Assignment 4

Continuing on our publication world, we have the following 5 tables in the database. In this assignment, you will use those tables to create views, stored procedures, and triggers to analyze author publications and university rankings in various areas of computer science. The tables already exist in the sample220P.db which you will use to test your queries against. Please note that the sample file above is only a small portion of the real data. You will use the sample to try out your queries. The answers to the sample queries are provided in autograder. Please note that we will execute your queries on the full (hidden) database to test their correctness. The sample database is part of the full database.

Submission Python file:
https://drive.google.com/file/d/1yh2FL7mTUSXyC9SCTaz9IP3MhABEWweh/view?usp=sharing
Autograder(You only need to upload your submission python file and database file this time):
https://drive.google.com/file/d/1eRSwABqduiALeRJk7KHowLMclkRhuXyk/view?usp=sharing
Documentation to write Stored Procedure in Python:
https://docs.google.com/document/d/1Q3w8HVFgFh7mc0tcEY-d_QM9ymNZnae9aIe_IJOnxD0/edit?usp=sharing
Sample Database file **DO NOT MODIFY DATABASE**:
https://drive.google.com/file/d/1eolUddnUpH4ofZubckfBrJPwb3vyX3fk/view?usp=sharing

Author Information Table:

This table contains the information of an author with his/her name, scholar, affiliation(university), homepage link, whether Turing award winner, whether is an ACM fellow, region of the affiliation, and country of the affiliation.

author('name', 'scholarid', 'affiliation', 'homepage', 'turing_award', 'acm_fellow', 'region', 'country')

Sample Data:

Xiyuan Chen ISK129sT Purdue University NULL 0 0 america us

Michael J Carey 7ahTS81 University of California, Irvine NULL 0 1 america us

Note that in the table above, scholarid is the key. Thus, we are only storing information about the last affiliation of the scholar since each scholar has only 1 row in the table.
Also, in the table, scholar names are also unique -- that is, no two scholars have the same name. In general that would not be true, but we have cleaned the data to make things simpler.

Publication Records Table:

This table contains the summary of the publication records of each scholar in each year. The attributes are the name of the scholar, the conference abbreviation, the year of that conference, and the count of papers that the scholar published in that conference edition.

pub_info('name', 'conference', 'year', 'count')

Sample:

Sharad Mehrotra SIGMOD 2 2024

Mike Carey PODS 1 1992

Xiyuan Chen ICDE 2 2006

Note that in this table, authors are referred to using their names and not their scholarid. This, however, does not cause ambiguity since remember that the scholar names are assumed to be unique. Attribute ‘conference’ is the conference abbreviation.

Field to Conference Table:

This table contains the major, which is ‘Computer Science’ in all rows for simplicity, the field under that major, and abbreviation of conference under that field.

field_conference('major', 'field', 'conference')

Sample:

Computer Science Databases PODS

Computer Science Design Automation DAC

Note: Attribute ‘conference’ is the conference abbreviation.

Conference Ranking Table:

This table contains the ranking of each conference. It includes the abbreviation of conference, the name of conference, and the rank (A*, A, B, C)

conference_ranking('conf_abbr', 'name', 'rank')

Sample:

ICIS International Conference on Interaction Sciences: Information Technology A* ACIS
Australasian Conference on Information Systems A

University Ranking Table:

This table contains the ranking of each US university. It includes the name and alias name of the university, its state, city, zip code, its type(institutional_control), the rank(number), whether there is a tied rank, its acceptance rate, annual tuition, and number of enrollment.

usnews_university_rankings('university_name', 'alias_names', 'state', 'city', 'zip', 'institutional_control', 'rank', 'tied', 'acceptance_rate', 'tuition', 'enrollment')

Sample:

Princeton University NJ Princeton 8544 private 1 FALSE 6 59710 5604

Now, please answer the following questions using SQL statements.

Question 1: Views

1. Create a View for University Publication Profiles

- Create a view named Universities_Pubs_Profile with the following attributes:

  - university_name

  - year

  - num_authors (number of authors associated with the university who make publication that year)

  - total_pubs (total number of publications by those authors that year)

  - avg_pubs (average number of publications by those authors that year)

  - num_Astar_authors (total number of authors associated with the university who published A* papers that year)

  - total_Astar_pubs (total number of A* publications by those authors that year)

  - avg_Astar_pubs (average number of A* publications by authors who published A* papers that year)

  Note: ** The average value will be rounded**

2. Create a View for Author Profiles

- Create a view named Author_Profile with the following attributes:

  - author_name

- year
- affiliated_university
- num_papers (number of papers published by the author that year)
- num_Astar_papers (number of A* papers published by the author that year)

Question 2: Stored Procedure for Top Authors

Create a Stored Procedure to List Top Authors

- Create a stored procedure named Get_Top_Authors that takes the following input parameters:
  - start_year (e.g., 2001)
  - end_year (e.g., 2004)
  - criteria (all papers / A* papers)
  - area (Databases / Machine learning /…/ ALL)
  - limit (e.g., 100)
- The procedure should print a list of top authors based on the criteria, ordered by the number of conference that the author successfully published during the specified period.
- Example: CALL Get_Top_Authors(2001, 2004, 'A*', 'Databases', 100) will produce a list of 100 top authors based on the how many times the author published in an A* conference in the Databases area.
- The defined procedure header should be question2(...)

Question 3: Stored Procedure for Top Universities

Create a Stored Procedure to Rank Universities

- Create a stored procedure named Get_Top_Universities that takes the following input parameters:
  - start_year (e.g., 2001)
  - end_year (e.g., 2004)
  - w_1 (weight for total A* papers)
  - w_2 (weight for number of A* authors)
  - w_3 (weight for total number of papers)

- w_4 (weight for total number of authors)

- area (Databases / Machine learning /…/ ALL)

- limit (e.g., 10)

- The procedure should output the top universities based on the weighted sum of each criterion for the specified date range.

- Example: CALL Get_Top_Universities(2001, 2004, 0.4, 0.4, 0.1, 0.1, 'ALL', 10) will produce a ranking of the top 10 universities based on the given weights.

- The defined procedure header should be question3(...)

Question 4: Insert Results into Rating Tables

Insert Query Results into New Tables

- Write a script to insert the results of the top university ranking into three new tables named:

  - Astar_Based_Rating

  - Balanced_Rating

  - General_Rating

- Each table should be based on a range from 2014 to 2024 for all areas with a limit of 50. Use the following weights for each table:

  - Astar_Based_Rating: Weights = (0.4, 0.4, 0.1, 0.1)

  - Balanced_Rating: Weights = (0.25, 0.25, 0.25, 0.25)

  - General_Rating: Weights = (0.1, 0.1, 0.4, 0.4)

- You should answer this question in 6 steps:

1. `CREATE TABLE IF NOT EXISTS Astar_Based_Rating (...`
2. `CREATE TABLE IF NOT EXISTS Balanced_Rating (...`
3. `CREATE TABLE IF NOT EXISTS General_Rating (...`
4. `INSERT INTO Astar_Based_Rating...`
5. `INSERT INTO Balanced_Rating...`
6. `INSERT INTO General_Rating...`

Question 5: Create a Trigger for Publication Updates

Create a Trigger for Publication Insertion:

- Trigger Name: Recalculate_Rankings_After_Insert

- Create a trigger on the pub_info table that, upon insertion of a new publication, checks if it affects the ranking of the universities in the Astar_Based_Rating, Balanced_Rating, or General_Rating tables.

- If the ranking changes, the trigger should recompute and update the corresponding university ratings.

Submission

Write your SQL statements for the above tasks in the given python file. Make sure to test your scripts thoroughly before submission. Include comments to explain your code where necessary.