CS348 – Spring 2020 - Project 2

Procedures and JDBC

Spring 2020

Due on: 03-30-2020 11:59 pm EST

Note: There will be a 10% penalty for each late calendar day. After five calendar days, the homework will not be accepted.

In this project, you are asked to complete 2 procedures with MYSQL Procedures and 2 JDBC Questions.

Notes:

1. The schema definition of database tables and sample test data are provided in tables.sql and data.sql, respectively. You need to use tables.sql to clean your database before you start this project because test data may be different from the data used in Project 1. Study and acquaint yourself with the schema (in tables.sql) and the data in the tables (in data.sql). This will make it easier to understand the required procedures.
2. You should finish all your work in answer.sql. Skeleton code for procedures is already provided in answer.sql. MySQL will give error messages if you do not finish all of the procedures, so you can comment the unfinished ones during development.
3. Please do not change the names of procedures in answer.sql.
4. Submit your answer via Blackboard.

1 Information about the University schema

1.1 Table: Professor

Professor(**ProfessorID**,Name,Position)

Professor contains the following attributes:

- ProfessorID: This attribute is an integer and the primary key for this table. Used to identify a Professor.
- Name: This attribute is a string that has the name of the Professor.
- Position: This attribute is a string that specifies the position/title of the professor.

1.2 Table: Department

Department(**DepartmentID**,Name,Head)

Department contains the following attributes:

- DepartmentID: This attribute is a integer and the primary key for this table. Used to identify a Department.
- Name: This attribute is a string that specifies which Department it is.
- Head: This attribute specifies the Professor Id of the Professor who is the head of the Department. This is a foreign key to Professor.

1.3 Table: Student

Student(**PUID**,Name,Address,Phone)

Student contains the following attributes:

- PUID: This attribute is a number and the primary key for this table. It specifies the PUID of the Student.
- Name: This attribute is a string that specifies the name of the Student.
- Address: This attribute is a string that specifies the address of the Student.
- Phone : This attribute specifies the phone number of the Student. This is a string.

1.4 Table: TA

TA(**TA_PUID**,Name,TA_Position,Department,SSN)

TA contains the following attributes:

- TA_PUID: This attribute is an integer and the primary key for this table. It specifies the PUID of the TA.
- Name: This attribute is a string that specifies the name of the TA.
- TA_Position: This attribute is a string that specifies the position of the TA. Position is either GTA or UTA.
- Department: This attribute is an integer. It specifies the department to which the student is associated with. This is a foreign key to Department.
- SSN: This attribute indicates the SSN of the TA. The SSN is an integer.

1.5 Table: Professor_Affiliation

Affiliated_With(**Professor**,**Department**,PrimaryAffiliation)

Professor_Affiliation contains the following attributes:

- Professor: This attribute is an integer which specifies the ProfessorID of the Professor. This is a foreign key to the Professor table.
- Department: This attribute is an integer that specifies the DepartmentID of the Department. This is a foreign key to the Department table.
- PrimaryAffiliation: This attribute is a boolean. It specifies for the given Professor whether it's his/her primary affiliation.(A Professor need not be a faculty of the department to teach a course affiliated with the department).

1.6 Table: Lecture

Lecture(**LectureID**,Student,TA,Professor,Start,End,Day)

Lecture contains the following attributes:

- LectureID: This attribute is an integer which specifies the LectureID.
- Student: This attribute is an integer that specifies the Student of the Student for whom the Lecture is booked. This is a foreign key to Students.
- TA: This attribute is an integer that specifies the TAID of the TA assigned to this Lecture. It is a foreign key to TA.
- Professor: This attribute is an integer that specifies the ProfessorID of the Professor assigned to this Lecture. It is a foreign key to Professor.
- Day: This attribute is a string that specifies the days when the Lecture happens. It could be either 'MWF' or 'TR'.
- Start: This attribute is of type time. It specifies the start time of the Lecture along with the date. To see how it looks, check **data.sql**.
- End: This attribute is of type time. It specifies the end time of the Lecture along with the date. The time for both the start and end is in 24 Hr format, so be careful while handling them.

2 Procedure and Functions

Note: Before starting your implementation please change the loginid to your Purdue username in test.sh,tables,data and drop.sql

1. **TA_NAMES**

Create a procedure TA_NAMES that accepts prof_id as input that will return the name of the professor, the TAs associated with the professor, the position of the TAs and a flag that says whether the professor is the head or not. (The name of the column you will print to stdout will be Head and it will either be "YES" or "NO" depending on whether the professor in question is the Head of a department or not).

Professor_Name	TA_NAME	TA_POSITION	HEAD
pname1	taname1	pos1	head1
pname2	taname2	pos2	head2

Table 1: Sample output for Question 1

2. **DEPT_DETAIL**

Create a function called DEPT_DETAIL that takes as a parameter a department id. For the id passed, retrieve the ID, name and position of the professor affiliated with the department who has the most lectures in a week.(**NOTE: Don't worry about the day attributes just work on the end and the start themselves. Eg: if MWF is the day and the lecture is 1 hr,just consider it as 1 hr instead of 3**.)

ProfessorID	Name	Position
id1	name1	pos1
id2	name2	pos2

Table 2: Sample output for Question 2

3 JDBC

For the following questions you will require JDBC to connect to the database and execute the queries. The connector jar file will be provided to you. In order to connect to the database you will need to provide your Purdue username and MYSQL password that you create at (`https://www.purdue.edu/apps/ics/a/MySQLAccount`).**(NOTE:You will need to supply your username and password as command line args for simplifying grading. The name of the file that you create should be Answers.java. Failure will result in penalty).**Implement the following questions:

1. Find the name of the TA who has the most number of Students. Display the details in the following format:

   Name is: *****
   Number of Students is : *****

2. Find the name,address and phone number of the Students who have a Lecture on 'MWF'.Also retrieve the name,position of the TA and the Professor. Display the details in the following format:

   Student Name : ****
   Student Address : ****
   Student Phone : ****
   Professor Name : ****
   Professor Position : ****
   TA name : ****
   TA Position : ****

   To compile the file run the following commands:
   javac -cp .:mysql-connector-java-8.0.17.jar Answers.java
   java -cp .:mysql-connector-java-8.0.17.jar Answers Purdueusername MYSQL-password