

Ramakrishna Mission Vidyamandira

(An Autonomous College Under University of Calcutta)

Computer Science (Honors) Semester IV 2025

Paper:

|  |
| --- |
| **Submitted by** |
| Class Roll Number: 261  B.Sc.  4th Semester  Batch: 2023-27 |

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SI NO.** | **Page start** | **Page end** | **D-O-S** | **SIGNATURE** |
| **Question 1** | **3** | **7** |  |  |
| **Question 2** | **8** | **15** |  |  |
| **Question 3** | **16** | **19** |  |  |
| **Question 4** | **20** | **24** |  |  |
| **Question 5** | **25** | **30** |  |  |

**Question 1:**

**STUDENTS (NAME, ROLL, ADDRESS, MAIN)**

**ADMISSION (ROLL, COURSE, SEMESTER)**

**FACULTY (COURSE, FACULTY, SEMESTER)**

**OFFERING (BRANCH, COURSE)**

**Write down the following SQL queries:**

**1. All courses taken by a given student.**

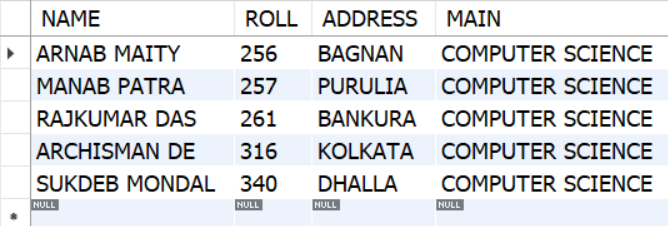
**2. Names of the students admitted in a particular course in a given semester.**

**3. Were two students (x and y) ever admitted in the same course in the same semester?**

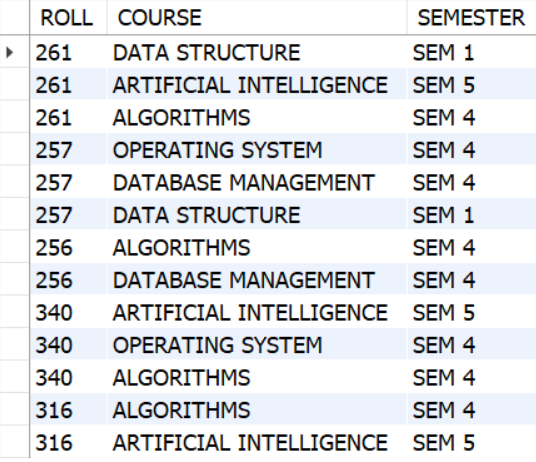
**4. Students who have taken all courses offered by a given faculty.**

**5. Find the name of the faculty who taught maximum courses.**

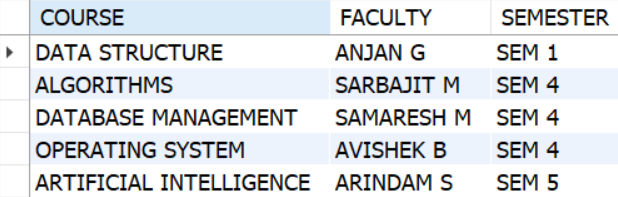
**Tables:**

**STUDENTS (NAME, ROLL, ADDRESS, MAIN)**

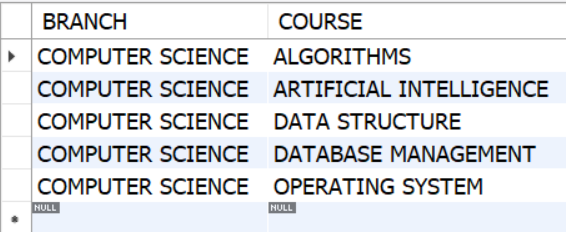
**ADMISSION (ROLL, COURSE, SEMESTER)**



**FACULTY (COURSE, FACULTY, SEMESTER)**



**OFFERING (BRANCH, COURSE)**



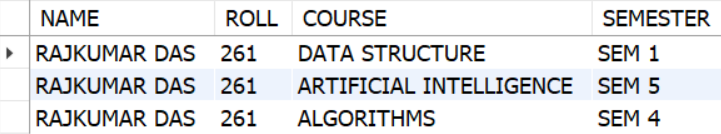
1. **All courses taken by a given student.**

**SELECT S.NAME,A.ROLL,A.COURSE,A.SEMESTER FROM STUDENTS AS S**

**JOIN ADMISSION AS A**

**ON S.ROLL = A.ROLL**

**WHERE S.NAME = 'RAJKUMAR DAS';**



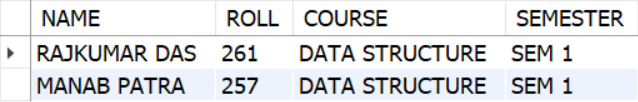
1. **Names of the students admitted in a particular course in a given semester.**

**SELECT S.NAME,A.ROLL,A.COURSE,A.SEMESTER FROM STUDENTS AS S**

**JOIN ADMISSION AS A**

**ON S.ROLL = A.ROLL**

**WHERE A.SEMESTER = 'SEM 1' AND A.COURSE = 'DATA STRUCTURE';**



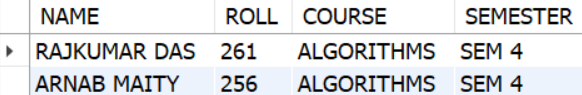
1. **Were two students (x and y) ever admitted in the same course in the same semester?**

**SELECT STUDENTS.NAME, ADMISSION.ROLL, ADMISSION.COURSE, ADMISSION.SEMESTER FROM STUDENTS**

**JOIN ADMISSION**

**ON ADMISSION.ROLL = STUDENTS.ROLL**

**WHERE ADMISSION.COURSE = "ALGORITHMS" AND ADMISSION.SEMESTER = 'SEM 4' AND ADMISSION.ROLL IN (261,256);**



1. **Students who have taken all courses offered by a given faculty.**

**SELECT S.NAME , A.COURSE FROM STUDENTS AS S**

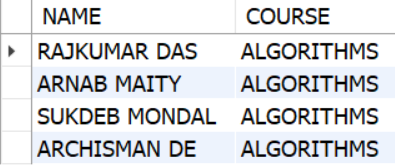
**JOIN ADMISSION AS A**

**ON S.ROLL = A.ROLL**

**WHERE A.COURSE IN (**

**SELECT COURSE FROM FACULTY WHERE FACULTY = 'SARBAJIT M'**

**);**

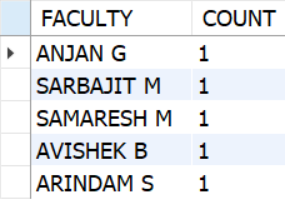


1. **Find the name of the faculty who taught maximum courses.**

**SELECT FACULTY, COUNT(\*) AS COUNT FROM FACULTY**

**GROUP BY FACULTY**

**ORDER BY COUNT(COURSE) DESC;**



**Question 2:**

**STUDENT (SID, SNAME, SEX, AGE, YEAR, GPA)**

**DEPT (DNAME, NUMPHDS)**

**PROF (PNAME, DNAME)**

**COURSE (CNO, CNAME, DNAME)**

**MAJOR (DNAME, SID)**

**SECTION (DNAME, CNO, SECTNO, PNAME)**

**ENROLL (SID, GRADE, DNAME, CNO, SECTNO)**

**Write down the following SQL queries:**

**1. List the names of professors who work in department that have fewer than 20 PhD.**

**2. List the names of students with the lowest GPA.**

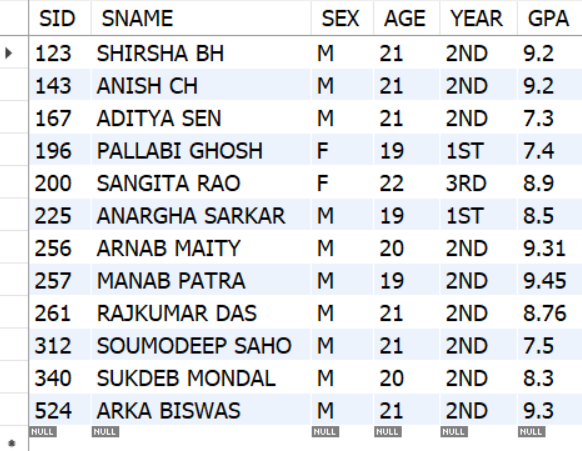
**3. List the name and sid of student enrolled in maximum classes.**

**4. List the name of student and department name whose sid=’Bsc 2017’**

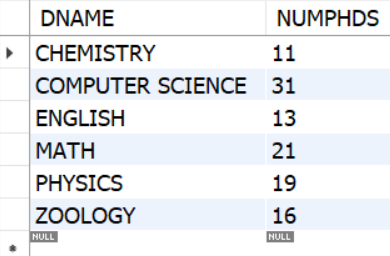
**5. List the names of students who are taking both a Computer Science and Mathematics**

**course.**

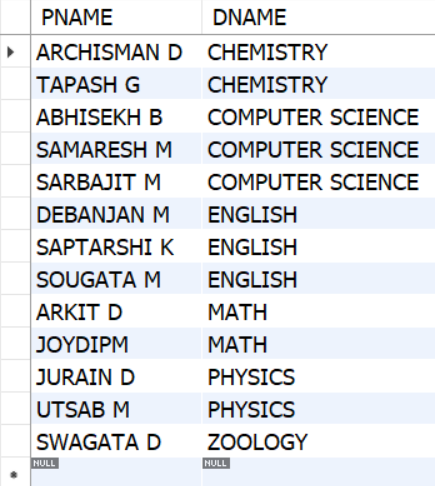
**STUDENT (SID, SNAME, SEX, AGE, YEAR, GPA)**



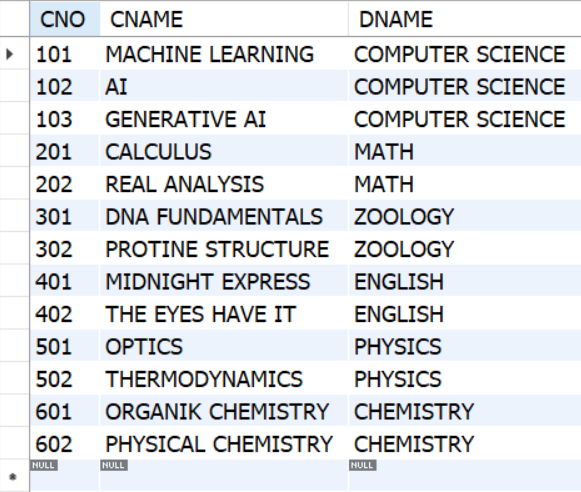
**DEPT (DNAME, NUMPHDS)**



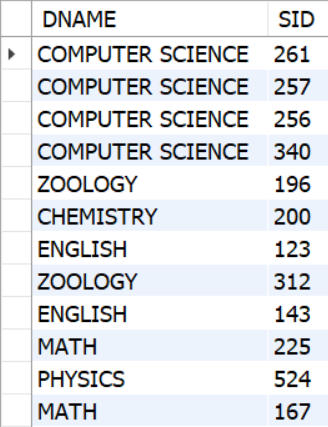
**PROF (PNAME, DNAME)**



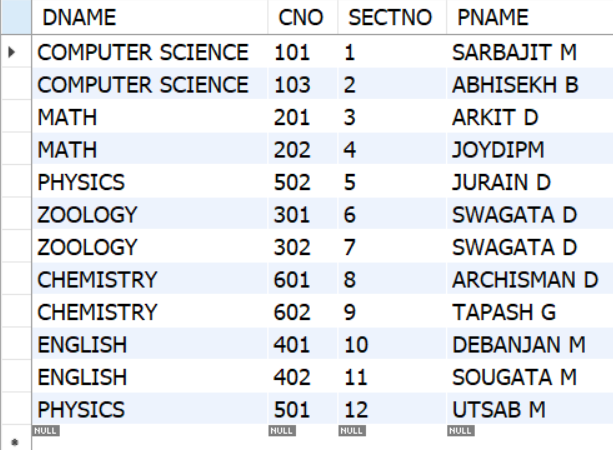
**COURSE (CNO, CNAME, DNAME)**



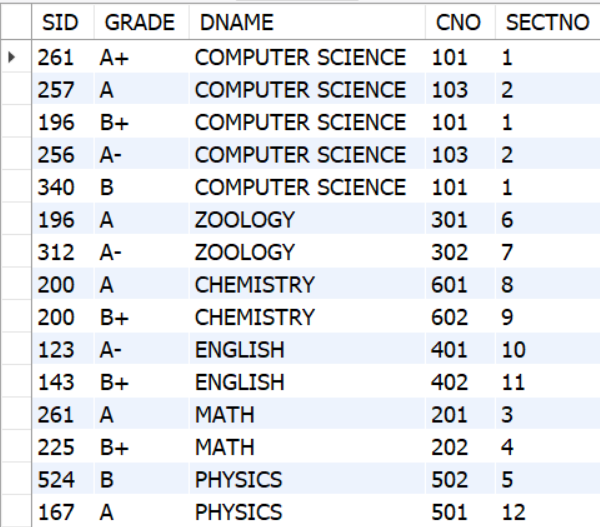
**MAJOR (DNAME, SID)**



**SECTION (DNAME, CNO, SECTNO, PNAME)**



**ENROLL (SID, GRADE, DNAME, CNO, SECTNO)**



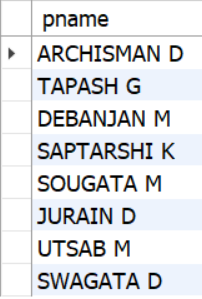
**1. List the names of professors who work in department that have fewer than 20 PhD.**

**select p.pname from prof as p**

**join dept as d**

**on p.dname = d.dname**

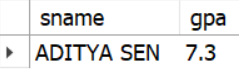
**where d.numphds < 20;**



**2. List the names of students with the lowest GPA.**

**select sname ,gpa from student**

**where gpa = (select min(gpa) from student);**



**3. List the name and sid of student enrolled in maximum classes.**

**SELECT s.sid, s.sname**

**FROM student s**

**JOIN enroll e ON s.sid = e.sid**

**GROUP BY s.sid, s.sname**

**HAVING COUNT(\*) = (**

**SELECT MAX(cnt)**

**FROM (**

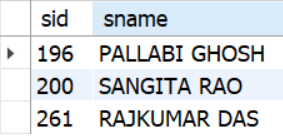
**SELECT COUNT(\*) AS cnt**

**FROM enroll**

**GROUP BY sid**

**) AS sub**

**);**



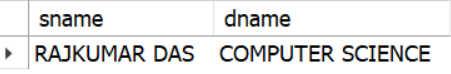
1. **List the name of student and department name whose sid=’261’**

**SELECT s.sname, m.dname**

**FROM student s**

**JOIN major m ON s.sid = m.sid**

**WHERE s.sid = 261;**



**5. List the names of students who are taking both a Computer Science and Mathematics course.**

**SELECT s.sname**

**FROM student s**

**WHERE s.sid IN (**

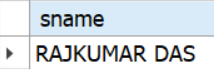
**SELECT sid FROM enroll WHERE dname = 'COMPUTER SCIENCE'**

**)**

**AND s.sid IN (**

**SELECT sid FROM enroll WHERE dname = 'MATH'**

**);**



**Question 3:**

**EMPLOYEE (EMP\_ID, EMP\_NAME, SALARY)**

**DEPARTMENT (DEPT\_ID, DEPT\_NAME, MGR\_ID, FLOOR\_NO)**

**Write down the following SQL queries:**

**1. Display the names of all employees who work on 10th floor and earn less than Rs. 10,000.**

**2.List names of all managers who manage two or more departments on the same floor.**

**3. Give 10% hike to the salary to every employee who works in sales department and display the appropriate record.**

**4.Display the names of employee who earn more than every employee of IT department.**

**5.Print the names of employees who earn more than Rs. 50,000 and work in either sales or marketing departments.**

**Question 4:**

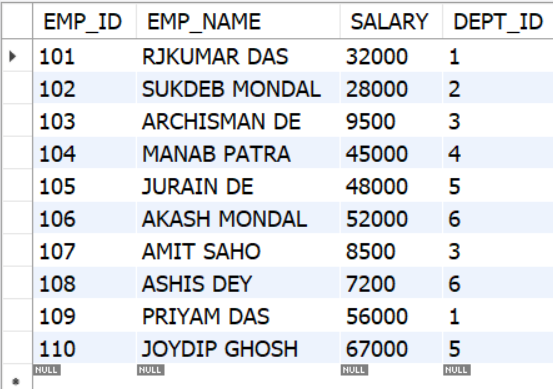
**DEPOSIT (BR\_NAME, ACC\_NO, CUST\_NAME, BALANCE)**

**CUSTOMER (CUST\_NAME, STREET, CITY)**

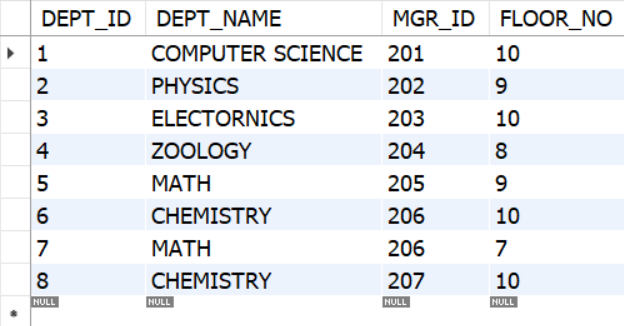
**BRANCH (BR\_NAME, ASSETS, CITY)**

**BORROW (BR\_NAME, LOAN\_NO, CUST\_NAME, AMOUNT)**

**EMPLOYEE (EMP\_ID, EMP\_NAME, SALARY)**



**DEPARTMENT (DEPT\_ID, DEPT\_NAME, MGR\_ID, FLOOR\_NO)**



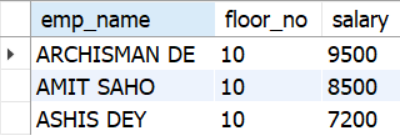
1. **Display the names of all employees who work on 10th floor and earn less than Rs. 10,000.**

**select e.emp\_name,d.floor\_no,e.salary from employee as e**

**join department as d**

**on e.dept\_id = d.dept\_id**

**where d.floor\_no = 10 and e.salary < 10000;**



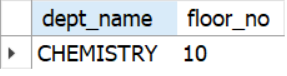
1. **List names of all managers who manage two or more departments on the same floor.**

**SELECT dept\_name, floor\_no**

**FROM department**

**GROUP BY dept\_name, floor\_no**

**HAVING COUNT(\*) = 2;**



1. **Give 10% hike to the salary to every employee who works in sales department and display the appropriate record.**

**SELECT e.emp\_name, e.salary, e.salary \* 1.10 AS hiked\_salary**

**FROM employee e**

**JOIN department d ON e.dept\_id = d.dept\_id**

**WHERE d.dept\_name = 'COMPUTER SCIENCE';**

**UPDATE employee**

**SET salary = salary \* 1.10**

**WHERE dept\_id IN (**

**SELECT dept\_id FROM department WHERE dept\_name = 'COMPUTER SCIENCE'**

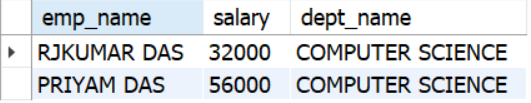
**);**

**SELECT e.emp\_name, e.salary,d.dept\_name FROM employee as e**

**join department as d**

**on e.dept\_id = d.dept\_id**

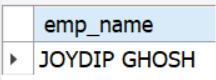
**where d.dept\_id = 1;**



1. **Display the names of employee who earn more than every employee of IT department.**

**select emp\_name from employee**

**where salary = (select max(salary) from employee);**



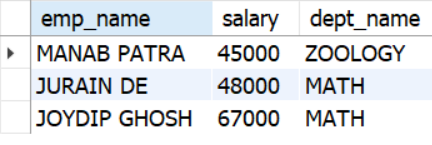
1. **Print the names of employees who earn more than Rs. 50,000 and work in either sales or marketing departments.**

**SELECT e.emp\_name,e.salary,d.dept\_name**

**FROM employee e**

**JOIN department d ON e.dept\_id = d.dept\_id**

**WHERE e.salary > 40000 AND d.dept\_name IN ('ZOOLOGY', 'MATH');**



**Question 4:**

**DEPOSIT (BR\_NAME, ACC\_NO, CUST\_NAME, BALANCE)**

**CUSTOMER (CUST\_NAME, STREET, CITY)**

**BRANCH (BR\_NAME, ASSETS, CITY)**

**BORROW (BR\_NAME, LOAN\_NO, CUST\_NAME, AMOUNT)**

**Write down the following SQL queries:**

**1. Find all customers who have either account or loan or both in SBI, Belur Math Branch**

**2. Find customers with names and cities in which they live who have taken loan from**

**SBI, Kolkata Branch.**

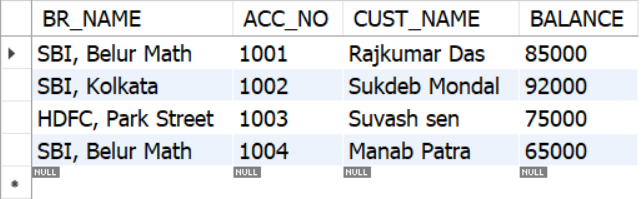
**3. Find customers name and amount including name of the branch for loan over**

**Rs.50,000.**

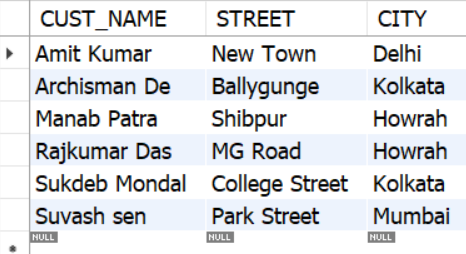
**4. Find customer having highest balance.**

**5. Find branch having highest loan.**

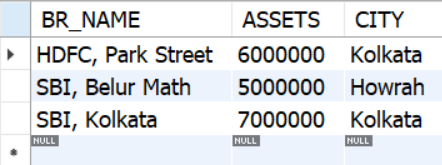
**DEPOSIT (BR\_NAME, ACC\_NO, CUST\_NAME, BALANCE)**



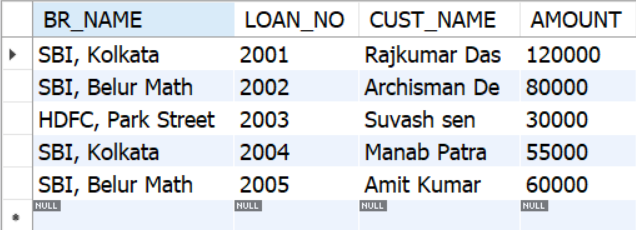
**CUSTOMER (CUST\_NAME, STREET, CITY)**



**BRANCH (BR\_NAME, ASSETS, CITY)**



**BORROW (BR\_NAME, LOAN\_NO, CUST\_NAME, AMOUNT)**



1. **Find all customers who have either account or loan or both in SBI, Belur Math Branch**

**SELECT DISTINCT cust\_name**

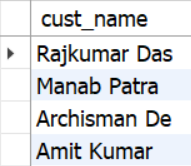
**FROM (**

**SELECT cust\_name FROM deposit WHERE br\_name = 'SBI, Belur Math'**

**UNION**

**SELECT cust\_name FROM borrow WHERE br\_name = 'SBI, Belur Math'**

**) AS result;**



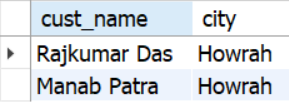
**2. Find customers with names and cities in which they live who have taken loan from SBI, Kolkata Branch.**

**SELECT c.cust\_name, c.city**

**FROM customer c**

**JOIN borrow b ON c.cust\_name = b.cust\_name**

**WHERE b.br\_name = 'SBI, Kolkata';**



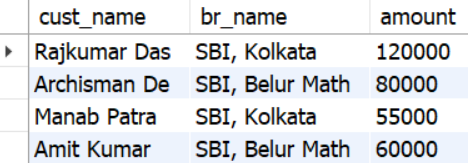
**3. Find customers name and amount including name of the branch for loan over**

**Rs.50,000.**

**SELECT b.cust\_name, b.br\_name, b.amount**

**FROM borrow b**

**WHERE b.amount > 50000;**

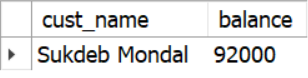


1. **Find customer having highest balance.**

**SELECT cust\_name, balance**

**FROM deposit**

**WHERE balance = (SELECT MAX(balance) FROM deposit);**



1. **Find branch having highest loan.**

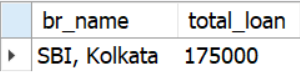
**SELECT br\_name, SUM(amount) AS total\_loan**

**FROM borrow**

**GROUP BY br\_name**

**ORDER BY total\_loan DESC**

**LIMIT 1;**



**Question 5:**

**STUDENT(SNUM: INTEGER, SNAME: STRING, MAJOR: STRING,LEVEL: STRING, AGE: INTEGER)**

**CLASS(NAME: STRING, MEETS AT: STRING, ROOM: STRING,FID: INTEGER)**

**ENROLLED(SNUM: INTEGER, CNAME: STRING)**

**FACULTY(FID: INTEGER, FNAME: STRING, DEPTID: INTEGER)**

**The meaning of these relations is straightforward; for example, Enrolled has one record per**

**student-class pair such that the student is enrolled in the class.**

**Write the following queries in SQL. No duplicates should be printed in any of the answers.**

1. **Find the names of faculty members for whom the combined enrollment of the courses that they teach is less than five.**

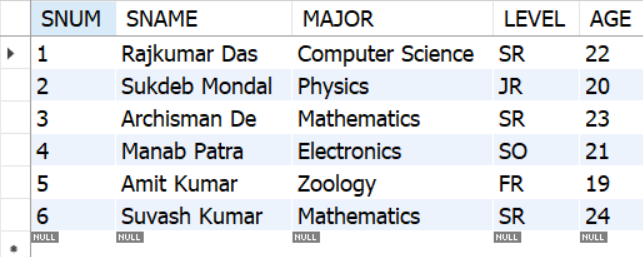
**2. For each level, print the level and the average age of students for that level.**

**3. For all levels except JR, print the level and the average age of students for that level.**

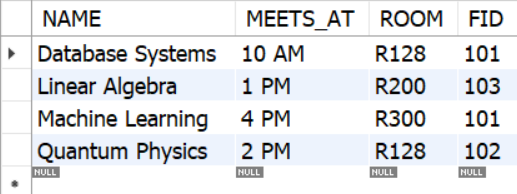
**4. For each faculty member that has taught classes only in room R128, print the faculty member's name and the total number of classes she or he has taught.**

**5. Find the names of students enrolled in the maximum number of classes.**

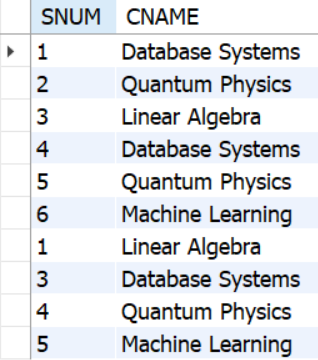
**STUDENT(SNUM: INTEGER, SNAME: STRING, MAJOR: STRING, LEVEL: STRING, AGE: INTEGER)**



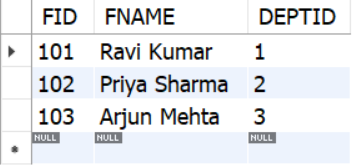
**CLASS(NAME: STRING, MEETS AT: STRING, ROOM: STRING,FID: INTEGER)**



**ENROLLED(SNUM: INTEGER, CNAME: STRING)**



**FACULTY(FID: INTEGER, FNAME: STRING, DEPTID: INTEGER)**



1. **Find the names of faculty members for whom the combined enrollment of the courses that they teach is less than five.**

**SELECT f.fname**

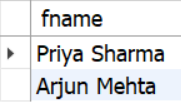
**FROM faculty f**

**JOIN class c ON f.fid = c.fid**

**LEFT JOIN enrolled e ON c.name = e.cname**

**GROUP BY f.fid, f.fname**

**HAVING COUNT(e.snum) < 5;**

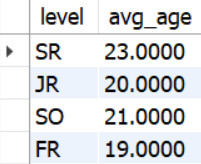


1. **For each level, print the level and the average age of students for that level.**

**SELECT level, AVG(age) AS avg\_age**

**FROM student**

**GROUP BY level;**



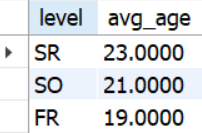
1. **For all levels except JR, print the level and the average age of students for that level.**

**SELECT level, AVG(age) AS avg\_age**

**FROM student**

**WHERE level <> 'JR'**

**GROUP BY level;**



1. **For each faculty member that has taught classes only in room R128, print the faculty member's name and the total number of classes she or he has taught.**

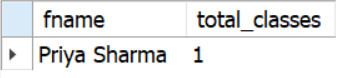
**SELECT f.fname, COUNT(\*) AS total\_classes**

**FROM faculty f**

**JOIN class c ON f.fid = c.fid**

**GROUP BY f.fid, f.fname**

**HAVING COUNT(DISTINCT c.room) = 1 AND MAX(c.room) = 'R128';**



**5. Find the names of students enrolled in the maximum number of classes.**

**SELECT s.sname**

**FROM student s**

**JOIN enrolled e ON s.snum = e.snum**

**GROUP BY s.snum, s.sname**

**HAVING COUNT(e.cname) = (**

**SELECT MAX(class\_count)**

**FROM (**

**SELECT COUNT(\*) AS class\_count**

**FROM enrolled**

**GROUP BY snum**

**) AS sub**

**);**

