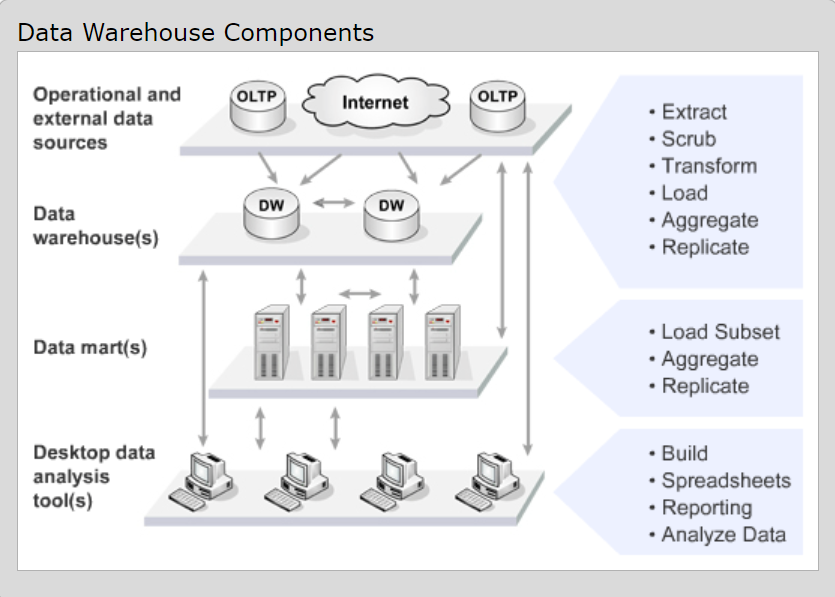
# 1. Data Warehouse Introduction

The definition of Data Warehouse is “An integrated, subject-oriented, time-variant, nonvolatile database that provides support for decision making”.

Data warehouses are part of a larger decision support infrastructure which is illustrated in the following figure.



From the figure you can see that data is loaded into data warehouses from operational (OLTP) databases and from other sources, which are often on the web. This processing to load the warehouses is usually referred to as Extract, Transform and Load, or ETL. ETL processing usually includes extracting the data from operational and other data sources, scrubbing to remove errors in the data, the computation and updating of aggregates such as daily and monthly summaries, transformation of the data into a dimensional or multidimensional form, and updating of the data warehouse database.

The figure also shows how data marts are loaded from data warehouses. The data marts are often designed to support particular functional areas such as finance, marketing, production, or sales. One advantage of loading data marts from data warehouses are that this localizes in the warehouse ETL the often complex processing to extract, scrub and reorganize the data for decision support. The data is often loaded into the marts after the warehouse ETL cycle has finished for a day, using queries that fetch just the data needed by each mart. Data may also be loaded from the warehouse into the marts on demand in response to user requests. In this way the users of the marts have nearly interactive access to all of the data in the warehouses, without the costs of storing that data in the mart databases. When marts are fed from warehouses the marts do not each have to deal with the considerable complexity of extracting the data from diverse sources, scrubbing it, and transforming it into useful information.

# 2. Dataset Description

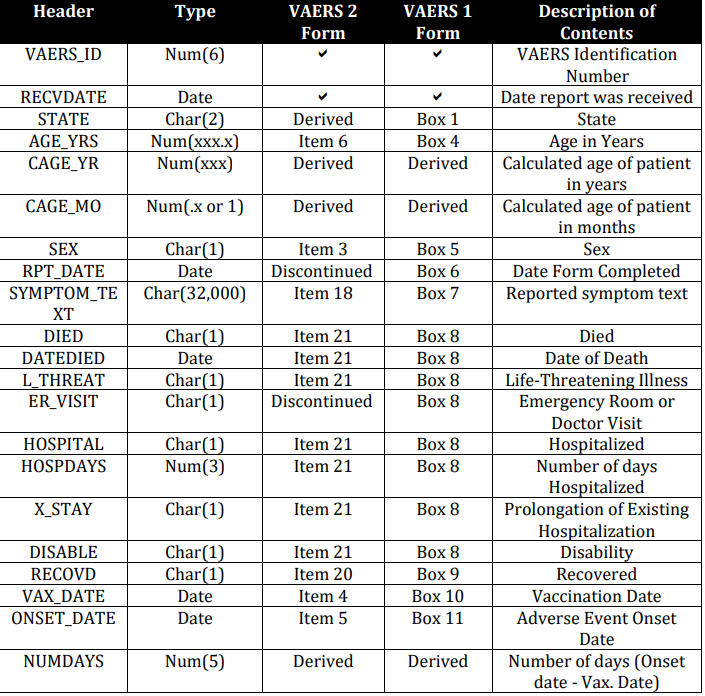
VAERS Data Sets （Vaccine Adverse Event Reporting System）is is directly downloaded from <https://www.kaggle.com/elenaeb/2021-vaers-vaccination-symptoms-adverse-data>

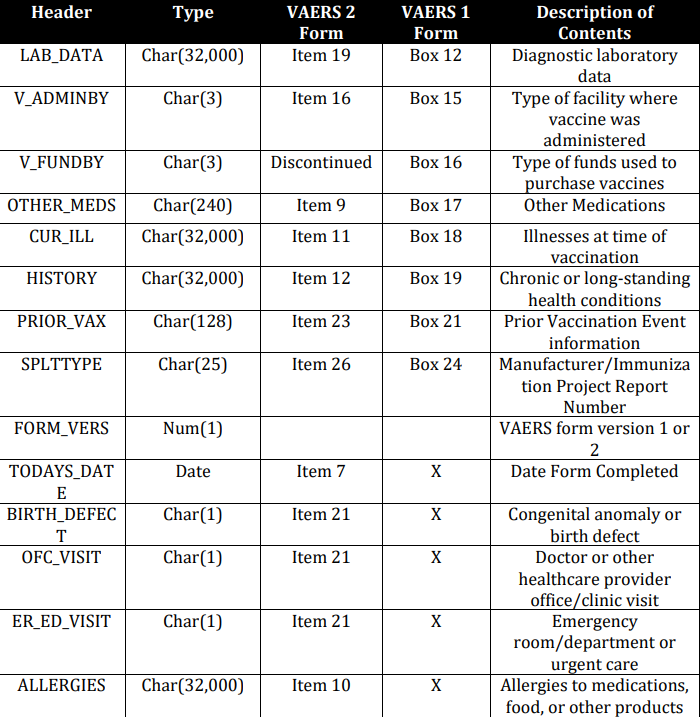
## Dataset Background

The Vaccine Adverse Event Reporting System (VAERS) was created by the Food and Drug Administration (FDA) and Centers for Disease Control and Prevention (CDC) to receive reports about adverse events that may be associated with vaccines. No prescription drug or biological product, such as a vaccine, is completely free from side effects. Vaccines protect many people from dangerous illnesses, but vaccines, like drugs, can cause side effects, a small percentage of which may be serious. VAERS is used to continually monitor reports to determine whether any vaccine or vaccine lot has a higher than expected rate of events. Doctors and other vaccine providers are encouraged to report adverse events, even if they are not certain that the vaccination was the cause. Since it is difficult to distinguish a coincidental event from one truly caused by a vaccine, the VAERS database will contain events of both types. In addition, it is often the case that more than one vaccine was administered, making it difficult to know to which of the vaccines the event might be attributed. In analyzing individual reports, researchers examine the medical information about the event, and obtain more specific information from the reporters whenever necessary. Patterns of reporting associated with vaccines and vaccine lots are also analyzed.

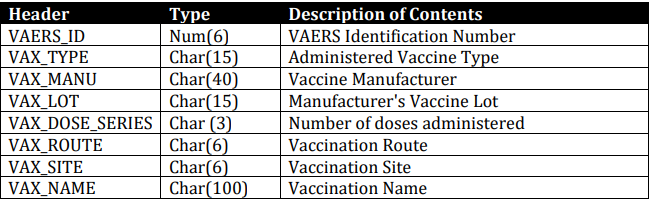
The downloadable VAERS public data set consists of three separate data files. These files are provided by calendar year beginning with the first VAERS reports reported in the latter part of 1990. In this project, I downloaded the dataset of 2021. The public data set is updated periodically, and the date of the update is referenced on the website. We currently accept the 2 versions of the VAERS form; fields in the VAERS 2 form are referred to as Items and Boxes in the VAERS 1 version. Comma-separated-value (CSV) files are industry-standard text files compatible with most of the major database or statistical analysis products on the market. Each data set is available for download in 2 formats: as three separate CSV files or as a compressed Zip file that contains the three CSV files listed for the specific year.

## Data information of file 2021VAERSDATA.csv

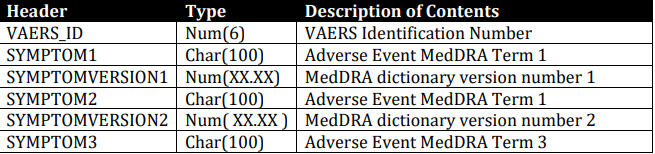


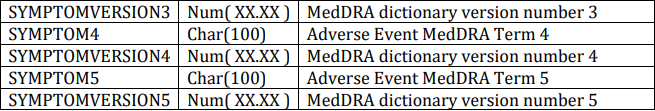


## Data information of file 2021VAERSVAX.csv



## Data information of file 2021VAERSSYMPTOMS.csv





# 3. Design the Data Warehouse

## Key Business questions

（1） What are the symptoms at different ages after getting the COVID-19 vaccine? Which symptoms are the most?

（2）Looking at the vaccines developed by different companies for COVID-19, which company developed the vaccines that produced the most adverse reactions? Which company's vaccine produced the fewest adverse reactions?

（3）What is the mortality rate after vaccination against COVID-19 in different states?

（4）Please find recovery rates and highest onset days after Covid-19 vaccination for men and women by age group.

（5）Which year has the greatest number of deaths and life threat at risk from vaccinations?

## Design Facts and Dimensions

Time dimension: time dimension, it has a primary key called TimeID, and three attributes: Day, Month and Year.

Vax dimension: vax dimension represents the vax information that the vaers data included.

Symptom dimension: symptom dimension represents the symptoms showing after taking some specific vax.

Age group dimension: age group dimension is defined by the certain age range. Such as child(0,6), teenager [6,18), youth [18,35), middle-aged[35,60) and elderly age [60, )

Sex dimension: the gender dimension, it represents the female and male. Some vaers may not show gender information, it is U.

Address dimension：the state info of the vaers information

Vaers dimension：it’s a junk dimension, it combined whether person are in danger, whether they die, whether they recover, etc.

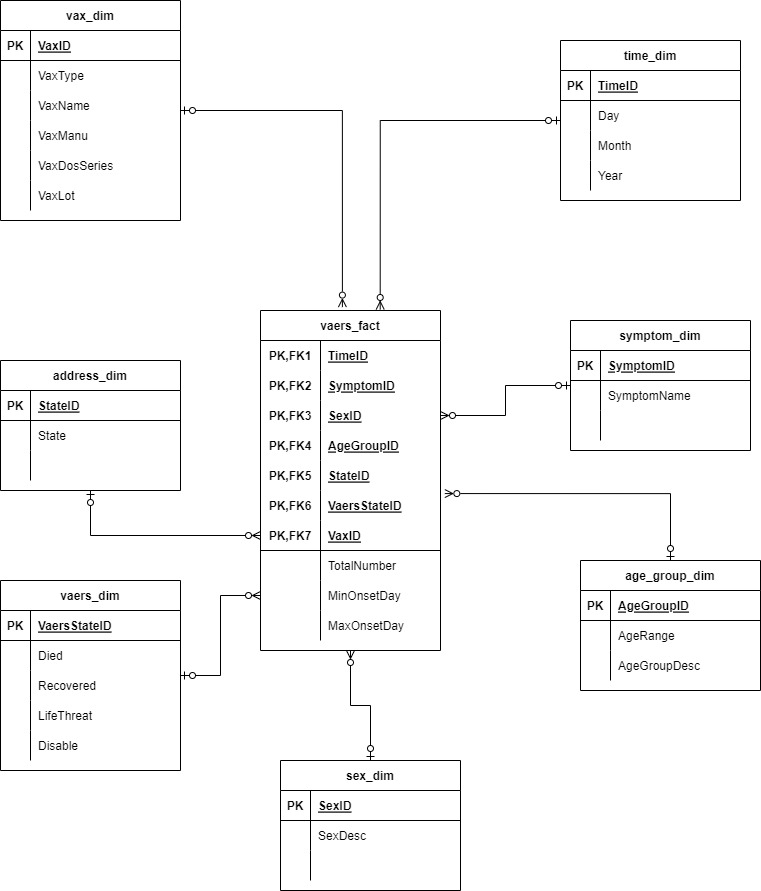
Vaers fact:

(1) total number: it measures the number of persons.

(2) min onset day: it measures the min value of vaers onset day.

(3) max onset day: it measures the max value of vaers onset day.

## Design ERD



# 4. Design and Implement ETL Processing

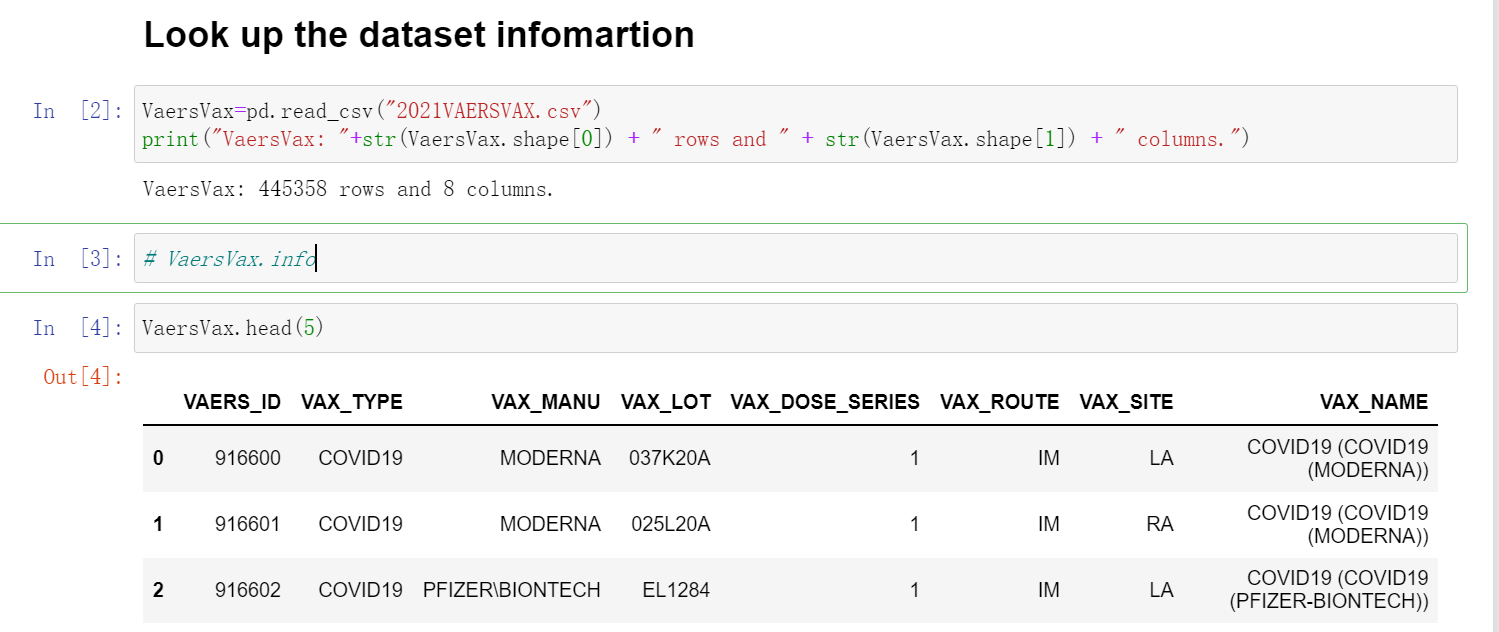
ETL means “Extract, Transform and Load” is the processing that collects data from operational databases and other data sources, cleanses it, transforms it, summarizes it, and loads it into a data warehouse.

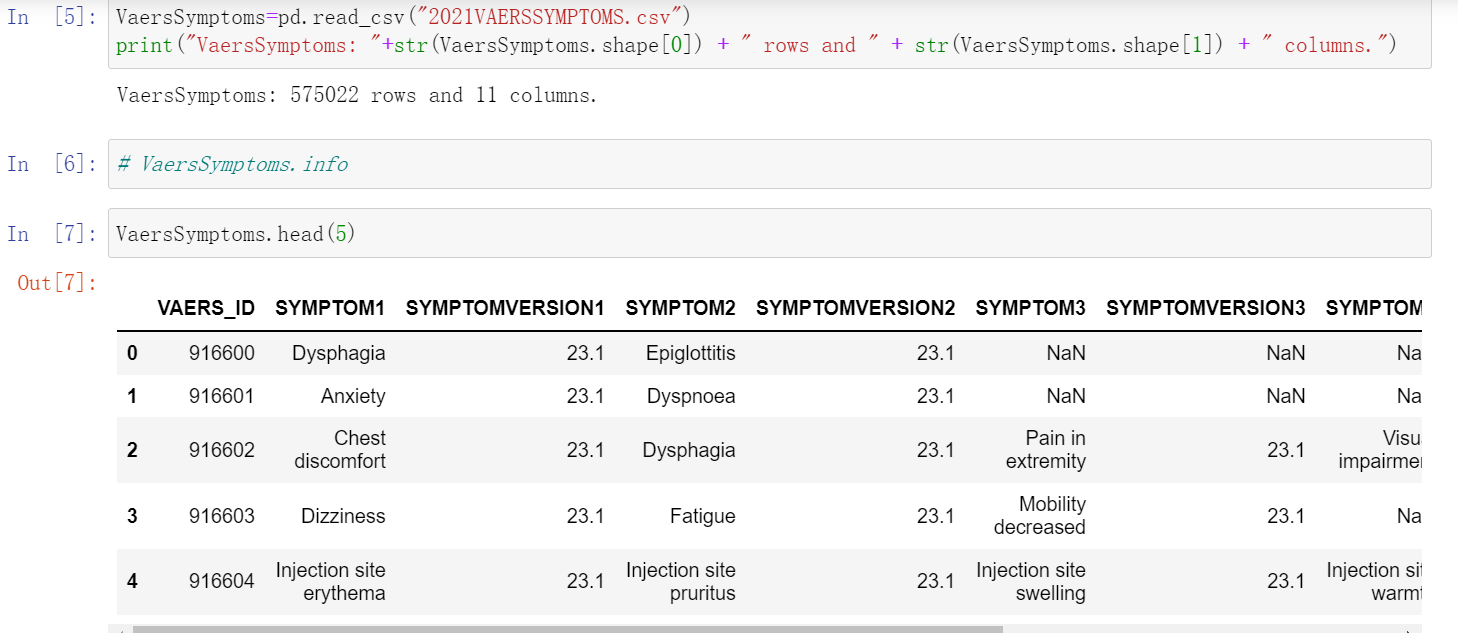
The way the data warehouse gets data may be from numerous sources. Much of the data loaded into data warehouses comes from the various Operational Data Stores, OLTP systems that are running active business processes. These are typically regular relational databases. It operates more or less as a snapshot of the organization as it is running. Because this data is already structured, you can expect that the extraction will be pretty consistent, which means reliable.

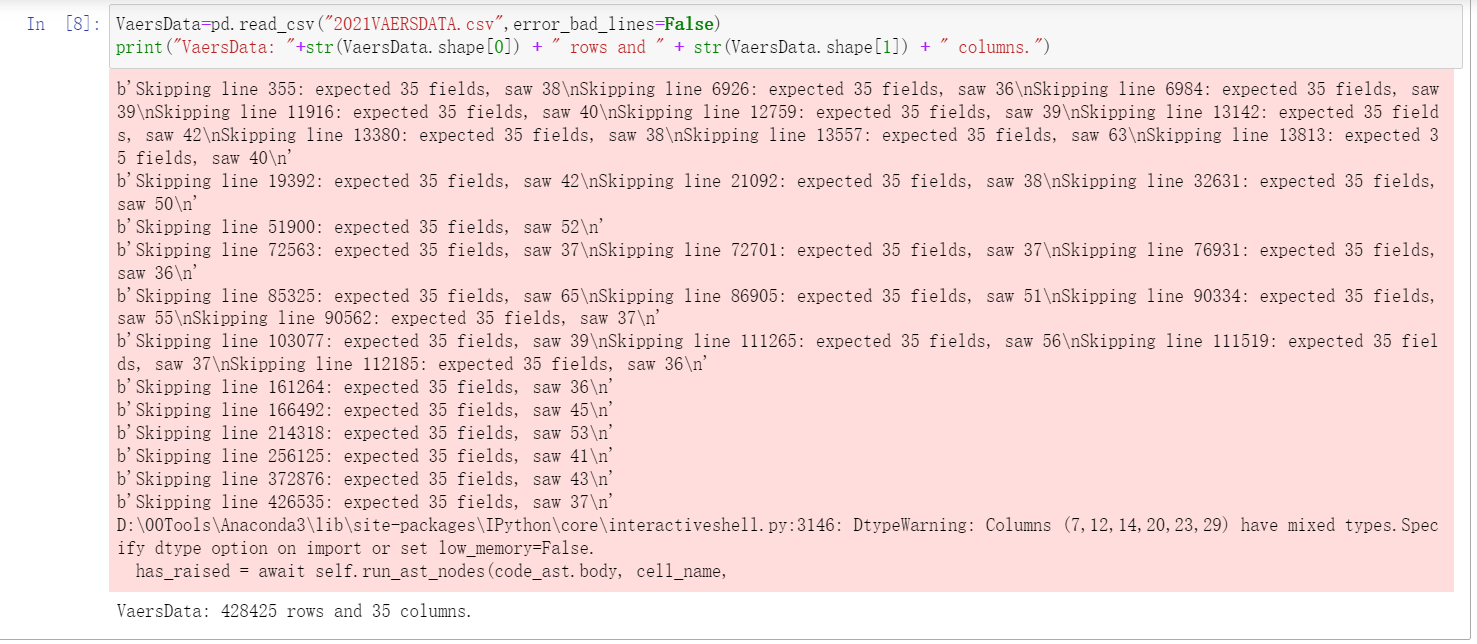
In my project, I choose the dataset as the data sources which is more likely a transactional dataset.

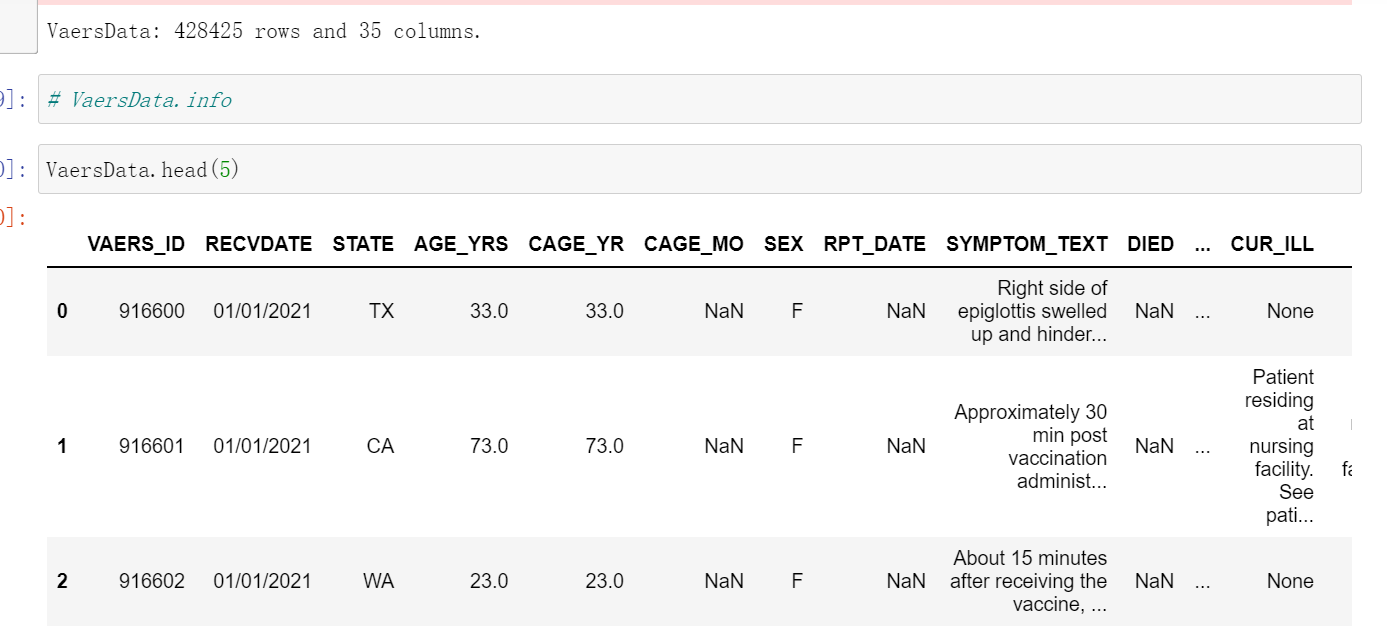
## Extraction

Assume that the dataset I downloaded is that I need to extract from. Here I use Python to extract the dataset. And the steps like following.









## Transformation

### Data Cleansing Concept

After the data extraction is completed, it is necessary to pay attention to whether the integrity and consistency of the data are satisfied, and the data often obtained may often include the following categories:

(1) Incomplete data

This type of data is mainly due to the lack of some information, such as the name of the supplier, the name of the branch, the lack of regional information of the customer, and the mismatch between the main table and the detailed table in the business system. For this type of data, the missing content is written into different Excel files and submitted to the customer, and is required to be completed within the specified time. It is written to the data warehouse after completion.

(2) Bad data

The reason for this type of error is that the business system is not sound enough, and it is directly written to the background database without making a judgment after receiving the input. Correct, date out of bounds, etc. This type of data should also be classified. For problems like full-width characters and invisible characters before and after the data, they can only be found by writing SQL statements, and then the customer is required to extract them after the business system is revised. Errors such as incorrect date format or date out-of-bounds will cause ETL to fail. This type of error needs to be picked out by SQL in the business system database, and handed over to the business department for correction within a time limit, and then extracted after correction.

(3) Duplicate data

For this type of data - especially in dimension tables - export all fields of duplicate data records for the client to confirm and organize.

Data cleaning is an iterative process that cannot be completed in a few days, only to constantly find and solve problems. As for whether to filter, whether to correct or not, the customer is generally required to confirm. For the filtered data, write the filtered data into an Excel file or write the filtered data into the data table. In the early stage of ETL development, you can send emails of the filtered data to the business units every day, urging them to quickly Correcting errors, it can also be used as a basis for future verification of data. Data cleaning needs to pay attention not to filter out useful data, carefully verify each filtering rule, and ask the user to confirm.

(4) Data cleansing

The process of re-examining and verifying data to remove duplicate information, correct existing errors, and provide data consistency.

Data cleansing is also seen from the name as "washing out" the "dirty", which refers to the last procedure to find and correct identifiable errors in data files, including checking data consistency, dealing with invalid and missing values, etc. Because the data in the data warehouse is a collection of data oriented to a certain topic, these data are extracted from multiple business systems and contain historical data, so it is inevitable that some data are wrong data, and some data are inconsistent with each other. Conflicts, these erroneous or conflicting data that we obviously don't want, are called "dirty data". We have to "wash out" the "dirty data" according to certain rules, which is data cleaning. The task of data cleaning is to filter those data that do not meet the requirements, and hand over the filtered results to the business department to confirm whether it is filtered out or corrected by the business unit before extracting. The data that does not meet the requirements are mainly divided into three categories: incomplete data, wrong data, and duplicate data. Data cleaning is different from questionnaire review, and data cleaning after input is generally done by computer rather than manually.

(5) Consistency check

Consistency check is to check whether the data meets the requirements according to the reasonable value range and mutual relationship of each variable, and find data that is beyond the normal range, logically unreasonable or contradictory. For example, a 0 value for a variable measured on a scale of 1-7 and a negative weight for weight should be considered outside the normal range. Computer software such as SPSS, SAS, and Excel can automatically identify each variable value that is out of range according to the defined value range. Answers with logical inconsistencies can come in many forms: for example, many respondents say they drive to work and report not having a car; or respondents report being heavy buyers and users of a brand, but at the same time familiar with it. A low score was given on the degree scale. When inconsistencies are found, the questionnaire serial number, record serial number, variable name, error category, etc. should be listed for further verification and correction.

(6) Handling of invalid and missing values

Due to survey, coding and entry errors, there may be some invalid and missing values ​​in the data that need to be handled appropriately. Commonly used processing methods are: estimation, whole case deletion, variable deletion and pairwise deletion.

Estimation. The easiest way to do this is to replace invalid and missing values ​​with the sample mean, median, or mode of a variable. This method is simple, but does not fully consider the existing information in the data, and the error may be large. Another way is to estimate through correlation analysis or logical inference between variables based on respondents' answers to other questions. For example, the ownership of a certain product may be related to household income, and the possibility of owning this product can be calculated based on the household income of the survey respondents.

Casewise deletion is the removal of samples with missing values. Since many questionnaires may have missing values, the result of this practice may lead to a significant reduction in the effective sample size, and the data that has been collected cannot be fully utilized. Therefore, it is only suitable for cases where key variables are missing, or the proportion of samples with invalid or missing values ​​is small.

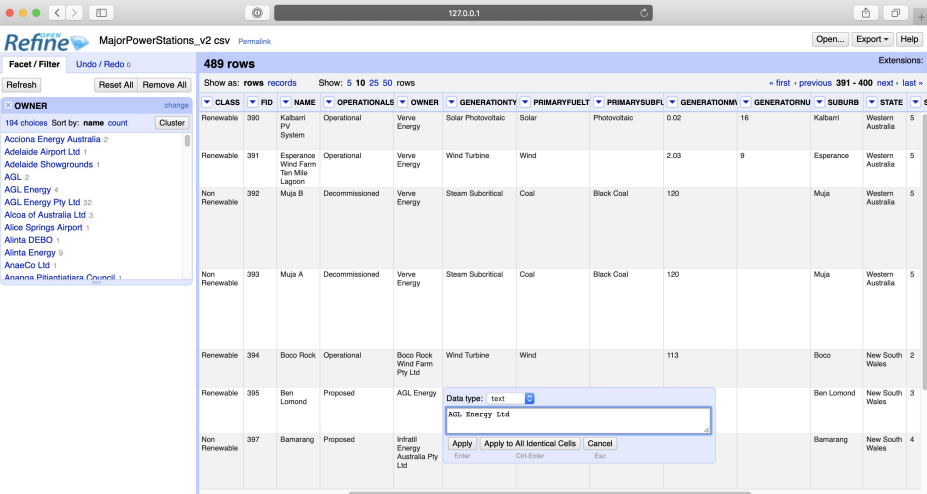
Variable deletion. If a variable has many invalid and missing values, and the variable is not particularly important to the problem under study, consider removing the variable. This practice reduces the number of variables available for analysis, but does not change the sample size.

Pairwise deletion is the use of a special code (usually 9, 99, 999, etc.) to represent invalid and missing values ​​while preserving all variables and samples in the dataset. However, only samples with complete answers are used in specific calculations, so different analyses have different effective sample sizes due to different variables involved. This is a conservative approach that preserves as much as possible the information available in the dataset.

Using different treatment methods may affect the results of the analysis, especially when the occurrence of missing values ​​is not random and the variables are clearly correlated. Therefore, in the survey, invalid and missing values ​​should be avoided as much as possible to ensure the integrity of the data.

### Data Cleansing Tools

Approach 1: Specific Data Cleansing Tools – Open Source Example: Open Refine Originally developed by Google. It allows to visually inspect and clean data with interactive user-interface. There are other commercial tools available for data cleansing.



Approach 2: Jupyter Notebooks and Python, SQL

Use jupyter notebooks and python to write code with Python and its libraries, to check for, and deal with, dirty data. It always look at the data first, before running any functions and always keep a copy of the original (before cleaning) data.

And Use SQL can easily check for and deal with missing, wrong data.

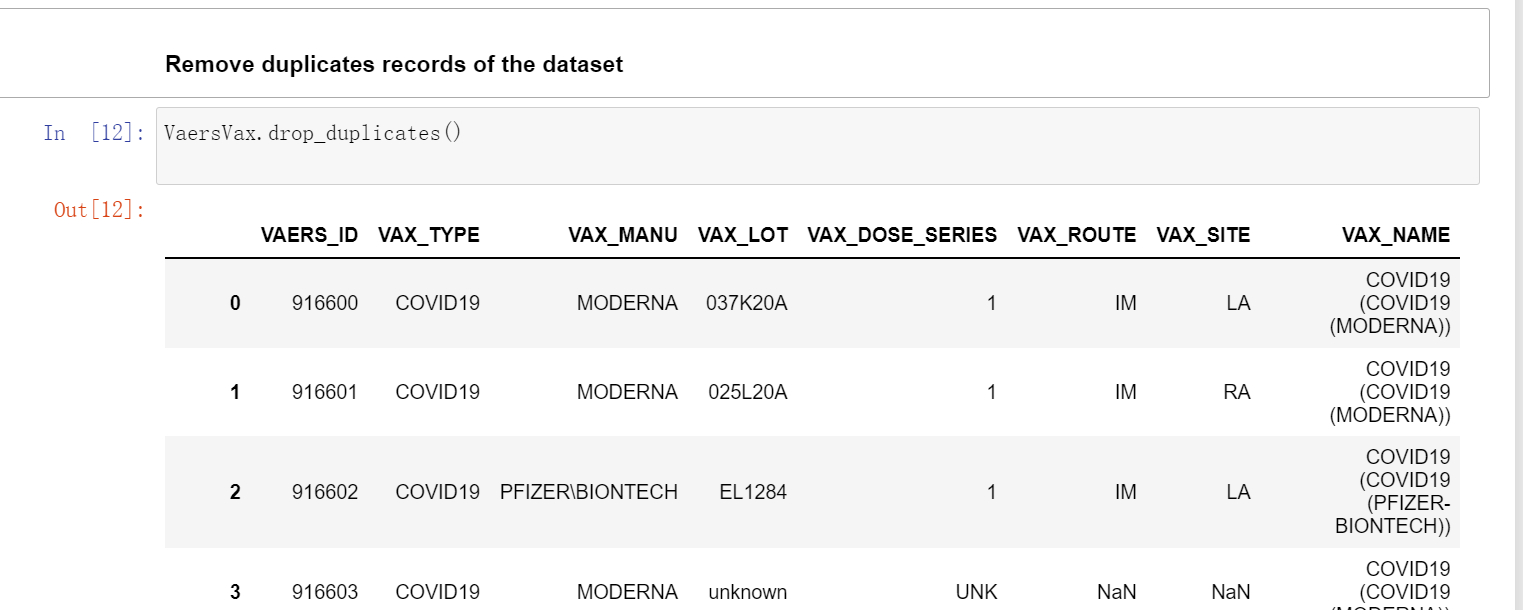
### Data Cleansing in Python And SQL

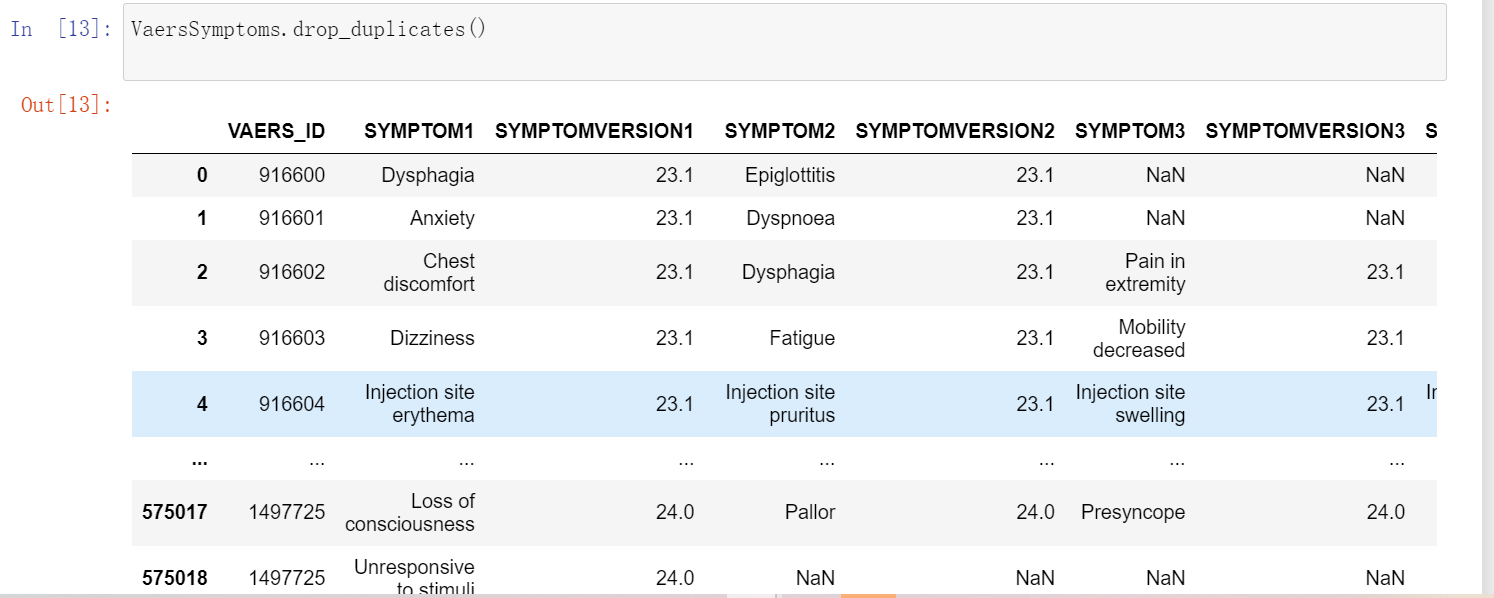
Of course, there are powerful ETL tools out there, but I do it for free in Python and SQL: (1) removing duplicated records (2) filtering of missing or inconsistent data (3) unifying semantic data representations (4) matching of entries from different sources.

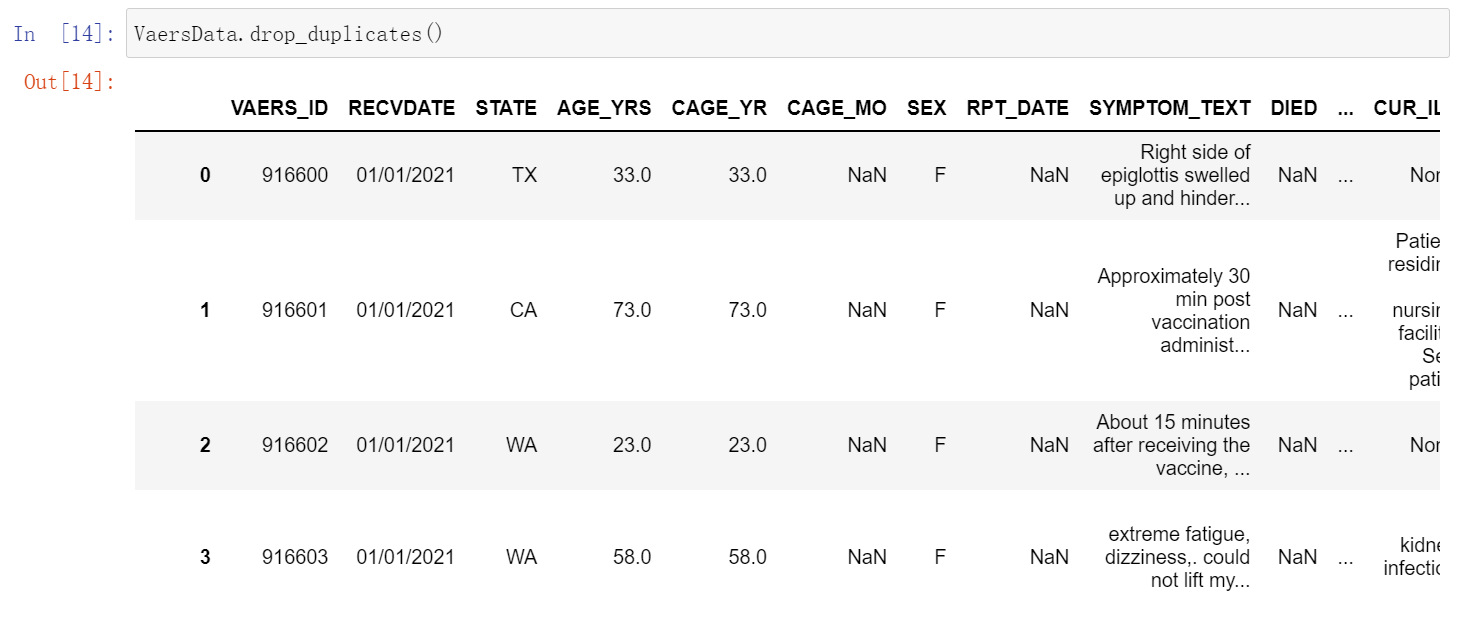
In my project, I use Pandas library to deal with data cleansing. Pandas provides various functions for handling duplicated/missing/wrong data.

Here I remove duplicates records of the dataset,

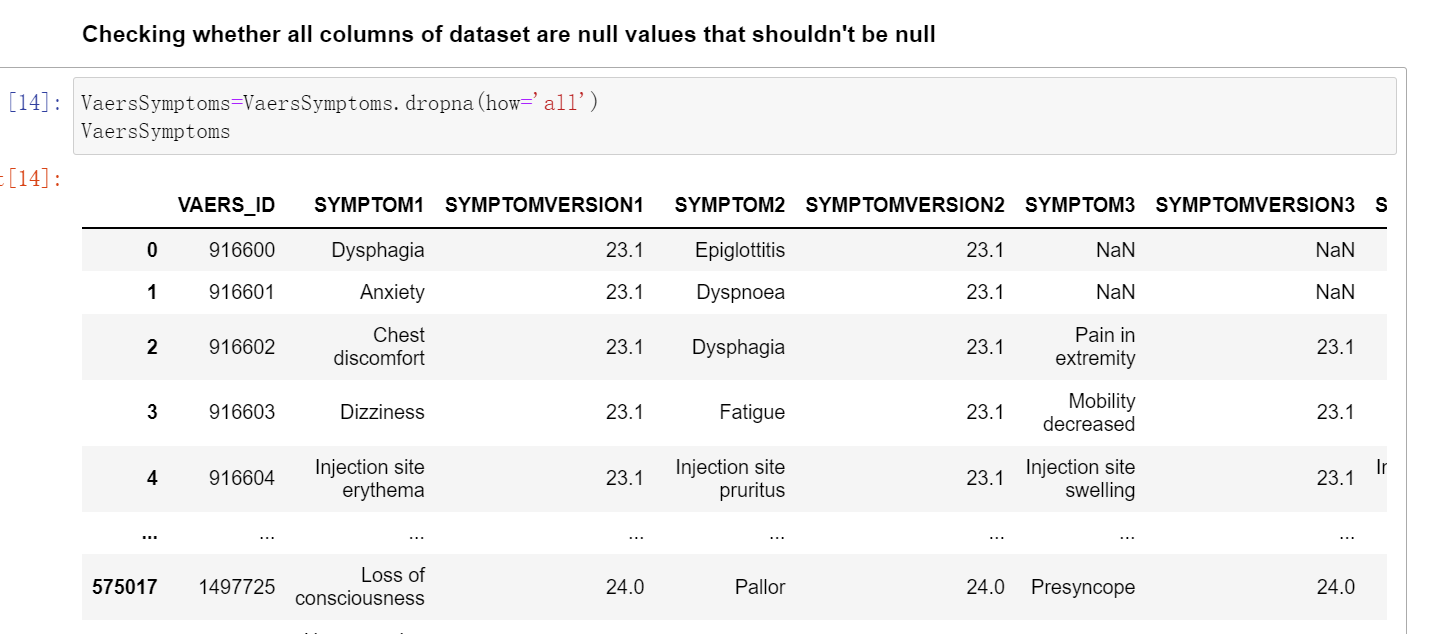
(1) Remove duplicates records of the dataset



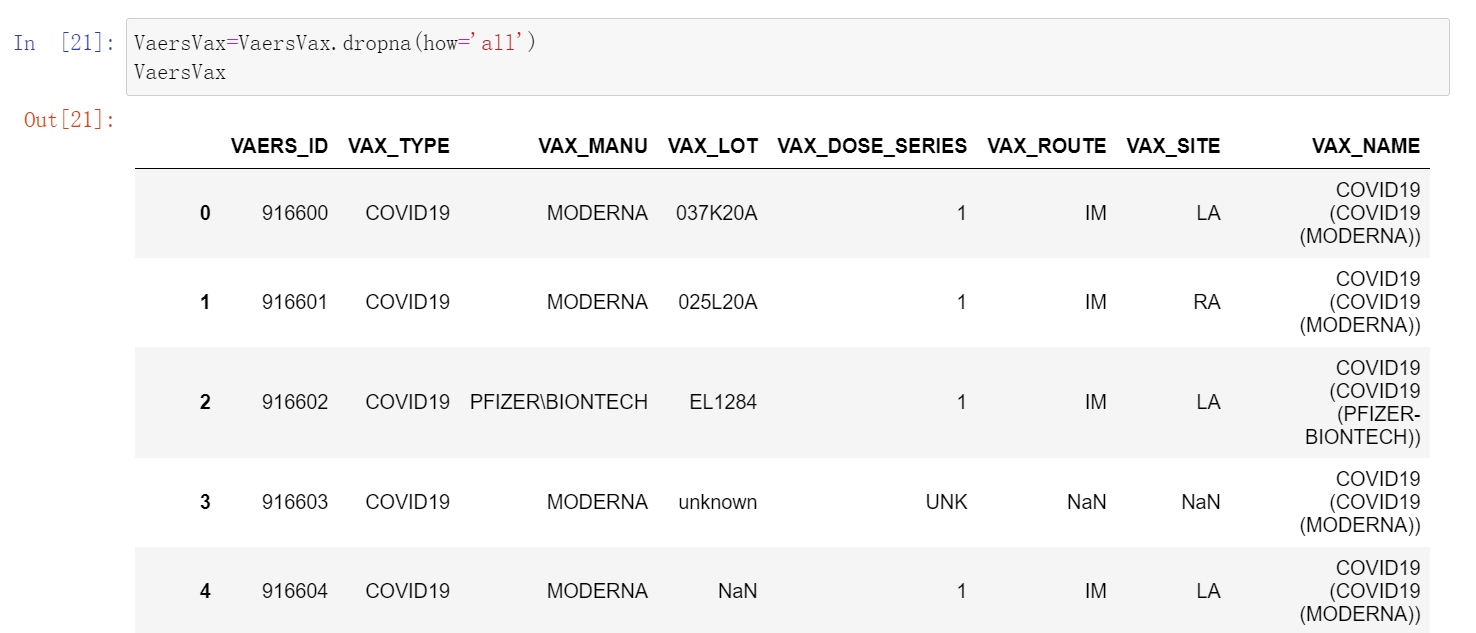




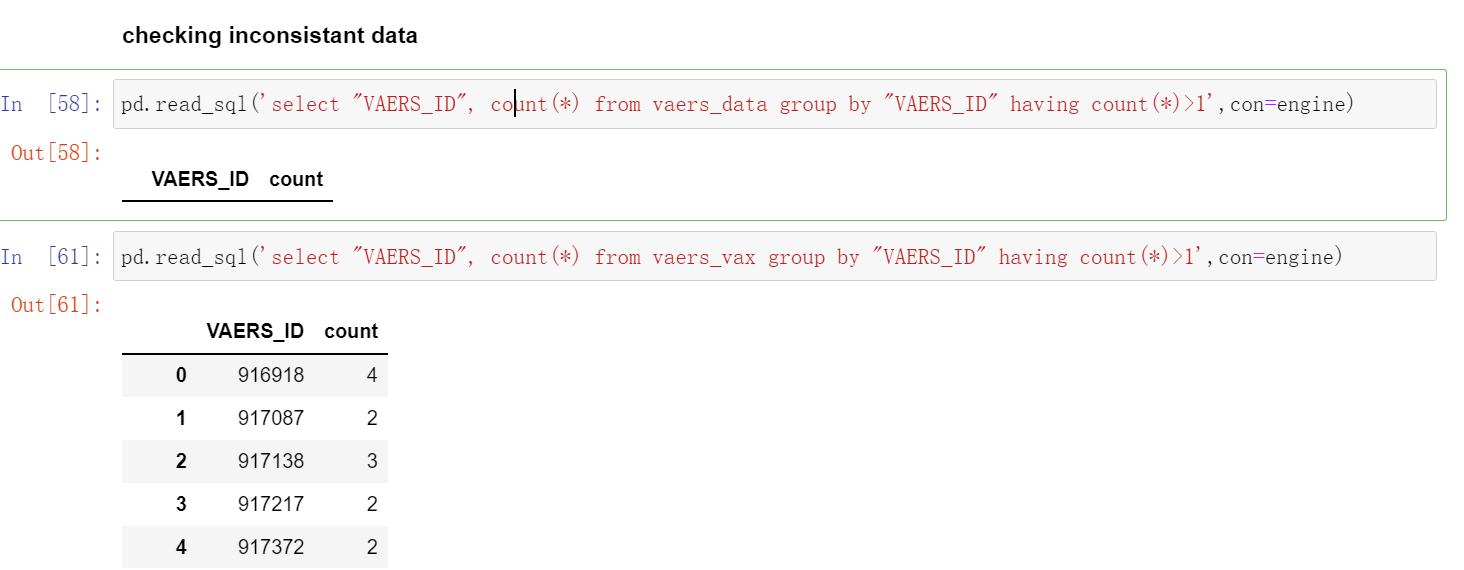
(2) Remove all null values in all columns

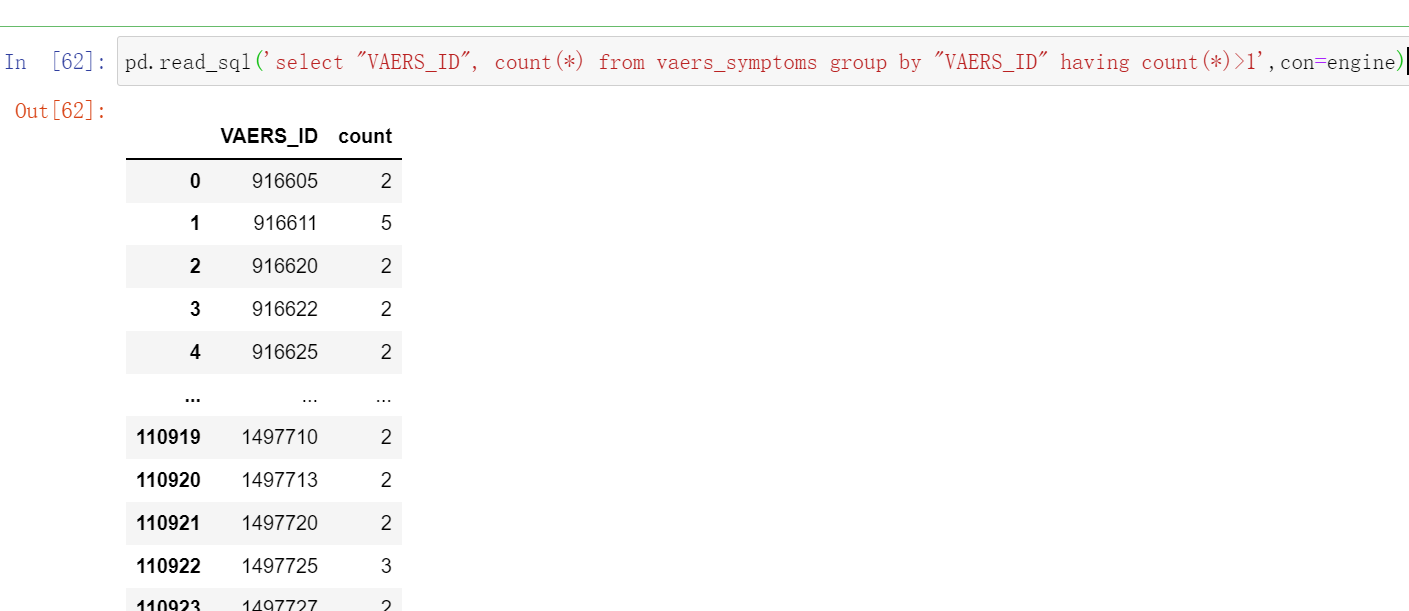






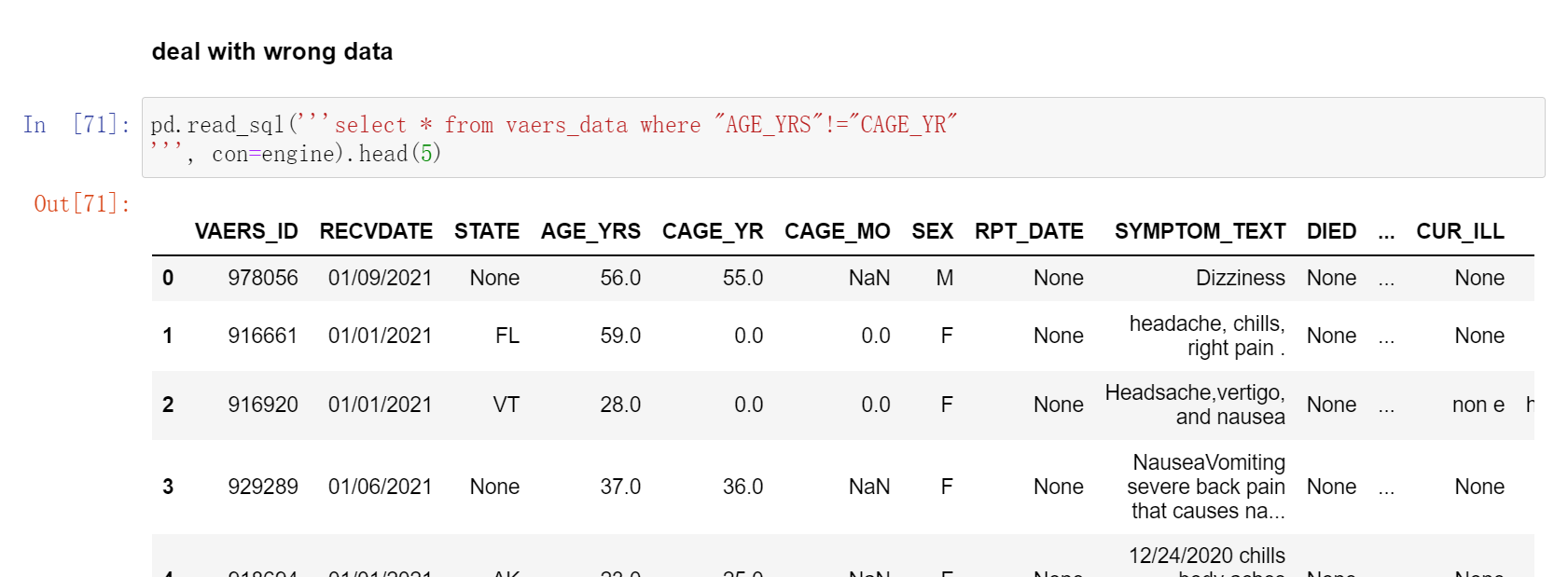
(3) checking inconsistent data and removing inconsistent data

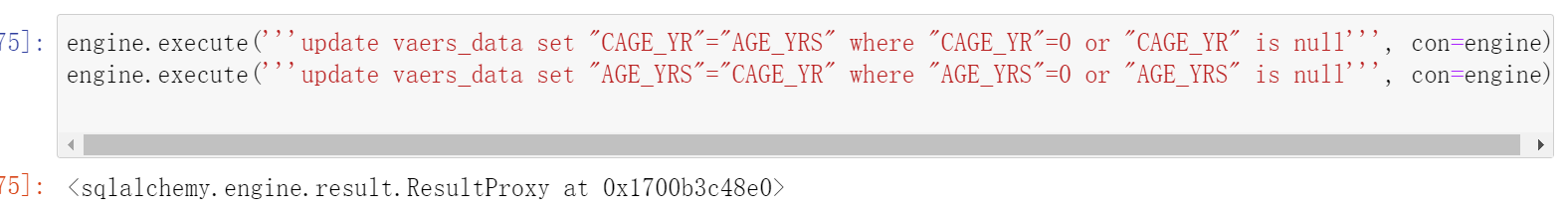




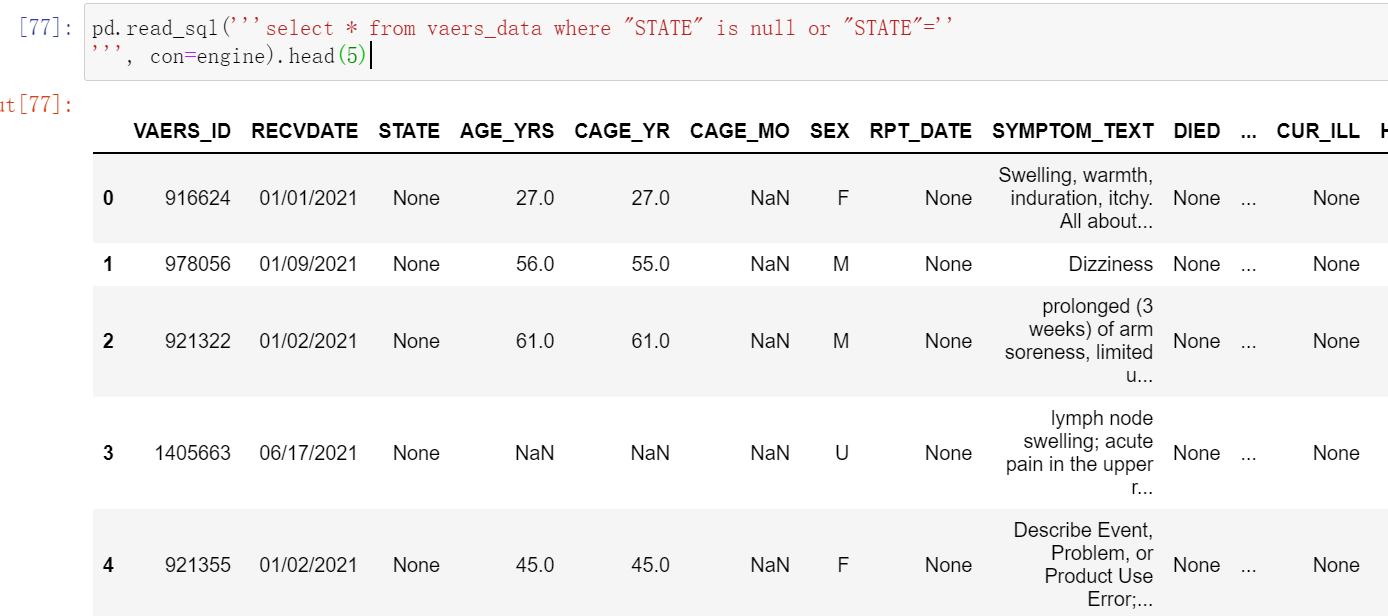


(4) deal with wrong data





(5) deal with missing data

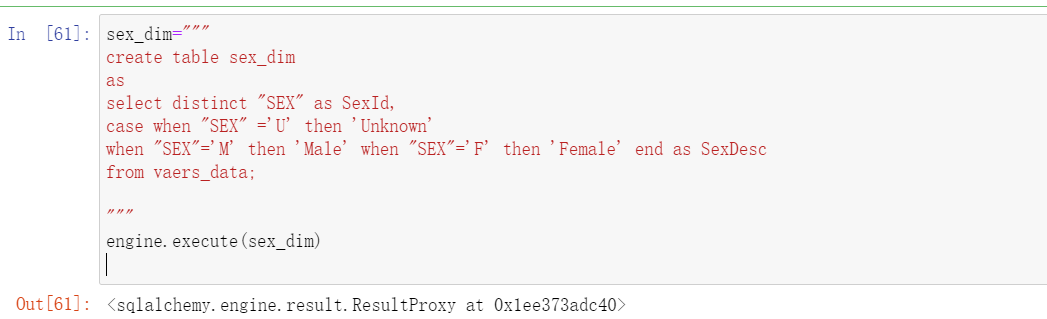


(6) Data type conversion

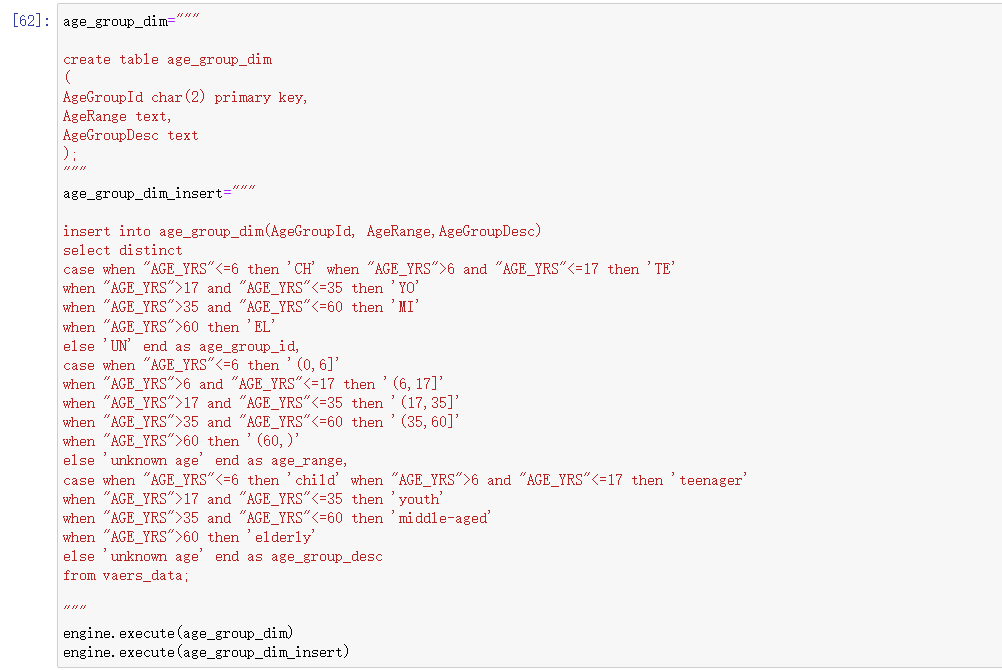


## Loading

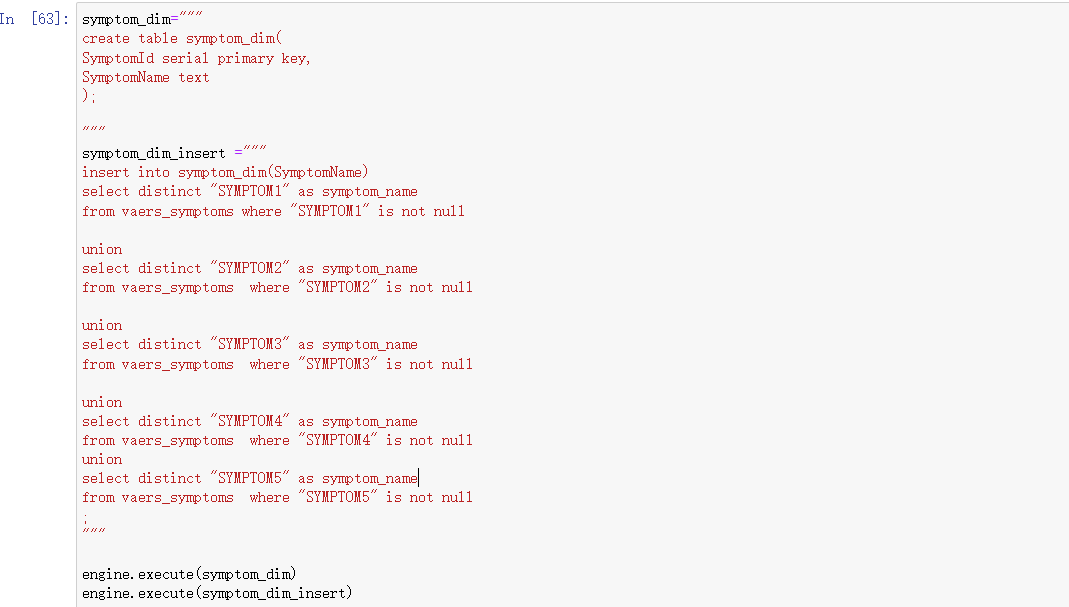
(1) Load sex dimension



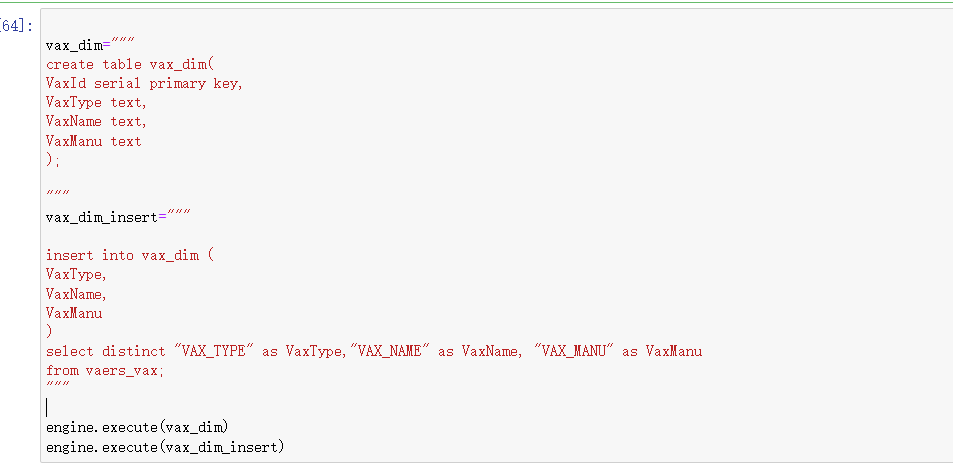
(2) Load age group dimension



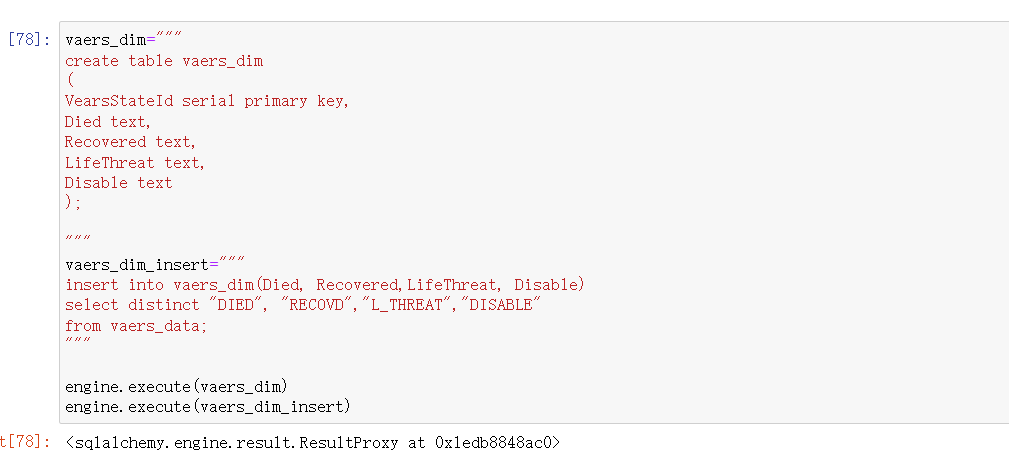
(3) Load symptom dimension



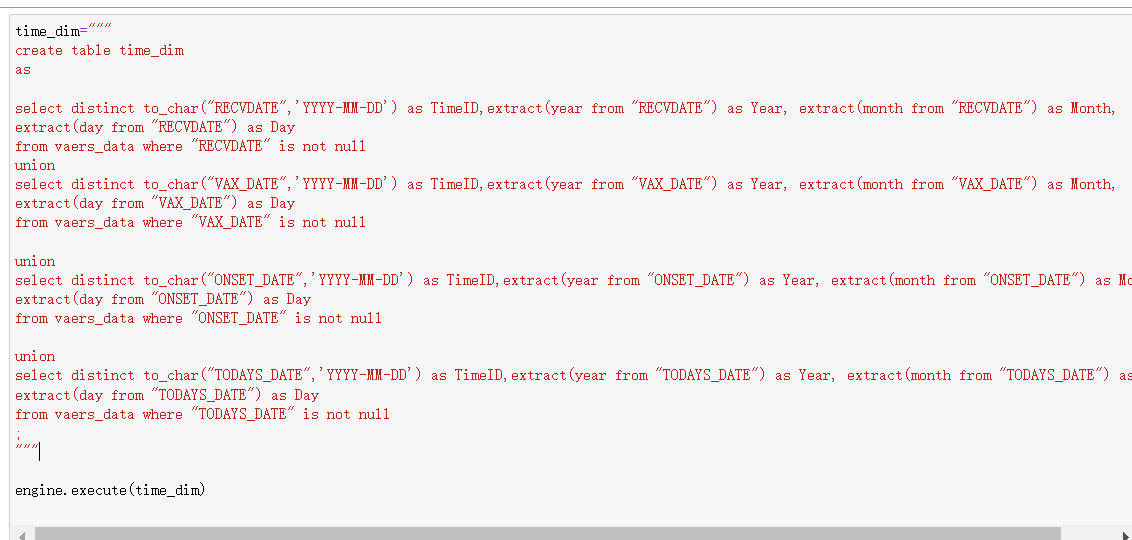
(4) load vax dimension



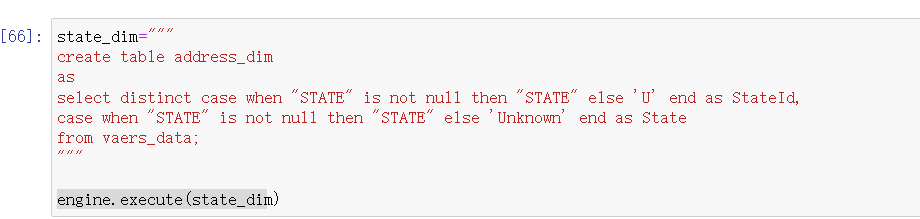
(5) load vaers dimension



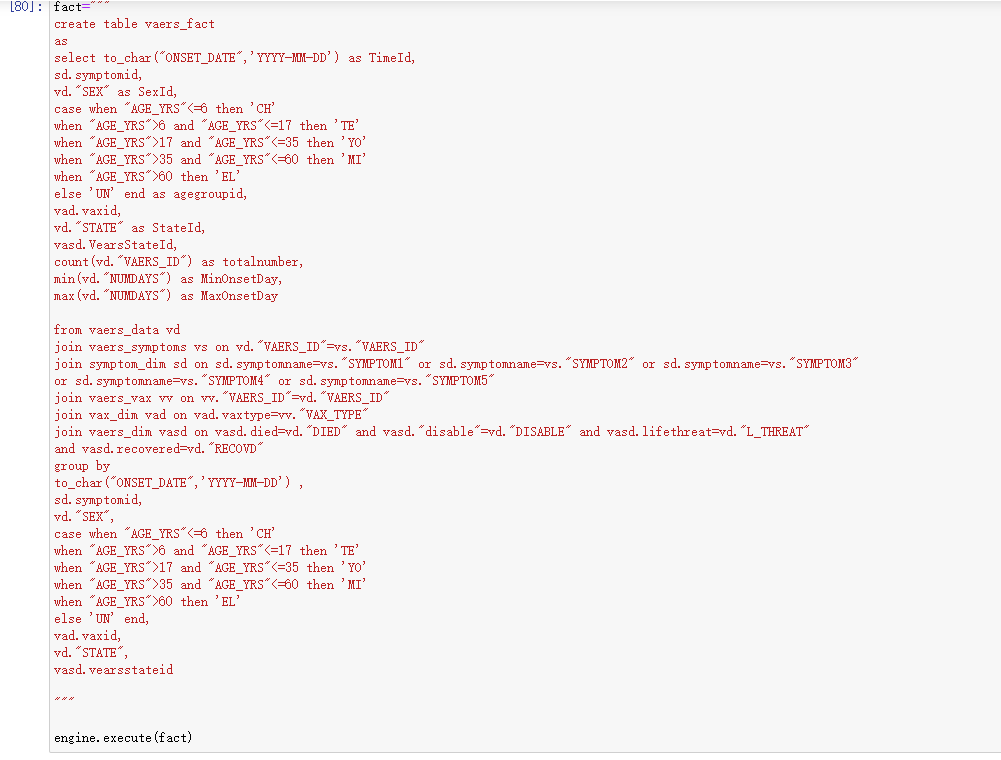
(6) load time dimension



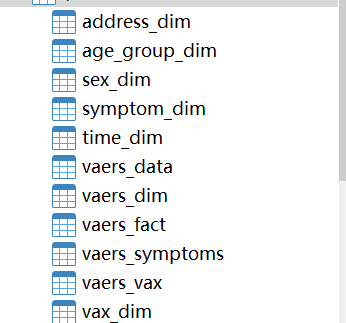
(7) load address dimension



(8) load fact table



### Loading result in database

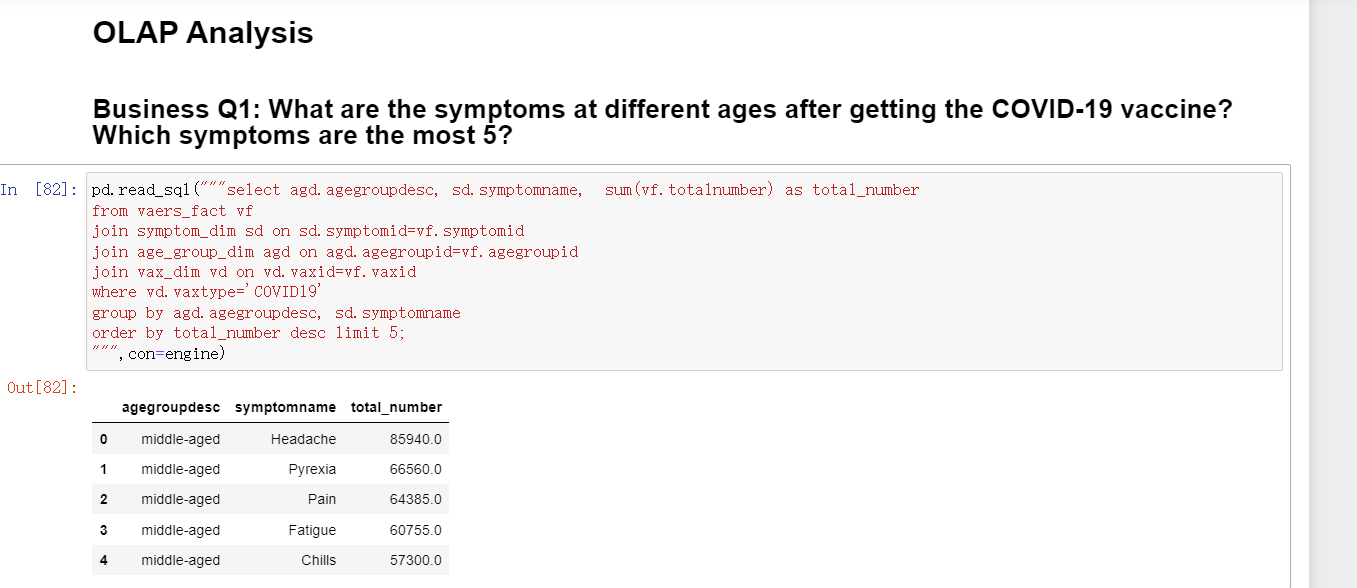


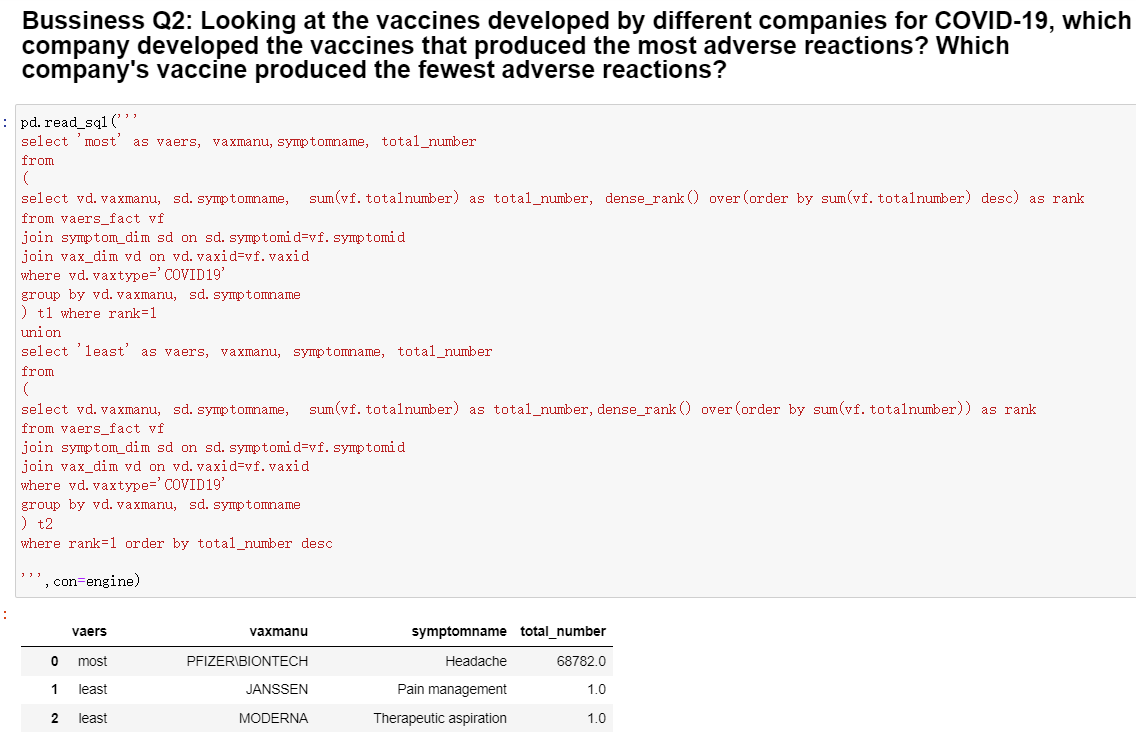
# 5. OLAP Analysis

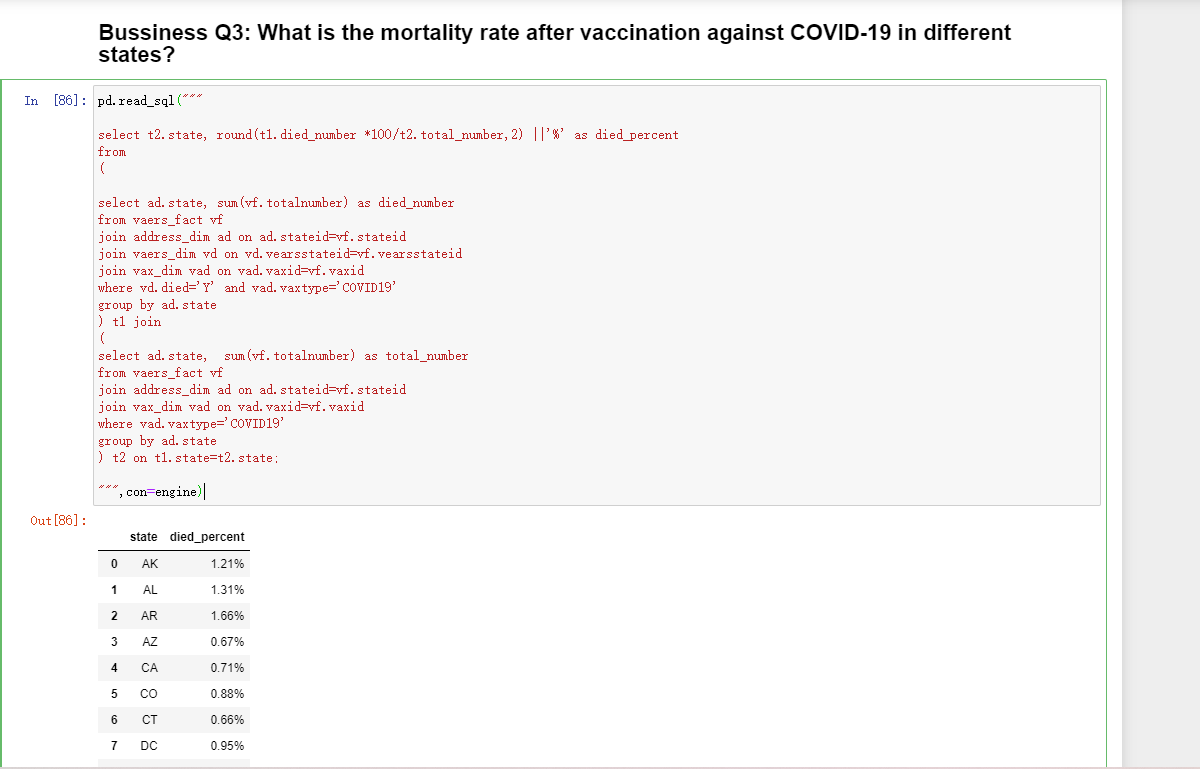
Online Analytic Processing: any decision support processing that responds to the user on a time scale that permits timely repeated queries. OLAP is distinct from overnight and other offline analytic processing.

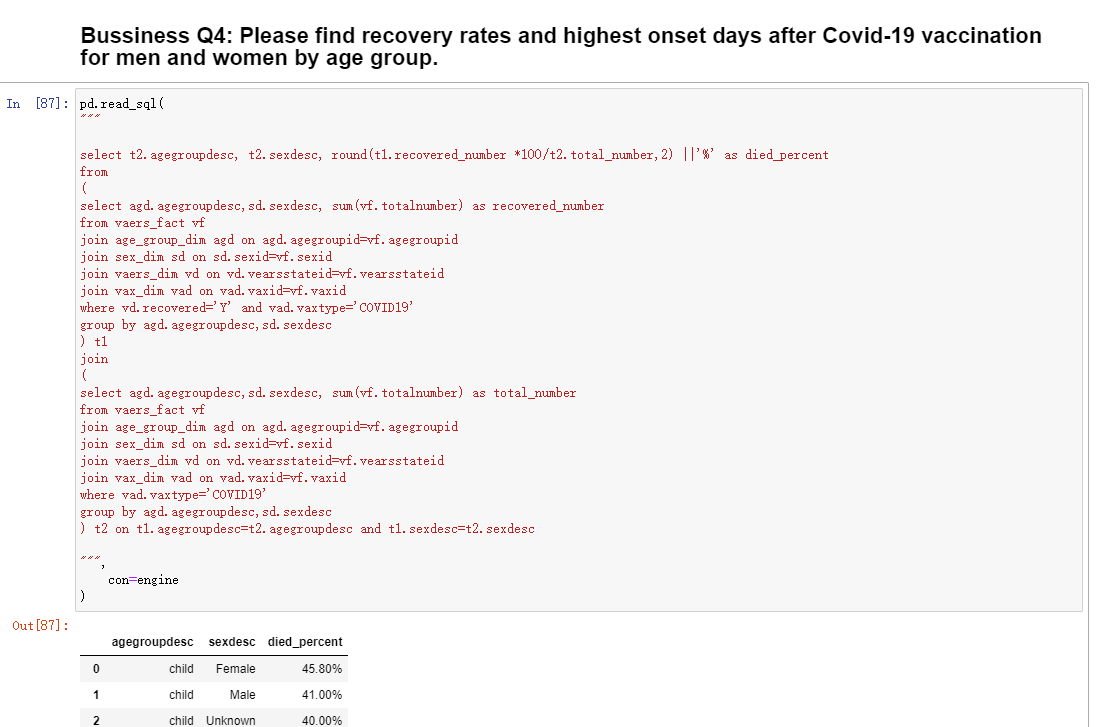
In my project, I raised some business questions based on some scenarios.

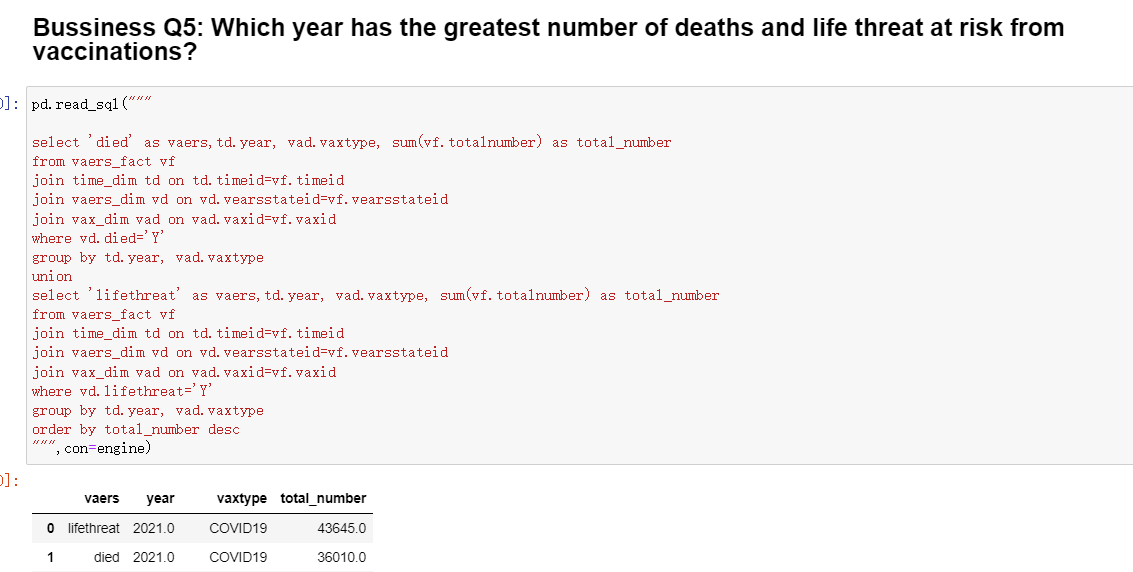
Here I use pandas to make queries based on the OLAP database.











# 6. Problem

When I finish my project, I encountered several problems that bothered me, but after I searched for the solution on the Internet, I finally solved it and successfully completed my project.

(1) When I extract data from my downloaded dataset files, I encountered the “UnicodeDecodeError”, which represented as: UnicodeDecodeError: 'utf-8' codec can't decode byte 0xa0 in position 0: invalid start byte.

I searched for the solution on the internet which tells me that the file encoding format is wrong. So I checked the file encoding and found it’s ANSI, it’s inconsistent with my python environment encoding UTF-8. So I update the dataset files encoding then problem is solved.

(2) When I extract data from my downloaded dataset files, I encountered the “ParserError”, it like “ParserError: Error tokenizing data. C error: Expected 35 fields in line 355, saw 38”. Here I check the dataset and found that the columns numbers are different for different rows in the same file. So I use “error\_bad\_lines=False” parameter to prevent this error.

# Appendix A

<https://learn.bu.edu/bbcswebdav/pid-9833813-dt-content-rid-61814506_1/courses/22sprgmetcs779_o1/course/module4/allpages.htm>

<https://www.kaggle.com/elenaeb/2021-vaers-vaccination-symptoms-adverse-data>

<https://vaers.hhs.gov/docs/VAERSDataUseGuide_November2020.pdf>

[API reference — pandas 1.4.1 documentation (pydata.org)](https://pandas.pydata.org/pandas-docs/stable/reference/index.html)

<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.to_sql.html>

<http://www.postgresql.org/docs/current/static/>