**Task2. Nested and Correlated Queries**

Consider the following database tables and write the solution for the given queries Using ANY, ALL, IN, EXISTS, NOT EXISTS, UNION, INTERSECT.

**Tables: Employee**(eid, ename, salary, doj, comm,did) **Department**(did, dname, location)

**Q15)** **Write a query to Find employees who earn more than the average salary:**

SELECT employee\_id, employee\_name, salary

FROM employees

WHERE salary > (SELECT AVG(salary) FROM employees);

**Q16).** **Write a query to Find departments that have more employees than the average department:**

SELECT d\_id

FROM employees

GROUP BY d\_id

HAVING COUNT(\*) > (SELECT AVG(employee\_count)

FROM (SELECT COUNT(\*) AS employee\_count

FROM employees

GROUP By d\_id) AS dept\_counts);

**Q16).** **Write a query to Find employees who earn more than anyone in department 10:**

SELECT employee\_id, employee\_name, salary

FROM employees

WHERE salary > ANY (SELECT salary FROM employees WHERE department\_id = 10);

**Q17).Write a query to Find employees whose salary is greater than the salary of all employees in department 20:**

SELECT employee\_id, employee\_name, salary

FROM employees

WHERE salary > ALL (SELECT salary FROM employees WHERE department\_id = 20);

**Q18).** **Write a query to Find employees who work in departments 10, 20, or 30:**

SELECT employee\_id, employee\_name, department\_id

FROM employees

WHERE department\_id IN (10, 20, 30);

**Q19).** **Write a query to Find departments that have at least one employee with a salary greater than 50,000:**

SELECT department\_id

FROM departments d

WHERE EXISTS (SELECT 1

FROM employees e

WHERE e.department\_id = d.department\_id

AND e.salary > 50000);

**Q20).** **Write a query to Find departments that do not have any employees with a salary greater than 50,000:**

SELECT department\_id

FROM departments d

WHERE NOT EXISTS (SELECT 1

FROM employees e

WHERE e.department\_id = d.department\_id

AND e.salary > 50000);

**Task3: Nested and Correlated Queries for the University System**

Consider the following database tables and write the solution for the given queries Using ANY, ALL, IN, EXISTS, NOT EXISTS, UNION, INTERSECT.

## Sample Data for University System

Student(sid, sname, ccode, dob, address)

Course(ccode, cname,did, fees)

Department(did, dname, location)

Faculty(fid, fname, sal, designation, doj, did)

**Student Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **sid** | **sname** | **ccode** | **dob** | **address** |
| **1** | **Alice** | **CS101** | **2002-01-01** | **Hyderabad** |
| **2** | **Bob** | **CS101** | **2003-02-02** | **Chennai** |
| **3** | **Charlie** | **CS102** | **2004-03-03** | **Bangalore** |
| **4** | **David** | **CS102** | **2005-04-04** | **Pune** |
| **5** | **Eve** | **CS102** | **2006-05-05** | **Delhi** |

**Course Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **ccode** | **cname** | **did** | **fees** |
| **CS101** | **Introduction to Computer Science** | **DEPT001** | **10000** |
| **CS102** | **Data Structures and Algorithms** | **DEPT001** | **12000** |
| **CS103** | **Object-Oriented Programming** | **DEPT002** | **15000** |

**Department Table**

|  |  |  |
| --- | --- | --- |
| **Did** | **dname** | **location** |
| **DEPT001** | **Computer Science** | **Hyderabad** |
| **DEPT002** | **Information Technology** | **Chennai** |

**Faculty Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **fid** | **fname** | **sal** | **designation** | **doj** | **did** |
| **101** | **Professor A** | **100000** | **Professor** | **2010-01-01** | **DEPT001** |
| **102** | **Professor B** | **90000** | **Associate Professor** | **2011-02-02** | **DEPT001** |
| **103** | **Professor C** | **80000** | **Assistant Professor** | **2012-03-03** | **DEPT002** |

## Nested Queries for the University System

### Q21: Find students who have enrolled in courses offered by a specific department.

**Expected Output:**

* Alice
* Bob
* Charlie
* David
* Eve

SELECT sname

FROM Student

WHERE ccode IN (

SELECT ccode

FROM Course

WHERE did = 'DEPT001'

);

### Q22: Find the names of faculty members who teach courses that have a fee greater than the average course fee.

**Expected Output:**

* Professor A
* Professor B

SELECT fname

FROM Faculty

WHERE fid IN (

SELECT fid

FROM Course

WHERE fees > (SELECT AVG(fees) FROM Course)

);

### 

### 

### 

### Q23: Find the names of students who have enrolled in all courses offered by a specific department.

**Expected Output:**

* Charlie
* David
* Eve

SELECT sname

FROM Student

WHERE NOT EXISTS (

SELECT ccode

FROM Course

WHERE did = 'DEPT001' AND ccode NOT IN (

SELECT ccode

FROM Student

WHERE sname = 'Student1'

)

);

### Q24: Find the names of departments where the average faculty salary is higher than the average student's course fee.

**Expected Output:**

* Computer Science

SELECT dname

FROM Department

WHERE did IN (

SELECT did

FROM Faculty

GROUP BY did

HAVING AVG(sal) > (

SELECT AVG(fees)

FROM Course

)

);

### Q25 Find the names of students who have enrolled in courses taught by faculty members with a salary greater than the average faculty salary.

**Expected Output:**

* **Alice**
* **Bob**
* **Charlie**
* **David**
* **Eve**

SELECT sname

FROM Student

WHERE ccode IN (

SELECT ccode

FROM Course

WHERE fid IN (

SELECT fid

FROM Faculty

WHERE sal > (SELECT AVG(sal) FROM Faculty)

)

);