

SQL QUERIES

NAME : ASWANTH RAMANA S M

DATE : 16.04.2024

TABLE :

```
create database aswanth;
use aswanth;
create table studies (
    PNAME VARCHAR(35),
    INSTITUTE VARCHAR(45),
    COURSE VARCHAR(35),
    COURSEFEE INT
);
INSERT INTO STUDIES (PNAME,INSTITUTE,COURSE,COURSEFEE) VALUES
('ANAND',"SABHARI","PGDCA",4500),('ATLAF',"COIT","DCA",7200),('JULIANA',"BITS","M
CA",22000),('KAMALA',"PRAGATHI","DCP",5000),('MARY',"SABHARI","PGDCA",4500),('N
ELSON',"PRAGATHI","DAP",6200),('PATRICK',"PRAGATHI","DCAP",5200),('QADIR',"APPL
E","HDCP",14000),('RAMESH',"SABHARI","PGDCA",4500),('RESECCA',"BRILLIANT","DCP",
11000),('REMITHA',"BDPS","DCS",6000),('VIJAYA',"BDPS","DCA",48000);
```

```
CREATE TABLE SOFTWARE(
    PNAME VARCHAR(45),
    TITLE VARCHAR(45),
    DEVELOPIN VARCHAR(45),
    SCOST FLOAT,
    DCOST INT,
    SOLD INT
);
INSERT INTO SOFTWARE (PNAME,TITLE,DEVELOPIN,SCOST,DCOST,SOLD) VALUES
('MARY',"README","CPP",100.00,1200,84),('ANAND',"PARACHUTES","BASIC",399.95,6000,
43),('ANAND',"VIDEOTITLING","PASCAL",7500.00,16000,9),('JULIANA',"INVENTORY","CO
BOL",3000.00,3500,0),('KAMALA',"PAYROLLPRG","DBASE",9000.00,20000,7),('MARY',"FIN
ANCIALACC","ORACLE",18000.00,85000,4),('MARY',"CODEGENERRATOP","C",4500.00,2000
0,23),('PATRICK',"README","CPP",300.00,1200,84),('QADIR',"BOMBSAWAY","ASSEMBL
Y",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",900.0
0,700,6);
SELECT * FROM SOFTWARE;
```

```
CREATE TABLE PROGRAMMER(
    PNAME VARCHAR(35),
    DOB DATE,
    DOJ DATE,
    SEX VARCHAR(3),
    PROF1 VARCHAR(35),
    PROF2 VARCHAR(35),
    SALARY INT
);
INSERT INTO PROGRAMMER (PNAME,DOB,DOJ,SEX,PROF1,PROF2,SALARY) 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",NULL,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);
SELECT * FROM PROGRAMMER;

```

QUERIES - 1

```

-- Q1 : Find out the selling cost of Packages developed in Pascal
SELECT SCOST FROM SOFTWARE WHERE DEVELOPIN = "PASCAL";

-- Q2 : Display the names ages of all programmers
SELECT PNAME,TIMESTAMPDIFF(YEAR,DOB,NOW()) FROM PROGRAMMER;

-- Q3 : Display the names and ages of all the programmers who have undergone training in DCS course.
SELECT P.PNAME, YEAR(CURRENT_DATE()) - YEAR(P.DOB) AS AGE
FROM PROGRAMMER P, STUDIES S
WHERE P.PNAME = S.PNAME
AND S.COURSE = 'DCS';

-- Q4 : What is the highest number of copies sold by a package
SELECT MAX(SOLD) AS HIGHEST_NUMBER_OF_COPIES FROM SOFTWARE;

-- Q5 : Display the names and date of births of all programmers born in january
SELECT DOB, PNAME FROM PROGRAMMER WHERE MONTH(DOB) = '01';

-- Q6 : Display the lowest course fee
SELECT MIN(COURSEFEE) AS LOWEST_COURSEFEE FROM STUDIES;

-- Q7 : How many programmers done the pgdca course
SELECT COUNT(PNAME) FROM STUDIES WHERE COURSE = "PGDCA";

-- Q8 : How much revenue has been earned thru sales of packages developed in c.
SELECT SUM(SCOST*SOLD) FROM SOFTWARE WHERE DEVELOPIN LIKE "C";

-- Q9 : Display the details of the software developed by Ramesh.
SELECT * FROM SOFTWARE WHERE PNAME = "RAMESH";

-- Q10 : How may programmers studied at Sabhari
SELECT COUNT(PNAME) FROM STUDIES WHERE INSTITUTE = "SABHARI";

-- Q11 : Display details of packages whose sales crossed the 2000 mark
SELECT * FROM SOFTWARE WHERE (DCOST*SOLD) > 2000;

-- Q12 : Find out the number of copies which should be sold in order to recover the development cost
of each package
SELECT TITLE, CEIL(DCOST / SCOST) AS COPIES_TO_SELL
FROM SOFTWARE;

-- Q13 : What is the cost of the costliest software developed in basic
SELECT MAX(DCOST) FROM SOFTWARE WHERE DEVELOPIN = "BASIC";

-- Q14 : Display the details of packages for which developments cost have been recovered.
SELECT * FROM SOFTWARE where SCOST*SOLD >= DCOST;

```

```

-- Q15 : How many packages developed in dbase
SELECT COUNT(TITLE) FROM SOFTWARE WHERE DEVELOPIN = "DBASE";

-- Q16 : How many programmers studies in Pragathi
SELECT COUNT(PNAME) FROM STUDIES WHERE INSTITUTE = "PRAGATHI";

-- Q17 : How many programmers paid 5000 to 10000 for their studies.
SELECT COUNT(PNAME) FROM STUDIES WHERE COURSEFEE >= 5000 AND COURSEFEE
<= 10000;

-- Q18 : What is avg course fee
SELECT AVG(COURSEFEE) FROM STUDIES;

-- Q19 : Display the detail the programmers knowing c
SELECT * FROM PROGRAMMER WHERE PROF1 = 'C' OR PROF2 = 'C';

-- Q20 : How many programmers know either Cobol or Pascal
SELECT COUNT(PNAME) AS PROGRAMMERS FROM PROGRAMMER WHERE PROF1 =
'COBOL' OR PROF2 = 'PASCAL' OR PROF1 = 'PASCAL' OR PROF2 = 'COBOL';

-- Q21 : How many programmers don't know Pascal and C
SELECT COUNT(PNAME) AS PROGRAMMERS FROM PROGRAMMER WHERE PROF1 !=
'PASCAL' AND PROF2 != 'C' AND PROF1 != 'C' AND PROF2 != 'PASCAL';

-- Q22 : How old is the oldest male programmers
SELECT MAX(TIMESTAMPDIFF(YEAR,DOB,NOW())) AS OLDEST_MALE_PROGRAMMER
FROM PROGRAMMER WHERE SEX = 'M';

-- Q23 : What is the average age of female programmers?
SELECT AVG(YEAR(CURRENT_DATE()) - YEAR(DOB)) AS AVERAGE_AGE
FROM PROGRAMMER
WHERE SEX = 'F';

-- Q24 : Calculate the experience in years for each programmers and. display along with the names in
descending order.
SELECT PNAME,ABS(TIMESTAMPDIFF(YEAR,DOB,DOJ)) FROM PROGRAMMER ORDER
BY ABS(TIMESTAMPDIFF(YEAR,DOB,DOJ)) DESC;

-- Q25 : Who are the programmers who celebrate their birthdays during the current month.
SELECT PNAME,DOB FROM PROGRAMMER WHERE MONTH(DOB) = '02';

-- Q26 : How many female programmers are there
SELECT COUNT(PNAME) AS FEMALE_PROGRAMMERS FROM PROGRAMMER WHERE
SEX = 'F';

-- Q27 : What are the languages by male programmers
SELECT PROF1 FROM PROGRAMMER WHERE SEX = 'M' UNION DISTINCT SELECT PROF2
FROM PROGRAMMER WHERE SEX = 'M';

-- Q28 : What is the average salary
SELECT AVG (SALARY) AS AVGSAL FROM PROGRAMMER;

-- Q29 : How many people draw salary 2000 to 4000
SELECT COUNT(PNAME) AS DRAW_SALARY FROM PROGRAMMER WHERE SALARY >=
2000 AND SALARY <= 4000;

```

```

-- Q30 : Display the details of those who don't know clipper Cobol or Pascal
SELECT * FROM PROGRAMMER WHERE PROF1!= 'COBOL' AND PROF2!= 'COBOL' AND
PROF1!= 'CLIPPER' AND PROF2!= 'CLIPPER' AND PROF1!= 'PASCAL' AND PROF2!=
'PASCAL';

-- Q31 : How many Female programmers knowing C are above 24 years of age?
SELECT COUNT(*) AS NUM_FEMALE_PROGRAMMERS
FROM PROGRAMMER
WHERE SEX = 'F'
AND PROF1 = 'C'
AND YEAR(CURRENT_DATE()) - YEAR(DOB) > 24;

-- Q32 : Who are the programmers who will be celebrating their Birthday within a week?
SELECT PNAME, DOB
FROM PROGRAMMER
WHERE DAYOFYEAR(DOB) BETWEEN DAYOFYEAR(CURRENT_DATE()) AND
DAYOFYEAR(DATE_ADD(CURRENT_DATE(), INTERVAL 7 DAY));

-- Q33 : Display the details of those with less than a year's experience?
SELECT * FROM PROGRAMMER
WHERE DATEDIFF(CURRENT_DATE(), DOJ) < 365;

-- Q34 : Display the details of those who will be completing 2 years of service this year?
SELECT * FROM PROGRAMMER
WHERE YEAR(DATE_ADD(DOJ, INTERVAL 2 YEAR)) = YEAR(CURRENT_DATE());

-- Q35 : Calculate the amount to be recovered for those packages whose development cost has not been
recovered?
SELECT TITLE, (DCOST - (SCOST * SOLD)) AS AMOUNT_TO_BE_RECOVERED FROM
SOFTWARE WHERE DCOST > (SCOST * SOLD);

-- Q36 : List the packages which have not been sold so far?
SELECT TITLE FROM SOFTWARE WHERE SOLD = 0;

-- Q37 : Find out the cost of the software developed by Mary?
SELECT SUM(DCOST) AS TOTAL_COST FROM SOFTWARE WHERE PNAME = 'MARY';

-- Q38 : Display the institute's names from the studies table without duplicates?
SELECT DISTINCT INSTITUTE FROM STUDIES;

-- Q39 : How many different courses are mentioned in the studies table?
SELECT COUNT(DISTINCT COURSE) AS NUM_COURSES FROM STUDIES;

-- Q40 : Display the names of the programmers whose names contain 2 occurrences of the letter A?
SELECT PNAME FROM PROGRAMMER WHERE PNAME LIKE '%A%A%A%';

-- Q41: Display the names of programmers whose names contain upto 5 characters?
SELECT PNAME FROM PROGRAMMER WHERE LENGTH(PNAME) <= 5;

-- Q42: How many female programmers knowing COBOL have more than 2 years experience?
SELECT COUNT(*) AS NUM_FEMALE_PROGRAMMERS FROM PROGRAMMER
WHERE SEX = 'F' AND PROF1 = 'COBOL' AND DATEDIFF(CURRENT_DATE(), DOJ) > 730;

-- Q43: What is the length of the shortest name in the programmer table?
SELECT MIN(LENGTH(PNAME)) AS SHORTEST_NAME_LENGTH FROM PROGRAMMER;

-- Q44: What is the average development cost of a package developed in COBOL?
SELECT AVG(DCOST) AS AVERAGE_DEVELOPMENT_COST
FROM SOFTWARE
WHERE DEVELOPIN = 'COBOL';

```

-- Q45: Display the name, sex, dob (DD/MM/YY format), doj for all the programmers without using conversion function?

```
SELECT PNAME, SEX,  
       CONCAT(DAY(DOB), '/', MONTH(DOB), '/', RIGHT(YEAR(DOB), 2)) AS DOB,  
       CONCAT(DAY(DOJ), '/', MONTH(DOJ), '/', RIGHT(YEAR(DOJ), 2)) AS DOJ  
FROM PROGRAMMER;
```

-- Q46: Who is the highest paid in c programmers.

```
SELECT * FROM PROGRAMMER WHERE DAY(DOB) = DAY(LAST_DAY(DOB));
```

-- Q47. What is the amount paid in salaries of the male programmers who do not know Cobol?

```
SELECT SUM(SALARY) AS TOTAL_SALARY  
FROM PROGRAMMER  
WHERE SEX = 'M' AND (PROF1 <> 'COBOL' OR PROF1 IS NULL)  
AND (PROF2 <> 'COBOL' OR PROF2 IS NULL);
```

-- Q48. Display the title, scost, dcost and difference between scost and dcost in descending order of difference?

```
SELECT TITLE, SCOST, DCOST, SCOST - DCOST AS DIFFERENCE FROM SOFTWARE  
ORDER BY DIFFERENCE DESC;
```

-- Q49. Display the name, dob, doj of those month of birth and month of joining are same?

```
SELECT PNAME, DOB, DOJ  
FROM PROGRAMMER  
WHERE MONTH(DOB) = MONTH(DOJ);
```

-- Q50. Display the names of the packages whose names contain more than 1 word?

```
SELECT TITLE  
FROM SOFTWARE  
WHERE TITLE LIKE '% %';
```

QUERIES - 2

-- Q51. Display THE NUMBER OF packages developed in EACH language.

```
SELECT DEVELOPIN AS LANGUAGE, COUNT(*) AS NUMBER_OF_PACKAGES  
FROM SOFTWARE  
GROUP BY DEVELOPIN;
```

-- Q52. Display THE NUMBER OF packages developed by EACH person.

```
SELECT PNAME, COUNT(*) AS NUMBER_OF_PACKAGES  
FROM SOFTWARE  
GROUP BY PNAME;
```

-- Q53. Display THE NUMBER OF male and female programmer.

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

-- Q54. Display THE COSTLIEST packages and HIGEST selling developed in EACH language.

```
SELECT DEVELOPIN AS LANGUAGE,  
       MAX(DCOST) AS COSTLIEST_PACKAGE,  
       MAX(SCOST) AS HIGHEST_SELLING  
FROM SOFTWARE  
GROUP BY DEVELOPIN;
```

-- Q55. Display THE NUMBER OF people BORN in EACH YEAR.

```
SELECT YEAR(DOB) AS BIRTH_YEAR, COUNT(*) AS NUMBER_OF_PEOPLE  
FROM PROGRAMMER  
GROUP BY YEAR(DOB);
```

-- Q56. Display THE NUMBER OF people JOINED in EACH YEAR.
SELECT YEAR(DOJ) AS JOIN_YEAR, COUNT(*) AS NUMBER_OF_PEOPLE
FROM PROGRAMMER
GROUP BY YEAR(DOJ);

-- Q57. Display THE NUMBER OF people BORN in EACH MONTH.
SELECT MONTH(DOB) AS BIRTH_MONTH, COUNT(*) AS NUMBER_OF_PEOPLE
FROM PROGRAMMER
GROUP BY MONTH(DOB);

-- Q58. Display THE NUMBER OF people JOINED in EACH MONTH.
SELECT MONTH(DOJ) AS JOIN_MONTH, COUNT(*) AS NUMBER_OF_PEOPLE
FROM PROGRAMMER
GROUP BY MONTH(DOJ);

-- Q59. Display the language wise COUNTS of prof1.
SELECT DEVELOPIN AS LANGUAGE, PROF1, COUNT(*) AS COUNT_OF_PROF1
FROM SOFTWARE
GROUP BY DEVELOPIN, PROF1;

-- Q60. Display the language wise COUNTS of prof2.
SELECT DEVELOPIN AS LANGUAGE, PROF2, COUNT(*) AS COUNT_OF_PROF2
FROM SOFTWARE
GROUP BY DEVELOPIN, PROF2;

-- Q61. Display THE NUMBER OF people in EACH salary group.
SELECT
CASE
WHEN SALARY < 3000 THEN 'Less than 3000'
WHEN SALARY >= 3000 AND SALARY < 5000 THEN '3000 - 4999'
WHEN SALARY >= 5000 AND SALARY < 7000 THEN '5000 - 6999'
WHEN SALARY >= 7000 AND SALARY < 9000 THEN '7000 - 8999'
ELSE 'More than 9000'
END AS SALARY_GROUP,
COUNT(*) AS NUMBER_OF_PEOPLE
FROM
PROGRAMMER
GROUP BY
SALARY_GROUP;

-- Q62. Display THE NUMBER OF people who studied in EACH institute.
SELECT INSTITUTE, COUNT(*) AS NUMBER_OF_PEOPLE
FROM STUDIES
GROUP BY INSTITUTE;

-- Q63. Display THE NUMBER OF people who studied in EACH course.
SELECT COURSE, COUNT(*) AS NUMBER_OF_PEOPLE
FROM STUDIES
GROUP BY COURSE;

-- Q64. Display the TOTAL development COST of the packages developed in EACH language.
SELECT DEVELOPIN AS LANGUAGE, SUM(DCOST) AS TOTAL_DEVELOPMENT_COST
FROM SOFTWARE
GROUP BY DEVELOPIN;

-- Q65. Display the selling cost of the package developed in EACH language.
SELECT DEVELOPIN AS LANGUAGE, SUM(SCOST) AS TOTAL_SELLING_COST
FROM SOFTWARE
GROUP BY DEVELOPIN;

-- Q66. Display the cost of the package developed by EACH programmer.
SELECT PNAME, SUM(DCOST) AS TOTAL_DEVELOPMENT_COST
FROM SOFTWARE
GROUP BY PNAME;

-- Q67. Display the sales values of the package developed in EACH programmer.
SELECT PNAME, SUM(SCOST * SOLD) AS TOTAL_SALES_VALUE
FROM SOFTWARE
GROUP BY PNAME;

-- Q68. Display the NUMBER of packages developed by EACH programmer.
SELECT PNAME, COUNT(*) AS NUMBER_OF_PACKAGES_DEVELOPED
FROM SOFTWARE
GROUP BY PNAME;

-- Q69. Display the sales COST of packages developed by EACH programmer language wise.
SELECT PNAME, DEVELOPIN AS LANGUAGE, SUM(SCOST * SOLD) AS
TOTAL_SALES_COST
FROM SOFTWARE
GROUP BY PNAME, DEVELOPIN;

-- Q70. Display EACH programmers name, costliest package and cheapest packages developed by
Him/Her.
SELECT PNAME,
 MAX(TITLE) AS COSTLIEST_PACKAGE,
 MIN(TITLE) AS CHEAPEST_PACKAGE
FROM SOFTWARE
GROUP BY PNAME;

-- Q71. Display EACH language name with AVERAGE development cost, AVERAGE cost, selling
cost and AVERAGE price per copy.
SELECT
 DEVELOPIN AS LANGUAGE,
 AVG(DCOST) AS AVERAGE_DEVELOPMENT_COST,
 AVG(DCOST + SCOST) AS AVERAGE_COST,
 AVG(SCOST) AS AVERAGE_SELLING_COST,
 AVG(SCOST / SOLD) AS AVERAGE_PRICE_PER_COPY
FROM SOFTWARE
GROUP BY DEVELOPIN;

-- Q72. Display EACH institute name with NUMBER of courses, AVERAGE cost per course.
SELECT
 INSTITUTE,
 COUNT(DISTINCT COURSE) AS NUMBER_OF_COURSES,
 AVG(COURSEFEE) AS AVERAGE_COST_PER_COURSE
FROM STUDIES
GROUP BY INSTITUTE;

-- Q73. Display EACH institute name with NUMBER of students.
SELECT
 INSTITUTE,
 COUNT(DISTINCT PNAME) AS NUMBER_OF_STUDENTS
FROM STUDIES
GROUP BY INSTITUTE;

-- Q74. Display names of male and female programmers.
SELECT DISTINCT PNAME, SEX FROM PROGRAMMER;

-- Q75. Display the programmer's name and their packages.

```
SELECT
  PNAME AS PROGRAMMER_NAME,
  (SELECT TITLE FROM SOFTWARE WHERE PNAME = P.PNAME) AS PACKAGE_NAME
FROM PROGRAMMER P;
```

-- Q76. Display the NUMBER of packages in EACH language.

```
SELECT DEVELOPIN AS LANGUAGE, COUNT(*) AS NUMBER_OF_PACKAGES
FROM SOFTWARE
GROUP BY DEVELOPIN;
```

-- Q77. Display the NUMBER of packages in EACH language for which development cost is less than 1000.

```
SELECT DEVELOPIN AS LANGUAGE, COUNT(*) AS NUMBER_OF_PACKAGES
FROM SOFTWARE
WHERE DCOST < 1000
GROUP BY DEVELOPIN;
```

-- Q78. Display the AVERAGE difference BETWEEN scost and dcost for EACH language.

```
SELECT DEVELOPIN AS LANGUAGE, AVG(SCOST - DCOST) AS AVERAGE_DIFFERENCE
FROM SOFTWARE
GROUP BY DEVELOPIN;
```

-- Q79. Display the TOTAL scost, dcost and amount TOBE recovered for EACH programmer for whose dcost HAS NOT YET BEEN recovered.

```
SELECT PNAME,
  SUM(SCOST) AS TOTAL_SCOST,
  SUM(DCOST) AS TOTAL_DCOST,
  SUM(CASE WHEN SCOST < DCOST THEN DCOST - SCOST ELSE 0 END) AS
  AMOUNT_TO_BE_RECOVERED
FROM SOFTWARE
GROUP BY PNAME;
```

-- Q80. Display highest, lowest and average salaries for THOSE earning MORE than 2000.

```
SELECT MAX(SALARY) AS HIGHEST_SALARY,
  MIN(SALARY) AS LOWEST_SALARY,
  AVG(SALARY) AS AVERAGE_SALARY
FROM PROGRAMMER
WHERE SALARY > 2000;
```

QUERIES - 3

-- Q81. Who is the highest paid C programmer?

```
SELECT PNAME
FROM PROGRAMMER
WHERE PROF1 = 'C'
ORDER BY SALARY DESC
LIMIT 1;
```

-- Q82. Who is the highest paid female cobol programmer?

```
SELECT PNAME
FROM PROGRAMMER
WHERE SEX = 'F' AND PROF1 = 'COBOL'
ORDER BY SALARY DESC
LIMIT 1;
```



```
-- Q83. Display the name of the HIGEST paid programmer for EACH language (prof1)
SELECT DEVELOPIN AS LANGUAGE, PNAME
FROM PROGRAMMER
WHERE (PROF1, SALARY) IN (SELECT PROF1, MAX(SALARY) FROM PROGRAMMER
GROUP BY PROF1)
GROUP BY LANGUAGE;
```

```
-- Q84. Who is the LEAST experienced programmer?
SELECT PNAME
FROM PROGRAMMER
ORDER BY DOJ
LIMIT 1;
```

```
-- Q85. Who is the MOST experienced programmer?
SELECT PNAME
FROM PROGRAMMER
ORDER BY DOJ DESC
LIMIT 1;
```

```
-- Q86. Which language is known by ONLY ONE programmer?
SELECT DEVELOPIN AS LANGUAGE
FROM (
    SELECT DEVELOPIN, COUNT(*) AS NUM_PROGRAMMERS
    FROM SOFTWARE
    GROUP BY DEVELOPIN
) AS LanguageCount
WHERE NUM_PROGRAMMERS = 1;
```

```
-- Q87. Who is the youngest programmer knowing DBASE?
SELECT PNAME
FROM PROGRAMMER
WHERE PROF1 = 'DBASE' OR PROF2 = 'DBASE'
ORDER BY DOB ASC
LIMIT 1;
```

```
-- Q88. Which institute has the most number of students?
SELECT INSTITUTE
FROM STUDIES
GROUP BY INSTITUTE
ORDER BY COUNT(DISTINCT PNAME) DESC
LIMIT 1;
```

```
-- Q89. Who is the above programmer?
SELECT PNAME
FROM STUDIES
WHERE INSTITUTE = (
    SELECT INSTITUTE
    FROM STUDIES
    GROUP BY INSTITUTE
    ORDER BY COUNT(DISTINCT PNAME) DESC
    LIMIT 1
)
GROUP BY PNAME
ORDER BY DOB ASC
LIMIT 1;
```

-- Q90. Which female programmer earns more than 3000/- but does not know C, C++, Oracle, or Dbase?

```
SELECT PNAME
FROM PROGRAMMER
WHERE SEX = 'F' AND SALARY > 3000
AND PROF1 NOT IN ('C', 'C++', 'ORACLE', 'DBASE')
AND (PROF2 NOT IN ('C', 'C++', 'ORACLE', 'DBASE') OR PROF2 IS NULL);
```

-- Q91. Which is the costliest course?

```
SELECT COURSE
FROM STUDIES
GROUP BY COURSE
ORDER BY MAX(COURSEFEE) DESC
LIMIT 1;
```

-- Q92. Which course has been done by most of the students?

```
SELECT COURSE
FROM STUDIES
GROUP BY COURSE
ORDER BY COUNT(*) DESC
LIMIT 1;
```

-- Q93. Display name of the institute and course Which has below AVERAGE course fee?

```
SELECT INSTITUTE, COURSE
FROM STUDIES
WHERE COURSEFEE < (SELECT AVG(COURSEFEE) FROM STUDIES);
```

-- Q94. Which institute conducts COSTLIEST course?

```
SELECT INSTITUTE
FROM STUDIES
WHERE COURSEFEE = (SELECT MAX(COURSEFEE) FROM STUDIES);
```

-- Q95. Which course has below AVERAGE number of students?

```
SELECT COURSE
FROM STUDIES
GROUP BY COURSE
HAVING COUNT(DISTINCT PNAME) < (SELECT AVG(STUDENT_COUNT) FROM (SELECT
COUNT(DISTINCT PNAME) AS STUDENT_COUNT FROM STUDIES GROUP BY COURSE) AS
AVG_STUDENT_COUNT);
```

-- Q96. Which institute conducts the above course?

```
SELECT INSTITUTE
FROM STUDIES
WHERE COURSE = (
    SELECT COURSE
    FROM STUDIES
    GROUP BY COURSE
    ORDER BY MAX(COURSEFEE) DESC
    LIMIT 1
);
```

-- Q97. Display names of the course WHOSE fees are within 1000(+ or -) of the AVERAGE fee.

```
SELECT COURSE
FROM STUDIES
WHERE COURSEFEE BETWEEN (SELECT AVG(COURSEFEE) - 1000 FROM STUDIES) AND
(SELECT AVG(COURSEFEE) + 1000 FROM STUDIES);
```

-- Q98. Which package has the HIGEST development cost?

```
SELECT TITLE
FROM SOFTWARE
WHERE DCOST = (SELECT MAX(DCOST) FROM SOFTWARE);
```

-- Q99. Which package has the LOWEST selling cost?

```
SELECT TITLE
FROM SOFTWARE
WHERE SCOST = (SELECT MIN(SCOST) FROM SOFTWARE);
```

-- Q100. Who developed the package, which has sold the LEAST number of copies?

```
SELECT PNAME
FROM SOFTWARE
WHERE SOLD = (SELECT MIN(SOLD) FROM SOFTWARE);
```

-- Q101. Which language was used to develop the package WHICH has the HIGEST sales amount?

```
SELECT DEVELOPIN
FROM SOFTWARE
WHERE SCOST = (SELECT MAX(SCOST) FROM SOFTWARE);
```

-- Q102. How many copies of the package that has the LEAST DIFFERENCE between development and selling cost were sold?

```
SELECT SOLD
FROM SOFTWARE
WHERE ABS(SCOST - DCOST) = (
    SELECT MIN(ABS(SCOST - DCOST))
    FROM SOFTWARE
);
```

-- Q103. Which is the COSTLIEAST package developed in PASCAL?

```
SELECT TITLE
FROM SOFTWARE
WHERE DEVELOPIN = 'PASCAL'
ORDER BY SCOST DESC
LIMIT 1;
```

-- Q104. Which language was used to develop the MOST NUMBER of package?

```
SELECT DEVELOPIN
FROM SOFTWARE
GROUP BY DEVELOPIN
ORDER BY COUNT(*) DESC
LIMIT 1;
```

-- Q105. Which programmer has developed the HIGEST NUMBER of package?

```
SELECT PNAME
FROM SOFTWARE
GROUP BY PNAME
ORDER BY COUNT(*) DESC
LIMIT 1;
```

-- Q106 : Who is the author of the COSTLIEST package?

```
SELECT PNAME
FROM SOFTWARE
WHERE SCOST = (SELECT MAX(SCOST) FROM SOFTWARE);
```

-- Q107 : Display names of packages WHICH have been sold LESS THAN the AVERAGE number of copies?

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

-- Q108 : Who are the female programmers earning MORE than the HIGHEST paid male programmers?

```
SELECT PNAME
FROM PROGRAMMER
WHERE SEX = 'F' AND SALARY > (
    SELECT MAX(SALARY)
    FROM PROGRAMMER
    WHERE SEX = 'M'
);
```

-- Q109 : Which language has been stated as prof1 by MOST of the programmers?

```
SELECT PROF1 AS LANGUAGE
FROM PROGRAMMER
GROUP BY PROF1
ORDER BY COUNT(*) DESC
LIMIT 1;
```

-- Q110 : Who are the authors of packages, WHICH have recovered MORE THAN double the development cost?

```
SELECT PNAME
FROM SOFTWARE
GROUP BY PNAME
HAVING SUM(SCOST) > 2 * SUM(DCOST);
```

-- Q111 : Display programmer names and CHEAPEST package developed by them in EACH language?

```
SELECT PNAME, MIN(TITLE) AS CHEAPEST_PACKAGE, DEVELOPIN AS LANGUAGE
FROM SOFTWARE
GROUP BY PNAME, DEVELOPIN;
```

-- Q112 : Who is the YOUNGEST male programmer born in 1965?

```
SELECT PNAME
FROM PROGRAMMER
WHERE SEX = 'M' AND DOB = (
    SELECT MIN(DOB)
    FROM PROGRAMMER
    WHERE SEX = 'M' AND YEAR(DOB) = 1965
);
```

-- Q113 : Display language used by EACH programmer to develop the HIGHEST selling and LOWEST selling package.

```
SELECT PNAME,
    (SELECT DEVELOPIN FROM SOFTWARE WHERE P.PNAME = SOFTWARE.PNAME AND
    SCOST = (SELECT MAX(SCOST) FROM SOFTWARE WHERE PNAME = P.PNAME)) AS
    HIGHEST_SELLING_LANGUAGE,
    (SELECT DEVELOPIN FROM SOFTWARE WHERE P.PNAME = SOFTWARE.PNAME AND
    SCOST = (SELECT MIN(SCOST) FROM SOFTWARE WHERE PNAME = P.PNAME)) AS
    LOWEST_SELLING_LANGUAGE
FROM PROGRAMMER P;
```

-- Q114 : Who is the OLDEST female programmer WHO joined in 1992?

```
SELECT PNAME
FROM PROGRAMMER
WHERE SEX = 'F' AND YEAR(DOJ) = 1992
ORDER BY DOB ASC
LIMIT 1;
```

```

-- Q115 : In WHICH year where the MOST NUMBER of programmer born?
SELECT YEAR(DOB) AS BIRTH_YEAR, COUNT(*) AS NUMBER_OF_PROGRAMMERS
FROM PROGRAMMER
GROUP BY YEAR(DOB)
ORDER BY NUMBER_OF_PROGRAMMERS DESC
LIMIT 1;

-- Q116 : In WHICH month did MOST NUMBRER of programmer join?
SELECT MONTH(DOJ) AS JOIN_MONTH, COUNT(*) AS NUMBER_OF_PROGRAMMERS
FROM PROGRAMMER
GROUP BY JOIN_MONTH
ORDER BY NUMBER_OF_PROGRAMMERS DESC
LIMIT 1;

-- Q117 : In WHICH language are MOST of the programmer's proficient?
SELECT PROF1 AS LANGUAGE
FROM PROGRAMMER
GROUP BY PROF1
ORDER BY COUNT(*) DESC
LIMIT 1;

-- Q118 : Who are the male programmers earning BELOW the AVERAGE salary of female
programmers?
SELECT PNAME FROM PROGRAMMER WHERE SEX = 'M' AND SALARY < (SELECT
AVG(SALARY) FROM PROGRAMMER WHERE SEX = 'F');

```

QUERIES - 4

```

-- Q119 : 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(*) > 1
);

-- Q120 : Display the details of software developed by male programmers earning MORE than 3000.
SELECT *
FROM SOFTWARE
WHERE PNAME IN (
    SELECT PNAME
    FROM PROGRAMMER
    WHERE SEX = 'M' AND SALARY > 3000
);

-- Q121 : Display details of packages developed in PASCAL by female programmers.
SELECT *
FROM SOFTWARE
WHERE PNAME IN (
    SELECT PNAME
    FROM PROGRAMMER
    WHERE SEX = 'F'
) AND DEVELOPIN = 'PASCAL';

```

-- Q122 : Display the details of the programmer WHO joined BEFORE 1990.
SELECT *
FROM PROGRAMMER
WHERE YEAR(DOJ) < 1990;

-- Q123 : Display details of software developed in C by female programmers of PRAGATHI.
SELECT *
FROM SOFTWARE
WHERE PNAME IN (
 SELECT PNAME
 FROM PROGRAMMER
 WHERE SEX = 'F' AND INSTITUTE = 'PRAGATHI'
) AND DEVELOPIN = 'C';

-- Q124 : Display NUMBER of packages NUMBER of copies sold and sales value of EACH programmer Institute-wise.
SELECT PNAME, INSTITUTE, COUNT(*) AS NUM_PACKAGES, SUM(SOLD) AS
TOTAL_COPIES_SOLD, SUM(SCOST * SOLD) AS SALES_VALUE
FROM SOFTWARE S
JOIN PROGRAMMER P ON S.PNAME = P.PNAME
GROUP BY PNAME, INSTITUTE;

-- Q125 : Display details of software developed in DBASE by male programmers WHO belong to the institute on which MOST NUMBER OF programmer's studies.
SELECT *
FROM SOFTWARE
WHERE DEVELOPIN = 'DBASE' AND PNAME IN (
 SELECT PNAME
 FROM PROGRAMMER
 WHERE SEX = 'M' AND INSTITUTE = (
 SELECT INSTITUTE
 FROM PROGRAMMER
 GROUP BY INSTITUTE
 ORDER BY COUNT(*) DESC
 LIMIT 1
)
);

-- Q126 : Display the details of the software that was developed by male programmers born BEFORE 1965 and female programmers born AFTER 1975.
SELECT *
FROM SOFTWARE
WHERE PNAME IN (
 SELECT PNAME
 FROM PROGRAMMER
 WHERE (SEX = 'M' AND YEAR(DOB) < 1965)
 OR (SEX = 'F' AND YEAR(DOB) > 1975)
);

-- Q127 : Display the details of the software that was developed in the language that is NOT the programmer's first proficiency.
SELECT *
FROM SOFTWARE
WHERE DEVELOPIN NOT IN (
 SELECT PROF1
 FROM PROGRAMMER
);

-- Q128 : Display details of software that was developed in the language which is NEITHER first NOR second proficiency of the programmer.

```
SELECT *
FROM SOFTWARE
WHERE DEVELOPIN NOT IN (
    SELECT PROF1 FROM PROGRAMMER
    UNION
    SELECT PROF2 FROM PROGRAMMER
);
```

-- Q129 : Display details of software developed by male students of SABHARI.

```
SELECT *
FROM SOFTWARE
WHERE PNAME IN (
    SELECT PNAME
    FROM PROGRAMMER
    WHERE SEX = 'M' AND INSTITUTE = 'SABHARI'
);
```

-- Q130 : Display the names of programmers WHO HAVE NOT developed any package.

```
SELECT PNAME
FROM PROGRAMMER
WHERE PNAME NOT IN (SELECT DISTINCT PNAME FROM SOFTWARE);
```

-- Q131 : What is the total cost of the software developed by the programmers by APPLE?

```
SELECT SUM(SCOST) AS TOTAL_COST
FROM SOFTWARE
WHERE PNAME IN (
    SELECT PNAME
    FROM PROGRAMMER
    WHERE INSTITUTE = 'APPLE'
);
```

-- Q132 : Who are the programmers WHO JOINED in the same day?

```
SELECT PNAME
FROM PROGRAMMER
GROUP BY PNAME, DOJ
HAVING COUNT(*) > 1;
```

-- Q133 : Who are the programmers WHO HAVE THE SAME PROF2?

```
SELECT PNAME
FROM PROGRAMMER
GROUP BY PROF2
HAVING COUNT(*) > 1;
```

-- Q134 : Display the total sales values of software, institutes-wise.

```
SELECT INSTITUTE, SUM(SCOST * SOLD) AS TOTAL_SALES_VALUE
FROM SOFTWARE S
JOIN PROGRAMMER P ON S.PNAME = P.PNAME
GROUP BY INSTITUTE;
```

-- Q135 : In which institutes did the person who developed the COSTLIEST package study?

```
SELECT INSTITUTE
FROM PROGRAMMER
WHERE PNAME IN (
    SELECT PNAME
    FROM SOFTWARE
    WHERE SCOST = (SELECT MAX(SCOST) FROM SOFTWARE)
);
```

-- Q136 : Which language listed in prof1 and prof2 HAS NOT BEEN used to develop any package?

```
SELECT DISTINCT LANGUAGE
FROM (
    SELECT PROF1 AS LANGUAGE FROM PROGRAMMER
    UNION
    SELECT PROF2 AS LANGUAGE FROM PROGRAMMER
) AS LANGUAGES
WHERE LANGUAGE NOT IN (SELECT DISTINCT DEVELOPIN FROM SOFTWARE);
```

-- Q137 : How much does the person WHO developed the HIGHEST selling package earn and WHAT course did he/she undergo?

```
SELECT P.PNAME, P.SALARY, S.TITLE, S.SCOST
FROM PROGRAMMER P, SOFTWARE S
WHERE P.PNAME = S.PNAME
AND S.SCOST = (SELECT MAX(SCOST) FROM SOFTWARE);
```

-- Q138 : How many months will it take for each programmer to recover the cost of the course underwent?

```
SELECT PNAME, SALARY / COURSEFEE AS MONTHS_TO_RECOVER
FROM PROGRAMMER, STUDIES
WHERE PROGRAMMER.PNAME = STUDIES.PNAME;
```

-- Q139 : Which is the COSTLIEST package developed by a person with under 3 year's experiences?

```
SELECT TITLE
FROM SOFTWARE
WHERE PNAME IN (
    SELECT PNAME
    FROM PROGRAMMER
    WHERE DATEDIFF(NOW(), DOJ) < 3*365
)
ORDER BY SCOST DESC
LIMIT 1;
```

-- Q140 : What is the AVERAGE salary for those WHOSE software's sales value is more than 50,000?

```
SELECT AVG(SALARY) AS AVERAGE_SALARY
FROM PROGRAMMER
WHERE PNAME IN (
    SELECT PNAME
    FROM SOFTWARE
    GROUP BY PNAME
    HAVING SUM(SCOST * SOLD) > 50000
);
```

-- Q141 : How many packages were developed by the students WHO studied in the institute that charge the LOWEST course fee?

```
SELECT COUNT(*) AS NUM_PACKAGES
FROM SOFTWARE
WHERE PNAME IN (
    SELECT PNAME
    FROM STUDIES
    WHERE COURSEFEE = (
        SELECT MIN(COURSEFEE)
        FROM STUDIES
    )
);
```


-- Q142 : How many packages were developed by the person WHO developed the CHEAPEST package? Where did he\she study?

```
SELECT COUNT(*) AS NUM_PACKAGES, INSTITUTE
FROM SOFTWARE, STUDIES
WHERE SOFTWARE.PNAME = STUDIES.PNAME
AND SCOST = (
    SELECT MIN(SCOST)
    FROM SOFTWARE
)
GROUP BY INSTITUTE;
```

-- Q143 : How many packages were developed by female programmers earning MORE than the HIGHEST paid male programmer?

```
SELECT COUNT(*) AS NUM_PACKAGES
FROM SOFTWARE
WHERE PNAME IN (
    SELECT PNAME
    FROM PROGRAMMER
    WHERE SEX = 'F' AND SALARY > (
        SELECT MAX(SALARY)
        FROM PROGRAMMER
        WHERE SEX = 'M'
    )
);
```

-- Q144 : How many packages were developed by the MOST experienced programmers from BDPS?

```
SELECT COUNT(*) AS NUM_PACKAGES
FROM SOFTWARE
WHERE PNAME IN (
    SELECT PNAME
    FROM PROGRAMMER
    WHERE INSTITUTE = 'BDPS'
    ORDER BY DATEDIFF(NOW(), DOJ) DESC
    LIMIT 1
);
```

-- Q145 : Who are the programmers WHO JOINED in the same day? List the programmers (from software table) and institutes they studied, including those WHO DIDN'T develop any package.

```
SELECT DISTINCT P.PNAME,
    CASE WHEN S.PNAME IS NOT NULL THEN S.INSTITUTE ELSE 'N/A' END AS
INSTITUTE
FROM PROGRAMMER P, SOFTWARE S
WHERE P.PNAME = S.PNAME
OR S.PNAME IS NULL;
```

-- Q146 : List each profit with the number of programmers having that prof1 and the number of packages developed in that prof1.

```
SELECT PROF1, COUNT(DISTINCT PNAME) AS NUM_PROGRAMMERS, COUNT(*) AS
NUM_PACKAGES
FROM PROGRAMMER
GROUP BY PROF1;
```

-- Q147 : List programmer names (from programmer table) and number of packages EACH developed.

```
SELECT PNAME, COUNT(*) AS NUM_PACKAGES
FROM SOFTWARE
GROUP BY PNAME;
```

```
-- Q148 : List all the details of programmers who have done a course at S.S.I.L.  
SELECT *  
FROM PROGRAMMER  
WHERE PNAME IN (  
    SELECT PNAME  
    FROM STUDIES  
    WHERE INSTITUTE = 'S.S.I.L.'  
);
```