create table Programmer (

name varchar(100) not null,

dob date not null,

doj date not null,

sex varchar(1),

prof1 varchar(20),

prof2 varchar(20),

salary integer not null

);

INSERT INTO Programmer (name, dob, doj, sex, prof1, prof2, salary)

VALUES

('Muhil', '1990-05-15', '2015-08-20', 'M', 'Java', 'Python', 75000),

('Prathi', '1988-12-10', '2014-07-12', 'F', 'C++', 'JavaScript', 80000),

('Harshini', '1995-02-28', '2018-04-30', F, 'Python', 'SQL', 70000),

('Priyanka', '1992-09-20', '2017-03-15', 'F', 'Java', 'C#', 85000),

('Nandhini', '1985-06-25', '2012-01-10', 'F', 'JavaScript', 'HTML/CSS', 90000),

('Liya', '1993-04-18', '2016-11-05', 'F', 'Python', 'Ruby', 78000),

('Susmitha', '1991-11-07', '2019-09-22', 'F', 'C#', 'Java', 82000),

('Tamil', '1989-08-12', '2013-06-08', 'M', 'SQL', 'Python', 76000),

('Hari', '1987-03-30', '2010-12-05', 'M', 'Java', 'JavaScript', 88000),

('Jessi', '1994-07-03', '2015-05-18', 'F', 'C++', 'Python', 83000);

select \* from Programmer;

create table software (

name varchar(20) not null,

title varchar(20) not null,

dev\_in varchar (20) not null,

scost integer,

dcost integer,

sold integer

);

INSERT INTO software (dev\_in, title, name, scost, dcost, sold)

VALUES

('Pascal', 'Word Processor', 'Microsoft Word', 50.00, 10.00, 1000),

('Adobe', 'Photo Editing', 'Adobe Photoshop', 200.00, 25.00, 500),

('Autodesk', '3D Modeling', 'Autodesk Maya', 3000.00, 500.00, 200),

('Epic Games', 'Game Development', 'Unreal Engine', 0.00, 0.00, 5000),

('Oracle', 'Database Management', 'Oracle Database', 5000.00, 1000.00, 100),

('Pascal', 'Accounting Software', 'QuickBooks', 200.00, 50.00, 1000),

('Salesforce', 'CRM', 'Salesforce CRM', 100.00, 20.00, 2000),

('Pascal', 'Virtualization', 'VMware vSphere', 4000.00, 800.00, 300),

('Symantec', 'Security Software', 'Norton Security', 50.00, 15.00, 1500),

('IBM', 'Business Intelligence', 'IBM Cognos', 3000.00, 600.00, 400);

CREATE TABLE Studies (

name VARCHAR(20) NOT NULL,

splace VARCHAR(19) NOT NULL,

course VARCHAR(15) NOT NULL,

ccost VARCHAR(15) NOT NULL

);

INSERT INTO Studies (name, splace, course, ccost) VALUES ('John', 'ABC College', 'BSc', '6000'),

('Muhil', 'XYZ Institute', 'MBA', '8000'),

('Parthi', 'LMN University', 'Engineering', '7000'),

('Abi', 'PQR Academy', 'Accounting', '5500'),

('Selvi', 'DEF School', 'IT', '6500'),

('Susma', 'GHI College', 'Marketing', '7500'),

('Hari', 'JKL Institute', 'Medicine', '9000'), ('Emma', 'NOP University', 'Law', '8500'), ('William', 'QRS Academy', 'Finance', '7200'), ('Olivia', 'TUV School', 'Psychology', '5800');

**QUERY I**

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

SELECT AVG(sold) AS average\_selling\_cost

FROM software

WHERE prof1 = 'Pascal';

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

SELECT name, EXTRACT(YEAR FROM CURRENT\_DATE) - EXTRACT(YEAR FROM date\_of\_birth) 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 CURRENT\_DATE) - EXTRACT(YEAR FROM p.date\_of\_birth) 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;

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

SELECT name, date\_of\_birth

FROM programmer

WHERE EXTRACT(MONTH FROM date\_of\_birth) = 1;

**6) Display lowest course fee.**

SELECT MIN(CAST(ccost AS DECIMAL)) AS lowest\_course\_fee

FROM studies;

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

SELECT COUNT(\*) AS num\_programmers\_with\_pgdca

FROM studies

WHERE course = 'PGDCA';

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

SELECT SUM(sold) AS total\_revenue

FROM software

WHERE prof1 = 'C';

9) Display the details of software developed by Muhil?

SELECT \*

FROM software

WHERE dev\_in = 'Muhil';

**10) How many programmers studied at HARI.**

SELECT COUNT(\*) AS num\_programmers\_at\_sabhari

FROM programmer

WHERE splace = 'HARI';

**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, dcost / scost AS copies\_to\_be\_sold

FROM software;

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

SELECT MAX(scost) AS costliest\_software\_price

FROM software

WHERE prof1 = 'BASIC';

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

SELECT \*

FROM software

WHERE scost >= dcost;

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

SELECT COUNT(\*) AS num\_packages\_developed\_in\_dbase

FROM software

WHERE prof1 = 'dbase';

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

SELECT COUNT(\*) AS num\_programmers\_at\_paragathi

FROM programmer

WHERE splace = 'prathi';

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

SELECT COUNT(\*) AS num\_programmers\_paid\_between\_5000\_to\_10000

FROM studies

WHERE CAST(ccost AS DECIMAL) BETWEEN 5000 AND 10000;

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

SELECT AVG(CAST(ccost AS DECIMAL)) 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 num\_programmers\_knowing\_cobol\_or

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

SELECT COUNT(\*) AS num\_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 EXTRACT(YEAR FROM CURRENT\_DATE) - EXTRACT(YEAR FROM MAX(date\_of\_birth)) AS age\_of\_oldest\_male\_programmer

FROM programmer

WHERE sex = 'M';

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

SELECT AVG(EXTRACT(YEAR FROM CURRENT\_DATE) - EXTRACT(YEAR FROM date\_of\_birth)) AS average\_age\_of\_female\_programmers

FROM programmer

WHERE sex = 'F';

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

SELECT name, EXTRACT(YEAR FROM CURRENT\_DATE) - EXTRACT(YEAR FROM date\_of\_join) 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 EXTRACT(MONTH FROM date\_of\_birth) = EXTRACT(MONTH FROM CURRENT\_DATE);

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

SELECT COUNT(\*) AS num\_female\_programmers

FROM programmer

WHERE sex = 'F';

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

SELECT DISTINCT prof1 AS language

FROM programmer

WHERE sex = 'M'

UNION

SELECT DISTINCT prof2 AS language

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 num\_programmers\_in\_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 num\_female\_programmers\_knowing\_c\_above\_24\_years

FROM programmer

WHERE sex = 'F' AND prof1 = 'C' AND EXTRACT(YEAR FROM CURRENT\_DATE) - EXTRACT(YEAR FROM date\_of\_birth) > 24;

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

SELECT name

FROM programmer

WHERE DATE\_TRUNC('week', date\_of\_birth) = DATE\_TRUNC('week', CURRENT\_DATE);

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

SELECT \*

FROM programmer

WHERE date\_of\_join >= CURRENT\_DATE - INTERVAL '1 year';

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

SELECT \*

FROM programmer

WHERE date\_of\_join <= CURRENT\_DATE - INTERVAL '2 year' AND date\_of\_join >= CURRENT\_DATE - INTERVAL '2 year' - INTERVAL '1 day';

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

SELECT name, (scost - dcost) \* sold 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 AS cost\_of\_software

FROM software

WHERE dev\_in = '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 num\_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 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 num\_female\_programmers\_knowing\_cobol\_with\_more\_than\_2\_years\_experience

FROM programmer

WHERE sex = 'F' AND prof1 = 'COBOL' AND EXTRACT(YEAR FROM CURRENT\_DATE) - EXTRACT(YEAR FROM date\_of\_join) > 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(scost) AS average\_development\_cost\_in\_cobol

FROM software

WHERE prof1 = 'COBOL';

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

SELECT name, sex, RIGHT('0' || date\_of\_birth::text, 2) || '/' || RIGHT('0' || EXTRACT(MONTH FROM date\_of\_birth)::text, 2) || '/' || RIGHT(EXTRACT(YEAR FROM date\_of\_birth)::text, 2) AS dob, doj

FROM programmer;

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

SELECT name

FROM programmer

WHERE EXTRACT(DAY FROM date\_of\_birth) = DATE\_TRUNC('MONTH', date\_of\_birth + INTERVAL '1 MONTH') - DATE\_TRUNC('MONTH', date\_of\_birth);

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

SELECT SUM(salary) AS total\_salary\_paid\_to\_male\_programmers

FROM programmer

WHERE sex = 'M' AND (prof1 != 'Cobol' OR prof1 IS NULL) AND (prof2 != 'Cobol' OR prof2 IS NULL);

**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, date\_of\_birth, date\_of\_join

FROM programmer

WHERE EXTRACT(MONTH FROM date\_of\_birth) = EXTRACT(MONTH FROM date\_of\_join);

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

SELECT title

FROM software

WHERE title LIKE '% %';

**QUERY II**

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

SELECT prof1 AS language, COUNT(\*) AS num\_packages\_developed

FROM programmer

GROUP BY prof1;

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

SELECT dev\_in AS programmer\_name, COUNT(\*) AS num\_packages\_developed

FROM software

GROUP BY dev\_in;

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

SELECT sex, COUNT(\*) AS num\_programmers

FROM programmer

GROUP BY sex;

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

SELECT prof1 AS language, MAX(dcost) AS costliest\_package, MAX(sold) AS highest\_selling\_package

FROM software

GROUP BY prof1;

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

SELECT EXTRACT(YEAR FROM date\_of\_birth) AS birth\_year, COUNT(\*) AS num\_people\_born

FROM programmer

GROUP BY EXTRACT(YEAR FROM date\_of\_birth)

ORDER BY EXTRACT(YEAR FROM date\_of\_birth);

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

SELECT EXTRACT(YEAR FROM date\_of\_join) AS join\_year, COUNT(\*) AS num\_people\_joined

FROM programmer

GROUP BY EXTRACT(YEAR FROM date\_of\_join)

ORDER BY EXTRACT(YEAR FROM date\_of\_join);

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

SELECT EXTRACT(MONTH FROM date\_of\_birth) AS birth\_month, COUNT(\*) AS num\_people\_born

FROM programmer

GROUP BY EXTRACT(MONTH FROM date\_of\_birth)

ORDER BY EXTRACT(MONTH FROM date\_of\_birth);

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

SELECT EXTRACT(MONTH FROM date\_of\_join) AS join\_month, COUNT(\*) AS num\_people\_joined

FROM programmer

GROUP BY EXTRACT(MONTH FROM date\_of\_join)

ORDER BY EXTRACT(MONTH FROM date\_of\_join);

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

SELECT prof1 AS language, COUNT(\*) AS num\_prof1

FROM programmer

GROUP BY prof1;

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

SELECT prof2 AS language, COUNT(\*) AS num\_prof2

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 num\_people

FROM programmer

GROUP BY FLOOR(salary / 1000) \* 1000;

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

SELECT splace AS institute, COUNT(\*) AS num\_people

FROM programmer

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 prof1 AS language, SUM(dcost) AS total\_development\_cost

FROM software

GROUP BY prof1;

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

SELECT prof1 AS language, SUM(sold) AS total\_selling\_cost

FROM software

GROUP BY prof1;

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

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

FROM software

GROUP BY dev\_in;

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

SELECT dev\_in AS programmer\_name, COUNT(\*) AS num\_packages\_developed

FROM software

GROUP BY dev\_in;

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

SELECT dev\_in AS programmer\_name, COUNT(\*) AS num\_packages\_developed

FROM software

GROUP BY dev\_in;

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

SELECT dev\_in AS programmer\_name, prof1 AS language, SUM(sold) AS total\_sales\_cost

FROM software

GROUP BY dev\_in, prof1;

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

SELECT dev\_in AS programmer\_name,

MAX(dcost) AS costliest\_package,

MIN(dcost) AS cheapest\_package

FROM software

GROUP BY dev\_in;

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

SELECT prof1 AS language,

AVG(dcost) AS avg\_development\_cost,

AVG(scost) AS avg\_cost,

AVG(sold) AS avg\_selling\_cost,

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

FROM software

GROUP BY prof1;

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

SELECT splace AS institute,

COUNT(\*) AS num\_courses,

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

FROM studies

GROUP BY splace;

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

SELECT splace AS institute, COUNT(\*) AS num\_students

FROM programmer

GROUP BY splace;

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

SELECT sex, COUNT(\*) AS num\_programmers

FROM programmer

GROUP BY sex;

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

SELECT dev\_in AS programmer\_name, COUNT(\*) AS num\_packages\_developed

FROM software

GROUP BY dev\_in;

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

SELECT prof1 AS language, COUNT(\*) AS num\_packages

FROM software

GROUP BY prof1;

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

SELECT prof1 AS language, COUNT(\*) AS num\_packages

FROM software

WHERE dcost < 1000

GROUP BY prof1;

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

SELECT prof1 AS language, AVG(scost - dcost) AS avg\_difference

FROM software

GROUP BY prof1;

**29) Display the TOTAL scost, dcsot and amount TOBE recovered for EACH programmer for whose dcost HAS NOT YET BEEN recovered.**

SELECT dev\_in AS programmer\_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 dev\_in

HAVING SUM(dcost) < SUM(scost);

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

SELECT MAX(salary) AS highest\_salary,

MIN(salary) AS lowest\_salary,

AVG(salary) AS average\_salary

FROM programmer

WHERE salary > 2000;

**Query 111:**

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

SELECT p.name, p.salary // using alias name programmer as p

FROM programmer p

JOIN software s ON p.name = s.dev\_in

WHERE (p.prof1 = 'C' OR p.prof2 = 'C')

ORDER BY p.salary DESC

LIMIT 1;

**2)Who is the highest paid female java programmer?**

SELECT p.name, p.salary

FROM programmer p

WHERE (p.prof1 = 'java' OR p.prof2 = 'java')

AND p.sex = 'F'

ORDER BY p.salary DESC

LIMIT 1;

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

SELECT p.prof1,(SELECT name

FROM programmer

WHERE prof1 = p.prof1

ORDER BY salary DESC

LIMIT 1) AS highest\_paid\_programmer

FROM programmer p

GROUP BY p.prof1;

**4) Who is the LEAST experienced programmer?**

SELECT name, date\_of\_join

FROM programmer

ORDER BY date\_of\_join DESC

LIMIT 1;

**5) Who is the MOST experienced programmer?**

SELECT name, date\_of\_join

FROM programmer

ORDER BY date\_of\_join

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 languages\_known

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 date\_of\_birth ASC

LIMIT 1;

As DBASE is not in the inserted value there is no result.

**8) Which institute has MOST NUMBER of students?**

SELECT splace AS institute, COUNT(name) AS num\_students

FROM studies

GROUP BY splace

ORDER BY num\_students DESC

LIMIT 1;

**9) 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 NOT (prof1 IN ('C', 'C++', 'Oracle', 'Dbase') OR prof2 IN ('C', 'C++', 'Oracle', 'Dbase'));

**10) Which is the COSTLIEST course?**

SELECT course

FROM studies

ORDER BY CAST(ccost AS DECIMAL) DESC

LIMIT 1;

**11) 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'

);

**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;

//When there are two equal number of students taken same courses then

SELECT course, COUNT(\*) AS num\_students

FROM studies

GROUP BY course

HAVING COUNT(\*) = (

SELECT MAX(num\_students)

FROM (

SELECT COUNT(\*) AS num\_students

FROM studies

GROUP BY course

) AS course\_counts

);

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

SELECT splace AS institute, course

FROM studies

WHERE CAST(ccost AS DECIMAL) < (

SELECT AVG(CAST(ccost AS DECIMAL))

FROM studies

);

**14) Which institute conducts COSTLIEST course?**

SELECT splace AS institute, course

FROM studies

WHERE CAST(ccost AS DECIMAL) = (

SELECT MAX(CAST(ccost AS DECIMAL))

FROM studies

);

**15) Which course has below AVERAGE number of students?**

SELECT course, COUNT(\*) AS num\_students

FROM studies

GROUP BY course

HAVING COUNT(\*) < (

SELECT AVG(num\_students)

FROM (

SELECT COUNT(\*) AS num\_students

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 ABS(CAST(ccost AS DECIMAL) - (

SELECT AVG(CAST(ccost AS DECIMAL))

FROM studies

)) <= 1000;

**18) Which package has the HIGEST development cost?**

SELECT name

FROM software

ORDER BY dcost DESC

LIMIT 1;

**19) Which package has the LOWEST selling cost?**

select name

from software

order by scost

limit 1;

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

select dev\_in

from software

order by sold asc

limit 1;

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

SELECT prof1

FROM programmer p

JOIN software s ON p.name = s.dev\_in

ORDER BY s.sold \* s.scost DESC

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(dcost - scost)

LIMIT 1;

**23) Which is the COSTLIEST package developed in PASCAL?**

SELECT name

FROM software

WHERE dev\_in IN (

SELECT name

FROM programmer

WHERE prof1 = 'Pascal'

)

ORDER BY dcost DESC

LIMIT 1;

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

SELECT prof1 AS language, COUNT(\*) AS num\_packages

FROM programmer

WHERE prof1 IS NOT NULL

GROUP BY prof1

ORDER BY COUNT(\*) DESC

LIMIT 1;

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

SELECT dev\_in AS programmer, COUNT(\*) AS num\_packages

FROM software

GROUP BY dev\_in

ORDER BY COUNT(\*) DESC

LIMIT 1;

**26) Who is the author of the COSTLIEST package?**

SELECT name AS author, title AS package\_title, scost AS selling\_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 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 p\_female.name, p\_female.salary

FROM programmer p\_female

JOIN (SELECT MAX(salary) AS max\_salary FROM programmer WHERE sex = 'M') p\_male

ON p\_female.salary > p\_male.max\_salary

WHERE p\_female.sex = 'F';

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

SELECT prof1 AS language, COUNT(\*) AS num\_programmers

FROM programmer

WHERE prof1 IS NOT NULL

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, title AS package\_title, dcost AS development\_cost, scost AS selling\_cost

FROM software

WHERE scost > (2 \* dcost);

**31) Display programmer names and CHEAPEST package developed by them in EACH language? 32) Who is the YOUNGEST male programmer born in 1965?**

SELECT p.name AS programmer\_name, s.name AS package\_name, s.title AS package\_title, s.dev\_in AS language, s.dcost AS development\_cost

FROM (

SELECT dev\_in, MIN(dcost) AS min\_cost

FROM software

GROUP BY dev\_in

) AS cheapest\_packages

JOIN software s ON cheapest\_packages.dev\_in = s.dev\_in AND cheapest\_packages.min\_cost = s.dcost

JOIN programmer p ON s.dev\_in = p.name;

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

WITH highest\_selling AS (

SELECT dev\_in, MAX(sold) AS max\_sold

FROM software

GROUP BY dev\_in

),

lowest\_selling AS (

SELECT dev\_in, MIN(sold) AS min\_sold

FROM software

GROUP BY dev\_in

)

SELECT p.name AS programmer\_name,

hs.dev\_in AS language\_highest\_selling,

ls.dev\_in AS language\_lowest\_selling

FROM programmer p

JOIN highest\_selling hs ON p.name = hs.dev\_in

JOIN lowest\_selling ls ON p.name = ls.dev\_in;

**34) Who is the OLDEST female programmer WHO joined in 1992 6**

SELECT name

FROM programmer

WHERE sex = 'F' AND date\_of\_join = '1992-01-01'

ORDER BY date\_of\_birth ASC

LIMIT 1;

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

SELECT YEAR(date\_of\_birth) AS birth\_year, COUNT(\*) AS num\_programmers

FROM programmer

GROUP BY YEAR(date\_of\_birth)

ORDER BY COUNT(\*) DESC

LIMIT 1;

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

SELECT MONTH(date\_of\_join) AS join\_month, COUNT(\*) AS num\_programmers

FROM programmer

GROUP BY MONTH(date\_of\_join)

ORDER BY COUNT(\*) DESC

LIMIT 1;

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

SELECT language, COUNT(\*) AS num\_programmers

FROM (

SELECT prof1 AS language FROM programmer

UNION ALL

SELECT prof2 AS language FROM programmer

) AS combined\_languages

WHERE language IS NOT NULL

GROUP BY language

ORDER BY COUNT(\*) DESC

LIMIT 1;

**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.dev\_in = p.name

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

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

SELECT s.\*

FROM software s

JOIN programmer p ON s.dev\_in = p.name

WHERE p.sex = 'F' AND s.dev\_in = 'PASCAL';

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

SELECT \*

FROM programmer

WHERE date\_of\_join < '1990-01-01';

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

SELECT s.\*

FROM software s

JOIN programmer p ON s.dev\_in = p.name

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

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

SELECT p.splace AS institute,

p.name AS programmer\_name,

COUNT(s.name) AS num\_packages,

SUM(s.sold) AS total\_copies\_sold,

SUM(s.sold \* (s.scost - s.dcost)) AS total\_sales\_value

FROM programmer p

LEFT JOIN software s ON p.name = s.dev\_in

GROUP BY p.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 programmer

WHERE sex = 'M'

GROUP BY splace

ORDER BY COUNT(\*) DESC

LIMIT 1

) AS most\_popular\_institute ON most\_popular\_institute.splace = p.splace

WHERE 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.dev\_in = p.name

WHERE (p.sex = 'M' AND p.date\_of\_birth < '1965-01-01')

OR (p.sex = 'F' AND p.date\_of\_birth > '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.dev\_in = 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.dev\_in = p.name

WHERE s.dev\_in NOT IN (p.prof1, p.prof2) AND (p.prof1 IS NOT NULL AND p.prof2 IS NOT NULL);

**11) Display details of software developed by male students of HARI.**

SELECT s.\*

FROM software s

JOIN programmer p ON s.dev\_in = p.name

WHERE p.sex = 'M' AND p.name = 'HARI';

**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.dev\_in

WHERE s.dev\_in IS NULL;

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

SELECT SUM(dcost) AS total\_cost

FROM software

WHERE dev\_in = 'APPLE';

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

SELECT p1.name, p2.name

FROM programmer p1

JOIN programmer p2 ON p1.name <> p2.name

WHERE p1.date\_of\_join = p2.date\_of\_join;

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

SELECT p1.name, p2.name

FROM programmer p1

JOIN programmer p2 ON p1.name <> p2.name

WHERE p1.prof2 = p2.prof2;

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

SELECT p.splace AS institute,

SUM(s.sold \* (s.scost - s.dcost)) AS total\_sales\_value

FROM programmer p

JOIN software s ON p.name = s.dev\_in

GROUP BY p.splace;

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

SELECT p.splace

FROM programmer p

JOIN software s ON p.name = s.dev\_in

WHERE s.scost - s.dcost = (

SELECT MAX(scost - dcost)

FROM software

);

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

SELECT DISTINCT prof\_language

FROM (

SELECT prof1 AS prof\_language FROM programmer

UNION

SELECT prof2 AS prof\_language FROM programmer

) AS all\_languages

WHERE prof\_language NOT IN (

SELECT DISTINCT dev\_language

FROM software

);

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

SELECT p.salary, s.course

FROM programmer p

JOIN software s ON p.name = s.dev\_in

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?**

**21) Which is the COSTLIEST package developed by a person with under 3 year’s expenences?**

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

SELECT p.name, s.course, CEIL(ccost / (salary / 12)) AS months\_to\_recover

FROM programmer p

JOIN studies s ON p.name = s.name;

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

WITH LowestFee AS (

SELECT MIN(CAST(ccost AS DECIMAL)) AS min\_fee

FROM studies

)

SELECT COUNT(\*) AS num\_packages

FROM software

WHERE dev\_in IN (

SELECT name

FROM programmer

WHERE splace IN (

SELECT splace

FROM studies

WHERE CAST(ccost AS DECIMAL) = (SELECT min\_fee FROM LowestFee)

)

);

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

SELECT p.name AS programmer\_name,

p.splace AS institute,

COUNT(\*) AS num\_packages\_developed

FROM programmer p

JOIN software s ON p.name = s.dev\_in

WHERE s.dcost = (

SELECT MIN(dcost)

FROM software

)

GROUP BY p.name, p.splace;

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

SELECT COUNT(\*) AS num\_packages\_developed

FROM software s

JOIN programmer p ON s.dev\_in = 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\_developed

FROM software s

JOIN programmer p ON s.dev\_in = p.name

WHERE p.splace = 'BDPS' AND p.date\_of\_join = (

SELECT MIN(date\_of\_join)

FROM programmer

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,

p.splace AS institute,

COALESCE(COUNT(s.name), 0) AS num\_packages\_developed

FROM programmer p

LEFT JOIN software s ON p.name = s.dev\_in

GROUP BY p.name, p.splace;

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

SELECT prof1 AS programming\_language,

COUNT(DISTINCT name) AS num\_programmers,

COUNT(name) AS num\_packages\_developed

FROM programmer

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.dev\_in

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 = 'S.S.I.L'

);