Assignment 2

CS 412: Introduction to Data Mining (Spring 2023)
Instructor: Hanghang Tong

Release date: Feb. 2nd, 2023
Due date: Feb. 14th, 2023

- This assignment will cover the content of Chapter #3 (Data Warehousing and Online Analytical Processing).
- Feel free to discuss with other members of the class when doing the homework. You should, however, write down your own solution **independently**. ***Very Important Notes*: (1) there is a fine line between collaboration and completing the assignment by yourself and (2) aiding others to cheat would have the same consequence as the cheating itself. Please try to keep the solution brief and clear.**
- Please use Piazza first if you have questions about the assignment. Also feel free to send us e-mails and come to office hours.
- The assignment is due at 11:59 PM on the due date. We will be using Piazza for collecting homework assignments. **Please do not hand in a scan of your hand-written solution, only the typed solution (e.g., Microsoft Word, Latex, etc) will be graded**. Contact the TAs if you are having technical difficulties in submitting the assignment. We do **NOT** accept late assignment!
- The assignment should be submitted as a **single** PDF file using the name convention yourNetID_ HW2.pdf. If you use additional source code for solving problems (Note that simply using Excel for computation is not allowed, as the Excel table might not show the intermediate steps well. Please use programming languages, e.g., python, Java, C++, etc.), you are required to submit them and use the file names to identify the corresponding questions. For instance, yourNetID_HW2_P1.py refers to the python source code for Problem 1, replace netid with your netid. Compress all the files (PDF and source code files) into one zip file. Submit the compressed file ONLY. (If you did not use any source code, submitting the PDF file without compression will be fine)
- For each question, **you will NOT get full credits if you only give out a final result.** Necessary calculation steps are required. If the result is not an integer, round your result to 4 decimal places.

Problem 1. OLAP (12 points)
We would like to build a data cube of the fruit data, and want to include the following dimensions and measures:

- 4 dimensions: (Variety, Location, Size, Maturity)
- 3 measures: (Price.mean, Price.min, Fruit.num_eq_min)

It indicates that the fruit price is a function of variety, location, size and maturity. Suppose we are aggregating from the Maturity 1-D cuboid to the apex cuboid. **The meanings of these measures are explained below.**

(1) (6 points) Show that Price.mean is an algebraic measure. What distributive measures do you need to use in calculating Price.mean? Here Price.mean denotes the mean value of all prices.

Price.mean is an algebraic measure as it requires aggregation of data values to calculate. The distributive measures needed to calculate Price.mean are sum(prices) and count(prices). The formula to calculate the mean price can be expressed as: Price.mean = sum(prices) / count(prices). This calculation follows the distributive property of division over addition, which states that division can be distributed over a sum.

(2) (6 points) Explain whether Fruit.num_eq_min is an algebraic measure or holistic measure. Please justify your answer. Here Fruit.num_eq_min means the number of fruits that have the lowest price. For example, given 5 fruits with prices: 10, 11, 23, 10, 23, then the the lowest price is Price.min = 10 and Fruit.num eq min = 2.

Fruit.num eq min is a holistic measure as it is calculated based on the distribution of the data values, rather than aggregating the data values themselves. It involves counting the number of values in the data set that are equal to the minimum value. This calculation does not follow any distributive property of arithmetic operations and requires a holistic view of the data. The calculation of Fruit.num eq min requires knowledge of the entire data set and cannot be calculated based on aggregated values alone.

Problem 2. Data Cube Concepts (26 points)
Suppose the base cuboid of this data cube contains two cells:

$$(a_1, a_2, a_3, a_4, a_5, a_6, a_7) : 1, (a_1, b_2, a_3, b_4, a_5, b_6, a_7) : 1$$

where $a_i \neq b_i$ for $i = 2, 4, 6$. Assume each dimension contains no concept hierarchy.

(1) (6 points) Please list all the (nonempty) closed cells in this data cube.

3, Two base cells and (a1, *, a3, *, a5, *,a7).

(2) (4 points) How many (nonempty) aggregate cells are there in this data cube?

Since there are 4 common dimentions, the total number of nonempty aggregate cells is

$$2 * (2^7 - 1) - 2^4 = 238$$

(3) (4 points) How many (nonempty) aggregate closed cells are there in this data cube? Please list them.

1, (a1, *, a3, *, a5, *,a7).

(4) (4 points) If we set minimum support = 2, how many (nonempty) aggregate cells are there in the corresponding iceberg cube?

These two base cells have common value in 4 dimensions; therefore, there are $2^4 = 16$ nonempty cells with support = 2 and all of them are aggregated cells.

(5) (8 points) What are the differences among star schema, snowflake schema, and fact constellations for modeling the data warehouses? Which schema do you suggest to model this cube? Please justify your answer.

In a Star Schema, the data is arranged with a central fact table as the focus, which holds the information to be analyzed. This central fact table is connected to several dimension tables that contain descriptive data about the elements being analyzed. These connections are referred to as "star joins." The Star Schema is user-friendly and provides quick query results because of its simplified structure.

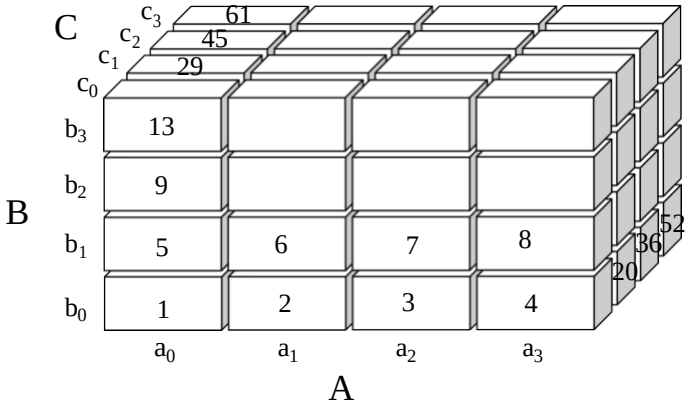
The Snowflake Schema is a variation of the Star Schema, in which the dimension tables have been structured in a way to minimize duplicate data. This leads to a more intricate schema, with several relationships between the fact table and dimension tables. While this complexity slows down query performance, it benefits the data by reducing storage requirements and improving data accuracy.

Fact Constellations are comparable to Star Schemas but with the added ability to have several fact tables within the same schema. Each fact table in this structure is designed to concentrate on a particular set of measures and dimensions. This offers greater adaptability in data modeling and analysis, however it may result in a more intricate and difficult to comprehend schema.

Based on the information about the cube provided, it is recommended to use a Star Schema to model it. The cube consists of only two base cells and does not have any hierarchical relationships between its dimensions, making the Star Schema the simplest and most straightforward option for organizing the data. Additionally, the Star Schema's denormalized structure will ensure fast query performance for the measures contained in the cube.

Problem 3. Data Cube Computation (20 points)

Assume our data is stored in a data cube with 3 dimensions A, B and C. We would like to do full cube computation using multi-way array aggregation. The lengths of dimensions A and B are 4000 and 400, respectively. The length of dimension C is an unknown value x ($x > 0$). We cut each dimension into quarters and get 64 chunks as follows.



(1) (4 points) If we follow the scan order $1-2-3-4-5-6-\cdots$, what is the memory requirement to compute the whole cube?

$$4000 * 400 + 4000 * \frac{x}{4} + 100 * \frac{x}{4} = 1.6 * 10^6 + 1025x$$

(2) (4 points) If we follow the scan order $1-5-9-13-2-6-\cdots$, what is the memory requirement to compute the whole cube?

$$400 * \frac{x}{4} + 1000 * \frac{x}{4} + 4000 * 400 = 1.6 * 10^6 + 350x$$

(3) (4 points) If we follow the scan order $1-17-33-49-5-21-\cdots$, what is the memory requirement to compute the whole cube?

$$1000 * x + 1000 * 100 + 400 * x = 1 * 10^5 + 1400x$$

(4) (4 points) What is the maximum **integer value** of *x* if (3) is the most memory-saving order **among (1), (2) and (3)**?

$$1 * 10^5 + 1400x < 1.6 * 10^6 + 350x$$

$$x = 1428$$

(5) (4 points) Besides the scan orders in (1)-(3), are there any other scan orders? If yes, please list all the other scan orders by specifying the first 10 chunks in their orders (you can assume the order always starts from chunk 1). If not, please justify your answer.

Yes, there are 3 other scane orders, they are:

1-2-3-4-17-18-19-20-…

1-17-33-49-2-18-34-50-…

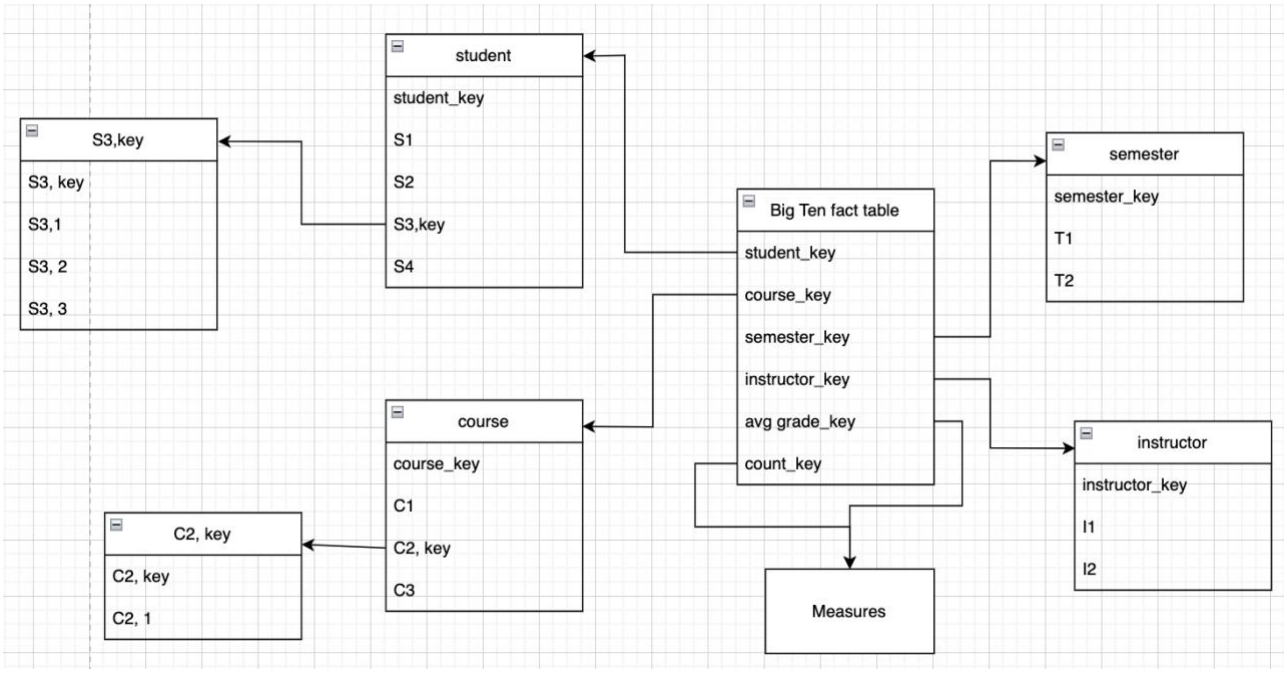
1-5-9-13-17-21-25-29-…

Problem 4. Snowflake Schema (18 points)

Suppose that a data warehouse for Big Ten universities consists of the four dimensions: *student*, *course*, *semester*, and *instructor*, and two measures: *avg grade* and *count*. Each dimension has a concept hierarchy, e.g., for *student*, the concept hierarchy is "*student* < *major* < *college* < *university* < *all*"; for *course*, the concept hierarchy is "*course* < *department* < *college* < *university* < *all* ", etc.

At the lowest conceptual level, i.e., for a given *student*, *course*, *semester*, *instructor* combination, the *avg grade* measure stores the actual course grade of the student. At higher levels of the concept hierarchy for one or more dimensions, *avg grade* stores the average grade for the given combination.

(1) (8 points) Assume that *student* has attributes S_1, S_2, $S_{3,key}$, S_4, where $S_{3,key}$ has attributes $S_{3,key}$, $S_{3,1}$, $S_{3,2}$, $S_{3,3}$; *course* has attributes C_1, $C_{2,key}$, C_3, where $C_{2,key}$ has attributes $C_{2,key}$, $C_{2,1}$; *semester* has attributes T_1, T_2, and *instructor* has attributes I_1, I_2. Draw a *snowflake schema* diagram for the data warehouse, where the dimension tables will be based on the attributes of the dimensions.



(2) (5 points) Starting with the base cuboid [*student, course, semester, instructor*], what specific OLAP operations (e.g., roll-up, drill down, etc.) should you perform in order to list the average grade of Computer Science courses (i.e., department = "Computer Science" for dimension *course*) for each student.

1. Rolling up semester and instructor, to reduce the dimensions, since these two attributes are not related to the requirement.
2. Rolling up course, and select the courses which provide by department of Computer Science.
3. Get the average grade for each student.

(3) (5 points) If each of the four dimensions in the data warehouse has five levels (including all), e.g., "*student* < *major* < *status* < *university* < *all* " for student, how many cuboids will the data cube contain (including the base and apex cuboids)? Clearly justify your answer. and we have the option of not having a dimension.

If each of the four dimensions in the data warehouse has five levels, the data cube will contain $5^4 = 625$ cuboids. Because there are only 5 possible choice for each 1 of 4 dimensions.

Problem 5. OLAP Operations (24 points)

In this problem, we explore some OLAP operations[1] with pandas[2], a famous data analysis tools for Python.

We analyze a dataset of student performance on Mathematics (file: student-mat.csv), which is a part of the Student Performance Data Set[3]. In the dataset, each record (row) is a student, each attribute (column) has different meaning with explanation in file student.txt.

(1) (4 points) Load the data. How many attributes does the dataset have? How many students (records) does the dataset have?
Hint: You need to set an appropriate delimiter sep when using pandas.read_csv().

```
df = pd.read_csv('/content/drive/MyDrive/412/a2/student-mat.csv',
delimiter=';')
#Question 1
num_attributes = len(df.columns)
num_records = len(df)
#display the number of attributes and number of records
print(num_attributes, num_records)
```

There are 33 attributes and 395 student records in the datatset.

(2) (4 points) What is the mean of all students' final grades (G3)? What is the median?
Hint: Here is a tutorial for calculating summary statistics.

```
#Question 2
# Mean of G3
mean_G3 = df['G3'].mean()
# Median of G3
median_G3 = df['G3'].median()
print(mean_G3, median_G3)
```

Mean of all students' final grades is 10.4152. Median of all students' final grade is 11.

(3) (8 points) Slicing and dicing.

(a) How many students are 16 years old? What is the mean of their final grade?

```
#Question 3 a
#Select the rows where the age is 16
df_16 = df[df['age'] == 16]
#Get the number of 16-year-old students
num_16 = len(df_16)
#Get the mean of the final grades of the 16-year-old students
mean_grade_16 = df_16['G3'].mean()
print(num_16, mean_grade_16)
```

There are 104 16 years old students, the mean of their final grade is 11.0288.

(b) How many students have age from 16 to 18 (both inclusive) and study time more than 5 hours a week simultaneously? What is the mean of their final grade?

```
#Question 3 b
#Select the rows where the age is between 16 and 18 (inclusive) and study
time is more than 5
df_selected = df[(df['age'] >= 16) & (df['age'] <= 18) & (df['studytime'] >=
3)]
#Get the number of students that meet the criteria
num_selected = len(df_selected)
#Get the mean of the final grades of the selected students
mean_grade_selected = df_selected['G3'].mean()
print(num_selected, mean_grade_selected)
```

There are 62 students' age are from 16 to 18 and study time more than 5 hours a week. The mean of their final grade is 11.7581.

(4) (8 points) Pivot table. The pivot table API in pandas can help us pivot and visualize two or more dimensions of the data cube. Draw a pivot table with two dimensions: study time and travel time, and show the mean of final grades. It is not necessary to convert the index (e.g., traveltime=1) to actual meaning (e.g., < 15 min). You can type the table or take a screenshot.
Hint: Check the pivot table API and specify values, index, columns, and aggfunc.

```
#Question 4
# Create the pivot table
table = df.pivot_table(values='G3', index='studytime', columns='traveltime',
aggfunc='mean')

# Print the pivot table
print(table)
```

traveltime	1	2	3	4
studytime				
1	10.866667	9.129032	8.090909	10.333333
2	10.413534	9.823529	10.181818	5.333333
3	11.382979	11.500000	12.000000	10.000000
4	11.705882	10.222222	NaN	13.000000

[1]pandas is not an OLAP system. We only use it to understand some operations on multidimensional data.

[2]https://pandas.pydata.org/docs/index.html

[3]The full dataset contains both Mathematics and Portuguese performance. We only use the Mathematics performance.