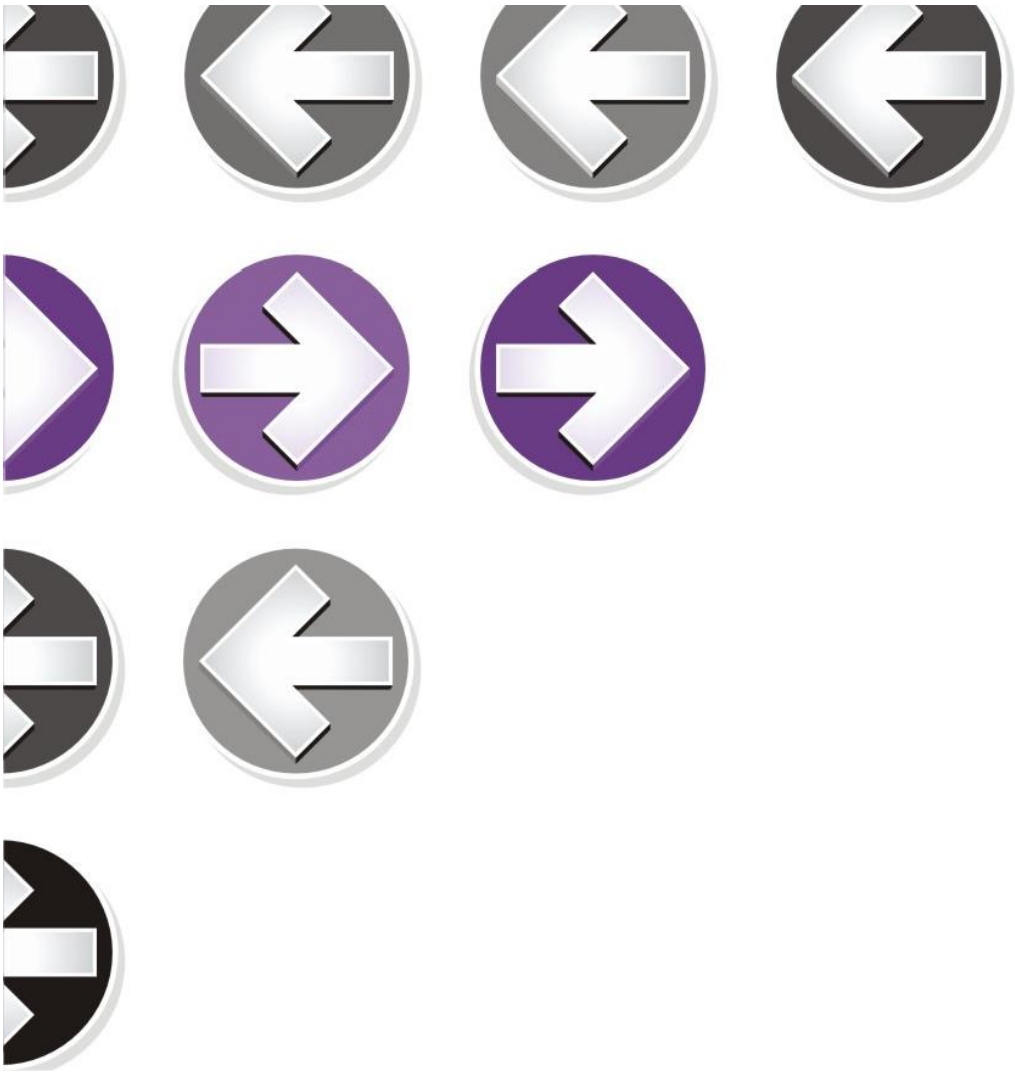


aspire SYSTEMS
Transforming Product Development

Assignments on SQL

This document contains few assignments on topics that covered under the SQL Course.

Talent Nurturing – Aspire Systems India Private Limited, Chennai

Summary

In this assignment, you are going to write SQL queries for each question that are given below. If you are having your own system you can install any database software like MySQL, MSSQL, Oracle, etc and create tables that are given below with the same structure. Sample data's that are provided below for those who are not having their own PC in home, to understand what type of data's will be there in the tables and you can insert the values as what you like. The purpose of this assignment is to assess your understanding of RDBMS concepts and how you are going to apply SQL Queries for each scenario that is given below.

Table Structure

1) Table Name	Programmer		
name	not null	varchar2(8)	name
dob	not null	date	date of birth
doj	not null	date	date of joining
sex	not null	varchar2(1)	male/ female
prof1		varchar2(8)	known language 1
prof2		varchar2(8)	known language 2
salary	not null	number(4)	salary

Sample Data						
somdutt	21-Apr-66	21-Apr-92	m	pascal	basic	3200

2) Table Name	Software		
name	not null	varchar2(8)	name
title	not null	varchar2(20)	developed project name
dev_in	not null	varchar2(8)	language developed
scost		number(7,2)	software cost
dcost		number(5)	development cost
sold		number(3)	number of software sold

Sample Data					
somdutt	parachutes	basic	399.95	6000	43

3) Table Name	Studies		
name	not null	varchar2(8)	name
splace	not null	varchar2(9)	studies place
course	not null	varchar2(5)	course studies
ccost	not null	varchar2(5)	course cost

Sample Data			
somdutt	sabhari	pgdca	4500
devdutt	bdps	dcs	5000

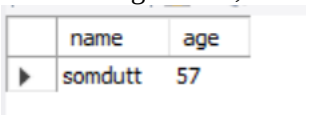
QUERIES - I

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, YEAR(CURDATE()) - YEAR(dob) - (RIGHT(CURDATE(), 5) < RIGHT(dob, 5)) AS age
FROM Programmer;
```



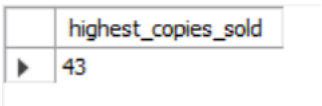
	name	age
▶	somdutt	57

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

```
SELECT p.name, YEAR(CURDATE()) - YEAR(dob) - (RIGHT(CURDATE(), 5) < RIGHT(dob, 5)) AS age
FROM Programmer p
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 MAX(sold) AS highest_copies_sold
FROM Software;
```



	highest_copies_sold
▶	43

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(*) AS pgdca_count
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_count
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_sell
FROM Software;
```

	name	copies_to_sell
▶	somdutt	16

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';
```

	costliest_software_price
▶	399.95

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

16) How many programmers studies at paragathi?

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

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

```
SELECT COUNT(*) AS programmers_count
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(*) AS programmers_count
FROM Programmer
WHERE prof1 IN ('Cobol', 'Pascal') OR prof2 IN ('Cobol', 'Pascal');
```

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

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

22) How old is the oldest male programmers?

```
   SELECT MAX(YEAR(CURDATE()) - YEAR(dob) - (RIGHT(CURDATE(), 5) < RIGHT(dob, 5))) AS
oldest_male_age
FROM Programmer
WHERE sex = 'male';
```

23) What is the average age of female programmers?

```
SELECT AVG(YEAR(CURDATE()) - YEAR(dob) - (RIGHT(CURDATE(), 5) < RIGHT(dob, 5))) AS
average_female_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,
      YEAR(CURDATE()) - YEAR(doj) - (RIGHT(CURDATE(), 5) < RIGHT(doj, 5)) 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_count
FROM Programmer
WHERE sex = 'female';
```

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

```
SELECT DISTINCT prof1 AS language
FROM Programmer
WHERE sex = 'male'
UNION
SELECT DISTINCT prof2 AS language
FROM Programmer
WHERE sex = 'male';
```

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_count
FROM Programmer
WHERE salary BETWEEN 2000 AND 4000;
```

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 programers knowing C are above 24 years of age?

```
SELECT COUNT(*) AS female_programmers_count
FROM Programmer
WHERE sex = 'female'
  AND (prof1 = 'C' OR prof2 = 'C')
  AND YEAR(CURDATE()) - YEAR(dob) > 24;
```

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

```
SELECT name, dob
FROM Programmer
WHERE WEEK(dob) = WEEK(CURDATE()) AND
    DAYOFYEAR(dob) >= DAYOFYEAR(CURDATE()) AND
    DAYOFYEAR(dob) <= DAYOFYEAR(DATE_ADD(CURDATE(), INTERVAL 7 DAY));
```

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

```
SELECT *
FROM Programmer
WHERE doj > DATE_SUB(CURDATE(), INTERVAL 1 YEAR);
```

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 name, (scost * sold) - dcost 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 scost AS software_cost
FROM Software
WHERE name = 'Mary';
```

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

```
SELECT DISTINCT splace AS institute_name
FROM Studies;
```

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

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

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

```
SELECT name
FROM Programmer
WHERE LENGTH(name) - LENGTH(REPLACE(name, 'a', '')) = 2;
```

41) Display the names of programmers whose names contain unto 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_count
FROM Programmer
WHERE sex = 'female'
  AND (prof1 = 'COBOL' OR prof2 = '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 formatted_dob,
    DATE_FORMAT(doj, '%d/%m/%y') AS formatted_doj
FROM Programmer;
```

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

```
SELECT name, dob
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_paid
FROM Programmer
WHERE sex = 'male' AND NOT (prof1 = 'Cobol' OR 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 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 - 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(scost) AS costliest_package,
MAX(sold) AS highest_selling_package
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 YEAR(dob)
ORDER 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 YEAR(doj)
ORDER 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 MONTH(dob)
ORDER 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 MONTH(doj)
```

ORDER BY join_month;

9) Display the language wise COUNTS of prof1.

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

10) Display the language wise COUNTS of prof2.

```
SELECT prof2 AS language, COUNT(*) AS count
FROM Programmer
GROUP BY prof2
ORDER BY count DESC;
```

11) Display THE NUMBER OF people in EACH 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) 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_selling_cost
FROM Software
GROUP BY name;
```

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

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

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

```
SELECT name, COUNT(*) AS number_of_packages
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 total_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(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 AS language_name,
  AVG(dcost) AS average_development_cost,
  AVG(scost) AS average_selling_cost,
  AVG(scost / sold) AS average_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_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(*) AS number_of_students
        FROM Studies
        GROUP BY splace;
```

24) Display names of male and female programmers.

```
SELECT name, sex
FROM Programmer
WHERE sex = 'male' OR sex = 'female';
```

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

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

26) Display the NUMBER of packages in EACH language.

```
SELECT dev_in AS language, COUNT(*) AS number_of_packages
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 number_of_packages
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_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_scost,
SUM(dcost) AS total_dcost,
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, 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 prof1='cobol'and Sex='Female' order by Salary desc limit 1;
```

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

```
SELECT P.Name, P.Prof1, P.Salary FROM Programmer P JOIN (SELECT Prof1, MAX(Salary) AS MaxSalary FROM Programmer GROUP BY Prof1 ) AS MaxSalaries ON P.Prof1 = MaxSalaries.Prof1 AND P.Salary = MaxSalaries.MaxSalary;
```

4) Who is the LEAST experienced programmer?

```
select Name, TIMESTAMPDIFF(YEAR, DOB, CURDATE()) as Experience from Programmer order by Experience limit 1;
```

5) Who is the MOST experienced programmer?

```
select Name, TIMESTAMPDIFF(YEAR, DOB, CURDATE()) as Experience from Programmer order by Experience desc limit 1;
```

6) Which language is known by ONLY ONE programmer?

```
select Name, Prof1 as Language from Programmer group by Name, Prof1 having count(*) = 1 union select Name, Prof2 as Language from Programmer group by Name, Prof2 having count(*) = 1;
```

7) Who is the YONGEST programmer knowing DBASE?

```
select Name, TIMESTAMPDIFF(YEAR, DOB, CURDATE()) as Experience from Programmer where Prof1='Dbase' or Prof2='Dbase'order by Experience limit 1;
```

8) Which institute has MOST NUMBER of students?

```
select Stu_place as Institute, count(*) as StudentCount from Studies group by Stu_place order by StudentCount desc limit 1;
```

9) Who is the above programmer?

```
SELECT *
FROM Programmer
WHERE name = (
  SELECT name
  FROM Studies
  WHERE splace = 'SSIL'
);
```

10) Which female programmer earns MORE than 3000/- but DOES NOT know C, C++, Oracle or Dbase?

```
SELECT *
FROM Programmer
```

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

11) Which is the COSTLIEST course?

```
SELECT course, MAX(ccost) AS cost
FROM Studies
GROUP BY course
ORDER BY cost DESC
LIMIT 1;
```

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

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

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

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

14) Which institute conducts COSTLIEST course?

```
SELECT splace AS institute, course
FROM Studies
WHERE ccost = (SELECT MAX(ccost) FROM Studies);
```

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_students);
```

16) Which institute conducts the above course?

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

```
SELECT course
```

```
FROM Studies
WHERE ccost BETWEEN (
   SELECT AVG(ccost) - 1000
   FROM Studies
) AND (
   SELECT AVG(ccost) + 1000
   FROM Studies
);
```

18) Which package has the HIGEST development cost?

```
SELECT *
FROM Software
ORDER BY dcost DESC
LIMIT 1;
```

19) Which package has the LOWEST selling cost?

```
SELECT *
FROM Software
ORDER BY scost ASC
LIMIT 1;
```

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

```
SELECT name
FROM Software
WHERE sold = (
   SELECT MIN(sold)
   FROM Software
);
```

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

```
SELECT dev_in
FROM Software
WHERE scost = (
   SELECT MAX(scost)
   FROM Software
);
```

22) How many copies of the package that has the LEAST DIFFRENCE 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 *
```

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

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

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

26) Who is the author of the COSTLIEST package?

```
SELECT name AS programmer_name, title AS package_title, scost AS software_cost
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, sold AS number_of_copies
FROM Software
WHERE sold < (SELECT AVG(sold) FROM Software);
```

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

```
SELECT *
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, COUNT(*) AS prof1_count
FROM Programmer
GROUP BY prof1
ORDER BY prof1_count DESC
LIMIT 1;
```

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

```
SELECT p.name
FROM Programmer p
JOIN Software s ON p.name = s.name
WHERE s.sold * s.scost > 2 * s.dcost;
```

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

```
SELECT p.name, s.language, MIN(s.scost) AS cheapest_package_cost
FROM Programmer p
JOIN Software s ON p.name = s.name
GROUP BY p.name, s.language;
```

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

```
SELECT name, dob
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.

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

```
SELECT name, dob
FROM Programmer
WHERE sex = 'Female' AND doj = '1992-01-01'
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 birth_year
  ORDER BY num_programmers_born 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 join_month
ORDER BY num_programmers_joined DESC
LIMIT 1;
```

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

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

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 = 'Male' 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 p.sex = 'Female' AND s.dev_in = '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.name = p.name
    JOIN Studies st ON p.name = st.name
    WHERE p.sex = 'Female'
    AND s.dev_in = 'C'
    AND st.splace = 'PRAGATHI';
```

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

```
        SELECT p.name, st.splace AS institute,
    COUNT(s.name) AS num_packages,
    SUM(s.sold) AS num_copies_sold,
    SUM(s.sold * s.scost) AS sales_value
        FROM Programmer p
        JOIN Software s ON p.name = s.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 programmer's studies.

```
        SELECT s.*
    FROM Software s
    JOIN Programmer p ON s.name = p.name
    JOIN Studies st ON p.name = st.name
    WHERE p.sex = 'Male'
    AND s.dev_in = 'DBASE'
    AND st.splace = (
       SELECT splace
       FROM Studies
       GROUP BY splace
       ORDER BY COUNT(*) DESC
       LIMIT 1
);
```

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 = 'Male' AND p.dob < '1965-01-01')
          OR (p.sex = 'Female' AND p.dob > '1975-12-31');
```

9) Display the details of the software that was developed in the language that is NOT the programmer's 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 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 NOT IN (p.prof1, p.prof2)
AND (p.prof1 IS NULL OR p.prof2 IS NULL);
```

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 = 'Male'
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) What is the total cost of the software developed by the programmers byAPPLE?

```
SELECT SUM(s.scost) 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) 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 prof2, COUNT(*) AS count
FROM Programmer
GROUP BY prof2
HAVING COUNT(*) > 1;
```

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

```
SELECT st.splace AS institute, SUM(s.sold * s.scost) 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 AS institute
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 language
FROM (
  SELECT prof1 AS language FROM Programmer
  UNION
  SELECT prof2 AS language FROM Programmer
) AS languages
WHERE language NOT IN (
  SELECT DISTINCT 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 year's expenences?

```
SELECT s.name, s.title, s.scost
FROM Software s
JOIN Programmer p ON s.name = p.name
WHERE DATEDIFF(CURRENT_DATE(), p.doj) < 3 * 365
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.name
WHERE s.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 s
JOIN Studies st ON s.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
FROM Software s
JOIN Studies st ON s.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(*) AS num_packages
FROM Software s
JOIN Programmer p ON s.name = 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.name = p.name
JOIN Studies st ON p.name = st.name
WHERE st.splace = 'BDPS'
AND p.doj <= (
    SELECT MAX(doj)
    FROM Programmer
    WHERE name IN (
        SELECT name
        FROM Studies
        WHERE 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, COALESCE(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 prof1,
    COUNT(DISTINCT p.name) AS num_programmers,
    COUNT(*) AS num_packages
FROM Programmer p
JOIN Software s ON p.name = s.name
GROUP BY prof1;
```

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

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

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

```
        SELECT *
FROM Programmer
WHERE name IN (
  SELECT name
  FROM Studies
  WHERE splace = 'SSIL'
);
```