TECHNICAL UNIVERSITY OF CRETE
ECE SCHOOL

DISTRIBUTED INFORMATION SYSTEMS AND APPLICATIONS WORKSHOP

DATABASES - PLI302 PHASE I
LABORATORY WORK – SPRING SEMESTER 2018-2019

TEACHER: Deligiannakis Antonios
LABORATORY **SUPPORT : Nektarios Moumoutzis, Nikolaos Pappas**

I. General description

The implementation of the database of a polytechnic department is requested. The database will contain all the data related to the department and its operation according to the following specifications. For this database you are given the ER diagram, the implementation in a relational PostgreSQL database and initial data. The relational database will be given to you in a backup file which you can restore in a PostgreSQL system that you will install on your personal computer. This database contains data on the fields, laboratories, courses taught in the department and random registrations for professors, laboratory staff and students.

Appropriate personal data should be maintained for all members of the department (faculty, laboratory staff and students). This data includes: AMKA which is unique for each person, first name, last name, patronymic, e-mail address. In particular, for students there should be information about the no. registry, date of registration. The professors belong to ranks (full-time, deputy, assistant, lecturer). The laboratory staff is also divided into grades (A, B, C, D). The department is organized research into specific areas (characterized by code, title and description). Workshops have also been set up in the department to support the educational process. Each laboratory belongs to a single field, has a unique code, specific title, description and is staffed by professors and laboratory staff. Each workshop is led by a professor who must necessarily belong to the highest level and covers one or more cognitive areas which are recorded in the form of three-letter codes (eg PLI, EKP, HRY...).

All courses are semester-long. For each course there is a unique code and information related to the title, description, the teaching units, the weekly hours of teaching, tutoring support and preparation of workshops and they have an indicative semester of execution (typical study year - winter, spring / typical year, typical season) . In each semester of the academic year, it is decided which courses will be taught. For each six-semester course, one or at most two teaching professors are designated and in the case that the course is a laboratory, the laboratory in which the assignments will be prepared and the laboratory staff to support them must be designated beforehand.

Grading rules

For each semester course, the grading rules are defined from which the final grade of each student is derived. The rules include:

1. **The participation percentage of the written exam in the final score. If the course does not is laboratory, participation rate is 100%.**
2. **If the course is laboratory and the student is required to have a laboratory grade above a minimum threshold, this threshold is recorded, otherwise this minimum threshold is zero.**

3. If the course is laboratory and the student is required to have a minimum grade in the written exam, then this limit is recorded. Otherwise, this minimum limit is zero.

The final score is as follows:

1. If the course is not laboratory, the final grade is equal to the grade of the written exam as the grade is calculated with 100% participation as already mentioned.
2. If the course is laboratory and the laboratory grade is strictly less than the relevant minimum, then zero (0) is automatically set as the written exam grade even if the update to a non-zero value is requested. The final score in this case is also zero.
3. If the course is a laboratory course and the written exam grade is strictly less than the relevant minimum, then the final grade is the written exam grade (the lab grade is not taken into account).
4. In any other case, the participation rate of the written exam is applied to combine the laboratory and written grades in deriving the final grade.

For a course to be considered successful and secured for a student, the student must have a final grade of five (5) or greater.

The courses are divided into two categories: (a) compulsory courses, and (b) optional compulsory courses. The first category includes core courses which provide basic knowledge and all of which must be successfully completed by a student to be able to graduate. The second category includes a large number of specialized courses, from which each student is required to choose and successfully complete a minimum number in order to graduate. Each course may have none or more prerequisite courses, and a student may not enroll in a semester course without successfully completing all prerequisite courses. Also, for each course there are one or more recommended courses which it is desirable (but not mandatory) for the student to know in order to attend it more easily. In each semester of the academic year, students apply

for registration (requested status) to attend the semester courses. Requests are created as proposed (status proposed) by the system. Each application is checked by the system that it meets the required conditions and is approved (approved status) or rejected (rejected status). In order for a student to be able to receive the graduation diploma, he must have completed the requirements set by the department, which are set each year and apply to students who register for the first time in that year. These conditions include the minimum number of elective compulsory courses and the minimum total number of teaching units. Students are required to prepare a thesis for which a title, description and grade

are recorded. A three-member committee is appointed by professors, one of whom is the work supervisor.

All diplomas are recorded in the system, they bear a unique serial number, the final grade of the diploma and the date of graduation. The diploma grade is calculated from the average of the grades of all courses required to obtain the diploma with a weighting factor of 80% and from the grade of the diploma thesis with a weighting factor of 20%. To calculate the course grade point average, each course grade is multiplied by the course weighting factor and the sum of the individual products is divided by the sum of all course weightings. The weighting factors are calculated according to the teaching units of each course, as shown in the table below:

Teaching Units	1-2	3-4	5
Gravity Factor	1	1.5	2

If a student has successfully completed more courses than the minimum number of courses required to obtain the diploma, the elective compulsory courses with the lowest passing grades are not counted towards the derivation of the final diploma grade, provided that all are fully satisfied the requirements for receiving a diploma from the remaining courses.

II. Implementation of the required functionality

Create a database in the PostgreSQL system and restore the data from the backup file that will be given to you. Then implement the following functionality:

1. Data management (building postgreSQL functions)

1.1. In the base given to you there are 2 tables Name, Surname containing Greek names and adjectives. Based on the data given to you in these tables, create the following functions (for teachers, laboratory staff and students – 3 functions) which will allow entering personal data of persons based on random selections of first and last names. These functions will accept as a parameter the number of records to be created. In addition, for the creation of students there will be a parameter for the date of enrollment in the department. The student registration number is of the format YYYYYYYY where YYYY is the year of registration date and YYYY is a unique (per year) sequence number. The rank of professors and laboratory staff members will be randomly selected from the respective set of possible values. The laboratory to which both are and are not included is also randomly selected from the set of laboratories recorded in the database.

1.2. Input grade for enrolled students in courses of a specific semester which is given as a parameter. A random integer from 1 to 10 will be entered as the written exam grade. Similarly, the laboratory grade will be entered if the course is laboratory. If grades already exist for some students, they are updated accordingly.

1.3. (*) Generate recommended enrollment for all students in semester courses of the current semester, taking into account their regular semester of study, prerequisites and previous courses that have not been passed. The number of proposed registrations ("proposed" status) will not be more than six per student, giving priority to courses of shorter semesters.

1.4. (*) Introduction of semester courses for a certain semester which is in "future" status. A semester course (CourseRun) is created for each course (Course) which course has a typical_season equal to the academic_season of that semester. The grading rules (grade_rules), teaching professors and (for lab courses) lab staff and lab (Lab) are copied from the most recent semester course for the same course (Course).

2. Data retrieval and calculations (building postgreSQL functions)

2.1. Retrieving the name and AMKA of professors and laboratory staff who belong to laboratories of a specific field for which the field code is given.

2.2. Retrieval of the courses together with the grade (the desired grade category will be given, i.e. written exam, lab grade or final grade) for the current semester and for a specific student (the student's AMKA is given).

2.3. Retrieving the registration number and registration year of the students who are registered in the current semester in optional compulsory courses.

2.4. Retrieve all codes of all compulsory courses marked YES or NO depending on whether a particular student whose registration number is given has passed them.

2.5. Retrieval of all elective compulsory courses (code and title) that are planned to be taught in the current semester but are not taught.

2.6. Find the teacher (AMKA and email) or teachers who have taught the most courses. Only the courses of completed semesters are counted.

2.7. Presentation of the honors rate per course (code) of a specific semester (determined by the combination of typical_year and typical_season). Only the scores of students who passed the course are considered. Those who passed the course with a final score greater than or equal to 8.5 are considered excellent.

2.8. Find the student load in the current semester. The load is calculated as the sum of the lecture, tutorial and laboratory hours for the courses in which a student is enrolled in the current semester.

2.9. (*) Retrieving course codes that depend directly or indirectly through prerequisite or recommended course relationships on a specific course whose code is given. In other words, it is requested to identify the courses that a student cannot attend or is not recommended to attend since he has not yet passed the specific course.

2.10.(*) Recovery of the AMKA of students who have successfully attended all compulsory courses taught in the current semester. 3.

Functionality with construction of triggers (triggers) in postgreSQL

2.10. Automatic correctness check when creating and updating semesters based on start and end dates so that it does not overlap with any other registered semester. Also, a correctness check should be done automatically for the chronological consistency of the status of the semesters. That is, not to violate the time order of the semesters based on the indication past, present, future, nor to allow the existence of multiple semesters with the indication present.

3.2. Automatic calculation of final grade and new status (pass / fail) of student registrations in courses when the necessary individual grades are completed or changed. For this purpose all the scoring rules described above in section I should be taken into account.

3.3. When inserting or updating the Semester table, the generated academic_year and academic_season attributes should be calculated accordingly.

3.4. Automatic check of student registration in a semester course so that the restrictions of prerequisite courses are satisfied and the total teaching units of the courses that the student will attend together with the course in question do not exceed 20 teaching units or the number of courses does not exceed 6. The check is triggered when inserting new "requested" records or when updating from 'proposed' to 'requested'. If the conditions are met, the status immediately becomes 'approved', while if the check fails then the status takes the value 'rejected'. Neither importing nor updating records to/from 'approved' or 'rejected' status is allowed as this will only be done via the trigger as already described.

4. Functionality using views

4.1. (*) Presentation of the supervisor and the members of the thesis committee for students who have not yet graduated. The view will have two fields (AMKA, Committee) where the second field will have the form: <surname1> <name1>, <surname2> <name2>, <surname3> <name3> where <surname1> <name1> are the elements of the supervisor and the rest of the committee members. When implementing the view keep in mind that there may be cases where committee members may not yet be registered in the database.

4.2. (*) Retrieve the number of students per year of enrollment for the last 10 years who meet graduation requirements and have not yet graduated. The result will be an array with columns: (year, count). If for any year this count is zero (0), the corresponding tuple will appear in the result with a value in the 'count' column equal to zero.

III. Deliverable

The deliverable of the A' phase of the laboratory work includes the database you implemented (backup file). You are not required to submit a written report.

The A' phase participates in a percentage of 70% in the final grade of the laboratory. The remaining 30% concerns the second phase of the work.

All functions marked with (*) can be delivered with the B' phase of the laboratory work.

~~Delivery date: April 12, 2019 Good luck!~~

IV. Install PostgreSQL and pgAdmin GUI

To install the PostgreSQL Database Management System and the pgAdmin GUI that you will use in the lab work, you can download the relevant installation file from the page https://www.enterprisedb.com/downloads/postgres-postgresql-downloads ~~Choose from the "Select your version"~~ list the latest version. The installation file contains both the PostgreSQL server and the pgAdmin GUI. Once it is downloaded to your computer run it and follow the instructions. Remember the password you will provide during the installation as you will be using it to connect to the server from pgAdmin. More information on the operation of the graphical environment and

the server will be given in the laboratories and tutorials of the course so that there is all the necessary support during the preparation of the laboratory work.

V. Installation of the database for the laboratory work

The speech is accompanied by a copy (backup) of the database that you will use in the laboratory work. To restore this copy to the server you will install on your computer, you must first log in through pgAdmin by putting 'localhost' as the host name/address and giving the password you set during installation. After logging in you will create a new database by right-clicking on the 'Databases' item under the node in the Servers hierarchy corresponding to your local server and selecting 'Create ÿ' 'Database'. Give the base a name of your choice and then right-click on the node that will be created with the name you gave and select 'Restore…'. In the dialog box that will appear, select the file corresponding to the copy of the database given to you (Filename field and selecting the file by pressing the '...' button). Finally you press the 'Restore' button and the database is created with all the tables, functions, formulas, etc. containing.

VI. Database Conceptual Diagram Following is

the relational entity diagram for the database you will use in the lab work. In addition to the tables that implement the entity types and relationship types of this diagram, the base also contains the Name, Surname tables with names and adjectives in the Greek language that you can use to create records for professors, students and laboratory staff.