**Title:** SQL ASSIGNMENT

**Author:** Janani.D.P

**Created Date:** 04.04.2024

**Last modified Date:** 14.04.2024

**Reviewed By:**

**Reviewed Date:**

**Table : Programmer**

CREATE TABLE `jananidb1`.`software` (

`name` VARCHAR(8) NOT NULL,

`dob` DATE NOT NULL,

`doj` DATE NOT NULL,

`sex` VARCHAR(1) NOT NULL,

`prof1` VARCHAR(8) NULL,

`prof2` VARCHAR(8) NULL,

`salary` INT NOT NULL);

**Table: Software**

CREATE TABLE `jananidb1`.`software` (

`name` VARCHAR(8) NOT NULL,

`title` VARCHAR(20) NOT NULL,

`dev\_in` VARCHAR(8) NOT NULL,

`scost` DECIMAL(7,2) NULL,

`dcost` INT(7) NULL,

`sold` INT(3) NULL);

**Table: Studies**

CREATE TABLE `jananidb1`.`studies` (

`name` VARCHAR(8) NOT NULL,

`splace` VARCHAR(9) NOT NULL,

`course` VARCHAR(5) NOT NULL,

`ccost` VARCHAR(5) NOT NULL);

QUERY 1:

1. SELECT AVG(scost\*sold) from software JOIN programmer ON software.name=programmer.name WHERE programmer.prof1="pascal";
2. SELECT name, timestampdiff(YEAR, dob, CURDATE()) AS age from programmer;
3. SELECT programmer.name,ABS(timestampdiff(YEAR,programmer.dob,CURDATE())) AS age from programmer JOIN studies ON programmer.name=studies.name WHERE studies.course="dcs";
4. SELECT MAX(sold) AS maximumsold from software;
5. SELECT name, dob from programmer WHERE MONTH(dob) = 1;
6. SELECT MIN(ccost) AS lowestcoursefee from studies;
7. SELECT COUNT(\*) AS pgdca\_programmers from studies WHERE course = 'PGDCA';
8. SELECT SUM(scost \* sold) AS revenue from software WHERE title = 'C';
9. SELECT \* from software WHERE dev\_in = 'Ramesh';
10. SELECT COUNT(\*) AS num\_programmers\_sabhari from studies WHERE splace = 'SABHARI';
11. SELECT \* FROM Software WHERE (sold \* scost) > 20000;
12. SELECT CEIL(dcost / scost) AS recover\_copy from software;
13. SELECT MAX(scost) AS costliest\_software from software WHERE dev\_in = 'BASIC';
14. SELECT \* from software WHERE sold >= dcost / scost;
15. SELECT COUNT(\*) FROM software WHERE dev\_in = 'dbase';
16. SELECT COUNT(\*) FROM studies WHERE splace = 'paragathi';
17. SELECT COUNT(\*) FROM studies WHERE CAST(ccost AS DECIMAL(7,2)) BETWEEN 5000 AND 10000;
18. SELECT AVG(CAST(ccost AS DECIMAL(7,2))) AS average\_course\_fee from studies;
19. SELECT \* FROM programmer WHERE prof1 = 'C' OR prof2 = 'C';
20. SELECT COUNT(\*) from programmer WHERE prof1 = 'Cobol' or prof2 = 'Pascal';
21. SELECT COUNT(\*) from programmer WHERE prof1 != 'Pascal' AND prof1 != 'C' AND prof2 != 'Pascal' AND prof2 != 'C';
22. SELECT MAX(TIMESTAMPDIFF(YEAR, dob, CURDATE())) AS eldest\_male\_programmer from programmer WHERE sex = 'M';
23. SELECT AVG(TIMESTAMPDIFF(YEAR, dob, CURDATE())) AS average\_femaleprog\_age from programmer WHERE sex = 'F';
24. SELECT name, TIMESTAMPDIFF(YEAR, doj, CURDATE()) AS experience from programmer ORDER BY experience DESC;
25. SELECT name, dob from programmer WHERE MONTH(dob) = MONTH(CURDATE());
26. SELECT COUNT(\*) from programmer WHERE sex = 'F';
27. SELECT AVG(salary) AS avg\_salary from programmer;
28. SELECT COUNT(\*) from programmer WHERE salary BETWEEN 2000 AND 4000;
29. SELECT COUNT(\*) AS draw\_2000\_to\_4000 from programmer WHERE salary BETWEEN 2000 AND 4000;
30. SELECT \* from programmer WHERE prof1 NOT IN ('Clipper', 'Cobol', 'Pascal') AND prof2 NOT IN ('Clipper', 'Cobol', 'Pascal');
31. SELECT COUNT(\*) from programmer WHERE sex = 'F' AND prof1 = 'C' AND TIMESTAMPDIFF(YEAR, dob, CURDATE()) > 24;
32. SELECT \* from programmer WHERE DATE\_ADD(dob, INTERVAL YEAR(CURDATE())-YEAR(dob) + IF(DAYOFYEAR(dob) > DAYOFYEAR(CURDATE()), 0, 1) YEAR) BETWEEN CURDATE() AND DATE\_ADD(CURDATE(), INTERVAL 7 DAY);
33. SELECT \* from programmer WHERE TIMESTAMPDIFF(MONTH, doj, CURDATE()) < 12;
34. SELECT \* from programmer WHERE TIMESTAMPDIFF(YEAR, doj, CURDATE()) = 2;
35. SELECT (dcost - (sold \* scost)) from software WHERE dcost > (sold \* scost);
36. SELECT \* FROM software WHERE sold IS NULL OR sold = 0;
37. SELECT SUM(scost) FROM software WHERE dev\_in = 'Mary';
38. SELECT DISTINCT splace from studies;
39. SELECT COUNT(DISTINCT course) from studies;
40. SELECT name from programmer WHERE LENGTH(name) - LENGTH(REPLACE(name, 'A', '')) = 2;
41. SELECT name from programmer WHERE LENGTH(name) <= 5;
42. SELECT COUNT(\*) from programmer WHERE sex = 'F' AND prof1 = 'COBOL' AND TIMESTAMPDIFF(YEAR, doj, CURDATE()) > 2;
43. SELECT MIN(CHAR\_LENGTH(name)) from programmer;
44. SELECT AVG(dcost) from software WHERE dev\_in = 'COBOL';
45. SELECT name, sex, DATE\_FORMAT(dob, '%d/%m/%y') AS dob, doj from programmer;
46. SELECT \* from programmer WHERE DAY(LAST\_DAY(dob)) = DAY(dob);
47. SELECT SUM(salary) FROM programmer WHERE sex = 'M' AND (prof1 != 'Cobol' AND prof2 != 'Cobol');
48. SELECT title, scost, dcost, (scost - dcost) AS difference from software ORDER BY difference DESC;
49. SELECT name, dob, doj FROM programmer WHERE MONTH(dob) = MONTH(doj);
50. SELECT name FROM software WHERE title LIKE '% %';

**QUERY 2**:

1. SELECT dev\_in, COUNT(\*) FROM software GROUP BY dev\_in;
2. SELECT name, COUNT(dev\_in) from software GROUP BY name;
3. SELECT sex, COUNT(\*) AS num\_programmers from programmer GROUP BY sex;
4. SELECT dev\_in, MAX(dcost) AS costliest\_package, MAX(sold) AS highest\_selling from software GROUP BY dev\_in;
5. SELECT YEAR(dob), COUNT(\*) FROM programmer GROUP BY YEAR(dob);
6. SELECT YEAR(doj), COUNT(\*) FROM programmer GROUP BY YEAR(doj);
7. SELECT MONTH(dob), COUNT(\*) FROM programmer GROUP BY MONTH(dob);
8. SELECT MONTH(doj), COUNT(\*) FROM programmer GROUP BY MONTH(doj);
9. SELECT distinct prof1, COUNT(\*) AS COUNT FROM programmer GROUP BY prof1;
10. SELECT distinct prof2, COUNT(\*) AS COUNT FROM programmer GROUP BY prof2;
11. SELECT salary, COUNT(\*) AS Count FROM programmer GROUP BY salary;
12. SELECT splace, COUNT(\*) AS num\_people FROM studies GROUP BY splace;
13. SELECT course, COUNT(\*) AS num\_people\_course from studies GROUP BY course;
14. SELECT dev\_in, SUM(dcost) AS tot\_dev\_cost from software GROUP BY dev\_in;
15. SELECT dev\_in, SUM(scost) AS tot\_selling\_cost from software GROUP BY dev\_in;
16. SELECT name, SUM(dcost) from software GROUP BY name;
17. SELECT name, SUM(sold) from software GROUP BY name;
18. SELECT name, COUNT(\*) from software GROUP BY name;
19. SELECT name, SUM(scost \* sold) from software GROUP BY name;
20. SELECT dev\_in, MAX(dcost) , MIN(dcost) FROM Software GROUP BY dev\_in;
21. SELECT dev\_in, AVG(dcost), AVG(scost), AVG(scost / sold) FROM software GROUP BY dev\_in;
22. SELECT splace, COUNT(course), AVG(ccost) FROM studies GROUP BY splace;
23. SELECT splace, COUNT(DISTINCT name) FROM studies GROUP BY splace;
24. SELECT DISTINCT name, sex FROM programmer;
25. SELECT name, dev\_in AS packages From software;
26. SELECT dev\_in, COUNT(\*) FROM software GROUP BY dev\_in;
27. SELECT dev\_in, COUNT(\*) FROM software WHERE dcost < 1000 GROUP BY dev\_in;
28. SELECT dev\_in, avg(scost - dcost) as avg\_difference From software GROUP BY dev\_in;
29. SELECT name, SUM(scost), SUM(dcost), SUM(scost - dcost) FROM software GROUP BY name HAVING SUM(scost) > SUM(dcost);
30. SELECT MAX(salary), MIN(salary), AVG(salary) From programmer where salary > 2000;

**QUERY 3:**

1. SELECT name, MAX(salary) AS highest\_salary FROM programmer WHERE prof1 = 'C' OR prof2 = 'C' group by name;
2. SELECT name, MAX(salary) AS high\_f\_cobol\_salary FROM programmer WHERE sex = 'F' AND (prof1 = 'COBOL' OR prof2 = 'COBOL') group by name;
3. SELECT prof1, name, MAX(salary) AS highest\_salary FROM programmer GROUP By prof1;
4. SELECT name, TIMESTAMPDIFF(YEAR, doj, CURDATE()) FROM programmer ORDER BY doj asc limit 1;
5. SELECT name, TIMESTAMPDIFF(YEAR, doj, CURDATE()) FROM programmer ORDER BY doj desc limit 1;
6. SELECT dev\_in AS language FROM software GROUP BY dev\_in HAVING COUNT(DISTINCT name) = 1;
7. SELECT name, dob FROM programmer WHERE prof1 = 'DBASE' OR prof2 = 'DBASE' ORDER BY dob ASC LIMIT 1;
8. SELECT splace, COUNT(\*) AS num\_students FROM studies GROUP BY splace ORDER BY num\_students DESC LIMIT 1;
9. SELECT name FROM programmer;
10. SELECT name, salary 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. SELECT course, MAX(ccost) From studies group by course;
12. SELECT course, COUNT(\*) FROM studies GROUP BY course ORDER BY COUNT(\*) DESC LIMIT 1;
13. SELECT splace, course FROM studies

WHERE ccost < (

SELECT avg(ccost)

From studies

)

1. SELECT splace, course, ccost FROM studies

WHERE ccost < (

SELECT avg(ccost)

From studies

)

1. SELECT course FROM Studies GROUP BY course HAVING COUNT(\*) < (SELECT AVG(count) FROM (SELECT COUNT(\*) AS count FROM Studies GROUP BY course) AS course\_count);
2. SELECT splace FROM studies WHERE course = (SELECT course FROM Studies GROUP BY course HAVING COUNT(\*) < (SELECT AVG(count) FROM (SELECT COUNT(\*) AS count FROM studies GROUP BY course) AS course\_count));
3. SELECT course FROM studies WHERE ccost BETWEEN (SELECT AVG(ccost) - 1000 FROM studies) AND (SELECT AVG(ccost) + 1000 FROM studies);
4. SELECT name, MAX(dcost) FROM software group by name;
5. SELECT name, Min(dcost) AS lowest\_development\_cost FROM software group by name;
6. SELECT name FROM software WHERE sold = (SELECT MIN(sold) FROM software);
7. SELECT dev\_in FROM software

WHERE scost \* sold = (

SELECT MAX(scost \* sold)

FROM software

);

1. SELECT sold FROM software

WHERE ABS(dcost - scost) = (

SELECT MIN(ABS(dcost - scost))

FROM software

);

1. SELECT name, MIN(dcost) FROM software WHERE dev\_in = 'Pascal' group by name;
2. SELECT dev\_in FROM software GROUP BY dev\_in ORDER BY COUNT(\*) DESC LIMIT 1;
3. SELECT name FROM software GROUP BY name ORDER BY COUNT(\*) DESC LIMIT 1;
4. SELECT name FROM software

WHERE dcost = (

SELECT MAX(dcost)

FROM software

);

1. SELECT name FROM software

WHERE sold < (

SELECT AVG(sold)

FROM software

);

1. SELECT name FROM programmer

WHERE sex = 'F' AND salary > (

SELECT MAX(salary)

FROM programmer

WHERE sex = 'M'

);

1. SELECT prof1 FROM programmer GROUP BY prof1 ORDER BY COUNT(\*) Desc limit 1;
2. SELECT name FROM software WHERE scost > (2 \* dcost) group by name;
3. SELECT name, dev\_in, MIN(scost) FROM software GROUP BY name, dev\_in;
4. SELECT name FROM programmer WHERE dob = (SELECT MAX(dob) FROM programmer WHERE YEAR(dob) = 1965 AND sex = 'M');
5. SELECT p.name,MAX(s1.dev\_in),MIN(s2.dev\_in) FROM programmer p LEFT JOIN software s1 ON p.name = s1.name LEFT JOIN software s2 ON p.name = s2.name GROUP BY p.name;
6. SELECT name FROM programmer WHERE dob = (SELECT Min(dob) FROM programmer WHERE YEAR(dob) = 1992 AND sex = 'F');
7. SELECT YEAR(dob) FROM Programmer GROUP BY YEAR(dob) ORDER BY COUNT(\*) DESC LIMIT 1;
8. SELECT DATE(doj) FROM Programmer GROUP BY DATE(doj) ORDER BY COUNT(\*) DESC LIMIT 1;
9. SELECT prof AS language, COUNT(\*) AS programmer\_count FROM (

select prof1 AS prof FROM programmer

Union ALL

SELECT prof2 FROM programmer

) AS proficiencies

GROUP BY prof

ORDER BY programmer\_count DESC

LIMIT 1;

1. SELECT name

FROM programmer

WHERE sex = 'M' AND salary < (

SELECT AVG(salary)

FROM programmer

WHERE sex = 'F'

);

**QUERY 4:**

1. SELECT \* FROM programmer WHERE salary IN (SELECT salary FROM programmer GROUP BY salary HAVING COUNT(\*) > 1);
2. SELECT s.\* FROM software s JOIN programmer p ON s.name = p.name WHERE p.sex = 'M' AND p.salary > 3000;
3. SELECT s.\* FROM software s JOIN programmer p ON s.name = p.name WHERE p.sex = 'M' AND s.dev\_in = 'Pascal';
4. SELECT \* from programmer where year(doj) < 1990;
5. SELECT s.\* FROM software s JOIN programmer p ON s.name = p.name JOIN studies t ON p.name = t.name WHERE s.dev\_in = 'C' AND p.sex = 'f' AND t.splace = 'Pragathi';
6. SELECT p.name,

COUNT(s.name) AS num\_packages,

COALESCE(SUM(s.sold), 0) AS num\_copies\_sold,

COALESCE(SUM(s.sold \* s.scost), 0) AS total\_sales\_value

FROM programmer p LEFT JOIN software s ON p.name = s.name GROUP BY p.name;

1. SELECT s.\* FROM software s JOIN programmer p ON s.name = p.name JOIN (

SELECT name

FROM programmer

GROUP BY name

ORDER BY COUNT(\*) DESC

LIMIT 1

) AS sub ON p.name = sub.name

WHERE s.dev\_in = 'DBASE' AND p.sex = 'M';

1. SELECT \* FROM software WHERE name IN (

SELECT name

FROM programmer

WHERE (sex = 'M' AND YEAR(dob) < 1965)

OR (sex = 'F' AND YEAR(dob) > 1975)

);

1. SELECT \* FROM software Where dev\_in NOT IN (SELECT prof1 FROM programmer);
2. SELECT s.\* FROM software s JOIN programmer p ON s.name = p.name WHERE s.dev\_in NOT IN (p.prof1, p.prof2);
3. SELECT s.\* FROM software s join programmer p ON s.name = p.name JOIN studies st ON p.name = st.name WHERE p.sex = 'M' AND st.splace = 'SABHARI';
4. SELECT programmer.name FROM programmer LEFT JOIN software ON programmer.name = software.name WHERE software.name IS NULL;
5. SELECT SUM(s.dcost) FROM software s JOIN studies st ON s.name = st.name WHERE st.splace = 'APPLE';
6. SELECT p1.name, p2.name, p1.doj FROM programmer p1 JOIN programmer p2 ON p1.doj = p2.doj AND p1.name <> p2.name;
7. SELECT p1.name, p2.name, p1.prof2 FROM programmer p1 JOIN programmer p2 ON p1.prof2 = p2.prof2 AND p1.name <> p2.name;
8. SELECT st.splace, SUM(s.sold \* s.scost) FROM software s JOIN studies st ON s.name = st.name GROUP BY st.splace;
9. SELECT st.splace FROM software s JOIN studies st ON s.name = st.name WHERE s.dcost = ( SELECT MAX(dcost) FROM software );
10. SELECT prof FROM (

SELECT prof1 AS prof FROM programmer

UNION

SELECT prof2 AS prof FROM programmer

) AS all\_profs WHERE prof NOT IN ( SELECT dev\_in FROM software );

1. SELECT p.name, p.salary, st.course FROM programmer p JOIN software s ON p.name = s.name JOIN studies st ON p.name = st.name WHERE s.sold = ( SELECT MAX(sold) FROM software);
2. SELECT p.name, st.ccost / p.salary FROM programmer p JOIN studies st ON p.name = st.name;
3. SELECT s.\* FROM software s JOIN (SELECT p.name FROM programmer p WHERE TIMESTAMPDIFF(YEAR, p.doj, CURDATE()) < 3) AS expenseunder3 ON s.name = expenseunder3.name ORDER BY s.dcost DESC LIMIT 1;
4. SELECT AVG(p.salary) FROM programmer p JOIN software s ON p.name = s.name WHERE s.sold \* s.scost > 50000;
5. SELECT COUNT(\*) FROM software s JOIN studies st ON s.name = st.name WHERE st.splace = (SELECT splace FROM studies ORDER BY ccost ASC LIMIT 1);
6. SELECT COUNT(\*), st.splace FROM software s JOIN studies st ON s.name = st.name WHERE s.dcost = (SELECT MIN(dcost) FROM software) GROUP BY st.splace;
7. SELECT COUNT(\*) AS package\_count FROM software s JOIN programmer p ON s.name = p.name WHERE p.sex = 'F' AND p.salary > (SELECT MAX(salary) FROM programmer WHERE sex = 'M');
8. SELECT COUNT(\*) FROM software s JOIN programmer p ON s.name = p.name

WHERE TIMESTAMPDIFF(YEAR, p.doj, CURDATE()) = (

SELECT MAX(TIMESTAMPDIFF(YEAR, doj, CURDATE())) FROM programmer WHERE p.name IN (SELECT name FROM studies WHERE splace = 'BDPS')

);

1. SELECT p.name, IFNULL(st.splace, 'No Institute') FROM programmer p LEFT JOIN studies st ON p.name = st.name ORDER BY p.name;
2. SELECT prof1,COUNT(DISTINCT p.name) AS programmers\_count,COUNT(s.name) AS packages\_count FROM programmer p LEFT JOIN software s ON p.name = s.name GROUP BY prof1;
3. SELECT p.name AS programmer\_name, COUNT(s.name) FROM programmer p LEFT JOIN software s ON p.name = s.name GROUP BY p.name;
4. SELECT \* FROM programmer WHERE name IN (SELECT name FROM studies WHERE splace = 'S.S.I.L');