Task :

1. We are creating a database for a university like MU for the website and server to access.
2. For lab 1 we will be creating a small section of the student database and a section of the campus database.

Database Name : **university_profile**

Note: please maintain the db and tables separate and don't delete them. We will be working on adding more data into the same tables and db. For lab hours use a different db. If you mess up the tables and the data you will lose marks if your output doesn't match. If you delete data be ready to re-insert everything.(or find a smart way to get it from your friends not gonna spoon feed)

Data Tables :

Student -

1. Should contain the following students and column values (exactly, as is).
2. **name** can be variable length at max upto 20 characters.
3. **rollno** must be unique and integer values.
4. All the **5 subjects** should be integers. note eng is english

name	rollno	age	math	sci	eng	social	sports
dwayne	33	21	98	94	91	96	55
john	58	18	70	87	77	98	67
dave	27	22	54	68	98	96	77
randy	56	19	69	75	65	67	98
kane	11	22	86	95	52	57	73
tom	50	21	76	84	62	74	81

Campus -

1. Should contain the following students and column values (exactly, as is).
2. **name** can be variable length at max upto 20 characters.
3. **cid** is college id should be unique integer
4. **loc** is location can be variable length at max upto 20 characters.
5. **cap** is capacity integer
6. Remaining 3 are types of schools in boolean (Note-mysql stores bool in 1,0 - read up)

name	cid	loc	cap	law	engg	buss
mec	101	hyd	1000	1	1	1
muc	104	mad	2000	0	1	1
mgt	107	bom	1500	1	0	1

Queries :

1. Management wants to know the cid of colleges in the uni that can support law.
2. Faculty wanted to inspect all the details of Dave excluding english and social.
3. Management wants to know the name and roll no of students who have qualified for scholarship. Qualification parameter : math OR sci above 80
4. A teacher for social wants to know what are the distinct marks she gave her students.
5. The english prof wishes to find the number of students failing in his class. Students below 70 fail. Count manually or pro tip : google count in sql. Both will be graded equally just trying to pique your curiosity.
6. The stem field wants to find the best qualified student to be head of their club as per marks. List the students name and rollno. (stem is sci and math in this case)
7. The Dept. of stem wants to know the college location of the university that does not provide engineering.
8. What is the ratio of students in the campus that provides stem and does not provide stem (query and show ratio by calc yourself, no need to divide just num/denom)
9. For all rounder find students who have score above 70 in all subj.
10. Find the count of students in each group if I split the students as per every 20 roll numbers in one group. Note you can run multiple queries and count individually for this one or see if you can google a faster way with sql itself. Both will be graded equally just trying to pique your curiosity.

Optional

11. Class teacher wants to find the student name and rollno with the maximum avg in all subjects. (google sum or avg)
12. An inquisitive student wants to find out the names of students whose names have the letter ‘a’ in the second position of their name
13. A math faculty wants to focus on the weak students first so he wants to see the names and marks of the students in asc order of marks

Method of submission :

1. Create a GitHub repository by the name 'dbms_lab' on your GitHub profile.
2. Create a folder in that repository by the name 'lab_1'.
3. Inside that folder, make an assign1.sql.
4. Put all your queries with comments on top of each query labelling as to which question they pertain to in the assignment in the 'assign1.sql' file, leave one line after each ans.
5. Optional problems in a separate optional.sql.
6. Put a text file in the same folder with the outputs of each query indexed by the q and leave a line between anss.
7. Share your GitHub repository links with me (next lab session will collect it).
8. Deadline is 6th Sep (Monday) night 11:59 pm.
9. Only the last commit on that day will be considered for submission, you can try making modifications after it but it won't be considered. 10. Bonus tip, try to keep the assignments in the repo as well so you can come back later (sometime in the future when applying for jobs or working on a project) and find all your data available in one place.