

Задание семинара 7

Лим В. БПИ225

Сначала написал docker-compose.yml:

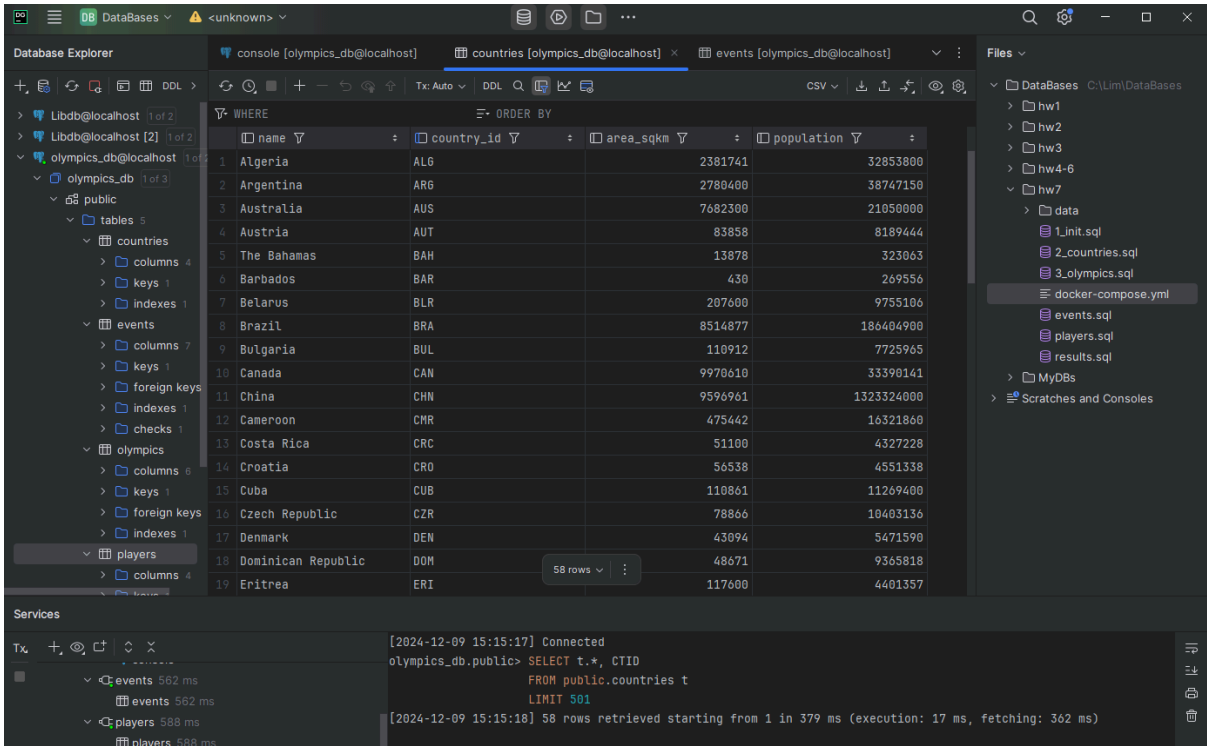
```
1  version: '3.8'
2
3  services:
4    postgres:
5      image: postgres:latest
6      environment:
7        POSTGRES_USER: user
8        POSTGRES_PASSWORD: password
9        POSTGRES_DB: olympics_db
10     ports:
11       - "5432:5432"
12     volumes:
13       - ./data:/var/lib/postgresql/data
14       - ./1_init.sql:/docker-entrypoint-initdb.d/1_init.sql
15       - ./2_countries.sql:/docker-entrypoint-initdb.d/2_countries.sql
16       - ./3_olympics.sql:/docker-entrypoint-initdb.d/3_olympics.sql
17       - ./events.sql:/docker-entrypoint-initdb.d/events.sql
18       - ./players.sql:/docker-entrypoint-initdb.d/players.sql
19       - ./results.sql:/docker-entrypoint-initdb.d/results.sql
20     networks:
21       - olympics_network
22
23  networks:
24    olympics_network:
25      driver: bridge
```

Затем поднял контейнер и подключился к бд:

```
C:\Lim\DataBases\hw7>docker-compose up -d
time="2024-12-09T15:12:45+03:00" level=warning msg="C:\\Lim\\DataBases\\hw7\\docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 15/15
  postgres Pulled                                         99.7s
  2cd360f3b7db Download complete                        0.6s
  627f580b7ad7 Download complete                       33.6s
  c5fdb20d8658 Download complete                       2.7s
  67c5fe618f0c Download complete                       2.2s
  002e1a8eb6f9 Download complete                       0.9s
  cfb3c2203f88 Download complete                      54.2s
  2cb801c39436 Download complete                       94.3s
  8f152c4aceed Download complete                       3.5s
  8d4265d09d9c Download complete                       2.2s
  e3a8293e92fd Download complete                       2.2s
  bc0965b23a04 Download complete                      66.0s
  a24f300391ed Download complete                      45.8s
  9e592465b243 Download complete                      30.2s
  c9cdd1fe82e4 Download complete                       3.5s
[+] Running 1/1
  Container hw7-postgres-1 Started                       1.0s

C:\Lim\DataBases\hw7>docker ps -all
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAME
a6e7330a20bb   postgres:latest "docker-entrypoint.s..." 13 seconds ago Up 12 seconds 0.0.0.0:5432->5432/tcp            hw7-postgres-1

C:\Lim\DataBases\hw7>
```



1. Для Олимпийских игр 2004 года сгенерируйте список (год рождения, количество игроков, количество золотых медалей), содержащий годы, в которые родились игроки, количество игроков, родившихся в каждый из этих лет, которые выиграли по крайней мере одну золотую медаль, и количество золотых медалей, завоеванных игроками, родившимися в этом году.

Запрос:

```
SELECT EXTRACT(YEAR FROM p.birthdate) AS birth_year,  
       COUNT(DISTINCT p.player_id) AS num_players,  
       SUM(CASE WHEN r.medal = 'GOLD' THEN 1 ELSE 0 END) AS gold_medals  
FROM players p  
JOIN results r ON p.player_id = r.player_id  
JOIN events e ON r.event_id = e.event_id  
JOIN olympics o ON e.olympic_id = o.olympic_id  
WHERE o.year = 2004  
GROUP BY birth_year  
ORDER BY birth_year;
```

Результат:

The screenshot shows a database client interface with a dark theme. The top panel displays the 'Database Explorer' on the left, showing a tree view of the database structure including 'foreign keys', 'indexes', 'checks', 'olympics', 'columns', 'keys', 'foreign keys', 'indexes', and 'players'. The main panel shows a SQL query in the 'console' tab, which is the same query as provided in the 'Запрос:' section. The bottom panel shows the 'Services' tab, which includes a 'Tx' section on the left and a 'Result 10' table on the right. The 'Tx' section shows a list of database services with their execution times. The 'Result 10' table displays the query results, with columns for 'birth_year', 'num_players', and 'gold_medals'.

	birth_year	num_players	gold_medals
1	1971	26	9
2	1972	15	3
3	1973	14	5
4	1974	22	9
5	1975	17	11
6	1976	12	17
7	1977	15	4
8	1978	29	15
9	1979	18	7
10	1980	18	12
11	1981	11	2
12	1982	8	2
13	1983	13	2
14	1984	17	7
15	1985	21	11

2. Перечислите все индивидуальные (не групповые) соревнования, в которых была ничья в счете, и два или более игрока выиграли золотую медаль.

Запрос:

```
SELECT e.event_id, e.name
FROM events e
JOIN results r ON e.event_id = r.event_id
WHERE e.is_team_event = 0
AND r.medal = 'GOLD'
GROUP BY e.event_id, e.name
HAVING COUNT(DISTINCT r.player_id) >= 2
      AND COUNT(DISTINCT r.result) = 1;
```

Результат:

The screenshot shows a database IDE interface. On the left, the 'Database Explorer' pane shows the 'olympics_db@localhost' database with tables 'events' and 'players'. The main editor displays a SQL query. The 'Services' pane at the bottom shows the execution of the query, with a table of results.

SQL Query:

```
SELECT e.event_id, e.name
FROM events e
JOIN results r ON e.event_id = r.event_id
WHERE e.is_team_event = 0
AND r.medal = 'GOLD'
GROUP BY e.event_id, e.name
HAVING COUNT(DISTINCT r.player_id) >= 2
      AND COUNT(DISTINCT r.result) = 1;
```

Results Table:

event_id	name
E108	50m Freestyle Men

3. Найдите всех игроков, которые выиграли хотя бы одну медаль (GOLD, SILVER и BRONZE) на одной Олимпиаде. (player-name, olympic-id).

Запрос:

```
SELECT p.name, o.olympic_id
FROM players p
JOIN results r ON p.player_id = r.player_id
JOIN events e ON r.event_id = e.event_id
JOIN olympics o ON e.olympic_id = o.olympic_id
WHERE r.medal IN ('GOLD', 'SILVER', 'BRONZE')
GROUP BY p.name, o.olympic_id;
```

Результат:

The screenshot shows a database console interface with a dark theme. The top panel displays a SQL query being executed. The query is as follows:

```
18 HAVING COUNT(DISTINCT r.player_id) >= 2
19 AND COUNT(DISTINCT r.result) = 1;
20
21 ✓ SELECT p.name, o.olympic_id
22 FROM players p
23 JOIN results r 1<->1..n: ON p.player_id = r.player_id
24 JOIN events e 1..n<->1: ON r.event_id = e.event_id
25 JOIN olympics o 1..n<->1: ON e.olympic_id = o.olympic_id
26 WHERE r.medal IN ('GOLD', 'SILVER', 'BRONZE')
27 GROUP BY p.name, o.olympic_id;
```

The bottom panel shows the results of the query in a table format. The table has two columns: 'name' and 'olympic_id'. The results are as follows:

	name	olympic_id
1	Stephanie Graf	SYD2000
2	Otylia Jedrzejczak	ATH2004
3	Erick Wainaina	SYD2000
4	Vladimir Andreyev	SYD2000
5	Nicola Pizzoni	SYD2000
6	Zersenay Tadesse	ATH2004
7	Irina Yatchenko	ATH2004
8	Lidia Simon	SYD2000
9	Frantz Kruger	SYD2000
10	Simone Cercato	ATH2004
11	Enefiok Udo-Obong	ATH2004
12	Yu Yang	ATH2004
13	Jai Taurima	SYD2000
14	Melissa Morrison	SYD2000
15	Klaas-Erik Zwering	ATH2004
16	Francis Obikwelu	ATH2004
17	Laure Manaudou	ATH2004
18	Michael Klim	SYD2000
19	Terrence Trammell	SYD2000
20	Alice Mills	ATH2004
21	Jane Saville	ATH2004

4. В какой стране был наибольший процент игроков (из перечисленных в наборе данных), чьи имена начинались с гласной?

Запрос:

```
SELECT p.country_id,  
       COUNT(CASE WHEN LOWER(SUBSTRING(p.name, 1, 1)) IN ('a', 'e', 'i', 'o', 'u') THEN  
1 END) * 100.0 / COUNT(p.player_id) AS percentage  
FROM players p  
GROUP BY p.country_id  
ORDER BY percentage DESC  
LIMIT 1;
```

Результат:

The screenshot shows a database IDE interface. On the left, the 'Database Explorer' pane shows a tree view of the 'olympics_db@localhost' database, with 'players' selected. The main editor shows a SQL query with line numbers 26 to 34. The query is the same as the one in the previous block. Below the editor, the 'Services' pane shows a 'console' window with a message 'console 407 ms'. On the right, the 'Output' pane shows 'Result 22' with a table of results.

country_id	percentage
1 FIN	100

5. Для Олимпийских игр 2000 года найдите 5 стран с минимальным соотношением количества групповых медалей к численности населения.

Запрос:

```
SELECT c.name,  
       c.population,  
       COUNT(r.medal) AS num_medals,  
       COUNT(r.medal) * 1.0 / c.population AS medal_per_capita  
FROM results r  
JOIN players p ON r.player_id = p.player_id  
JOIN events e ON r.event_id = e.event_id  
JOIN olympics o ON e.olympic_id = o.olympic_id  
JOIN countries c ON p.country_id = c.country_id  
WHERE o.year = 2000  
      AND e.is_team_event = 1  
GROUP BY c.name, c.population  
ORDER BY medal_per_capita ASC  
LIMIT 5;
```

Результат:

The screenshot shows a database management tool interface. The top panel displays a SQL query in a console window. The query is a SELECT statement that joins the results, players, events, olympics, and countries tables to find the top 5 countries with the lowest ratio of group medals to population for the 2000 Olympics. The bottom panel shows the results of the query in a table format.

Database: olympics_db@localhost

Query:

```
SELECT c.name,  
       c.population,  
       COUNT(r.medal) AS num_medals,  
       COUNT(r.medal) * 1.0 / c.population AS medal_per_capita  
FROM results r  
JOIN players p ON r.player_id = p.player_id  
JOIN events e ON r.event_id = e.event_id  
JOIN olympics o ON e.olympic_id = o.olympic_id  
JOIN countries c ON p.country_id = c.country_id  
WHERE o.year = 2000  
      AND e.is_team_event = 1 -- групповые медали  
GROUP BY c.name, c.population  
ORDER BY medal_per_capita ASC  
LIMIT 5;
```

Results:

name	population	num_medals	medal_per_capita
1 Russia	143201600	4	0.000000027932648797220143
2 Nigeria	131529700	4	0.000000030411382372194265
3 Japan	128084700	4	0.000000031229334963504618
4 Brazil	186404900	8	0.000000042917326744093101
5 Germany	82689210	8	0.000000096747810264483117