

TITLE: SQL ASSESSMENT
AUTHOR: RAMAPRABA J
CREATED DATE: 07-04-2024

QUERIES – I

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

```
SELECT AVG(scost) AS average_selling_cost 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 AS programmer_name, TIMESTAMPDIFF(YEAR, dob, CURDATE()) AS age FROM Programmer WHERE name IN (SELECT name FROM Studies WHERE course = 'DCS');
```

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

```
SELECT MAX(sold) AS highest_copies_sold FROM Software;
```

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

```
SELECT DOB, NAME FROM PROGRAMMER WHERE EXTRACT(month from dob) LIKE 'JAN';
```

- 6) Display lowest course fee.**

```
SELECT MIN(ccost) AS lowest_course_fee FROM Studies;
```

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

```
SELECT COUNT(*) 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 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(*) AS sabhari_programmers FROM Studies WHERE splace = 'sabhari';
```

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

```
SELECT * FROM Software WHERE sold > 20000;
```

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

```
SELECT name, CEIL(dcost / scost) AS copies_to_recover_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 >= dcost / scost;
```

15) How many packages were developed in dbase?

```
SELECT COUNT(*) AS dbase_packages_count FROM Software WHERE dev_in = 'dbase';
```

16) How many programmers studies at paragathi?

```
SELECT COUNT(*) AS paragathi_programmers FROM Studies WHERE splace = 'paragathi';
```

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

```
SELECT COUNT(*) AS programmers_paid_between_5000_and_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 p.* FROM Programmer p JOIN Studies s ON p.name = s.name WHERE
s.course = '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 is the oldest male programmers?

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

23) What is the average age of female programmers?

```
SELECT AVG(TIMESTAMPDIFF(YEAR, dob, CURDATE())) AS average_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 AS programmer_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 EXTRACT (MONTH FROM dob) =
EXTRACT (MONTH FROM SYSDATE);
```

26) How many female programmers are there?

```
SELECT COUNT(*) AS female_programmers_count FROM Programmer WHERE  
sex = 'female';
```

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

```
SELECT 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_within_salary_range 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 FROM Programmer WHERE  
sex = 'female' AND TIMESTAMPDIFF(YEAR, dob, CURDATE()) > 24 AND ('C' IN  
(prof1, prof2));
```

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

```
SELECT * FROM programmer WHERE DAY(dob) BETWEEN DAY(CURDATE())  
AND DAY(CURDATE() + INTERVAL 7 DAY) AND MONTH(dob) =  
MONTH(CURDATE());
```

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

```
SELECT name, sex, dob, doj FROM Programmer WHERE TIMESTAMPDIFF(YEAR,  
doj, CURDATE()) < 1;
```

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

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

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

```
SELECT SUM(scost - dcost) AS amount_to_be_recovered FROM Software WHERE  
scost > 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 FROM Studies;
```

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

```
SELECT COUNT(DISTINCT course) AS unique_courses_count 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 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 female_programmers_cobol_experience FROM Programmer  
WHERE sex = 'female' AND TIMESTAMPDIFF(YEAR, doj, CURDATE()) > 2 AND  
( 'COBOL' IN (prof1, prof2));
```

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, 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(LAST_day(dob)=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_paid 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, scost - dcost AS cost_difference FROM Software ORDER  
BY cost_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 EXTRACT(MONTH FROM  
dob) = EXTRACT(MONTH FROM doj);
```

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

```
SELECT title FROM Software WHERE INSTR(title, ' ') > 0;
```

QUERIES - 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 HIGHEST selling developed in EACH language.

```
SELECT dev_in AS language, MAX(scost) AS CostPack, MAX(sold) AS Highselling_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_born FROM Programmer GROUP BY YEAR(dob) ORDER BY birth_year;
```

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

```
SELECT YEAR(doj) AS joined_year, COUNT(*) AS num_people_born FROM Programmer GROUP BY YEAR(dob) ORDER BY joined_year;
```

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

```
SELECT MONTH(dob) AS birth_month, COUNT(*) AS num_people_born FROM Programmer GROUP BY MONTH(dob) ORDER BY birth_month;
```

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

```
SELECT MONTH(doj) AS join_month, COUNT(*) AS num_people_joined FROM Programmer GROUP BY MONTH(doj) ORDER BY join_month;
```

9) Display the language wise COUNTS of prof1.

```
SELECT dev_in AS language, prof1, COUNT(*) AS prof1_count FROM Software GROUP BY dev_in, prof1;
```

10) Display the language wise COUNTS of prof2.

```
SELECT dev_in AS language, prof2, COUNT(*) AS prof2_count FROM Software GROUP BY dev_in, prof2;
```

11) Display THE NUMBER OF people in EACH salary group

```
SELECT salary, COUNT(name) AS people FROM programmer GROUP BY salary
```

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

```
SELECT splace, COUNT(name) AS Institute FROM programmer GROUP BY splace
```

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

```
SELECT course, COUNT(name) AS people FROM programmer 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) AS total_selling_cost FROM Software  
GROUP BY dev_in;
```

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

```
SELECT name, SUM(scost) AS total_package_cost FROM Software GROUP BY  
name;
```

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

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


20) Display EACH programmers name, costliest package and cheapest packages developed by Him/Her

```
SELECT name AS programmer_name, MAX(scost) AS costliest_package_cost,  
MIN(scost) AS cheapest_package_cost 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_name,AVG(dcost) AS average_development_cost,  
AVG(scost) AS average_cost, AVG(scost + dcost) AS avg_selling_cost,AVG((scost +  
dcost) / sold) AS avg_price_per_copyFROM Software GROUP BY dev_in;
```

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

```
SELECT splace AS institute_name, COUNT(course) AS number_of_courses,  
AVG(ccost) AS average_cost_per_course FROM Studies GROUP BY splace;
```

23) Display EACH institute name with NUMBER of students.

```
SELECT splace AS institute_name, COUNT(name) AS number_of_students FROM  
Studies GROUP BY splace;
```

24) Display names of male and female programmers.

```
SELECT name, sex FROM Programmer;
```

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

```
SELECT p.name AS programmer_name, s.name AS package_name FROM  
Programmer p JOIN Software s ON p.name = s.name;
```

26) Display the NUMBER of packages in EACH language.

```
SELECT dev_in AS language, COUNT(title) AS package_count FROM Software  
GROUP BY dev_in; output
```

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

```
SELECT dev_in AS language, COUNT(title) 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 average_cost_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 AS programmer_name, SUM(scost) AS total_selling_cost, SUM(dcost) AS total_development_cost, SUM(scost) - SUM(dcost) AS amount_to_be_recovered FROM Software GROUP BY name HAVING SUM(scost) > SUM(dcost);
```

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 - III

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 = 'female' AND prof1 = 'COBOL' ORDER BY salary DESC LIMIT 1;
```

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

```
SELECT prof1 AS language, name AS highest_paid_programmer FROM Programmer WHERE (prof1, salary) IN (SELECT prof1, MAX(salary) FROM Programmer GROUP BY prof1);
```

4) Who is the LEAST experienced programmer?

```
SELECT name FROM Programmer ORDER BY doj LIMIT 1;
```

5) Who is the MOST experienced programmer?

```
SELECT name FROM Programmer ORDER BY doj DESC LIMIT 1;
```

6) Which language is known by ONLY ONE programmer?

```
SELECT language FROM (SELECT prof1 AS language FROM Programmer
UNION ALL SELECT prof2 AS language FROM Programmer) AS
languageknow_by_onlyone_programmer GROUP BY language HAVING
COUNT(*) = 1;
```

7) Who is the YONGEST programmer knowing DBASE?

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

8) Which institute has MOST NUMBER of students?

```
SELECT splace AS institute_name, COUNT(name) AS student_count FROM
Studies GROUP BY splace ORDER BY student_count DESC LIMIT 1;
```

9) Who is the above programmer?

```
SELECT name AS programmer_name FROM Programmer 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 = 'female' AND salary > 3000
AND prof1 NOT IN ('C', 'C++', 'Oracle', 'Dbase') OR prof2 NOT IN ('C', 'C++',
'Oracle', 'Dbase');
```

11) Which is the COSTLIEST course?

```
SELECT course, MAX(ccost) AS costliest_course FROM Studies;
```

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

```
SELECT course, COUNT(name) AS student_count FROM Studies GROUP BY
course ORDER BY student_count DESC LIMIT 1;
```

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

```
SELECT splace AS institute_name, course FROM Studies WHERE ccost <
(SELECT AVG(ccost) FROM Studies);
```

14) Which institute conducts COSTLIEST course?

```
SELECT splace AS institute_name, MAX( ccost) AS costliest_course FROM
Studies GROUP BY splace LIMIT 1;
```

15) Which course has below AVERAGE number of students?

```
SELECT course FROM Studies GROUP BY course HAVING COUNT(name) <
(SELECT  AVG(student_count)  FROM  (SELECT  COUNT(name)  AS
student_count FROM Studies GROUP BY course) AS avg_counts);
```

16) Which institute conducts the above course?

```
SELECT splace AS institute_name, course FROM Studies WHERE course IN
(SELECT course FROM Studies GROUP BY course HAVING COUNT(name) <
(SELECT  AVG(student_count)  FROM  (SELECT  COUNT(name)  AS
student_count FROM Studies GROUP BY course) AS avg_counts));
```

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

```
SELECT course FROM Studies GROUP BY course HAVING ABS(AVG(ccost) -
ccost) <= 1000;
```

18) Which package has the HIGHEST development cost?

```
SELECT title AS package_name, dcost AS development_cost FROM Software
ORDER BY dcost DESC LIMIT 1;
```

19) Which package has the LOWEST selling cost?

```
SELECT title AS package_name, dcost AS development_cost FROM Software
ORDER BY dcost ASC LIMIT 1;
```

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

```
SELECT name AS developer_name FROM Programmer WHERE name IN
(SELECT dev_in FROM Software ORDER BY sold LIMIT 1);
```

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

```
SELECT dev_in AS language_used FROM Software WHERE sold = (SELECT
MAX(sold) FROM Software);
```

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

```
SELECT sold FROM Software WHERE ABS(scost - dcost) = (SELECT  
MIN(ABS(scost - dcost))FROM Software);
```

23) Which is the COSTLIEAST package developed in PASCAL?

```
SELECT title AS package_name, scost AS software_cost FROM Software  
WHERE dev_in = 'PASCAL' ORDER BY scost DESC LIMIT 1;
```

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

```
SELECT dev_in AS language_used, COUNT(*) AS package_count FROM  
Software GROUP BY dev_in  
ORDER BY package_count DESC LIMIT 1;
```

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

```
SELECT dev_in AS programmer_name, COUNT(*) AS package_count FROM  
Software GROUP BY dev_in ORDER BY package_count DESC LIMIT 1;
```

26) Who is the author of the COSTLIEST package?

```
SELECT name AS programmer_name FROM Programmer WHERE name IN  
(SELECT dev_in FROM Software ORDER BY scost DESC LIMIT 1);
```

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

```
SELECT title AS package_name FROM Software WHERE sold < (SELECT  
AVG(sold) FROM Software);
```

28) Who are the female programmers earning MORE than the HIGEST paid male programmers?

```
SELECT name FROM Programmer WHERE sex = 'female' AND salary >  
(SELECT MAX(salary) FROM Programmer WHERE sex = 'male');
```

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

```
SELECT prof1 AS language, COUNT(*) AS count FROM Programmer GROUP  
BY prof1 ORDER BY count DESC LIMIT 1;
```

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

```
SELECT name AS author FROM Programmer WHERE name IN (SELECT dev_in FROM Software WHERE scost > 2 * dcost);
```

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

```
SELECT name,title FROM Software WHERE dcost IN (SELECT MIN(dcost) FROM Software GROUP BY dev_in)
```

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

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

33) Display language used by EACH programmer to develop the HIGEST selling and LOWEST selling package.

```
SELECT name,dev_in FROM Software WHERE sold IN (SELECT MAX(sold) FROM Software GROUP BY name) UNION SELECT name,dev_in FROM Software WHERE sold IN (SELECT MIN(sold) FROM Software GROUP BY name);
```

34) Who is the OLDEST female programmer WHO joined in 1992

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

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

```
SELECT YEAR(dob) AS birth_year, COUNT(*) AS num_programmers_born FROM Programmer GROUP BY YEAR(dob) ORDER BY COUNT(*) DESC LIMIT 1;
```

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

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

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

```
SELECT prof1 AS language, COUNT(*) AS num_programmers FROM Programmer GROUP BY prof1 ORDER BY COUNT(*) DESC LIMIT 1;
```

38) Who are the male programmers earning BELOW the AVERAGE salary of female programmers?

```
SELECT name AS programmer_name, sex, salary FROM Programmer WHERE  
sex = 'male' AND salary < (SELECT AVG(salary) FROM Programmer WHERE  
sex = 'female');
```

QUERIES – IV

1) Who is the highest paid C programmer?

```
SELECT name FROM Programmer WHERE prof1 = 'C' or prof='C' ORDER BY  
salary DESC LIMIT 1;
```

2) Display the details of software developed by male programmers earning MORE than 3000

```
SELECT * FROM Software WHERE dev_in IN (SELECT name FROM  
Programmer WHERE sex = 'male' AND salary > 3000);
```

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

```
SELECT * FROM Software WHERE dev_in IN (SELECT name FROM  
Programmer WHERE sex = 'female') AND dev_in IN (SELECT name FROM  
Programmer WHERE prof1 = 'PASCAL' OR prof2 = 'PASCAL');
```

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

```
SELECT * FROM Programmer WHERE YEAR(doj) < 1990;
```

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

```
SELECT s.* FROM Software s JOIN Programmer p ON s.dev_in = p.name  
WHERE p.sex = 'female' AND (p.prof1 = 'C' OR p.prof2 = 'C') AND p.splace =  
'PRAGATHI';
```

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

```
SELECT s.splace AS institute_name, p.name AS programmer_name,  
COUNT(s.title) AS num_packages, SUM(s.sold) AS total_copies_sold,  
SUM(s.sold * s.scost) AS total_sales_value FROM Programmer p LEFT JOIN
```

Studies st ON p.name = st.name LEFT JOIN Software s ON p.name = s.dev_in
GROUP BY s.splace, p.name;

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

```
SELECT s.* FROM Software s JOIN Programmer p ON s.dev_in = p.name JOIN  
(SELECT splace, COUNT(*) AS num_programmers FROM Studies GROUP BY splace  
ORDER BY num_programmers DESC LIMIT 1) AS most_programmers ON p.splace =  
most_programmers.splace WHERE s.dev_in = p.name AND s.dev_in IN (SELECT name  
FROM Programmer WHERE sex = 'male') AND s.dev_in IN (SELECT name FROM  
Programmer WHERE name = p.name AND p.prof1 = 'DBASE');
```

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

```
SELECT * FROM Software WHERE dev_in IN (SELECT name FROM  
Programmer WHERE sex = 'male' AND YEAR(dob) < 1965) AND dev_in IN (  
SELECT name FROM Programmer WHERE sex = 'female' AND YEAR(dob) >  
1975);
```

- 9) Display the details of the software that was developed in the language that is NOT the programmer's first proficiency.**

```
SELECT * FROM Software WHERE dev_in IN (SELECT name FROM  
Programmer WHERE prof1 <> dev_in AND (prof2 IS NULL OR prof2 <>  
dev_in));
```

- 10) Display details of software that was developed in the language which is NEITHER first NOR second proficiency of the programmer.**

```
SELECT s.* FROM Software s JOIN Programmer p ON s.dev_in = p.name  
WHERE 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.dev_in = p.name JOIN  
Studies st ON p.name = st.name WHERE p.sex = 'male' AND st.splace =  
'SABHARI';
```

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

```
SELECT name FROM Programmer WHERE name NOT IN (SELECT dev_in  
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 dev_in IN (
SELECT name FROM Programmer WHERE splace = 'APPLE');
```

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

```
SELECT a.name,a.doj FROM Programmer a,programmer b WHERE a.doj=b.doj
and a.name <> b.name;
```

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

```
SELECT unique(a.name),a.prof2 FROM Programmer a,programmer b where
a.prof2=b.prof2 and a.name <> b.name;
```

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

```
SELECT st.splace AS institute_name, SUM(s.sold * s.scost) AS total_sales_value
FROM Software s JOIN Programmer p ON s.dev_in = 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 AS institute_name FROM Software s JOIN Programmer p ON
s.dev_in = 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 language FROM (SELECT prof1 AS language FROM
Programmer UNION SELECT prof2 AS language FROM Programmer) AS
languages WHERE language 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 AS programmer_name, p.salary AS programmer_salary,
st.course AS course_name FROM Programmer p JOIN Software s ON p.name =
s.dev_in 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 AS programmer_name, st.course AS course_name, st.ccost AS
course_cost, p.salary AS monthly_salary, ROUND(st.ccost / p.salary, 2) 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 year's experiences?

```
SELECT s.title AS package_name, MAX(s.scost) AS package_cost FROM
Software s JOIN Programmer p ON s.dev_in = p.name WHERE
TIMESTAMPDIFF(YEAR, p.doj, CURDATE()) < 3 ORDER BY s.scost DESC
LIMIT 1;
```

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

```
SELECT AVG(p.salary) AS average_salary FROM Programmer p JOIN Software
s ON p.name = s.dev_in WHERE s.sold * s.scost > 50000;
```

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

```
SELECT COUNT(s.title) AS num_packages FROM Software s JOIN
Programmer p ON s.dev_in = p.name JOIN Studies st ON p.name = st.name
WHERE st.splace = (SELECT splace FROM Studies
GROUP BY splace ORDER BY ccost ASC LIMIT 1);
```

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 AS institute_name FROM
Software s JOIN Programmer p ON s.dev_in = p.name JOIN Studies st ON
p.name = st.name WHERE s.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(s.title) AS num_packages FROM Software s JOIN
Programmer p ON s.dev_in = p.name WHERE p.sex = 'female' AND p.salary >
(SELECT MAX(salary) FROM Programmer WHERE sex = 'male');
```

26) How many packages were developed by the MOST experienced programmers from BDPS?

```
SELECT COUNT(*) AS num_packages FROM Software s JOIN Programmer p ON
s.dev_in = p.name WHERE TIMESTAMPDIFF(YEAR, p.doj, CURDATE()) = (SELECT
MAX(TIMESTAMPDIFF(YEAR, p1.doj, CURDATE())) FROM Programmer p1
WHERE p1.splace = 'BDPS');
```

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

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

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

```
SELECT p.profl AS proficiency, COUNT(DISTINCT p.name) AS
num_programmers, COUNT(s.title) AS num_packages FROM Programmer p
LEFT JOIN Software s ON p.profl = s.dev_in GROUP BY p.profl;
```

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

```
SELECT p.name AS programmer_name, COUNT(s.title) AS
num_packages_developed FROM Programmer p LEFT JOIN Software s ON
p.name = s.dev_in GROUP BY p.name ORDER BY
programmer_name;
```

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

```
SELECT p.*, s.splace AS institute_name, s.course AS course_name, s.ccost AS
course_cost FROM Programmer p JOIN Studies s ON p.name = s.name WHERE
s.splace = 'S.S.I.L';
```