Design Decisions

In phase 1, we got one feedback which is **adding more follow-up questions**. The changes we made are adding two follow-up questions for each question. In addition, we also slightly change each question a little bit.

Below are the questions we change to:

1. Which of the following attributes is associated with the outcome of the previous marketing campaign(poutcome)?
    - age
    - job
    - marital
    - education
    - existence credit
    - Housing
    - Loan
    - pcontact(number of times contacted of the previous campaign)
    - Employment variation rate
    - Consumer price index
    - Consumer confidence index
    - Euribor 3 month rate
    - Existence of subscription of a term deposit

    Follow up question:
    - if there is an association, does the average poutcome increase or decrease when the attribute increase?
    - For each attribute above, what value(group) results in the lowest or highest poutcome?
2. Which kind of people has the lowest or highest average poutcome?

    Follow up question:
    - What value of personal information(age, job, marital, education, existence credit, Housing, Loan) does the group of people have?
    - What value of social and economic attributes does the group of people have?
3. Does existence credit vary by different marital, education and job groups?

    Follow up question:
    - if there is an association, does the existence credit increase or decrease when the attribute increase?
    - For each attribute above, what value(group) results in the lowest or highest existence credit?

In addition to the change we made due to TA's feedback, we also **change Integrity Constraint** to:

- Person[ID] $\subseteq$ MaritalCredit[ID]
- Person[ID] $\subseteq$ Outcome[ID]

- Index[ID] ⊆ Outcome[ID]

We added the last two constraints because we found out that questions 1 and 2 we want to investigate require these two constraints. We also removed some constraints because we found they are meaningless. For example, we deleted MaritalCredit[ID] ⊆ Index[ID] because we don't need these two constraints in any question.

The last change we made is that we added two more attributes in MaritalCredit table because we change the third question a little bit so we need more attributes in this table.

Data Cleaning

We used pandas for the cleaning process. The code for cleaning is in "main.py".
Firstly, importing the original CSV data file as a dataframe named "data".
Secondly, only keep columns age, job, marital, education, default, housing, loan, previous, poutcome, emp.var.rate, cons.price.idx, cons.conf.idx, Euribor3m, subscribed by deleting other columns because we are only interested in these columns.
Then, change "unknown" to NULL in columns marital, default, housing, loan, and change "nonexistent" to NULL in column poutcome. The reason is that "unknown" and "nonexistent" both mean that we do not have the value for them.
After that, assign each row an index starting from 0 and name the column of indices as "ID".
Thirdly, we split the dataframe "data" into small dataframes according to the schema we set up in phase 1.

- create a dataframe called "outcome" that only contains the column poutcome of data, and set the name to "ID" for the column of indexing. Then, drop any rows with the missing value of poutcome(NULL).
- create a dataframe called "selected_poutcome" by using "merge" to inner join "outcome" with "data" to keep the rows that do not have a missing value for poutcome with all the columns of both dataframes.
- create a dataframe called "person" that contains the columns 'age', 'job', 'marital', 'education', 'default', 'housing', 'loan', 'previous' of selected_poutcome, and set the name to "ID" for the column of indexing with matching indices from selected_poutcome. Drop any rows with all the columns missing if necessary.
- create a dataframe called "index" that contains the columns 'emp.var.rate', 'cons.price.idx', 'cons.conf.idx', 'euribor3m', 'subscribed' of selected_poutcome, and set the name to "ID" for the column of indexing with matching indices from selected_poutcome. Drop any rows with all the columns missing if necessary.
- create a dataframe called "MaritalCredit" that only contains the column 'marital', 'education', 'job', 'default' of data, and set the name to "ID" for the column of indexing. Drop any rows with all the columns missing if necessary.

Lastly, save each dataframe outcome, person, index, MaritalCredit into a CSV file with the same name,