**Assignment for SQL Queries**

**1) Find out the SELLING COST AVERAGE for the packages developed in PASCAL?**

SELECT AVG(scost) AS avg\_selling\_cost

FROM Software

WHERE dev\_in = 'pascal';

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

SELECT name, TRUNC(MONTHS\_BETWEEN(SYSDATE, dob) / 12) AS age

FROM Programmer;

**3) Display the names and ages of all the programmers who have undergone training in DCS course.**

SELECT p.name,year(current\_date()) - year(p.dob) AS Age

FROM programmer p

INNER JOIN studies s on p.name=s.name

WHERE s.course='dcs';

**4) What is the highest numbers of copies sold by a package?**

SELECT dev\_in,MAX(sold) AS higher\_copies

FROM Software

GROUP BY dev\_in

order by higher\_copies desc limit 1;

**5) Display the names and date of birth of all the programmer born in JANUARY.**

SELECT name, dob

FROM Programmer

WHERE MONTH(dob) = 1;

**6) Display lowest course fee.**

SELECT MIN(ccost) AS lowest\_course\_fee

FROM Studies;

**7) How many programmer has done PGDCA course.**

SELECT COUNT(DISTINCT name) AS pgdca\_programmers

FROM Studies

WHERE course = 'pgdca';

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

SELECT SUM(scost \* sold) AS total\_revenue

FROM Software

WHERE dev\_in = 'c';

**9) Display the details of software developed by Ramesh?**

SELECT \*

FROM Software

WHERE name = 'ramesh';

**10) How many programmers studied at SABHARI?**

SELECT COUNT(DISTINCT name) AS sabhari\_programmers

FROM Studies

WHERE splace = 'sabharri';

**11) Display the details of PACKAGES whose sales crossed the 20000 mark.**

SELECT \*

FROM Software

WHERE (scost \* sold) > 20000;

**12) Find out the number of copies which should be sold in order to recover the development cost of each package.**

SELECT title, dcost / scost AS copies\_needed\_to\_recover

FROM Software;

**13) What is the price of the costliest software developed in BASIC?**

SELECT MAX(scost) AS costliest\_software

FROM Software

WHERE dev\_in = 'basic';

**14) Display the details of packages for which development cost has been recovered.**

SELECT \*

FROM Software

WHERE scost \* sold >= dcost;

**15) How many packages were developed in dbase?**

SELECT COUNT(\*) AS dbase\_packages

FROM Software

WHERE dev\_in = 'dbase';

**16) How many programmers studies at paragathi?**

SELECT COUNT(DISTINCT name) AS paragathi\_programmers

FROM Studies

WHERE splace = 'paragathi';

**17) How many programmers paid 5000 to 10000 for their course?**

SELECT COUNT(DISTINCT name) AS programmers\_paid\_5000\_to\_10000

FROM Studies

WHERE ccost BETWEEN 5000 AND 10000;

**18) What is the average course fee?**

SELECT AVG(ccost) AS average\_course\_fee

FROM Studies;

**19) Display the details of programmers knowing c?**

SELECT \*

FROM Programmer

WHERE prof1 = 'c' OR prof2 = 'c';

**20) How many programmers know either Cobol or Pascal?**

SELECT COUNT(DISTINCT name) AS cobol\_or\_pascal\_programmers

FROM Programmer

WHERE prof1 IN ('cobol', 'pascal') OR prof2 IN ('cobol', 'pascal');

**21) How many programmers don't know Pascal & C?**

SELECT COUNT(DISTINCT name) AS programmers\_not\_knowing\_pascal\_or\_c

FROM Programmer

WHERE prof1 NOT IN ('pascal', 'c') AND prof2 NOT IN ('pascal', 'c');

**22) How old is the oldest male programmers?**

SELECT name, TIMESTAMPDIFF(YEAR, dob, CURDATE()) AS age

FROM Programmer

WHERE sex = 'm'

ORDER BY age DESC

LIMIT 1;

**23) What is the average age of female programmers?**

SELECT AVG(TIMESTAMPDIFF(YEAR, dob, CURDATE())) AS average\_female\_age

FROM Programmer

WHERE sex = 'f';

**24) Calculate the experience in years for each programmers and display along with the names in descending order?**

SELECT name, TIMESTAMPDIFF(YEAR, doj, CURDATE()) AS experience\_years

FROM Programmer

ORDER BY experience\_years DESC;

**25) Who are the programmers who celebrate their birthday during the current month?**

SELECT name

FROM Programmer

WHERE MONTH(dob) = MONTH(CURDATE());

**26) How many female programmers are there?**

SELECT COUNT(\*) AS female\_programmers

FROM Programmer

WHERE sex = 'f';

**27) What are the languages known by the male programmers?**

SELECT DISTINCT prof1, prof2

FROM Programmer

WHERE sex = 'm';

**28) What is the Average salary?**

SELECT AVG(salary) AS average\_salary

FROM Programmer;

**29) How many people draw 2000 to 4000?**

SELECT COUNT(\*) AS people\_drawing\_2000\_to\_4000

FROM Programmer

WHERE salary BETWEEN 2000 AND 4000;

**30) Display the details of those who don't know Clipper, Cobol or Pascal?**

SELECT \*

FROM Programmer

WHERE prof1 NOT IN ('clipper', 'cobol', 'pascal') AND prof2 NOT IN ('clipper', 'cobol', 'pascal');

**31) How many Female programmers knowing C are above 24 years of age?**

SELECT COUNT(\*) AS FemaleProgrammers

FROM (

SELECT \*, YEAR(CURDATE()) - YEAR(dob) AS age

FROM Programmer) AS p

WHERE sex = 'Female' AND prof1 = 'C' AND age > 24;

**32) Who are the programmers who will be celebrating their Birthday within a week?**

SELECT name, dob

FROM Programmer

WHERE DAYOFYEAR(dob) BETWEEN DAYOFYEAR(current\_date()) AND DAYOFYEAR(current\_date() + INTERVAL 7 DAY);

**33) Display the details of those with less than a year's experience?**

SELECT \*

FROM Programmer

WHERE YEAR(CURDATE()) - YEAR(doj) < 1;

**34) Display the details of those who will be completing 2 years of service this year?**

SELECT \*

FROM Programmer

WHERE YEAR(CURDATE()) - YEAR(doj) < 2;

**35) Calculate the amount to be recovered for those packages whose development cost has not been recovered?**

SELECT name, title, (dcost - (scost \* sold)) AS AmountToRecover

FROM Software

WHERE (scost \* sold) < dcost;

**36) List the packages which have not been sold so far?**

SELECT \*

FROM Software

WHERE sold = 0;

**37) Find out the cost of the software developed by Mary?**

SELECT scost

FROM Software

WHERE name = 'Mary';

**38) Display the institute’s names from the studies table without duplicates?**

SELECT DISTINCT splace AS InstituteNames

FROM Studies;

**39) How many different courses are mentioned in the studies table?**

SELECT COUNT(DISTINCT course) AS TotalCourses

FROM Studies;

**40) Display the names of the programmers whose names contain 2 occurrences of the letter A?**

SELECT name

FROM Programmers

WHERE name LIKE '%A%A%' AND LENGTH(name) - LENGTH(REPLACE(name, 'A', '')) = 2;

**41) Display the names of programmers whose names contain upto 5 characters?**

SELECT name

FROM Programmer

WHERE LENGTH(name) <= 5;

**42) How many female programmers knowing COBOL have more than 2 years experience?**

SELECT COUNT(\*) AS Experienced\_Females

FROM Programmer

WHERE sex = 'F' AND prof1 = 'COBOL' AND (YEAR(CURDATE()) - YEAR(doj)) > 2;

**43) What is the length of the shortest name in the programmer table?**

SELECT MIN(LENGTH(name)) AS Shortest\_Name\_Length

FROM Programmer;

**44) What is the average development cost of a package developed in COBOL?**

SELECT AVG(dcost) AS Average\_Development\_Cost

FROM Software

WHERE dev\_in = 'COBOL';

**45) Display the name, sex, dob (DD/MM/YY format), doj for all the programmers without using conversion function?**

SELECT name,sex, DATE\_FORMAT(dob, '%d/%m/%y') AS DOB, doj

FROM Programmer;

**46) Who are the programmers who were born on the last day of the month?**

SELECT name

FROM Programmer

WHERE DAY(dob) = last\_day(dob);

**47) What is the amount paid in salaries of the male programmers who do not know Cobol?**

SELECT SUM(salary) AS Total\_Salary

FROM Programmer

WHERE sex = 'M' AND prof1 != 'COBOL';

**48) Display the title, scost, dcost and difference between scost and dcost in descending order of difference?**

SELECT title, scost, dcost, (scost - dcost) AS Difference

FROM Software

ORDER BY Difference DESC;

**49) Display the name, dob, doj of those month of birth and month of joining are same?**

SELECT name, dob, doj

FROM Programmer

WHERE MONTH(dob) = MONTH(doj);

**50) Display the names of the packages whose names contain more than 1 word?**

SELECT title

FROM Software

WHERE title LIKE '% %';

**Query II**

**1) Display THE NUMBER OF packages developed in EACH language.**

SELECT dev\_in AS language, COUNT(\*) AS package\_count

FROM Software

GROUP BY dev\_in;

**2) Display THE NUMBER OF packages developed by EACH person.**

SELECT name, COUNT(\*) AS package\_count

FROM Software

GROUP BY name;

**3) Display THE NUMBER OF male and female programmer.**

SELECT sex, COUNT(\*) AS programmer\_count

FROM Programmer

GROUP BY sex;

**4) Display THE COSTLIEST packages and HIGEST selling developed in EACH language.**

SELECT dev\_in AS language, MAX(dcost) AS costliest\_package, MAX(sold) AS highest\_selling FROM Software

GROUP BY dev\_in;

**5) Display THE NUMBER OF people BORN in EACH YEAR.**

SELECT YEAR(dob) AS birth\_year, COUNT(\*) AS people\_count

FROM Programmer

GROUP BY birth\_year;

**6) Display THE NUMBER OF people JOINED in EACH YEAR.**

SELECT YEAR(doj) AS join\_year, COUNT(\*) AS people\_count

FROM Programmer

GROUP BY join\_year;

**7) Display THE NUMBER OF people BORN in EACH MONTH.**

SELECT MONTH(dob) AS birth\_month, COUNT(\*) AS people\_count

FROM Programmer

GROUP BY birth\_month;

**8) Display THE NUMBER OF people JOINED in EACH MONTH.**

SELECT MONTH(doj) AS join\_month, COUNT(\*) AS people\_count

FROM Programmer

GROUP BY join\_month;

**9) Display the language wise COUNTS of prof1.**

SELECT prof1 AS language, COUNT(\*) AS prof1\_count

FROM Programmer

GROUP BY prof1;

**10) Display the language wise COUNTS of prof2.**

SELECT prof2 AS language, COUNT(\*) AS prof2\_count

FROM Programmer

GROUP BY prof2;

**11) Display THE NUMBER OF people in EACH salary group.**

SELECT FLOOR(salary / 1000) \* 1000 AS salary\_group, COUNT(\*) AS people\_count

FROM Programmer

GROUP BY salary\_group;

**12) Display THE NUMBER OF people who studied in EACH institute.**

SELECT splace AS institute, COUNT(\*) AS people\_count

FROM Studies

GROUP BY splace;

**13) Display THE NUMBER OF people who studied in EACH course.**

SELECT course, COUNT(\*) AS people\_count

FROM Studies

GROUP BY course;

**14) Display the TOTAL development COST of the packages developed in EACH language.**

SELECT dev\_in AS language, SUM(dcost) AS total\_development\_cost

FROM Software

GROUP BY dev\_in;

**15) Display the selling cost of the package developed in EACH language.**

SELECT dev\_in AS language, SUM(scost \* sold) AS total\_selling\_cost

FROM Software

GROUP BY dev\_in;

**16) Display the cost of the package developed by EACH programmer.**

SELECT name, SUM(dcost) AS total\_cost

FROM Software

GROUP BY name;

**17) Display the sales values of the package developed inEACH programmer.**

SELECT name, SUM(dcost) AS total\_cost

FROM Software

GROUP BY name;

**18) Display the NUMBER of packages developed by EACH programmer.**

SELECT name, COUNT(\*) AS package\_count

FROM Software

GROUP BY name;

**19) Display the sales COST of packages developed by EACH programmer language wise.**

SELECT name, dev\_in AS language, SUM(scost \* sold) AS sales\_cost

FROM Software

GROUP BY name, dev\_in;

**20) Display EACH programmers name,costliest package and cheapest packages developed by**

**Him/Her.**

SELECT name, MAX(dcost) AS costliest\_package, MIN(dcost) AS cheapest\_package

FROM Software

GROUP BY name;

**21) Display EACH language name with AVERAGE development cost, AVERAGE cost, selling cost and AVERAGE price per copy.**

SELECT dev\_in AS language,

AVG(dcost) AS avg\_development\_cost,

AVG(scost) AS avg\_selling\_cost,

AVG(scost / sold) AS avg\_price\_per\_copy

FROM Software

GROUP BY dev\_in;

**22) Display EACH institute name with NUMBER of courses, AVERAGE cost per course.**

SELECT splace AS institute, COUNT(course) AS course\_count,

AVG(ccost) AS avg\_cost\_per\_course

FROM Studies

GROUP BY splace;

**23) Display EACH institute name with NUMBER of students.**

SELECT splace AS institute, COUNT(\*) AS student\_count

FROM Studies

GROUP BY splace;

**5**

**24) Display names of male and female programmers.**

SELECT name, sex

FROM Programmer;

**25) Display the programmer's name and their packages.**

SELECT Programmer.name AS programmer\_name, Software.title AS package\_name

FROM Programmer

JOIN Software ON Programmer.name = Software.name;

**26) Display the NUMBER of packages in EACH language.**

SELECT dev\_in AS language, COUNT(\*) AS package\_count

FROM Software

GROUP BY dev\_in;

**27) Display the NUMBER of packages in EACH language for which development cost is less than 1000.**

SELECT dev\_in AS language, COUNT(\*) AS package\_count

FROM Software

WHERE dcost < 1000

GROUP BY dev\_in;

**28) Display the AVERAGE difference BETWEEN scost and dcost for EACH language.**

SELECT dev\_in AS language, AVG(scost - dcost) AS avg\_difference

FROM Software

GROUP BY dev\_in;

**29) Display the TOTAL scost, dcsot and amount TOBE recovered for EACH programmer for**

**whose dcost HAS NOT YET BEEN recovered.**

SELECT name, SUM(scost \* sold) AS total\_selling\_cost,

SUM(dcost) AS total\_development\_cost,

SUM(dcost - (scost \* sold)) AS amount\_to\_be\_recovered

FROM Software

WHERE scost \* sold < dcost

GROUP BY name;

**30) Display highest, lowest and average salaries for THOSE earning MORE than 2000.**

SELECT MAX(salary) AS highest\_salary,

MIN(salary) AS lowest\_salary,

AVG(salary) AS average\_salary

FROM Programmer

WHERE salary > 2000;

**QUERY III**

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

SELECT name, salary

FROM Programmer

WHERE prof1 = 'C' OR prof2 = 'C'

ORDER BY salary DESC

LIMIT 1;

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

SELECT name, salary

FROM Programmer

WHERE sex = 'f' AND (prof1 = 'COBOL' OR prof2 = 'COBOL')

ORDER BY salary DESC

LIMIT 1;

**3. Highest paid programmer for each language (prof1)**

SELECT prof1 AS language, name, MAX(salary) AS max\_salary

FROM Programmer

GROUP BY prof1;

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

SELECT name, doj

FROM Programmer

ORDER BY doj DESC

LIMIT 1;

**5. Who is the most experienced programmer?**

SELECT name, doj

FROM Programmer

ORDER BY doj ASC

LIMIT 1;

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

SELECT language

FROM (

SELECT prof1 AS language, COUNT(\*) AS count

FROM Programmer

GROUP BY prof1

UNION

SELECT prof2 AS language, COUNT(\*) AS count

FROM Programmer

GROUP BY prof2

) AS languages

WHERE count = 1;

**7. Who is the youngest programmer knowing DBASE?**

SELECT name, dob

FROM Programmer

WHERE prof1 = 'DBASE' OR prof2 = 'DBASE'

ORDER BY dob DESC

LIMIT 1;

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

SELECT splace AS institute, COUNT(\*) AS student\_count

FROM Studies

GROUP BY splace

ORDER BY student\_count DESC

LIMIT 1;

**9. Who is the above programmer?**

SELECT name

FROM Studies

WHERE splace = (

SELECT splace

FROM Studies

GROUP BY splace

ORDER BY COUNT(\*) DESC

LIMIT 1

);

**10. Which female programmer earns more than 3000 but does not know C, C++, Oracle, or DBASE?**

SELECT name

FROM Programmer

WHERE sex = 'f' AND salary > 3000

AND prof1 NOT IN ('C', 'C++', 'Oracle', 'DBASE')

AND prof2 NOT IN ('C', 'C++', 'Oracle', 'DBASE');

**11. Which is the costliest course?**

SELECT course, ccost

FROM Studies

ORDER BY ccost DESC

LIMIT 1;

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

SELECT course, COUNT(\*) AS student\_count

FROM Studies

GROUP BY course

ORDER BY student\_count DESC

LIMIT 1;

**13. Name of the institute and course with below average course fee**

SELECT splace, course

FROM Studies

WHERE ccost < (SELECT AVG(ccost) FROM Studies);

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

SELECT splace, course, ccost

FROM Studies

ORDER BY ccost DESC

LIMIT 1;

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

SELECT course

FROM Studies

GROUP BY course

HAVING COUNT(\*) < (

SELECT AVG(student\_count)

FROM (

SELECT course, COUNT(\*) AS student\_count

FROM Studies

GROUP BY course

) AS subquery

);

**16. Which institute conducts the above course?**

SELECT splace

FROM Studies

WHERE course = (

SELECT course

FROM Studies

GROUP BY course

HAVING COUNT(\*) < (

SELECT AVG(student\_count)

FROM (

SELECT course, COUNT(\*) AS student\_count

FROM Studies

GROUP BY course

) AS subquery

)

);

**17. Names of the course whose fees are within 1000 (+ or -) of the average fee**

SELECT course

FROM Studies

WHERE ABS(ccost - (SELECT AVG(ccost) FROM Studies)) <= 1000;

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

SELECT title, dcost

FROM Software

ORDER BY dcost DESC

LIMIT 1;

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

SELECT title, scost

FROM Software

ORDER BY scost ASC

LIMIT 1;

**20. Who developed the package with the least number of copies sold?**

SELECT name, title

FROM Software

ORDER BY sold ASC

LIMIT 1;

**21. Which language was used to develop the package with the highest sales amount?**

SELECT dev\_in AS language

FROM Software

ORDER BY (scost \* sold) DESC

LIMIT 1;

**22. Copies sold of the package with the least difference between development and selling cost**

SELECT sold

FROM Software

ORDER BY ABS(scost - dcost) ASC

LIMIT 1;

**23. Costliest package developed in Pascal**

SELECT title, dcost

FROM Software

WHERE dev\_in = 'Pascal'

ORDER BY dcost DESC

LIMIT 1;

**24. Language used to develop the most number of packages**

SELECT dev\_in AS language, COUNT(\*) AS package\_count

FROM Software

GROUP BY dev\_in

ORDER BY package\_count DESC

LIMIT 1;

**25. Programmer who developed the highest number of packages**

SELECT name, COUNT(\*) AS package\_count

FROM Software

GROUP BY name

ORDER BY package\_count DESC

LIMIT 1;

**26. Author of the costliest package**

SELECT name, title

FROM Software

ORDER BY dcost DESC

LIMIT 1;

**27. Packages sold less than the average number of copies**

SELECT title

FROM Software

WHERE sold < (SELECT AVG(sold) FROM Software);

**28. Female programmers earning more than the highest paid male programmer**

SELECT name

FROM Programmer

WHERE sex = 'f' AND salary > (

SELECT MAX(salary)

FROM Programmer

WHERE sex = 'm'

);

**29. Language stated as prof1 by most programmers**

SELECT prof1 AS language, COUNT(\*) AS count

FROM Programmer

GROUP BY prof1

ORDER BY count DESC

LIMIT 1;

**30. Authors of packages that recovered more than double the development cost**

SELECT name

FROM Software

WHERE (scost \* sold) > 2 \* dcost;

**31. Programmer names and cheapest package developed by them in each language**

SELECT name, dev\_in AS language, MIN(dcost) AS cheapest\_package\_cost

FROM Software

GROUP BY name, dev\_in;

**32. Youngest male programmer born in 1965**

SELECT name, dob

FROM Programmer

WHERE sex = 'm' AND YEAR(dob) = 1965

ORDER BY dob DESC

LIMIT 1;

**33. Language used by each programmer to develop the highest and lowest selling package**

SELECT name,

(SELECT dev\_in FROM Software AS s1 WHERE s1.name = p.name ORDER BY sold DESC LIMIT 1) AS highest\_selling\_language,

(SELECT dev\_in FROM Software AS s2 WHERE s2.name = p.name ORDER BY sold ASC LIMIT 1) AS lowest\_selling\_language

FROM Programmer p;

**34. Oldest female programmer who joined in 1992**

SELECT name, dob

FROM Programmer

WHERE sex = 'f' AND YEAR(doj) = 1992

ORDER BY dob ASC

LIMIT 1;

**35. Year with the most number of programmers born**

SELECT YEAR(dob) AS birth\_year, COUNT(\*) AS programmer\_count

FROM Programmer

GROUP BY birth\_year

ORDER BY programmer\_count DESC

LIMIT 1;

**36. Month with the most number of programmers joining**

SELECT MONTH(doj) AS join\_month, COUNT(\*) AS programmer\_count

FROM Programmer

GROUP BY join\_month

ORDER BY programmer\_count DESC

LIMIT 1;

**37. Language most programmers are proficient in**

SELECT language, COUNT(\*) AS programmer\_count

FROM (

SELECT prof1 AS language FROM Programmer

UNION ALL

SELECT prof2 AS language FROM Programmer

) AS languages

GROUP BY language

ORDER BY programmer\_count DESC

LIMIT 1;

**38. Male programmers earning below the average salary of female programmers**

SELECT name

FROM Programmer

WHERE sex = 'm' AND salary < (

SELECT AVG(salary)

FROM Programmer

WHERE sex = 'f'

);

**QUERY IV**

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

SELECT \*

FROM Programmer

WHERE salary IN (

SELECT salary

FROM Programmer

GROUP BY salary

HAVING COUNT(\*) > 1

);

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

SELECT s.\*

FROM Software s

JOIN Programmer p ON s.name = p.name

WHERE p.sex = 'm' AND p.salary > 3000;

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

SELECT s.\*

FROM Software s

JOIN Programmer p ON s.name = p.name

WHERE s.dev\_in = 'Pascal' AND p.sex = 'f';

**4. Display the details of the programmer who joined before 1990**

SELECT \*

FROM Programmer

WHERE doj < '1990-01-01';

**5. Display details of software developed in C by female programmers of PRAGATHI**

SELECT s.\*

FROM Software s

JOIN Programmer p ON s.name = p.name

JOIN Studies st ON p.name = st.name

WHERE s.dev\_in = 'C' AND p.sex = 'f' AND st.splace = 'PRAGATHI';

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

SELECT st.splace AS institute, p.name, COUNT(s.title) AS package\_count, SUM(s.sold) AS total\_copies, SUM(s.sold \* s.scost) AS sales\_value

FROM Software s

JOIN Programmer p ON s.name = p.name

JOIN Studies st ON p.name = st.name

GROUP BY st.splace, p.name;

**7. Display details of software developed in DBASE by male programmers from the institute with the most programmers**

SELECT s.\*

FROM Software s

JOIN Programmer p ON s.name = p.name

WHERE s.dev\_in = 'DBASE' AND p.sex = 'm' AND p.name IN (

SELECT name

FROM Studies

WHERE splace = (

SELECT splace

FROM Studies

GROUP BY splace

ORDER BY COUNT(\*) DESC

LIMIT 1

)

);

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

SELECT \*

FROM Software

WHERE name IN (

SELECT name

FROM Programmer

WHERE (sex = 'm' AND dob < '1965-01-01') OR (sex = 'f' AND dob > '1975-01-01')

);

**9. Display details of software developed in a language not the programmer’s first proficiency**

SELECT s.\*

FROM Software s

JOIN Programmer p ON s.name = p.name

WHERE s.dev\_in != p.prof1;

**10. Display details of software developed in a language neither first nor second proficiency of the programmer**

SELECT s.\*

FROM Software s

JOIN Programmer p ON s.name = p.name

WHERE s.dev\_in != p.prof1 AND s.dev\_in != p.prof2;

**11. Display details of software developed by male students of SABHARI**

SELECT s.\*

FROM Software s

JOIN Programmer p ON s.name = p.name

JOIN Studies st ON p.name = st.name

WHERE p.sex = 'm' AND st.splace = 'SABHARI';

**12. Display the names of programmers who have not developed any package**

SELECT p.name

FROM Programmer p

LEFT JOIN Software s ON p.name = s.name

WHERE s.name IS NULL;

**13. Total cost of software developed by programmers from APPLE**

SELECT SUM(s.dcost) AS total\_cost

FROM Software s

JOIN Programmer p ON s.name = p.name

JOIN Studies st ON p.name = st.name

WHERE st.splace = 'APPLE';

**14. Programmers who joined on the same day**

SELECT \*

FROM Programmer

WHERE doj IN (

SELECT doj

FROM Programmer

GROUP BY doj

HAVING COUNT(\*) > 1

);

**15. Programmers who have the same prof2**

SELECT \*

FROM Programmer

WHERE prof2 IN (

SELECT prof2

FROM Programmer

GROUP BY prof2

HAVING COUNT(\*) > 1

);

**16. Total sales values of software, institute-wise**

SELECT st.splace AS institute, SUM(s.sold \* s.scost) AS total\_sales

FROM Software s

JOIN Programmer p ON s.name = p.name

JOIN Studies st ON p.name = st.name

GROUP BY st.splace;

**17. Institute where the person who developed the costliest package studied**

SELECT st.splace

FROM Studies st

JOIN Programmer p ON st.name = p.name

JOIN Software s ON p.name = s.name

WHERE s.dcost = (SELECT MAX(dcost) FROM Software);

**18. Language listed in prof1 and prof2 not used to develop any package**

SELECT DISTINCT p.prof1 AS language

FROM Programmer p

WHERE p.prof1 NOT IN (SELECT DISTINCT dev\_in FROM Software)

UNION

SELECT DISTINCT p.prof2

FROM Programmer p

WHERE p.prof2 NOT IN (SELECT DISTINCT dev\_in FROM Software);

**19. Salary and course of the person who developed the highest selling package**

SELECT p.salary, st.course

FROM Programmer p

JOIN Software s ON p.name = s.name

JOIN Studies st ON p.name = st.name

WHERE s.sold = (SELECT MAX(sold) FROM Software);

**20. Months to recover course cost for each programmer**

SELECT p.name, CEIL(st.ccost / p.salary) AS months\_to\_recover

FROM Programmer p

JOIN Studies st ON p.name = st.name;

**21. Costliest package developed by a person with under 3 years' experience**

SELECT \*

FROM Software s

JOIN Programmer p ON s.name = p.name

WHERE TIMESTAMPDIFF(YEAR, p.doj, NOW()) < 3

ORDER BY s.dcost DESC

LIMIT 1;

**22. Average salary of programmers whose software sales value is more than 50,000**

SELECT AVG(p.salary) AS average\_salary

FROM Programmer p

JOIN Software s ON p.name = s.name

GROUP BY p.name

HAVING SUM(s.sold \* s.scost) > 50000;

**23. Packages developed by students from the institute with the lowest course fee**

SELECT COUNT(\*)

FROM Software s

JOIN Programmer p ON s.name = p.name

JOIN Studies st ON p.name = st.name

WHERE st.splace = (SELECT splace FROM Studies ORDER BY ccost ASC LIMIT 1);

**24. Packages by the person who developed the cheapest package and their institute**

SELECT COUNT(s.title) AS package\_count, st.splace

FROM Software s

JOIN Programmer p ON s.name = p.name

JOIN Studies st ON p.name = st.name

WHERE s.dcost = (SELECT MIN(dcost) FROM Software);

**25. Packages by female programmers earning more than the highest paid male programmer**

SELECT COUNT(\*)

FROM Software s

JOIN Programmer p ON s.name = p.name

WHERE p.sex = 'f' AND p.salary > (SELECT MAX(salary) FROM Programmer WHERE sex = 'm');

**26. Packages by most experienced programmers from BDPS**

SELECT COUNT(\*)

FROM Software s

JOIN Programmer p ON s.name = p.name

JOIN Studies st ON p.name = st.name

WHERE st.splace = 'BDPS' AND p.doj = (SELECT MIN(doj) FROM Programmer);

**27. Programmers and institutes they studied, including those who didn't develop any package**

SELECT p.name, st.splace

FROM Programmer p

LEFT JOIN Studies st ON p.name = st.name;

**28. Each prof1 with number of programmers and number of packages developed**

SELECT p.prof1 AS language, COUNT(DISTINCT p.name) AS programmer\_count, COUNT(s.title) AS package\_count

FROM Programmer p

LEFT JOIN Software s ON p.name = s.name AND s.dev\_in = p.prof1

GROUP BY p.prof1;

**29. Programmer names and number of packages they developed**

SELECT p.name, COUNT(s.title) AS package\_count

FROM Programmer p

LEFT JOIN Software s ON p.name = s.name

GROUP BY p.name;

**30. All details of programmers who have done a course at S.S.I.L**

SELECT \*

FROM Programmer p

WHERE p.name IN (

SELECT name

FROM Studies

WHERE splace = 'S.S.I.L'

);