**Assignment – SQL Queries**

**1. Find out the selling cost average for packages developed in Pascal.**

SELECT AVG(SellingCost) AS AverageSellingCost FROM SOFTWARE WHERE Language = 'Pascal';

**2. Display the names and ages of all programmers.**

SELECT Name, Age FROM PROGRAMMER;

**3. Display the names of those who have done the DAP Course.**

SELECT DISTINCT Name

FROM STUDIES

WHERE Course = 'DAP';

**4. Display the names and date of birth of all programmers born in January.**

SELECT Name, DateOfBirth

FROM PROGRAMMER

WHERE MONTH(DateOfBirth) = 1;

**5. What is the highest number of copies sold by a package?**

SELECT MAX(CopiesSold) AS HighestCopiesSold

FROM SOFTWARE;

**6. Display lowest course fee.**

SELECT MIN(Fee) AS LowestCourseFee

FROM STUDIES;

**7. How many programmers have done the PGDCA Course?**

SELECT COUNT(\*) AS NumberOfProgrammers

FROM STUDIES

WHERE Course = 'PGDCA';

**8. How much revenue has been earned through sales of packages developed in C?**

SELECT SUM(SellingCost) AS TotalRevenue

FROM SOFTWARE

WHERE Language = 'C';

**9. Display the details of the software developed by Ramesh.**

SELECT \*

FROM SOFTWARE

WHERE DeveloperName = 'Ramesh';

**10. How many programmers studied at Sabhari?**

SELECT COUNT(\*) AS NumberOfProgrammers

FROM STUDIES

WHERE Institution = 'Sabhari';

**11. Display details of packages whose sales crossed the 2000 mark.**

SELECT \*

FROM SOFTWARE

WHERE SellingCost > 2000;

**12. Display the details of packages for which development costs have been recovered.**

SELECT \*

FROM SOFTWARE

WHERE SellingCost >= DevelopmentCost;

**13. What is the cost of the costliest software development in Basic?**

SELECT MAX(DevelopmentCost) AS CostliestDevelopmentCost

FROM SOFTWARE

WHERE Language = 'Basic';

**14. How many packages have been developed in dBase?**

SELECT COUNT(\*) AS NumberOfPackages

FROM SOFTWARE

WHERE Language = 'dBase';

**15. How many programmers studied in Pragathi?**

SELECT COUNT(\*) AS NumberOfProgrammers

FROM STUDIES

WHERE Institution = 'Pragathi';

**16. How many programmers paid 5000 to 10000 for their course?**

SELECT COUNT(\*) AS NumberOfProgrammers

FROM STUDIES

WHERE Fee BETWEEN 5000 AND 10000;

**17. What is the average course fee?**

SELECT AVG(Fee) AS AverageCourseFee

FROM STUDIES;

**18. Display the details of the programmers knowing C.**

SELECT \*

FROM PROGRAMMER

WHERE Skills LIKE '%C%';

**19. How many programmers know either COBOL or Pascal?**

SELECT COUNT(\*) AS NumberOfProgrammers

FROM PROGRAMMER

WHERE Skills LIKE '%COBOL%' OR Skills LIKE '%Pascal%';

**20. How many programmers don’t know Pascal and C?**

SELECT COUNT(\*) AS NumberOfProgrammers

FROM PROGRAMMER

WHERE Skills NOT LIKE '%Pascal%'

AND Skills NOT LIKE '%C%';

**21. How old is the oldest male programmer?**

SELECT MAX(Age) AS OldestMaleProgrammerAge

FROM PROGRAMMER

WHERE Gender = 'Male';

**22. What is the average age of female programmers?**

SELECT AVG(Age) AS AverageFemaleProgrammerAge

FROM PROGRAMMER

WHERE Gender = 'Female';

**23. Calculate the experience in years for each programmer and display with their names in descending order.**

SELECT Name,

DATEDIFF(YEAR, StartDate, '2023-10-01') AS ExperienceInYears

FROM PROGRAMMER

ORDER BY ExperienceInYears DESC;

**24. Who are the programmers who celebrate their birthdays during the current month?**

SELECT Name

FROM PROGRAMMER

WHERE MONTH(DateOfBirth) = MONTH(CURRENT\_DATE())

AND YEAR(DateOfBirth) = YEAR(CURRENT\_DATE());

**25. How many female programmers are there?**

SELECT COUNT(\*) AS NumberOfFemaleProgrammers

FROM PROGRAMMER

WHERE Gender = 'Female';

**26. What are the languages studied by male programmers?**

SELECT DISTINCT Skills

FROM PROGRAMMER

WHERE Gender = 'Male';

**27. What is the average salary?**

SELECT AVG(Salary) AS AverageSalary

FROM SALARY;

**28. How many people draw a salary between 2000 to 4000?**

SELECT COUNT(\*) AS NumberOfPeople

FROM SALARY

WHERE Salary BETWEEN 2000 AND 4000;

**29. Display the details of those who don’t know Clipper, COBOL or Pascal.**

SELECT \*

FROM PROGRAMMER

WHERE Skills NOT LIKE '%Clipper%'

AND Skills NOT LIKE '%COBOL%'

AND Skills NOT LIKE '%Pascal%';

**30. Display the cost of packages developed by each programmer.**

SELECT DeveloperName, SUM(DevelopmentCost) AS TotalDevelopmentCost

FROM SOFTWARE

GROUP BY DeveloperName;

**31. Display the sales value of the packages developed by each programmer.**

SELECT DeveloperName, SUM(SellingCost) AS TotalSalesValue

FROM SOFTWARE

GROUP BY DeveloperName;

**32. Display the number of packages sold by each programmer.**

SELECT DeveloperName, SUM(CopiesSold) AS TotalPackagesSold

FROM SOFTWARE

GROUP BY DeveloperName;

**33. Display the sales cost of the packages developed by each programmer language wise.**

SELECT DeveloperName, Language, SUM(SellingCost) AS TotalSalesCost

FROM SOFTWARE

GROUP BY DeveloperName, Language

ORDER BY DeveloperName, Language;

**34. Display each language name with the average development cost, average selling cost and average price per copy.**

SELECT

Language,

AVG(DevelopmentCost) AS AvgDevelopmentCost,

AVG(SellingCost) AS AvgSellingCost,

AVG(SellingCost / CopiesSold) AS AvgPricePerCopy

FROM SOFTWARE

GROUP BY Language;

**35. Display each programmer’s name and the costliest and cheapest packages developed by him or her.**

SELECT

DeveloperName,

MAX(DevelopmentCost) AS CostliestPackage,

MIN(DevelopmentCost) AS CheapestPackage

FROM SOFTWARE

GROUP BY DeveloperName;

**36. Display each institute’s name with the number of courses and the average cost per course.**

SELECT

Institution,

COUNT(\*) AS NumberOfCourses,

AVG(Fee) AS AvgCostPerCourse

FROM STUDIES

GROUP BY Institution;

**37. Display each institute’s name with the number of students.**

SELECT Institution, COUNT(\*) AS NumberOfStudents

FROM STUDIES

GROUP BY Institution;

**38. Display names of male and female programmers along with their gender.**

SELECT Name, Gender FROM PROGRAMMER;

**39. Display the name of programmers and their packages.**

SELECT P.Name AS ProgrammerName, S.PackageName AS PackageName

FROM PROGRAMMER P

JOIN SOFTWARE S ON P.Name = S.DeveloperName;

**40. Display the number of packages in each language except C and C++.**

SELECT Language, COUNT(\*) AS NumberOfPackages

FROM SOFTWARE

WHERE Language NOT IN ('C', 'C++')

GROUP BY Language;

**41. Display the number of packages in each language for which development cost is less than 1000.**

SELECT Language, COUNT(\*) AS NumberOfPackages

FROM SOFTWARE

WHERE DevelopmentCost < 1000

GROUP BY Language;

**42. Display the average difference between SCOST and DCOST for each package.**

SELECT

PackageName,

AVG(SCOST - DCOST) AS AvgCostDifference

FROM SOFTWARE

GROUP BY PackageName;

**43. Display the total SCOST, DCOST and the amount to be recovered for each programmer whose cost has not yet been recovered.**

SELECT

DeveloperName,

SUM(SCOST) AS TotalSCOST,

SUM(DCOST) AS TotalDCOST,

SUM(SCOST - DCOST) AS AmountToBeRecovered

FROM SOFTWARE

GROUP BY DeveloperName

HAVING SUM(SCOST - DCOST) > 0;

**44. Display the highest, lowest and average salaries for those earning more than 2000.**

SELECT

MAX(Salary) AS HighestSalary,

MIN(Salary) AS LowestSalary,

AVG(Salary) AS AverageSalary

FROM SALARY

WHERE Salary > 2000;

**45. Who is the highest paid C programmer?**

SELECT P.Name AS ProgrammerName, S.Salary

FROM PROGRAMMER P

JOIN SALARY S ON P.Name = S.Name

WHERE P.Skills LIKE '%C%'

ORDER BY S.Salary DESC

LIMIT 1;

**46. Who is the highest paid female COBOL programmer?**

SELECT P.Name AS ProgrammerName, S.Salary

FROM PROGRAMMER P

JOIN SALARY S ON P.Name = S.Name

WHERE P.Gender = 'Female' AND P.Skills LIKE '%COBOL%'

ORDER BY S.Salary DESC

LIMIT 1;

**47. Display the names of the highest paid programmers for each language.**

WITH MaxSalaries AS (

SELECT

P.Skills AS Language,

MAX(S.Salary) AS MaxSalary

FROM PROGRAMMER P

JOIN SALARY S ON P.Name = S.Name

GROUP BY P.Skills

)

SELECT P.Name AS ProgrammerName, M.Language, M.MaxSalary

FROM PROGRAMMER P

JOIN SALARY S ON P.Name = S.Name

JOIN MaxSalaries M ON P.Skills = M.Language AND S.Salary = M.MaxSalary;

**48. Who is the least experienced programmer?**

SELECT Name AS LeastExperiencedProgrammer

FROM PROGRAMMER

ORDER BY HireDate ASC

LIMIT 1;

**49. Who is the most experienced male programmer knowing PASCAL?**

SELECT Name AS MostExperiencedMalePascalProgrammer

FROM PROGRAMMER

WHERE Gender = 'Male' AND Skills LIKE '%PASCAL%'

ORDER BY Experience DESC

LIMIT 1;

**50. Which language is known by only one programmer?**

SELECT Skills AS Language, COUNT(\*) AS NumberOfProgrammers

FROM PROGRAMMER

GROUP BY Skills

HAVING COUNT(\*) = 1;

**51. Who is the above programmer referred in 50?**

SELECT P.Name AS ProgrammerName

FROM PROGRAMMER P

WHERE P.Skills = (

SELECT Skills

FROM PROGRAMMER

GROUP BY Skills

HAVING COUNT(\*) = 1

);

**52. Who is the youngest programmer knowing dBase?**

SELECT Name AS YoungestDBaseProgrammer

FROM PROGRAMMER

WHERE Skills LIKE '%dBase%'

ORDER BY DOB DESC

LIMIT 1;

**53. Which female programmer earning more than 3000 does not know C, C++, Oracle or dBase?**

SELECT Name AS FemaleProgrammer

FROM PROGRAMMER

WHERE Gender = 'Female'

AND Salary > 3000

AND Skills NOT LIKE '%C%'

AND Skills NOT LIKE '%C++%'

AND Skills NOT LIKE '%Oracle%'

AND Skills NOT LIKE '%dBase%';

**54. Which institute has the most number of students?**

SELECT Institution, COUNT(\*) AS NumberOfStudents

FROM STUDIES

GROUP BY Institution

ORDER BY NumberOfStudents DESC

LIMIT 1;

**55. What is the costliest course?**

SELECT CourseName AS CostliestCourse

FROM STUDIES

ORDER BY CourseFee DESC

LIMIT 1;

**56. Which course has been done by the most number of students?**

SELECT CourseName AS MostPopularCourse, COUNT(\*) AS NumberOfStudents

FROM STUDIES

GROUP BY CourseName

ORDER BY NumberOfStudents DESC

LIMIT 1;

**57. Which institute conducts the costliest course?**

SELECT C.Institution AS ConductingInstitute, C.CourseName AS CostliestCourse

FROM STUDIES S

JOIN COURSE C ON S.CourseName = C.CourseName

ORDER BY C.CourseFee DESC

LIMIT 1;

**58. Display the name of the institute and the course which has below average course fee.**

WITH AvgCourseFee AS (

SELECT AVG(CourseFee) AS AverageFee

FROM COURSE

)

SELECT S.Institution AS Institute, S.CourseName AS CourseName

FROM COURSE S, AvgCourseFee

WHERE S.CourseFee < AvgCourseFee.AverageFee;

**59. Display the names of the courses whose fees are within 1000 (+ or -) of the average fee.**

WITH AvgCourseFee AS (

SELECT AVG(CourseFee) AS AverageFee

FROM COURSE

)

SELECT CourseName

FROM COURSE, AvgCourseFee

WHERE CourseFee BETWEEN (AvgCourseFee.AverageFee - 1000) AND (AvgCourseFee.AverageFee + 1000);

**60. Which package has the highest development cost?**

SELECT PackageName AS CostliestPackage

FROM SOFTWARE

ORDER BY DCOST DESC

LIMIT 1;

**61. Which course has below average number of students?**

WITH AvgNumberOfStudents AS (

SELECT AVG(NumberOfStudents) AS AverageStudents

FROM STUDIES

)

SELECT CourseName

FROM STUDIES, AvgNumberOfStudents

WHERE NumberOfStudents < AvgNumberOfStudents.AverageStudents;

**62. Which package has the lowest selling cost?**

SELECT PackageName AS CheapestPackage

FROM SOFTWARE

ORDER BY SCOST ASC

LIMIT 1;

**63. Who developed the package that has sold the least number of copies?**

SELECT DeveloperName

FROM SOFTWARE

WHERE CopiesSold = (

SELECT MIN(CopiesSold)

FROM SOFTWARE

);

**64. Which language has been used to develop the package which has the highest sales amount?**

SELECT LanguageUsed

FROM SOFTWARE

WHERE SalesAmount = (

SELECT MAX(SalesAmount)

FROM SOFTWARE

);

**65. How many copies of the package that has the least difference between development and selling cost were sold?**

SELECT CopiesSold

FROM SOFTWARE

WHERE DifferenceBetweenCosts = (

SELECT MIN(DifferenceBetweenCosts)

FROM SOFTWARE

);

**66. Which is the costliest package developed in Pascal?**

SELECT PackageName AS CostliestPascalPackage

FROM SOFTWARE

WHERE LanguageUsed = 'Pascal'

ORDER BY DCOST DESC

LIMIT 1;

**67. Which language was used to develop the most number of packages?**

SELECT LanguageUsed, COUNT(\*) AS PackageCount

FROM SOFTWARE

GROUP BY LanguageUsed

ORDER BY PackageCount DESC

LIMIT 1;

**68. Which programmer has developed the highest number of packages?**

SELECT DeveloperName, COUNT(\*) AS PackageCount

FROM SOFTWARE

GROUP BY DeveloperName

ORDER BY PackageCount DESC

LIMIT 1;

**69. Who is the author of the costliest package?**

SELECT DeveloperName AS AuthorOfCostliestPackage

FROM SOFTWARE

ORDER BY SCOST DESC

LIMIT 1;

**70. Display the names of the packages which have sold less than the average number of copies.**

WITH AvgCopiesSold AS (

SELECT AVG(CopiesSold) AS AverageCopies

FROM SOFTWARE

)

SELECT PackageName

FROM SOFTWARE, AvgCopiesSold

WHERE CopiesSold < AvgCopiesSold.AverageCopies;

**71. Who are the authors of the packages which have recovered more than double the development cost?**

SELECT DeveloperName

FROM SOFTWARE

WHERE (SCOST - DCOST) > (2 \* DCOST);

**72. Display the programmer names and the cheapest packages developed by them in each language.**

SELECT

P.programmer\_name,

S.language,

MIN(S.package\_cost) AS cheapest\_package\_cost

FROM

PROGRAMMER P

JOIN

SOFTWARE S ON P.programmer\_id = S.programmer\_id

GROUP BY

P.programmer\_name, S.language;

**73. Display the language used by each programmer to develop the highest selling and lowest selling package.**

WITH HighestSelling AS (

SELECT

P.programmer\_name,

S.language AS highest\_selling\_language,

S.package\_name AS highest\_selling\_package,

S.sales AS highest\_selling\_sales

FROM

PROGRAMMER P

JOIN

SOFTWARE S ON P.programmer\_id = S.programmer\_id

WHERE

S.sales = (

SELECT MAX(sales)

FROM SOFTWARE

WHERE programmer\_id = S.programmer\_id

)

),

LowestSelling AS (

SELECT

P.programmer\_name,

S.language AS lowest\_selling\_language,

S.package\_name AS lowest\_selling\_package,

S.sales AS lowest\_selling\_sales

FROM

PROGRAMMER P

JOIN

SOFTWARE S ON P.programmer\_id = S.programmer\_id

WHERE

S.sales = (

SELECT MIN(sales)

FROM SOFTWARE

WHERE programmer\_id = S.programmer\_id

)

)

SELECT

H.programmer\_name,

H.highest\_selling\_language,

H.highest\_selling\_package AS highest\_selling\_package,

L.lowest\_selling\_language,

L.lowest\_selling\_package AS lowest\_selling\_package

FROM

HighestSelling H

JOIN

LowestSelling L ON H.programmer\_name = L.programmer\_name;

**74. Who is the youngest male programmer born in 1965?**

SELECT TOP 1

programmer\_name

FROM

PROGRAMMER

WHERE

gender = 'Male'

AND YEAR(date\_of\_birth) = 1965

ORDER BY

date\_of\_birth DESC;

**75. Who is the oldest female programmer who joined in 1992?**

SELECT TOP 1

programmer\_name

FROM

PROGRAMMER

WHERE

gender = 'Female'

AND YEAR(joining\_date) = 1992

ORDER BY

date\_of\_birth ASC;

**76. In which year was the most number of programmers born?**

SELECT TOP 1

YEAR(date\_of\_birth) AS birth\_year,

COUNT(\*) AS programmer\_count

FROM

PROGRAMMER

GROUP BY

YEAR(date\_of\_birth)

ORDER BY

programmer\_count DESC;

**77. In which month did the most number of programmers join?**

SELECT TOP 1

MONTH(joining\_date) AS joining\_month,

COUNT(\*) AS programmer\_count

FROM

PROGRAMMER

GROUP BY

MONTH(joining\_date)

ORDER BY

programmer\_count DESC;

**78. In which language are most of the programmer’s proficient?**

SELECT TOP 1

language\_proficiency AS proficient\_language,

COUNT(\*) AS programmer\_count

FROM

PROGRAMMER

GROUP BY

language\_proficiency

ORDER BY

programmer\_count DESC;

**79. Who are the male programmers earning below the average salary of female programmers?**

SELECT

programmer\_name,

gender,

salary

FROM

PROGRAMMER

WHERE

gender = 'Male'

AND salary < (

SELECT AVG(salary)

FROM PROGRAMMER

WHERE gender = 'Female'

);

**80. Who are the female programmers earning more than the highest paid?**

SELECT

programmer\_name,

gender,

salary

FROM

PROGRAMMER

WHERE

gender = 'Female'

AND salary > (

SELECT MAX(salary)

FROM PROGRAMMER

);

**81. Which language has been stated as the proficiency by most of the programmers?**

SELECT TOP 1

language\_proficiency,

COUNT(\*) AS proficiency\_count

FROM

PROGRAMMER

GROUP BY

language\_proficiency

ORDER BY

proficiency\_count DESC;

**82. Display the details of those who are drawing the same salary.**

SELECT

programmer\_name,

salary

FROM

PROGRAMMER

WHERE

salary IN (

SELECT salary

FROM PROGRAMMER

GROUP BY salary

HAVING COUNT(\*) > 1

)

ORDER BY

salary;

**83. Display the details of the software developed by the male programmers earning more than 3000.**

SELECT

S.software\_name,

S.language,

S.package\_cost,

P.programmer\_name,

P.salary

FROM

SOFTWARE S

JOIN

PROGRAMMER P ON S.programmer\_id = P.programmer\_id

WHERE

P.gender = 'Male'

AND P.salary > 3000;

**84. Display the details of the packages developed in Pascal by the female programmers.**

SELECT

S.software\_name,

S.package\_cost,

P.programmer\_name,

P.gender

FROM

SOFTWARE S

JOIN

PROGRAMMER P ON S.programmer\_id = P.programmer\_id

WHERE

S.language = 'Pascal'

AND P.gender = 'Female';

**85. Display the details of the programmers who joined before 1990.**

SELECT

programmer\_name,

joining\_date

FROM

PROGRAMMER

WHERE

YEAR(joining\_date) < 1990;

**86. Display the details of the software developed in C by the female programmers at Pragathi.**

SELECT

S.software\_name,

S.package\_cost,

P.programmer\_name,

P.gender,

L.market

FROM

SOFTWARE S

JOIN

PROGRAMMER P ON S.programmer\_id = P.programmer\_id

JOIN

LOCATION L ON P.location\_id = L.location\_id

WHERE

S.language = 'C'

AND P.gender = 'Female'

AND L.market = 'Pragathi';

**87. Display the number of packages, number of copies sold and sales value of each programmer institute wise.**

SELECT

P.institute,

COUNT(S.software\_id) AS number\_of\_packages,

SUM(S.sales) AS number\_of\_copies\_sold,

SUM(S.sales \* S.package\_cost) AS sales\_value

FROM

PROGRAMMER P

JOIN

SOFTWARE S ON P.programmer\_id = S.programmer\_id

JOIN

LOCATION L ON P.location\_id = L.location\_id

GROUP BY

P.institute;

**88. Display the details of the software developed in dBase by male programmers who belong to the institute in which the most number of programmers studied.**

WITH MostPopulousInstitute AS (

SELECT TOP 1

L.institute,

COUNT(\*) AS programmer\_count

FROM

PROGRAMMER P

JOIN

LOCATION L ON P.location\_id = L.location\_id

WHERE

P.gender = 'Male'

GROUP BY

L.institute

ORDER BY

programmer\_count DESC

)

SELECT

S.software\_name,

S.package\_cost,

P.programmer\_name,

P.gender,

L.institute

FROM

SOFTWARE S

JOIN

PROGRAMMER P ON S.programmer\_id = P.programmer\_id

JOIN

LOCATION L ON P.location\_id = L.location\_id

JOIN

MostPopulousInstitute MPI ON L.institute = MPI.institute

WHERE

S.language = 'dBase'

AND P.gender = 'Male';

**89. Display the details of the software developed by the male programmers born before 1965 and female programmers born after 1975.**

SELECT

S.software\_name,

S.language,

S.package\_cost,

P.programmer\_name,

P.gender,

P.date\_of\_birth

FROM

SOFTWARE S

JOIN

PROGRAMMER P ON S.programmer\_id = P.programmer\_id

WHERE

(P.gender = 'Male' AND YEAR(P.date\_of\_birth) < 1965)

OR (P.gender = 'Female' AND YEAR(P.date\_of\_birth) > 1975);

**90. Display the details of the software that has been developed in the language which is neither the first nor the second proficiency of the programmers.**

SELECT

S.software\_name,

S.language,

S.package\_cost,

P.programmer\_name,

P.language\_proficiency AS programmer\_first\_proficiency,

(

SELECT TOP 1

language\_proficiency

FROM

PROGRAMMER P2

WHERE

P2.programmer\_id = P.programmer\_id

AND P2.language\_proficiency <> P.language\_proficiency

ORDER BY

(CASE WHEN P2.language\_proficiency = P.language\_proficiency THEN 1 ELSE 0 END) DESC

) AS programmer\_second\_proficiency

FROM

SOFTWARE S

JOIN

PROGRAMMER P ON S.programmer\_id = P.programmer\_id

WHERE

S.language NOT IN (P.language\_proficiency, programmer\_second\_proficiency);

**91. Display the details of the software developed by the male students at Sabhari.**

SELECT

S.software\_name,

S.language,

S.package\_cost,

P.programmer\_name,

P.gender,

L.market AS location

FROM

SOFTWARE S

JOIN

PROGRAMMER P ON S.programmer\_id = P.programmer\_id

JOIN

LOCATION L ON P.location\_id = L.location\_id

WHERE

P.gender = 'Male'

AND L.market = 'Sabhari';

**92. Display the names of the programmers who have not developed any packages.**

SELECT

P.programmer\_name

FROM

PROGRAMMER P

LEFT JOIN

SOFTWARE S ON P.programmer\_id = S.programmer\_id

WHERE

S.software\_id IS NULL;

**93. What is the total cost of the software developed by the programmers of Apple?**

SELECT

SUM(S.package\_cost) AS total\_cost

FROM

SOFTWARE S

JOIN

PROGRAMMER P ON S.programmer\_id = P.programmer\_id

JOIN

LOCATION L ON P.location\_id = L.location\_id

WHERE

L.market = 'Apple';

**94. Who are the programmers who joined on the same day?**

SELECT

joining\_date,

STRING\_AGG(programmer\_name, ', ') AS joined\_programmers

FROM

PROGRAMMER

GROUP BY

joining\_date

HAVING

COUNT(\*) > 1;

**95. Who are the programmers who have the same Prof2?**

SELECT

Prof2,

STRING\_AGG(programmer\_name, ', ') AS programmers\_with\_same\_Prof2

FROM

PROGRAMMER

GROUP BY

Prof2

HAVING

COUNT(\*) > 1;

**96. Display the total sales value of the software institute wise.**

SELECT

L.institute AS institute\_name,

SUM(S.sales \* S.package\_cost) AS total\_sales\_value

FROM

SOFTWARE S

JOIN

PROGRAMMER P ON S.programmer\_id = P.programmer\_id

JOIN

LOCATION L ON P.location\_id = L.location\_id

GROUP BY

L.institute;

**97. In which institute does the person who developed the costliest package study?**

SELECT TOP 1

L.institute AS institute\_name,

S.software\_name AS costliest\_package\_name,

S.package\_cost AS costliest\_package\_cost

FROM

SOFTWARE S

JOIN

PROGRAMMER P ON S.programmer\_id = P.programmer\_id

JOIN

LOCATION L ON P.location\_id = L.location\_id

ORDER BY

S.package\_cost DESC;

**98. Which language listed in Prof1, Prof2 has not been used to develop any package?**

SELECT DISTINCT

CASE

WHEN 'LanguageA' NOT IN (P.Prof1, P.Prof2) THEN 'LanguageA'

WHEN 'LanguageB' NOT IN (P.Prof1, P.Prof2) THEN 'LanguageB'

WHEN 'LanguageC' NOT IN (P.Prof1, P.Prof2) THEN 'LanguageC'

-- Add more languages as needed

END AS missing\_language

FROM

PROGRAMMER P

LEFT JOIN

SOFTWARE S ON P.programmer\_id = S.programmer\_id

WHERE

S.programmer\_id IS NULL;

**99. How much does the person who developed the highest selling package earn and what course did he/she undergo?**

SELECT TOP 1

P.programmer\_name,

P.earnings,

C.course\_name

FROM

SOFTWARE S

JOIN

PROGRAMMER P ON S.programmer\_id = P.programmer\_id

JOIN

COURSE C ON P.course\_id = C.course\_id

ORDER BY

S.sales DESC;

**100. What is the average salary for those whose software sales is more than 50,000?**

SELECT

AVG(P.earnings) AS average\_salary

FROM

PROGRAMMER P

JOIN

(

SELECT DISTINCT

S.programmer\_id

FROM

SOFTWARE S

WHERE

S.sales > 50000

) HighSalesProgrammers ON P.programmer\_id = HighSalesProgrammers.programmer\_id;

**101. How many packages were developed by students who studied in institutes that charge the lowest course fee?**

WITH LowestCourseFee AS (

SELECT

MIN(course\_fee) AS min\_fee

FROM

COURSE

)

SELECT

COUNT(S.software\_id) AS package\_count

FROM

SOFTWARE S

JOIN

PROGRAMMER P ON S.programmer\_id = P.programmer\_id

JOIN

LOCATION L ON P.location\_id = L.location\_id

JOIN

COURSE C ON P.course\_id = C.course\_id

JOIN

LowestCourseFee LC ON C.course\_fee = LC.min\_fee;

**102. How many packages were developed by the person who developed the cheapest package? Where did he/she study?**

SELECT

P.programmer\_name,

COUNT(S.software\_id) AS package\_count,

L.institute AS institute\_name

FROM

SOFTWARE S

JOIN

PROGRAMMER P ON S.programmer\_id = P.programmer\_id

JOIN

LOCATION L ON P.location\_id = L.location\_id

WHERE

S.package\_cost = (

SELECT MIN(package\_cost) FROM SOFTWARE

)

GROUP BY

P.programmer\_name, L.institute;

**103. How many packages were developed by female programmers earning more than the highest paid male programmer?**

WITH HighestMaleEarnings AS (

SELECT

MAX(earnings) AS highest\_earnings

FROM

PROGRAMMER

WHERE

gender = 'Male'

)

SELECT

COUNT(S.software\_id) AS package\_count

FROM

SOFTWARE S

JOIN

PROGRAMMER P ON S.programmer\_id = P.programmer\_id

JOIN

HighestMaleEarnings HME ON 1=1

WHERE

P.gender = 'Female'

AND P.earnings > HME.highest\_earnings;

**104. How many packages are developed by the most experienced programmers from BDPS?**

WITH MostExperiencedProgrammers AS (

SELECT

P.programmer\_id,

P.programmer\_name

FROM

PROGRAMMER P

JOIN

LOCATION L ON P.location\_id = L.location\_id

WHERE

L.institute = 'BDPS'

ORDER BY

P.years\_of\_experience DESC

LIMIT 1

)

SELECT

COUNT(S.software\_id) AS package\_count

FROM

SOFTWARE S

JOIN

MostExperiencedProgrammers MEP ON S.programmer\_id = MEP.programmer\_id;

**105. List the programmers (from the software table) and the institutes they studied at.**

SELECT

P.programmer\_name,

L.institute AS institute\_name

FROM

SOFTWARE S

JOIN

PROGRAMMER P ON S.programmer\_id = P.programmer\_id

JOIN

LOCATION L ON P.location\_id = L.location\_id;

**106. List each PROF with the number of programmers having that PROF and the number of the packages in that PROF.**

SELECT

PROF,

COUNT(DISTINCT P.programmer\_id) AS programmer\_count,

COUNT(S.software\_id) AS package\_count

FROM

PROGRAMMER P

LEFT JOIN

SOFTWARE S ON P.programmer\_id = S.programmer\_id

GROUP BY

PROF

ORDER BY

PROF;

**107. List the programmer names (from the programmer table) and number of packages each has developed.**

SELECT

P.programmer\_name,

COUNT(S.software\_id) AS package\_count

FROM

PROGRAMMER P

LEFT JOIN

SOFTWARE S ON P.programmer\_id = S.programmer\_id

GROUP BY

P.programmer\_name

ORDER BY

P.programmer\_name;