# CS246: Database Management Systems Lab

**Lab # 10 (?? Questions, ?? Marks)**
**Lab session: AL1**
**Held on: 18-Mar-2024 (Mon)**
**Lab Timings: 14:00 to 17:00 Hours Pages: ??**
**Submission time: 16:45 Hrs, 17-Feb-2024**
**Instructors Dr. V. Vijaya Saradhi**
**Head TAs Adithya K Moorthy & Laxita Agrawal**
**Department of CSE, IIT Guwahati**

1. This lab requires the understanding of C programming APIs to connect to the `MySQL` server.
2. Attached are PDF files which help understand the APIs.
3. Two template C programs are shared
   (a) To establish database connection
   (b) To retrieve data from a table
4. Compile the program named `dbconnect.c` as: `gcc dbconnect.c -lmysqlclient`
5. You must perform insertion as individual statements. That is write several insert statements. **Any attempt to import all the data into tables leads to ZERO marks.**
6. Write all the SQL statements in a file named with `your_roll_number`.sql file name & extension and upload. Note replace the text `your_roll_number` with appropriate roll number. If you have several files, appropriately name them by prepending your roll number.
7. You must submit every file that is used to implement the following lab problem.
8. This lab theme is centered around chapter 5: Advanced SQL of the text book *Database System Concepts* Abraham Silberschatz, Henry F Korth & S. Sudarshan. Additional material on database connectivity APIs in C programming language is shared along with the assignment.

**Question 1:** (86 points)

Using `MySQL` and `MySQL C APIs` perform the following tasks:

**Task 01** (4 marks) Database connection API structures

1. (1 mark) Declare a pointer to structure `MYSQL` which holds information about database connection.
2. (1 mark) Initialize the above structure with the `mysql_init()` function which takes input `NULL`

3. (1 mark) Pass the database parameters namely `host, user name, password, database name` to the function `mysql_real_connect()`. Assume the last three parameters namely `port, unix_socket` and `client_flag` as `0, NULL, 0` respectively.
4. (1 mark) Create a database named *week10* using the API `mysql_query` function
5. An example program to establish database connection is shared.

**Task 02** (8 marks) **Create the following tables** using the `mysql_query` function

1. (2 marks) A table `student18` containing the following

| | | |
|---|---|---|
| $1^{st}$ column | `name` | string of characters of fixed size 100 |
| $2^{nd}$ column | `roll_number` | string of characters of fixed size 10 |

   with `roll_number` as primary key.
2. (2 marks) A table `course18` containing the following

| | | |
|---|---|---|
| $1^{st}$ column | `semester` | integer |
| $2^{nd}$ column | `cid` | string of characters of fixed size 7 |
| $3^{rd}$ column | `name` | string of characters of fixed size 100 |
| $4^{th}$ column | `l` | integer |
| $5^{th}$ column | `t` | integer |
| $6^{th}$ column | `p` | integer |
| $7^{th}$ column | `c` | integer |

   with `cid` as primary key.
3. (2 marks) A table `grade18` containing the following

| | | |
|---|---|---|
| $1^{st}$ column | `roll_number` | string of characters of fixed size 10 |
| $2^{nd}$ column | `cid` | string of characters of fixed size 7 |
| $3^{rd}$ column | `letter_grade` | string of characters of fixed size 2 |

   with `roll_number` and `cid` together form primary key
4. (2 marks) A table `curriculum` containing the following

| | | |
|---|---|---|
| $1^{st}$ column | `dept` | string of characters of fixed size 3 |
| $2^{nd}$ column | `number` | integer |
| $3^{rd}$ column | `cid` | string of characters of fixed size 7 |

**Task 03** (8 marks) **populate data** using the `mysql_query` function

1. (2 marks) Populate data from file `student18.csv` into table `student18`
2. (2 marks) Populate data from file `course18.csv` into table `course18`
3. (2 marks) Populate data from file `grade18.csv` into table `grade18`
4. (2 marks) populate data from the file `curriculum.csv` into table `curriculum`

**Task 04** (64) Write the following SQL queries in C program

1. (6 marks) List all the rows in the `student18` table using `C API`. To achieve this, perform the following
   (a) (1 mark) Declare a pointer to the structure `MYSQL_RES` say `student_res_set`
   (b) (1 mark) Write the query to list all rows of the table `student18` and invoke the function `mysql_query`

(c) (1 mark) Assign to `student_res_set` variable the value returned by the function `mysql_store_result` which take the pointer to the structure `MYSQL`

(d) (1 mark) Declare a variable of type `MYSQL_row`.

(e) (2 marks) Fetch every row from the `student_res_set` and print all the fields of the row.

2. (6 marks) List all the rows in the `course18` table using `C API`. To achieve this, repeat the steps given above.
3. (6 marks) List all the rows in the `grade18` table using `C API`. To achieve this, repeat the steps given above.
4. (6 marks) List all the rows in the `curriculum` table using `C API`. To achieve this, repeat the steps given above.
5. (10 marks) Write a C function `compute_spi` which takes pointer to `MYSQL` structure and `roll_number` as input and computes semester-wise SPI of the corresponding student. Limit the SPI precision to 2 digits after decimal.
6. (10 marks) Write a C function `compute_cpi` which takes pointer to `MYSQL` structure and `roll_number` as input and computes semester-wise CPI of the corresponding student. Limit the SPI precision to 2 digits after decimal.
7. (10 marks) Write a C function that perform the following: for every semester, check the `roll_number` has taken prescribed **core courses** given in the `curriculum`. Print only those roll numbers who have not taken the priscribed number of core courses.
   Following are the `core courses`.
   - Semester I: All courses
   - Semester II: All courses
   - Semester III: All courses
   - Semester IV: All courses except minor course.
   - Semester V: All courses except minor course.
   - Semester VI: All courses except minor course.
   - Semester VII: Only `CS498`.
   - Semester VIII: Only `CS499`.
8. (10 marks) Write a C function that perform the following: for every semester, check the `roll_number` has taken prescribed open elective courses given in the `curriculum`. `cid`'s which start with `OE` are considered open electives. Print the `roll_number` of students who have not taken priscribed number of open electives.
9. (1 mark) free the result set memory
10. (1 mark) close the database connection