

### Employee Table:

- Entity Name: Employee
- Attributes:
  - Emp\_ID (Primary Key)
  - Emp\_Name
  - Emp\_Expertise
  - Emp\_Designation

### Project Table:

- Entity Name: Project
- Attributes:
  - Project\_ID (Primary Key)
  - Project\_Name
  - Project\_Category

### Work Table:

- Entity Name: Work
- Attributes:
  - Work\_Code (Primary Key)
  - Emp\_ID (Foreign Key referencing Employee.Emp\_ID)
  - Project\_ID (Foreign Key referencing Project.Project\_ID)
  - Technology

### Doctor entity:

- Attributes
  - Dr\_id (Primary Key)
  - Dr\_Name
  - Dr\_specialization

### Patient entity:

- Attributes
  - P\_id (Primary Key)
  - P\_Name
  - P\_Disease
  - P\_Allergies
  - P\_City

### Test entity:

- Attributes
  - Test\_id (Primary Key)
  - P\_id (Foreign Key referring to Patient)
  - Test\_Name

### Treatment entity:

- Attributes
  - Treatment\_id (Primary Key)
  - P\_id (Foreign Key referring to Patient)
  - Dr\_id (Foreign Key referring to Doctor)
  - Admit\_Date

Q 1. Retrieve the projects with their names and the total years of experience of all employees working on them.

- A. `SELECT Project_Name, AVG(Emp_Expertiese) AS Total_Experience FROM Project INNER JOIN Work ON Project.Project_ID = Work.Project_ID INNER JOIN Employee ON Work.Emp_ID = Employee.Emp_ID GROUP BY Project_Name`
- B. `SELECT Project_Name, MAX(Emp_Expertiese) AS Total_Experience FROM Project INNER JOIN Work ON Project.Project_ID = Work.Project_ID INNER JOIN Employee ON Work.Emp_ID = Employee.Emp_ID GROUP BY Project_Name`
- C. `SELECT Project_Name, SUM(Emp_Expertiese) AS Total_Experience FROM Project INNER JOIN Work ON Project.Project_ID = Work.Project_ID INNER JOIN Employee ON Work.Emp_ID = Employee.Emp_ID GROUP BY Project_Name`
- D. `SELECT Project_Name, MIN(Emp_Expertiese) AS Total_Experience FROM Project INNER JOIN Work ON Project.Project_ID = Work.Project_ID INNER JOIN Employee ON Work.Emp_ID = Employee.Emp_ID GROUP BY Project_Name`

Q 2. Update the designation of employees to 'Senior Developer' if they have more than 5 years of experience, 'Junior Developer' if they have more than 2 years of experience, and 'Intern' otherwise.

- A. `UPDATE Employee SET Emp_Designation = CASE WHEN Emp_Expertiese > 2 THEN 'Intern' WHEN Emp_Expertiese > 5 THEN 'Senior Developer' ELSE 'Junior Developer' END`
- B. `UPDATE Employee SET Emp_Designation = CASE WHEN Emp_Expertiese > 2 THEN 'Intern' WHEN Emp_Expertiese > 5 THEN 'Junior Developer' ELSE 'Senior Developer' END`
- C. `UPDATE Employee SET Emp_Designation = CASE WHEN Emp_Expertiese > 5 THEN 'Senior Developer' WHEN Emp_Expertiese > 2 THEN 'Junior Developer' ELSE 'Intern' END`
- D. `UPDATE Employee SET Emp_Designation = CASE WHEN Emp_Expertiese > 5 THEN 'Junior Developer' WHEN Emp_Expertiese > 2 THEN 'Intern' ELSE 'Senior Developer' END`

Q 3. Find the doctors who have treated the most patients and show the total number of patients they have treated.

- A. `SELECT D.Dr_Name, COUNT(T.P_id) AS Total_Patients FROM Doctor D INNER JOIN Treatment T ON D.Dr_id = T.Dr_id GROUP BY D.Dr_id, D.Dr_Name HAVING COUNT(T.P_id) = (SELECT MAX(Total_Patients) FROM (SELECT D.Dr_id, COUNT(T.P_id) AS Total_Patients FROM Doctor D INNER JOIN Treatment T ON D.Dr_id = T.Dr_id GROUP BY D.Dr_id) AS Subquery);`
- B. `SELECT D.Dr_Name, COUNT(T.P_id) AS Total_Patients FROM Doctor D INNER JOIN Treatment T ON D.Dr_id = T.Dr_id GROUP BY D.Dr_id, D.Dr_Name HAVING COUNT(T.P_id) =`

(SELECT AVG(Total\_Patients) FROM (SELECT D.Dr\_id, COUNT(T.P\_id) AS Total\_Patients FROM Doctor D INNER JOIN Treatment T ON D.Dr\_id = T.Dr\_id GROUP BY D.Dr\_id) AS Subquery);

C. SELECT D.Dr\_Name, COUNT(T.P\_id) AS Total\_Patients FROM Doctor D INNER JOIN Treatment T ON D.Dr\_id = T.Dr\_id GROUP BY D.Dr\_id, D.Dr\_Name HAVING COUNT(T.P\_id) = (SELECT MIN(Total\_Patients) FROM (SELECT D.Dr\_id, COUNT(T.P\_id) AS Total\_Patients FROM Doctor D INNER JOIN Treatment T ON D.Dr\_id = T.Dr\_id GROUP BY D.Dr\_id) AS Subquery);

D. SELECT D.Dr\_Name, COUNT(T.P\_id) AS Total\_Patients FROM Doctor D INNER JOIN Treatment T ON D.Dr\_id = T.Dr\_id GROUP BY D.Dr\_id, D.Dr\_Name HAVING COUNT(T.P\_id) = (SELECT SUM(Total\_Patients) FROM (SELECT D.Dr\_id, COUNT(T.P\_id) AS Total\_Patients FROM Doctor D INNER JOIN Treatment T ON D.Dr\_id = T.Dr\_id GROUP BY D.Dr\_id) AS Subquery);

Q 4. List the doctors who have treated patients with 'Heart Disease' and 'Hypertension,' and show their names and the total number of such patients.

A. SELECT D.Dr\_Name, COUNT(DISTINCT T.P\_id) AS Total\_Patients FROM Doctor D INNER JOIN Treatment T ON D.Dr\_id = T.Dr\_id INNER JOIN Patient P ON T.P\_id = P.P\_id WHERE P.P\_Disease IN ('Heart Disease', 'Hypertension') GROUP BY D.Dr\_Name HAVING COUNT(DISTINCT T.P\_id) >= 2;

B. SELECT D.Dr\_Name, COUNT(DISTINCT T.P\_id) AS Total\_Patients FROM Doctor D INNER JOIN Treatment T ON D.Dr\_id = T.Dr\_id INNER JOIN Patient P ON T.P\_id = P.P\_id WHERE P.P\_Disease IN ('Heart Disease', 'Hypertension') GROUP BY D.Dr\_Name HAVING COUNT(DISTINCT T.P\_id) > 2;

C. SELECT D.Dr\_Name, COUNT(DISTINCT T.P\_id) AS Total\_Patients FROM Doctor D INNER JOIN Treatment T ON D.Dr\_id = T.Dr\_id INNER JOIN Patient P ON T.P\_id = P.P\_id WHERE P.P\_Disease IN ('Cancer', 'Diabetes') GROUP BY D.Dr\_Name HAVING COUNT(DISTINCT T.P\_id) > 2;

D. SELECT D.Dr\_Name, COUNT(DISTINCT T.P\_id) AS Total\_Patients FROM Doctor D INNER JOIN Treatment T ON D.Dr\_id = T.Dr\_id INNER JOIN Patient P ON T.P\_id = P.P\_id WHERE P.P\_Disease IN ('Cancer', 'Diabetes') GROUP BY D.Dr\_Name HAVING COUNT(DISTINCT T.P\_id) >= 2;

Q 5. Find the doctors who have treated patients with 'Asthma' and 'Allergies,' and show their names and the total number of such patients.

A. SELECT D.Dr\_Name, COUNT(DISTINCT T.P\_id) AS Total\_Patients FROM Doctor D INNER JOIN Treatment T ON D.Dr\_id = T.Dr\_id INNER JOIN Patient P ON T.P\_id = P.P\_id WHERE

P.P\_Disease IN ('Asthma', 'Allergies') GROUP BY D.Dr\_id, D.Dr\_Name HAVING  
COUNT(DISTINCT T.P\_id) = 1;

B. SELECT D.Dr\_Name, COUNT(DISTINCT T.P\_id) AS Total\_Patients FROM Doctor D INNER  
JOIN Treatment T ON D.Dr\_id = T.Dr\_id INNER JOIN Patient P ON T.P\_id = P.P\_id WHERE  
P.P\_Disease IN ('Asthma', 'Allergies') GROUP BY D.Dr\_id, D.Dr\_Name HAVING  
COUNT(DISTINCT T.P\_id) = 2;

C. SELECT D.Dr\_Name, COUNT(DISTINCT T.P\_id) AS Total\_Patients FROM Doctor D INNER  
JOIN Treatment T ON D.Dr\_id = T.Dr\_id INNER JOIN Patient P ON T.P\_id = P.P\_id WHERE  
P.P\_Disease IN ('Cancer', 'Hypertension') GROUP BY D.Dr\_id, D.Dr\_Name HAVING  
COUNT(DISTINCT T.P\_id) = 1;

D. SELECT D.Dr\_Name, COUNT(DISTINCT T.P\_id) AS Total\_Patients FROM Doctor D INNER  
JOIN Treatment T ON D.Dr\_id = T.Dr\_id INNER JOIN Patient P ON T.P\_id = P.P\_id WHERE  
P.P\_Disease IN ('Cancer', 'Hypertension') GROUP BY D.Dr\_id, D.Dr\_Name HAVING  
COUNT(DISTINCT T.P\_id) = 2;

Q 6. Retrieve the names of doctors who have not treated patients with 'Cancer' and have an  
experience of more than 10 years.

A. SELECT D.Dr\_Name FROM Doctor D WHERE D.Dr\_specialization IN (SELECT DISTINCT  
D.Dr\_specialization FROM Doctor D INNER JOIN Treatment T ON D.Dr\_id = T.Dr\_id INNER  
JOIN Patient P ON T.P\_id = P.P\_id WHERE P.P\_Disease = 'Cancer') AND D.Dr\_Experience <= 10;

B. SELECT D.Dr\_Name FROM Doctor D WHERE D.Dr\_specialization IN (SELECT DISTINCT  
D.Dr\_specialization FROM Doctor D INNER JOIN Treatment T ON D.Dr\_id = T.Dr\_id INNER  
JOIN Patient P ON T.P\_id = P.P\_id WHERE P.P\_Disease = 'Cancer') AND D.Dr\_Experience > 10;

C. SELECT D.Dr\_Name FROM Doctor D WHERE D.Dr\_specialization NOT IN (SELECT DISTINCT  
D.Dr\_specialization FROM Doctor D INNER JOIN Treatment T ON D.Dr\_id = T.Dr\_id INNER  
JOIN Patient P ON T.P\_id = P.P\_id WHERE P.P\_Disease = 'Cancer') AND D.Dr\_Experience <= 10;

D. SELECT D.Dr\_Name FROM Doctor D WHERE D.Dr\_specialization NOT IN (SELECT DISTINCT  
D.Dr\_specialization FROM Doctor D INNER JOIN Treatment T ON D.Dr\_id = T.Dr\_id INNER  
JOIN Patient P ON T.P\_id = P.P\_id WHERE P.P\_Disease = 'Cancer') AND D.Dr\_Experience > 10;

Q 7. Calculate the average age of patients who are treated by doctors specializing in 'Pediatrics.'

A. SELECT AVG(YEAR(CURRENT\_DATE) - YEAR(P.P\_Birthdate)) AS Avg\_Age FROM Patient P  
INNER JOIN Treatment T ON P.P\_id = T.P\_id INNER JOIN Doctor D ON T.Dr\_id = D.Dr\_id  
WHERE D.Dr\_specialization = 'Oncology';

B. SELECT AVG(YEAR(CURRENT\_DATE) - YEAR(P.P\_Birthdate)) AS Avg\_Age FROM Patient P  
INNER JOIN Treatment T ON P.P\_id = T.P\_id INNER JOIN Doctor D ON T.Dr\_id = D.Dr\_id  
WHERE D.Dr\_specialization = 'Pediatrics';

C. SELECT AVG(YEAR(CURRENT\_DATE) - YEAR(P.P\_Birthdate)) AS Avg\_Age FROM Patient P  
INNER JOIN Treatment T ON P.P\_id = T.P\_id INNER JOIN Doctor D ON T.Dr\_id = D.Dr\_id  
WHERE D.Dr\_specialization = 'Cardiology';

D. SELECT AVG(YEAR(CURRENT\_DATE) - YEAR(P.P\_Birthdate)) AS Avg\_Age FROM Patient P  
INNER JOIN Treatment T ON P.P\_id = T.P\_id INNER JOIN Doctor D ON T.Dr\_id = D.Dr\_id  
WHERE D.Dr\_specialization = 'Surgery';

Q 8. Retrieve the names of employees and their corresponding designations, but if the designation is 'Manager,' display it as 'MGR,' 'Senior Developer' as 'Sr. Dev,' and 'Junior Developer' as 'Jr. Dev.'

A. SELECT Emp\_Name, CASE WHEN Emp\_Designation = 'Manager' THEN 'MGR' WHEN  
Emp\_Designation = 'Junior Developer' THEN 'Jr. Dev' WHEN Emp\_Designation = 'Senior  
Developer' THEN 'Sr. Dev' ELSE Emp\_Designation END FROM Employee

B. SELECT Emp\_Name, CASE Emp\_Designation WHEN 'Manager' THEN 'MGR' WHEN 'Senior  
Developer' THEN 'Sr. Dev' WHEN 'Junior Developer' THEN 'Jr. Dev' ELSE Emp\_Designation  
END FROM Employee

C. SELECT Emp\_Name, CASE WHEN Emp\_Designation = 'Manager' THEN 'MGR' WHEN  
Emp\_Designation = 'Senior Developer' THEN 'Sr. Dev' WHEN Emp\_Designation = 'Junior  
Developer' THEN 'Jr. Dev' ELSE Emp\_Designation END FROM Employee

D. SELECT Emp\_Name, CASE Emp\_Designation WHEN 'Manager' THEN 'MGR' WHEN 'Junior  
Developer' THEN 'Jr. Dev' WHEN 'Senior Developer' THEN 'Sr. Dev' ELSE Emp\_Designation  
END FROM Employee

Q 9. List the employees who have worked on projects in multiple project categories and classify them as 'Versatile' employees.

A. SELECT Emp\_Name FROM Employee WHERE Emp\_ID IN (SELECT Emp\_ID FROM Work  
GROUP BY Emp\_ID HAVING COUNT(DISTINCT Project\_id) > 1)

B. SELECT Emp\_Name FROM Employee WHERE Emp\_ID IN (SELECT Emp\_ID FROM Work  
GROUP BY Emp\_ID HAVING COUNT(Project\_id) > 1)

- C. `SELECT Emp_Name FROM Employee WHERE Emp_ID IN (SELECT Emp_ID FROM Work GROUP BY Emp_ID HAVING COUNT(DISTINCT Project_id) = 1)`
- D. `SELECT Emp_Name FROM Employee WHERE Emp_ID IN (SELECT Emp_ID FROM Work GROUP BY Emp_ID HAVING COUNT(Project_id) = 1)`

Q 10. Find the doctors who have treated patients from 'Chicago' and have performed at least one 'CT Scan' test.

- A. `SELECT D.Dr_Name FROM Doctor D INNER JOIN Treatment T ON D.Dr_id = T.Dr_id INNER JOIN Patient P ON T.P_id = P.P_id INNER JOIN Test TS ON P.P_id = TS.P_id WHERE P.P_City = 'New York City' AND TS.Test_Name = 'MRI';`
- B. `SELECT D.Dr_Name FROM Doctor D INNER JOIN Treatment T ON D.Dr_id = T.Dr_id INNER JOIN Patient P ON T.P_id = P.P_id INNER JOIN Test TS ON P.P_id = TS.P_id WHERE P.P_City = 'Chicago' AND TS.Test_Name = 'CT Scan';`
- C. `SELECT D.Dr_Name FROM Doctor D INNER JOIN Treatment T ON D.Dr_id = T.Dr_id INNER JOIN Patient P ON T.P_id = P.P_id INNER JOIN Test TS ON P.P_id = TS.P_id WHERE P.P_City = 'New York City' AND TS.Test_Name = 'CT Scan';`
- D. `SELECT D.Dr_Name FROM Doctor D INNER JOIN Treatment T ON D.Dr_id = T.Dr_id INNER JOIN Patient P ON T.P_id = P.P_id INNER JOIN Test TS ON P.P_id = TS.P_id WHERE P.P_City = 'Chicago' AND TS.Test_Name = 'MRI';`

Q 11. Update the salaries of employees by adding 10% for employees with more than 3 years of experience, 15% for employees with more than 5 years, and 20% for employees with more than 10 years.

- A. `UPDATE Employee SET Emp_Salary = Emp_Salary * 1.2 WHERE Emp_Expertiese > 3`
- B. `UPDATE Employee SET Emp_Salary = Emp_Salary * 1.15 WHERE Emp_Expertiese > 5`
- C. `UPDATE Employee SET Emp_Salary = Emp_Salary * 1.2 WHERE Emp_Expertiese > 10`
- D. `UPDATE Employee SET Emp_Salary = Emp_Salary * 1.1 WHERE Emp_Expertiese > 3`

Q 12. List the projects and their categories, but if the project category is 'Development,' display it as 'DEV,' 'Research' as 'RSR,' and 'Management' as 'MGT.'

- A. SELECT Project\_Name, CASE Project\_Categoery WHEN 'Development' THEN 'DEV' WHEN 'Management' THEN 'MGT' WHEN 'Research' THEN 'RSR' ELSE Project\_Categoery END FROM Project
- B. SELECT Project\_Name, CASE Project\_Categoery WHEN 'Development' THEN 'MGT' WHEN 'Research' THEN 'DEV' WHEN 'Management' THEN 'RSR' ELSE Project\_Categoery END FROM Project
- C. SELECT Project\_Name, CASE WHEN Project\_Categoery = 'Development' THEN 'DEV' WHEN Project\_Categoery = 'Research' THEN 'RSR' WHEN Project\_Categoery = 'Management' THEN 'MGT' ELSE Project\_Categoery END FROM Project
- D. SELECT Project\_Name, CASE Project\_Categoery WHEN 'Development' THEN 'DEV' WHEN 'Research' THEN 'RSR' WHEN 'Management' THEN 'MGT' ELSE Project\_Categoery END FROM Project

Q 13. Retrieve the projects and their categories with the count of employees working on each project.

- A. SELECT Project\_Name, Project\_Categoery, COUNT(Emp\_ID) AS Employee\_Count FROM Project LEFT JOIN Work ON Project.Project\_ID = Work.Project\_ID GROUP BY Project\_Name, Project\_Categoery
- B. SELECT Project\_Name, Project\_Categoery, MAX(Emp\_ID) AS Employee\_Count FROM Project RIGHT JOIN Work ON Project.Project\_ID = Work.Project\_ID GROUP BY Project\_Name, Project\_Categoery
- C. SELECT Project\_Name, Project\_Categoery, SUM(Emp\_ID) AS Employee\_Count FROM Project INNER JOIN Work ON Project.Project\_ID = Work.Project\_ID GROUP BY Project\_Name, Project\_Categoery
- D. SELECT Project\_Name, Project\_Categoery, AVG(Emp\_ID) AS Employee\_Count FROM Project LEFT JOIN Work ON Project.Project\_ID = Work.Project\_ID GROUP BY Project\_Name, Project\_Categoery

Q 14. List the projects that have more than two employees assigned to them.

- A. SELECT Project\_Name FROM Project WHERE Project\_ID IN (SELECT Project\_ID FROM Work GROUP BY Project\_id HAVING COUNT(Emp\_ID) = 2)
- B. SELECT Project\_Name FROM Project WHERE Project\_ID IN (SELECT Project\_ID FROM Work GROUP BY Project\_id HAVING COUNT(Emp\_ID) < 2)

- C. `SELECT Project_Name FROM Project WHERE Project_ID IN (SELECT Project_ID FROM Work GROUP BY Project_id HAVING COUNT(Emp_ID) > 2)`
- D. `SELECT Project_Name FROM Project WHERE Project_ID IN (SELECT Project_ID FROM Work GROUP BY Project_id HAVING COUNT(Emp_ID) = 3)`

Q 15. List all employees who have worked on projects in the 'Development' category.

- A. `SELECT Emp_Name FROM Employee WHERE Emp_ID IN (SELECT Emp_ID FROM Work WHERE Project_id = (SELECT Project_ID FROM Project WHERE Project_Categoery <> 'Development'))`
- B. `SELECT Emp_Name FROM Employee WHERE Emp_ID IN (SELECT Emp_ID FROM Work WHERE Project_id IN (SELECT Project_ID FROM Project WHERE Project_Categoery <> 'Development'))`
- C. `SELECT Emp_Name FROM Employee WHERE Emp_ID IN (SELECT Emp_ID FROM Work WHERE Project_id = (SELECT Project_ID FROM Project WHERE Project_Categoery = 'Development'))`
- D. `SELECT Emp_Name FROM Employee WHERE Emp_ID IN (SELECT Emp_ID FROM Work WHERE Project_id IN (SELECT Project_ID FROM Project WHERE Project_Categoery = 'Development'))`

Q 16. Identify employees who have worked on projects with a budget exceeding \$75,000, and if they have less than 5 years of experience, grant them a bonus of 5% of the project's budget.

- A. `UPDATE Employee SET Emp_Bonus = Project_Budget * 0.05 FROM Employee INNER JOIN Work ON Employee.Emp_ID = Work.Emp_ID INNER JOIN Project ON Work.Project_ID = Project.Project_ID WHERE Project_Budget >= 75000 AND Emp_Expertiese >= 5`
- B. `UPDATE Employee SET Emp_Bonus = Project_Budget * 0.05 FROM Employee INNER JOIN Work ON Employee.Emp_ID = Work.Emp_ID INNER JOIN Project ON Work.Project_ID = Project.Project_ID WHERE Project_Budget >= 75000 AND Emp_Expertiese < 5`
- C. `UPDATE Employee SET Emp_Bonus = Project_Budget * 0.05 FROM Employee INNER JOIN Work ON Employee.Emp_ID = Work.Emp_ID INNER JOIN Project ON Work.Project_ID = Project.Project_ID WHERE Project_Budget > 75000 AND Emp_Expertiese >= 5`
- D. `UPDATE Employee SET Emp_Bonus = Project_Budget * 0.05 FROM Employee INNER JOIN Work ON Employee.Emp_ID = Work.Emp_ID INNER JOIN Project ON Work.Project_ID = Project.Project_ID WHERE Project_Budget > 75000 AND Emp_Expertiese < 5`



Q 17. Retrieve the names of patients who are allergic to 'Penicillin' and are treated by doctors specializing in 'Allergy.'

- A. `SELECT P.P_Name FROM Patient P WHERE P.P_Allergies LIKE '%Penicillin%' AND P.P_id IN (SELECT T.P_id FROM Treatment T INNER JOIN Doctor D ON T.Dr_id = D.Dr_id WHERE D.Dr_specialization = 'Allergy')`
- B. `SELECT P.P_Name FROM Patient P INNER JOIN Treatment T ON P.P_id = T.P_id INNER JOIN Doctor D ON T.Dr_id = D.Dr_id WHERE P.P_Allergies LIKE '%Penicillin%' AND D.Dr_specialization = 'Allergy'`
- C. `SELECT P.P_Name FROM Patient P INNER JOIN Treatment T ON P.P_id = T.P_id INNER JOIN Doctor D ON T.Dr_id = D.Dr_id WHERE P.P_Allergies LIKE '%Penicillin%' AND D.Dr_specialization = 'Allergy'`
- D. `SELECT P.P_Name FROM Patient P INNER JOIN Treatment T ON P.P_id = T.P_id INNER JOIN Doctor D ON T.Dr_id = D.Dr_id WHERE P.P_Allergies LIKE '%Penicillin%' AND P.P_id IN (SELECT T.P_id FROM Treatment T INNER JOIN Doctor D ON T.Dr_id = D.Dr_id WHERE D.Dr_specialization = 'Allergy')`

Q 18. Find the doctors who have treated patients from 'San Francisco' and have had at least one patient from 'New York City.'

- A. `SELECT D.Dr_Name FROM Doctor D INNER JOIN Treatment T1 ON D.Dr_id = T1.Dr_id INNER JOIN Patient P1 ON T1.P_id = P1.P_id WHERE P1.P_City = 'San Francisco' AND EXISTS (SELECT 1 FROM Treatment T2 INNER JOIN Patient P2 ON T2.P_id = P2.P_id WHERE T2.Dr_id = D.Dr_id AND P2.P_City = 'New York City');`
- B. `SELECT D.Dr_Name FROM Doctor D INNER JOIN Treatment T1 ON D.Dr_id = T1.Dr_id INNER JOIN Patient P1 ON T1.P_id = P1.P_id WHERE P1.P_City = 'San Francisco' AND NOT EXISTS (SELECT 1 FROM Treatment T2 INNER JOIN Patient P2 ON T2.P_id = P2.P_id WHERE T2.Dr_id = D.Dr_id AND P2.P_City = 'Los Angeles');`
- C. `SELECT D.Dr_Name FROM Doctor D INNER JOIN Treatment T1 ON D.Dr_id = T1.Dr_id INNER JOIN Patient P1 ON T1.P_id = P1.P_id WHERE P1.P_City = 'New York City' AND EXISTS (SELECT 1 FROM Treatment T2 INNER JOIN Patient P2 ON T2.P_id = P2.P_id WHERE T2.Dr_id = D.Dr_id AND P2.P_City = 'San Francisco');`
- D. `SELECT D.Dr_Name FROM Doctor D INNER JOIN Treatment T1 ON D.Dr_id = T1.Dr_id INNER JOIN Patient P1 ON T1.P_id = P1.P_id WHERE P1.P_City = 'Los Angeles' AND NOT EXISTS (SELECT 1 FROM Treatment T2 INNER JOIN Patient P2 ON T2.P_id = P2.P_id WHERE T2.Dr_id = D.Dr_id AND P2.P_City = 'Chicago');`

Q 19. List the patients who have been treated by doctors from the same city as the patient and show the names of the doctors.

- A. `SELECT P.P_Name, D.Dr_Name FROM Patient P INNER JOIN Treatment T ON P.P_id = T.P_id INNER JOIN Doctor D ON T.Dr_id = D.Dr_id WHERE P.P_City = D.Dr_City AND P.P_id = D.Dr_id;`
- B. `SELECT P.P_Name, D.Dr_Name FROM Patient P INNER JOIN Treatment T ON P.P_id = T.P_id INNER JOIN Doctor D ON T.Dr_id = D.Dr_id WHERE P.P_City <> D.Dr_City AND P.P_id != D.Dr_id;`
- C. `SELECT P.P_Name, D.Dr_Name FROM Patient P INNER JOIN Treatment T ON P.P_id = T.P_id INNER JOIN Doctor D ON T.Dr_id = D.Dr_id WHERE P.P_City <> D.Dr_City AND P.P_id = D.Dr_id;`
- D. `SELECT P.P_Name, D.Dr_Name FROM Patient P INNER JOIN Treatment T ON P.P_id = T.P_id INNER JOIN Doctor D ON T.Dr_id = D.Dr_id WHERE P.P_City = D.Dr_City AND P.P_id != D.Dr_id;`

Q 20. Find the projects with the highest total budget. If multiple projects have the same highest budget, choose the one with the most employees.

- A. `SELECT Project_Name FROM Project ORDER BY Project_Budget DESC, (SELECT COUNT(*) FROM Work WHERE Project.Project_ID = Work.Project_ID) DESC LIMIT 1`
- B. `SELECT Project_Name FROM Project ORDER BY Project_Budget DESC, (SELECT COUNT(*) FROM Work WHERE Project.Project_ID = Work.Project_ID) ASC LIMIT 1`
- C. `SELECT Project_Name FROM Project ORDER BY Project_Budget ASC, (SELECT COUNT(*) FROM Work WHERE Project.Project_ID = Work.Project_ID) DESC LIMIT 1`
- D. `SELECT Project_Name FROM Project ORDER BY Project_Budget ASC, (SELECT COUNT(*) FROM Work WHERE Project.Project_ID = Work.Project_ID) ASC LIMIT 1`

Q 21. Retrieve the names of doctors who have treated the same patient more than once, and show the patient's name and the number of treatments.

- A. `SELECT D.Dr_Name, P.P_Name, COUNT(T.Treatment_id) AS Total_Treatments FROM Doctor D INNER JOIN Treatment T ON D.Dr_id = T.Dr_id INNER JOIN Patient P ON T.P_id =`

P.P\_id WHERE T.P\_id IN (SELECT T.P\_id FROM Treatment T GROUP BY T.P\_id HAVING COUNT(Treatment\_id) = 1) GROUP BY D.Dr\_Name, P.P\_Name;

B. SELECT D.Dr\_Name, P.P\_Name, COUNT(T.Treatment\_id) AS Total\_Treatments FROM Doctor D INNER JOIN Treatment T ON D.Dr\_id = T.Dr\_id INNER JOIN Patient P ON T.P\_id = P.P\_id WHERE T.P\_id IN (SELECT T.P\_id FROM Treatment T GROUP BY T.P\_id HAVING COUNT(T.Treatment\_id) > 1) GROUP BY D.Dr\_Name, P.P\_Name;

C. SELECT D.Dr\_Name, P.P\_Name, COUNT(T.Treatment\_id) AS Total\_Treatments FROM Doctor D INNER JOIN Treatment T ON D.Dr\_id = T.Dr\_id INNER JOIN Patient P ON T.P\_id = P.P\_id WHERE T.P\_id IN (SELECT T.P\_id FROM Treatment T GROUP BY T.P\_id HAVING COUNT(Treatment\_id) = 2) GROUP BY D.Dr\_Name, P.P\_Name;

D. SELECT D.Dr\_Name, P.P\_Name, COUNT(T.Treatment\_id) AS Total\_Treatments FROM Doctor D INNER JOIN Treatment T ON D.Dr\_id = T.Dr\_id INNER JOIN Patient P ON T.P\_id = P.P\_id WHERE T.P\_id NOT IN (SELECT T.P\_id FROM Treatment T GROUP BY T.P\_id HAVING COUNT(Treatment\_id) > 1) GROUP BY D.Dr\_Name, P.P\_Name;

Q 22. Calculate the average project duration in days, but if a project has a duration of more than 90 days, categorize it as 'Long-Term,' between 30 and 90 days as 'Medium-Term,' and less than 30 days as 'Short-Term.'

A. SELECT AVG(Project\_Duration), CASE WHEN AVG(Project\_Duration) >= 30 THEN 'Medium-Term' WHEN AVG(Project\_Duration) > 90 THEN 'Long-Term' ELSE 'Short-Term' END FROM Project

B. SELECT AVG(Project\_Duration), CASE WHEN AVG(Project\_Duration) >= 30 THEN 'Long-Term' WHEN AVG(Project\_Duration) > 90 THEN 'Medium-Term' ELSE 'Short-Term' END FROM Project

C. SELECT AVG(Project\_Duration), CASE WHEN AVG(Project\_Duration) > 90 THEN 'Long-Term' WHEN AVG(Project\_Duration) >= 30 THEN 'Medium-Term' ELSE 'Short-Term' END FROM Project

D. SELECT AVG(Project\_Duration), CASE WHEN AVG(Project\_Duration) > 90 THEN 'Short-Term' WHEN AVG(Project\_Duration) >= 30 THEN 'Medium-Term' ELSE 'Long-Term' END FROM Project

Q 23. Update the designation of employees with more than 10 years of experience to 'Senior Developer.'

A. UPDATE Employee SET Emp\_Designation = 'Senior Developer' WHERE Emp\_Expertiese < 10

- B. UPDATE Employee SET Emp\_Designation = 'Senior Developer' WHERE Emp\_Expertiese > 10
- C. UPDATE Employee SET Emp\_Designation = 'Senior Developer' WHERE Emp\_Expertiese >= 10
- D. UPDATE Employee SET Emp\_Designation = 'Senior Developer' WHERE Emp\_Expertiese = 10

Q 24. Find the projects that have at least one employee with expertise in 'Java.'

- A. SELECT Project\_Name FROM Project WHERE Project\_ID IN (SELECT Project\_ID FROM Work WHERE Emp\_ID IN (SELECT Emp\_ID FROM Employee WHERE Emp\_Expertiese = 'Java'))
- B. SELECT Project\_Name FROM Project WHERE Project\_ID IN (SELECT Project\_ID FROM Work WHERE Emp\_ID IN (SELECT Emp\_ID FROM Employee WHERE Emp\_Expertiese = 'Java'))
- C. SELECT Project\_Name FROM Project WHERE Project\_ID IN (SELECT Project\_ID FROM Work WHERE Emp\_ID IN (SELECT Emp\_ID FROM Employee WHERE Emp\_Expertiese LIKE '%Java%'))
- D. SELECT Project\_Name FROM Project WHERE Project\_ID IN (SELECT Project\_ID FROM Work WHERE Emp\_ID IN (SELECT Emp\_ID FROM Employee WHERE Emp\_Expertiese IN ('Java', 'Java Script')))

Q 25. Identify employees who have worked on projects with a budget exceeding \$100,000 and have more than 5 years of experience. Update their designation to 'Project Lead.'

- A. UPDATE Employee SET Emp\_Designation = 'Project Lead' WHERE Emp\_ID IN (SELECT Emp\_ID FROM Work INNER JOIN Project ON Work.Project\_ID = Project.Project\_ID WHERE Project\_Budget >= 100000 AND Emp\_Expertiese >= 5)
- B. UPDATE Employee SET Emp\_Designation = 'Project Lead' WHERE Emp\_ID IN (SELECT Emp\_ID FROM Work INNER JOIN Project ON Work.Project\_ID = Project.Project\_ID WHERE Project\_Budget >= 100000 AND Emp\_Expertiese > 5)
- C. UPDATE Employee SET Emp\_Designation = 'Project Lead' WHERE Emp\_ID IN (SELECT Emp\_ID FROM Work INNER JOIN Project ON Work.Project\_ID = Project.Project\_ID WHERE Project\_Budget > 100000 AND Emp\_Expertiese >= 5)
- D. UPDATE Employee SET Emp\_Designation = 'Project Lead' WHERE Emp\_ID IN (SELECT Emp\_ID FROM Work INNER JOIN Project ON Work.Project\_ID = Project.Project\_ID WHERE Project\_Budget > 100000 AND Emp\_Expertiese > 5)

Q 26. Find the projects that have employees with expertise in both 'Java' and 'Python.'

- A. `SELECT Project_Name FROM Project WHERE Project_ID IN (SELECT Project_ID FROM Work WHERE Emp_ID IN (SELECT Emp_ID FROM Employee WHERE Emp_Expertiese = 'Java')) AND Project_ID IN (SELECT Project_ID FROM Work WHERE Emp_ID IN (SELECT Emp_ID FROM Employee WHERE Emp_Expertiese = 'Python'))`
- B. `SELECT Project_Name FROM Project WHERE Project_ID IN (SELECT Project_ID FROM Work WHERE Emp_ID IN (SELECT Emp_ID FROM Employee WHERE Emp_Expertiese = 'Java') OR Emp_ID IN (SELECT Emp_ID FROM Employee WHERE Emp_Expertiese = 'Python'))`
- C. `SELECT Project_Name FROM Project WHERE Project_ID IN (SELECT Project_ID FROM Work WHERE Emp_ID IN (SELECT Emp_ID FROM Employee WHERE Emp_Expertiese = 'Java') AND Project_ID IN (SELECT Project_ID FROM Work WHERE Emp_ID IN (SELECT Emp_ID FROM Employee WHERE Emp_Expertiese = 'Python')))`
- D. `SELECT Project_Name FROM Project WHERE Project_ID IN (SELECT Project_ID FROM Work WHERE Emp_ID IN (SELECT Emp_ID FROM Employee WHERE Emp_Expertiese = 'Java') AND Project_ID IN (SELECT Project_ID FROM Work WHERE Emp_ID IN (SELECT Emp_ID FROM Employee WHERE Emp_Expertiese = 'Python')))`

Q 27. Calculate the average salary for employees in each department, but if the average salary is below \$50,000, categorize the department as 'Low Pay,' between \$50,000 and \$75,000 as 'Average Pay,' and above \$75,000 as 'High Pay.'

- A. `SELECT Department, AVG(Emp_Salary), CASE WHEN AVG(Emp_Salary) < 50000 THEN 'Average Pay' WHEN AVG(Emp_Salary) >= 50000 AND AVG(Emp_Salary) <= 75000 THEN 'Low Pay' ELSE 'High Pay' END FROM Employee GROUP BY Department`
- B. `SELECT Department, AVG(Emp_Salary), CASE WHEN AVG(Emp_Salary) < 50000 THEN 'Low Pay' WHEN AVG(Emp_Salary) >= 50000 AND AVG(Emp_Salary) <= 75000 THEN 'Average Pay' ELSE 'High Pay' END FROM Employee GROUP BY Department`
- C. `SELECT Department, CASE WHEN AVG(Emp_Salary) < 50000 THEN 'High Pay' WHEN AVG(Emp_Salary) >= 50000 AND AVG(Emp_Salary) <= 75000 THEN 'Low Pay' ELSE 'Average Pay' END FROM Employee GROUP BY Department`
- D. `SELECT Department, CASE WHEN AVG(Emp_Salary) < 50000 THEN 'Low Pay' WHEN AVG(Emp_Salary) >= 50000 AND AVG(Emp_Salary) <= 75000 THEN 'High Pay' ELSE 'Average Pay' END FROM Employee GROUP BY Department`

Q 28. Calculate the total bonus for employees who have more than 3 years of experience. If the bonus is above \$2,000, categorize it as 'High Bonus,' between \$1,000 and \$2,000 as 'Medium Bonus,' and below \$1,000 as 'Low Bonus.'

- A. SELECT Emp\_Name, SUM(Emp\_Bonus), CASE WHEN SUM(Emp\_Bonus) > 2000 THEN 'High Bonus' WHEN SUM(Emp\_Bonus) >= 1000 THEN 'Medium Bonus' ELSE 'Low Bonus' END FROM Employee WHERE Emp\_Experience > 3 GROUP BY Emp\_Name
- B. SELECT Emp\_Name, SUM(Emp\_Bonus), CASE WHEN SUM(Emp\_Bonus) >= 1000 THEN 'Medium Bonus' WHEN SUM(Emp\_Bonus) > 2000 THEN 'Low Bonus' ELSE 'High Bonus' END FROM Employee WHERE Emp\_Experience > 3 GROUP BY Emp\_Name
- C. SELECT Emp\_Name, SUM(Emp\_Bonus), CASE WHEN SUM(Emp\_Bonus) > 1000 THEN 'Low Bonus' WHEN SUM(Emp\_Bonus) >= 2000 THEN 'Medium Bonus' ELSE 'High Bonus' END FROM Employee WHERE Emp\_Experience > 3 GROUP BY Emp\_Name
- D. SELECT Emp\_Name, SUM(Emp\_Bonus), CASE WHEN SUM(Emp\_Bonus) >= 1000 THEN 'Low Bonus' WHEN SUM(Emp\_Bonus) > 2000 THEN 'Medium Bonus' ELSE 'High Bonus' END FROM Employee WHERE Emp\_Experience > 3 GROUP BY Emp\_Name

Q 29. Identify the patients who have been treated by doctors from the same city and specialization as the patient, and show the names of the doctors.

- A. SELECT P.P\_Name, D.Dr\_Name FROM Patient P INNER JOIN Treatment T ON P.P\_id = T.P\_id INNER JOIN Doctor D ON T.Dr\_id = D.Dr\_id WHERE P.P\_City <> D.Dr\_City AND P.P\_Specialization <> D.Dr\_Specialization GROUP BY P.P\_Name, D.Dr\_Name;
- B. SELECT P.P\_Name, D.Dr\_Name FROM Patient P INNER JOIN Treatment T ON P.P\_id = T.P\_id INNER JOIN Doctor D ON T.Dr\_id = D.Dr\_id WHERE P.P\_City = D.Dr\_City AND P.P\_Specialization <> D.Dr\_Specialization GROUP BY P.P\_Name, D.Dr\_Name;
- C. SELECT P.P\_Name, D.Dr\_Name FROM Patient P INNER JOIN Treatment T ON P.P\_id = T.P\_id INNER JOIN Doctor D ON T.Dr\_id = D.Dr\_id WHERE P.P\_City = D.Dr\_City AND P.P\_Specialization = D.Dr\_Specialization GROUP BY P.P\_Name, D.Dr\_Name;
- D. SELECT P.P\_Name, D.Dr\_Name FROM Patient P INNER JOIN Treatment T ON P.P\_id = T.P\_id INNER JOIN Doctor D ON T.Dr\_id = D.Dr\_id WHERE P.P\_City <> D.Dr\_City AND P.P\_Specialization = D.Dr\_Specialization GROUP BY P.P\_Name, D.Dr\_Name;

Q 30. List the employees who have not been assigned to any projects yet.

- A. SELECT Emp\_Name FROM Employee WHERE Emp\_ID IS NULL
- B. SELECT Emp\_Name FROM Employee WHERE Emp\_ID NOT IN (SELECT Emp\_ID FROM Work)
- C. SELECT Emp\_Name FROM Employee WHERE Emp\_ID IN (SELECT Emp\_ID FROM Work)
- D. SELECT Emp\_Name FROM Employee WHERE Emp\_ID = NULL

Q 31. List the doctors who have treated patients from 'New York City' and have performed more than 3 different tests.

- A. SELECT DISTINCT D.Dr\_Name FROM Doctor D INNER JOIN Treatment T ON D.Dr\_id = T.Dr\_id INNER JOIN Patient P ON T.P\_id = P.P\_id INNER JOIN Test TS ON P.P\_id = TS.P\_id WHERE P.P\_City = 'New York City' HAVING COUNT(DISTINCT TS.Test\_Name) <= 3;
- B. SELECT DISTINCT D.Dr\_Name FROM Doctor D INNER JOIN Treatment T ON D.Dr\_id = T.Dr\_id INNER JOIN Patient P ON T.P\_id = P.P\_id INNER JOIN Test TS ON P.P\_id = TS.P\_id WHERE P.P\_City = 'New York City' GROUP BY D.Dr\_id, D.Dr\_Name HAVING COUNT(DISTINCT TS.Test\_Name) <= 3;
- C. SELECT DISTINCT D.Dr\_Name FROM Doctor D INNER JOIN Treatment T ON D.Dr\_id = T.Dr\_id INNER JOIN Patient P ON T.P\_id = P.P\_id INNER JOIN Test TS ON P.P\_id = TS.P\_id WHERE P.P\_City = 'New York City' HAVING COUNT(DISTINCT TS.Test\_Name) > 3;
- D. SELECT DISTINCT D.Dr\_Name FROM Doctor D INNER JOIN Treatment T ON D.Dr\_id = T.Dr\_id INNER JOIN Patient P ON T.P\_id = P.P\_id INNER JOIN Test TS ON P.P\_id = TS.P\_id WHERE P.P\_City = 'New York City' GROUP BY D.Dr\_id, D.Dr\_Name HAVING COUNT(DISTINCT TS.Test\_Name) > 3;

Q 32. Retrieve the names of patients who have not had any tests and are treated by doctors with less than 5 years of experience.

- A. SELECT P.P\_Name FROM Patient P WHERE P.P\_id NOT IN (SELECT T.P\_id FROM Test T) AND P.P\_id NOT IN (SELECT D.P\_id FROM Doctor D WHERE D.Dr\_Experience < 5);
- B. SELECT P.P\_Name FROM Patient P WHERE P.P\_id IN (SELECT T.P\_id FROM Test T) AND P.P\_id NOT IN (SELECT D.P\_id FROM Doctor D WHERE D.Dr\_Experience < 5);
- C. SELECT P.P\_Name FROM Patient P WHERE P.P\_id IN (SELECT T.P\_id FROM Test T) AND P.P\_id IN (SELECT D.P\_id FROM Doctor D WHERE D.Dr\_Experience < 5);
- D. SELECT P.P\_Name FROM Patient P WHERE P.P\_id NOT IN (SELECT T.P\_id FROM Test T) AND P.P\_id IN (SELECT D.P\_id FROM Doctor D WHERE D.Dr\_Experience < 5);

Q 33. Find the employees who have the same designation as their project category.

- A. `SELECT Emp_Name FROM Employee WHERE Emp_Designation = (SELECT Project_Categoery FROM Project WHERE Project_ID = Work.Project_ID) AND Emp_ID = Work.Emp_ID`
- B. `SELECT Emp_Name FROM Employee WHERE Emp_Designation = (SELECT Project_Categoery FROM Project WHERE Project_ID = Work.Project_ID) AND Emp_ID = (SELECT Emp_ID FROM Work WHERE Emp_Designation = Project_Categoery)`
- C. `SELECT Emp_Name FROM Employee WHERE Emp_Designation = (SELECT Project_Categoery FROM Project WHERE Project_ID = Work.Project_ID)`
- D. `SELECT Emp_Name FROM Employee WHERE Emp_Designation = (SELECT Project_Categoery FROM Project WHERE Project_ID = Work.Project_ID) AND Emp_ID = (SELECT Emp_ID FROM Work WHERE Emp_Designation = Project_Categoery)`

Q 34. Find the total number of patients admitted by each doctor, but only if the doctor has treated more than 10 patients.

- A. `SELECT D.Dr_Name, COUNT(T.P_id) AS Total_Patients FROM Doctor D LEFT JOIN Treatment T ON D.Dr_id = T.Dr_id GROUP BY D.Dr_id, D.Dr_Name HAVING COUNT(T.P_id) >= 10;`
- B. `SELECT D.Dr_Name, COUNT(T.P_id) AS Total_Patients FROM Doctor D LEFT JOIN Treatment T ON D.Dr_id = T.Dr_id GROUP BY D.Dr_id, D.Dr_Name HAVING COUNT(T.P_id) <= 10;`
- C. `SELECT D.Dr_Name, COUNT(T.P_id) AS Total_Patients FROM Doctor D LEFT JOIN Treatment T ON D.Dr_id = T.Dr_id GROUP BY D.Dr_id, D.Dr_Name HAVING COUNT(T.P_id) > 10;`
- D. `SELECT D.Dr_Name, COUNT(T.P_id) AS Total_Patients FROM Doctor D LEFT JOIN Treatment T ON D.Dr_id = T.Dr_id GROUP BY D.Dr_id, D.Dr_Name HAVING COUNT(T.P_id) < 10;`

Q 35. Find the patients who have been treated by doctors from the 'Bay Area' and have had both 'X-ray' and 'MRI' tests.



- A. `SELECT P.P_Name, P.P_id FROM Patient P INNER JOIN Treatment T ON P.P_id = T.P_id  
INNER JOIN Doctor D ON T.Dr_id = D.Dr_id INNER JOIN Test TS ON P.P_id = TS.P_id WHERE  
P.P_City = 'Bay Area' AND TS.Test_Name IN ('X-ray', 'MRI') GROUP BY P.P_id, P.P_Name  
HAVING COUNT(DISTINCT TS.Test_Name) = 2;`
- B. `SELECT P.P_Name, P.P_id FROM Patient P INNER JOIN Treatment T ON P.P_id = T.P_id  
INNER JOIN Doctor D ON T.Dr_id = D.Dr_id INNER JOIN Test TS ON P.P_id = TS.P_id WHERE  
P.P_City = 'Bay Area' AND TS.Test_Name IN ('X-ray', 'CT Scan') GROUP BY P.P_id, P.P_Name  
HAVING COUNT(DISTINCT TS.Test_Name) = 2;`
- C. `SELECT P.P_Name, P.P_id FROM Patient P INNER JOIN Treatment T ON P.P_id = T.P_id  
INNER JOIN Doctor D ON T.Dr_id = D.Dr_id INNER JOIN Test TS ON P.P_id = TS.P_id WHERE  
P.P_City = 'New York City' AND TS.Test_Name IN ('X-ray', 'MRI') GROUP BY P.P_id, P.P_Name  
HAVING COUNT(DISTINCT TS.Test_Name) = 2;`
- D. `SELECT P.P_Name, P.P_id FROM Patient P INNER JOIN Treatment T ON P.P_id = T.P_id  
INNER JOIN Doctor D ON T.Dr_id = D.Dr_id INNER JOIN Test TS ON P.P_id = TS.P_id WHERE  
P.P_City = 'Bay Area' AND TS.Test_Name IN ('X-ray', 'MRI') GROUP BY P.P_id, P.P_Name  
HAVING COUNT(DISTINCT TS.Test_Name) = 1;`

Q 36. Calculate the total cost of treatment for patients who have been admitted more than once and are treated by doctors specializing in 'Surgery.'

- A. `SELECT T.P_id, SUM(Treatment_Cost) AS Total_Cost FROM Treatment T INNER JOIN  
Doctor D ON T.Dr_id = D.Dr_id WHERE D.Dr_specialization = 'Oncology' AND T.P_id IN  
(SELECT P_id FROM Treatment GROUP BY P_id HAVING COUNT(Treatment_id) <= 1) GROUP  
BY T.P_id;`
- B. `SELECT T.P_id, SUM(Treatment_Cost) AS Total_Cost FROM Treatment T INNER JOIN  
Doctor D ON T.Dr_id = D.Dr_id WHERE D.Dr_specialization = 'Surgery' AND T.P_id IN (SELECT  
P_id FROM Treatment GROUP BY P_id HAVING COUNT(Treatment_id) <= 1) GROUP BY T.P_id;`
- C. `SELECT T.P_id, SUM(Treatment_Cost) AS Total_Cost FROM Treatment T INNER JOIN  
Doctor D ON T.Dr_id = D.Dr_id WHERE D.Dr_specialization = 'Surgery' AND T.P_id IN (SELECT  
P_id FROM Treatment GROUP BY P_id HAVING COUNT(Treatment_id) > 1) GROUP BY T.P_id;`
- D. `SELECT T.P_id, SUM(Treatment_Cost) AS Total_Cost FROM Treatment T INNER JOIN  
Doctor D ON T.Dr_id = D.Dr_id WHERE D.Dr_specialization = 'Oncology' AND T.P_id IN  
(SELECT P_id FROM Treatment GROUP BY P_id HAVING COUNT(Treatment_id) > 1) GROUP BY  
T.P_id;`

Q 37. Find the doctors who have not treated patients with 'Diabetes' and specialize in 'Cardiology' or 'Endocrinology.'

- A. `SELECT D.Dr_Name, D.Dr_id FROM Doctor D WHERE D.Dr_specialization IN ('Cardiology', 'Endocrinology') AND D.Dr_id IN (SELECT DISTINCT T.Dr_id FROM Treatment T INNER JOIN Patient P ON T.P_id = P.P_id WHERE P.P_Disease = 'Diabetes');`
- B. `SELECT D.Dr_Name, D.Dr_id FROM Doctor D WHERE D.Dr_specialization IN ('Cardiology', 'Endocrinology') AND D.Dr_id NOT IN (SELECT DISTINCT T.Dr_id FROM Treatment T INNER JOIN Patient P ON T.P_id = P.P_id WHERE P.P_Disease = 'Diabetes');`
- C. `SELECT D.Dr_Name, D.Dr_id FROM Doctor D WHERE D.Dr_specialization IN ('Oncology', 'Surgery') AND D.Dr_id NOT IN (SELECT DISTINCT T.Dr_id FROM Treatment T INNER JOIN Patient P ON T.P_id = P.P_id WHERE P.P_Disease = 'Diabetes');`
- D. `SELECT D.Dr_Name, D.Dr_id FROM Doctor D WHERE D.Dr_specialization IN ('Oncology', 'Surgery') AND D.Dr_id IN (SELECT DISTINCT T.Dr_id FROM Treatment T INNER JOIN Patient P ON T.P_id = P.P_id WHERE P.P_Disease = 'Diabetes');`

Q 38. Retrieve the projects with the highest number of employees working on them.

- A. `SELECT Project_Name FROM Project WHERE Project_ID IN (SELECT Project_id FROM Work GROUP BY Project_id ORDER BY COUNT(Emp_ID) DESC LIMIT 1)`
- B. `SELECT Project_Name FROM Project WHERE Project_ID IN (SELECT Project_id FROM Work GROUP BY Project_id HAVING COUNT(Emp_ID) = SUM(COUNT(Emp_ID)))`
- C. `SELECT Project_Name FROM Project WHERE Project_ID IN (SELECT Project_id FROM Work GROUP BY Project_id HAVING COUNT(Emp_ID) = MIN(COUNT(Emp_ID)))`
- D. `SELECT Project_Name FROM Project WHERE Project_ID IN (SELECT Project_id FROM Work GROUP BY Project_id HAVING COUNT(Emp_ID) = MAX(COUNT(Emp_ID)))`

Q 39. List the employees who are working on projects with a specific technology, e.g., 'SQL Server.'

- A. `SELECT Emp_Name FROM Employee WHERE Emp_ID IN (SELECT Emp_ID FROM Work WHERE Technology = 'SQL Server')`
- B. `SELECT Emp_Name FROM Employee WHERE Emp_ID IN (SELECT Emp_ID FROM Work WHERE Technology LIKE '%SQL Server%')`
- C. `SELECT Emp_Name FROM Employee WHERE Emp_ID IN (SELECT Emp_ID FROM Work WHERE Technology IN ('SQL Server', 'SQL'))`

D. SELECT Emp\_Name FROM Employee WHERE Emp\_ID IN (SELECT Emp\_ID FROM Work WHERE Technology = 'MySQL')

Q 40. Retrieve the employee names who are working on projects that are not in the 'Research' category.

A. SELECT Emp\_Name FROM Employee WHERE Emp\_ID IN (SELECT Emp\_ID FROM Work WHERE Project\_id IN (SELECT Project\_ID FROM Project WHERE Project\_Categoery = 'Development'))

B. SELECT Emp\_Name FROM Employee WHERE Emp\_ID IN (SELECT Emp\_ID FROM Work WHERE Project\_id IN (SELECT Project\_ID FROM Project WHERE Project\_Categoery <> 'Research'))

C. SELECT Emp\_Name FROM Employee WHERE Emp\_ID IN (SELECT Emp\_ID FROM Work WHERE Project\_id IN (SELECT Project\_ID FROM Project WHERE Project\_Categoery = 'Development'))

D. SELECT Emp\_Name FROM Employee WHERE Emp\_ID IN (SELECT Emp\_ID FROM Work WHERE Project\_id NOT IN (SELECT Project\_ID FROM Project WHERE Project\_Categoery = 'Research'))

Q 41. List the doctors who have not treated patients from 'Los Angeles' and have performed more than 5 tests in total.

A. SELECT D.Dr\_Name FROM Doctor D LEFT JOIN Treatment T ON D.Dr\_id = T.Dr\_id LEFT JOIN Patient P ON T.P\_id = P.P\_id LEFT JOIN Test TS ON P.P\_id = TS.P\_id WHERE P.P\_City = 'Los Angeles' GROUP BY D.Dr\_id, D.Dr\_Name HAVING COUNT(DISTINCT TS.Test\_id) <= 5;

B. SELECT D.Dr\_Name FROM Doctor D LEFT JOIN Treatment T ON D.Dr\_id = T.Dr\_id LEFT JOIN Patient P ON T.P\_id = P.P\_id LEFT JOIN Test TS ON P.P\_id = TS.P\_id WHERE P.P\_City <> 'Los Angeles' OR P.P\_id IS NULL GROUP BY D.Dr\_id, D.Dr\_Name HAVING COUNT(DISTINCT TS.Test\_id) > 5;

C. SELECT D.Dr\_Name FROM Doctor D LEFT JOIN Treatment T ON D.Dr\_id = T.Dr\_id LEFT JOIN Patient P ON T.P\_id = P.P\_id LEFT JOIN Test TS ON P.P\_id = TS.P\_id WHERE P.P\_City = 'Los Angeles' GROUP BY D.Dr\_id, D.Dr\_Name HAVING COUNT(DISTINCT TS.Test\_id) > 5;

D. SELECT D.Dr\_Name FROM Doctor D LEFT JOIN Treatment T ON D.Dr\_id = T.Dr\_id LEFT JOIN Patient P ON T.P\_id = P.P\_id LEFT JOIN Test TS ON P.P\_id = TS.P\_id WHERE P.P\_City <> 'Los Angeles' OR P.P\_id IS NULL GROUP BY D.Dr\_id, D.Dr\_Name HAVING COUNT(DISTINCT TS.Test\_id) <= 5;

Q 42. Find the projects where the total budget exceeds \$100,000, and categorize them as 'High Budget.' If the budget is between \$50,000 and \$100,000, categorize them as 'Medium Budget,' and below \$50,000 as 'Low Budget.'

A. SELECT Project\_Name, CASE WHEN Project\_Budget >= 50000 AND Project\_Budget <= 100000 THEN 'Low Budget' WHEN Project\_Budget > 100000 THEN 'High Budget' ELSE 'Medium Budget' END FROM Project

B. SELECT Project\_Name, CASE WHEN Project\_Budget >= 50000 AND Project\_Budget <= 100000 THEN 'High Budget' WHEN Project\_Budget > 100000 THEN 'Medium Budget' ELSE 'Low Budget' END FROM Project

C. SELECT Project\_Name, CASE WHEN Project\_Budget > 100000 THEN 'High Budget' WHEN Project\_Budget >= 50000 AND Project\_Budget <= 100000 THEN 'Medium Budget' ELSE 'Low Budget' END FROM Project

D. SELECT Project\_Name, CASE WHEN Project\_Budget >= 50000 AND Project\_Budget <= 100000 THEN 'Medium Budget' WHEN Project\_Budget > 100000 THEN 'High Budget' ELSE 'Low Budget' END FROM Project

Q 43. List the patients who have had multiple treatments and show the names of their treating doctors for each treatment.

A. SELECT P.P\_Name, T.P\_id, D.Dr\_Name, T.Dr\_id FROM Patient P INNER JOIN Treatment T ON P.P\_id = T.P\_id INNER JOIN Doctor D ON T.Dr\_id = D.Dr\_id WHERE T.P\_id IN (SELECT T.P\_id FROM Treatment T GROUP BY T.P\_id HAVING COUNT(Treatment\_id) > 1);

B. SELECT P.P\_Name, T.P\_id, D.Dr\_Name, T.Dr\_id FROM Patient P INNER JOIN Treatment T ON P.P\_id = T.P\_id INNER JOIN Doctor D ON T.Dr\_id = D.Dr\_id WHERE T.P\_id IN (SELECT T.P\_id FROM Treatment T GROUP BY T.P\_id HAVING COUNT(Treatment\_id) >= 2);

C. SELECT P.P\_Name, T.P\_id, D.Dr\_Name, T.Dr\_id FROM Patient P INNER JOIN Treatment T ON P.P\_id = T.P\_id INNER JOIN Doctor D ON T.Dr\_id = D.Dr\_id WHERE T.P\_id IN (SELECT T.P\_id FROM Treatment T GROUP BY T.P\_id HAVING COUNT(Treatment\_id) <= 1);

D. SELECT P.P\_Name, T.P\_id, D.Dr\_Name, T.Dr\_id FROM Patient P INNER JOIN Treatment T ON P.P\_id = T.P\_id INNER JOIN Doctor D ON T.Dr\_id = D.Dr\_id WHERE T.P\_id IN (SELECT T.P\_id FROM Treatment T GROUP BY T.P\_id HAVING COUNT(Treatment\_id) < 2);

Q 44. Retrieve a list of employees with their project details (Emp\_Name, Project\_Name).

- A. SELECT Emp\_Name, Project\_Name FROM Employee RIGHT JOIN Work ON Employee.Emp\_ID = Work.Emp\_ID RIGHT JOIN Project ON Work.Project\_ID = Project.Project\_ID
- B. SELECT Emp\_Name, Project\_Name FROM Employee INNER JOIN Work ON Employee.Emp\_ID = Work.Emp\_ID INNER JOIN Project ON Work.Project\_ID = Project.Project\_ID
- C. SELECT Emp\_Name, Project\_Name FROM Employee LEFT JOIN Work ON Employee.Emp\_ID = Work.Emp\_ID LEFT JOIN Project ON Work.Project\_ID = Project.Project\_ID
- D. SELECT Emp\_Name, Project\_Name FROM Employee CROSS JOIN Work CROSS JOIN Project

Q 45. Retrieve the names of employees who have more than 5 years of experience.

- A. SELECT Emp\_Name FROM Employee WHERE Emp\_Expertiese = 5
- B. SELECT Emp\_Name FROM Employee WHERE Emp\_Expertiese < 5
- C. SELECT Emp\_Name FROM Employee WHERE Emp\_Expertiese > 5
- D. SELECT Emp\_Name FROM Employee WHERE Emp\_Expertiese >= 5

Q 46. Calculate the total salary of each employee, including a bonus of \$500 for employees with the designation 'Manager' and \$200 for employees with the designation 'Senior Developer.'

- A. SELECT Emp\_Name, Emp\_Salary + CASE WHEN Emp\_Desigation = 'Manager' THEN 500 WHEN Emp\_Desigation = 'Senior Developer' THEN 200 ELSE 0 END FROM Employee
- B. SELECT Emp\_Name, Emp\_Salary + CASE WHEN Emp\_Desigation = 'Senior Developer' THEN 200 WHEN Emp\_Desigation = 'Manager' THEN 500 ELSE 0 END FROM Employee
- C. SELECT Emp\_Name, Emp\_Salary + CASE WHEN Emp\_Desigation = 'Manager' THEN 200 WHEN Emp\_Desigation = 'Senior Developer' THEN 500 ELSE 0 END FROM Employee
- D. SELECT Emp\_Name, Emp\_Salary + CASE WHEN Emp\_Desigation = 'Senior Developer' THEN 500 WHEN Emp\_Desigation = 'Manager' THEN 200 ELSE 0 END FROM Employee

Q 47. Retrieve the projects with their respective statuses, but if a project started before 2022, categorize it as 'Old Project,' otherwise as 'Recent Project.'

- A. `SELECT Project_Name, CASE WHEN Project_Start_Date < '2022-01-01' THEN 'Old Project' ELSE 'Recent Project' END FROM Project`
- B. `SELECT Project_Name, CASE WHEN Project_Start_Date < '2022-01-01' THEN 'Recent Project' ELSE 'Old Project' END FROM Project`
- C. `SELECT Project_Name, CASE WHEN Project_Start_Date > '2022-01-01' THEN 'Old Project' ELSE 'Recent Project' END FROM Project`
- D. `SELECT Project_Name, CASE WHEN Project_Start_Date > '2022-01-01' THEN 'Recent Project' ELSE 'Old Project' END FROM Project`

Q 48. List the employees who are assigned to projects with the word 'Database' in their project name.

- A. `SELECT Emp_Name FROM Employee WHERE Emp_ID IN (SELECT Emp_ID FROM Work WHERE Project_id IN (SELECT Project_ID FROM Project WHERE Project_Name LIKE '%Database%'))`
- B. `SELECT Emp_Name FROM Employee WHERE Emp_ID IN (SELECT Emp_ID FROM Work WHERE Project_id = (SELECT Project_ID FROM Project WHERE Project_Name LIKE 'Database'))`
- C. `SELECT Emp_Name FROM Employee WHERE Emp_ID IN (SELECT Emp_ID FROM Work WHERE Project_id = (SELECT Project_ID FROM Project WHERE Project_Name = 'Database'))`
- D. `SELECT Emp_Name FROM Employee WHERE Emp_ID IN (SELECT Emp_ID FROM Work WHERE Project_id = (SELECT Project_ID FROM Project WHERE Project_Name IN ('Database', 'Data Storage')))`

Q 49. Calculate the average treatment cost for patients who have been admitted more than once and are treated by doctors specializing in 'Cardiology.'

- A. `SELECT AVG(Treatment_Cost) AS Avg_Treatment_Cost FROM Treatment T INNER JOIN Patient P ON T.P_id = P.P_id INNER JOIN Doctor D ON T.Dr_id = D.Dr_id WHERE P.P_id IN (SELECT P_id FROM Treatment GROUP BY P_id HAVING COUNT(Treatment_id) > 1) AND D.Dr_Specialization = 'Cardiology';`

B. SELECT AVG(Treatment\_Cost) AS Avg\_Treatment\_Cost FROM Treatment T INNER JOIN Patient P ON T.P\_id = P.P\_id INNER JOIN Doctor D ON T.Dr\_id = D.Dr\_id WHERE P.P\_id IN (SELECT P\_id FROM Treatment GROUP BY P\_id HAVING COUNT(Treatment\_id) <= 1) AND D.Dr\_Specialization <> 'Cardiology';

C. SELECT AVG(Treatment\_Cost) AS Avg\_Treatment\_Cost FROM Treatment T INNER JOIN Patient P ON T.P\_id = P.P\_id INNER JOIN Doctor D ON T.Dr\_id = D.Dr\_id WHERE P.P\_id IN (SELECT P\_id FROM Treatment GROUP BY P\_id HAVING COUNT(Treatment\_id) <= 1) AND D.Dr\_Specialization = 'Cardiology';

D. SELECT AVG(Treatment\_Cost) AS Avg\_Treatment\_Cost FROM Treatment T INNER JOIN Patient P ON T.P\_id = P.P\_id INNER JOIN Doctor D ON T.Dr\_id = D.Dr\_id WHERE P.P\_id IN (SELECT P\_id FROM Treatment GROUP BY P\_id HAVING COUNT(Treatment\_id) > 1) AND D.Dr\_Specialization <> 'Cardiology';

Q 50. Retrieve the names of doctors who have treated patients from the 'Bay Area' and have performed at least one 'MRI' test.

A. SELECT DISTINCT D.Dr\_Name FROM Doctor D INNER JOIN Treatment T ON D.Dr\_id = T.Dr\_id INNER JOIN Patient P ON T.P\_id = P.P\_id INNER JOIN Test TS ON P.P\_id = TS.P\_id WHERE P.P\_City = 'New York City' AND TS.Test\_Name = 'CT Scan';

B. SELECT DISTINCT D.Dr\_Name FROM Doctor D INNER JOIN Treatment T ON D.Dr\_id = T.Dr\_id INNER JOIN Patient P ON T.P\_id = P.P\_id INNER JOIN Test TS ON P.P\_id = TS.P\_id WHERE P.P\_City = 'Bay Area' AND TS.Test\_Name = 'MRI';

C. SELECT DISTINCT D.Dr\_Name FROM Doctor D INNER JOIN Treatment T ON D.Dr\_id = T.Dr\_id INNER JOIN Patient P ON T.P\_id = P.P\_id INNER JOIN Test TS ON P.P\_id = TS.P\_id WHERE P.P\_City = 'New York City' AND TS.Test\_Name = 'MRI';

D. SELECT DISTINCT D.Dr\_Name FROM Doctor D INNER JOIN Treatment T ON D.Dr\_id = T.Dr\_id INNER JOIN Patient P ON T.P\_id = P.P\_id INNER JOIN Test TS ON P.P\_id = TS.P\_id WHERE P.P\_City = 'Bay Area' AND TS.Test\_Name = 'CT Scan';

Q 51. Calculate the average treatment duration for patients admitted by doctors specializing in 'Oncology' and who have had more than one treatment.

A. SELECT AVG(Days) AS Avg\_Duration FROM (SELECT P\_id, D.Dr\_id, DATEDIFF(Admit\_Date, MIN(Admit\_Date)) AS Days FROM Treatment T INNER JOIN Doctor D ON T.Dr\_id = D.Dr\_id WHERE D.Dr\_specialization = 'Oncology' GROUP BY T.P\_id, D.Dr\_id HAVING COUNT(Treatment\_id) > 1) AS Subquery;

B. SELECT AVG(Days) AS Avg\_Duration FROM (SELECT P\_id, D.Dr\_id, DATEDIFF(Admit\_Date, MIN(Admit\_Date)) AS Days FROM Treatment T INNER JOIN Doctor D ON T.Dr\_id = D.Dr\_id WHERE D.Dr\_specialization = 'Cardiology' GROUP BY T.P\_id, D.Dr\_id HAVING COUNT(Treatment\_id) > 1) AS Subquery;

C. SELECT AVG(Days) AS Avg\_Duration FROM (SELECT P\_id, D.Dr\_id, DATEDIFF(Admit\_Date, MIN(Admit\_Date)) AS Days FROM Treatment T INNER JOIN Doctor D ON T.Dr\_id = D.Dr\_id WHERE D.Dr\_specialization = 'Cardiology' GROUP BY T.P\_id, D.Dr\_id HAVING COUNT(Treatment\_id) <= 1) AS Subquery;

D. SELECT AVG(Days) AS Avg\_Duration FROM (SELECT P\_id, D.Dr\_id, DATEDIFF(Admit\_Date, MIN(Admit\_Date)) AS Days FROM Treatment T INNER JOIN Doctor D ON T.Dr\_id = D.Dr\_id WHERE D.Dr\_specialization = 'Oncology' GROUP BY T.P\_id, D.Dr\_id HAVING COUNT(Treatment\_id) <= 1) AS Subquery;

Q 52. Identify the patients with the highest number of admissions, and show the details of their treating doctors.

A. SELECT P.P\_Name, P.P\_id, D.Dr\_Name, D.Dr\_id FROM Patient P INNER JOIN (SELECT T.P\_id, T.Dr\_id, COUNT(Treatment\_id) AS Admissions FROM Treatment T GROUP BY T.P\_id, T.Dr\_id HAVING Admissions = (SELECT MAX(Admissions) FROM (SELECT P\_id, Dr\_id, COUNT(Treatment\_id) AS Admissions FROM Treatment T GROUP BY P\_id, Dr\_id) AS Subquery)) AS Subquery ON P.P\_id = Subquery.P\_id INNER JOIN Doctor D ON Subquery.Dr\_id = D.Dr\_id;

B. SELECT P.P\_Name, P.P\_id, D.Dr\_Name, D.Dr\_id FROM Patient P INNER JOIN (SELECT T.P\_id, T.Dr\_id, COUNT(Treatment\_id) AS Admissions FROM Treatment T GROUP BY T.P\_id, T.Dr\_id HAVING Admissions = (SELECT MIN(Admissions) FROM (SELECT P\_id, Dr\_id, COUNT(Treatment\_id) AS Admissions FROM Treatment T GROUP BY P\_id, Dr\_id) AS Subquery)) AS Subquery ON P.P\_id = Subquery.P\_id INNER JOIN Doctor D ON Subquery.Dr\_id = D.Dr\_id;

C. SELECT P.P\_Name, P.P\_id, D.Dr\_Name, D.Dr\_id FROM Patient P INNER JOIN (SELECT T.P\_id, T.Dr\_id, COUNT(Treatment\_id) AS Admissions FROM Treatment T GROUP BY T.P\_id, T.Dr\_id HAVING Admissions = (SELECT MAX(Admissions) FROM (SELECT P\_id, Dr\_id, COUNT(Treatment\_id) AS Admissions FROM Treatment T GROUP BY P\_id, Dr\_id) AS Subquery)) AS Subquery ON P.P\_id = Subquery.P\_id INNER JOIN Doctor D ON Subquery.Dr\_id = D.Dr\_id;

D. SELECT P.P\_Name, P.P\_id, D.Dr\_Name, D.Dr\_id FROM Patient P INNER JOIN (SELECT T.P\_id, T.Dr\_id, COUNT(Treatment\_id) AS Admissions FROM Treatment T GROUP BY T.P\_id, T.Dr\_id HAVING Admissions = (SELECT AVG(Admissions) FROM (SELECT P\_id, Dr\_id, COUNT(Treatment\_id) AS Admissions FROM Treatment T GROUP BY P\_id, Dr\_id) AS Subquery)) AS Subquery ON P.P\_id = Subquery.P\_id INNER JOIN Doctor D ON Subquery.Dr\_id = D.Dr\_id;



Q 53. Calculate the average number of tests performed on patients who are allergic to 'Peanuts' and are treated by doctors with a specialization of 'Allergy.'

- A. `SELECT AVG(Num_Tests) AS Avg_Tests FROM (SELECT P.P_id, COUNT(TS.Test_Name) AS Num_Tests FROM Patient P INNER JOIN Treatment T ON P.P_id = T.P_id INNER JOIN Doctor D ON T.Dr_id = D.Dr_id INNER JOIN Test TS ON P.P_id = TS.P_id WHERE P.P_Allergies LIKE '%Peanuts%' AND D.Dr_specialization = 'Orthopedics' GROUP BY P.P_id) AS Subquery;`
- B. `SELECT AVG(Num_Tests) AS Avg_Tests FROM (SELECT P.P_id, COUNT(TS.Test_Name) AS Num_Tests FROM Patient P INNER JOIN Treatment T ON P.P_id = T.P_id INNER JOIN Doctor D ON T.Dr_id = D.Dr_id INNER JOIN Test TS ON P.P_id = TS.P_id WHERE P.P_Allergies LIKE '%Peanuts%' AND D.Dr_specialization = 'Pediatrics' GROUP BY P.P_id) AS Subquery;`
- C. `SELECT AVG(Num_Tests) AS Avg_Tests FROM (SELECT P.P_id, COUNT(TS.Test_Name) AS Num_Tests FROM Patient P INNER JOIN Treatment T ON P.P_id = T.P_id INNER JOIN Doctor D ON T.Dr_id = D.Dr_id INNER JOIN Test TS ON P.P_id = TS.P_id WHERE P.P_Allergies LIKE '%Peanuts%' AND D.Dr_specialization = 'Cardiology' GROUP BY P.P_id) AS Subquery;`
- D. `SELECT AVG(Num_Tests) AS Avg_Tests FROM (SELECT P.P_id, COUNT(TS.Test_Name) AS Num_Tests FROM Patient P INNER JOIN Treatment T ON P.P_id = T.P_id INNER JOIN Doctor D ON T.Dr_id = D.Dr_id INNER JOIN Test TS ON P.P_id = TS.P_id WHERE P.P_Allergies LIKE '%Peanuts%' AND D.Dr_specialization = 'Allergy' GROUP BY P.P_id) AS Subquery;`

Q 54. Find the projects where the project category is 'Development.'

- A. `SELECT Project_Name FROM Project WHERE Project_Categoery IN ('Development', 'Software')`
- B. `SELECT Project_Name FROM Project WHERE Project_Categoery = 'Research'`
- C. `SELECT Project_Name FROM Project WHERE Project_Categoery LIKE 'Develop%'`
- D. `SELECT Project_Name FROM Project WHERE Project_Categoery = 'Development'`

Q 55. Calculate the average project cost, but if a project cost is over \$50,000, categorize it as 'Expensive,' between \$20,000 and \$50,000 as 'Moderate,' and below \$20,000 as 'Inexpensive.'

- A. `SELECT AVG(Project_Cost), CASE WHEN AVG(Project_Cost) > 50000 THEN 'Expensive' WHEN AVG(Project_Cost) >= 20000 THEN 'Moderate' ELSE 'Inexpensive' END FROM Project`

- B. `SELECT AVG(Project_Cost), CASE WHEN AVG(Project_Cost) > 50000 THEN 'Inexpensive' WHEN AVG(Project_Cost) >= 20000 THEN 'Moderate' ELSE 'Expensive' END FROM Project`
- C. `SELECT AVG(Project_Cost), CASE WHEN AVG(Project_Cost) > 50000 THEN 'Moderate' WHEN AVG(Project_Cost) >= 20000 THEN 'Expensive' ELSE 'Inexpensive' END FROM Project`
- D. `SELECT AVG(Project_Cost), CASE WHEN AVG(Project_Cost) > 50000 THEN 'Moderate' WHEN AVG(Project_Cost) >= 20000 THEN 'Inexpensive' ELSE 'Expensive' END FROM Project`

Q 56. Find the projects and their categories along with the employee assigned to each project.

- A. `SELECT Project_Name, Project_Categoery, Emp_Name FROM Project CROSS JOIN Work CROSS JOIN Employee`
- B. `SELECT Project_Name, Project_Categoery, Emp_Name FROM Project RIGHT JOIN Work ON Project.Project_ID = Work.Project_ID RIGHT JOIN Employee ON Work.Emp_ID = Employee.Emp_ID`
- C. `SELECT Project_Name, Project_Categoery, Emp_Name FROM Project INNER JOIN Work ON Project.Project_ID = Work.Project_ID INNER JOIN Employee ON Work.Emp_ID = Employee.Emp_ID`
- D. `SELECT Project_Name, Project_Categoery, Emp_Name FROM Project LEFT JOIN Work ON Project.Project_ID = Work.Project_ID LEFT JOIN Employee ON Work.Emp_ID = Employee.Emp_ID`

Q 57. Retrieve the names of patients who have had more than 5 different tests and are treated by doctors with more than 15 years of experience.

- A. `SELECT P.P_Name FROM Patient P INNER JOIN (SELECT P_id, COUNT(DISTINCT Test_Name) AS Num_Tests FROM Test GROUP BY P_id HAVING Num_Tests > 5) AS Subquery ON P.P_id = Subquery.P_id INNER JOIN Treatment T ON P.P_id = T.P_id INNER JOIN Doctor D ON T.Dr_id = D.Dr_id WHERE D.Dr_Experience <= 15;`
- B. `SELECT P.P_Name FROM Patient P INNER JOIN (SELECT P_id, COUNT(DISTINCT Test_Name) AS Num_Tests FROM Test GROUP BY P_id HAVING Num_Tests <= 5) AS Subquery ON P.P_id = Subquery.P_id INNER JOIN Treatment T ON P.P_id = T.P_id INNER JOIN Doctor D ON T.Dr_id = D.Dr_id WHERE D.Dr_Experience > 15;`
- C. `SELECT P.P_Name FROM Patient P INNER JOIN (SELECT P_id, COUNT(DISTINCT Test_Name) AS Num_Tests FROM Test GROUP BY P_id HAVING Num_Tests > 5) AS Subquery ON P.P_id = Subquery.P_id INNER JOIN Treatment T ON P.P_id = T.P_id INNER JOIN Doctor D ON T.Dr_id = D.Dr_id WHERE D.Dr_Experience > 15;`

D. `SELECT P.P_Name FROM Patient P INNER JOIN (SELECT P_id, COUNT(DISTINCT Test_Name) AS Num_Tests FROM Test GROUP BY P_id HAVING Num_Tests <= 5) AS Subquery ON P.P_id = Subquery.P_id INNER JOIN Treatment T ON P.P_id = T.P_id INNER JOIN Doctor D ON T.Dr_id = D.Dr_id WHERE D.Dr_Experience <= 15;`

Q 58. List the employees who have worked on projects with 'Database' in the project name and are designated as 'Developer.'

A. `SELECT Emp_Name FROM Employee WHERE Emp_ID IN (SELECT Emp_ID FROM Work WHERE Project_id IN (SELECT Project_ID FROM Project WHERE Project_Name LIKE '%Database%')) AND Emp_Designation = 'Developer'`

B. `SELECT Emp_Name FROM Employee WHERE Emp_ID IN (SELECT Emp_ID FROM Work WHERE Project_id IN (SELECT Project_ID FROM Project WHERE Project_Name LIKE '%Database%')) AND Emp_Designation = 'Developer'`

C. `SELECT Emp_Name FROM Employee WHERE Emp_ID IN (SELECT Emp_ID FROM Work WHERE Project_id IN (SELECT Project_ID FROM Project WHERE Project_Name = 'Database') AND Emp_Designation = 'Developer')`

D. `SELECT Emp_Name FROM Employee WHERE Emp_ID IN (SELECT Emp_ID FROM Work WHERE Project_id IN (SELECT Project_ID FROM Project WHERE Project_Name = 'Database')) AND Emp_Designation = 'Developer'`

Q 59. List all employees who have worked on projects with a total duration of more than 180 days and have designations of 'Manager' or 'Senior Developer.'

A. `SELECT Emp_Name FROM Employee WHERE Emp_ID IN (SELECT Emp_ID FROM Work GROUP BY Emp_ID HAVING SUM(Project_Duration) > 180) AND (Emp_Designation = 'Junior Developer' OR Emp_Designation = 'Manager')`

B. `SELECT Emp_Name FROM Employee WHERE Emp_ID IN (SELECT Emp_ID FROM Work GROUP BY Emp_ID HAVING SUM(Project_Duration) > 180) AND (Emp_Designation = 'Senior Developer' OR Emp_Designation = 'Junior Developer')`

C. `SELECT Emp_Name FROM Employee WHERE Emp_ID IN (SELECT Emp_ID FROM Work GROUP BY Emp_ID HAVING SUM(Project_Duration) > 180) AND (Emp_Designation = 'Senior Developer' OR Emp_Designation = 'Manager')`

D. `SELECT Emp_Name FROM Employee WHERE Emp_ID IN (SELECT Emp_ID FROM Work GROUP BY Emp_ID HAVING SUM(Project_Duration) > 180) AND (Emp_Designation = 'Manager' OR Emp_Designation = 'Junior Developer')`

Q 60. Find the employees who have the same expertise as another employee with Emp\_ID 101.

- A. `SELECT Emp_Name FROM Employee WHERE Emp_Expertiese IN (SELECT DISTINCT Emp_Expertiese FROM Employee WHERE Emp_ID = 101) AND Emp_ID <> 101`
- B. `SELECT Emp_Name FROM Employee WHERE Emp_Expertiese IN (SELECT Emp_Expertiese FROM Employee WHERE Emp_ID = 101) AND Emp_ID <> 101`
- C. `SELECT Emp_Name FROM Employee WHERE Emp_Expertiese = (SELECT Emp_Expertiese FROM Employee WHERE Emp_ID = 101) AND Emp_ID <> 101`
- D. `SELECT Emp_Name FROM Employee WHERE Emp_Expertiese = (SELECT DISTINCT Emp_Expertiese FROM Employee WHERE Emp_ID = 101) AND Emp_ID <> 101`

Q 61. Calculate the average years of experience for all employees and categorize them as 'Junior,' 'Intermediate,' or 'Senior.'

- A. `SELECT AVG(Emp_Expertiese) AS Average_Experience, CASE WHEN AVG(Emp_Expertiese) < 5 THEN 'Junior' WHEN AVG(Emp_Expertiese) >= 5 AND AVG(Emp_Expertiese) <= 10 THEN 'Intermediate' ELSE 'Senior' END AS Experience_Category FROM Employee`
- B. `SELECT AVG(Emp_Expertiese) AS Average_Experience, CASE WHEN AVG(Emp_Expertiese) < 5 THEN 'Junior' WHEN AVG(Emp_Expertiese) >= 5 AND AVG(Emp_Expertiese) <= 9 THEN 'Intermediate' ELSE 'Senior' END AS Experience_Category FROM Employee`
- C. `SELECT AVG(Emp_Expertiese) AS Average_Experience, CASE WHEN AVG(Emp_Expertiese) < 5 THEN 'Junior' WHEN AVG(Emp_Expertiese) >= 5 AND AVG(Emp_Expertiese) <= 15 THEN 'Intermediate' ELSE 'Senior' END AS Experience_Category FROM Employee`
- D. `SELECT AVG(Emp_Expertiese) AS Average_Experience, CASE WHEN AVG(Emp_Expertiese) < 6 THEN 'Junior' WHEN AVG(Emp_Expertiese) >= 6 AND AVG(Emp_Expertiese) <= 10 THEN 'Intermediate' ELSE 'Senior' END AS Experience_Category FROM Employee`

Q 62. List the patients who have had more tests than their age and show the total number of tests they've undergone.

- A. `SELECT P.P_Name, COUNT(T.Test_id) AS Total_Tests FROM Patient P INNER JOIN Test T ON P.P_id = T.P_id WHERE COUNT(T.Test_id) >= YEAR(CURRENT_DATE) - YEAR(P.P_Birthdate) GROUP BY P.P_Name;`

B. SELECT P.P\_Name, COUNT(T.Test\_id) AS Total\_Tests FROM Patient P INNER JOIN Test T ON P.P\_id = T.P\_id WHERE COUNT(T.Test\_id) <= YEAR(CURRENT\_DATE) - YEAR(P.P\_Birthdate) GROUP BY P.P\_Name;

C. SELECT P.P\_Name, COUNT(T.Test\_id) AS Total\_Tests FROM Patient P INNER JOIN Test T ON P.P\_id = T.P\_id WHERE COUNT(T.Test\_id) > YEAR(CURRENT\_DATE) - YEAR(P.P\_Birthdate) GROUP BY P.P\_Name;

D. SELECT P.P\_Name, COUNT(T.Test\_id) AS Total\_Tests FROM Patient P INNER JOIN Test T ON P.P\_id = T.P\_id WHERE COUNT(T.Test\_id) = YEAR(CURRENT\_DATE) - YEAR(P.P\_Birthdate) GROUP BY P.P\_Name;

Q 63. Find the doctors who have treated patients with 'Hypertension' and have performed more 'CT Scan' tests than 'X-ray' tests.

A. SELECT D.Dr\_Name FROM Doctor D INNER JOIN Treatment T ON D.Dr\_id = T.Dr\_id INNER JOIN Patient P ON T.P\_id = P.P\_id INNER JOIN Test TS ON P.P\_id = TS.P\_id WHERE P.P\_Disease = 'Hypertension' AND TS.Test\_Name = 'CT Scan' GROUP BY D.Dr\_id, D.Dr\_Name HAVING COUNT(TS.Test\_id) > (SELECT COUNT(TS.Test\_id) FROM Test TS WHERE TS.Test\_Name = 'X-ray');

B. SELECT D.Dr\_Name FROM Doctor D INNER JOIN Treatment T ON D.Dr\_id = T.Dr\_id INNER JOIN Patient P ON T.P\_id = P.P\_id INNER JOIN Test TS ON P.P\_id = TS.P\_id WHERE P.P\_Disease = 'Asthma' AND TS.Test\_Name = 'CT Scan' GROUP BY D.Dr\_id, D.Dr\_Name HAVING COUNT(TS.Test\_id) > (SELECT COUNT(TS.Test\_id) FROM Test TS WHERE TS.Test\_Name = 'X-ray');

C. SELECT D.Dr\_Name FROM Doctor D INNER JOIN Treatment T ON D.Dr\_id = T.Dr\_id INNER JOIN Patient P ON T.P\_id = P.P\_id INNER JOIN Test TS ON P.P\_id = TS.P\_id WHERE P.P\_Disease = 'Hypertension' AND TS.Test\_Name = 'X-ray' GROUP BY D.Dr\_id, D.Dr\_Name HAVING COUNT(TS.Test\_id) > (SELECT COUNT(TS.Test\_id) FROM Test TS WHERE TS.Test\_Name = 'CT Scan');

D. SELECT D.Dr\_Name FROM Doctor D INNER JOIN Treatment T ON D.Dr\_id = T.Dr\_id INNER JOIN Patient P ON T.P\_id = P.P\_id INNER JOIN Test TS ON P.P\_id = TS.P\_id WHERE P.P\_Disease = 'Asthma' AND TS.Test\_Name = 'X-ray' GROUP BY D.Dr\_id, D.Dr\_Name HAVING COUNT(TS.Test\_id) > (SELECT COUNT(TS.Test\_id) FROM Test TS WHERE TS.Test\_Name = 'CT Scan');

Q 64. Calculate the total treatment cost for patients who have been admitted on or after '2023-01-01' and have 'Diabetes.'

A. SELECT T.P\_id, SUM(Treatment\_Cost) AS Total\_Cost FROM Treatment T INNER JOIN Patient P ON T.P\_id = P.P\_id WHERE P.P\_Disease = 'Cancer' AND T.Admit\_Date < '2023-01-01' GROUP BY T.P\_id;

B. SELECT T.P\_id, SUM(Treatment\_Cost) AS Total\_Cost FROM Treatment T INNER JOIN Patient P ON T.P\_id = P.P\_id WHERE P.P\_Disease = 'Diabetes' AND T.Admit\_Date < '2023-01-01' GROUP BY T.P\_id;

C. SELECT T.P\_id, SUM(Treatment\_Cost) AS Total\_Cost FROM Treatment T INNER JOIN Patient P ON T.P\_id = P.P\_id WHERE P.P\_Disease = 'Cancer' AND T.Admit\_Date >= '2023-01-01' GROUP BY T.P\_id;

D. SELECT T.P\_id, SUM(Treatment\_Cost) AS Total\_Cost FROM Treatment T INNER JOIN Patient P ON T.P\_id = P.P\_id WHERE P.P\_Disease = 'Diabetes' AND T.Admit\_Date >= '2023-01-01' GROUP BY T.P\_id;

Q 65. Identify employees who have worked on more than 3 projects and have designations of either 'Senior Developer' or 'Manager,' and grant them a performance bonus of \$1,000.

A. UPDATE Employee SET Emp\_Bonus = 1000 WHERE Emp\_ID IN (SELECT Emp\_ID FROM Work GROUP BY Emp\_ID HAVING COUNT(Project\_ID) > 3) AND (Emp\_Designation = 'Junior Developer' OR Emp\_Designation = 'Manager')

B. UPDATE Employee SET Emp\_Bonus = 1000 WHERE Emp\_ID IN (SELECT Emp\_ID FROM Work GROUP BY Emp\_ID HAVING COUNT(Project\_ID) > 3) AND (Emp\_Designation = 'Senior Developer' OR Emp\_Designation = 'Junior Developer')

C. UPDATE Employee SET Emp\_Bonus = 1000 WHERE Emp\_ID IN (SELECT Emp\_ID FROM Work GROUP BY Emp\_ID HAVING COUNT(Project\_ID) = 3) AND (Emp\_Designation = 'Senior Developer' OR Emp\_Designation = 'Manager')

D. UPDATE Employee SET Emp\_Bonus = 1000 WHERE Emp\_ID IN (SELECT Emp\_ID FROM Work GROUP BY Emp\_ID HAVING COUNT(Project\_ID) > 3) AND (Emp\_Designation = 'Senior Developer' OR Emp\_Designation = 'Manager')

Q 66. List the patients who have been treated by doctors from 'San Francisco' and have had both 'Blood Test' and 'X-ray' tests.

A. SELECT P.P\_Name FROM Patient P WHERE P.P\_City = 'San Francisco' AND P.P\_id IN (SELECT P\_id FROM Test WHERE Test\_Name = 'Blood Test') AND P.P\_id IN (SELECT P\_id FROM Test WHERE Test\_Name = 'X-ray');

B. SELECT P.P\_Name FROM Patient P WHERE P.P\_City = 'San Francisco' AND P.P\_id NOT IN (SELECT P\_id FROM Test WHERE Test\_Name = 'Blood Test') AND P.P\_id NOT IN (SELECT P\_id FROM Test WHERE Test\_Name = 'X-ray');

C. SELECT P.P\_Name FROM Patient P WHERE P.P\_City = 'New York City' AND P.P\_id IN (SELECT P\_id FROM Test WHERE Test\_Name = 'Blood Test') AND P.P\_id IN (SELECT P\_id FROM Test WHERE Test\_Name = 'X-ray');

D. SELECT P.P\_Name FROM Patient P WHERE P.P\_City = 'New York City' AND P.P\_id NOT IN (SELECT P\_id FROM Test WHERE Test\_Name = 'Blood Test') AND P.P\_id NOT IN (SELECT P\_id FROM Test WHERE Test\_Name = 'X-ray');

Q 67. List the projects and their categories, but if the project category is 'Management,' display it as 'MGT,' 'Development' as 'DEV,' and 'Research' as 'RSR.'

A. SELECT Project\_Name, CASE Project\_Categoery WHEN 'Management' THEN 'DEV' WHEN 'Development' THEN 'MGT' WHEN 'Research' THEN 'RSR' ELSE Project\_Categoery END FROM Project

B. SELECT Project\_Name, CASE Project\_Categoery WHEN 'Development' THEN 'DEV' WHEN 'Management' THEN 'RSR' WHEN 'Research' THEN 'MGT' ELSE Project\_Categoery END FROM Project

C. SELECT Project\_Name, CASE Project\_Categoery WHEN 'Management' THEN 'MGT' WHEN 'Development' THEN 'DEV' WHEN 'Research' THEN 'RSR' ELSE Project\_Categoery END FROM Project

D. SELECT Project\_Name, CASE Project\_Categoery WHEN 'Development' THEN 'RSR' WHEN 'Management' THEN 'MGT' WHEN 'Research' THEN 'DEV' ELSE Project\_Categoery END FROM Project

Q 68. Calculate the average number of tests performed on patients from 'Los Angeles' who have had treatments by doctors with more than 10 years of experience.

A. SELECT AVG(COUNT(T.Test\_id)) AS Avg\_Tests FROM Patient P INNER JOIN Treatment T ON P.P\_id = T.P\_id INNER JOIN Doctor D ON T.Dr\_id = D.Dr\_id WHERE P.P\_City = 'New York City' AND D.Dr\_Experience <= 10 GROUP BY P.P\_id;

B. SELECT AVG(COUNT(T.Test\_id)) AS Avg\_Tests FROM Patient P INNER JOIN Treatment T ON P.P\_id = T.P\_id INNER JOIN Doctor D ON T.Dr\_id = D.Dr\_id WHERE P.P\_City = 'Los Angeles' AND D.Dr\_Experience > 10 GROUP BY P.P\_id;

C. SELECT AVG(COUNT(T.Test\_id)) AS Avg\_Tests FROM Patient P INNER JOIN Treatment T ON P.P\_id = T.P\_id INNER JOIN Doctor D ON T.Dr\_id = D.Dr\_id WHERE P.P\_City = 'New York City' AND D.Dr\_Experience > 10 GROUP BY P.P\_id;

D. SELECT AVG(COUNT(T.Test\_id)) AS Avg\_Tests FROM Patient P INNER JOIN Treatment T ON P.P\_id = T.P\_id INNER JOIN Doctor D ON T.Dr\_id = D.Dr\_id WHERE P.P\_City = 'Los Angeles' AND D.Dr\_Experience <= 10 GROUP BY P.P\_id;

Q 69. Identify employees who have the highest salary in their respective departments and update their designations to 'Department Head.'

A. UPDATE Employee SET Emp\_Designation = 'Department Head' WHERE Emp\_Salary = (SELECT MAX(Emp\_Salary) FROM Employee)

B. UPDATE Employee SET Emp\_Designation = 'Department Head' WHERE Emp\_Salary IN (SELECT MAX(Emp\_Salary) FROM Employee)

C. UPDATE Employee SET Emp\_Designation = 'Department Head' WHERE (Department, Emp\_Salary) IN (SELECT Department, MAX(Emp\_Salary) FROM Employee GROUP BY Department)

D. UPDATE Employee SET Emp\_Designation = 'Department Head' WHERE Emp\_Salary IN (SELECT MAX(Emp\_Salary) FROM Employee GROUP BY Department)

Q 70. Update the designation of employees to 'Manager' if they have more than 5 years of experience, 'Senior Developer' if they have more than 10 years of experience, and 'Junior Developer' otherwise.

A. UPDATE Employee SET Emp\_Designation = CASE WHEN Emp\_Experience > 5 THEN 'Manager' WHEN Emp\_Experience > 10 THEN 'Senior Developer' ELSE 'Junior Developer' END

B. UPDATE Employee SET Emp\_Designation = CASE WHEN Emp\_Experience > 10 THEN 'Senior Developer' WHEN Emp\_Experience > 5 THEN 'Manager' ELSE 'Junior Developer' END

C. UPDATE Employee SET Emp\_Designation = CASE WHEN Emp\_Experience > 5 THEN 'Senior Developer' WHEN Emp\_Experience > 10 THEN 'Manager' ELSE 'Junior Developer' END

D. UPDATE Employee SET Emp\_Designation = CASE WHEN Emp\_Experience > 10 THEN 'Manager' WHEN Emp\_Experience > 5 THEN 'Senior Developer' ELSE 'Junior Developer' END