SQL ASSIGNMENT

NAME : KODEESWARAN M

DATE : 2024-04-19

TABLE PROGRAMMER:

```
CREATE TABLE Programmer(
Name varchar(8) NOT NULL PRIMARY KEY,
DOB date NOT NULL,
DOJ date NOT NULL,
SEX char(1) NOT NULL,
PROF1 varchar(8) NOT NULL,
PROF2 varchar(8) NOT NULL,
SALARY numeric(4) NOT NULL,
constraint chk_sex check (sex in ('M', 'F'))
);
```

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),
DCost DECIMAL(5,0),
Sold DECIMAL(3,0),
FOREIGN KEY(Name) REFERENCES programmer(Name)
);
```

TABLE STUDIES:

```
CREATE TABLE Studies(
Name varchar(8) NOT NULL,
SPlace varchar(9) NOT NULL,
COURSE VARCHAR(5),
CCost VARCHAR(5),
FOREIGN KEY(Name) REFERENCES programmer(Name)
);
```

```
INSERT INTO programmers VALUES('somdutt', '1966-04-21', '1992-04-21', 'M', 'pascal', 'basic', 3200);
INSERT INTO programmer
VALUES ("ANAND","1966-04-12","1992-04-21","M","PASCAL","BASIC",3200),
("ALTAF","1964-07-02","1990-11-13","M","CLIPPER","COBOL",2800),
("JULIANA","1960-01-13","1990-04-21","F","COBOL","DBASE",3000),
("KAMALA","1968-10-30","1992-01-02","F","C","DBASE",2900),
("MARY","1970-06-24","1991-02-01","F","CPP","ORACLE",4500),
("NELSON","1985-09-11","1989-03-11","M","COBOL","DBASE",2500),
("PATTRICK","1965-11-10","1990-04-21","M","PASCAL"," ",2800),
("QADIR","1965-08-31","1990-04-21","M","ASSEMBLY","C",3000),
("RAMESH","1967-05-03","1991-02-26","M","PASCAL","DBASE",3200),
("REBECCA","1997-01-01","1990-12-01","F","BASIC","COBOL",2500),
("REMITHA","1970-04-19","1993-04-20","F","C","ASSEMBLY",3000),
("REVATHI","1969-12-02","1992-01-02","F","PASCAL","BASIC",3200),
("VIJAYA","1965-12-14","1992-05-02","F","FOXPRO","C",4500);

INSERT INTO software values('somdutt', 'parachutes', 'basic', 399.95,6000,43);
INSERT INTO software
VALUES ("MARY","README","CPP",100.00,1200,84),
("ANAND","PARACHUTES","BASIC",399.95,6000,43),
("ANAND","VIDEOTITLING","PASCAL",7500.00,16000,9),
("JULIANA","INVENTORY","COBOL",3000.00,3500,0),
("KAMALA","PAYROLLPRG","DBASE",9000.00,20000,7),
("MARY","FINANCIALACC","ORACLE",18000.00,85000,4),
("MARY","CODEGENERRATOP","C",4500.00,20000,23),
("PATTRICK","README","CPP",300.00,1200,84),
("QADIR","BOMBSAWAY","ASSEMBLY",750.00,5000,11),
("QADIR","VACCINES","C",1900.00,3400,21),
("RAMESH","HOTLIMGMT","DBASE",12000.00,35000,4),
("RAMESH","DEADLEE","PASCAL",599.95,4500,73),
```

("REMITHA","PCUTITLES","C",725.00,5000,51),

("REMITHA","TSRHELPPKG","ASSEMBLY",2500.00,6000,7),

("REVATHI","HOSPITALMGT","PASCAL",1100.00,75000,2),

("VIJAYA","TSREDITOR","C",900.00,700,6);

INSERT INTO studies VALUES('somdutt', 'sabhari', 'pgdca', '4500');

INSERT INTO studies VALUES("ALTAF","COIT","DCA",7200),

("JULIANA","BITS","MCA",22000),

("KAMALA","PRAGATHI","DCP",5000),

("MARY","SABHARI","PGDCA",4500),

("NELSON","PRAGATHI","DAP",6200),

("PATTRICK","PRAGATHI","DCAP",5200),

("QADIR","APPLE","HDCP",14000),

("RAMESH","SABHARI","PGDCA",4500),

("REBECCA","BRILLIANT","DCP",11000),

("REMITHA","BDPS","DCS",6000),

("VIJAYA","BDPS","DCA",48000);

SELECT *FROM programmer;

SELECT *FROM software;

SELECT *FROM studies;

QUERIES I

1) select avg(sold) as Cost from software where dev_in='pascal' ;

2) select Name, floor(datediff(curdate(),DOB)/365) as Age from Programmer ;

3) select Programmer.Name as name, floor(datediff(curdate(),DOB)/365) as Age from Programmer join studies on Programmer.Name = studies.Name where studies.course='DCS';

4) select max(sold) as MaxSales from software;

5) select name, DOB from programmer where month(DOB)=4;

6) select min(ccost) as MinCost from studies;

7) select count(name) as Count from studies where course = 'PGDCA';

8) select count(name) as Count from studies where course = 'PGDCA';

9) SELECT * FROM SOFTWARE WHERE NAME = "RAMESH";

10) SELECT COUNT(NAME) FROM STUDIES WHERE splace = "SABHARI";

11) SELECT * FROM SOFTWARE WHERE (DCOST*SOLD) > 20000;

12) SELECT CEIL(DCost / SCost) AS copires FROM Software;

13) SELECT MAX(DCOST) FROM SOFTWARE WHERE dev_in = "BASIC";

14) SELECT * FROM SOFTWARE where SCOST*SOLD >= DCOST;

15) SELECT COUNT(*) FROM SOFTWARE WHERE DEV_IN = "DBASE";

16) SELECT COUNT(*) FROM STUDIES WHERE SPLACE = "PRAGATHI";

17) SELECT COUNT(*) FROM STUDIES WHERE CCOST >= 5000 AND CCOST <= 10000;

18) SELECT AVG(CCOST) FROM STUDIES;

19) SELECT * FROM PROGRAMMER WHERE PROF1 = 'C' OR PROF2 = 'C';

20) SELECT COUNT(NAME) FROM PROGRAMMER WHERE PROF1 = 'COBOL' OR PROF1 = 'PASCAL' OR PROF2 = 'PASCAL' OR PROF2 = 'COBOL';

21) SELECT COUNT(NAME) FROM PROGRAMMER WHERE PROF1 <> 'PASCAL' AND PROF1 <> 'C' AND PROF2 <> 'C' AND PROF2 <> 'PASCAL';

22) select max(floor(datediff(curdate(),DOB)/365)) as Age from Programmer where SEX='M' ;

23) select AVG(floor(datediff(curdate(),DOB)/365)) as Age from Programmer where SEX='F' ;

24) SELECT NAME,ABS(floor(datediff(curdate(),DOJ)/365)) AS EXP FROM PROGRAMMER ORDER BY EXP DESC;

25) SELECT NAME FROM PROGRAMMER WHERE MONTH(DOB) = MONTH(curdate());

26) SELECT COUNT(*) AS FEMALE_PROGRAMMERS FROM PROGRAMMER WHERE SEX = 'F';

27) SELECT PROF1 AS LANG FROM PROGRAMMER WHERE SEX = 'M' UNION DISTINCT SELECT PROF2 FROM PROGRAMMER WHERE SEX = 'M';

28) SELECT AVG (SALARY) AS AVERAGE FROM PROGRAMMER;

29) SELECT COUNT(NAME) AS COUNT FROM PROGRAMMER WHERE SALARY >= 2000 AND SALARY <= 4000;

30) SELECT * FROM PROGRAMMER WHERE PROF1!= 'COBOL' AND PROF2!= 'COBOL' AND PROF1!= 'CLIPPER' AND PROF2!= 'CLIPPER' AND PROF1!= 'PASCAL' AND PROF2!= 'PASCAL';

31) SELECT COUNT(NAME) FROM PROGRAMMER WHERE (PROF1='C' OR PROF2='C') AND (floor(datediff(curdate(),DOB)/365))>24;

32) SELECT NAME, DOB FROM PROGRAMMER WHERE DATEDIFF((DATE_ADD(DOB, INTERVAL CEIL(datediff(curdate(),DOB)/365.25) YEAR)),CURDATE()) BETWEEN 0 AND 6;

SELECT DATEDIFF((DATE_ADD(DOB, INTERVAL CEIL(datediff(curdate(),DOB)/365.25) YEAR)),CURDATE()) FROM PROGRAMMER;

33) SELECT NAME FROM PROGRAMMER WHERE ABS(floor(datediff(curdate(),DOJ)/365)) < 1;

34) SELECT NAME FROM PROGRAMMER WHERE ABS(floor(datediff(curdate(),DOJ)/365)) = 2;

35) SELECT DCOST-(SCOST*SOLD) AS AMOUNT FROM SOFTWARE WHERE DCOST-(SCOST*SOLD) >0;

36) SELECT * FROM SOFTWARE WHERE SOLD=0;

37) SELECT SUM(DCOST) AS TOTAL FROM SOFTWARE WHERE NAME = 'MARY';

38) SELECT DISTINCT SPLACE FROM STUDIES;

39) SELECT COUNT(DISTINCT COURSE) FROM STUDIES;

40) SELECT NAME FROM PROGRAMMER WHERE NAME LIKE '%A%A%';

41) SELECT NAME FROM PROGRAMMER WHERE CHAR_LENGTH(NAME) <=5;

42) SELECT NAME FROM PROGRAMMER WHERE SEX='F' AND (PROF1='COBOL' OR PROF2='COBOL') AND ABS(floor(datediff(curdate(),DOJ)/365)) > 2;

43) SELECT MIN(CHAR_LENGTH(NAME)) FROM PROGRAMMER;

44) SELECT AVG(DCOST) AS AVG FROM SOFTWARE WHERE DEV_IN='COBOL';

45) select name, sex,

```
CONCAT(
   RIGHT(CONCAT('0', DAY(dob)), 2), '/',
   RIGHT(CONCAT('0', MONTH(dob)), 2), '/',
   RIGHT(YEAR(dob), 2)
) as DOB,
CONCAT(
   RIGHT(CONCAT('0', DAY(doj)), 2), '/',
   RIGHT(CONCAT('0', MONTH(doj)), 2), '/',
   RIGHT(YEAR(doj), 2)
) as DOJ FROM Programmer;
```

46) SELECT NAME FROM PROGRAMMER WHERE DOB=LAST_DAY(DOB);

47) SELECT NAME, SALARY , SEX FROM PROGRAMMER WHERE SEX='M' AND (PROF1 <> 'COBOL' AND PROF2 <> 'COBOL');

48) SELECT TITLE ,SCOST,DCOST,ABS(SCOST-DCOST) AS COST FROM SOFTWARE ORDER BY COST DESC;

49) SELECT NAME,DOB,DOJ FROM PROGRAMMER WHERE MONTH(DOB)=MONTH(DOJ);

50) SELECT TITLE FROM SOFTWARE WHERE LOCATE(' ', TITLE) >0;

QUERIES II

1) SELECT DEV_IN , COUNT(*) AS PACKAGES FROM SOFTWARE GROUP BY DEV_IN;

2) SELECT NAME,COUNT(*) AS PACKAGES FROM SOFTWARE GROUP BY NAME;

3) SELECT SEX, COUNT(*) AS COUNT FROM PROGRAMMER GROUP BY SEX;

4) SELECT DEV_IN, MAX(SCOST*SOLD) FROM SOFTWARE GROUP BY DEV_IN;

5) SELECT YEAR(DOB), COUNT(*) AS TOTAL FROM PROGRAMMER GROUP BY YEAR(DOB) ORDER BY YEAR(DOB);

6) SELECT YEAR(DOJ), COUNT(*) AS TOTAL FROM PROGRAMMER GROUP BY YEAR(DOJ);

7) SELECT MONTH(DOB), COUNT(*) AS TOTAL FROM PROGRAMMER GROUP BY MONTH(DOB) ORDER BY MONTH(DOB);

8) SELECT MONTH(DOJ), COUNT(*) AS TOTAL FROM PROGRAMMER GROUP BY MONTH(DOJ) ORDER BY MONTH(DOJ);

9) SELECT PROF1, COUNT(PROF1) AS COUNT FROM PROGRAMMER GROUP BY PROF1;

10) SELECT PROF2, COUNT(PROF2) AS COUNT FROM PROGRAMMER GROUP BY PROF2 ORDER BY PROF2;

11) SELECT SALARY, COUNT(NAME) AS TOTAL FROM PROGRAMMER GROUP BY SALARY ORDER BY SALARY;

12) SELECT SPLACE, COUNT(NAME) AS TOTAL FROM STUDIES GROUP BY SPLACE ORDER BY SPLACE;

13) SELECT COURSE, COUNT(NAME) AS TOTAL FROM STUDIES GROUP BY COURSE ORDER BY COURSE;

14) SELECT DEV_IN, SUM(DCOST) FROM SOFTWARE GROUP BY DEV_IN;

15) SELECT DEV_IN, SUM(SCOST) FROM SOFTWARE GROUP BY DEV_IN;

16) SELECT NAME, SUM(SCOST) FROM SOFTWARE GROUP BY NAME;

17) SELECT NAME, SUM(SCOST*SOLD) FROM SOFTWARE GROUP BY NAME;

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 AS programmer_name, MAX(SCost) AS costliest_package, MIN(SCost) AS cheapest_package FROM SOFTWARE s 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 SEX,NAME FROM PROGRAMMER WHERE SEX='M' UNION SELECT SEX,NAME FROM PROGRAMMER WHERE SEX='F';

25) SELECT NAME, GROUP_CONCAT(TITLE) FROM SOFTWARE GROUP BY NAME;

26) SELECT DEV_IN, COUNT(TITLE) FROM SOFTWARE GROUP BY DEV_IN;

27) SELECT DEV_IN, COUNT(DCOST) FROM SOFTWARE WHERE DCOST<1000 GROUP BY DEV_IN ;

28) SELECT DEV_IN, AVG(ABS(SCOST-DCOST))AS AVG FROM SOFTWARE GROUP BY DEV_IN;

29) SELECT NAME, SUM(SCOST), SUM(DCOST), SUM(DCOST-(SCOST*SOLD)) AS RECOVERY FROM SOFTWARE GROUP BY NAME HAVING SUM(DCOST)>SUM(SCOST*SOLD);

30) SELECT MAX(SALARY), MIN(SALARY), AVG(SALARY) FROM PROGRAMMER WHERE SALARY > 2000;

QUERY III

1) SELECT NAME,SALARY FROM PROGRAMMER WHERE PROF1='C' OR PROF2='C' ORDER BY SALARY DESC LIMIT 1;

2) SELECT NAME,SALARY FROM PROGRAMMER WHERE (PROF1='COBOL' OR PROF2='COBOL') AND SEX='F' ORDER BY SALARY DESC LIMIT 1;

3) SELECT PROF1, MAX(SALARY), GROUP_CONCAT(NAME) FROM PROGRAMMER GROUP BY PROF1 ORDER BY PROF1;

4) SELECT ABS(floor(datediff(curdate(),DOJ)/365)) AS EXP,GROUP_CONCAT(NAME) AS NAME FROM PROGRAMMER GROUP BY EXP ORDER BY EXP LIMIT 1 ;

5) SELECT ABS(floor(datediff(curdate(),DOJ)/365)) AS EXP,GROUP_CONCAT(NAME) AS NAME FROM PROGRAMMER GROUP BY EXP ORDER BY EXP DESC LIMIT 1 ;

6) SELECT lang FROM (

SELECT PROF1 AS lang

FROM PROGRAMMER

UNION ALL

SELECT PROF2 AS lang

FROM PROGRAMMER

) languages

GROUP BY lang

HAVING COUNT(*) = 1;

7) SELECT NAME,floor(datediff(curdate(),DOB)/365) AS AGE FROM PROGRAMMER WHERE PROF1='DBASE' OR PROF2='DBASE' ORDER BY AGE LIMIT 1 ;

8) SELECT SPLACE, COUNT(*) AS TOTAL FROM STUDIES GROUP BY SPLACE ORDER BY TOTAL DESC LIMIT 1;

9) WITH TOTAL AS (

SELECT SPlace

FROM STUDIES

GROUP BY SPlace

ORDER BY COUNT(*) DESC

LIMIT 1

)

SELECT DISTINCT NAME

FROM STUDIES

WHERE SPlace = (SELECT SPlace FROM TOTAL);

10) SELECT NAME , SEX FROM PROGRAMMER WHERE SEX='F' AND SALARY>3000 AND ((PROF1 <> 'C' AND PROF1 <> 'C++' AND PROF1<>'ORACLE' AND PROF1<>'DBASE') AND (PROF2 <> 'C' AND PROF2 <> 'C++' AND PROF2<>'ORACLE' AND PROF2<>'DBASE'));

11) SELECT SPLACE,CCOST FROM STUDIES WHERE CCOST=(SELECT MAX(CAST(CCOST AS DECIMAL(10,2))) FROM STUDIES);

12) SELECT COURSE, COUNT(*) AS TOTAL FROM STUDIES GROUP BY COURSE ORDER BY TOTAL DESC LIMIT 1;

13) SELECT COURSE, SPLACE, CCOST FROM STUDIES WHERE CCOST < (SELECT AVG(CCOST) FROM STUDIES);

14) SELECT SPLACE, MAX(CCOST) FROM STUDIES GROUP BY SPLACE ORDER BY MAX(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 AvgNumStudents

);

16) SELECT SPlace AS Institute

FROM STUDIES

WHERE COURSE IN (

  SELECT COURSE

  FROM STUDIES

  GROUP BY COURSE

  HAVING COUNT(*) < (

    SELECT AVG(Num_Students)

    FROM (

      SELECT COUNT(*) AS Num_Students

      FROM STUDIES

      GROUP BY COURSE

) AS AvgNumStudents

)

);

17) SELECT COURSE FROM STUDIES WHERE ABS(CCOST - (SELECT AVG(CCOST) FROM STUDIES)) <= 1000;

18) SELECT TITLE, DCOST FROM SOFTWARE ORDER BY DCOST DESC LIMIT 1;

19) SELECT TITLE, SCOST FROM SOFTWARE ORDER BY SCOST LIMIT 1;

20) SELECT NAME,SOLD FROM SOFTWARE WHERE SOLD = (SELECT MIN(SOLD) FROM SOFTWARE);

21) SELECT DEV_IN,SCOST FROM SOFTWARE WHERE SCOST = (SELECT MAX(SCOST) FROM SOFTWARE);

22) SELECT SOLD,TITLE FROM SOFTWARE WHERE TITLE = (SELECT TITLE FROM SOFTWARE WHERE (DCOST-SCOST)=(SELECT MIN(DCOST-SCOST) FROM SOFTWARE));

23) SELECT TITLE FROM SOFTWARE WHERE DCOST = (SELECT MAX(DCOST)FROM SOFTWARE WHERE DEV_IN LIKE 'PASCAL');

24) SELECT DEV_IN FROM SOFTWARE GROUP BY DEV_IN HAVING DEV_IN = (SELECT MAX(DEV_IN) FROM SOFTWARE);

25) SELECT NAME FROM SOFTWARE GROUP BY NAME HAVING NAME = (SELECT MAX(NAME) FROM SOFTWARE);

26) SELECT NAME FROM SOFTWARE WHERE SCOST = (SELECT MAX(SCOST) FROM SOFTWARE);

27) SELECT TITLE FROM SOFTWARE WHERE SOLD < (SELECT AVG(SOLD)FROM SOFTWARE);

28) SELECT NAME, SALARY FROM PROGRAMMER WHERE SEX = 'F' AND SALARY > (SELECT MAX(SALARY) FROM PROGRAMMER WHERE SEX = 'M');

29) SELECT PROF1, COUNT(*) FROM PROGRAMMER GROUP BY PROF1 ORDER BY COUNT(*) DESC LIMIT 1;

30) SELECT DISTINCT NAME FROM SOFTWARE WHERE SOLD*SCOST > 2*DCOST;

31) SELECT NAME,TITLE FROM SOFTWARE WHERE DCOST IN (SELECT MIN(DCOST) FROM SOFTWARE GROUP BY DEV_IN);

32) select name,floor(DATEDIFF(CURDATE(), DOB)/365) as age from programmer where year(dob)='1965' and sex='M' limit 1;

33) SELECT NAME, DEV_IN FROM SOFTWARE WHERE SOLD IN (SELECT MAX(SOLD) FROM SOFTWARE GROUP BY NAME) UNION SELECT NAME, DEV_IN FROM SOFTWARE WHERE SOLD IN (SELECT MIN(SOLD) FROM SOFTWARE GROUP BY NAME);

34) select name, 2024-year(dob) as age from programmer where year(doj)='1992' and sex='F' order by age desc limit 1;

35) select year(dob) ,count(year(dob)) as count from programmer group by year(dob) order by count(*) desc limit 1;

36) select month(doj) as month,count(month(doj)) as count from programmer group by month(doj) order by count(*) desc limit 1;

37) select prof1,prof2,count(name) as count from programmer group by prof1,prof2 order by count desc limit 2;

38) SELECT NAME FROM PROGRAMMER WHERE SEX LIKE 'M' AND SALARY < (SELECT(AVG(SALARY)) FROM PROGRAMMER WHERE SEX LIKE 'F');

QUERY IV

1) SELECT * FROM Programmer WHERE Salary IN (

SELECT Salary FROM Programmer GROUP BY Salary HAVING COUNT(*) > 1);

2) SELECT * FROM Software WHERE Dev_In IN (

SELECT Name FROM Programmer WHERE Sex = 'M' AND Salary > 3000);

3) SELECT * FROM Software WHERE Dev_In IN

(SELECT Name FROM Programmer WHERE Sex = 'F') AND

Title IN (SELECT Name FROM Programmer WHERE Prof1 = 'PASCAL' OR Prof2 = 'PASCAL');

4) SELECT * FROM Programmer WHERE DOJ < '1990-01-01';

5) SELECT * FROM Software

WHERE Dev_In IN (SELECT Name FROM Programmer WHERE Sex = 'F'

AND (Prof1 = 'C' OR Prof2 = 'C'))

AND Dev_In IN (SELECT Name FROM Studies WHERE SPlace = 'PRAGATHI');

6) SELECT s.Dev_In, p.SPlace, COUNT(s.Title) AS NumPackages, SUM(s.Sold) AS NumCopiesSold, SUM(s.DCost * s.Sold) AS SalesValue

FROM Software s

INNER JOIN Studies p ON s.Name = p.Name

GROUP BY s.Dev_In, p.SPlace;

7) SELECT *FROM Software

WHERE Title IN (

SELECT Title FROM Software WHERE Dev_In IN (

SELECT Name FROM Programmer WHERE Sex = 'M')

GROUP BY Title ORDER BY COUNT(*) DESC LIMIT);

8) SELECT * FROM Software

WHERE Dev_In IN (

SELECT Name FROM Programmer WHERE Sex = 'M' AND DOB < '1965-01-01')

OR Dev_In IN (SELECT Name FROM Programmer WHERE Sex = 'F' AND DOB > '1975-01-01'

);

9) SELECT * FROM Software

WHERE Title IN (SELECT Title FROM Software WHERE Dev_In IN

(

SELECT Name FROM Programmer WHERE Title NOT IN (Prof1, Prof2)

)

);

10) SELECT * FROM Software

WHERE Title IN (SELECT Title FROM Software WHERE Dev_In IN

(

SELECT Name FROM Programmer WHERE Title NOT IN (Prof1, Prof2)

)

);

11) SELECT * FROM Software

WHERE Dev_In IN (SELECT Name FROM ProgrammermWHERE Sex = 'M')

AND Dev_In IN (

SELECT Name FROM Studies WHERE SPlace = 'SABHARI'

);

12) SELECT Name FROM Programmer

WHERE Name NOT IN (SELECT DISTINCT Name FROM Software

);

```
13) SELECT SUM(DCost) AS TotalCost
FROM Software WHERE Dev_In = 'APPLE';

14) SELECT Name FROM Programmer
GROUP BY DOJ HAVING COUNT(*) > 1;

15) SELECT Name FROM Programmer
GROUP BY PROF2 HAVING COUNT(*) > 1;

16) SELECT p.SPlace, SUM(s.DCost * s.Sold) AS TotalSalesValue
FROM Software s
INNER JOIN Studies p ON s.Name = p.Name
GROUP BY p.SPlace;

17) SELECT p.SPlace FROM Software s
INNER JOIN Studies p ON s.Name = p.Name
WHERE s.DCost = (
   SELECT MAX(DCost) FROM Software
);

18) SELECT DISTINCT Prof1 AS Language FROM Programmer
WHERE Prof1 NOT IN (  SELECT DISTINCT TitleFROM Software )
UNION
SELECT DISTINCT Prof2 AS Language FROM Programmer
WHERE Prof2 NOT IN (
   SELECT DISTINCT Title FROM Software
);

19) SELECT p.Name, p.Salary, s.COURSE
FROM Programmer p INNER JOIN Software s
 ON p.Name = s.Name WHERE s.Sold = (
```

```
   SELECT MAX(Sold) FROM Software
);
```

20)
```
SELECT p.Name, DATEDIFF(MONTH, p.DOJ, GETDATE()) AS MonthsToRecover
FROM Programmer p
INNER JOIN Studies s ON p.Name = s.Name;
```

21)
```
SELECT Title AS CostliestPackage FROM Software
WHERE DCost = (
   SELECT MAX(DCost) FROM Software
   WHERE Name IN (
      SELECT Name FROM Programmer
      WHERE DATEDIFF(YEAR, DOJ, GETDATE()) < 3  ) );
```

22)
```
SELECT AVG(Salary) AS AverageSalary FROM Programmer
WHERE Name IN (
   SELECT Name FROM Software WHERE DCost * Sold > 50000
);
```

23)
```
SELECT COUNT(Title) AS NumPackages FROM Software
WHERE Name IN (
   SELECT Name FROM Studies
   WHERE CCost = ( SELECT MIN(COST) FROM Studies )
);
```

24)
```
SELECT p.Name, COUNT(*) AS NumPackages, s.SPlace
FROM Programmer p
INNER JOIN Software s ON p.Name = s.Name
WHERE DCost = ( SELECT MIN(DCost) FROM Software )
GROUP BY p.Name, s.SPlace;
```

25)
```
SELECT COUNT(*) AS NumPackages FROM Software
```

```
WHERE Name IN (
   SELECT Name FROM Programmer
   WHERE Sex = 'F' AND Salary > (
      SELECT MAX(Salary) FROM Programmer  WHERE Sex = 'M'
   )
);
```

26)
```
SELECT COUNT(*) AS NumPackagesFROM Software
WHERE Name IN (
   SELECT Name FROM Programmer
   WHERE DOJ = (
      SELECT MAX(DOJ) FROM Programmer
      WHERE Name IN (
         SELECT Name FROM Studies
         WHERE SPlace = 'BDPS'
      )
   )
);
```

27)
```
SELECT p.Name, s.SPlace FROM Programmer p
LEFT JOIN Studies s ON p.Name = s.Name;
```

28)
```
SELECT Prof1 AS Proficiency, COUNT(Name) AS NumProgrammers, (
   SELECT COUNT(Title) FROM Software
   WHERE Dev_In IN (SELECT Name FROM Programmer WHERE Prof1 = p.Prof1  )
) AS NumPackages
FROM Programmer p GROUP BY Prof1;
```

29)
```
SELECT p.Name, COUNT(s.Title) AS NumPackages FROM Programmer p
LEFT JOIN Software s ON p.Name = s.Name
GROUP BY p.Name;
```

30) SELECT p.* FROM Programmer p

INNER JOIN Studies s ON p.Name = s.Name

WHERE s.SPlace = 'S.S.I.L';