**Title of the application**: RDBMS Assignment

**Author**: S. Jayashree

**Created date**: 03/04/2024

**Last Modified date**:08/04/2024

**Reviewed by**:

**Reviewed Date**:

Table: Programmer

CREATE TABLE `programmer` (

`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 `software` (

`name` VARCHAR(8) NOT NULL,

`title` VARCHAR(20) NOT NULL,

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

`scost` DECIMAL(7,2) NULL,

`dcost` INT(5) NULL,

`sold` INT(3) NULL);

Table: Studies

CREATE TABLE `studies` (

`name` VARCHAR(8) NOT NULL,

`splace` VARCHAR(9) NOT NULL,

`course` VARCHAR(5) NOT NULL,

`ccost` VARCHAR(5) NOT NULL);

QUERIES I:

1. SELECT AVG(scost\*sold) FROM software JOIN programmer ON software.name=programmer.name WHERE programmer.prof1="pascal";
2. SELECT name, ABS(timestampdiff(YEAR,dob,CURDATE())) AS age FROM programmer;
3. SELECTprogrammer.name,ABS(timestampdiff(YEAR,programmer.dob,CURDATE())) AS age FROM programmer JOIN studies ON programmer.name=studies.name WHERE studies.course="dcs";
4. null
5. SELECT \* from programmer where month(dob)=1;
6. SELECT MIN(ccost) FROM studies;
7. SELECT COUNT(\*) from studies where course="pgdca"; [1]
8. SELECT scost\*sold AS Total\_revenue from software where dev\_in="c";
9. SELECT dev\_in from software where name="ramesh";
10. SELECT COUNT(\*) from studies where splace="sabhari";
11. SELECT \* FROM software WHERE scost\*sold>20000;
12. SELECT CEIL(dcost/scost) as Copies FROM software;
13. SELECT max(dcost) from software where dev\_in="C";
14. SELECT dev\_in from software where CEIL(dcost/scost)>sold;
15. SELECT COUNT(\*) FROM software WHERE dev\_in=”dbase”;
16. SELECT COUNT(\*) FROM studies WHERE splace="paragathi";
17. SELECT COUNT(\*) FROM studies WHERE ccost between 5000 and 10000;
18. SELECT avg(ccost) from studies;
19. SELECT \* from programmer where prof1 or prof2 = "c";
20. SELECT count(\*) FROM programmer WHERE prof1 = "pascal" or "Cobol" and prof2="pascal" or "Cobol";
21. SELECT count(\*) from programmer where NOT prof1 = "pascal" or "Cobol" and prof2="pascal" or "Cobol";
22. SELECT max(ABS(timestampdiff(YEAR,dob,CURDATE()))) from programmer where sex="m";
23. SELECT avg(ABS(timestampdiff(YEAR,dob,CURDATE()))) from programmer where sex="f";
24. SELECT name,ABS(timestampdiff(YEAR,doj,CURDATE())) from programmer order by name desc;
25. SELECT name FROM Programmer WHERE MONTH(dob) = MONTH(CURRENT\_DATE());
26. SELECT count(\*) from programmer where sex="f";
27. SELECT distinct prof1,prof2 from programmer where sex="m";
28. SELECT avg(salary) from programmer;
29. SELECT name from programmer where salary between 2000 and 4000;
30. SELECT name from programmer where not prof1="Clipper"or "Cobol" or "Pascal" and prof2="Clipper"or "Cobol" or "Pascal"; (or) SELECT name FROM programmer WHERE NOT (prof1 IN ('Clipper', 'Cobol', 'Pascal') AND prof2 IN ('Clipper', 'Cobol', 'Pascal'));
31. SELECT COUNT(\*) FROM programmer WHERE sex = 'f' AND prof1 = 'C' AND ABS(TIMESTAMPDIFF(YEAR, dob, CURDATE())) > 24;
32. SELECT name FROM programmer WHERE DATE\_ADD(dob, INTERVAL YEAR(CURRENT\_DATE())-YEAR(dob) YEAR) BETWEEN CURDATE() AND DATE\_ADD(CURDATE(), INTERVAL 7 DAY);
33. SELECT name FROM programmer WHERE ABS(timestampdiff(year,doj,CURDATE()))<2;
34. SELECT name from programmer where year(doj)=year(curdate())-2 and month(doj)<=month(curdate()) and day(doj)<=day(curdate());
35. SELECT (dcost-(scost\*sold)) from software where sold\*scost<dcost;
36. SELECT dev\_in from software where sold=0;
37. SELECT scost FROM software WHERE name="mary";
38. SELECT distinct splace from studies;
39. SELECT count(course) from studies;
40. SELECT name FROM programmer WHERE name REGEXP '(.\*a.\*){2}';
41. SELECT name FROM programmer WHERE CHAR\_LENGTH(name)<=5;
42. SELECT name FROM programmer WHERE sex=’f’ AND prof1 or prof2 =”COBOL” AND ABS(timestampdiff(year,doj,CURDATE()))>2;
43. SELECT MIN(CHAR\_LENGTH(name)) FROM programmer;
44. SELECT AVG(dcost) FROM software WHERE dev\_in=”COBAL”;
45. SELECT name, sex, dob,doj FROM programmer;
46. SELECT name from programmer where DAYOFMONTH(dob)=DAY(LASTDAY(dob));
47. SELECT salary FROM programmer WHERE NOT prof1 or prof2=”COBAL”;
48. SELECT title, scost, dcost, (scost - dcost) AS cost\_diff FROM software ORDER BY cost\_diff desc;
49. SELECT name, dob, doj FROM software WHERE MONTH(dob)=MONTH(doj);
50. SELECT name FROM packages WHERE name LIKE '% %';

QUERIES II

1. SELECT dev\_in AS language, COUNT(\*) AS num\_packages 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), MAX(scost) 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 Count FROM studies GROUP BY splace;
13. SELECT course, COUNT(\*) AS Count FROM studies GROUP BY course;
14. SELECT dev\_in, SUM(dcost) FROM software GROUP BY dev\_in;
15. SELECT dev\_in, scost FROM software GROUP BY dev\_in;
16. SELECT dev\_in, dcost FROM software GROUP BY dev\_in;
17. SELECT distinct sold AS total\_sales, name FROM software;
18. SELECT name, count(\*) from software group by name;
19. SELECT name,dev\_in, sum(scost) from software group by name,dev\_in;
20. SELECT name,MAX(scost),MIN(scost) from software group by name;
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(name) FROM studies group by splace;
24. SELECT name from programmer where sex=’m’ or sex=’f’ ;
25. SELECT name, dev\_in as packages from software;
26. SELECT dev\_in , count(\*) from software group by dev\_in;
27. SELECT distinct dev\_in, COUNT(\*) from software where MIN(dcost);
28. SELECT dev-in, Avg(scost-dcost) from software group by dev\_in;
29. SELECT sum(scost) as total\_scost, sum(dcost) as total\_dcost, sum(scost-dcost) from software where scost>dcost group by name;
30. SELECT MAX(salary), Min(salary), avg(salary) from programmer where salary>2000;

QUERIES -III

1. SELECT name from programmer where salary=(select max(salary) from programmer) and prof1="c" or prof2="c";
2. SELECT name from programmer WHERE salary=(select max(salary) from programmer) and prof1="cobal" or prof2="cobal" and sex=”f”;
3. SELECT max(salary), prof1 from programmer group by prof1;
4. SELECT name, timestampdiff(Year,doj,CURDATE()) FROM programmer ORDER BY timestampdiff(Year,doj,CURDATE())ASC LIMIT 1;
5. SELECT MAX(ABS(timestampdiff(Year,doj,CURDATE()))) AS Experienced\_Programmer FROM programmer;
6. SELECT prof1 AS language\_known FROM Programmer GROUP BY prof1 HAVING COUNT(\*) = 1 UNION SELECT prof2 AS language\_known FROM programmer GROUP BY prof2 HAVING COUNT(\*) = 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. -
10. SELECT name, salary FROM programmer WHERE sex = 'f' AND salary > 3000 AND NOT (prof1 IN ('C', 'C++', 'Oracle', 'Dbase') OR prof2 IN ('C', 'C++', 'Oracle', 'Dbase'));
11. SELECT course, ccost FROM studies order BY ccost desc limit 1;
12. SELECT course, count(\*) from studies order by count(\*) desc limit 1;
13. SELECT name, splace, course from studies WHERE ccost < (SELECT avg(ccost) from studies) order by ccost;
14. SELECT splace from studies order by ccost desc limit 1;
15. 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. -
17. SELECT dev\_in,dcost from software order by dcost desc limit 1;
18. SELECT dev\_in,scost from software order by scost asc limit 1;
19. SELECT name, sold from software order by sold asc limit 1;
20. SELECT dev\_in, scost from software order by scost desc limit 1;
21. SELECT name, sold FROM software WHERE ABS(dcost - scost) = ( SELECT MIN(ABS(dcost - scost)) FROM software) ORDER BY sold DESC LIMIT 1;
22. SELECT dev\_in, sold FROM software WHERE (scost - dcost) = (SELECT MIN(scost - dcost) FROM software);
23. SELECT dev\_in , MAX(dcost) from software where dev\_in='PASCAL';
24. SELECT dev\_in, COUNT(\*) AS num\_packages FROM software GROUP BY dev\_in ORDER BY num\_packages DESC LIMIT 1;
25. SELECT name, count(dev\_in) from software group by name order by count(dev\_in) desc limit 1;
26. SELECT name from software where scost=(SELECT MAX(scost) from software);
27. SELECT dev\_in FROM software WHERE sold< (SELECT AVG(sold) FROM software );
28. SELECT name, sex, salary FROM programmer WHERE sex= 'f' AND salary > (SELECT MAX(salary) FROM programmer WHERE sex= 'm');
29. 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. SELECT name FROM software WHERE scost > (2 \* dcost);
31. SELECT name, dev\_in, MIN(scost) AS cheapest\_package\_cost FROM software GROUP BY name, dev\_in;
32. SELECT name, dob FROM programmer WHERE sex = 'm' AND dob = (SELECT MIN(dob) FROM programmer WHERE sex = 'm' AND dob >= '1965-01-01' AND dob <= '1965-12-31');
33. SELECT name, dev\_in, MAX(scost) AS highest\_selling\_package, MIN(scost) AS lowest\_selling\_package FROM software GROUP BY name, dev\_in;
34. SELECT name, dob FROM programmer WHERE sex = 'f' AND doj >= '1992-01-01' AND doj <= '1992-12-31' ORDER BY dob DESCLIMIT 1;
35. SELECT YEAR(dob) AS birth\_year, COUNT(\*) AS num\_programmers FROM programmer GROUP BY birth\_year ORDER BY num\_programmers DESC LIMIT 1;
36. SELECT MONTHNAME(doj) AS join\_month, COUNT(\*) AS join\_count FROM programmer GROUP BY doj ORDER BY join\_count DESC LIMIT 1;
37. SELECT prof1, prof2, COUNT(\*) AS proficiency\_count FROM programmers WHERE prof1 IS NOT NULL AND prof2 IS NOT NULL GROUP BY prof1, prof2 ORDER BY proficiency\_count DESC LIMIT 1;
38. SELECT name, salary FROM programmer WHERE sex= 'm' AND salary < (SELECT AVG(salary) FROM programmer WHERE sex= 'f');

QUERY – IV

1. SELECT \* FROM programmer WHERE salary IN ( SELECT salary FROM programmer GROUP BY salary HAVING COUNT(\*) > 1);
2. SELECT dev\_in FROM software JOIN programmer ON software.name=programmer.name WHERE sex = 'm' AND salary > 3000;
3. SELECT dev\_in, dcost, scost from software join programmer on software.name=programmer.name WHERE sex = 'f' and 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 s.name, i.splace, COUNT(\*) AS num\_packages, SUM(s.sold) AS total\_copies\_sold, SUM(s.sold \* s.scost) AS total\_sales\_value FROM Software s JOIN studies i ON s.name = i.name GROUP BY s.name, i.splace;
7. 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 = ( SELECT splace FROM Studies GROUP BY splace ORDER BY COUNT(\*) DESC LIMIT 1 ) AND s.dev\_in = 'DBASE';
8. SELECT s.\* FROM Software s JOIN Programmer p ON s.name = p.name WHERE (p.sex = 'male' AND p.dob < '1965-01-01') OR (p.sex = 'female' AND p.dob > '1975-12-31');
9. SELECT s.\* FROM Software s JOIN Programmer p ON s.dev\_in != p.prof1 WHERE s.dev\_in IS NOT NULL;
10. SELECT s.\* FROM Software s LEFT JOIN Programmer p ON s.dev\_in = p.prof1 OR s.dev\_in = p.prof2 WHERE p.name IS NULL;
11. SELECT s.\* FROM software s JOIN programmer p ON s.name = p.name JOIN studies t ON p.name = t.name WHERE p.sex = 'm' AND t.splace = 'sabhari';
12. SELECT p.name FROM programmer p WHERE p.name NOT IN ( SELECT DISTINCT name FROM software );
13. SELECT SUM(s.scost) AS Total\_Cost FROM Programmer p JOIN Software s ON p.name = s.name WHERE p.name = 'APPLE';
14. SELECT name, doj FROM Programmer p WHERE EXISTS ( SELECT 1 FROM Programmer WHERE name <> p.name AND doj = p.doj);
15. SELECT p1.name AS Programmer1,

p2.name AS Programmer2,

p1.prof2 AS Shared\_Prof2

FROM Programmer p1

JOIN Programmer p2 ON p1.prof2 = p2.prof2

WHERE p1.name < p2.name;

1. SELECT st.splace , SUM(s.sold) AS Total\_Sales\_Value FROM Studies st JOIN Software s ON st.name = s.name GROUP BY st.splace;
2. SELECT st.splace AS Study\_Location

FROM Studies st

JOIN (

SELECT name

FROM Software

WHERE scost = (

SELECT MAX(scost)

FROM Software

)

LIMIT 1

) AS costliest\_package\_developer ON st.name = costliest\_package\_developer.name;

1. SELECT DISTINCT prof

FROM (

SELECT prof1 AS prof FROM Programmer

UNION

SELECT prof2 AS prof FROM Programmer

) AS all\_languages

LEFT JOIN Software ON all\_languages.prof = Software.dev\_in

WHERE Software.dev\_in IS NULL;

1. SELECT p.name AS Developer\_Name,

p.salary AS Salary,

st.course AS Course\_Underwent

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

);

1. SELECT p.name AS Programmer\_Name,

CEIL(st.ccost / (p.salary / 12)) AS Months\_To\_Recover FROM Programmer p JOIN Studies st ON p.name = st.name;

1. SELECT software.name, Max(software.dcost) from software join programmer on software.name=programmer.name where timestampdiff(year,doj,curdate())=3 group by name;
2. SELECT AVG(p.salary) AS AverageSalary

FROM Programmer p

JOIN Software s ON p.name = s.name

WHERE s.sold > 50000;

1. SELECT COUNT(s.title) AS PackagesDeveloped

FROM Software s

JOIN Studies st ON s.name = st.name

WHERE st.splace = (

SELECT splace

FROM Studies

ORDER BY ccost ASC

LIMIT 1

);

1. SELECT p.name AS Developer\_Name,

COUNT(s.title) AS PackagesDeveloped,

s.splace AS Study\_Location

FROM Programmer p

JOIN Software s ON p.name = s.name

WHERE s.dcost = (

SELECT MIN(dcost)

FROM Software

)

GROUP BY p.name, s.splace;

1. SELECT COUNT(s.title) AS PackagesDeveloped

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'

);

1. SELECT COUNT(s.title) AS PackagesDeveloped

FROM Software s

WHERE s.name IN (

SELECT p.name

FROM Programmer p

WHERE p.prof1 = 'BDPS' OR p.prof2 = 'BDPS'

ORDER BY p.dob DESC

LIMIT 1

);

1. SELECT p.name AS Programmer\_Name,

COALESCE(s.splace, 'Not Applicable') AS Institute\_Studied\_At

FROM Programmer p

LEFT JOIN Studies s ON p.name = s.name;

1. SELECT p.prof1 AS Programming\_Language,

COUNT(p.name) AS Number\_of\_Programmers,

SUM(s.sold) AS Number\_of\_Packages,

SUM(s.scost) - SUM(s.dcost) AS Profit

FROM Programmer p

JOIN Software s ON p.prof1 = s.dev\_in

GROUP BY p.prof1;

1. SELECT p.name, COUNT(s.name) AS no\_packages FROM Programmer p LEFT JOIN Software s ON p.name = s.name GROUP BY p.name;
2. SELECT p.\* FROM programmer p JOIN studies s ON p.name = s.name WHERE s.splace = 'S.S.I.L.';