SQL queries (Abitha T from BIT)

QUERIES 1:

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

SELECT AVG(scost) AS selling_cost_average FROM Software WHERE dev_in = 'pascal';

2) Display the names and ages of all programmers.

SELECT name,TIMESTAMPDIFF(YEAR, dob, CURDATE()) AS age FROM Programmer;

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

SELECT NAME, TIMESTAMPDIFF(YEAR, DOB, CURDATE()) AS AGE FROM STUDIES WHERE NAME LIKE '%H';

SELECT p.name,TIMESTAMPDIFF(YEAR, p.dob, CURDATE()) AS age FROM Programmer p JOIN Studies s ON p.name = s.name
WHERE s.course = 'DCS';

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

SELECT MAX(SOLD) FROM SOFTWARE;

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

SELECT name, dob FROM Programmer WHERE MONTH(dob) = 1;

6) Display lowest course fee.

SELECT MIN (CCOST) FROM STUDIES;

7) How many programmers have done PGDCA courses?

SELECT COUNT (NAME) FROM STUDIES WHERE COURSE LIKE 'PGDCA';

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

SELECT SUM (SOLD*SCOST) FROM SOFTWARE WHERE DEV_IN LIKE '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 (NAME) AS PROGRAMMERS FROM STUDIES WHERE SPLACE='SABHARI';

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

SELECT * FROM SOFTWARE WHERE (SOLD*SCOST)>20000;

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

SELECT title, CEIL(dcost / scost) AS copies_to_recover_development_cost FROM Software;

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

SELECT MAX(scost) AS costliest_software_price FROM Software WHERE dev_in = 'BASIC';

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

SELECT * FROM Software WHERE sold * scost >= dcost;

15) How many packages were developed in dbase?

SELECT COUNT (TITLE) AS TOTAL FROM SOFTWARE WHERE DEV_IN='DBASE';

16) How many programmers study at paragathi?

SELECT COUNT (NAME) FROM STUDIES WHERE SPLACE='PRAGATHI';

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

SELECT COUNT (NAME) AS NO_OF_PROGRAMMERS FROM STUDIES WHERE CCOST>=5000 AND CCOST<=10000;

18) What is the average course fee?

SELECT AVG (CCOST) AS AVERAGE COST 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(*) AS cobol_or_pascal_programmers FROM Programmer WHERE prof1 = 'Cobol' OR prof2 = 'Cobol' OR prof1 = 'Pascal' OR prof2 = 'Pascal';

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

SELECT COUNT(*) AS programmers_not_knowing_pascal_and_c FROM Programmer WHERE prof1 NOT IN ('Pascal', 'C') AND prof2 NOT IN ('Pascal', 'C');

22) How old are the oldest male programmers?

SELECT MAX(TIMESTAMPDIFF(YEAR, dob, CURDATE())) AS oldest_male_programmer_age FROM Programmer WHERE sex = 'male';

23) What is the average age of female programmers?

SELECT AVG(TIMESTAMPDIFF(YEAR, dob, CURDATE())) AS average_female_programmer_age FROM Programmer WHERE sex = 'female';

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_in_years FROM Programmer ORDER BY experience_in_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 (NAME) FEMALE_PROG 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 AVGSAL FROM PROGRAMMER;

29) How many people draw 2000 to 4000?

SELECT NAME 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 female_programmers_above_24_knowing_c FROM Programmer WHERE sex = 'female' AND TIMESTAMPDIFF(YEAR, dob, CURDATE()) > 24 AND (prof1 = 'C' OR prof2 = 'C');

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

SELECT name FROM Programmer WHERE DAYOFYEAR(dob) BETWEEN DAYOFYEAR(CURDATE()) AND DAYOFYEAR(CURDATE() + INTERVAL 7 DAY);

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

SELECT * FROM Programmer WHERE TIMESTAMPDIFF(MONTH, doj, CURDATE()) < 12;

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

SELECT * FROM Programmer WHERE YEAR(doj) = YEAR(CURDATE()) - 2;

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

SELECT title, (dcost - (scost * sold)) AS amount_to_be_recovered 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 SUM(scost) FROM Software WHERE name = 'Mary';
```

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

```
SELECT DISTINCT splace FROM Studies;
```

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

```
SELECT COUNT(DISTINCT course) AS different_courses FROM Studies;
```

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

```
SELECT name FROM Programmer WHERE name LIKE '%a%a%';
```

41) Display the names of programmers whose names contain up to 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(*) FROM Programmer WHERE sex = 'female' AND
TIMESTAMPDIFF(YEAR, doj, CURDATE()) > 2 AND (prof1 = 'COBOL' OR prof2 = 'COBOL');
```

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

```
SELECT MIN(LENGTH(NAME)) FROM PROGRAMMER;
```

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

```
SELECT AVG(DCOST) 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, DATE_FORMAT(doj,
'%d/%m/%y') AS 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) = DAY(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 = 'male' AND (prof1 != 'Cobol' AND prof2 != 'Cobol');

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

SELECT TITLE, SCOST, DCOST, DCOST -SCOST DIFF FROM SOFTWARE ORDER BY 4 DESC;

49) Display the name, dob, doj of those months of birth and month of joining are the 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 '% %';

QUERIES 2:

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

SELECT dev_in, COUNT(*) AS num_packages FROM Software GROUP BY dev_in;

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

SELECT name, COUNT(*) AS num_packages FROM Software GROUP BY name;

3) Display THE NUMBER OF male and female programmers.

SELECT sex, COUNT(*) AS num_programmers FROM Programmer GROUP BY sex;

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

SELECT dev_in, MAX(scost) 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 num_people FROM Programmer GROUP BY YEAR(dob);

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

SELECT YEAR(doj) AS join_year, COUNT(*) AS num_people FROM Programmer GROUP BY YEAR(doj);

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

SELECT MONTH(dob) AS birth_month, COUNT(*) AS num_people FROM Programmer GROUP BY MONTH(dob);

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

SELECT MONTH(doj) AS join_month, COUNT(*) AS num_people FROM Programmer GROUP BY MONTH(doj);

9) Display the language wise COUNTS of prof1.

SELECT prof1, COUNT(*) AS num_programmers FROM Programmer GROUP BY prof1;

10) Display the language wise COUNTS of prof2.

SELECT prof2, COUNT(*) AS num_programmers FROM Programmer GROUP BY prof2;

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

SELECT FLOOR(salary / 1000) AS salary_group, COUNT(*) AS num_people FROM Programmer GROUP BY salary_group;

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

SELECT splace, COUNT(*) AS num_people FROM Studies GROUP BY splace;

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

SELECT course, COUNT(*) AS num_people FROM Studies GROUP BY course;

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

SELECT dev_in, 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, SUM(scost) 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_development_cost FROM Software GROUP BY name;

17) Display the sales values of the package developed by EACH programmer.

SELECT name, SUM(sold * scost) AS total_sales_value FROM Software GROUP BY name;

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

SELECT name, COUNT(*) AS num_packages FROM Software GROUP BY name;

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

SELECT name, dev_in, SUM(sold * scost) AS total_sales_cost FROM Software GROUP BY name, dev_in;

20) Display EACH programmers name,costliest package,cheapest packages developed by His/Her?

SELECT name, MAX(scost) AS costliest_package, MIN(scost) 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, AVG(dcost) AS avg_development_cost, AVG(scost) AS avg_cost, AVG(sold * scost) AS avg_selling_cost, AVG(scost) / AVG(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, COUNT(DISTINCT course) AS num_courses, AVG(ccost) AS avg_cost_per_course FROM Studies GROUP BY splace;

23) Display EACH institute name with NUMBER of students.

SELECT SPLACE,COUNT(NAME) FROM STUDIES GROUP BY SPLACE;

24) Display names of male and female programmers.

SELECT NAME,SEX FROM PROGRAMMER ORDER BY SEX;

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

SELECT NAME,TITLE FROM SOFTWARE ORDER BY NAME;

26) Display the NUMBER of packages in EACH language.

SELECT COUNT(TITLE),DEV_IN FROM SOFTWARE GROUP BY DEV_IN;

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

SELECT COUNT(TITLE),DEV_IN FROM SOFTWARE WHERE DCOST<1000 GROUP BY DEV_IN;

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

SELECT DEV_IN,AVG(DCOST - SCOST) 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) AS total_selling_cost, SUM(dcost) AS total_development_cost, SUM(sold * scost) AS amount_to_be_recovered FROM Software WHERE dcost > (sold * scost) 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;

QUERIES 3:

1) Who is the highest paid C programmer?

SELECT name 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 FROM Programmer WHERE sex = 'f' AND (prof1 = 'COBOL' OR prof2 = 'COBOL') ORDER BY salary DESC LIMIT 1;

3) Display the name of the HIGHEST paid programmer for EACH language (prof1)

SELECT dev_in, name FROM Software JOIN Programmer ON Software.name = Programmer.name GROUP BY dev_in ORDER BY scost DESC;

4) Who is the LEAST experienced programmer?

SELECT name FROM Programmer ORDER BY dob LIMIT 1;

5) Who is the MOST experienced programmer?

SELECT name FROM Programmer ORDER BY dob DESC LIMIT 1;

6) Which language is known by ONLY ONE programmer?

SELECT dev_in FROM Software GROUP BY dev_in HAVING COUNT(*) = 1;

7) Who is the YONGEST programmer knowing DBASE?

SELECT name FROM Programmer WHERE prof1 = 'DBASE' OR prof2 = 'DBASE' ORDER BY dob DESC LIMIT 1;

8) Which institute has MOST NUMBER of students?

SELECT splace FROM Studies GROUP BY splace ORDER BY COUNT(*) DESC LIMIT 1;

9) Which female programmer earns MORE than 3000/- but DOESN'T 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');

10) Which is the COSTLIEST course?

SELECT course FROM Studies GROUP BY course ORDER BY AVG(ccost) DESC LIMIT 1;

11) Which course has been done by MOST of the students?

SELECT course FROM Studies GROUP BY course ORDER BY COUNT(*) DESC LIMIT 1;

12) Display name of the institute and course Which has below AVERAGE course fee?

SELECT splace, course FROM Studies GROUP BY splace, course HAVING AVG(ccost) < (AVG(ccost) FROM Studies);

13) Which institute conducts the COSTLIEST course?

SELECT splace FROM Studies GROUP BY splace ORDER BY MAX(ccost) DESC LIMIT 1;

14) Which course has below AVERAGE number of students?

SELECT course FROM Studies GROUP BY course HAVING COUNT(*) < (SELECT AVG(COUNT(*)) FROM Studies GROUP BY course);

15) Which institute conducts the above course?

SELECT splace FROM Studies WHERE course = (SELECT course FROM Studies GROUP BY course HAVING COUNT(*) < (SELECT AVG(COUNT(*)) FROM Studies GROUP BY course)) LIMIT 1;

16) Display names of the course WHOSE fees are within 1000(+ or -) of the AVERAGE fee.

SELECT course FROM Studies GROUP BY course HAVING AVG(ccost) BETWEEN (SELECT AVG(ccost) - 1000 FROM Studies) AND (SELECT AVG(ccost) + 1000 FROM Studies);

17) Which package has the HIGHEST development cost?

SELECT title FROM Software ORDER BY dcost DESC LIMIT 1;

18) Which package has the LOWEST selling cost?

SELECT title FROM Software ORDER BY scost ASC LIMIT 1;

19) Who developed the package, which has sold the LEAST number of copies?

SELECT name FROM Software ORDER BY sold ASC LIMIT 1;

20) Which language was used to develop the package WHICH has the HIGEST sales amount?

SELECT dev_in FROM Software ORDER BY (scost * sold) DESC LIMIT 1;

21) How many copies of the package that has the LEAST DIFFRENCE between development and selling cost were sold?

SELECT sold FROM Software ORDER BY (scost - dcost) ASC LIMIT 1;

22) Which is the COSTLIEAST package developed in PASCAL?

SELECT title FROM Software WHERE dev_in = 'PASCAL' ORDER BY scost DESC LIMIT 1;

23) Which language was used to develop the MOST NUMBER of package?

SELECT dev_in FROM Software GROUP BY dev_in ORDER BY COUNT(*) DESC LIMIT 1;

24) Which programmer has developed the HIGEST NUMBER of package?

SELECT name FROM Software GROUP BY name ORDER BY COUNT(*) DESC LIMIT 1;

25) Who is the author of the COSTLIEST package?

SELECT name FROM Software ORDER BY scost DESC LIMIT 1;

26) Display names of packages WHICH have been sold LESS THAN the AVERAGE number of copies?

SELECT title FROM Software GROUP BY title HAVING SUM(sold) < (SELECT AVG(sold) FROM Software);

27) Who are the female programmers earning MORE than the HIGHEST paid male programmers?

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

28) Which language has been stated as prof1 by MOST of the programmers?

SELECT prof1 FROM Programmer GROUP BY prof1 ORDER BY COUNT(*) DESC LIMIT 1;

29) Who are the authors of packages, WHICH have recovered MORE THAN double the development cost?

SELECT name FROM Software GROUP BY name HAVING SUM(scost * sold) > (2 * SUM(dcost));

30) Display programmer names and CHEAPEST package developed by them in EACH language?

SELECT name, dev_in, MIN(scost) AS cheapest_package_cost FROM Software GROUP BY name, dev_in;

31) Who is the YOUNGEST male programmer born in 1965?

SELECT name FROM Programmer WHERE sex = 'm' AND dob = (SELECT MIN(dob) FROM Programmer WHERE sex = 'm' AND YEAR(dob) = 1965);

32) Display language used by EACH programmer to develop the HIGHEST selling and LOWEST selling package.

SELECT name,
(SELECT dev_in FROM Software WHERE name = p.name ORDER BY sold DESC LIMIT 1) AS language_highest_selling,
(SELECT dev_in FROM Software WHERE name = p.name ORDER BY sold ASC LIMIT 1) AS language_lowest_selling
FROM Programmer p;

33) Who is the OLDEST female programmer WHO joined in 1992?

SELECT name FROM Programmer WHERE sex = 'f' AND YEAR(doj) = 1992 ORDER BY dob ASC LIMIT 1;

34) In WHICH year where the MOST NUMBER of programmer born?

SELECT YEAR(dob) AS birth_year, COUNT(*) AS num_programmers FROM Programmer

GROUP BY birth_year ORDER BY COUNT(*) DESC LIMIT 1;

35) In WHICH month did MOST NUMBRER of programmer join?

SELECT MONTH(doj) AS join_month, COUNT(*) AS num_programmers FROM Programmer GROUP BY join_month ORDER BY COUNT(*) DESC LIMIT 1;

36) In WHICH language are MOST of the programmer's proficient?

SELECT dev_in AS language, COUNT(*) AS num_programmers FROM Software GROUP BY dev_in ORDER BY COUNT(*) DESC LIMIT 1;

37) Who are the 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');

QUEREIS 4:

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 earing MORE than 3000.

select software.* from programmer p,software s where p.name=s.name and salary>3000 and sex='m';

3) Display details of packages developed in PASCAL by female programmers.

SELECT * FROM Software WHERE dev_in = 'PASCAL' AND name IN (SELECT name FROM Programmer WHERE sex = 'f');

4) Display the details of these programmer WHO joined BEFORE 1990.

select * from programmer where to_char(doj,'yy')<90;

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 st.splace = 'PRAGATHI' AND p.sex = 'f' AND s.dev_in = 'C';
```

6) Display NUMBER of packages NUMBER of copies sold and sales value of EACH programmer Institute-wise.

```
SELECT p.name, st.splace, COUNT(*) AS num_packages, SUM(s.sold) AS
total_copies_sold, SUM(s.scost * s.sold) AS total_sales_value
FROM Software s
JOIN Programmer p ON s.name = p.name
JOIN Studies st ON p.name = st.name
GROUP BY p.name, st.splace;
```

7) Display details of software developed in DBASE by male programmers WHO belong to the institute on which MOST NUMBER OF programmers studies.

```
SELECT s.*
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 GROUP BY splace ORDER BY
COUNT(*) DESC LIMIT 1)
AND p.sex = 'm' AND s.dev_in = 'DBASE';
```

8) Display the details of the software that was developed by male programmers born BEFORE 1965 and female programmers born AFTER 1975.

```
SELECT s.*
FROM Software s
JOIN Programmer p ON s.name = p.name
WHERE (p.sex = 'm' AND YEAR(p.dob) < 1965) OR (p.sex = 'f' AND YEAR(p.dob) > 1975);
```

9) Display the details of the software that was developed in the language that is NOT the programmers first proficiency.

```
SELECT s.*
FROM Software s
JOIN Programmer p ON s.name = p.name
```

WHERE s.dev_in NOT IN (p.prof1, p.prof2);

10) Display details of software that was developed in the language which is NITHER first NOR second proficiency of the programmer.

```
SELECT s.* FROM Software s
JOIN Programmer p ON s.name = p.name
WHERE s.dev_in NOT IN (p.prof1, p.prof2) AND s.dev_in NOT IN (p.prof1, 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 st.splace = 'SABHARI' AND p.sex = 'm';
```

12) Display the names of programmers WHO HAVE NOT developed any package.

```
SELECT name FROM Programmer WHERE name NOT IN (SELECT DISTINCT name
FROM Software);
```

13) What is the total cost of the software developed by the programmers by APPLE?

```
SELECT SUM(scost) AS total_cost FROM Software WHERE name IN (SELECT name
FROM Programmer WHERE name LIKE '%APPLE%');
```

14) Who are the programmers WHO JOINED in the same day?

```
SELECT name, doj FROM Programmer GROUP BY doj HAVING COUNT(*) > 1;
```

15) Who are the programmers WHO HAVE THE SAME PROF2?

```
SELECT name, prof2 FROM Programmer GROUP BY prof2 HAVING COUNT(*) > 1;
```

16) Display the total sales values of software, institutes-wise.

```
SELECT st.splace, SUM(s.scost * s.sold) AS total_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;
```

17) In which institutes did the person who developed the COSTLIEST package study?

```
SELECT st.splace
FROM Software s
JOIN Programmer p ON s.name = p.name
JOIN Studies st ON p.name = st.name
WHERE s.scost = (SELECT MAX(scost) FROM Software);
```

18) Which language listed in prof1 and prof2 HAS NOT BEEN used to develop any package?

```
SELECT DISTINCT lang
FROM (
   SELECT prof1 AS lang FROM Programmer
   UNION
   SELECT prof2 AS lang FROM Programmer
) AS langs
WHERE lang NOT IN (SELECT dev_in FROM Software);
```

19) How much does the person WHO developed the HIGHEST selling package earn and WHAT course did he/she undergo?

```
SELECT p.name, 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) How many months will it take for each programmer to recover the cost of the course underwent?

```
SELECT p.name, ROUND(st.ccost / p.salary * 12) AS months_to_recover
FROM Programmer p
JOIN Studies st ON p.name = st.name;
```

21) Which is the COSTLIEST package developed by a person with under 3 years expenence?

```
SELECT title FROM Software WHERE name IN (SELECT name FROM Programmer
WHERE YEAR(CURRENT_DATE()) - YEAR(doj) < 3) ORDER BY scost DESC LIMIT 1;
```

22) What is the AVERAGE salary for those WHOSE software's sales value is more than 50,000?

SELECT AVG(salary) AS avg_salary FROM Programmer WHERE name IN (SELECT name FROM Software GROUP BY name HAVING SUM(scost * sold) > 50000);

23) How many packages were developed by the students WHO studied in the institute that Charge the LOWEST course fee?

SELECT COUNT(*) AS num_packages FROM Software WHERE name IN (SELECT name FROM Studies WHERE ccost = (SELECT MIN(ccost) FROM Studies));

24) How many packages were developed by the person WHO developed the CHEAPEST package. Where did he\she study?

```
SELECT COUNT(*) AS num_packages, st.splace
FROM Software s
JOIN Studies st ON s.name = st.name
WHERE scost = (SELECT MIN(scost) FROM Software)
GROUP BY st.splace;
```

25) How many packages were developed by female programmers earning MORE than the HIGHEST paid male programmer?

```
SELECT COUNT(*) AS num_packages 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) How many packages were developed by the MOST experienced programmers from BDPS.

```
SELECT COUNT(*) AS num_packages FROM Software
WHERE name IN (SELECT name FROM Programmer WHERE sex = 'm' AND dob = (SELECT MAX(dob) FROM Programmer WHERE sex = 'm' AND st.splace = 'BDPS'));
```

27) List the programmers (from software table) and institutes they studied, including those WHO DIDN'T develop any package.

```
SELECT p.name, IFNULL(st.splace, 'No Institute') AS institute
FROM Programmer p
LEFT JOIN Studies st ON p.name = st.name;
```

28) List each profit with the number of programmers having that prof1 and the number of packages developed in that prof1.

```
SELECT prof, COUNT(DISTINCT p.name) AS num_programmers, COUNT(s.name) AS
num_packages
FROM (
    SELECT prof1 AS prof, name FROM Programmer
    UNION ALL
    SELECT prof2 AS prof, name FROM Programmer
) AS profs
LEFT JOIN Software s ON profs.name = s.name
GROUP BY prof;
```

29) List programmer names (from programmer table) and number of packages EACH developed.

```
SELECT name, COUNT(*) AS num_packages FROM Software GROUP BY name;
```

30) List all the details of programmers who has done a course at S.S.I.L.

```
SELECT * FROM Programmer WHERE name IN (SELECT name FROM Studies WHERE
splace = 'S.S.I.L');
```