DBMS Lab Sample Questions

1) Create a database named "COLLEGE" in MySQL and perform the following operations on it:
i) Create a table - FACULTY(facultyid,facultyname,dept,email-id,mobileno)
ii) Create a user "temp" and grant all permissions.
iii) Insert sample records into the table
iv) Write a Java Program to display all record in the table
v) Delete all records in the table
vi) Delete the database

2) Create a database named "COLLEGE" in MySQL and perform the following operations on it:
i) Create a table - FACULTY(facultyid,facultyname,dept,email-id,mobileno)
ii) Create a user "temp" and grant all permissions.
iii) Write a Java Program to insert sample records in the table
iv) Display all records in the table.
v) Delete all records in the table
vi) Delete the database

3) Create a database named "COLLEGE" in MySQL and perform the following operations on it:
i) Create a table - FACULTY(facultyid,facultyname,dept,email-id,mobileno)
ii) Create a user "temp" and grant all permissions.
iii) Insert sample records into the table
iv) Write a Java Program to delete a sample record in the table
v) Display all records in the table.
vi) Delete the database

4) Use MongoDB and create a sample database named "COMPANY" and perform the following operations on it :
i) Create a collection named "EMPLOYEE"
ii) list available databases and collections
iii) Create sample documents with fields : empid,empname,dept,email-id,mobileno,salary
iv) list all documents in the collection with salary > 25000
v) delete all documents in the collection
vi) delete the collection and database

5) Create the following tables with the mapping given below.
a. stu_details (reg_no, stu_name, DOB, address, city)
b. mark_details (reg_no, mark1, mark2, mark3, total)
(i) Alter the table mark_details to add a column average with data type as long.
(ii) Display the months between the DOB and till date.
(iii) Using alter command drop the column address from the table stu_details.
(iv) Write a stored procedures for finding the grade of the student.
(pass - >= 45%, First class - >=60%, Distinction - >=80%)
(v) Create a backup of the database.

6) Create the following tables with the mapping given below.
a. emp_details (emp_no, emp_name, DOB, address, doj, mobile_no, dept_no, salary).
b. dept_details (dept_no, dept_name, location).
(i) Display the months between the doj and till date.
(ii) Alter the table emp_details to add a primary key constraint on emp_no.
(iii) Create a new user u1. Give u1 permissions for insert and update only. Remove from u1 the pemission for update.
(iv) Create a backup of the database.
(v) Write a stored function that calculates the age of the employee.

DBMS Lab Sample Questions

7) Consider the following employee and department tables.
EMPLOYEE(empno, ename, designation, manager, hiredate, salary, commission,deptno)
DEPARTMENT(deptno, dname, location)
i. Create the above tables by properly specifying the primary keys and foreign keys and enter at least five tuples for each relation.
ii. List the names of employees whose name contain substring ‘LA’.
iii. List the details of employees of salary are greater than or equal to the average salary of employee table.
iv. Create a view which consists of details of all ‘SALESMAN’.
v. Write a stored function to display the experience of all employees in years.
vi. Create a backup of the database as employee.sql.
vii. Import employee.sql to database ‘employee1’.

8) Create the following tables with the mapping given below.
a. stu_details (reg_no, stu_name, DOB, address, city)
b. mark_details (reg_no, mark1, mark2, mark3, total)
(i). Display only those rows whose total ranges between 250 and 300.
(ii). Illustrate the use of TCL commands.
(iii). Delete the row whose reg_no=161.
(iv). Display all details whose names begins with 'a'.
(v). Write a trigger that keeps track of the old values before updation.
(vi). Drop the table mark_details.
(vii). Create a new user u3. Give u3 permissions for insert and update only. Remove from u1 the pemission for insert and update.

9) Consider the following tables.
SAILOR(sid, sname, rating, age)
BOATS(bid, bname, colour)
RESERVES(sid, bid, day)
i. Create the above tables by properly specifying the primary keys and foreign keys and enter at least five tuples for each relation.
ii. List the sailors in the descending order of their rating.
Iii. Create a view that contains sid and rating. Also create another view that contains bid and colour.
v. Write a stored procedure for classifying the sailor rating.
(high - >=8, medium - >=6, low - <5)
vi. Create a new user u2. Give u2 permissions for insert and update only. Remove from u1 the pemission for insert.