

Project: OlympicDB — Phase 3 —

Release Date: Apr. 04, 2024

Due: 8:00 PM, Apr. 12, 2024

Purpose of the project

The primary goal of this project is to realize a database system that supports the operations of OlympicDB, a database system for managing the teams, team members, awards, etc. of the Olympic Games. This project has you follow the typical development cycle of a database application (i.e., conceptual design, logical design, and implementation of an application).

The secondary goal is to learn how to work as a member of a team which designs and develops a relatively large, real database application.

Project objectives

The OlympicDB application will be developed in three phases: (1) The OlympicDB database design; (2) The OlympicDB data manipulation; (3) The OlympicDB analytical queries; and (4) The OlympicDB Java App.

You will implement your application program using PostgreSQL, Java and JDBC. The assignment focuses on the database component and not on the user interface (UI). Hence, NO HTML or other graphical user interface is required for this project and carry no bonus points.

It is expected that all members of a team will be involved in all aspects of the project development and contribute equally. Division of labor to data engineering (db component) and software engineering (java component) is not acceptable since each member of the team will be evaluated on both components.

Analytical Queries for OlympicDB

In Phase 2, you defined the relational schema of the OlympicDB database using PostgreSQL and familiarized yourself with all the powerful features of SQL/PL, by implementing various data manipulation tasks using views, common table expressions, triggers, stored procedures, and functions.

In this phase, you will extend your OlympicDB system from Phase 2 by implementing the following operations for handling analytical queries.

1. **countryRankings**

Rank all of the countries (country_code, country_name, participationRank, olympiadCount) in OlympicsDB based on the number of Olympiads that the country has participated in. *Hint: Consider using a view.*

2. **mostSuccessfulParticipantsInOlympiad**

Given an *olympiad_num* and integer *k*, display the top-*k* athletes (participantID, totalPoints) based on the points associated with each medal that the athlete earned. If the *olympiad_num* is set to 'ALL', the operation should list the top-*k* athletes across all Olympiads stored in OlympicDB. Otherwise, the operation should only consider athletes who participated in the provided Olympiad when displaying the top-*k* athletes. For both cases, athletes should be sorted in descending order by the total number of points earned.

3. topSports

Display the top- k sports (sport_id, sport_name, teamCount) with respect to the number of teams who participated in an event for the sport in the past x years. Integers x and k are input parameters to this function. Note that 1 year is defined as 365 days counting back from the current system clock.

4. connectedCoaches

Given two coach_ids, c_1 and c_2 , find a path, if one exists, between c_1 and c_2 with at most 2 hops between them. A hop is defined as two coaches having coached at the same Olympiad (olympiad_num). The output of this task should be a string representing the path that connects c_1 to c_2 with all intermediate hops. For example, ' $c_1 \rightarrow c_3 \rightarrow c_2$ ' where c_3 is a coach that connects c_1 to c_2 .

Project Submission

Your submission should contain SQL components for the analytical queries. Specifically,

- **analytical.sql** the script containing the implementation for the analytical queries.

The project will be collected by submitting your GitHub repository to Gradescope. Given that Phase 3 builds on top of the OlympicsDB system from Phase 2, you should utilize the same GitHub repository from Phase 2.

To turn in your code, you must do three things by the deadline:

1. Make a commit to your project repository that represents what should be graded as your group's submission for that phase.
2. Push that commit to the GitHub repository that you have shared with the TAs
3. Submit your GitHub repository (including all necessary SQL files) to Gradescope under the Project Phase 3 assignment.

To submit to Gradescope, you will need to:

- Select the appropriate assignment submission link (as you've previously done with homework assignments).
- On the "Submit Programming Assignment" window that appears, choose "GitHub." If this is your first time submitting to Gradescope via GitHub, you will be prompted to link your GitHub account and authenticate.
- Select your team's GitHub repository in the dropdown (searching will filter the repositories listed).
- Select the branch of your GitHub repository with the code your team wishes to submit (typically just the main branch). Then click the green upload button in the bottom left of the window.
- After uploading your team's submission, you will be taken to the submission results page. The next step is to add each team member to the assignment to allow for a single linked submission. Note: **There should only be one submission per team, i.e., every team member does not need to submit.**
- To link team members, click "Add Group Member" in the top right corner of this page.
- On the new window, add all of your corresponding team members to the group, then hit the green "Save" button in the bottom left of this window.

- If done correctly, every team member should receive a confirmation email from Gradescope.

Multiple submissions are allowed for each part for each team. The last submission before the corresponding deadline will be graded. *NO late submission is allowed.*

Grading

The project will be graded on correctness (e.g., coping with violation of integrity constraints), robustness (e.g., coping with failed transactions) and readability. You will not be graded on efficient code with respect to speed although bad programming will certainly lead to incorrect programs. Programs that fail to compile or run or connect to the database server earn **zero** and *no partial points*.

Academic Honesty

The work in this assignment is to be done *independently* by each team. Discussions with other students or teams on the project should be limited to understanding the statement of the problem. Cheating in any way, including giving your work to someone else will result in an F for the course and a report to the appropriate University authority.

Enjoy the second phase of your class project!