

2024Olympics

October 21, 2024

0.1 0. Set up

Set up connection to DuckDB and load MySQL dataset into it.

```
[1]: import duckdb
import pandas as pd
```

```
[3]: %load_ext sql
conn = duckdb.connect()
%sql conn --alias duckdb
%sql ATTACH '<path>/olympics2024.db' as olympics
```

The sql extension is already loaded. To reload it, use:

```
%reload_ext sql
```

Running query in 'duckdb'

```
[3]: +-----+
| Success |
+-----+
+-----+
```

DuckDB's Python API has some limitation such as ATTACH command doesn't take effect. So database name has to be specified through variable in Magic SQL

```
[59]: db = 'olympics'
```

0.2 1. Sample DuckDB

List all medal types in alphabetical order.

```
[53]: %sql SELECT distinct(name) FROM {{db}}.medal_info ORDER BY name;
```

Running query in 'duckdb'

```
[53]: +-----+
|      name      |
+-----+
| Bronze Medal |
| Gold Medal  |
```

```
| Silver Medal |
+-----+
```

0.3 2. Successful Coaches

Find all successful coaches who have won at least one medal. List them in descending order by medal number, then by name alphabetically.

- Details: A medal is credited to a coach if it shares the same country and discipline with the coach, regardless of the gender or event. Consider to use `winner_code` of one medal to decide its country.
- Output Format: COACH_NAME|MEDAL_NUMBER

```
[54]: %%sql
SELECT
    C.name AS COACH_NAME,
    SUM(NUM) AS MEDAL_NUMBER
FROM
    ((
        SELECT M.discipline, T.country_code, COUNT(*) AS NUM FROM {{db}}.medals
        ↪M
        INNER JOIN (SELECT DISTINCT country_code, discipline, code FROM {{db}}.
        ↪teams) T ON winner_code = T.code
        GROUP BY M.discipline, T.country_code ORDER BY NUM DESC
    )
    UNION
    (
        SELECT M.discipline, A.country_code, COUNT(*) AS NUM FROM {{db}}.medals
        ↪M
        INNER JOIN {{db}}.athletes A ON winner_code = code
        GROUP BY M.discipline, A.country_code ORDER BY NUM DESC
    )) TMP
INNER JOIN {{db}}.coaches C ON C.discipline = TMP.discipline AND C.country_code
    ↪= TMP.country_code
GROUP BY C.name, C.discipline, C.country_code
ORDER BY MEDAL_NUMBER DESC, C.name ASC;
```

Running query in 'duckdb'

```
[54]: +-----+
|      COACH_NAME      | MEDAL_NUMBER |
+-----+
| BRECKENRIDGE Grant |      9      |
|   CAESAR Syque     |      9      |
|   CAREY Brian      |      9      |
|   CHEN Wei         |      9      |
| GAARENSTROOM Jordan |      9      |
|   GRABA Jess       |      9      |
```

HE Hua	9	
JIN Weiguo	9	
LANDI Cecile	9	
LANDI Laurent	9	

Truncated to displaylimit of 10.

0.4 3. Judo Athlete Medals

Find all athletes in Judo discipline, and also list the number of medals they have won. Sort output in descending order by medal number first, then by name alphabetically.

```
[55]: %%sql
SELECT
    ATHLETE_NAME,
    SUM(MEDAL_NUMBER) AS MEDAL_NUMBER
FROM
    ((
        SELECT ANY_VALUE(A.name) AS ATHLETE_NAME, COUNT(*) AS MEDAL_NUMBER FROM
        ↳{{db}}.athletes A
        INNER JOIN (
            SELECT T.team, M.winner_code, T.athletes_code FROM {{db}}.medals M
            INNER JOIN {{db}}.teams T ON winner_code = code AND T.discipline =
            ↳'Judo'
        ) TMP ON TMP.athletes_code = A.code
        GROUP BY TMP.athletes_code
    )
    UNION ALL
    (
        SELECT ANY_VALUE(A.name) AS ATHLETE_NAME, COUNT(*) AS MEDAL_NUMBER FROM
        ↳{{db}}.medals M
        INNER JOIN {{db}}.athletes A ON winner_code = code AND discipline =
        ↳'Judo'
        GROUP BY M.winner_code
    ))
GROUP BY ATHLETE_NAME
ORDER BY MEDAL_NUMBER DESC, ATHLETE_NAME ASC;
```

Running query in 'duckdb'

```
[55]: +-----+-----+
|    ATHLETE_NAME    | MEDAL_NUMBER |
+-----+-----+
|    ABE Hifumi      |            2 |
| AGBEGNENOU Clarisse |            2 |
|    BOUKLI Shirine  |            2 |
|    BUCHARD Amandine |            2 |
```

CYSIQUE Sarah Leonie	2	
DICKO Romane	2	
FUNAKUBO Haruka	2	
GABA Joan-Benjamin	2	
HASHIMOTO Soichi	2	
HUH Mimi	2	

+-----+

Truncated to displaylimit of 10.

0.5 4. Athletics Venue Athletes

For all venues that have hosted **Athletics** discipline competitions, list all athletes who have competed at these venues, and sort them by the distance from their nationality country to the country they represented in descending order, then by name alphabetically.

```
[56]: %%sql
SELECT
    A.name AS ATHLETE_NAME,
    A.country_code AS REPRESENTED_COUNTRY_CODE,
    A.nationality_code AS NATIONALITY_COUNTRY_CODE
FROM {{db}}.athletes A
INNER JOIN
    ((
        SELECT DISTINCT participant_code AS code FROM {{db}}.results
        WHERE
            venue IN (SELECT venue FROM {{db}}.venues WHERE disciplines LIKE
↳ '%Athletics%') AND
            participant_code IN (SELECT code FROM {{db}}.athletes)
    )
    UNION
    (
        SELECT DISTINCT athletes_code AS code FROM {{db}}.results
        INNER JOIN {{db}}.teams ON participant_code = code
        WHERE venue IN (SELECT venue FROM {{db}}.venues WHERE disciplines LIKE
↳ '%Athletics%')
    )) TMP
    ON TMP.code = A.code
INNER JOIN {{db}}.countries C
    ON A.country_code = C.code AND C.Latitude IS NOT NULL AND C.Longitude IS
↳ NOT NULL
INNER JOIN {{db}}.countries C2
    ON A.nationality_code = C2.code AND C2.Latitude IS NOT NULL AND C2.
↳ Longitude IS NOT NULL
ORDER BY sqrt((C.Latitude-C2.Latitude)**2 + (C.Longitude-C2.Longitude)**2)
↳ DESC, A.name ASC;
```

Running query in 'duckdb'

```
[56]: +-----+-----+-----+
|      ATHLETE_NAME      | REPRESENTED_COUNTRY_CODE | NATIONALITY_COUNTRY_CODE |
+-----+-----+-----+
|    GREEN Joseph      |          GUM             |          USA             |
| TUGADE-WATSON Regine |          GUM             |          USA             |
|    CABANG John       |          PHI             |          ESP             |
| IAKOPO Filomenaleonisa |        ASA              |          USA             |
|    HOWELL Davonte    |          CAY             |          GBR             |
|    JALLOW Sanu      |          GAM             |          USA             |
| BRATHWAITE Rikkoi    |          IVB             |          GBR             |
|    HODGE Adaejah     |          IVB             |          GBR             |
|    McMASTER Kyron   |          IVB             |          GBR             |
| PERINCHIEF Jah-Nhai  |          BER             |          GBR             |
+-----+-----+-----+
```

Truncated to displaylimit of 10.

0.6 5. Top 5 Rank Country Per Day

For each day, find the country with the highest number of appearances in the top 5 ranks (inclusive) of that day. For these countries, also list their population rank and GDP rank. Sort the output by date in ascending order.

```
[57]: %sql
WITH
  CCODE AS (
    (SELECT DISTINCT participant_code, country_code FROM {{db}}.results INNER
    ↪JOIN {{db}}.athletes ON participant_code = code)
    UNION
    (SELECT DISTINCT participant_code, country_code FROM {{db}}.results INNER
    ↪JOIN {{db}}.teams ON participant_code = code)
  ),
  TOP5 AS (
    SELECT * FROM
      (SELECT
        R.date DATE,
        CC.country_code COUNTRY_CODE,
        count(R.*) TOP5_APPEARANCES,
        row_number() OVER (PARTITION BY DATE ORDER BY TOP5_APPEARANCES
    ↪DESC, COUNTRY_CODE ASC) AS rownum
      FROM {{db}}.results R
      LEFT JOIN CCODE CC ON CC.participant_code = R.participant_code
      WHERE R.rank IS NOT NULL AND R.rank <= 5
      GROUP BY DATE, COUNTRY_CODE)
    WHERE rownum = 1
  ),
```

```

COUNTRYRANK AS (
  SELECT
    code,
    rank() OVER (ORDER BY "GDP ($ per capita)" DESC) GDP_RANK,
    rank() OVER (ORDER BY Population DESC) POPULATION_RANK
  FROM {{db}}.countries
)

SELECT DATE, COUNTRY_CODE, TOP5_APPEARANCES, GDP_RANK, POPULATION_RANK
FROM TOP5
LEFT JOIN COUNTRYRANK ON COUNTRY_CODE = code
ORDER BY DATE ASC;

```

Running query in 'duckdb'

```

[57]: +-----+-----+-----+-----+-----+
|  DATE  | COUNTRY_CODE | TOP5_APPEARANCES | GDP_RANK | POPULATION_RANK |
+-----+-----+-----+-----+-----+
| 2024-07-25 | KOR      | 7      | 38     | 22     |
| 2024-07-27 | USA      | 27     | 2      | 3      |
| 2024-07-28 | USA      | 24     | 2      | 3      |
| 2024-07-29 | USA      | 28     | 2      | 3      |
| 2024-07-30 | USA      | 34     | 2      | 3      |
| 2024-07-31 | GBR      | 24     | 18     | 20     |
| 2024-08-01 | USA      | 45     | 2      | 3      |
| 2024-08-02 | USA      | 47     | 2      | 3      |
| 2024-08-03 | USA      | 48     | 2      | 3      |
| 2024-08-04 | USA      | 40     | 2      | 3      |
+-----+-----+-----+-----+-----+

```

Truncated to displaylimit of 10.

0.7 6. Big Progress Country Female Teams

List the five countries with the greatest improvement in the number of gold medals compared to the Tokyo Olympics. For each of these five countries, list all their all-female teams. Sort the output first by the increased number of gold medals in descending order, then by country code alphabetically, and last by team code alphabetically.

- Details: When calculating all-female teams, if the `athlete_code` in a record from the `teams` table is not found in the `athletes` table, please ignore this record as if it doesn't exist.
- Hints: You might find Lateral Joins in DuckDB useful: find out the 5 countries with largest progress first, and then use lateral join to find their all-female reams.
- Output Format: COUNTRY_CODE|INCREASED_GOLD_MEDAL_NUMBER|TEAM_CODE

```

[58]: %%sql
WITH
  PARISGOLD AS (

```

```

SELECT country_code, sum(num) num FROM (
    (SELECT country_code, count(*) num FROM {{db}}.medals
    LEFT JOIN {{db}}.athletes T ON winner_code = T.code
    WHERE medal_code = 1 AND country_code IS NOT NULL
    GROUP BY country_code)
    UNION ALL
    (SELECT country_code, count(DISTINCT code) num FROM {{db}}.medals
    LEFT JOIN {{db}}.teams T ON winner_code = T.code
    WHERE medal_code = 1 AND country_code IS NOT NULL
    GROUP BY country_code))
GROUP BY country_code
),

TOKYOGOLD AS (
    SELECT country_code, gold_medal FROM {{db}}.tokyo_medals
),

ALLFEMALE AS (
    SELECT
        DISTINCT T.country_code,
        T.code,
        sum(CASE WHEN A.gender = 1 THEN 0 ELSE 1 END) num
    FROM {{db}}.teams T
    LEFT JOIN {{db}}.athletes A ON T.athletes_code = A.code
    WHERE T.athletes_code IN (SELECT code FROM {{db}}.athletes)
    GROUP BY T.country_code, T.code HAVING num = 0
),

RESULT AS (
    SELECT
        P.country_code COUNTRY_CODE,
        P.num-T.gold_medal INCREASED_GOLD_MEDAL_NUMBER,
        AF.code TEAM_CODE,
        rank_dense() OVER (ORDER BY P.num-T.gold_medal DESC) rank
    FROM PARISGOLD P
    LEFT JOIN TOKYOGOLD T ON P.country_code = T.country_code
    LEFT JOIN ALLFEMALE AF ON P.country_code = AF.country_code
    ORDER BY INCREASED_GOLD_MEDAL_NUMBER DESC
)

SELECT COUNTRY_CODE, INCREASED_GOLD_MEDAL_NUMBER, TEAM_CODE FROM RESULT
WHERE rank <= 5
ORDER BY INCREASED_GOLD_MEDAL_NUMBER DESC, COUNTRY_CODE ASC, TEAM_CODE ASC;

```

Running query in 'duckdb'

```
[58]: +-----+-----+-----+
| COUNTRY_CODE | INCREASED_GOLD_MEDAL_NUMBER | TEAM_CODE |
+-----+-----+-----+
| KOR          | 7                            | ARCWTEAM3---KOR01 |
| KOR          | 7                            | BDMWDOUBLES-KOR01 |
| KOR          | 7                            | BDMWDOUBLES-KOR02 |
| KOR          | 7                            | FENWTEAMEPEEKOR01 |
| KOR          | 7                            | FENWTEAMSABRKOR01 |
| KOR          | 7                            | GARWTEAM----KOR01 |
| KOR          | 7                            | HBLWTEAM7---KOR01 |
| KOR          | 7                            | SWAWTEAM2---KOR01 |
| KOR          | 7                            | TTEWTEAM----KOR01 |
| FRA          | 6                            | ARCWTEAM3---FRA01 |
+-----+-----+-----+
```

Truncated to displaylimit of 10.

```
[ ]:
```