Project 2 Report

Shizhen Li, Bowen Du

Introduction:

The data is collected from Kaggle.com, https://www.kaggle.com/cdc/behavioral-risk-factor-surveillance-system#2011.csv, it is published by CDC, it is a survey contains more than 400k people around the US, which contains their health situation including healthcare coverage, weight, overweight, drinking, smoking. They called it the BRFSS, "The objective of the BRFSS is to collect uniform, state-specific data on preventive practices and risk behaviors that are linked to chronic diseases injuries, and preventable infectious diseases in the adult population. Factors assessed by the BRDSS include tobaccos use, health care coverage, HIV/AIDs knowledge or prevention, physical activity and fruit and vegetable consumption. Data are collected for a random sample of adults through a telephone survey".

The database itself contains more than 500 attributes, and 500k row. But not all of them are useful information, for example, they record who is the interviewer, land - phone or cellphone, etc. What we want to do is to collect information from this database which can be analyzed directly. Therefore, we drop unnecessary attributes and leaves those key elements in the database. At the end, we selected 64 attributes which are the key to this whole project.

Use Case:

Please go to our website vandycs101.com. Our backend is running 24 hours on an EC2 instance, which can handle all the request from our website. The database server is also running 24 hours on a RDS-MYSQL instance Currently, our application shows data for CA, TN. For one, I live in TN. And CA has the largest participants around the US, which includes around 20k people. The landing page gives detailed information about how we designed our application. The interactive button at the top of our application is to let the user pick a STATE, CA or TN, to view detailed information about those two states. The graph section includes multiple graphs to provide better visualization based on all states' data. It is fairly easy to play around with the our application. Note: the application only has a landing page now, the rest of the application is being reviewed by the AWS and should be deployed soon. Please email me if you aren't able to access the website.

Dataset:

This is the most difficult part of our project; we have to search what the variables mean and then make a decision whether we should include it or not. After spending a huge amount of effort of choosing the right data for my use. Here are the attributes we will be using:

(General info): _STATE, AGE, MARITAL, EDUCA, EMPLOY, WEIGHT2, GENHLTH, PHYSHLTH, MENTHLTH, HLTHPLN1, MEDCOST, CHECKUP1, HIVTST6, DIABAGE2
(state, age, marital, education, employment, weigh, physical health, general health, mental health, insurance cover, skip doc because cost, last time doc checkup, tested for HIV) 13

(Chronic conditions): BPMEDS, TOLDHI2, ASTHMA3, CHCSCNCR, CHCOCNCR, HAVARTH3, ADDEPEV2, CHCKIDNY, CHCVISON, DIABETE3,
(high blood pressure, high blood cholesterol, asthma, skin cancer, other cancers, arthritis,
Depression disorder, kidney disease, vison impairment, diabetes, diabetes age) 11

(Tobacco Use): SMOKDAY2
(smoke every day, some days, or not at all) 1

(Fresh consumption): FRUITJU1, FRUIT1, FVBEANS, FVGREEN, FVORANG
(drink 100%pure fruit juice, how many times eat fruit, eat beans, dark vegetables, orange colored vegies) 5

(Exercise): EXERANY2, EXRACT01, EXEROFT1, QLACTLM2, USEEQUIP, LMTJOIN3
(participate in any physical activities, what type, how many times, are you limited, need equipment, limited because joint or arthritis) 6

(Alcohol Use): ALCDAY5, MAXDRNKS
(drink how many times, largest number of drinks at one time) 2

(Testing Variable): IMPAGE, _RFHLTH, _HCVU651, _RFHYPE5, _DRDXAR1, _SMOKER3, WTKG3, _RFBMI5, _EDUCAG, FTJUDA1_, FRUTDA1_, BEANDAY_, GRENDAY_, ORNGDAY_, VEGEDA1_, _TOTINDA, _FLSHOT5, DROCDY3_, _RFBING5, _DRNKDY4, _RFDRHV4, _RFDRMN4, _AIDTST3
(age, health, health coverage, high blood pressure, arthritis, four level of smokers, weight in kg, overweight, education, fruit juice, fruit intake, bean intake, vegies intake per day, doing exercise, aged 65+ have flu shot, drink occasions per day, binge drinker, total achol per day, heavy drinker, adult who are heavy drinker, HIV tested).

Relations:
For general information, a primary key person_id.
person_id → _STATE, AGE, MARITAL, EDUCA, EMPLOY, WEIGHT2, GENHLTH, PHYSHLTH, MENTHLTH, HLTHPLN1, MEDCOST, CHECKUP1, HIVTST6, DIABAGE2

For Chronic, a primary key chronic_id
chronic_id → BPMEDS, TOLDHI2, ASTHMA3, CHCSCNCR, CHCOCNCR, HAVARTH3, ADDEPEV2, CHCKIDNY, CHCVISON, DIABETE3

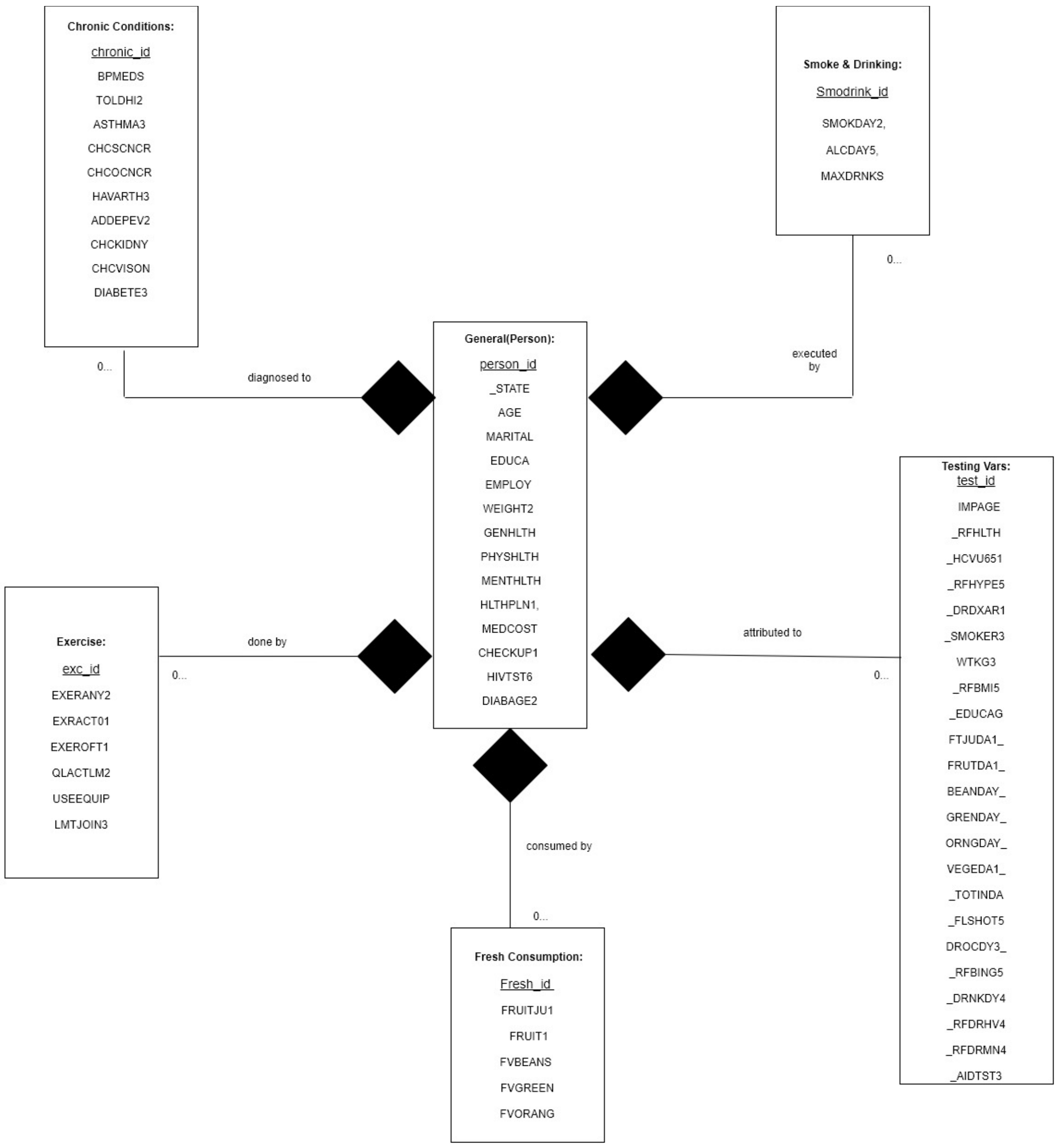
For smoking and drinking, a primary key smodrink_id,
Smodrink_id → SMOKDAY2, ALCDAY5, MAXDRNKS

For Fresh consumption, a primary key fresh_id,
Fresh_id → FRUITJU1, FRUIT1, FVBEANS, FVGREEN, FVORANG

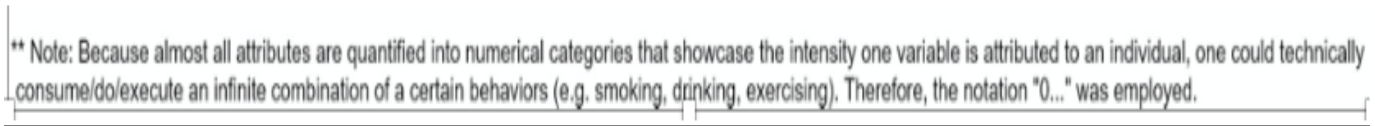
For Exercise, a primary key exc_id.
exc_id → EXERANY2, EXRACT01, EXEROFT1, QLACTLM2, USEEQUIP, LMTJOIN3

For testing, a primary key test_id.
test_id → IMPAGE, _RFHLTH, _HCVU651, _RFHYPE5, _DRDXAR1, _SMOKER3, WTKG3, _RFBMI5, _EDUCAG, FTJUDA1_, FRUTDA1_, BEANDAY_, GRENDAY_, ORNGDAY_, VEGEDA1_, _TOTINDA, _FLSHOT5, DROCDY3_, _RFBING5, _DRNKDY4, _RFDRHV4, _RFDRMN4, _AIDTST3

UML:

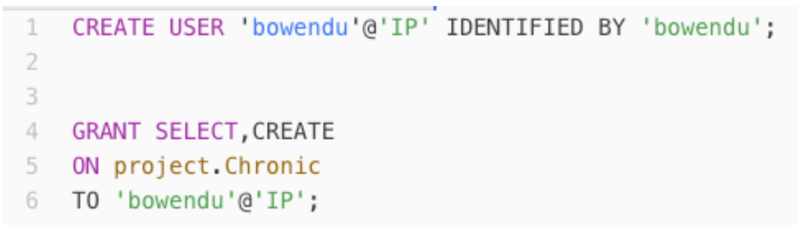


** Note: Because almost all attributes are quantified into numerical categories that showcase the intensity one variable is attri
consume/do/execute an infinite combination of a certain behaviors (e.g. smoking, drinking, exercising). Therefore, the notatio


** Note: Because almost all attributes are quantified into numerical categories that showcase the intensity one variable is attributed to an individual, one could technically consume/do/execute an infinite combination of a certain behaviors (e.g. smoking, drinking, exercising). Therefore, the notation "0..." was employed.

Sharing Database:

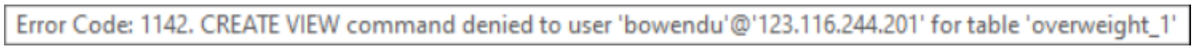
This is another difficult aspect we are facing, since me and my teammates are in different time zone. We worked on our databased mostly without 24 communications, so we could only tell each other what we did on the database. For example, our database is not consistent, some of the query can run on my MacOS but has issues at his Windows. Therefore, I launched AWS-RDS instance for our database, as a result, it solves all the query problems and cost us less time to communicate. Another cool feature is that after I set up my connection, I can create a new user just from my SQL file!!!!!!!! (all the grant is in submitted file)



```sql
CREATE USER 'bowendu'@'IP' IDENTIFIED BY 'bowendu';


GRANT SELECT,CREATE
ON project.Chronic
TO 'bowendu'@'IP';
```

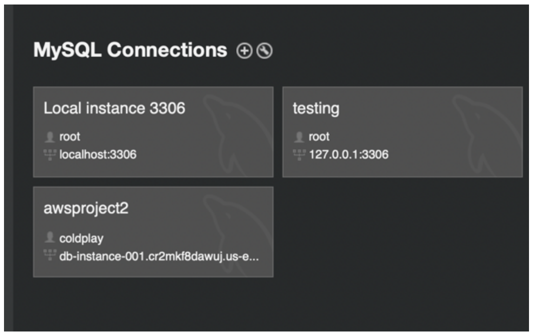
And give him permission for the database. While we are testing this functionality, I only gave him limited access. Therefore, he got a nice 'access denied error' when creating views. We know this functionality is not required but be able to working on the same database server is just fantastic experience and mimic real work experience.



```
Error Code: 1142. CREATE VIEW command denied to user 'bowendu'@'123.116.244.201' for table 'overweight_1'
```

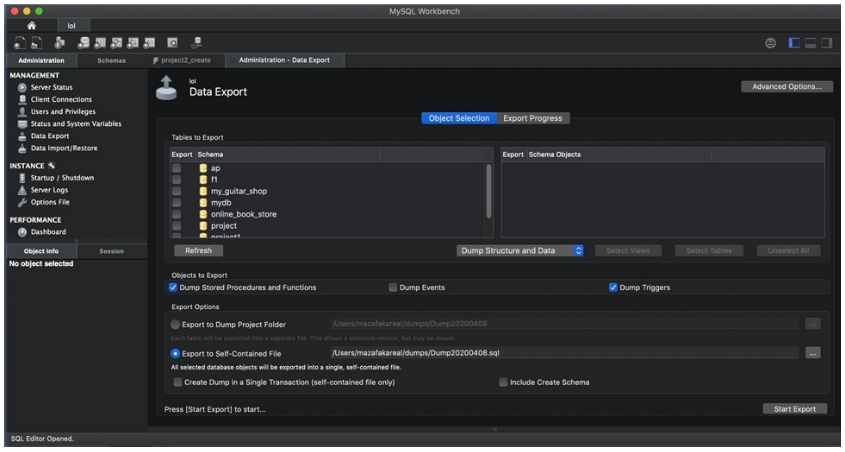
Using DUMP:

First: establish the connection using workbench.



MySQL Connections

Local instance 3306 root localhost:3306	testing root 127.0.0.1:3306
awsproject2 coldplay db-instance-001.cr2mkf8dawuj.us-e...	



No object selected

lol
Data Export

Tables to Export

Export	Schema
☐	ap
☐	f1
☐	my_guitar_shop
☐	mydb
☐	online_book_store
☐	project
☐	project1

Export	Schema Objects

Dump Structure and Data

Objects to Export

[✓] Dump Stored Procedures and Functions ☐ Dump Events [✓] Dump Triggers

Export Options

○ Export to Dump Project Folder /Users/mazafakareal/dumps/Dump20200408

Each table will be exported into a separate file. This allows a selective restore, but may be slower.

● Export to Self-Contained File /Users/mazafakareal/dumps/Dump20200408.sql

All selected database objects will be exported into a single, self-contained file.

☐ Create Dump in a Single Transaction (self-contained file only) ☐ Include Create Schema

Press [Start Export] to start...

SQL Editor Opened.

Second: export from local schema as dump, and import that dump into aws mysql instance. From there, our work is in sync to the end.

Summary:

Me and Bowen both dedicated huge amount of efforts into this project. Debugging on the Frontend and backend is just painful.

Our video link: https://www.youtube.com/watch?v=FWI5juuZX10&t=51s

Bowen Du: 50% effort
Shizhen Li: 50% effort

Bowen works mainly on Backend while I work on frontend. And we split the work for database in half. Backend cost a huge amount of time just to debug and running the queries.

Database side, Bowen worked on views, mega table, UML, index.
I worked on SP, create smaller tables, normalization. I recorded the video because Bowen is not able to access YouTube based on where he is now.