INFO 153 HW6

This assignment is to load data from CSV, populate them in a SQLite database, and run in-database analytics with SQL.

## Data Source

The data is given in the CSV format, available at:

<https://www.kaggle.com/uciml/default-of-credit-card-clients-dataset>

According to the site:

This dataset contains information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005.

Write **Python code with the SQLite** module to do the following.

## SQLite Database Setup

Create a new database, connect to it, and create a table structure corresponding to data fields in the CSV data source.

## Load Data into SQLite

Read data from the CSV file, use a loop to read each CSV line (data instance), and insert it into the SQLite table.

## In-database Query and Analytics with SQLite Data

Code in Python with related SQL statements to do the following:

1. Update the data so that marriage=2 (single) and marriage=3 (others) are merged into 2 (single).
2. Remove all data records with negative BILL\_AMT values (in any of the BILL\_AMT1 through BILL\_AMT6;
3. Select and show the first 10 records in the database table, using SELECT … LIMIT… ;
4. Select and show all records with a BILL\_AMT1 amount greater than 500k;
5. Compute the total number of records, average AGE, min LIMIT\_BAL, max LIMIT\_BAL in the data;ao
6. Count the # records, average AGE, min LIMIT\_BAL, max LIMIT\_BAL for default.payment.next.month=0 (no default) vs. default.payment.next.month=1 (default), using GROUP BY;
7. Count the # records, average AGE, min LIMIT\_BAL, max LIMIT\_BAL for each marriage group (1, 2), again using GROUP BY;
8. Count the # records in each marriage group who will default (1) vs. not default (0).

## Python with NoSQL / MongoDB

Write Python code to load the same into MongoDB and conduct the same analysis above.

## Bonus:

Create two additional tables for PaymentDefault (with DefaultCode and DefaultName fields), and Marriage (with MCode and MGroupName) for mapping related code to meaningful names (e.g. code 1 for “Married” in the Marriage table). Use JOIN in tasks 6, 7, and 8 above to run the statistics based on meaningful names (rather than numeric code).