ECE4721J - Homework 4

Methods and Tools for Big Data

Kexuan Huang 518370910126

July 19, 2022

Ex.1 Reminders on database

1. Explain what is a Join operation, and describe its most common types.[1]

`JOIN` is an SQL clause used to query and access data from multiple tables, based on logical relationships between those tables Basically, we have 5 types of `JOIN`:

- `INNER JOIN`
- `LEFT OUTER JOIN`
- `RIGHT OUTER JOIN`
- `SELF JOIN`
- `CROSS JOIN`

2. What is an aggregate operation?[2]

An aggregation operation computes a single value from a collection of values. An example of an aggregation operation is calculating the average daily temperature from a month's worth of daily temperature values.

3. Write at least three advanced nested queries on the weather database.

For schema setup, please refer to `README.md`

[1] devart
[2] Microsoft Docs

3.1 Top 5 stations with highest daily average temperature

SQL:

```sql
SELECT station.s_name AS station, weather.w_value AS value
FROM weather
    INNER JOIN station ON station.s_id = weather.w_station
WHERE weather.w_type = 'TAVG'
    AND LENGTH(weather.w_value) > 0
ORDER BY CAST(weather.w_value AS INTEGER) DESC
LIMIT 5;
```

Output:

```
+--------------------------+-------+
|         station          | value |
+--------------------------+-------+
| ELK CREEK OREGON         | 572   |
| BEVERLY HILLS CALIFORNIA | 567   |
| BEVERLY HILLS CALIFORNIA | 544   |
| COLORADO CITY COLORADO   | 492   |
| ELK CREEK OREGON         | 466   |
+--------------------------+-------+
5 rows selected (3.581 seconds)
```

3.2 Top 5 station with lowest daily minimum temperature on Augest 25, 2017

SQL:

```sql
SELECT station.s_name AS station,
    weather.w_value AS value
FROM weather
    INNER JOIN station ON station.s_id = weather.w_station
WHERE weather.w_type = 'TMIN'
    AND LENGTH(weather.w_value) > 0
    AND weather.w_value <> -999
    AND weather.w_date = '20170825'
ORDER BY CAST(weather.w_value AS INTEGER)
LIMIT 5;
```

Output:

```
+----------------------------+-------+
|          station           | value |
+----------------------------+-------+
| VOSTOK                     | -750  |
| SAN ANTONIO INCARNATE WORD | -728  |
| PROGRESS                   | -362  |
| SYOWA                      | -329  |
| MIRNYJ                     | -324  |
+----------------------------+-------+
5 rows selected (3.691 seconds)
```

3.3 Top 5 date with highest average temperature in Shanghai

SQL:

```sql
SELECT country.c_name AS country,
    station.s_name AS station,
    weather.w_date AS day,
    weather.w_value AS value
FROM station
    INNER JOIN country ON SUBSTR(station.s_id, 1, 2) = country.
       c_fips
    INNER JOIN weather ON station.s_id = weather.w_station
WHERE station.s_name LIKE 'SHANGHAI%'
    AND weather.w_type = 'TAVG'
ORDER BY CAST(weather.w_value AS INTEGER) DESC
LIMIT 5;
```

Output:

```
+---------+-------------------+----------+-------+
| country |      station      |   day    | value |
+---------+-------------------+----------+-------+
| China   | SHANGHAI/HONGQIAO | 20170721 | 356   |
| China   | SHANGHAI/HONGQIAO | 20170724 | 354   |
| China   | SHANGHAI          | 20170724 | 353   |
| China   | SHANGHAI/HONGQIAO | 20170725 | 353   |
| China   | SHANGHAI/HONGQIAO | 20170720 | 351   |
+---------+-------------------+----------+-------+
5 rows selected (2.094 seconds)
```

Ex.2 Holidays!

1. Define what is “perfect weather” according to you.

Perfect weather options for me:

1. Average temperature: 15°C ~ 25°C
2. Maximum temperature: 30°C
3. Minimum temperature: 10°C
4. Precipitation: 10% ~ 20%

And I wanna go on July and Auguest!

2. Using Drill, with or without R, determine the perfect location of your next holidays.

SQL:

```
SELECT DISTINCT(country.c_name) AS country,
    country.c_continent AS continent
FROM station
    INNER JOIN country ON SUBSTR(station.s_id, 1, 2) = country.
       c_fips
    INNER JOIN weather ON station.s_id = weather.w_station
WHERE (
        weather.w_date > 20170701
        AND weather.w_date < 20170831
        AND (
            (
                weather.w_type = 'TAVG'
                AND CAST(weather.w_value AS FLOAT) > 150
                AND CAST(weather.w_value AS FLOAT) < 300
            )
            OR (
                weather.w_type = 'TMAX'
                AND CAST(weather.w_value AS FLOAT) < 30
            )
            OR (
                weather.w_type = 'TIN'
                AND CAST(weather.w_value AS FLOAT) > 10
            )
            OR (
                weather.w_type = 'PRCP'
                AND CAST(weather.w_value AS FLOAT) > 10
                AND CAST(weather.w_value AS FLOAT) < 20
            )
        )
    )
LIMIT 3;
```

Output:

```
+---------+-----------+
| country | continent |
+---------+-----------+
| Belize  | NA        |
| Fiji    | OC        |
| Japan   | AS        |
+---------+-----------+
3 rows selected (3.743 seconds)
```

Fiji looks good to me!

Ex.3 Data visualisation

1. Plot the temperature variation for each continent.

SQL:

```sql
CREATE TABLE dfs.tmp.variation AS
SELECT country.c_continent AS continent,
    SUBSTR(weather.w_date, 5, 2) AS month,
    AVG(CAST(weather.w_value AS FLOAT)) AS temperature
FROM weather
    INNER JOIN country
        ON SUBSTR(weather.w_station, 1, 2) = country.c_fips
GROUP BY country.c_continent, month
ORDER BY country.c_continent, month;
```

Output: a CSV file under `/tmp/variation/0_0_0.csv` in the following format:

```csv
continent,month,temperature
AF,01,197.2585854968666
AF,02,208.32279406108162
AF,03,219.8021733168792
AF,04,221.49289368959637
AF,05,222.44915687276443
AF,06,216.14217875115176
AF,07,213.0046727330218
AF,08,211.13295474100843
AF,09,218.6969151670951
AF,10,217.42438489819006
AF,11,205.95837657524092
AF,12,198.06032209791206
AN,01,-13.24591977869986
AN,02,-34.87149606299213
AN,03,-64.83542788749251
AN,04,-86.39857227840571
AN,05,-83.21605117766792
AN,06,-113.84666666666666
AN,07,-114.73870682019486
AN,08,-124.0599938781757
AN,09,-111.12093435836783
AN,10,-72.07099012543368
AN,11,-37.66158958737192
AN,12,-13.127147766323024
AS,01,83.21173870897132
AS,02,94.07081743554168
AS,03,124.55971918876755
...
```

Read the CSV data into RStudio and load the package ggplot for plotting the results:

```
require(ggplot2)
df <- read.csv("/data/variation.csv")
```

Plot the results:

```
ggplot(data = df, aes(x = factor(month), y = temperature / 10,
    color = continent)) +
     geom_line(aes(group = continent)) +
     geom_point() +
     xlab("Month") +
     ylab("Average Temperature (°C)")
```

Output:

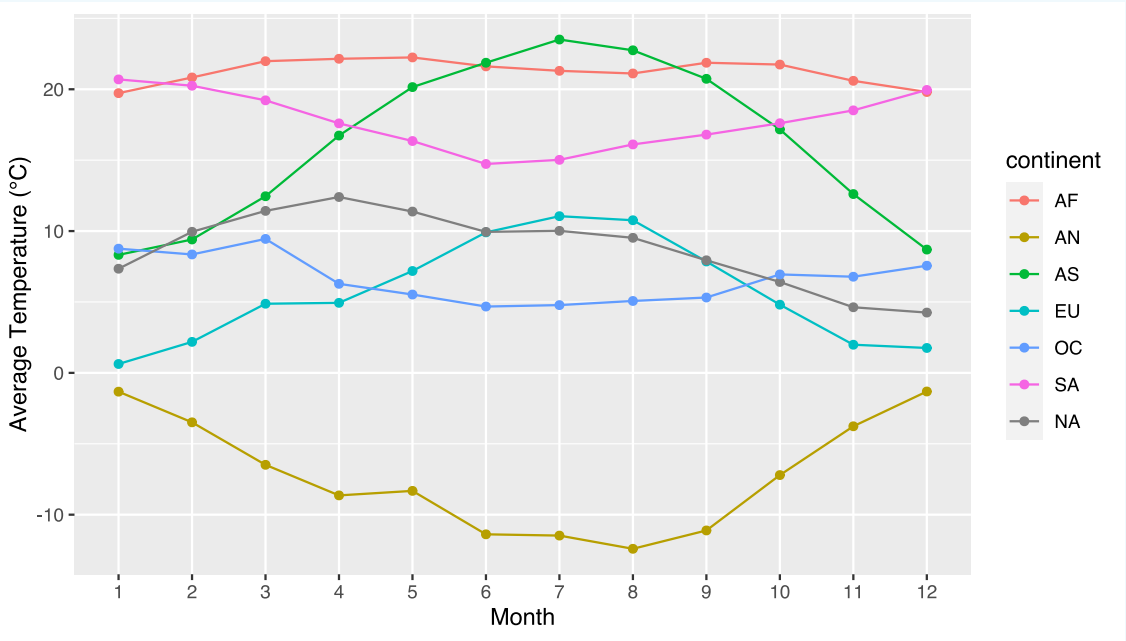

Figure 1: Temperature Variation

2. Plot the average temperature for each continent.

Plot the results:

```
ggplot(df, aes(continent, temperature / 10, fill = continent)) +
    geom_bar(position = "dodge", stat = "summary", fun = "mean") +
    xlab("Continent") +
    ylab("Average Temperature (°C)")
```

Output:

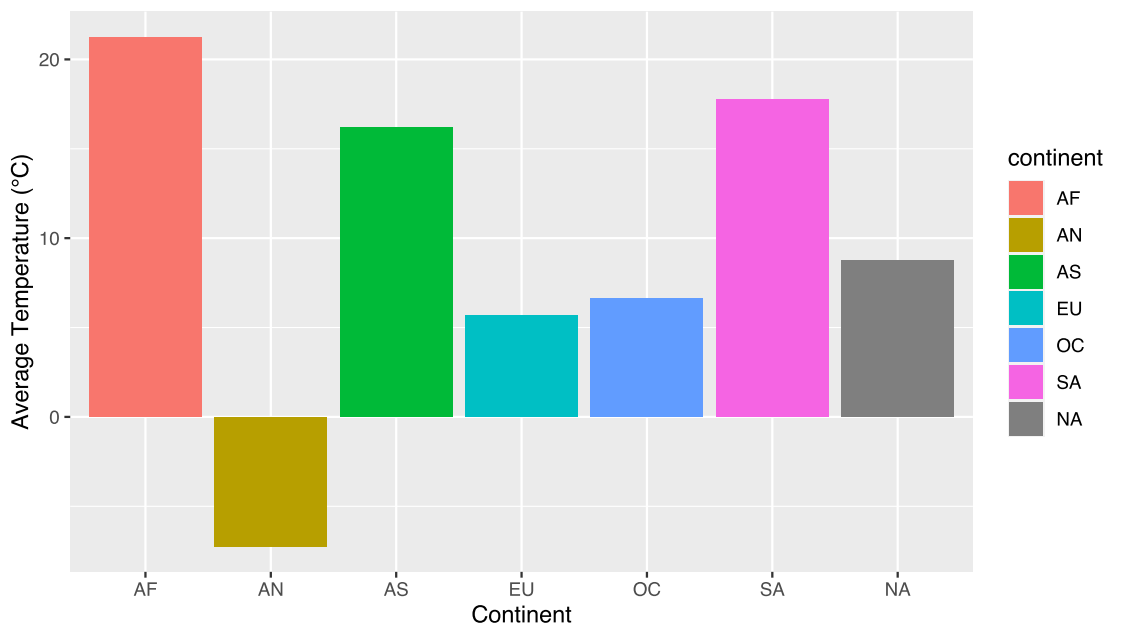

Figure 2: Average Temperature