DBMS Lab
2021

Final Exam
[Marks 25, Duration 90 mnts]

Instruction

1. Create one pdf as <name>_<id>_DBMSLab21_FinalExam.pdf to submit. You also need to submit the SQL file as <name>_<id>_DBMSLab21_FinalExam.sql and submit the Python file <name>_<id>_DBMSLab21_FinalExam.py
2. Put questions in the order given here, and paste screenshots of results next to it such a way that your solution is also clearly visible.
3. In case it is required, write the solution explicitly along with a justification.
4. You can look into the manuals of MariaDB and MongoDB for finding the correct syntax only.

Part-I: Use MariaDB and university database [18]

1. Show the name of the departments which have more number instructors than number of students. [**2**]
2. Show the names of all students who have taken any course from the *Comp. Sci.* department. [**2**]
3. Show the name of the instructors and the courses they have taught if the number of courses taught by the instructor is more than 1. [**3**]
4. Show the names of the students who have taken courses from departments which are different from the department of their enrolment. Show the names of the students, names of the courses, departments of the courses, and departments of enrolments. [**3**]

5. Create a table *oddeven* that contains one integer field and one varchar(5) field. Create a procedure *poe*, that accepts two integers say a,b. Then *poe* inserts into table *oddeven* all integers between and including a and b. For the second field *poe* inserts 'odd' or 'even' depending on the integer value in the first field. For example { {1,'odd'}, {2,'even'} }. [**4**]
6. Create a function *get_user* which will return the username of the currently logged in user. Create a user *Snow* with password *white.* Demonstrate the function *get_user* using Snow. Use *get_user* in a query of your choice. [**4**]

Part-II: MariaDB and canteen database [4]

1. Create a database *canteen*. Create a table *menu* with attributes *id* int not null, and *name* varchar(50) not null, *type* that can take value between ' healthy' , and ' unhealthy'. Create another table *customerorder* with attribute *id* not null , and *count* int not null. Create a table price that will contain *id* of a dish in the menu and *amount* float. [**1**]
2. Create a trigger *init_price* that will check the type of the newly added dish in the menu, and will automatically initialize a price in the correct table. For healthy foods price is 10, and for unhealthy foods price is 15. [**3**]

Part-III: Use MongoDB and primer database [3]

1. Show the number of restaurants in Manhatttan for each cuisine in ascending order if the number is more than 100 and less than 500. [**3**]