**Lab: SQL Practice**

**Objective**: students learn how to use SQL to set up databases and retrieve information from the database.

**Note: the instruction in this lab is using MySQL but you are free to use any other DBMS that offers similar functionalities. If you are using any other DBMS, please set up tables accordingly and populate data into tables in a similar way using other DBMS tool, then you could jump to step 7. If you have using Mysql, please start from step 1.**

In this lab, you will use MySQL(or any other DBMS) to create a database and execute SQL queries.

(Setting up: 4 points, queries: 21 points)

**Step 1: Getting started**

For this assignment, we will use the university dataset. We have all the csv files for this database compressed and posted on Canvas. To download the dataset, do the following steps:

1. Login to Class Canvas
2. Click on Week 5 to download the dataset under Lab (university.zip).
3. Save and extracted the downloaded dataset to a folder in your directory (for example: c:\data)

On successful execution, following contents are extracted:

* \*.csv (files)
* readme.pdf file explaining structure of tables in the database.

**Step 2: Getting familiar with MySQL**

Please follow the instruction on how to install and set up MySQL server on your Windows computers posted on Canvas.

**Step 3: Understanding the data**

Please refer to readme.pdf for the description of each table.

**Step 4: Login into MySQL and create the university database.** Depending on how you set it up, you could login into MySQL from Command line (cmd from Windows to take you to system prompt) or you could use MysqlWorkBench. The following instruction is for Command line:

C:\Data> mysql –u <username> -p

Password: \*\*\*\*\*\*\*

Please replace <your username> with the actual user name in your set up.

mysql> create database university;

mysql> show databases; // you should see university database listed here

**Step 5: Create the following tables in MySQL**

1. Class: (cname, meets\_at, room, fid) PK: cname

|  |  |
| --- | --- |
| Field name | Data type |
| Cname | Varchar(255) |
| meets\_at | Varchar(255) |
| Room | Varchar(255) |
| Fid | Bigint |

1. Enroll (snum, cname) PK (snum, cname)

|  |  |
| --- | --- |
| Field name | Data type |
| Snum | Bigint |
| Cname | Varchar(255) |

1. Faculty(fid, fname, deptid) PK(fid)

|  |  |
| --- | --- |
| Field name | Data type |
| Fid | Bigint |
| Fname | Varchar(255) |
| Deptid | Bigint |

1. Student (snum, sname, major, level, age) PK (snum)

|  |  |
| --- | --- |
| Field name | Data type |
| Snum | Bigint |
| Sname | Varchar(255) |
| Major | Varchar(255) |
| Level | Varchar(255) |
| Age | int |

**Step 6: Populate data into those tables:**

**Important note for Mysql server 8.0:**

*You may need to login into mysql as a root user and issue:*

SET GLOBAL local\_infile = true;

*Before you could load data from a text file in console.*

After you downloaded all the csv files from Canvas, please logout from MySQL console:

Mysql> exit

You are then brought back to command line. At the command line, please type:

mysql -u <your username> -p --local-infile university

You will be asked to type in your password. Please replace <your username> with the actual user name in your set up.

Then after login successfully inside the MySQL, load a csv file into a table as follows:

mysql> load data local infile <path to the csv file> into table <table name> fields terminated by ',' lines terminated by '\n';

Example:

mysql> load data local infile ‘:\\data\\student.csv' into table student fields terminated by ',' lines terminated by '\r\n';

You will load all 4 files into 4 corresponding table.

**Step 7**: Checking these tables to make sure the number of attributes and instances are correct as they are shown in the csv files. Using a set of select \* from <table name> command to do that.

**Step 8: Writing queries (1 point/a query)**

**Please fill in the SQL and Result columns of the following table for each query.**

|  |  |  |
| --- | --- | --- |
| Query | SQL | Result |
| 1. Find the student id of all students whose name starts with “M”. | SELECT  Snum  FROM  Student  WHERE  Sname LIKE "M%"; |  |
| 2. Find the name of the classes that meet at room “R12”. | SELECT  Cname  FROM  Class  WHERE  Room = 'R12'; |  |
| 3. Find the name of all faculty members who are working at department “20”. | SELECT  Fname  FROM  Faculty  WHERE  Deptid = 20; |  |
| 4. Find the names of all Juniors (level='JR') who are currently enrolled in “Database Systems”. | SELECT  S.Sname  FROM  Student AS S,  Enroll AS E  where  S.Snum = E.Snum  AND S.Level = 'JR'  AND E.Cname = 'Database Systems'; |  |
| 5. Find the names of all Juniors (level='JR') who are enrolled in a class taught by “I. Teach”. | SELECT  S.Sname  FROM  Student AS S,  Enroll AS E,  Class AS C,  Faculty AS F  WHERE  S.Level = 'JR'  AND E.Snum = S.Snum  AND E.Cname = C.Cname  AND C.Fid = F.Fid  AND F.Fname = 'I. Teach'; |  |
| 6. Find the names of all classes that either meet in room “R128” or meet MWF | SELECT  Cname  FROM  Class  WHERE  Room = 'R128'  OR meets\_at LIKE 'MWF%'; |  |
| 7. Find all the names of all classes taught by Elizabeth Taylor. | SELECT Cname  FROM  Class as C,  Faculty AS F  WHERE  C.Fid = F.Fid  AND F.Fname = 'Elizabeth Taylor' |  |
| 8.Find the names, rooms and schedule of all enrolled classes form Joseph Thompson. | SELECT  C.Cname,  C.Room,  C.meets\_at  FROM  Student AS S,  Enroll AS E,  Class AS C  WHERE  S.Sname = 'Joseph Thompson'  AND S.Snum = E.Snum  AND E.Cname = C.Cname; |  |
| 9. Find the names of all faculty members who teach at “R128”. | SELECT  Cname  FROM  Class AS C,  Faculty AS F  WHERE  C.Fid = F.Fid  AND C.Room = 'R128'; |  |
| 10. Find all the pairs of classes that meet at the same time (produce pairs in alphabetic order). | SELECT  CONCAT(C1.Cname, ', ', C2.Cname) AS class\_pair  FROM  Class AS C1,  Class AS C2  WHERE  C1.meets\_at = C2.meets\_at  AND C1.Cname < C2.Cname  ORDER BY  class\_pair; |  |
| 11.Find the age of the oldest student who is either a History major or enrolled in a course taught by “I. Teach”. | SELECT  MAX(S.Age)  FROM  Student AS S,  Enroll AS E,  Class AS C,  Faculty AS F  WHERE  Major = 'History'  OR (  E.Snum = S.Snum  AND E.Cname = C.Cname  AND C.Fid = F.Fid  AND F.Fname = 'I. Teach'  ); |  |
| 12.Find the names of all classes that either meet in room “R128” or have five or more students enrolled. | SELECT  C.Cname  FROM  Class AS C  LEFT JOIN (  SELECT  Cname,  COUNT(Snum) AS student\_count  FROM  Enroll  GROUP BY  Cname  ) AS T ON T.Cname = C.Cname  WHERE  (C.Room = 'R128')  OR (T.student\_count >= 5); |  |
| 13.Find the names of all students who are enrolled in two classes that meet at the same time. | SELECT  DISTINCT S.Sname  FROM  Student AS S,  Enroll AS E  RIGHT JOIN (  SELECT  DISTINCT C1.Cname AS c1name,  C2.Cname AS c2name  FROM  Class AS C1,  Class AS C2  WHERE  C1.meets\_at = C2.meets\_at  AND C1.Cname < C2.Cname  ) AS T ON E.Cname = T.c1name  WHERE  E.Snum = S.Snum; |  |
| 14.Find the names of faculty members for whom the combined enrollment of the courses that they teach is less than five. | SELECT  F.Fname  FROM  Faculty F  LEFT JOIN (  SELECT  COUNT(E.Snum) AS num\_enroll,  C.Fid  FROM  Enroll AS E,  Class AS C  where  E.Cname = C.Cname  GROUP BY  Fid  ) AS T ON F.Fid = T.Fid  WHERE  num\_enroll < 5; |  |
| 15.For each level (FR, SO, JR, SR), print the level and the average age of students for that level. | SELECT  S.Level,  AVG(S.Age)  FROM  Student AS S  GROUP BY  S.Level; |  |
| 16.For all levels except JR, print the level and the average age of students for that level. | SELECT  S.Level,  AVG(S.Age)  FROM  Student AS S  WHERE  S.Level <> 'JR'  GROUP BY  S.Level; |  |
| 17.For each faculty member that has taught classes only in room “R128”, print the faculty member’s name and the total number of classes she or he has taught. | SELECT F.Fname, COUNT(C.Cname) AS class\_num  FROM Faculty AS F  JOIN Class AS C ON F.Fid = C.Fid  WHERE C.Fid NOT IN (  SELECT C.Fid  FROM Class AS C  WHERE C.Room <> 'R128'  )  AND C.Room = 'R128'  GROUP BY F.Fid, F.Fname; |  |
| 18.Find the names of students enrolled in the maximum number of classes. | SELECT S.Sname  FROM Student AS S  JOIN Enroll AS E ON S.Snum = E.Snum  GROUP BY S.Sname  HAVING COUNT(E.Cname) = (  SELECT MAX(ClassCount)  FROM (  SELECT COUNT(E.Cname) AS ClassCount  FROM Enroll AS E  GROUP BY E.Snum  ) AS MaxClasses  ); |  |
| 19.Find the names of students not enrolled in any class. | SELECT  S.Sname  FROM  Student AS S  WHERE  S.Snum NOT IN (  SELECT  DISTINCT E.Snum  FROM  Enroll AS E  ); |  |
| 20.For each age value that appears in Students, find the level value that appears most often. For example, if there are more FR level students aged 18 than SR, JR, or SO students aged 18, you should print the pair (18, FR). | WITH LevelCounts AS (  SELECT  Age,  Level,  COUNT(\*) AS Count  FROM  Student  GROUP BY  Age,  Level  ),  MaxCounts AS (  SELECT  Age,  MAX(Count) AS MaxCount  FROM  LevelCounts  GROUP BY  Age  )  SELECT  L.Age,  L.Level  FROM  LevelCounts AS L  JOIN MaxCounts AS M ON L.Age = M.Age  AND L.Count = M.MaxCount  ORDER BY  L.Age ASC; |  |
| 21.Find the average age of students who enroll in classes taught by “I. Teach” | SELECT AVG(S.Age) AS avg\_age  FROM Student AS S  JOIN Enroll AS E ON S.Snum = E.Snum  JOIN Class AS C ON E.Cname = C.Cname  JOIN Faculty AS F ON C.Fid = F.Fid  WHERE F.Fname = 'I. Teach'; |  |

**Step 9**: Filling in the table in Step 8 and submit this table to the Lab assignment by the due date.