TECHNICAL UNIVERISTY OF CRETE
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

LABORATORY OF DISTRIBUTED
INFORMATION SYSTEMS AND APPLICATIONS

DATABASES

SECOND PHASE OF LABORATORY PROJECT – SPRING SEMESTER 2021-2022

The second phase of the laboratory work includes the following tasks:

1. Java application communication with PostgreSQL database management system via JDBC.

To implement the application you will use the Eclipse environment and the JDBC driver provided with the postgreSQL server..

You will need to implement a class that provides methods for the following:

1. Connection/Disconnection to/from a specific database on a postgreSQL server (IP address, database name, username, password).
2. Display of the grades on a student's lab assignments for a semester. The student's code (amka) and the semester (academic_year, academic_season) will be given as input arguments. Calling the method should result in viewing the student's data and a list containing the course code, assignment title and grade.
3. Import random lab assignments and work groups for a specific semester course. This method will accept as input arguments the course code (course_code), the serial execution (serial_number), the number of lab assignments (LabModule) and the number of work groups to be formed. When entering data you will take the following into account:
   i. The determination of the task type will be done randomly with a probability of 0.2 for project type and 0.8 for lab_exercise type.
   ii. The maximum number of participants per group for each task will be chosen randomly from 2 to 6.
   iii. The evaluation grade will be selected following a normal distribution on a scale of 0-10, with a mean of 6 and a standard deviation of 2.

   Hint: To generate random number values following the above, you will make use of the methods of the java.util.Random class: nextDouble (i), nextInt (ii) and nextGaussian (iii).
4. Insertion of random lab assignments and workgroups for all current semester lab courses (CourseRun) of a particular subject. This method will take as input arguments the subject (PLH, THL, etc.), the number of lab assignments (LabModule) and the number of workgroups to be formed for each course run. (**Note**: You should use the method you implemented in question 3)

2. Query performance analysis - physical design:

Study the following query: Find the full names of students with an entry_date after 1/9/2010 and before 1/9/2011 who have participated in teams for a project-type laboratory project with a grade greater than 8. Follow the steps below:

- **Step 1**: Use the database backup file that accompanies the essay and which can be downloaded from the link http://147.27.41.88:5000/sharing/NJ75GuyeN. This database already contains more than 53,000 tuples in the Person and Student tables, about 1.5 million tuples in the Joins table, more than 370,000 tuples in the Workgroup table, and about 1,800 tuples in the LabModule table..
- **Step 2**: Study the plan and execution times using EXPLAIN ANALYSIS and note your results and observations in your report. Then sequentially try to create any appropriate index(es) that you think may speed up the execution of the query and study the execution plan again.

What index(es) do you choose, what type and why? Test and note in your report the differences between different types of indexes as derived from the results provided by EXPLAIN ANALYSIS. What do you observe? Which index(es) do you think is more appropriate? Also try to possibly speed up the query further by using the clustering feature. What do you observe? Note all your observations in your report. In addition, try to selectively disable link computation algorithms (if used) and record in your report your observations regarding the change in execution plans and their performance. Do this both before and after you decide to create indexes. Finally, try, both before and after creating indexes, to change the order of the links (selectivities) and report the changes you observe in the execution plans and times.

- **Step 3**: Use the implementation of request (1.4) to increase the database data set by hundreds of thousands by entering random lab assignments and workgroups for all semester-long lab courses (CourseRun) of the subjects HRY, SYS, ENE. **Delete any indexes you created in Step (2) before creating the new student tuples**. After you have created them, re-examine the execution plans before and after creating the indexes following the previous steps and note for each step the observations justifying your choices in your report.

Notes:

- Before using EXPLAIN or EXPLAIN ANALYSE, disable the ability to create parallel execution plans using the command:

SET max_parallel_workers_per_gather = 0;

- Run EXPLAIN ANALYSIS a few times before deciding on the execution time of a query as the times change each time. In your report put the detailed results of the EXPLAIN ANALYSIS for at least three (3) runs of each query.

Deliverable

You will deliver a deliverable in zip format that includes:

1. A Backup of the Phase B Database.

2. The code for as many of the requests with (*) of Phase A as you delivered with Phase B as well as a brief description in which you indicate the correspondence of functions, etc. that you implemented with the requests.

3. The Eclipse project of the external application (export to .zip) that you implemented in the request 1 of phase B.

4. A detailed implementation report in pdf format for Phase B question 2 in which you justify the indexes you created and all other actions you took and the results of using the EXPLAIN ANALYSIS command and the difference in execution times you observed by following the instructions in the pronunciation.

Deadline: 30/05/2022