
ECE4721J - Homework 4

Methods and Tools for Big Data

Kexuan Huang 518370910126

July 4, 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](#)

¹devart

²Microsoft Docs

3.1 Top 5 stations with highest daily average temperature

SQL:

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

Output:

```
1 +-----+
2 |          station          | value |
3 +-----+
4 | ELK CREEK OREGON          | 572   |
5 | BEVERLY HILLS CALIFORNIA  | 567   |
6 | BEVERLY HILLS CALIFORNIA  | 544   |
7 | COLORADO CITY COLORADO    | 492   |
8 | ELK CREEK OREGON          | 466   |
9 +-----+
10 5 rows selected (3.581 seconds)
```

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

SQL:

```
1 SELECT station.s_name AS station,
2     weather.w_value AS value
3 FROM weather
4     INNER JOIN station ON station.s_id = weather.w_station
5 WHERE weather.w_type = 'TMIN'
6     AND LENGTH(weather.w_value) > 0
7     AND weather.w_value <> -999
8     AND weather.w_date = '20170825'
9 ORDER BY CAST(weather.w_value AS INTEGER)
10 LIMIT 5;
```

Output:

```
1 +-----+-----+
2 |          station          | value |
3 +-----+-----+
4 | VOSTOK                    | -750  |
5 | SAN ANTONIO INCARNATE WORD | -728  |
6 | PROGRESS                   | -362  |
7 | SYOWA                      | -329  |
8 | MIRNYJ                     | -324  |
9 +-----+-----+
10 5 rows selected (3.691 seconds)
```

3.3 Top 5 date with highest average temperature in Shanghai

SQL:

```

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

Output:

```

1  +-----+-----+-----+-----+
2  | country | station | day | value |
3  +-----+-----+-----+-----+
4  | China  | SHANGHAI/HONGQIAO | 20170721 | 356 |
5  | China  | SHANGHAI/HONGQIAO | 20170724 | 354 |
6  | China  | SHANGHAI | 20170724 | 353 |
7  | China  | SHANGHAI/HONGQIAO | 20170725 | 353 |
8  | China  | SHANGHAI/HONGQIAO | 20170720 | 351 |
9  +-----+-----+-----+-----+
10 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 August!

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

SQL:

```
1 SELECT DISTINCT(country.c_name) AS country,
2     country.c_continent AS continent
3 FROM station
4     INNER JOIN country ON SUBSTR(station.s_id, 1, 2) = country.
5     c_fips
6     INNER JOIN weather ON station.s_id = weather.w_station
7 WHERE (
8     weather.w_date > 20170701
9     AND weather.w_date < 20170831
10    AND (
11        (
12            weather.w_type = 'TAVG'
13            AND CAST(weather.w_value AS FLOAT) > 150
14            AND CAST(weather.w_value AS FLOAT) < 300
15        )
16        OR (
17            weather.w_type = 'TMAX'
18            AND CAST(weather.w_value AS FLOAT) < 30
19        )
20        OR (
21            weather.w_type = 'TIN'
22            AND CAST(weather.w_value AS FLOAT) > 10
23        )
24        OR (
25            weather.w_type = 'PRCP'
26            AND CAST(weather.w_value AS FLOAT) > 10
27            AND CAST(weather.w_value AS FLOAT) < 20
28        )
29    )
30 LIMIT 3;
```

Output:

```
1 +-----+-----+
2 | country | continent |
3 +-----+-----+
4 | Belize  | NA        |
5 | Fiji    | OC        |
6 | Japan   | AS        |
7 +-----+-----+
8 3 rows selected (3.743 seconds)
```

Fiji looks good to me!

Ex.3 Data visualisation

1. Plot the temperature variation for each continent.

SQL:

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

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

```
1 continent,month,temperature
2 AF,01,197.2585854968666
3 AF,02,208.32279406108162
4 AF,03,219.8021733168792
5 AF,04,221.49289368959637
6 AF,05,222.44915687276443
7 AF,06,216.14217875115176
8 AF,07,213.0046727330218
9 AF,08,211.13295474100843
10 AF,09,218.6969151670951
11 AF,10,217.42438489819006
12 AF,11,205.95837657524092
13 AF,12,198.06032209791206
14 AN,01,-13.24591977869986
15 AN,02,-34.87149606299213
16 AN,03,-64.83542788749251
17 AN,04,-86.39857227840571
18 AN,05,-83.21605117766792
19 AN,06,-113.84666666666666
20 AN,07,-114.73870682019486
21 AN,08,-124.0599938781757
22 AN,09,-111.12093435836783
23 AN,10,-72.07099012543368
24 AN,11,-37.66158958737192
25 AN,12,-13.127147766323024
26 AS,01,83.21173870897132
27 AS,02,94.07081743554168
28 AS,03,124.55971918876755
29 ...
```


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

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

Plot the results:

```
1 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:

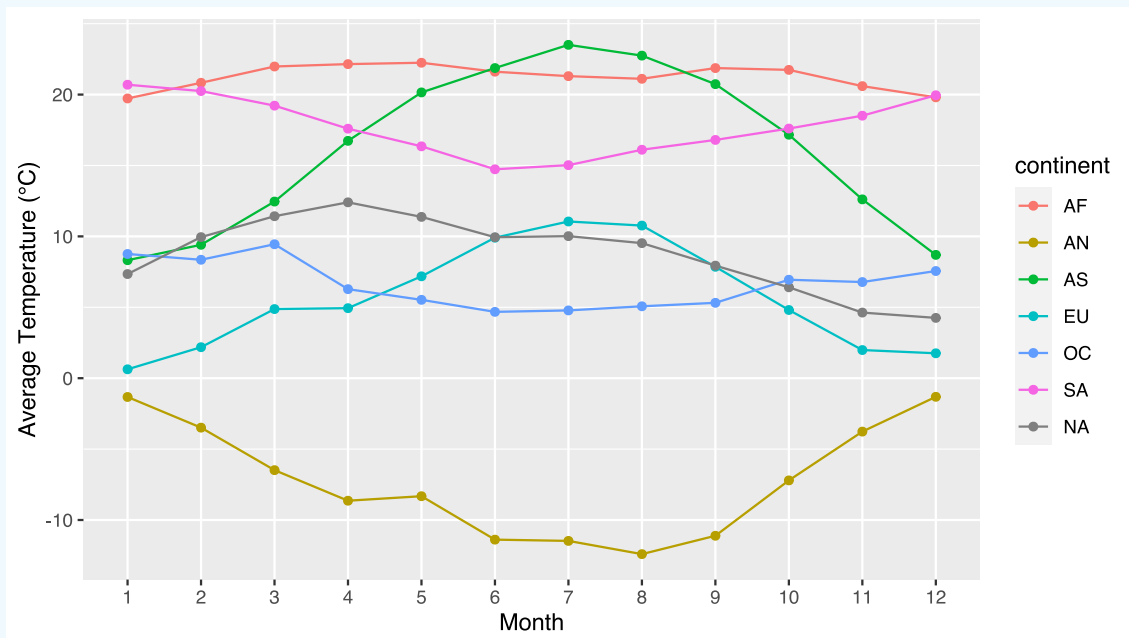


Figure 1: Temperature Variation

2. Plot the average temperature for each continent.

Plot the results:

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

Output:

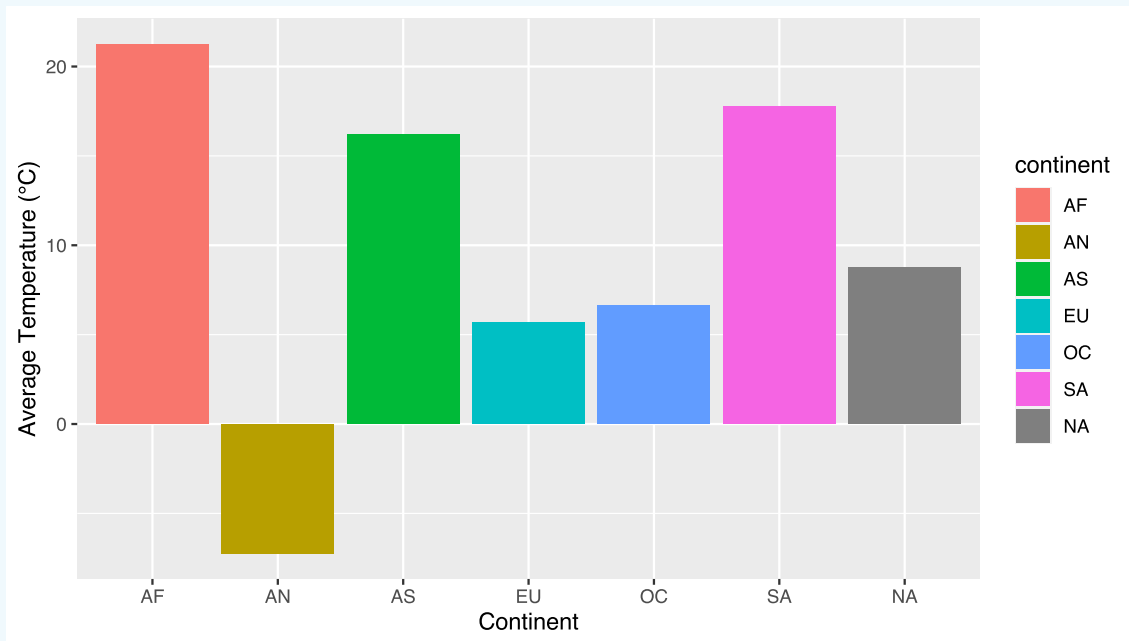


Figure 2: Average Temperature