

**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 MysqWorkBench. 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

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

Field name	Data type
Snum	Bigint
Cname	Varchar(255)

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

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

4. 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”.	<code>SELECT Snum FROM Student WHERE Sname <b>LIKE</b> "M%";</code>	+-----+  Snum  +-----+  51135593  +-----+  280158572  +-----+  451519864  +-----+
2. Find the name of the classes that meet at room “R12”.	<code>SELECT Cname FROM Class WHERE Room = 'R12';</code>	+-----+  Cname  +-----+  Introductory Latin  +-----+  Organic Chemistry  +-----+
3. Find the name of all faculty members who are working at department “20”.	<code>SELECT Fname FROM Faculty WHERE DeptId = 20;</code>	+-----+  Fname  +-----+  Mary Johnson  +-----+  I. Teach  +-----+  David Anderson  +-----+  Linda Davis  +-----+  Ulysses Teach  +-----+
4. Find the names of all Juniors (level='JR') who are currently enrolled in “Database Systems”.	<code>SELECT S.Sname FROM Student AS S, Enroll AS E where S.Snum = E.Snum AND S.Level = 'JR' AND E.Cname = 'Database Systems';</code>	+-----+  Sname  +-----+  Christopher Garcia  +-----+  Paul Hall  +-----+

5. Find the names of all Juniors (level='JR') who are enrolled in a class taught by "I. Teach".	<pre><b>SELECT</b> S.Sname <b>FROM</b> Student AS S, Enroll AS E, Class AS C, Faculty AS F <b>WHERE</b> S.Level = 'JR' <b>AND</b> E.Snum = S.Snum <b>AND</b> E.Cname = C.Cname <b>AND</b> C.Fid = F.Fid <b>AND</b> F.Fname = 'I. Teach';</pre>	<table border="1"> <thead> <tr><th>Sname</th></tr> </thead> <tbody> <tr><td>Christopher Garcia</td></tr> <tr><td>Paul Hall</td></tr> </tbody> </table>	Sname	Christopher Garcia	Paul Hall							
Sname												
Christopher Garcia												
Paul Hall												
6. Find the names of all classes that either meet in room "R128" or meet MWF	<pre><b>SELECT</b> Cname <b>FROM</b> Class <b>WHERE</b> Room = 'R128' <b>OR</b> meets_at <b>LIKE</b> 'MWF%';</pre>	<table border="1"> <thead> <tr><th>Cname</th></tr> </thead> <tbody> <tr><td>Archaeology of the Incas</td></tr> <tr><td>Dairy Herd Management</td></tr> <tr><td>Data Structures</td></tr> <tr><td>Database Systems</td></tr> <tr><td>Introduction to Math</td></tr> <tr><td>Introductory Latin</td></tr> <tr><td>Orbital Mechanics</td></tr> <tr><td>Patent Law</td></tr> <tr><td>Urban Economics</td></tr> </tbody> </table>	Cname	Archaeology of the Incas	Dairy Herd Management	Data Structures	Database Systems	Introduction to Math	Introductory Latin	Orbital Mechanics	Patent Law	Urban Economics
Cname												
Archaeology of the Incas												
Dairy Herd Management												
Data Structures												
Database Systems												
Introduction to Math												
Introductory Latin												
Orbital Mechanics												
Patent Law												
Urban Economics												
7. Find all the names of all classes taught by Elizabeth Taylor.	<pre><b>SELECT</b> Cname <b>FROM</b> Class <b>as</b> C, Faculty AS F <b>WHERE</b> C.Fid = F.Fid <b>AND</b> F.Fname = 'Elizabeth Taylor'</pre>	<table border="1"> <thead> <tr><th>Cname</th></tr> </thead> <tbody> <tr><td>Multivariate Analysis</td></tr> <tr><td>Patent Law</td></tr> </tbody> </table>	Cname	Multivariate Analysis	Patent Law							
Cname												
Multivariate Analysis												
Patent Law												
8. Find the names, rooms and schedule of all enrolled classes form Joseph Thompson.	<pre><b>SELECT</b> C.Cname, C.Room, C.meets_at <b>FROM</b> Student AS S, Enroll AS E, Class AS C <b>WHERE</b> S.Sname = 'Joseph Thompson' <b>AND</b> S.Snum = E.Snum <b>AND</b> E.Cname = C.Cname;</pre>	<table border="1"> <thead> <tr><th>Cname</th><th>Room</th><th>meets_at</th></tr> </thead> <tbody> <tr><td>Database Systems</td><td>1320 DCL</td><td>MWF 12:30-1:45</td></tr> <tr><td>Operating System Design</td><td>20 AVW</td><td>TuTh 12-1:20</td></tr> </tbody> </table>	Cname	Room	meets_at	Database Systems	1320 DCL	MWF 12:30-1:45	Operating System Design	20 AVW	TuTh 12-1:20	
Cname	Room	meets_at										
Database Systems	1320 DCL	MWF 12:30-1:45										
Operating System Design	20 AVW	TuTh 12-1:20										

9. Find the names of all faculty members who teach at “R128”.	<pre><b>SELECT</b> Cname <b>FROM</b> Class <b>AS</b> C, Faculty <b>AS</b> F <b>WHERE</b> C.Fid = F.Fid <b>AND</b> C.Room = 'R128';</pre>	<pre>+-----+   Cname +-----+   Archaeology of the Incas   Dairy Herd Management   Data Structures   Introduction to Math   Patent Law +-----+</pre>
10. Find all the pairs of classes that meet at the same time (produce pairs in alphabetic order).	<pre><b>SELECT</b> CONCAT(C1.Cname, ', ',  C2.Cname) <b>AS</b> class_pair <b>FROM</b> Class <b>AS</b> C1, Class <b>AS</b> C2 <b>WHERE</b> C1.meets_at = C2.meets_at <b>AND</b> C1.Cname &lt; C2.Cname <b>ORDER BY</b> class_pair;</pre>	<pre>+-----+   class_pair +-----+   American Political Parties, Multivariate Analysis   Archaeology of the Incas, Introductory Latin   Artificial Intelligence, Psychology   Dairy Herd Management, Optical Electronics   Dairy Herd Management, Organic Chemistry   Optical Electronics, Organic Chemistry +-----+</pre>
11. Find the age of the oldest student who is either a History major or enrolled in a course taught by “I. Teach”.	<pre><b>SELECT</b> MAX(S.Age) <b>FROM</b> Student <b>AS</b> S, Enroll <b>AS</b> E, Class <b>AS</b> C, Faculty <b>AS</b> F <b>WHERE</b> Major = 'History' <b>OR</b> ( E.Snum = S.Snum <b>AND</b> E.Cname = C.Cname <b>AND</b> C.Fid = F.Fid <b>AND</b> F.Fname = 'I. Teach' );</pre>	<pre>+-----+   MAX(S.Age)   +-----+   20   +-----+</pre>

12.Find the names of all classes that either meet in room “R128” or have five or more students enrolled.	<pre><code>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 &gt;= 5);</code></pre>	<pre>+-----+   Cname +-----+   Archaeology of the Incas   Dairy Herd Management   Data Structures   Database Systems   Introduction to Math   Operating System Design   Patent Law +-----+</pre>
13.Find the names of all students who are enrolled in two classes that meet at the same time.	<pre><code>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 &lt; C2.Cname ) AS T ON E.Cname = T.c1name WHERE E.Snum = S.Snum;</code></pre>	<pre>+-----+   Sname +-----+   Juan Rodriguez   Luis Hernandez +-----+</pre>
14.Find the names of faculty members for whom the combined enrollment of the courses that they teach is less than five.	<pre><code>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</code></pre>	<pre>+-----+   Fname +-----+   Elizabeth Taylor   Jennifer Thomas   Richard Jackson   William Moore   Patricia Jones   Mary Johnson   John Williams +-----+</pre>

	) AS T ON F.Fid = T.Fid <b>WHERE</b> num_enroll < 5;											
15. For each level (FR, SO, JR, SR), print the level and the average age of students for that level.	<pre> SELECT S.Level, AVG(S.Age) FROM Student AS S GROUP BY S.Level;</pre>	<table border="1"> <thead> <tr> <th>Level</th> <th>AVG(S.Age)</th> </tr> </thead> <tbody> <tr> <td>SR</td> <td>20.7143</td> </tr> <tr> <td>JR</td> <td>19.5000</td> </tr> <tr> <td>SO</td> <td>18.4000</td> </tr> <tr> <td>FR</td> <td>17.6667</td> </tr> </tbody> </table>	Level	AVG(S.Age)	SR	20.7143	JR	19.5000	SO	18.4000	FR	17.6667
Level	AVG(S.Age)											
SR	20.7143											
JR	19.5000											
SO	18.4000											
FR	17.6667											
16. For all levels except JR, print the level and the average age of students for that level.	<pre> SELECT S.Level, AVG(S.Age) FROM Student AS S <b>WHERE</b> S.Level &lt;&gt; 'JR' GROUP BY S.Level;</pre>	<table border="1"> <thead> <tr> <th>Level</th> <th>AVG(S.Age)</th> </tr> </thead> <tbody> <tr> <td>SR</td> <td>20.7143</td> </tr> <tr> <td>SO</td> <td>18.4000</td> </tr> <tr> <td>FR</td> <td>17.6667</td> </tr> </tbody> </table>	Level	AVG(S.Age)	SR	20.7143	SO	18.4000	FR	17.6667		
Level	AVG(S.Age)											
SR	20.7143											
SO	18.4000											
FR	17.6667											
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.	<pre> SELECT F.Fname, COUNT(C.Cname) AS class_num FROM Faculty AS F JOIN Class AS C ON F.Fid = C.Fid <b>WHERE</b> C.Fid NOT IN ( SELECT C.Fid FROM Class AS C <b>WHERE</b> C.Room &lt;&gt; 'R128' ) <b>AND</b> C.Room = 'R128' GROUP BY F.Fid, F.Fname;</pre>	<table border="1"> <thead> <tr> <th>Fname</th> <th>class_num</th> </tr> </thead> <tbody> <tr> <td>Robert Brown</td> <td>1</td> </tr> </tbody> </table>	Fname	class_num	Robert Brown	1						
Fname	class_num											
Robert Brown	1											
18. Find the names of students enrolled in the maximum number of classes.	<pre> SELECT S.Sname FROM Student AS S JOIN Enroll AS E ON S.Snum = E.Snum GROUP BY S.Sname <b>HAVING</b> COUNT(E.Cname) = ( SELECT MAX(ClassCount) FROM ( SELECT COUNT(E.Cname) AS ClassCount FROM Enroll AS E GROUP BY E.Snum ) AS MaxClasses );</pre>	<table border="1"> <thead> <tr> <th>Sname</th> </tr> </thead> <tbody> <tr> <td>Juan Rodriguez</td> </tr> <tr> <td>Luis Hernandez</td> </tr> <tr> <td>Ana Lopez</td> </tr> </tbody> </table>	Sname	Juan Rodriguez	Luis Hernandez	Ana Lopez						
Sname												
Juan Rodriguez												
Luis Hernandez												
Ana Lopez												

19.Find the names of students not enrolled in any class.	<pre> SELECT S.Sname FROM Student AS S WHERE S.Snum NOT IN ( SELECT DISTINCT E.Snum FROM Enroll AS E ); </pre>	<table border="1"> <thead> <tr><th>Sname</th></tr> </thead> <tbody> <tr><td>Maria White</td></tr> <tr><td>Charles Harris</td></tr> <tr><td>Angela Martinez</td></tr> <tr><td>Thomas Robinson</td></tr> <tr><td>Margaret Clark</td></tr> <tr><td>Dorthy Lewis</td></tr> <tr><td>Daniel Lee</td></tr> <tr><td>Nancy Allen</td></tr> <tr><td>Mark Young</td></tr> <tr><td>Donald King</td></tr> <tr><td>George Wright</td></tr> <tr><td>Steven Green</td></tr> <tr><td>Edward Baker</td></tr> </tbody> </table>	Sname	Maria White	Charles Harris	Angela Martinez	Thomas Robinson	Margaret Clark	Dorthy Lewis	Daniel Lee	Nancy Allen	Mark Young	Donald King	George Wright	Steven Green	Edward Baker
Sname																
Maria White																
Charles Harris																
Angela Martinez																
Thomas Robinson																
Margaret Clark																
Dorthy Lewis																
Daniel Lee																
Nancy Allen																
Mark Young																
Donald King																
George Wright																
Steven Green																
Edward Baker																
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).	<pre> 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; </pre>	<table border="1"> <thead> <tr><th>Age</th><th>Level</th></tr> </thead> <tbody> <tr><td>17</td><td>FR</td></tr> <tr><td>18</td><td>FR</td></tr> <tr><td>19</td><td>SO</td></tr> <tr><td>20</td><td>JR</td></tr> <tr><td>21</td><td>SR</td></tr> <tr><td>22</td><td>SR</td></tr> </tbody> </table>	Age	Level	17	FR	18	FR	19	SO	20	JR	21	SR	22	SR
Age	Level															
17	FR															
18	FR															
19	SO															
20	JR															
21	SR															
22	SR															

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';
```

avg_age
18.6000

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