only one table in SQL that contains the specific information about each tweets/post, it will be easier to set up and to maintain by only monitoring this table. For Mongo on the other hands, it will require to set up and maintain on more than one table which will cause more time consuming and more human efforts.

Regarding to task two, I believe SQL database is easier to work with. Since the prompt is asking us to find all the distinct users from the database and by discovering the SQL database, we know that SQL database has a table called ‘users’ that contains all the users’ information. So, by conducting a single line query on SQL to find all the distinct/unique value in the table by the primary key ‘user\_id’, we then can get the final result as a list of users in the database with no duplicates. However, the Mongo database does not have a table that has all the user’s information. It will require to add up all the users from all the five different databases and to find the unique value from it.

Regarding to task one, I believe Mongo database is easier to work with. Since the prompt on this specific question is asking us to find the list of users who have retweeted a post that was originally from Elon Musk. After exploring the database on Mongo, we discover that there is a ‘twitter\_retweets’ table that has the retweets information. And from this table, there is a group called ‘retweeted\_status’ that contains all the information about the original tweet where the user retweeted from. So, by finding the user’s screen name equal to ‘elonmusk’ from the ‘quoted\_status’ under the group of ‘retweeted\_status’, we will be able all the list of users who have retweeted Elon Musk.

The way to construct the database between SQL and Mongo varies a lot. Mongo has total of five different tables which has a lot more complex structure than the SQL database does. So when encountering problems like adding new fields to the database, Mongo will be relatively difficult than the SQL database.

Here we have the web page with advanced functionality

So, the whole webpage was created by Python from scratch.

As mention before, the webpage is also in the testing phase right now.

The reason we are creating this website is that We are hoping to find out some of the possible advance functions that can be conducted by python, and on the later stages, we are planning to integrate these advanced functions into the web app that was introduced by Yan on the previous slides.

As we can see the screen shot on the slides, we have Implemented the multiple filtering function on the navigation bar, which is located on the left side of the webpage.

For researchers who are interested in searching more than 1 inputs at a time for a specific category, they can simply do it on the web page. As you see from the example, you can input both the art and culture on the subject field and just by clicking the search button at the bottom, you will get all the collections that contains both art and culture on the subject field at once.

And like wise, for other categories fields, you can also input more than one inputs and it will get it to work.

**Next**

On this slide, we have some of the sample python script that we have used for the project. On the left, we have created a python function that can check for the validity of the links for the collections.

By just simply by running the code, we can detect which links are not working. The reason we are checking the links periodically is because, for some institutions and schools, they may update or change their websites address for some reasons, and it will very important and useful for us to always make sure that our data is up to date.

On the right-hand side of the screen, we have a sample portion of the python code for creating this webpage.

**On Next Side, Yan will introduce our metadata**

Here, we have some of our potential future concerns and plans for the project.

We are planning to expand our geographic scope from US and Canada to the rest of the world.

We also plan to continue cleaning and clarifying the data, as well as to seek some enhancements on the searching and browsing functions.

We also like to continue collecting recommendations for the digital collections on Chinese studies and to gather feedback from users.

We have also created the online survey form for it.

As you can see on the right-hand side, there is a QR code for the survey form.

You can simply use the camera on your cellphone to scan the code, and it will bring you directly to the link.

Or if you find it easier by using the link underneath it, we are certainly welcome to do so.

**Next**

First of all, I have computed the R-squared figure which is 95.53%. It is a goodness-of-fit measure for this linear regression model and 95.53% indicates that the model explains 95.53% of the variability of the response data around its mean.

I also computed the F-value which is 96.14. It is also a pretty good measure since it is close to 1. And also, since I got the p-value is very small and almost equal to zero, I am rejecting the null hypothesis in favor of the alternative hypothesis that at least one coefficient is not equal to zero. Although the constant variable has a p-value of 11.6%, the other two variables of “Days Absent” and “Year with company” have a much lower p-value of both at about 1%. I then can reject the null hypothesis that those two coefficients are zero.

I then also got the result of the Durbin-Watson Statistics and which it is 1.46. It is something that I need to care about since it is not in between 1.5 to 2.5.

The four in one plot that I created talks about the normality of the residuals and the constant variables of the residuals. The normal probability plot shows that a normal distribution with a mean of zero and standard deviation of 0.9197 fits the residuals well. But visually there might be a problem because the points are not following very closely at the straight line. The histogram shows the distribution of the residuals. The scatter plot indicates that the residual spots are pretty out-spread instead of having a constant bandwidth. The versus order plot visually shows that the plots are not very chaotic. The more chaotic of the graph, the more likely that the residuals aren’t infected individually distributed. Since the VIF of the two variables are 4.82 which is kind of high and over 1, there are correlation between these two and might to remove one in the later stage.

The probability of residual indicates that most of points follow the straight line; and since they are all within the two bounds which means there are no outliners detected.

For the DFIT threshold of 1.0, I do not find any data point that are between the positive and negative of the threshold. For the TRES threshold, I found two points which are at index 8 and 10. The TRES value are 1.848 and 1.475 which both exceed the threshold value of 1.397 at the significance level of 20%.

As for the results conducted on the Cereal sheet, we can say that there is not enough evidence to conclude there is a difference between the average of treatment 1, which is the Wheat and treatment 3, which is the Maize. The reason behind this is because we can see from the graph that the confidence interval of treatment 1 contains zero, and there is overlapping area between treatment 1 and treatment 3. However, there is enough evidence to say that the average of treatment 3 is different than the average of treatment 2 and treatment 4. As we can visually see from the graph that the confident intervals of treatment 2 and treatment 4 do not contain zero, and there are no overlapping areas. The next step will be to conduct a normality plot of the residuals and before that, we can conclude that treatment 2 and treatment 4 will have more thiamin content on average.

As we visually see from the residual plot, all points fall within the confidence boundaries hence there is no outliers detected. In addition, the p-value is bigger than 0.25 so we fail to reject the hypothesis test that the residuals are normally distributed at the mean of 0 and standard deviation of 0.8848.

We then used the Bartlett’s test to make conclusion on the equality of treatment means and to test the equality of the variances. As we are assuming the assumption that we previously stated is correct, we used the results coming from the Bartlett’s test to confirm that with the p-value of 81%, we fail to reject the null hypothesis that the variances with residuals term is constant. We finally conclude that the treatment 2 Barley and treatment 4 Oats will contain more Thiamin.

So, for some of the benefits and advantage that we have designed the database

First, we have minimized the size of the main table from 21 columns into 7 columns which will only contain the restaurant level information.

And as we introduced earlier in the previous slides, the rest of the columns are divided into tables with different features, so we can just simply link between tables to get the desired information. No unnecessary information will be shown in the result.

Also, it will be relatively easy to maintain the database. Since the dataset is divided into tables, if there is an issue coming from one table, we can just fix that table instead of fixing the entire data set. And it will also be relatively easy to update information, like adding more content to a table or delete values from particular table.

Third, It is also convenient for end users to search for information just by know the characteristics of the tables.

And lastly, if in the later stage, there is a need to add more tables to the database, it will also be convenient to do so, just by simply following the method we used to built for the previous tables.

**So Here on these slides, we have some example queries on how to use the database to find out information.**

On this one, we are trying to find out which countries in the Zomato database have the most restaurants.