SQL Queries Assignment

Problem Statement:

Consider yourself to be Sam and you have been given the below tasks to complete using the Table – STUDIES, SOFTWARE and PROGRAMMER

Tasks To Be Performed:

```
-- 1. Find out the selling cost average for packages developed in Pascal

select AVG(SCOST) as "Avg Selling Cost"
from Software
where DEVELOPIN = 'PASCAL'

-- 2. Display the names and ages of all programmers.

select PName ,
DATEDIFF(YY,DOB,GETDATE()) as "Age of Employeee"
from Programmer


-- 3. Display the names of those who have done the DAP Course

select Pname,Course
from Studies
where COURSE = 'DAP'

-- 4. Display the names and date of birth of all programmers born in January.

Select PNAME, DOB
From Programmer
```

```sql
Where Month(DOB) = 01

-- 5. What is the highest number of copies sold  by a packages ?

Select Max(SOLD) As "Highest Selling Copies"
from Software


-- 6. Display lowest course fee.

Select
Min(Course_Fee) as "Lowest Course Fee"
from Studies

-- 7. How many programmers have done the PGDCA Course?

Select COUNT(PNAME) as "PGDCA Programmers "From Studies
where COURSE = ('PGDCA')


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


Select SUM(SCOST * SOLD) AS 'Total Revenue' from Software
where DEVELOPIN = 'C'

-- 9. Display the details of the software developed by Ramesh.

Select * from Software
where PNAME = 'RAMESH'
```

-- 10. How many programmers studied at Sabhari?

```
Select Count(PNAME) as "Count of Sabhari Students " From Studies
where INSTITUTE = 'SABHARI'
```

-- 11. Display details of packages whose sales crossed the 2000 mark.

```
Select * from Software
where (SCOST) > 2000;
```

-- 12. Display the details of packages for which development costs have been recovered.

```
Select* from Software
Where (SCOST*SOLD) > DCOST;
```

-- 13. What is the cost of the costliest software development in Basic?

```
Select MAX(DCOST) AS "COSTLIEST DEVELOPEMENT" from Software
WHERE DEVELOPIN = 'BASIC'
```

-- 14. How many packages have been developed in dBase?

```
SELECT COUNT(TITLE) 'SOFTWARE IN DBASE' FROM Software
WHERE DEVELOPIN = 'DBASE'
```

-- 15. How many programmers studied in Pragathi?

```
SELECT COUNT(PNAME) 'STUDENTS FROM Pragathi' FROM Studies
WHERE INSTITUTE = 'Pragathi'
```

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

SELECT * FROM Studies
WHERE COURSE_FEE BETWEEN 5000 AND 10000

-- 17. What is the average course fee?

SELECT AVG(COURSE_FEE) AS "AVG COURSE FEE" FROM Studies

-- 18. Display the details of the programmers knowing C.

SELECT * FROM Programmer
WHERE 'C'= PROF1 OR 'C'=PROF2

-- 19. How many programmers know either COBOL or Pascal?

SELECT COUNT(PNAME) "COBAL & PASKAL PROGRAMERS" FROM Programmer
WHERE PROF1 IN ('COBOL','PASCAL') OR PROF2 IN ('COBOL','PASCAL')

-- 20. How many programmers don’t know Pascal and C?
SELECT COUNT(PNAME) "NO KNOWLEDGE OF C & PASKAL" FROM Programmer
WHERE PROF1 NOT IN ('C','PASCAL') AND PROF2 NOT IN ('C','PASCAL')

-- 21. How old is the oldest male programmer?
SELECT MIN(DOB)
FROM Programmer
WHERE GENDER = 'M'

-- 22. What is the average age of female programmers?
select
AVG(DATEDIFF(YY,DOB,GETDATE())) as "Age of Employeee"
```

```
from Programmer

WHERE GENDER = 'F'
```

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

```
Select PName ,

DATEDIFF(YY,DOJ,GETDATE()) as "Experience of Employeee"

from Programmer

order by PNAME desc
```

-- 24. Who are the programmers who celebrate their birthdays during the current month?

```
SELECT * FROM Programmer WHERE MONTH(DOB) = MONTH(getdate())
```

-- 25. How many female programmers are there?

```
select count(PNAME) as 'Female Programmers'from Programmer

where gender = 'F'
```

-- 26. What are the languages studied by male programmers?

```
select PNAME, Prof1, Prof2 from Programmer

where GENDER = 'M';
```

-- 27. What is the average salary?

```
SELECT AVG(SALARY) AS "AVG SALARY" FROM Programmer
```

-- 28. How many people draw a salary between 2000 to 4000?

```sql
SELECT COUNT(PNAME) AS "2K TO 4K SALARY EMP" FROM Programmer

WHERE SALARY BETWEEN 2000 AND 4000


-- 29. 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')


-- 30. Display the cost of packages developed by each programmer


SELECT PNAME, SUM(DCOST) AS "DEVELOPING COST"

FROM SOFTWARE

GROUP BY PNAME


-- 31. Display the sales value of the packages developed by each programmer.


SELECT PNAME , SUM(SCOST) AS "SALES DEVELOPED"

FROM Software

GROUP BY PNAME


-- 32. Display the number of packages sold by each programmer.


SELECT PNAME , SUM(SOLD) AS "PACKAGE SOLD"

FROM Software

GROUP BY PNAME


-- 33. Display the sales cost of the packages developed by each programmer language wise.


SELECT PNAME,

DEVELOPIN,

SUM(SCOST) AS "SALES COST BY LANGAUGE"
```

```
FROM Software
GROUP BY DEVELOPIN,PNAME
```

-- 34. Display each language name with the average development cost,average selling cost and average price per copy.

```
SELECT
AVG(DCOST),
AVG(SCOST),
AVG(SCOST)
FROM Software
```

-- 35. Display each programmer’s name and the costliest and cheapest packages developed by him or her.

```
SELECT PNAME,
MAX(DCOST) AS "COSLIEST DEVELOPEMENT",
MIN(DCOST) AS "CHEAPEST DEVELOPEMENT"
FROM Software
GROUP BY PNAME
```

-- 36. Display each institute’s name with the number of courses and the average cost per course.

```
SELECT INSTITUTE,
COUNT(COURSE) AS "NUMBER OF COURSES ",
AVG(COURSE_FEE) AS "AVERAGE COURSE FEE"
FROM Studies
GROUP BY INSTITUTE
```

-- 37. Display each institute’s name with the number of students.

```
SELECT
```

```sql
INSTITUTE,
COUNT(PNAME) AS "NUMBER OF STUDENT"
FROM Studies
GROUP BY INSTITUTE

-- 38. Display names of male and female programmers along with their gender.

SELECT
PNAME,
GENDER
FROM Programmer

-- 39. Display the Name of Programmers and Their Packages.

SELECT
PNAME,
SALARY
FROM Programmer

-- 40. Display the number of packages in each language except C and C++.

SELECT
DEVELOPIN AS 'LANGAUGE',
COUNT(TITLE) AS 'NUMBER OF PACKAGES'
FROM Software
WHERE DEVELOPIN != 'C' AND DEVELOPIN != 'C++'
GROUP BY DEVELOPIN

-- 41. Display the number of packages in each language for which development cost is less than 1000.

SELECT
```

```sql
DEVELOPIN AS "LANGAUGE",
COUNT(TITLE) AS 'PACKAGE'
FROM Software
WHERE DCOST >= 1000
GROUP BY DEVELOPIN

-- 42. Display the average difference between SCOST and DCOST for each package

SELECT
TITLE,
AVG(DCOST - SCOST) AS "AVG(SCOST- DCOST)"
FROM Software
GROUP BY TITLE

-- 43. Display the total SCOST, DCOST and the amount to be recovered for each programmer whose cost has not yet been recovered.

SELECT
PNAME,
SUM(SCOST) 'TOTAL SELLING COST',
SUM(DCOST)'TOTAL DEVELOPING COST',
SUM(DCOST-(SCOST * SOLD)) 'COST NEED TO RECOVER'
FROM Software
GROUP BY PNAME
HAVING SUM(DCOST) > SUM(SOLD * SCOST)

-- 44. Display the highest, lowest and average salaries for those earning more than 2000

SELECT
MAX(SALARY) AS " MAX SALARY",
MIN(SALARY) AS "MIN SALARY",
AVG(SALARY) AS "AVG SALARY"
FROM Programmer
WHERE SALARY >= 2000

-- 45. Who is the Highest Paid C Programmers?

SELECT *
FROM Programmer
WHERE
SALARY =(SELECT MAX (SALARY) FROM Programmer
        WHERE PROF1 = 'C' OR PROF2 = 'C')

-- 46. Who is the highest paid female COBOL programmer?

SELECT *
FROM Programmer
WHERE
SALARY = (SELECT MAX(SALARY) FROM Programmer
         WHERE  (PROF1 = 'COBOL' OR PROF2 = 'COBOL')) AND (GENDER = 'F')

-- 47. Display the names of the highest paid programmers for each language.

 WITH CTC AS (
 SELECT PNAME, SALARY, PROF1 AS PROF FROM programmer
 UNION
 SELECT PNAME, SALARY, PROF2 FROM programmer
 )
 SELECT p1.PNAME, p1.PROF, p1.SALARY
 FROM CTC as p1
 LEFT JOIN CTC as
 p2
```

```sql
  ON p1.PROF = p2.PROF AND p1.SALARY < p2.SALARY
  WHERE p2.PNAME IS NULL;

-- 48. Who is the least experienced programmer?

--SELECT min(
--case
--when dateadd(year, datediff(YEAR, doj, getdate()), doj)>getdate()
--then datediff(YEAR, doj, getdate()) - 1
--else
--datediff(YEAR, doj, getdate()) end) As experience
--from programmer;

Select
MIN(DATEDIFF(YY,DOJ,GETDATE())) as "Minimun Experience of Employeee"
from Programmer

-- 49. Who is the most experienced male programmer knowing PASCAL?

Select
MAX(DATEDIFF(YY,DOJ,GETDATE())) as "Maximum Experience with pascal"
from Programmer
WHERE PROF1 = 'PASCAL' OR PROF2 = 'PASCAL'

--SELECT max(
--case
--when dateadd(year, datediff(YEAR, doj, getdate()), doj)>getdate()
--then datediff(YEAR, doj, getdate()) - 1
--else
--datediff(YEAR, doj, getdate()) end) As experience
--from programmer
```

```
--where prof1='pascal' or prof2='pascal';

-- 50. Which language is known by only one programmer?

  SELECT PROF1 FROM Programmer
  GROUP BY PROF1
  HAVING PROF1 NOT IN
  (SELECT PROF2 FROM Programmer)
  AND COUNT (PROF1) = 1
  UNION
  SELECT PROF2 FROM Programmer
  GROUP BY PROF2
  HAVING PROF2 NOT IN
  (SELECT PROF1 FROM Programmer)
  AND COUNT(PROF2)=1;

-- 51. Who is the above programmer referred in 50?

  CREATE TABLE PSLang(PROF VARCHAR(20))

Select * from pslang

INSERT INTO PSLang
SELECT PROF1 FROM programmer
GROUP BY PROF1 HAVING
PROF1 NOT IN (SELECT PROF2 FROM programmer)
AND COUNT(PROF1)=1
UNION
SELECT PROF2 FROM programmer
GROUP BY PROF2 HAVING
PROF2 NOT IN (SELECT PROF1 FROM programmer)
```

```
AND COUNT(PROF2)=1

SELECT PNAME, PROF FROM programmer
INNER JOIN PSLang ON
PROF=PROF1 OR PROF=PROF2


-- 52. Who is the youngest programmer knowing dBase?


SELECT pname, prof1, prof2,
case
when dateadd(year, datediff(YEAR, dob, getdate()), dob)>getdate()
then datediff(YEAR, dob, getdate()) - 1
else
datediff(YEAR, dob, getdate()) end As Age
from programmer where dob = (SELECT max(dob) from programmer where prof1='dbase' or prof2='dbase');

-- 53. Which female programmer earning more than 3000 does not know C,C++, Oracle or dBase?

SELECT *
FROM Programmer
WHERE (PROF1 NOT IN ('C','C++','ORACLE','DBASE')
  AND  PROF2 NOT IN ('C','C++','ORACLE','DBASE'))
  AND( SALARY >=3000 AND  GENDER = 'F')

-- 54. Which Institute has most number of Students?


CREATE TABLE InstStudNo (InstituteName VARCHAR(20), StdNo INT)
```

```sql
INSERT INTO InstStudNo
SELECT INSTITUTE,COUNT(PNAME) FROM studies GROUP BY INSTITUTE

SELECT InstituteName,StdNo AS COUNT_OF_STUDENTS FROM InstStudNo
WHERE StdNo = (SELECT MAX(StdNo) FROM InstStudNo)

select * from InstStudNo

-- 55.What is the costliest course?

SELECT
COURSE,
COURSE_FEE
FROM Studies
WHERE COURSE_FEE= (SELECT MAX(COURSE_FEE) FROM Studies)


-- 56. Which course has been done by the most of the Students?

CREATE TABLE MOSTCDONE (CourNam VARCHAR(20), StdNo INT)

INSERT INTO MOSTCDONE
SELECT COURSE,COUNT(PNAME) FROM studies GROUP BY COURSE

select * from MOSTCDONE

SELECT CourNam,StdNo AS COUNT_OF_STUDENTS FROM MOSTCDONE WHERE StdNo =
(SELECT MAX(StdNo) FROM MOSTCDONE)

-- 57. Which Institute conducts costliest course.
```

```
SELECT
INSTITUTE,
COURSE_FEE
FROM Studies
WHERE COURSE_FEE = (SELECT MAX(COURSE_FEE) FROM Studies)

-- 58. Display the name of the institute and the course which has below average course fee.

SELECT
INSTITUTE,
COURSE_FEE
FROM Studies
WHERE COURSE_FEE < (SELECT AVG(COURSE_FEE) FROM Studies)

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

  SELECT
  COURSE
  FROM Studies
  WHERE
  COURSE_FEE < (SELECT AVG(COURSE_FEE) + 1000 FROM Studies)
  AND
  COURSE_FEE < (SELECT AVG(COURSE_FEE) - 1000 FROM Studies)

-- 60. Which package has the highest development cosT /

SELECT *
FROM Software
WHERE DCOST = (SELECT MAX(DCOST) FROM Software)
```

```sql
-- 61. Which course has below AVG number of Students?

  CREATE TABLE AVGSTU(BAVGCRS VARCHAR(20),BAVG INT)

  INSERT INTO AVGSTU
  SELECT COURSE,COUNT(PNAME) FROM STUDIES GROUP BY COURSE
  SELECT BAVGCRS,BAVG FROM AVGSTU WHERE BAVG <=(SELECT AVG (BAVG) FROM AVGSTU)

-- 62. Which package has the lowest selling cost ?

SELECT
TITLE,
SCOST
FROM Software
WHERE SCOST = (SELECT MIN(SCOST) FROM Software)

-- 63. Who developed the package that has sold the least number of copies?

  SELECT
  PNAME,
  TITLE,
  SOLD
  FROM Software
  WHERE SOLD = (SELECT MIN(SOLD)FROM SOFTWARE)

-- 64. Which language has been used to develop the package which has the highest sales amount?

SELECT
PNAME,
TITLE,
DEVELOPIN

FROM Software

WHERE SCOST = (SELECT MAX(SCOST) FROM Software)


-- 65. How many copies of package that has the least difference between development and selling cost where sold.


SELECT

SOLD ,

TITLE

FROM Software

WHERE TITLE = (SELECT TITLE FROM Software WHERE (DCOST- SCOST) =(SELECT MIN(DCOST-SCOST)FROM SOFTWARE))


-- 66. Which is the costliest package developed in Pascal?


SELECT

TITLE ,

DCOST

FROM Software

WHERE DCOST= (SELECT MAX(DCOST) FROM Software WHERE DEVELOPIN = 'PASCAL')


-- 67. Which language was used to develop the most number of Packages.


SELECT

DEVELOPIN

FROM Software

GROUP BY DEVELOPIN

HAVING DEVELOPIN = (SELECT MAX(DEVELOPIN) FROM Software )


-- 68. Which programmer has developed the highest number of packages?
```

```sql
   SELECT
PNAME
FROM Software
GROUP BY PNAME
HAVING PNAME = (SELECT MAX(PNAME) FROM Software )

-- 69. Who is the author of the costliest package?

  SELECT
  PNAME,
  TITLE,
  DCOST
  FROM Software
  WHERE DCOST = (SELECT MAX(DCOST) FROM Software)

-- 70. Display the names of the packages, which have sold less than the AVG number of copies.

  SELECT
  TITLE
  FROM Software
  WHERE SOLD < (SELECT AVG(SOLD) FROM Software)

-- 71. Who are the authors of the packages which have recovered more than double the development
cost?

  SELECT
  DISTINCT(PNAME)
  FROM Software
  WHERE SCOST*SOLD > (2*DCOST)

-- 72. Display the programmer names and the cheapest packages developed by them in each language.
```

```
  SELECT

  PNAME,

  TITLE,

  DEVELOPIN,

  DCOST

  FROM Software

  WHERE DCOST IN (SELECT MIN(DCOST) FROM Software GROUP BY DEVELOPIN)
```

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

```
  SELECT

  PNAME,

  DEVELOPIN

  FROM Software

  WHERE SOLD IN (SELECT MAX (SOLD) FROM Software GROUP BY PNAME)

  UNION

  SELECT

  PNAME,

  DEVELOPIN

  FROM Software

  WHERE SOLD IN (SELECT MIN (SOLD) FROM Software GROUP BY PNAME)
```

-- 74. Who is the youngest male programmer born in 1965?

```
Select

PNAME,

DOB,

DATEDIFF(YY,DOB,GETDATE()) as "Employeee age"

from Programmer
```

```
WHERE DOB = (SELECT MIN(DOB) FROM Programmer WHERE YEAR(DOB) = '1965' AND GENDER = 'M')

--SELECT pname,
--case
--when dateadd(year, datediff(YEAR, dob, getdate()), dob)>getdate()
--then datediff(YEAR, dob, getdate()) - 1
--else
--datediff(YEAR, dob, getdate()) end As Age
--from programmer where dob = (SELECT MIN(dob) from programmer where YEAR(dob)='1965' and GENDER='M');

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

 Select
PNAME,
DOJ,
DATEDIFF(YY,DOB,GETDATE()) as "Oldest Employeee age"
from Programmer
WHERE DOB = (SELECT MIN(DOB) FROM Programmer WHERE YEAR(DOJ) = '1992' AND GENDER = 'F')

--SELECT pname,
--case
--when dateadd(year, datediff(YEAR, dob, getdate()), dob)>getdate()
--then datediff(YEAR, dob, getdate()) - 1
--else
--datediff(YEAR, dob, getdate()) end As Age
--from programmer where dob = (SELECT min(dob) from programmer where YEAR(doj)='1992' and GENDER='F');

-- 76. In which year was the most number of programmers born?

   CREATE TABLE TEMP (BIRTH_YEAR INT , SCOUNT INT)
 INSERT INTO TEMP
 SELECT
 YEAR(DOB) AS 'BIRTH YEAR',
 COUNT(PNAME)
 FROM Programmer
 GROUP BY YEAR(DOB)

 SELECT
 BIRTH_YEAR,
 SCOUNT
 FROM TEMP WHERE SCOUNT = (SELECT MAX(SCOUNT) FROM TEMP)

 SELECT * FROM TEMP

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

CREATE TABLE MST_JN_PN( JOIN_YEAR INT, PCOUNT INT)
INSERT INTO MST_JN_PN
SELECT
YEAR(DOJ) AS "JOINING YEAR",
COUNT(PNAME)
FROM Programmer
GROUP BY YEAR(DOJ
)

SELECT
JOIN_YEAR,
PCOUNT
FROM MST_JN_PN WHERE PCOUNT = (SELECT MAX(PCOUNT) FROM MST_JN_PN)

-- 78. In which language are most of the programmer’s proficient?
```

```
 CREATE TABLE #PROF1_LNG(PR1 VARCHAR(20),PR1_CNT INT)
 CREATE TABLE #PROF2_LNG(PR2 VARCHAR(20),PR2_CNT INT)

 INSERT INTO #PROF1_LNG
 SELECT PROF1, COUNT(PNAME) FROM Programmer GROUP BY PROF1
 UNION ALL
 SELECT PROF2, COUNT(PNAME) FROM Programmer GROUP BY PROF2

 INSERT INTO #PROF2_LNG
 SELECT PR1 , SUM(PR1_CNT) FROM #PROF1_LNG GROUP BY PR1

 SELECT
 PR2,
 PR2_CNT
 FROM #PROF2_LNG
 WHERE PR2_CNT = (SELECT MAX (PR2_CNT) FROM #PROF2_LNG)

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

 SELECT
 PNAME
 FROM Programmer
 WHERE GENDER = 'M'
 AND
 SALARY < (SELECT (AVG(SALARY)) FROM Programmer WHERE GENDER = 'F' )

-- 80. Who are the female programmers earning more than the highest paid?

 SELECT
 PNAME
```

```
  FROM Programmer
  WHERE
  GENDER = 'F'
  AND
  SALARY > (SELECT (MAX(SALARY)) FROM Programmer WHERE GENDER = 'M')
```

-- 81. Which language has been stated as the proficiency by most of the programmers?

```
  SELECT PROF1 FROM Programmer GROUP BY PROF1 HAVING PROF1 = (SELECT MAX (PROF1) FROM Programmer)
  UNION
  SELECT PROF2 FROM Programmer GROUP BY PROF2 HAVING PROF2 = (SELECT MAX (PROF2) FROM Programmer)
```

-- 82. 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(SALARY) > 1 )
```

-- 83. Display the details of the software developed by the male programmers earning more than 3000.

```
  SELECT *
  FROM Programmer P
  FULL OUTER JOIN Software S
  ON S.PNAME = P.PNAME
  WHERE SALARY > 3000 AND GENDER = 'M'

  --select * from programmer p,software s
  --where p.pname=s.pname and salary>3000 and gender='m';
```

-- 84. Display the details of the packages developed in Pascal by the female programmers.

```
  SELECT *
  FROM Software S
  FULL OUTER JOIN Programmer P
  ON P.PNAME = S.PNAME
  WHERE DEVELOPIN = 'PASCAL' AND GENDER = 'F'

  --select s.* from programmer p,software s
  --where p.pname=s.pname and gender='f' and DEVELOPIN ='pascal';

-- 85. Display the details of the Programmers who joined before 1990.

  SELECT * FROM Programmer
  WHERE YEAR(DOJ) < '1990'

-- 86. Display the details of the software developed in C by the female programmers at Pragathi.

  SELECT * FROM Software SS
  FULL OUTER JOIN Studies ST
  ON SS.PNAME = ST.PNAME
  FULL OUTER JOIN Programmer P
  ON ST.PNAME = P.PNAME
  WHERE DEVELOPIN = 'C' AND INSTITUTE = 'PRAGATHI' AND GENDER = 'F'

-- 87. Display the number of packages, number of copies sold and sales value of each programmer institute wise.

  SELECT
  INSTITUTE,
  COUNT(DEVELOPIN) AS 'DEVELOPING LANGAUGES',
  COUNT(SOLD) AS 'SOLD COPIES',
  SUM(SOLD * SCOST) AS 'SALES REVENUE'

  FROM Software SS

  INNER JOIN Studies ST

  ON SS.PNAME = ST.PNAME

  GROUP BY INSTITUTE


  --Select studies.institute,

  --count(software.developin) AS developin,

  --count(software.sold) As Sold,

  --sum(software.sold*software.scost) AS sales

  --from software,studies

  --where software.pname =studies.pname group by studies.institute;
```

-- 88. Display the details of the software developed in DBASE by Male Programmers, who belong to the institute in which most number of Programmers studied.

```
   CREATE TABLE #INST (INS VARCHAR(20), CNT INT)


INSERT INTO #INST

SELECT INSTITUTE, COUNT(PNAME) FROM studies GROUP BY INSTITUTE


SELECT distinct SW.* FROM software AS SW, programmer AS PG, studies AS ST, #INST

WHERE DEVELOPIN='DBASE' AND GENDER='M' AND SW.PNAME = PG.PNAME

AND INSTITUTE = INS AND CNT= (SELECT MAX(CNT) FROM #INST)
```

-- 89. Display the details of the software Developed by the male programmers Born before 1965 and female programmers born after 1975.

```
  SELECT * FROM Programmer P

  LEFT JOIN Software S

  ON S.PNAME = P.PNAME
  WHERE ((GENDER =  'M' AND YEAR(DOB) < 1965) OR (GENDER = 'F' AND YEAR(DOB) > 1975))
```

-- 90. Display the details of the software that has developed in the language which is neither the first nor the second proficiency of the programmers.

```
  SELECT * FROM Programmer P
  RIGHT JOIN Software S
  ON S.PNAME = P.PNAME
  WHERE PROF1 != DEVELOPIN AND PROF2 != DEVELOPIN



  --select s.* from programmer p,software s
  --where s.pname=p.pname and (developin <> prof1 and developin <> prof2);
```

-- 91. Display the details of the software developed by the male students of Sabhari.

```
  SELECT * FROM Programmer P
  JOIN Studies S
  ON
  P.PNAME = S.PNAME
  WHERE GENDER = 'M' AND INSTITUTE = 'SABHARI'
```

-- 92. Display the names of the programmers who have not developed any packages.

```
  SELECT PNAME FROM  Programmer
  WHERE PNAME NOT IN (SELECT PNAME FROM Software)
```

-- 93. What is the total cost of the Software developed by the programmers of Apple?

```
  --SELECT SUM(DCOST) AS 'TOTAL DEVELOPEMENT COST' FROM Software SS,Studies ST
  --WHERE SS.PNAME = ST.PNAME
  --AND
  --INSTITUTE = 'APPLE'

  SELECT SUM(DCOST) AS "APPLE S.DEVELOPEMENT COST" FROM Software SS
  JOIN Studies ST
  ON SS.PNAME = ST.PNAME
  WHERE INSTITUTE = 'APPLE'



-- 94. Who are the programmers who joined on the same day?

  SELECT
  A.PNAME,
  A.DOJ
  FROM Programmer A, Programmer B
  WHERE A.DOJ = B.DOJ AND A.PNAME != B.PNAME

-- 95. Who are the programmers who have the same Prof2?

  SELECT
  DISTINCT(A.PNAME),
  A.PROF2
  FROM Programmer A, Programmer B
  WHERE A.PROF2 = B.PROF2 AND A.PNAME != B.PNAME

-- 96. Display the total sales value of the software, institute wise.

  SELECT TITLE,INSTITUTE,SUM(SOLD * SCOST) AS 'SALES VALUE'FROM Software SS
  INNER JOIN Studies ST
  ON SS.PNAME = ST.PNAME
  GROUP BY TITLE,INSTITUTE
```

-- 97. In which institute does the person who developed the costliest package study?

```
  SELECT
  INSTITUTE
  FROM Software SS,Studies ST
  WHERE SS.PNAME = ST.PNAME
  GROUP BY INSTITUTE,DCOST
  HAVING DCOST = (SELECT MAX(DCOST) FROM Software)

  --select institute from software st,studies s
  --where s.pname=st.pname
  --group by institute,dcost
  --having dcost=(select max(dcost) from software);

  --98. Which language listed in prof1, prof2 has not been used to develop any package.

  select prof1 from programmer where prof1 not in(select developin from software)
  union
  select prof2 from programmer where prof2 not in(select developin from software);
```

-- 99. How much does the person who developed the highest selling package earn and what course did HE/SHE undergo.

```
select p1.salary,s2.course from programmer p1,software s1,studies s2
where p1.pname=s1.pname and s1.pname=s2.pname and scost=(select max(scost) from software);
```

-- 100. What is the AVG salary for those whose software sales is more than 50,000/-.

```
select avg(salary) from programmer p,software s
where p .pname=s.pname and sold*scost>50000;
```

-- 101. How many packages were developed by students, who studied in institute that charge the lowest course fee?

```
select s.pname, count(title) As packages from software s,studies st
where s.pname=st.pname group by s.pname,COURSE_FEE having min(COURSE_FEE)=(select min(COURSE_FEE) from studies);
```

-- 102. How many packages were developed by the person who developed the cheapest package, where did HE/SHE study?

```
select count(developin) from programmer p,software s
where s .pname=p.pname group by developin having min(dcost)=(select min(dcost) from software);
```

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

```
select count(developin) from programmer p,software s
where s.pname=p.pname and gender='f' and salary>(select max(salary) from programmer p,software s
where s.pname=p.pname and gender='m');
```

-- 104. How many packages are developed by the most experienced programmer form BDPS.

```
select count(*) from software s,programmer p
where p.pname=s.pname group by doj having min(doj)=(select min(doj)
from studies st,programmer p, software s
where p.pname=s.pname and st.pname=p.pname and (institute='bdps'));
```

-- 105. List the programmers (form the software table) and the institutes they studied, including those WHO DIDN'T develop any package.

```
select pname,institute from studies
where pname not in(select pname from software);
```

-- 106. List each PROF with the number of Programmers having that PROF and the number of the packages in that PROF.

```
select count(*),sum(scost*sold-dcost) "PROFIT" from software
```

```
where developin in (select prof1 from programmer) group by developin;
```

-- 107. List the programmer names (from the programmer table) and No. Of Packages each has developed.

```
select s.pname,count(developin) from programmer p1,software s

where p1.pname=s.pname group by s.pname;
```