

MSDS 7330 File Organization and Database Management Homework MySQL

Due Week 7

Name: Cory Nichols

This is a homework assignment for MSDS 7330, File Organization and Database Management. For this assignment, turn in a single pdf file containing all of your answers. The file should be named ;yourLastName;HW-MySQL.pdf. For example, the file name for my homework assignment would be 'EngelsHW-MySQL.pdf'. Insert your answer pages into this file with the answer for Question 1 inserted immediately after Question 1 and before Question 2, the answer for Question 2 inserted immediately after Question 2, and the answer for Question 3 inserted immediately after Question 3. You may insert a front page containing your name and date if you do not wish to or cannot electronically add that information to the first page of this homework sheet.

Collaboration is expected and encouraged; however, each student must hand in their own homework assignment. To the greatest extent possible, answers should not be copied but, instead, should be written in your own words. Copying answers from anywhere is plagiarism, this includes copying text directly from the textbook. Do not copy answers. Always use your own words and your own code. Directly under each question list all persons with whom you collaborated and list all resources used in arriving at your answer. Resources include but are not limited to the textbook used for this course, papers read on the topic, and Google search results. Don't forget to place your name on the first page of the pdf document.

Question 1:

MySQL Database

Download and install the MySQL Community Server database program on your computer. MySQL Community Server is a free download from <https://www.mysql.com>. If you are running a Microsoft operating system, you may download and install MySQL Workbench as well. If you are using a Mac, you may download the free app Sequel Pro from <http://www.sequelpro.com>. Both MySQL Workbench and Sequel Pro are visualization applications for accessing MySQL databases.

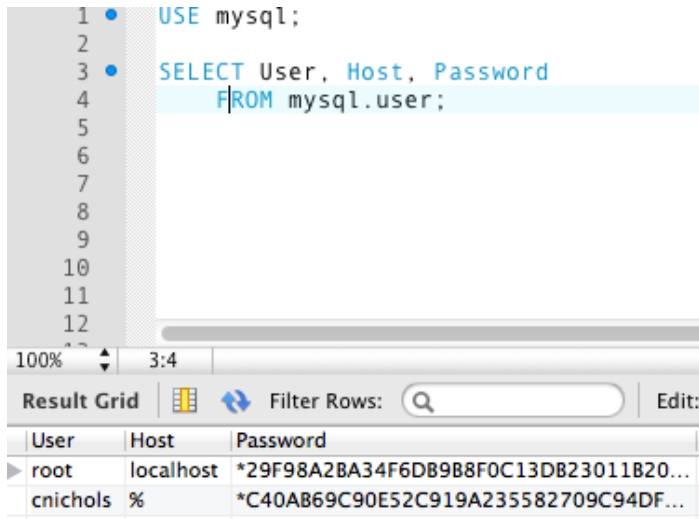
Once you have installed MySQL, be sure to set the password for your user account on the MySQL database. And, be sure to give your account the privileges needed to create and modify databases. The MySQL reference manual, available from <https://www.mysql.com>, provides in-depth instructions on how to install and configure your MySQL software.

Once you have installed and configured MySQL, select the mysql database by executing the “USE mysql” command. Then, run the query “SELECT User, Host, Password FROM mysql.user;” from the command line.

Capture the resulting output as a screen capture or grab and turn in the resulting pdf showing both the query and the results.

Collaborators: Cory Nichols

Resources: MySQL

QUERY OUTPUT FOR QUESTION 1:

```
1 • USE mysql;
2
3 • SELECT User, Host, Password
4 FROM mysql.user;
```

100% 3:4

Result Grid Filter Rows: Edit:

User	Host	Password
root	localhost	*29F98A2BA34F6DB9B8F0C13DB23011B20...
cnichols	%	*C40AB69C90E52C919A235582709C94DF...

Question 2:

The file baseball salaries 2003.txt contains salary information for certain professional baseball players from the year 2003. Write a SQL script that processes this file to determine, for each position, the average salary of the players in that position. Note that the seven player positions that can occur in the input file are “Catcher”, “First Baseman”, “Outfielder”, “Pitcher”, “Second Baseman”, “Shortstop” and “Third Baseman”.

Your script should create an appropriate table in your database (you may want to create a new database just for this problem) and populate it using the data in the input file. It should then execute a single SQL query whose output has the schema (position, avgSal). The output should appear sorted in descending order of average salary.

Capture the resulting output as a screen capture or grab. Turn in a pdf of your script and the results.

Collaborators: Cory Nichols

Resources: mySQL

Script: I used LOAD DATA LOCAL INFILE to load the baseball salaries.

```
-- QUESTION 2
6 CREATE DATABASE Baseball;
7
8 USE Baseball;
9
10 CREATE TABLE Stats
11 (TEAM VARCHAR(50),
12  PLAYER VARCHAR(50),
13  SALARY DECIMAL(20,5),
14  POSITION VARCHAR(20));
15
16 LOAD DATA LOCAL INFILE
17 '/Users/patrickcorynichols/Desktop/Data Science/Database Mgmt/Homework/baseball_salaries_2003.txt'
18 INTO TABLE Stats
19 FIELDS TERMINATED BY ':'
20 LINES TERMINATED BY '\n'
21 IGNORE 3 LINES
22 (Team, Player, Salary, Position);
23
24 UPDATE STATS
25 SET PLAYER = REPLACE(Player, '"', '');
26
27 SELECT POSITION, CONCAT('$', ROUND(AVG(SALARY),2)) avgSal
28 FROM STATS
29 GROUP BY POSITION
30 ORDER BY AvgSal DESC;
```

QUERY OUTPUT FOR QUESTION 2:

```
30
31 • SELECT POSITION, CONCAT('$',ROUND(AVG(SALARY),2)) avgSal
32 FROM STATS
33 GROUP BY POSITION
34 ORDER BY AvgSal DESC;
35
36
37
38
39
40
41
42
43
44
45
```

100% 5:40

Result Grid Filter Rows: Export:

POSITION	avgSal
Outfielder	\$4050024.41
First Baseman	\$3591402.63
Shortstop	\$2953382.23
Third Baseman	\$2461333.33
Pitcher	\$2179134.99
Second Baseman	\$1307750.00
Catcher	\$1172669.44

Result 5

Action Output

	Time	Action	Response	Duration / Fetch Time
✓ 71	15:36:36	USE B...	0 row(s) affected	0.00020 sec
✗ 72	15:36:43	SELE...	Error Code: 1054. Unknown column 'Avg_Salary' in 'order clause'	0.055 sec
✓ 73	15:37:06	SELE...	7 row(s) returned	0.074 sec / 0.00001...

Question 3:

Create a database called “University”. Load the textbook relations into your *University* database. (I’ve loaded all relations to the database, so all are represented below)

Access these relations and obtain answers for the following queries:

- 1) Produce a list of all the students in the student relation, including their ID, name and department name, sorted into ascending order by their name.
- 2) Produce a list of the names and salaries of professors in the Comp. Sci. and Elec. Eng. departments ordered by decreasing salary.

Capture the sequence of queries and resulting output thereby demonstrating your database in operation. Turn in a pdf of your database in operation.

Collaborators: Cory Nichols

Resources: mySQL

CREATE DATABASE and CREATE TABLE syntax:

```
CREATE DATABASE university;
USE university;

CREATE table classroom
  (building varchar(15),
   room_number varchar(7),
   capacity numeric(4,0),
   PRIMARY KEY(building, room_number));

CREATE TABLE department
  (dept_name varchar(20),
   building varchar(15),
   budget numeric(12,2) check(budget>0),
   primary key (dept_name));

CREATE TABLE course
  (course_id varchar(8),
   title varchar(50),
   dept_name varchar(20),
   credits numeric(2,0) check(credits>0),
   PRIMARY KEY(course_id),
   FOREIGN KEY(dept_name) REFERENCES department(dept_name)
   ON DELETE SET NULL);
```

```

CREATE TABLE instructor
  (ID VARCHAR(5),
   name VARCHAR(20) not null,
   dept_name VARCHAR(20),
   salary NUMERIC(8,2) CHECK(salary>29000),
   PRIMARY KEY(ID),
   FOREIGN KEY(dept_name) REFERENCES department(dept_name)
     ON DELETE SET NULL);

CREATE TABLE section
  (course_id VARCHAR(8),
   sec_id VARCHAR(8),
   semester VARCHAR(6) check(semester IN('Fall','Winter','Spring','Summer')),
   year NUMERIC(4,0) check(year > 1701 AND year < 2100),
   building varchar(15),
   room_number VARCHAR(7),
   time_slot_id VARCHAR(4),
   PRIMARY KEY(course_id, sec_id, semester, year),
   FOREIGN KEY(course_id) REFERENCES course(course_id)
     ON DELETE CASCADE,
   FOREIGN KEY(building,room_number) REFERENCES classroom(building,room_number)
     ON DELETE SET NULL);

CREATE table teaches
  (ID varchar(5),
   course_id VARCHAR(8),
   sec_id VARCHAR(8),
   semester VARCHAR(6),
   year numeric(4,0),
   PRIMARY KEY(ID, course_id, sec_id, semester, year),
   FOREIGN KEY(course_id, sec_id, semester, year)
     REFERENCES section(course_id, sec_id, semester, year)
     ON DELETE CASCADE,
   FOREIGN KEY(ID) REFERENCES instructor(ID));
-- ON DELETE SET NULL); -- this operation is not allowed in mySQL because ID is in the primary key

CREATE TABLE student
  (ID varchar(5),
   name varchar(20) NOT NULL,
   dept_name varchar(20),
   tot_cred numeric(3,0) check(tot_cred>=0),
   PRIMARY KEY(ID),
   FOREIGN KEY(dept_name) REFERENCES department(dept_name)
     ON DELETE set null);

```

```
CREATE TABLE takes
  (ID varchar(5),
   course_id varchar(8),
   sec_id varchar(8),
   semester varchar(6),
   year numeric(4,0),
   grade varchar(2),
   PRIMARY KEY(ID, course_id, sec_id, semester, year),
   FOREIGN KEY(course_id, sec_id, semester, year)
     REFERENCES section(course_id, sec_id, semester, year)
     ON DELETE CASCADE,
   FOREIGN KEY(ID)
     REFERENCES student(ID)
     ON DELETE CASCADE);

CREATE TABLE advisor
  (s_id varchar(5),
   i_id varchar(5),
   PRIMARY KEY(s_id),
   FOREIGN KEY (i_id) REFERENCES instructor(ID)
     ON DELETE SET NULL,
   FOREIGN KEY(s_id) REFERENCES student(ID)
     ON DELETE CASCADE);

CREATE TABLE prereq
  (course_id varchar(8),
   prereq_id varchar(8),
   PRIMARY KEY(course_id, prereq_id),
   FOREIGN KEY(course_id)
     REFERENCES course(course_id)
     ON DELETE CASCADE,
   FOREIGN KEY(prereq_id)
     REFERENCES course(course_id));

CREATE TABLE timeslot
  (time_slot_id varchar(4),
   day varchar(1) check(day in('M','T','W','R','F','S','U')),
   start_time time,
   end_time time,
   PRIMARY KEY (time_slot_id, day, start_time));
```


INSERT SYNTAX:

```
INSERT INTO department(dept_name, building, budget) VALUES
('Biology', 'Watson', 90000),
('Comp. Sci.', 'Taylor', 100000),
('Elec. Eng.', 'Taylor', 85000),
('Finance', 'Painter', 120000),
('History', 'Painter', 50000),
('Music', 'Packard', 80000),
('Physics', 'Watson', 70000)
;
```

```
INSERT INTO student
(ID, name, dept_name, tot_cred)
VALUES
(00129, 'Zhang', 'Comp. Sci.', 102),
(19991, 'Brandt', 'History', 80),
(12345, 'Shankar', 'Comp. Sci.', 32),
(23121, 'Chavez', 'Finance', 110),
(44553, 'Peltier', 'Physics', 56),
(45678, 'Levy', 'Physics', 46),
(54321, 'Williams', 'Comp. Sci.', 54),
(55739, 'Sanchez', 'Music', 38),
(70557, 'Snow', 'Physics', 0),
(76543, 'Brown', 'Comp. Sci.', 58),
(76653, 'Aoi', 'Elec. Eng.', 60),
(98765, 'Bourikas', 'Elec. Eng.', 98),
(98988, 'Tanaka', 'Biology', 120);
```

```
INSERT INTO instructor
(ID, name, dept_name, salary)
VALUES
(10101, 'Srinivasan', 'Comp. Sci.', 65000),
(12121, 'Wu', 'Finance', 90000),
(15151, 'Mozart', 'Music', 40000),
(22222, 'Einstein', 'Physics', 95000),
(32343, 'El Said', 'History', 60000),
(33456, 'Gold', 'Physics', 87000),
(45565, 'Katz', 'Comp. Sci.', 75000),
(58583, 'Califieri', 'History', 62000),
(76543, 'Singh', 'Finance', 80000),
(76766, 'Crick', 'Biology', 72000),
(83821, 'Brandt', 'Comp. Sci.', 92000),
(98345, 'Kim', 'Elec. Eng.', 80000);
```

Query Results for Question 3:

190				195	
191				196	SELECT name, salary
192	SELECT id, name, dept_name			197	FROM Instructor
193	FROM Student			198	WHERE dept_name IN ('Comp. Sci.', 'Elec. Eng.')
194	ORDER BY Name ASC;			199	ORDER BY salary DESC;
195				200	
100%	1:192			201	
Result Grid	Filter Rows:			202	
id	name	dept_name		100%	1:200
76653	Aoi	Elec. Eng.		Result Grid	Filter Rows:
98765	Bourikas	Elec. Eng.			Export:
19991	Brandt	History		name	salary
76543	Brown	Comp. Sci.		Brandt	92000.00
23121	Chavez	Finance		Kim	80000.00
45678	Levy	Physics		Katz	75000.00
44553	Peltier	Physics		Srinivasan	65000.00
55739	Sanchez	Music			
12345	Shankar	Comp. Sci.			
70557	Snow	Physics			
98988	Tanaka	Biology			
54321	Williams	Comp. Sci.			
129	Zhang	Comp. Sci.			