CS 348 - Homework 1 (version 3)

SQL
(160 Points)

Fall 2020

Due on: 09/18/2020

You may do this assignment individually or you may work with one partner. That is, this assignment is to be completed by individuals or by teams of two students. You should only talk to the instructor, the TA, and your partner about this assignment. You may also post questions (and not answers) to Campuswire.

Submit your answers using the hw1.py skeleton file included with this assignment (look at the homework tab in week 2 on Brightspace). The first and most important step is to write your Purdue username in the username function. Then, write your queries in the query1 to query16 functions. Each student in a team must submit their own copy of the homework on Brightspace. Each student must submit only one file.

There will be a 10% penalty if the homework is submitted 24 hours after the due date , a 15% penalty if the homework is submitted 48 hours after the due date, or a 20% penalty if the homework is submitted 72 hours after the due date. The homework will not be accepted after 72 hours, as a solution will be posted by then.

Given below is a relational schema about the details of a Student Job Application database. Use the following database to provide SQL queries corresponding to the questions below.

Note:

1. The schema definition of the tables and sample test data are provided in ***tables.sql*** and ***data.sql***, respectively.

2. You may test your queries against the sample records given in data.sql, but we will use different data for grading. You may add/modify the sample data to test your queries for corner cases. Each query below includes the expected answer when issuing the query against the sample test data.

3. Do not use PL/SQL for this project, use only the main SQL construct for all questions (nested / subqueries are allowed).

4. Write your answer to each query in **hw1.py**.

Student(**StudentId**, *StudentName*, *SchoolId*, *BirthDate*, *Grade*)
School(**SchoolId**, *SchoolName*, *Address*)
Company(**CompId**, *CompName*, *Address*)
Recruiter(**RecId**, *RecName*)
Job(**JobId**, *CompId*, *JobNum*, *JobTitle*, *Salary*, *OfferYear*)
Internship(**StudentId**, **CompId**, **RecId**, *OfferYear*)
JobApplication(**JobId**, **StudentId**, *ApplicationDate*)

1. Find out the recruiters' names, their ids, and the number of internships they have. Only show the results for recruiters who have at least 2 internships. (10 points)

RecName	RecId	NumOfInternships
Rec1	1	2

2. Find out the names of companies that did not offer jobs in the year 2020. Do not list duplicate company names. (10 points)

CompName
Microsoft
Facebook
UC Berkely
Spotify

3. List all students who have the highest age. Include the birth year and age for all of those students. Age is the difference between 2020 and the birth year. (15 points)

StudentName	BirthYear	Age
student8	1989	31

4. Find the number of jobs. (5 points)

NumberOfJobs
7

5. List companies that received applications from bright students. A student is considered bright if they have the highest grade (GPA) in their school. Eliminate duplicates in your result. (15 points)

CompName
Amazon

6. List names of students who have the highest grade in their school. For each of those students, include their school name and their grade. Sort your result by student grades in descending order. (10 points)

StudentName	SchoolName	MaxGrade
student8	University of Illinois	92.00
student1	Purdue	90.90
student3	Indiana University	70.70
student4	Ball State University	60.60

7. Find the names of students who applied to a job with a salary between 120k and 150k inclusive. Only consider jobs offered between 2019 and 2020 inclusive. List the student names and their respective salaries. (10 points)

StudentName	Salary
student1	150000
student2	130000
student3	150000
student4	130000

8. List each company name, its lowest salary, and the job titles of the lowest salary. (10 points)

CompName	JobTitle	Salary
Amazon	job3	110000
Microsoft	job7	90000
Facebook	job5	120000

9. Find the names of students who did an internship in 2019. List the company name and the school they attended. Sort your result by company name in ascending order. (10 points)

CompName	StudentName	SchoolName
Amazon	student1	Purdue
Facebook	student3	Indiana University

10. List the name of each school, the number of students who attend it, and the average grade of its students. Sort your result by by average grade in ascending order. Only show schools with more than one student and whose average grade is above 60. (10 points)

SchoolName	NumOfStudent	AverageGrade
University of Illinois	2	66.200000
Purdue	3	82.483333

11. Display a report of student internships. List student name, grade, and the recruiter id of all student's internships. Include students who did not have internships as well. (10 points)

StudentName	Grade	RecId
student1	90.90	1
student1	90.90	1
student2	80.80	2
student3	70.70	3
student4	60.60	NULL
student5	50.50	NULL
student6	75.75	NULL
student7	40.40	NULL
student8	92.00	NULL

12. Find students who applied to jobs in or after 2018 (including 2018). List the name and school name for each student. (10 points)

StudentName	SchoolName
student3	Indiana University
student4	Ball State University
student1	Purdue University

13. List the number of jobs each company has (not job applications). (10 points)

CompName	NumJobs
Amazon	3
Microsoft	2
Facebook	2

14. List student names for students born in 1992 and 1993. (10 points)

StudentName	BirthYear
student2	1992
student3	1992
student4	1993

15. List company names for companies that offer jobs with salary less than '120000'. (10 points)

CompName
Amazon
Microsoft

16. Find out the total salary for job1 and job2. (10 points)

SUM(Salary)
280000

Submission instructions:
Please fill out the functions (username, partnerUsername and 16 queries) in **hw1.py** and upload the file to **Brightspace**.

Your resulting tables should have the attributes in the same order as appeared in the sample answers. The order of rows is not important unless a question requests a specific order.

Make sure to **test** that python prints out the strings correctly.