**Database Management Systems**

**Course code: CS/AI 3103**

**Assignment -3**

**Note:** Consider the database or tables you have created in assignment-1 to solve this assignment.

**Instructions to open mysql and using created database**

1. **Start mysql using**

$ mysql -u root –p

(enter password if prompted)

1. **Use database inside which you have created and store the tables given in the assignment-1**

use studentdb;

1. **Use the following table created in assignment-1 to solve this assignment**

Student (snum, sname, major, standing, age)

Faculty (fid, fname, deptid)

Class (name, meets, room, fid)

Enrolled (snum, cname)

1. **Create a table employee in studentdb database**

Create table employee(empid numeric(9,0),empname varchar(20), department varchar(20),salary numeric(10,0), bonus numeric(10,0));

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| empid | empname | department | salary | bonus |
| 1 | A | HR | 30000 | 6000 |
| 2 | B | Admin | 25000 | 5000 |
| 3 | C | HR | 50000 | 9000 |
| 4 | D | Admin | 25000 | 6000 |
| 5 | E | Account | 10000 | 4000 |
| 6 | F | Account | 12000 | 7000 |
| 7 | G | HR | 55000 | 12000 |

1. **Exercise on SELECT query:**

**Syntax: SELECT \* FROM Table\_name;**

**Or**

**Syntax: SELECT column 1, column 2, … ,column n FROM table\_name;**

1. Write the sql query to display all the records from the student table.

**Solution: SELECT \* FROM student;**

1. Write the sql query to fetch ‘snum’, ‘sname’ and ‘age’ columns from the student table.

**Solution: SELECT snum, sname, age FROM student;**

1. **Exercise on SELECT DISTINCT**

**Syntax: SELECT distinct column 1, column 2 FROM table\_name;**

1. Write the sql query to select the distinct ages from student table.

**Solution: SELECT DISTINCT age FROM student;**

1. **Exercise on SELECT WHERE**

**Syntax: SELECT column 1, column 2 FROM table\_name WHERE condition;**

1. Write the sql query to find the names of faculties whose age is 21.

**Solution: SELECT sname FROM student WHERE age=21;**

1. Write the sql query to find the names of students whose age is greater than 20.

**Solution: SELECT sname FROM student WHERE age>20;**

1. Write the sql query to find the names of the faculties in faculty table whose department id is greater than equal to 19.

**Solution: SELECT fname FROM faculty WHERE deptid>=19;**

1. Write the sql query to find the names of the faculties in faculty table whose department id is less than equal to 18.

**Solution: SELECT fname FROM faculty WHERE deptid<=18;**

1. Write the sql query to find the names of the faculties in faculty table whose department id is not equal to 20.

**Solution: SELECT fname FROM faculty WHERE deptid<>20;**

1. **Exercise on SELECT WHERE (AND, OR, NOT)**

**Syntax: SELECT column 1, column 2 FROM table\_name WHERE Condition1 AND Condition2;**

**Syntax: SELECT column 1, column 2 FROM table\_name WHERE Condition1 OR Condition2;**

**Syntax: SELECT column 1, column 2 FROM table\_name WHERE NOT Condition;**

1. Write the sql query to fetch the names of the students from student table where major is ENGLISH and age is 21.

**Solution: SELECT sname FROM student WHERE major='ENGLISH' AND age=21;**

1. Write the sql query to display the names of students from student table where major is Computer Science or standing is JR.

**Solution: SELECT sname FROM student WHERE major='Computer Science' OR standing='JR';**

1. Write the sql query to display the names of the students from student table whose standing is not SO.

**Solution: SELECT sname FROM student WHERE NOT standing='SO';**

1. Write the sql query to find the names of the student from the student table whose major is Computer Science OR standing is SR AND age is greater than or equal to 20.

**Solution: SELECT sname FROM student WHERE major='Computer Science' OR standing='SR' AND age>=20;**

1. Write the sql query to find the names of the student from the student table whose major is Computer Science OR standing is JR OR age is greater than 19.

**Solution: SELECT sname FROM student WHERE major='Computer Science' OR standing='JR' OR age>19;**

1. **Exercise on SELECT (Arithmetic Expressions)**

Syntax: SELECT column1 Arithmetic Operator column2 AS new column\_name FROM table\_name;

Or

Syntax: SELECT column1 Arithmetic Operator column2 AS new column\_name FROM table\_name WHERE condition;

1. Write an SQL query to display the total salary of each employee adding the Salary with bonus value.

**Solution: SELECT salary+bonus AS emptotalsalary FROM employee;**

1. Write an SQL query to display the total salary of each employee adding the Salary with bonus value where employee salary greater than 25000.

**Solution: SELECT salary+bonus AS emptotalsalary FROM employee WHERE salary>=25000;**

**Additional queries:**

1. Write the sql to find the names of all Juniors (level=JR) who are enrolled in a class taught by Ivana Teach.

**Solution: SELECT DISTINCT S.Sname**

**FROM Student S, Class C, Enrolled E, Faculty F**

**WHERE S.snum = E.snum AND E.cname = C.name AND C.fid = F.fid AND F.fname = 'Ivana Teach' AND S.standing = 'JR';**

1. Write the sql to print the level and the average age of students for that level ordered by the average age, for all levels except JR.

**Solution: SELECT S.standing, AVG(S.age)**

**FROM Student S**

**WHERE S.standing <> 'JR'**

**GROUP BY s.standing;**

1. Write the sql to find the total number of distinct students taught by a faculty member in all the courses he/she teaches.

**Solution: SELECT faculty.fid,count(distinct snum)**

**FROM faculty, class,enrolled**

**WHERE faculty.fid=class.fid and class.name=enrolled.cname**

**GROUP BY faculty.fid;**

1. Write the query to fetch the number of employees working in the department ‘HR’.

**Solution: SELECT COUNT(\*) FROM employee WHERE department='HR';**

1. Write the query to find all the employees whose salary is between 5000 to 50000.

**Solution: SELECT \* FROM Employee WHERE Salary BETWEEN '5000' AND '50000';**

1. Write the query to find the names of employees that begin with ‘A’.

**Solution: SELECT \* FROM Employee WHERE Empname LIKE 'A%';**

1. Write the query to retrieve employee name and department along with total salaries paid for each of them.

**Solution: SELECT empname, department, SUM(salary) from Employee GROUP BY empname, department;**