Project 2:

CMPE 321, Introduction to Database Systems, Spring 2024

Due: 15.04.2024, 23:59

1 Project Description

In this project we have provided you with a MySQL database file called VolleyDB.sql. The database is "almost" identical to the sample data from Project 1, with minor changes and new insertions. The database consists of 9 tables, whose information is provided in the next section. The primary keys are already provided for you.

Aside from the insertions/updates that are wanted in the queries, do not make any changes to the database as we will directly use the VolleyDB.sql file for checking.

You need to implement 20 different SQL queries and provide the SQL files. We will also provide you with the expected results (the tables with column names and contents) in .txt files, so please don't forget to check your results.

2 Tables

- **Player**
    - username VARCHAR(512),
    - password VARCHAR(512),
    - name VARCHAR(512),
    - surname VARCHAR(512),
    - date_of_birth VARCHAR(512),
    - height INT,
    - weight INT,
    - PRIMARY KEY (username)
- **PlayerPositions**
    - player_positions_id INT,
    - username VARCHAR(512),
    - position INT,

- – PRIMARY KEY (player_positions_id)
- **PlayerTeams**
  - – player_teams_id INT,
  - – username VARCHAR(512),
  - – team INT,
  - – PRIMARY KEY (player_teams_id)
- **Team**
  - – team_ID INT,
  - – team_name VARCHAR(512),
  - – coach_username VARCHAR(512),
  - – contract_start VARCHAR(512),
  - – contract_finish VARCHAR(512),
  - – channel_ID INT,
  - – channel_name VARCHAR(512), P
  - – PRIMARY KEY (team_ID)
- **Coach**
  - – username VARCHAR(512),
  - – password VARCHAR(512),
  - – name VARCHAR(512),
  - – surname VARCHAR(512),
  - – nationality VARCHAR(512),
  - – PRIMARY KEY (username)
- **Position**
  - – position_ID INT,
  - – position_name VARCHAR(512),
  - – PRIMARY KEY (position_ID)
- **Jury**
  - – username VARCHAR(512),
  - – password VARCHAR(512),
  - – name VARCHAR(512),
  - – surname VARCHAR(512),
  - – nationality VARCHAR(512),

- – PRIMARY KEY (username)
- **MatchSession**
  - – session_ID INT,
  - – team_ID INT,
  - – stadium_ID INT,
  - – stadium_name VARCHAR(512),
  - – stadium_country VARCHAR(512),
  - – time_slot INT,
  - – date VARCHAR(512),
  - – assigned_jury_username VARCHAR(512),
  - – rating DOUBLE,
  - – PRIMARY KEY (session_ID)
- **SessionSquads**
  - – squad_ID INT,
  - – session_ID INT,
  - – played_player_username VARCHAR(512),
  - – position_ID INT,
  - – PRIMARY KEY (squad_ID)

3 Queries

1. Find the number of players and display this as *player_count*. The required column names are respectively: *player_count*.

2. List the match sessions which are played before 2024 (exclusive). Display date in DD/MM/YYYY format. The required column names are respectively: *session_ID*, *assigned_jury_username*, *rating*, *date*. Sort the results by *date* in ascending order.

3. List all the fields of the match sessions with the minimum *rating*. Display Date in DD.MM.YYYY format. The required column names are respectively: *session_ID*, *team_ID*, *stadium_ID*, *stadium_name*, *stadium_country*, *time_slot*, *date*, *assigned_jury_username*, *rating*.

4. List *assigned_jury_username* and *stadium_name* of the match sessions with the maximum *rating*. The required column names are respectively: *assigned_jury_username*, *stadium_name*. Sort the results by *assigned_jury_username* in descending order.

5. Find the average rating of all match sessions. The required column names are respectively: *average_rating*.

6. List all teams together with the number of players in each *Team*. Notice that in the Team table, each team_id is unique. However, different ids may have the same team names as different agreements with coaches in the past were needed to be recorded. Note that there may be empty teams. The required column names are respectively: *team_name*, *coach_name*, *coach_surname*, *player_count*.

7. Insert a new position, whose *position_ID* is "5" and *position_name* is "Middle Hitter". No required column names. No required output.

8. Find the names and surnames of coaches who have directed more than or equal to 2 match sessions. The required column names are respectively: *name*, *surname*. Sort by *surname* in descending order.

9. Find the names and surnames of players who were born in the same year as "Ebrar Karakurt" and who are taller than her. The required column names are respectively: *name*, *surname*. Sort by *surname* in ascending order.

10. List all the coaches' names who directed a single team non-stop between 02.09.2024-31.12.2025 (inclusive) and that team has agreement with "Digiturk". The required column names are respectively: *name*, *surname*, *channel_name*, *contract_start*, *contract_finish*. Sort by *name* in ascending order.

11. List all match sessions that were not directed by Daniele Santarelli and were not played in UK. The required column names are respectively: *session_ID*, *name*, *surname*, *stadium_name*, *stadium_country*, *team_name*. Sort by *session_id* in ascending order.

12. For each year, find and list the coach who has directed the team with the highest average rating in that year. The required column names are respectively: *name*, *surname*, *year*, *average_rating*. Sort by *year* in ascending order.

13. For each stadium, find the coach who has directed the highest number of matches in that stadium. Please note that there can be multiple maximums. You can check the provided result table. The required column names are respectively: *stadium_name*, *name*, *surname*, *directed_count*.

14. List all the matches that were played after 2023 (meaning starting from 01.01.2024 inclusive), and have not been directed by "Ferhat Akbaş". The required column names are respectively: *session_ID*, *name*, *surname*. Sort by *session_id* in ascending order.

15. Find the jury who has rated the highest number of match sessions in the database. The required column names are respectively: *name*, *surname*, *rated_sessions*.

16. Find the average rating of match sessions for each coach, and list the coaches in descending order of their average ratings. The required column names are respectively: *name*, *surname*, *average_rating*. Sort by *name* in descending order.

17. For each team name, find the coach who has signed a contract that has the longest period with that team, and has never directed a match with a rating lower than 4.7 (4.7 is not acceptable) with ANY team. Please note that if there are no coaches who meet these conditions, show the coach name, surname, and day count as NULL or None. The required column names are respectively: *team_name*, *name*, *surname*, *day_count*.

18. Find the names of coaches who have directed at least one match in each stadium. The required column names are respectively: *name*, *surname*, *played_count*.

19. Return the players who have played in *stadium_name* "GD Voleybol Arena" as *position_name* "Libero". The required column names are respectively: *name*, *surname*.

20. List all *player_ID*'s with the column *more_than_one* which is either TRUE if that player plays more than one position or FALSE otherwise. The required column names are respectively: *name*, *surname*, *more_than_one*. Sort in ascending order by *name*.

   The output of each query is given in the **output_query-index.txt** files (e.g. *output1.txt, output2.txt, ... , output20.txt*.

4 Submission

- This project can be implemented either individually or as a team of two people. You are free to change teams in upcoming projects.

- Write each query to a file named query**i**.sql (e.g. query1.sql,...,query20.sql).

- Include a comment in each file that explains the reasoning that led you to the query in detail.

- The .sql files that do not contain explanatory comments will receive 0 points.

- Put all .sql files into a folder.

- Name the folder as **StudentID1_StudentID2** if you are working as a group.

- Name the folder as **StudentID1** if you are working as an individual.

- Zip the folder for submission and name the .zip file with the same name given to the folder.

- Submit .zip file through **Moodle** until the deadline.

- Each group should submit **one** .zip file.

- Do not include any other files in your .zip file.

- **Any other submission method is not allowed.**

- 10 points will be deducted in case of non-compliance to any of the naming and folder conventions explained above.

- Do not inject your observations into your queries to skip some essential steps! Do not obtain values manually from the database and inject them into the queries! This kind of query will receive 0 points. Furthermore, additional penalties will be imposed as a violation of this condition will be considered dishonest behavior.

- The files must follow the specific naming procedure:
  - Student1ID_Student2ID.zip
    * Student1ID_Student2ID
      (This is a folder)
      · query1.sql
      · query2.sql
      · ...
      · query20.sql

- No points will be deducted because of floating point precision or empty/null lines.

5 Late Submission Policy

We will accept late submissions, however;

- One day late (even one minute will be considered a day late) would mean -30 points penalty.

- Two days late (even one minute and a day will be considered two days late) would mean -60 points penalty.

- Moodle will close after two days. No other submission method will be accepted.

6 Academic Honesty

Please read carefully the academic honesty part of the syllabus as we give utmost importance to academic honesty.