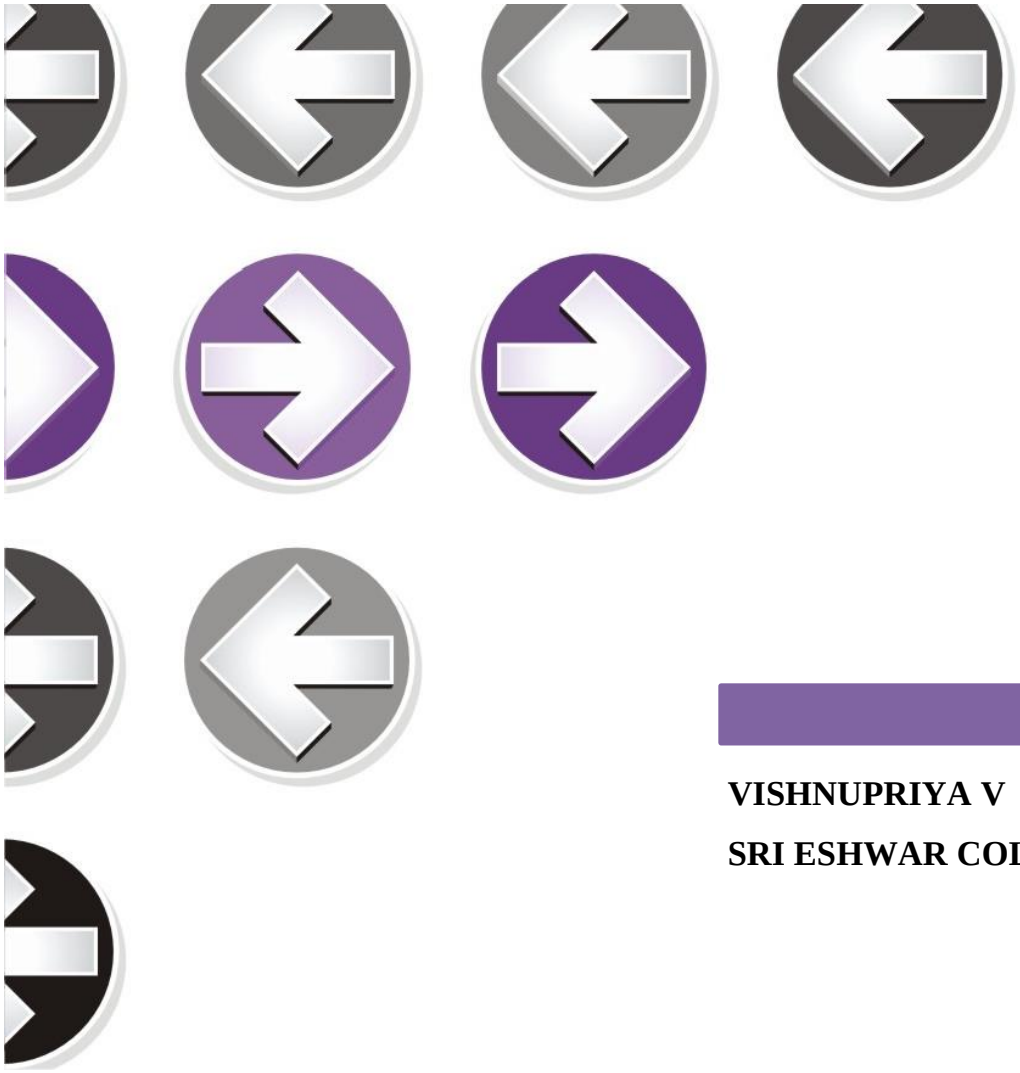

VISHNUPRIYA V

SRI ESHWAR COLLEGE OF ENGINEERING

Assignments on SQL

This document contains few assignments on topics thatcovered 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

```
CREATE TABLE programer(
   name VARCHAR(8) NOT NULL,
   dob DATE NOT NULL,
   doj DATE NOT NULL,
   sex VARCHAR(1) NOT NULL,
   prof1 VARCHAR(8),
   prof2 VARCHAR(8),
   salary INT
);
```

```
INSERT INTO programmer VALUES ('somdutt', '1966-01-21', '2021-04-1992', 'M', 'pascal', 'basic', 3200),
('Nivas', '1990-04-15', '2021-08-20', 'M', 'Java', 'Python', 10000),
('Aparna', '1988-09-23', '2020-07-10', 'F', 'C++', 'JavaScript', 12000),
('Kumar', '1995-12-30', '2022-01-05', 'M', 'Ruby', 'PHP', 13000),
('Priya', '1992-06-18', '2023-03-15', 'F', 'Swift', 'Kotlin', 14000),
('Kavin', '1987-03-11', '2024-02-02', 'M', 'C#', 'SQL', 15000);
```

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

```
CREATE TABLE Software (
   name VARCHAR(8) NOT NULL,
   title VARCHAR(20) NOT NULL,
   dev_in VARCHAR(8) NOT NULL,
   scost DECIMAL(7,2),
   dcost DECIMAL(5),
   sold INT
);


INSERT INTO Software (name, title, dev_in, scost, dcost, sold)
VALUES
('Nikitha', 'Inventory', 'Java', 499.99, 8000, 55),
('Arun', 'Calculator', 'C++', 299.50, 5000, 30),
('Mithul', 'E-commerce', 'Python', 899.75, 10000, 70),
('Ilyash', 'Messaging', 'Swift', 399.25, 7000, 45),
('Danya', 'Analytics', 'SQL', 599.80, 9000, 60)
```

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

```
CREATE TABLE Studies (
   name VARCHAR(8) NOT NULL,
   splace VARCHAR(9) NOT NULL,
   course VARCHAR(5) NOT NULL,
   ccost VARCHAR(5) NOT NULL
);

INSERT INTO Studies (name, splace, course, ccost)
VALUES
('somdutt', 'sabhari', 'pgdca', '4500'),
('devdutt', 'bdps', 'dcs', '5000'),
('vishnu', 'pollachi', 'bca', '6000'),
('sneha', 'coimbatore', 'mca', '7000'),
('maha', 'chennai', 'btech', '8000');
```

QUERIES – I

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

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

2) Display the names and ages of all programmers.

```
SELECT name,
   EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM dob) AS age
FROM Programmer;
```

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

```
SELECT p.name, EXTRACT(YEAR FROM sysdate) - EXTRACT(YEAR FROM p.dob) 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 Max_Copies_Sold
FROM Software;
```

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

```
SELECT name, dob
FROM Programmer
WHERE TO_CHAR(dob, 'MM') = '01';
```

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
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)
FROM Software;
```

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

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

16) How many programmers studies at paragathi?

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

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

```
SELECT COUNT(*)
FROM Studies
WHERE ccost BETWEEN '5000' AND '10000';
```

18) What is the average course fee?

```
SELECT AVG(ccost)
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(*)
FROM Programmer
WHERE prof1 IN ('cobol', 'pascal') OR prof2 IN ('cobol', 'pascal');
```

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

```
SELECT COUNT(*)
FROM Programmer
WHERE NOT (prof1 IN ('pascal', 'C') OR prof2 IN ('pascal', 'C'));
```

22) How old is the oldest male programmer?

```
SELECT MAX(EXTRACT(YEAR FROM sysdate) - EXTRACT(YEAR FROM dob))
FROM Programmer
WHERE sex = 'm';
```

23) What is the average age of female programmers?

```
SELECT AVG(EXTRACT(YEAR FROM sysdate) - EXTRACT(YEAR FROM dob))
```

```
FROM Programmer
WHERE sex = 'f';
```

24) Calculate the experience in years for each programmer and display along with the names in descending order.

```
SELECT name, EXTRACT(YEAR FROM sysdate) - EXTRACT(YEAR FROM doj)
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 TO_CHAR(dob, 'MM') = TO_CHAR(sysdate, 'MM');
```

26) How many female programmers are there?

```
SELECT COUNT(*)
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(*)
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(*)
FROM Programmer
WHERE sex = 'f' 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
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 DATEDIFF(CURDATE(), doj) < 365;
```

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

```
SELECT *
FROM Programmer
WHERE DATEDIFF(CURDATE(), doj) >= 730 AND DATEDIFF(CURDATE(), doj) < 1095;
```

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

```
SELECT name, (scost * sold) - dcost
WHERE (scost * sold) < dcost;
```

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

```
SELECT title
FROM Software
WHERE sold = 0;
```

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

```
SELECT scost AS Cost
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)
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(*)
FROM Programmer
```

```
WHERE sex = 'f' AND (prof1 = 'COBOL' OR prof2 = 'COBOL') AND DATEDIFF(CURDATE(), doj) > 730;
```

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)
FROM Programmer
WHERE sex = 'm' 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 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 '% %';
```

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 programmers.

```
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(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 YEAR(dob);
```

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);
```

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);
```

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);
```

9) Display the language wise COUNTS of prof1.

```
SELECT prof1 AS Language, COUNT(*) AS Count
FROM Programmer
GROUP BY prof1;
```

10) Display the language wise COUNTS of prof2.

```
SELECT prof2 AS Language, COUNT(*) AS 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) 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 in EACH programmer.
SELECT name, SUM(scost) AS Total_Selling_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) AS Total_Selling_Cost
FROM Software
GROUP BY name, dev_in;

20) Display EACH programmer's name, costliest package and cheapest packages developed by Him/Her.
SELECT s.name AS Programmer_Name, MAX(s.scost) AS Costliest_Package, MIN(s.scost) AS
Cheapest_Package
FROM Software s
JOIN Programmer p ON s.name = p.name
GROUP BY s.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 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, COUNT(DISTINCT 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, COUNT(*) 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.title 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(*) 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 Average_Difference
FROM Software
GROUP BY dev_in;
```

29) Display the TOTAL scost, dcost and amount TO BE 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(scost - 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 = '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 DISTINCT ON (prof1) name
FROM Programmer
ORDER BY prof1, salary DESC;
```

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

7) Who is the YOUNGEST 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 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 Programmer
WHERE prof1 = (
   SELECT prof1
   FROM Programmer
   WHERE prof1 = 'DBASE' OR prof2 = 'DBASE'
   ORDER BY dob 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
FROM Studies
ORDER BY ccost DESC
LIMIT 1;
```

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

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

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

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

14) Which institute conducts COSTLIEST course?

```
SELECT splace AS Institute
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(COUNT(*)) FROM Studies GROUP BY course);
```

16) Which institute conducts the above course?

```
SELECT splace AS Institute
FROM Studies
GROUP BY splace
HAVING COUNT(*) < (SELECT AVG(COUNT(*)) FROM Studies GROUP BY 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 title
FROM Software
ORDER BY dcost DESC
LIMIT 1;
```

19) Which package has the LOWEST selling cost?

```
SELECT title
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)
LIMIT 1;
```

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

```
SELECT dev_in AS Language
FROM Software
WHERE scost = (SELECT MAX(scost) FROM Software)
LIMIT 1;
```

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

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

23) Which is the COSTLIEAST package developed in PASCAL?

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

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

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

26) Who is the author of the COSTLIEST package?

```
SELECT name
FROM Software
WHERE scost = (SELECT MAX(scost) FROM Software)
LIMIT 1;
```

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

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

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

```
SELECT prof1 AS Language
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
FROM Software
GROUP BY name
HAVING SUM(scost) > (2 * SUM(dcost));
```

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

```
SELECT s.name AS Programmer_Name, s.title AS Cheapest_Package, s.dev_in AS Language
FROM Software s
JOIN (
   SELECT name, dev_in, MIN(scost) AS Min_Selling_Cost
   FROM Software
   GROUP BY name, dev_in
) AS min_cost ON s.name = min_cost.name AND s.dev_in = min_cost.dev_in AND s.scost =
min_cost.Min_Selling_Cost;
```

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

```
SELECT name
FROM Programmer
WHERE sex = 'm' AND YEAR(dob) = 1965
ORDER BY dob ASC
LIMIT 1;
```

33) 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 AND scost = MAX(scost)) AS
Highest_Selling_Language,
     (SELECT dev_in FROM Software WHERE name = p.name AND scost = MIN(scost)) AS
Lowest_Selling_Language
FROM Programmer p;
```

34) 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;
```

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

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

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

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

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

```
SELECT prof1 AS Language, COUNT(*) AS Programmer_Count
```

```
FROM Programmer
GROUP BY prof1
UNION ALL
SELECT prof2 AS Language, COUNT(*) AS Programmer_Count
FROM Programmer
GROUP BY prof2
ORDER BY Programmer_Count DESC
LIMIT 1;
```

38) 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');
```

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

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 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 *
FROM Software
WHERE dev_in = 'C' AND name IN (
   SELECT name
   FROM Programmer
   WHERE sex = 'f' AND splace = 'PRAGATHI'
);
```

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

```
SELECT p.name AS Programmer_Name, p.splace AS Institute, COUNT(s.title) AS Packages_Developed,
SUM(s.sold) AS Total_Copies_Sold, SUM(s.sold * s.scost) AS Total_Sales_Value
FROM Programmer p
LEFT JOIN Software s ON p.name = s.name
GROUP BY p.name, p.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 *
FROM Software
WHERE dev_in = 'DBASE' AND name IN (
   SELECT name
   FROM Programmer
   WHERE sex = 'm' AND splace = (
      SELECT splace
      FROM Programmer
      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 *
FROM Software
WHERE name IN (
   SELECT name
   FROM Programmer
   WHERE (sex = 'm' AND YEAR(dob) < 1965)
     OR (sex = 'f' 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 NOT IN (
   SELECT prof1
   FROM Programmer
   UNION
   SELECT prof2
   FROM Programmer
);
```

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

```
SELECT *
FROM Software
WHERE dev_in NOT IN (
   SELECT prof1
   FROM Programmer
   UNION
   SELECT prof2
```

```
   FROM Programmer
);
```

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

```
SELECT *
FROM Software
WHERE name IN (
   SELECT name
   FROM Programmer
   WHERE sex = 'm' AND splace = 'SABHARI'
);
```

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 splace = 'APPLE'
);
```

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

```
SELECT name
FROM Programmer
GROUP BY name
HAVING COUNT(DISTINCT doj) = 1;
```

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

```
SELECT name
FROM Programmer
GROUP BY name
HAVING COUNT(DISTINCT prof2) = 1;
```

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

```
SELECT p.splace AS Institute, SUM(s.sold * s.scost) AS Total_Sales_Value
FROM Programmer p
LEFT JOIN Software s ON p.name = s.name
GROUP BY p.splace;
```

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

```
SELECT splace AS Institute
FROM Programmer
WHERE name IN (
   SELECT name
   FROM Software
   WHERE 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, p.salary, s.title, s.sold, s.scost, s.dcost, s.dev_in
FROM Programmer p
JOIN Software s ON p.name = s.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, s.title, ROUND(s.scost / (p.salary / 12), 2) AS Months_To_Recover
FROM Programmer p
JOIN Software s ON p.name = s.name;
```

21) Which is the COSTLIEST package developed by a person with under 3 year's experience?

```
SELECT *
FROM Software
WHERE name IN (
   SELECT name
   FROM Programmer
   WHERE DATE_PART('year', CURRENT_DATE) - DATE_PART('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(p.salary) AS Average_Salary
FROM Programmer p
JOIN Software s ON p.name = s.name
GROUP BY p.salary
HAVING SUM(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(*) AS Package_Count
FROM Software
WHERE name IN (
   SELECT name
   FROM Programmer
   WHERE splace = (
      SELECT splace
      FROM Studies
      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 Package_Count, p.splace AS Institute
FROM Software s
JOIN Programmer p ON s.name = p.name
WHERE scost = (
   SELECT MIN(scost)
   FROM Software
)
GROUP BY p.splace;
```

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

```
SELECT COUNT(*) AS Package_Count
FROM Software
WHERE name IN (
   SELECT name
   FROM Programmer
   WHERE sex = 'f' AND 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 Package_Count
FROM Software
WHERE name IN (
   SELECT name
   FROM Programmer
   WHERE splace = 'BDPS'
   ORDER BY doj DESC
   LIMIT 1
);
```

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

```
SELECT p.name, p.splace AS Institute
FROM Programmer p
LEFT JOIN Software s ON p.name = s.name;
```

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

```
SELECT prof1 AS Language, COUNT(DISTINCT name) AS Programmer_Count, COUNT(title) AS
Package_Count
FROM Programmer p
LEFT 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, COUNT(s.title) AS Package_Count
FROM Programmer p
LEFT JOIN Software s ON p.name = s.name
GROUP BY p.name;
```

30) List all the details of programmers.

```
SELECT *
FROM Programmer;
```