**Part 1: Simple Nested Queries**

**Problem 1: Find the names of all employees who work in departments located in 'New York'.**

sql

Copy code

SELECT EmployeeName

FROM Employees

WHERE DepartmentID IN (

SELECT DepartmentID

FROM Departments

WHERE Location = 'New York'

);

**Problem 2: List the employees who are not in the 'Marketing' department.**

sql

Copy code

SELECT EmployeeName

FROM Employees

WHERE DepartmentID NOT IN (

SELECT DepartmentID

FROM Departments

WHERE DepartmentName = 'Marketing'

);

**Problem 3: Find employees who earn more than the highest salary in the 'IT' department.**

sql

Copy code

SELECT EmployeeName

FROM Employees

WHERE Salary > (

SELECT MAX(Salary)

FROM Employees

WHERE DepartmentID = (

SELECT DepartmentID

FROM Departments

WHERE DepartmentName = 'IT'

)

);

**Problem 4: Find the employees who earn more than everyone in the 'Finance' department.**

sql

Copy code

SELECT EmployeeName

FROM Employees

WHERE Salary > ALL (

SELECT Salary

FROM Employees

WHERE DepartmentID = (

SELECT DepartmentID

FROM Departments

WHERE DepartmentName = 'Finance'

)

);

**Part 2: Aggregate Functions with GROUP BY and HAVING**

**Problem 1: Count the number of employees in each department.**

sql

Copy code

SELECT D.DepartmentName, COUNT(E.EmployeeID) AS EmployeeCount

FROM Departments D

LEFT JOIN Employees E ON D.DepartmentID = E.DepartmentID

GROUP BY D.DepartmentName;

**Problem 2: Find the highest and lowest salary in each department.**

sql

Copy code

SELECT D.DepartmentName, MAX(E.Salary) AS HighestSalary, MIN(E.Salary) AS LowestSalary

FROM Departments D

LEFT JOIN Employees E ON D.DepartmentID = E.DepartmentID

GROUP BY D.DepartmentName;

**Problem 3: Calculate the total salary expenditure in each department.**

sql

Copy code

SELECT D.DepartmentName, SUM(E.Salary) AS TotalSalary

FROM Departments D

LEFT JOIN Employees E ON D.DepartmentID = E.DepartmentID

GROUP BY D.DepartmentName;

**Problem 4: Find the average salary in each department, but only for departments with more than 2 employees.**

sql

Copy code

SELECT D.DepartmentName, AVG(E.Salary) AS AverageSalary

FROM Departments D

LEFT JOIN Employees E ON D.DepartmentID = E.DepartmentID

GROUP BY D.DepartmentName

HAVING COUNT(E.EmployeeID) > 2;

**Problem 5: Find the departments where the total salary expenditure exceeds 200,000.**

sql

Copy code

SELECT D.DepartmentName, SUM(E.Salary) AS TotalSalary

FROM Departments D

LEFT JOIN Employees E ON D.DepartmentID = E.DepartmentID

GROUP BY D.DepartmentName

HAVING SUM(E.Salary) > 200000;

**Problem 6: Find the department with the highest average salary.**

sql

Copy code

SELECT D.DepartmentName, AVG(E.Salary) AS AverageSalary

FROM Departments D

LEFT JOIN Employees E ON D.DepartmentID = E.DepartmentID

GROUP BY D.DepartmentName

ORDER BY AverageSalary DESC

LIMIT 1;

**Problem 7: List the employees who have the highest salary in each department.**

sql

Copy code

SELECT E.EmployeeName, D.DepartmentName, E.Salary

FROM Employees E

JOIN Departments D ON E.DepartmentID = D.DepartmentID

WHERE E.Salary = (

SELECT MAX(Salary)

FROM Employees

WHERE DepartmentID = E.DepartmentID

);

**Problem 8: Find the departments where the average salary is above 60,000 and there are more than 3 employees.**

sql

Copy code

SELECT D.DepartmentName, AVG(E.Salary) AS AverageSalary, COUNT(E.EmployeeID) AS EmployeeCount

FROM Departments D

LEFT JOIN Employees E ON D.DepartmentID = E.DepartmentID

GROUP BY D.DepartmentName

HAVING AVG(E.Salary) > 60000 AND COUNT(E.EmployeeID) > 3;

**Problem 9: Find the departments that have an average salary between 50,000 and 80,000.**

sql

Copy code

SELECT D.DepartmentName, AVG(E.Salary) AS AverageSalary

FROM Departments D

LEFT JOIN Employees E ON D.DepartmentID = E.DepartmentID

GROUP BY D.DepartmentName

HAVING AVG(E.Salary) BETWEEN 50000 AND 80000;

**Problem 10: Find the department with the most employees.**

sql

Copy code

SELECT D.DepartmentName, COUNT(E.EmployeeID) AS EmployeeCount

FROM Departments D

LEFT JOIN Employees E ON D.DepartmentID = E.DepartmentID

GROUP BY D.DepartmentName

ORDER BY EmployeeCount DESC

LIMIT 1;

**Part 3: University Merit Calculation**

**Find the first two toppers based on total merit.**

sql

Copy code

SELECT S.StudentID, S.StudentName,

(TR.TestScore \* 0.30 + AER.AnnualExamScore \* 0.70) AS FinalMeritScore

FROM Student S

JOIN TestResult TR ON S.StudentID = TR.StudentID

JOIN AnnualExamResult AER ON S.StudentID = AER.StudentID

ORDER BY FinalMeritScore DESC

LIMIT 2;

**Find the last student based on total merit.**

sql

Copy code

SELECT S.StudentID, S.StudentName,

(TR.TestScore \* 0.30 + AER.AnnualExamScore \* 0.70) AS FinalMeritScore

FROM Student S

JOIN TestResult TR ON S.StudentID = TR.StudentID

JOIN AnnualExamResult AER ON S.StudentID = AER.StudentID

ORDER BY FinalMeritScore ASC

LIMIT 1;

**Part 4: Joins and Select Variations**

**Cross Product**

sql

Copy code

SELECT \*

FROM Student, Course;

**Inner Join**

sql

Copy code

SELECT S.StudentID, S.FirstName, C.CourseName

FROM Student S

INNER JOIN Enrollment E ON S.StudentID = E.StudentID

INNER JOIN Course C ON E.CourseID = C.CourseID;

**Natural Join**

sql

Copy code

SELECT Student.StudentID, Student.FirstName, Course.CourseName

FROM Student

NATURAL JOIN Enrollment

NATURAL JOIN Course;

**Left Outer Join**

sql

Copy code

SELECT S.StudentID, S.FirstName, C.CourseName

FROM Student S

LEFT OUTER JOIN Enrollment E ON S.StudentID = E.StudentID

LEFT OUTER JOIN Course C ON E.CourseID = C.CourseID;

**Right Outer Join**

sql

Copy code

SELECT S.StudentID, S.FirstName, C.CourseName

FROM Student S

RIGHT OUTER JOIN Enrollment E ON S.StudentID = E.StudentID

RIGHT OUTER JOIN Course C ON E.CourseID = C.CourseID;

**Equi Join**

sql

Copy code

SELECT S.StudentID, S.FirstName, C.CourseName

FROM Student S

JOIN Enrollment E ON S.StudentID = E.StudentID

JOIN Course C ON E.CourseID = C.CourseID;

These queries are designed to solve all the scenarios described. Let me know if any further clarification or adjustments are needed!