NAME: KAVI BALAJI E

COLLEGE: BIT

ASSIGNMENT ON SQL QUERIES

FIRST CREATE TABLE FOR PROGRAMMER

```
CREATE TABLE Programmer (
    name VARCHAR2(8) NOT NULL,
    dob DATE NOT NULL,
    doj DATE NOT NULL,
    gender  VARCHAR2(1) NOT NULL,
    prof1 VARCHAR2(8),
    prof2 VARCHAR2(8),
    salary NUMBER(4) NOT NULL,
    CONSTRAINT CHECK (gender(‘male’,’female’));
    CONSTRAINT pk_programmer PRIMARY KEY (name)
);
INSERT INTO Programmer (name, dob, doj, gender, prof1, prof2, salary)
VALUES (('somdutt', '1966-04-21', '1992-04-21', 'm', 'pascal', 'basic', 3200),
('John', '1975-10-15', '1999-08-25', 'm', 'java', 'python', 4500),
 ('Alice', '1982-03-28', '2005-09-12', 'f', 'c++', 'COBOL', 3800),
 ('Emma', '1990-07-12', '2012-05-30', 'f', 'python', 'java', 4200));
```

TABLE FOR SOFTWARE

```
CREATE TABLE Software (
    name VARCHAR2(8) NOT NULL,
    title VARCHAR2(20) NOT NULL,
    dev_in VARCHAR2(8) NOT NULL,
    scost NUMBER(7,2),
    dcost NUMBER(5),
    sold NUMBER(3),
    CONSTRAINT pk_software PRIMARY KEY (name),
    CONSTRAINT fk_dev_in FOREIGN KEY (dev_in) REFERENCES Programmer (prof1)
);
INSERT INTO Software (name, title, dev_in, scost, dcost, sold)
VALUES ('somdutt', 'parhchutes', 'basic', 399.95, 6000, 43);
```

TABLE STUDIES

```
CREATE TABLE Studies (
   name VARCHAR2(8) NOT NULL,
   splace VARCHAR2(9) NOT NULL,
   course VARCHAR2(5) NOT NULL,
   ccost VARCHAR2(5) NOT NULL
);
INSERT INTO Studies (name, splace, course, ccost)
VALUES( ('somdutt', 'sabhari', 'pgdca', '4500'),
 ('devdutt', 'bdps', 'dcs', '500'));
```

QUERIES I

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

Answer:

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

2.Display the names and ages of all programmers.

Answer:

```
SELECT name, FLOOR((SYSDATE - dob) / 365.25) AS age FROM Programmer;
```

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

Answer:

```
SELECT p.name, FLOOR((SYSDATE - p.dob) / 365.25) 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?

Answer:

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

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

Answer:

```
SELECT name, TO_CHAR(dob, 'DD-MON-YYYY') AS date_of_birth FROM Programmer WHERE
TO_CHAR(dob, 'MM') = '01';
```

6.Display lowest course fee.

Answer:

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

7.How many programmer has done PGDCA course.

Answer:

```
SELECT COUNT(*) AS pgdca_count FROM Studies WHERE course = 'pgdca';
```

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

Answer:

```
SELECT SUM(sold * scost) AS total_revenue FROM Software WHERE dev_in = 'c';
```

9.Display the details of software developed by Ramesh?

Answer:

```
SELECT * FROM Software WHERE name = 'ramesh';
```

10.How many programmers studied at SABHARI.

Answer:

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

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

Answer:

```
SELECT name, CEIL(dcost / scost) AS copies_to_recover_cost FROM Software;
```

12.What is the price of the costliest software developed in BASIC?

Answer:

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

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

Answer:

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

14.How many packages were developed in dbase?

Answer:

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

15.How many programmers studies at paragathi?

Answer:

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

16.How many programmers paid 5000 to 10000 for their course?

Answer:

```
SELECT COUNT(*) AS programmers_paid_5000_to_10000 FROM Studies WHERE ccost BETWEEN
5000 AND 10000;
```

17.What is the average course fee?

Answer:

```
SELECT AVG(ccost) AS average_course_fee FROM Studies;
```

18.Display the details of programmers knowing c?

Answer:

```
SELECT p.* FROM Programmer p JOIN Studies s ON p.name = s.name WHERE s.course = 'c';
```

19.How many programmers know either Cobol or Pascal?

Answer:

```
SELECT COUNT(DISTINCT name) AS programmers_knowing_cobol_or_pascal FROM Studies WHERE
course IN ('cobol', 'pascal');
```

20.How many programmers don't know Pascal & C?

Answer:

```
SELECT COUNT(*) AS programmers_not_knowing_pascal_and_c FROM Programmer WHERE name
NOT IN ( SELECT DISTINCT name FROM Studies WHERE course IN ('pascal', 'c') );
```

21.How old is the oldest male programmer?

Answer:

```
SELECT FLOOR((SYSDATE - MAX(dob)) / 365.25) AS oldest_male_programmer_age FROM
Programmer WHERE gender = 'm';
```

22.What is the average age of female programmers?

Answer:

```
SELECT AVG(FLOOR((SYSDATE - dob) / 365.25)) AS average_female_programmer_age FROM
Programmer WHERE gender = 'f';
```

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

Answer:

```
SELECT name, FLOOR((SYSDATE - doj) / 365.25) AS experience_years FROM Programmer ORDER BY
experience_years DESC;
```

24.How many female programmers are there?

Answer:

```
SELECT COUNT(*) AS female_programmers_count FROM Programmer WHERE gender = 'f';
```

25.What are the languages known by the male programmers?

Answer:

```
SELECT DISTINCT prof1 AS known_language FROM Programmer WHERE gender = 'm' UNION SELECT
DISTINCT prof2 AS known_language FROM Programmer WHERE gender = 'm';
```

26.What is the Average salary?

Answer:

```
SELECT AVG(salary) AS average_salary FROM Programmer;
```

27.How many people draw 2000 to 4000?

Answer:

```
SELECT COUNT(*) AS people_with_salary_2000_to_4000 FROM Programmer WHERE salary
BETWEEN 2000 AND 4000;
```

28.Display the details of those who don't know Clipper, Cobol, or Pascal?

Answer:

```
SELECT * FROM Programmer WHERE name NOT IN ( SELECT name FROM Studies WHERE course IN
('clipper', 'cobol', 'pascal') );
```

29.How many Female programmers knowing C are above 24 years of age?

Answer:

```
SELECT COUNT(*) AS female_programmers_knowing_c_above_24 FROM Programmer p JOIN Studies
s ON p.name = s.name WHERE p.gender = 'f' AND s.course = 'c' AND FLOOR((SYSDATE - p.dob) /
365.25) > 24;
```

30.Who are the programmers who will be celebrating their Birthday within a week?

Answer:

```
SELECT * FROM Programmer WHERE EXTRACT(MONTH FROM dob) = EXTRACT(MONTH FROM
SYSDATE + INTERVAL '7' DAY) AND EXTRACT(DAY FROM dob) BETWEEN EXTRACT(DAY FROM
SYSDATE) AND EXTRACT(DAY FROM SYSDATE + INTERVAL '7' DAY);
```

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

Answer:

```
SELECT * FROM Programmer WHERE SYSDATE - doj < INTERVAL '1' YEAR;
```

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

Answer:

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

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

Answer:

```
SELECT name, (dcost - (sold * scost)) AS amount_to_be_recovered FROM Software WHERE dcost > sold * scost;
```

34.List the packages which have not been sold so far?

Answer:

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

35.Find out the cost of the software developed by Mary?

Answer:

```
SELECT scost AS software_cost FROM Software WHERE name = 'Mary';
```

36.Display the institute’s names from the studies table without duplicates?

Answer:

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

37.How many different courses are mentioned in the studies table?

Answer:

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

38.Display the names of the programmers whose names contain 2 occurrences of the letter A?

Answer:

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

39.Display the names of programmers whose names contain up to 5 characters?

Answer:

```
SELECT name FROM Programmer WHERE LENGTH(name) <= 5;
```

40.How many female programmers knowing COBOL have more than 2 years experience?

Answer:

```
SELECT COUNT(*) AS female_cobol_programmers_with_more_than_2_years_exp FROM
Programmer p JOIN Studies s ON p.name = s.name WHERE p.gender = 'f' AND s.course = 'cobol' AND
FLOOR((SYSDATE - p.doj) / 365.25) > 2;
```

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

Answer:

```
SELECT MIN(LENGTH(name)) AS shortest_name_length FROM Programmer;
```

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

Answer:

```
SELECT name, gender, TO_CHAR(dob, 'DD/MM/YY') AS dob, TO_CHAR(doj, 'DD/MM/YY') AS doj
FROM Programmer;
```

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

Answer:

```
SELECT * FROM Programmer WHERE EXTRACT(DAY FROM dob) = LAST_DAY(dob);
```

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

Answer:

```
SELECT SUM(salary) AS total_salary_paid FROM Programmer WHERE gender = 'm' AND name NOT IN
( SELECT name FROM Studies WHERE course = 'cobol' );
```

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

Answer:

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 the same?

Answer:

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

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

Answer:

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

QUERIES II:

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

Answer:

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

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

Answer:

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

3.Display THE NUMBER OF male and female programmers.

Answer:

SELECT gender, COUNT(*) AS number_of_programmers FROM Programmer GROUP BY gender;

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

Answer:

SELECT dev_in AS language, MAX(scost) AS costliest_package, MAX(sold) AS highest_selling FROM Software GROUP BY dev_in;

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

Answer:

SELECT EXTRACT(YEAR FROM dob) AS birth_year, COUNT(*) AS number_of_people_born FROM Programmer GROUP BY EXTRACT(YEAR FROM dob);

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

Answer:

SELECT EXTRACT(YEAR FROM doj) AS joining_year, COUNT(*) AS number_of_people_joined FROM Programmer GROUP BY EXTRACT(YEAR FROM doj);

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

Answer:

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

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

Answer:

SELECT EXTRACT(MONTH FROM doj) AS joining_month, COUNT(*) AS number_of_people_joined FROM Programmer GROUP BY EXTRACT(MONTH FROM doj);

9.Display the language wise COUNTS of prof1.

Answer:

SELECT prof1 AS language, COUNT(*) AS prof1_count FROM Programmer GROUP BY prof1;

10.Display the language wise COUNTS of prof2.

Answer:

SELECT prof2 AS language, COUNT(*) AS prof2_count FROM Programmer GROUP BY prof2;

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

Answer:

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

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

Answer:

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

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

Answer:

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

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

Answer:

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.

Answer:

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.

Answer:

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.

Answer:

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

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

Answer:

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

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

Answer:

```
SELECT name, dev_in AS language, SUM(scost) AS total_selling_cost FROM Software GROUP BY
name, dev_in;
```

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

Answer:

```
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.

Answer:

```
SELECT dev_in AS language, AVG(dcost) AS average_development_cost, AVG(scost) AS average_cost,
SUM(scost) AS total_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.

Answer:

```
SELECT splace AS institute_name, 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.

Answer:

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

24.Display names of male and female programmers.

Answer:

```
SELECT name, gender FROM Programmer;
```

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

Answer:

```
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.

Answer:

```
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.

Answer:

```
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.

Answer:

```
SELECT dev_in AS language, AVG(scost - dcost) AS average_cost_difference FROM Software GROUP BY dev_in;
```

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

Answer:

```
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.

Answer:

```
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?

Answer:

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

2.Who is the highest paid female COBOL programmer?

Answer:

```
SELECT name FROM Programmer WHERE gender = 'female' AND prof1 = 'COBOL' ORDER BY salary DESC LIMIT 1;
```

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

Answer:

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

4.Who is the least experienced programmer?

Answer:

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

5.Who is the most experienced programmer?

Answer:

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

6.Which language is known by only one programmer?

Answer:

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

7.Who is the youngest programmer knowing DBASE?

Answer:

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

8.Which institute has the most number of students?

Answer:

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

9.Who is the above programmer?

Answer:

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

10.Which female programmer earns more than 3000/- but does not know C, C++, Oracle, or DBASE?

Answer:

```
SELECT name FROM Programmer WHERE gender = 'female' AND salary > 3000 AND prof1 NOT IN ('C', 'C++', 'Oracle', 'DBASE');
```

11.Which is the costliest course?

Answer:

```
SELECT course FROM Studies ORDER BY ccost DESC LIMIT 1;
```

12.Which course has been done by most of the students?

Answer:

```
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?

Answer:

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

14.Which institute conducts the costliest course?

Answer:

```
SELECT splace AS institute_name FROM Studies WHERE ccost = (SELECT MAX(ccost) FROM Studies);
```

15.Which course has below average number of students?

Answer:

```
SELECT course FROM Studies GROUP BY course HAVING COUNT(*) < (SELECT AVG(count) FROM
(SELECT COUNT(*) AS count FROM Studies GROUP BY course) AS sub);
```

16.Which institute conducts the above course?

Answer:

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

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

Answer:

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

18.Which package has the highest development cost?

Answer:

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

19.Which package has the lowest selling cost?

Answer:

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

20.Who developed the package which has sold the least number of copies?

Answer:

```
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 highest sales amount?

Answer:

```
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 difference between development and selling cost were sold?

Answer:

```
SELECT sold FROM Software WHERE ABS(scost - dcost) = (SELECT MIN(ABS(scost - dcost)) FROM
Software) LIMIT 1;
```

23.Which is the costliest package developed in Pascal?

Answer:

```
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 packages?

Answer:

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

25.Which programmer has developed the highest number of packages?

Answer:

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

26.Who is the author of the costliest package?

Answer:

```
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?

Answer:

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

28.Who are the female programmers earning more than the highest paid male programmers?

Answer:

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

29.Which language has been stated as prof1 by most of the programmers?

Answer:

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?

Answer:

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?

Answer:

SELECT p.name AS programmer_name, s.title AS cheapest_package FROM Programmer p JOIN Software s ON p.name = s.name WHERE (s.dev_in, s.scost) IN (SELECT dev_in, MIN(scost) FROM Software GROUP BY dev_in);

32.Who is the youngest male programmer born in 1965?

Answer:

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

33.Display language used by each programmer to develop the highest selling and lowest selling package.

Answer:

SELECT name, (SELECT dev_in FROM Software WHERE name = p.name AND scost = (SELECT MAX(scost) FROM Software WHERE name = p.name)) AS highest_selling_language, (SELECT dev_in FROM Software WHERE name = p.name AND scost = (SELECT MIN(scost) FROM Software WHERE name = p.name)) AS lowest_selling_language FROM Programmer p;

34.Who is the oldest female programmer who joined in 1992?

Answer:

SELECT name FROM Programmer WHERE gender = 'female' AND doj = (SELECT MAX(doj) FROM Programmer WHERE gender = 'female' AND EXTRACT(YEAR FROM doj) = 1992);

35.In which year were the most number of programmers born?

Answer:

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

36.In which month did the most number of programmers join?

Answer:

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

37.In which language are most of the programmers proficient?

Answer:

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

38.Who are the male programmers earning below the average salary of female programmers?

Answer:

```
SELECT name FROM Programmer WHERE gender = 'male' AND salary < (SELECT AVG(salary) FROM Programmer WHERE gender = 'female');
```

QUERIES IV:

1.Display the details of those who are drawing the same salary.

Answer:

```
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.

Answer:

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

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

Answer:

```
SELECT * FROM Software WHERE name IN ( SELECT name FROM Programmer WHERE gender = 'female' ) AND dev_in = 'PASCAL';
```

4.Display the details of the programmer who joined before 1990.

Answer:

```
SELECT * FROM Programmer WHERE doj < DATE '1990-01-01';
```

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

Answer:

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

6.Display number of packages, number of copies sold, and sales value of each programmer Institute-wise.

Answer:

```
SELECT p.splace AS institute, p.name AS programmer_name, COUNT(*) AS num_packages, SUM(sold)
AS total_copies_sold, SUM(scost) AS total_sales_value FROM Programmer p JOIN Software s ON
p.name = s.name 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 programmers studies.

Answer:

```
SELECT * FROM Software WHERE name IN ( SELECT name FROM Programmer WHERE gender =
'male' AND splace = ( SELECT splace FROM Programmer GROUP BY splace ORDER BY COUNT(*) DESC
LIMIT 1 ) ) AND 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.

Answer:

```
SELECT * FROM Software WHERE name IN ( SELECT name FROM Programmer WHERE (gender =
'male' AND dob < DATE '1965-01-01') OR (gender = 'female' AND dob > DATE '1975-01-01') );
```

9.Display the details of the software that was developed in the language that is not the programmer's first proficiency.

Answer:

```
SELECT * FROM Software WHERE dev_in NOT IN ( SELECT prof1 FROM Programmer );
```

10.Display details of software that was developed in the language which is neither first nor second proficiency of the programmer.

Answer:

```
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.

Answer:

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

12.Display the names of programmers who have not developed any package.

Answer:

```
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?

Answer:

```
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?

Answer:

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

15.Who are the programmers who have the same prof2?

Answer:

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

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

Answer:

```
SELECT p.splace AS institute, SUM(s.sold * s.scost) AS total_sales_value FROM Programmer p 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?

Answer:

```
SELECT p.splace AS institute FROM Programmer p JOIN Software s ON p.name = s.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?

Answer:

```
SELECT prof1 AS language FROM Programmer WHERE prof1 NOT IN ( SELECT dev_in FROM Software
UNION 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?

Answer:

```
SELECT p.salary AS earnings, s.prof1 AS course 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?

Answer:

```
 SELECT name, salary / ccost AS months_to_recover FROM Programmer p FULLJOIN Studies s ON
p.name = s.name;
```

21.Which is the costliest package developed by a person with under 3 years' experience?

Answer:

```
SELECT * FROM Software WHERE name IN ( SELECT name FROM Programmer WHERE
(EXTRACT(YEAR FROM CURRENT_DATE) - EXTRACT(YEAR FROM 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?

Answer:

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

23.How many packages were developed by the students who studied in the institute that charges the lowest course fee?

Answer:

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

24.How many packages were developed by the person who developed the cheapest package? Where did he/she study?

Answer:

```
SELECT COUNT(*) AS num_packages FROM Software WHERE name = ( SELECT name FROM Software
WHERE scost = (SELECT MIN(scost) FROM Software) );
```

25.How many packages were developed by female programmers earning more than the highest paid male programmer?

Answer:

```
SELECT COUNT(*) AS num_packages FROM Software WHERE name IN ( SELECT name FROM
Programmer WHERE gender = 'female' AND salary > (SELECT MAX(salary) FROM Programmer WHERE
gender = 'male') );
```

26.How many packages were developed by the most experienced programmers from BDPS?

Answer:

```
SELECT COUNT(*) AS num_packages FROM Software WHERE name IN ( SELECT name FROM
Programmer WHERE splace = 'BDPS' ORDER BY EXTRACT(YEAR FROM CURRENT_DATE) -
EXTRACT(YEAR FROM doj) DESC LIMIT 1 );
```

27.List the programmers (from software table) and institutes they studied, including those who didn't develop any package.

Answer:

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

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

Answer:

```
SELECT prof1 AS proficiency, COUNT(DISTINCT name) AS num_programmers, COUNT(*) AS
num_packages FROM Programmer JOIN Software ON Programmer.name = Software.name GROUP BY
prof1;
```

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

Answer:

```
SELECT p.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.

Answer:

```
SELECT p.* FROM Programmer p JOIN Studies s ON p.name = s.name WHERE s.splace = 'S.S.I.L';
```