

Assignments on SQL

This document contains few assignments on topics that covered under the SQL Course.

**Talent Nurturing – Aspire Systems India Private Limited, Chennai**

**Summary**

In this assignment, you are going to write SQL queries for each question that are given below. If you are having your own system you can install any database software like MySQL, MSSQL, Oracle, etc and create tables that are given below with the same structure. Sample data’s that are provided below for those who are not having their own PC in home, to understand what type of data’s will be there in the tables and you can insert the values as what you like. The purpose of this assignment is to assess your understanding of RDBMS concepts and how you are going to apply SQL Queries for each scenario that is given below.

# Table Structure

|  |  |  |  |
| --- | --- | --- | --- |
| 1) Table Name | Programmer |  | |
|  | | | |
| name | not null | varchar2(8) | name |
| dob | not null | date | date of birth |
| doj | not null | date | date of joining |
| sex | not null | varchar2(1) | male/ female |
| prof1 |  | varchar2(8) | known language 1 |
| prof2 |  | varchar2(8) | known language 2 |
| salary | not null | number(4) | salary |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sample Data | |  | | | | | |
|  | | | | | | | |
| somdutt | 21-Apr-66 | | 21-Apr-92 | m | pascal | basic | 3200 |

|  |  |  |  |
| --- | --- | --- | --- |
| 2) Table Name | Software |  | |
|  | | | |
| name | not null | varchar2(8) | name |
| title | not null | varchar2(20) | developed project name |
| dev\_in | not null | varchar2(8) | language developed |
| scost |  | number(7,2) | software cost |
| dcost |  | number(5) | development cost |
| sold |  | number(3) | number of software sold |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sample Data |  | | | | |
|  | | | | | |
| somdutt | parachutes | basic | 399.95 | 6000 | 43 |

2

|  |  |  |  |
| --- | --- | --- | --- |
| 3) Table Name | Studies |  | |
|  | | | |
| name | not null | varchar2(8) | name |
| splace | not null | varchar2(9) | studies place |
| course | not null | varchar2(5) | course studies |
| ccost | not null | varchar2(5) | course cost |

|  |  |  |  |
| --- | --- | --- | --- |
| Sample Data |  | | |
|  | | | |
| somdutt | sabhari | pgdca | 4500 |
| devdutt | bdps | dcs | 5000 |

**Creating Table and Inserting Values:**

CREATE TABLE Programmer (

name varchar(8) NOT NULL,

dob date NOT NULL,

doj date NOT NULL,

sex varchar(1) NOT NULL,

prof1 varchar(8),

prof2 varchar(8),

salary integer

);

CREATE TABLE Software (

xname varchar(8) NOT NULL,

title varchar(29) NOT NULL,

dev\_in varchar(8) NOT NULL,

scost float,

dcost float,

sold float

);

CREATE TABLE Studies (

name varchar(8) NOT NULL,

splace varchar(9) NOT NULL,

course varchar(5) NOT NULL,

ccost integer

);

INSERT INTO Programmer VALUES ('somdutt', '1965-04-21', '1992-04-21', 'm', 'java', 'dbase', 3250), ('devdutt', '1978-04-21', '1995-04-21', 'm', 'basic', 'c', 3250), ('caldutt', '1978-06-15', '2022-01-21', 'm', 'dbase', 'c', 3250), ('ramesh', '1988-06-10', '1993-05-21', 'm', 'c', 'pascal', 4250), ('rscah', '1970-07-21', '2021-09-21', 'm', 'cobol', 'pascal', 5250), ('calste', '1988-09-10', '2001-06-21', 'm', 'c', 'java', 3450), ('raname', '1998-11-21', '2020-07-21', 'f', 'clipper', 'pascal', 5250), ('rani', '1965-05-21', '2015-12-21', 'f', 'pascal', 'cobol', 3450), ('mary', '1999-11-21', '2021-11-21', 'f', 'c', 'python', 4250), ('raja', '1990-06-30', '2019-05-21', 'f', 'c++', 'cobol', 6450);

I

NSERT INTO Software VALUES ('somdutt', 'parachutes', 'cobol', 399.95, 6000.0, 43.0), ('devdutt', 'chaca', 'pascal', 199.95, 4000.0, 56.0), ('mary', 'dalce', 'c', 499.95, 4400.0, 8.0), ('somdutt', 'malda', 'dbase', 599.95, 5400.0, 35.0), ('caldutt', 'other', 'c', 99.95, 2700.0, 60.0), ('ramesh', 'rast', 'basic', 49.95, 800.0, 60.0), ('ramesh', 'odart', 'c', 79.95, 3400.0, 40.0), ('mary', 'cast', 'cobol', 75.95, 1400.0, 33.0), ('palead', 'dadr jas', 'java', 75.95, 1400.0, 0.0);

INSERT INTO Studies VALUES ('somdutt', 'sabhari', 'pgdca', 4500), ('devdutt', 'bdps', 'dcs', 6000), ('devdutt', 'sabhari', 'pgdca', 4500),('mary', 'bdps', 'pgdca', 6040), ('mary', 'soria', 'dcs', 5260);

**QUERIES - I**

1. **Find out the SELLING COST AVERAGE for the packages developed in PASCAL?**

SELECT AVG(scost) AS 'Selling cost'

FROM Software

WHERE dev\_in='pascal';

1. **Display the names and ages of all programmers.**

SELECT name,CAST(DATEDIFF(CURRENT\_TIMESTAMP, dob)/365.25 AS DECIMAL) AS Age

FROM Programmer;

1. **Display the names and ages of all the programmers who have undergone training in DCS course.**

SELECT Programmer.name, CAST(DATEDIFF(CURRENT\_TIMESTAMP, Programmer.dob)/365.25 AS DECIMAL) AS Age FROM Programmer JOIN Studies ON Programmer.name=Studies.name

WHERE Studies.course='dcs';

1. **What is the highest numbers of copies sold by a package?**

SELECT MAX(sold) AS 'Highest copies'

FROM Software;

1. **Display the names and date of birth of all the programmer born in JANUARY**.

SELECT name, dob

FROM Programmer

WHERE MONTH(dob) = 1;

1. **Display lowest course fee.**

SELECT MIN(ccost) AS 'course fee'

FROM Studies;

1. **How many programmer has done PGDCA course.**

SELECT COUNT(name) AS 'PGDCA course'

FROM Studies

WHERE course = 'pgdca';

1. **How much revenue has been earned through sales of packages in C.**

SELECT SUM(sold\*scost) AS 'C packages'

FROM Software

WHERE dev\_in='c';

1. **Display the details of software developed by Ramesh?**

SELECT \*

FROM Software

WHERE name='ramesh';

1. **How many programmers studied at SABHARI.**

SELECT COUNT(name) AS 'Programmers Studied '

FROM Studies

WHERE splace = 'sabhari';

1. **Display the details of PACKAGES whose sales crossed the 20000 mark.**

SELECT \* FROM Software

WHERE sold\*scost > 20000;

1. **Find out the number of copies which should be sold in order to recover the development cost of each package.**

SELECT name, CAST(dcost/scost AS DECIMAL) AS 'Development cost'

FROM Software;

1. **What is the price of the costliest software developed in BASIC?**

SELECT MAX(scost) AS 'Basic packages'

FROM Software

WHERE dev\_in = 'basic';

1. **Display the details of packages for which development cost has been recovered.**

SELECT \* FROM Software

WHERE dcost < sold\*scost;

1. **How many packages were developed in dbase?**

SELECT COUNT(name) AS 'Packages developed'

FROM Software

WHERE dev\_in = 'dbase';

1. **How many programmers studies at paragathi?**

SELECT COUNT(name) AS 'Programmers studied'

FROM Studies

WHERE splace = 'paragathi';

1. **How many programmers paid 5000 to 10000 for their course?**

SELECT COUNT(name) AS 'Programmers paid'

FROM Studies

WHERE ccost > 5000 AND ccost < 10000;

1. **What is the average course fee?**

SELECT AVG(ccost) AS 'Course fee'

FROM Studies;

1. **Display the details of programmers knowing c?**

SELECT \* FROM Programmer

WHERE prof1 = 'c' OR prof2 = 'c';

1. **How many programmers know either Cobol or Pascal?**

SELECT COUNT(name) AS 'Programmers'

FROM Programmer

WHERE prof1 = 'pascal' OR prof1 = 'cobol' OR prof2 = 'pascal' OR prof2 = 'cobol';

1. **How many programmers don't know Pascal & C?**

SELECT COUNT(name) AS 'Programmers'

FROM Programmer

WHERE prof1 <> 'pascal' AND prof1 <> 'c' OR prof2 <> 'pascal' AND prof2 <> 'c';

1. **How old is the oldest male programmers?**

SELECT MAX(CAST(DATEDIFF(CURRENT\_TIMESTAMP, dob)/365.25 AS DECIMAL))

AS 'Oldest Programmers' FROM Programmer WHERE sex = 'm';

1. **What is the average age of female programmers?**

SELECT AVG(CAST(DATEDIFF(CURRENT\_TIMESTAMP, dob)/365.25 AS DECIMAL))

AS 'Average age' FROM Programmer WHERE sex = 'f';

1. **Calculate the experience in years for each programmers and display along with the names in descending order?**

SELECT name, CAST(DATEDIFF(CURRENT\_TIMESTAMP, doj)/365.25 AS DECIMAL) AS 'Experience' FROM Programmer ORDER BY CAST(DATEDIFF(CURRENT\_TIMESTAMP, doj)/365.25 AS DECIMAL) DESC;

1. **Who are the programmers who celebrate their birthday during the current month?**

SELECT name, dob FROM Programmer

WHERE MONTH(CURRENT\_TIMESTAMP) = MONTH(dob);

1. **How many female programmers are there?**

SELECT COUNT(xname) AS 'Female Programmers'

FROM Programmer WHERE sex = 'f';

1. **What are the languages known by the male programmers?**

SELECT prof1 AS 'Languages Known'

FROM Programmer WHERE sex = 'm' UNION SELECT prof2

FROM Programmer WHERE sex = 'm'

GROUP BY prof1, prof2;

1. **What is the Average salary?**

SELECT AVG(salary) AS 'Average salary'

FROM Programmer;

1. **How many people draw 2000 to 4000?**

SELECT COUNT(salary) AS 'People draw'

FROM Programmer WHERE salary >= 2000 AND salary <= 4000;

1. **Display the details of those who don't know Clipper, Cobol or Pascal?**

SELECT \* FROM Programmer

WHERE prof1 <> 'cobol' AND prof1 <> 'pascal' AND prof1 <> 'clipper' AND prof2 <> 'cobol'

AND prof2 <> 'pascal' AND prof2 <> 'clipper';

1. **How many Female programmers knowing C are above 24 years of age?**

SELECT COUNT(name) AS 'Female programmers' FROM Programmer WHERE sex = 'f' AND CAST(DATEDIFF(CURRENT\_TIMESTAMP, dob)/365.25 AS DECIMAL) > 24;

1. **Who are the programmers who will be celebrating their Birthday within a week?**

SELECT name FROM Programmer

WHERE MONTH(CURRENT\_TIMESTAMP) = MONTH(dob)

AND DAY(CURRENT\_TIMESTAMP)- DAY(dob) < 7

AND DAY(CURRENT\_TIMESTAMP)-DAY(dob) > -7;

1. **) Display the details of those with less than a year's experience?**

SELECT \* FROM Programmer

WHERE DATEDIFF(CURRENT\_TIMESTAMP, doj) < 365;

1. **) Display the details of those who will be completing 2 years of service this year?**

SELECT \* FROM Programmer

WHERE DATEDIFF(CURRENT\_TIMESTAMP, doj) < 365\*2

AND DATEDIFF(CURRENT\_TIMESTAMP, doj) > 365;

1. **) Calculate the amount to be recovered for those packages whose development cost has not been recovered?**

SELECT name, dcost-scost\*sold AS 'Amount Left'

FROM Software WHERE scost\*sold < dcost;

1. **List the packages which have not been sold so far?**

SELECT title FROM Software WHERE sold = 0;

1. **Find out the cost of the software developed by Mary?**

SELECT SUM(dcost) AS 'Cost' FROM Software

WHERE name = 'mary';

1. **Display the institute’s names from the studies table without duplicates?**

SELECT name FROM Studies GROUP BY name;

1. **How many different courses are mentioned in the studies table?**

SELECT COUNT(distinct course) AS 'Study table'

FROM Studies;

1. **Display the names of the programmers whose names contain 2 occurrences of the letter A?**

SELECT name FROM Programmer

WHERE LENGTH(name) - LENGTH(REPLACE(name, 'a', '')) = 2;

1. **Display the names of programmers whose names contain unto 5 characters?**

SELECT name FROM Programmer WHERE LENGTH(name) >= 5;

1. **How many female programmers knowing COBOL have more than 2 years experience?**

SELECT COUNT(name) AS 'Female programmers'

FROM Programmer WHERE prof1 = 'cobol' OR prof2 = 'cobol' AND CAST(DATEDIFF(CURRENT\_TIMESTAMP, doj)/365.25 AS DECIMAL) > 2;

1. **What is the length of the shortest name in the programmer table?**

SELECT MIN(LENGTH(name)) AS 'Lenght' FROM Programmer;

1. **What is the average development cost of a package developed in COBOL?**

SELECT AVG(dcost) AS 'Average cost' FROM Software

WHERE dev\_in = 'cobol';

1. **Display the name, sex, dob (DD/MM/YY format), doj for all the programmers without using conversion function?**

SELECT name, DATE\_FORMAT(dob,"%d/%m/%Y") as 'dob', doj FROM Programmer;

1. **Who are the programmers who were born on the last day of the month?**

SELECT name FROM Programmer

WHERE (MONTH(dob) IN (1,3,5,7,8,10,12) AND DAY(dob) = 31) OR (MONTH(dob) IN (4,6,9,11)

AND DAY(dob) = 30) OR (MONTH(dob) IN (2) AND DAY(dob) IN (28,29));

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

SELECT salary FROM Programmer

WHERE prof1 <> 'cobol' AND prof2 <> 'cobol';

1. **Display the title, scost, dcost and difference between scost and dcost in descending order of difference?**

SELECT title, scost, dcost, dcost-scost FROM Software ORDER BY dcost-scost DESC;

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

SELECT name, dob, doj FROM Programmer

WHERE MONTH(dob) = MONTH(doj);

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

SELECT title FROM Software

WHERE (LENGTH(title) - LENGTH(REPLACE(title, ' ', ''))) > 0;

# QUERIES - II

1. **Display THE NUMBER OF packages developed in EACH language.**

SELECT dev\_in, COUNT(name) AS 'Packages'

FROM Software GROUP BY dev\_in;

1. **Display THE NUMBER OF packages developed by EACH person.**

SELECT name, COUNT(name) AS 'Packages'

FROM Software GROUP BY name;

1. **Display THE NUMBER OF male and female programmer.**

SELECT sex, COUNT(name) AS 'Programmers'

FROM Programmer GROUP BY sex;

1. **Display THE COSTLIEST packages and HIGEST selling developed in EACH language.**

SELECT dev\_in, MAX(dcost) AS 'Cost', MAX(sold) AS 'Selling package'

FROM Software GROUP BY dev\_in;

1. **Display THE NUMBER OF people BORN in EACH YEAR.**

SELECT YEAR(dob) AS 'Year', COUNT(name) AS 'People'

FROM Programmer GROUP BY YEAR(dob);

1. **Display THE NUMBER OF people JOINED in EACH YEAR.**

SELECT YEAR(doj) AS 'Year',COUNT(name) AS 'People'

FROM Programmer GROUP BY YEAR(doj);

1. **Display THE NUMBER OF people BORN in EACH MONTH.**

SELECT MONTH(dob) AS 'Month',COUNT(name) AS 'People'

FROM Programmer GROUP BY MONTH(dob);

1. **Display THE NUMBER OF people JOINED in EACH MONTH.**

SELECT MONTH(doj) AS 'Month',COUNT(name) AS 'People'

FROM Programmer GROUP BY MONTH(doj);

1. **Display the language wise COUNTS of prof1.**

SELECT prof1, COUNT(name) AS 'Count'

FROM Programmer GROUP BY prof1;

1. **Display the language wise COUNTS of prof2.**

SELECT prof2, COUNT(name) AS 'Count'

FROM Programmer GROUP BY prof2;

1. **Display THE NUMBER OF people in EACH salary group.**

SELECT salary, COUNT(name) AS 'Count'

FROM Programmer GROUP BY salary;

1. **Display THE NUMBER OF people who studied in EACH institute.**

SELECT splace, COUNT(name) AS 'Count'

FROM Studies GROUP BY splace;

1. **Display THE NUMBER OF people who studied in EACH course.**

SELECT course, COUNT(name) AS 'Count'

FROM Studies GROUP BY course;

1. **Display the TOTAL development COST of the packages developed in EACH language.**

SELECT dev\_in, SUM(dcost) AS 'Cost'

FROM Software GROUP BY dev\_in;

1. **Display the selling cost of the package developed in EACH language.**

SELECT dev\_in, SUM(scost) AS 'Selling prices'

FROM Software GROUP BY dev\_in;

1. **Display the cost of the package developed by EACH programmer.**

SELECT name, SUM(dcost) AS 'Development cost'

FROM Software GROUP BY name;

1. **Display the sales values of the package developed in EACH programmer.**

SELECT name, SUM(dcost\*sold) AS 'Sales'

FROM Software GROUP BY name;

1. **Display the NUMBER of packages developed by EACH programmer.**

SELECT name, COUNT(title) AS 'Packages'

FROM Software GROUP BY name;

1. **Display the sales COST of packages developed by EACH programmer language wise.**

SELECT name, dev\_in, SUM(dcost) AS 'Cost'

FROM Software GROUP BY name, dev\_in ORDER BY name;

1. **Display EACH programmers name, costliest package and cheapest packages developed by Him/Her.**

SELECT name, MAX(dcost) AS 'Cost', MIN(dcost) AS 'Cheapest package'

FROM Software GROUP BY name;

1. **Display EACH language name with AVERAGE development cost, AVERAGE cost, selling cost and AVERAGE price per copy.**

SELECT dev\_in, AVG(dcost) AS 'Development', AVG(scost) AS 'Selling'

FROM Software GROUP BY dev\_in;

1. **Display EACH institute name with NUMBER of courses, AVERAGE cost per course.**

4

SELECT course, COUNT(distinct splace) AS 'Courses', AVG(ccost) AS 'Average cost'

FROM Studies GROUP BY course;

1. **Display EACH institute name with NUMBER of students.**

SELECT splace, COUNT(name) AS 'Students'

FROM Studies GROUP BY splace;

1. **Display names of male and female programmers.**

SELECT name, sex FROM Programmer ORDER BY sex;

1. **Display the programmer's name and their packages.**

SELECT name, title FROM Software ORDER BY name;

1. **Display the NUMBER of packages in EACH language.**

SELECT dev\_in, COUNT(title) AS 'Number of packages'

FROM Software GROUP BY dev\_in;

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

SELECT dev\_in, COUNT(title) AS 'Number of packages'

FROM Software WHERE dcost < 1000 GROUP BY dev\_in;

1. **Display the AVERAGE difference BETWEEN scost and dcost for EACH language.**

SELECT dev\_in, AVG(dcost - scost) AS 'Average'

FROM Software GROUP BY dev\_in;

1. **Display the TOTAL scost, dcsot and amount TOBE recovered for EACH programmer for whose dcost HAS NOT YET BEEN recovered.**

SELECT name, SUM(scost) AS 'selling', SUM(dcost) AS 'Cost',

SUM(dcost - sold\*scost) 'Left amount' FROM Software WHERE dcost > sold\*scost

GROUP BY name;

1. **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 - III

1. **Who is the highest paid C programmer?**

SELECT name AS 'C Programmer'

FROM Programmer WHERE salary IN (

SELECT MAX(salary)

FROM Programmer

WHERE prof1 = 'c' OR prof2 = 'c'

);

1. **Who is the highest paid female cobol programmer?**

SELECT name AS 'Cobol Programmer'

FROM Programmer WHERE salary IN (

SELECT MAX(salary)

FROM Programmer

WHERE prof1 = 'cobol' OR prof2 = 'cobol' AND sex = 'f'

);

1. **Display the name of the HIGEST paid programmer for EACH language (prof1)**

SELECT p1.name, p1.prof1, p1.salary

FROM Programmer p1

LEFT JOIN Programmer p2

ON p1.prof1 = p2.prof1 AND p1.salary < p2.salary

WHERE p2.name IS NULL

ORDER BY p1.prof1;

1. **Who is the LEAST experienced programmer?**

SELECT name FROM Programmer

WHERE ROUND(DATEDIFF(CURRENT\_TIMESTAMP, doj)/365.25)

IN (SELECT ROUND(MIN(DATEDIFF(CURRENT\_TIMESTAMP, doj)/365.25)) FROM Programmer);

1. **Who is the MOST experienced programmer?**

SELECT name FROM Programmer

WHERE ROUND(DATEDIFF(CURRENT\_TIMESTAMP, doj)/365.25)

IN (SELECT ROUND(MAX(DATEDIFF(CURRENT\_TIMESTAMP, doj)/365.25)) FROM Programmer);

1. **Which language is known by ONLY ONE programmer?**

WITH result AS (

SELECT prof1 'pro'

FROM Programmer

UNION ALL

SELECT prof2

FROM Programmer

)

SELECT pro

FROM

GROUP BY pro

HAVING COUNT(pro) = 1;

1. **Who is the YONGEST programmer knowing DBASE?**

SELECT name FROM Programmer

WHERE ROUND(DATEDIFF(CURRENT\_TIMESTAMP, dob))

IN (SELECT ROUND(MIN(DATEDIFF(CURRENT\_TIMESTAMP, dob)))

FROM Programmer

WHERE prof1 = 'dbase' OR prof2 = 'dbase');

1. **Which institute has MOST NUMBER of students?**

WITH result AS (

SELECT splace, COUNT(course) AS 'course\_c'

FROM Studies

GROUP BY splace

)

SELECT splace

FROM result

WHERE course\_c IN (

SELECT MAX(course\_c) FROM result);

1. **Who is the above programmer?**

SELECT name FROM Programmer LIMIT 1;

1. **Which female programmer earns MORE than 3000/- but DOES NOT know C, C++, Oracle or Dbase?**

SELECT name FROM Programmer

WHERE sex = 'f' AND salary > 3000 AND prof1 NOT IN ('c','c++','oracle','dbase')

AND prof2 NOT IN ('c','c++','oracle','dbase');

1. **Which is the COSTLIEST course?**

WITH result AS (

SELECT course, MAX(ccost) AS 'course\_cost'

FROM Studies

GROUP BY course

)

SELECT course

FROM result

WHERE course\_cost IN (

SELECT MAX(course\_cost) FROM result);

1. **Which course has been done by MOST of the students?**

WITH result AS (

SELECT course, COUNT(xname) AS 'student\_c'

FROM Studies

GROUP BY course

)

SELECT course

FROM result

WHERE student\_c IN (

SELECT MAX(student\_c) FROM result);

1. **Display name of the institute and course Which has below AVERAGE course fee?**

SELECT DISTINCT splace, course

FROM Studies

WHERE ccost < (SELECT AVG(ccost) FROM Studies);

1. **Which institute conducts COSTLIEST course?**

WITH result AS (

SELECT course, MAX(ccost) AS 'course\_cost'

FROM Studies

GROUP BY course

)

SELECT DISTINCT splace

FROM Studies

WHERE ccost IN (

SELECT MAX(course\_cost) FROM result);

1. **Which course has below AVERAGE number of students?**

WITH result AS (

SELECT course, COUNT(name) AS 'student\_c'

FROM Studies

GROUP BY course

)

SELECT course

FROM result

WHERE student\_c < (

SELECT AVG(student\_c) FROM );

1. **Which institute conducts the above course?**

SELECT splace FROM Studies LIMIT 1;

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

SELECT course AS 'Average cost' FROM Studies

WHERE ccost BETWEEN (SELECT AVG(ccost) FROM Studies)+1000 AND (SELECT AVG(ccost)

FROM Studies)-1000;

1. **Which package has the HIGEST development cost?**

SELECT title FROM Software

WHERE dcost IN (SELECT MAX(dcost) FROM Software);

1. **Which package has the LOWEST selling cost?**

SELECT title FROM Software

WHERE scost IN (SELECT MIN(scost) FROM Software);

1. **Who developed the package, which has sold the LEAST number of copies?**

SELECT name FROM Software

WHERE sold IN (SELECT MIN(sold) FROM Software);

1. **Which language was used to develop the package WHICH has the HIGEST sales amount?**

SELECT dev\_in FROM Software

WHERE sold IN (SELECT MAX(sold) FROM Software);

1. **How many copies of the package that has the LEAST DIFFRENCE between development and selling cost were sold?**

SELECT sold FROM Software WHERE ABS(dcost-sold\*scost)

IN (SELECT MIN(ABS(dcost-sold\*scost)) FROM Software);

1. **Which is the COSTLIEAST package developed in PASCAL?**

SELECT title FROM Software

WHERE dcost IN (SELECT MAX(dcost) FROM Software WHERE dev\_in = 'pascal');

1. **Which language was used to develop the MOST NUMBER of package?**

WITH result AS (

SELECT dev\_in, COUNT(title) AS 'pack\_count'

FROM Software

GROUP BY dev\_in

)

SELECT dev\_in

FROM result

WHERE pack\_count IN (

SELECT MAX(pack\_count) FROM result);

1. **Which programmer has developed the HIGEST NUMBER of package?**

WITH result AS (

SELECT name, COUNT(title) AS 'pack\_count'

FROM Software

GROUP BY name

)

SELECT name

FROM result

WHERE pack\_count IN (

SELECT MAX(pack\_count) FROM result);

1. **Who is the author of the COSTLIEST package?**

SELECT name FROM Software

WHERE dcost IN (SELECT MAX(dcost) FROM Software);

1. **Display names of packages WHICH have been sold LESS THAN the AVERAGE number of copies?**

SELECT COUNT(title) AS 'Average' FROM Software

WHERE sold < (SELECT AVG(sold) FROM Software);

1. **Who are the female programmers earning MORE than the HIGEST paid male programmers?**

SELECT name FROM Programmer

WHERE salary > (SELECT MAX(salary) FROM Programmer WHERE sex = 'm');

1. **Which language has been stated as prof1 by MOST of the programmers?**

WITH result AS (

SELECT prof1, COUNT(name) AS 'pro\_count'

FROM Programmer

GROUP BY prof1

)

SELECT prof1

FROM result

WHERE pro\_count IN (

SELECT MAX(pro\_count) FROM result);

1. **Who are the authors of packages, WHICH have recovered MORE THAN double the development cost?**

SELECT DISTINCT name FROM Software

WHERE sold\*scost > 2\*dcost;

1. **Display programmer names and CHEAPEST package developed by them in EACH language?**
2. **Who is the YOUNGEST male programmer born in 1965?**

SELECT name FROM Programmer

WHERE DATEDIFF(CURRENT\_TIMESTAMP, dob) IN (SELECT MIN(DATEDIFF(CURRENT\_TIMESTAMP, dob)) FROM Programmer WHERE YEAR(dob) = 1965);

1. **Display language used by EACH programmer to develop the HIGEST selling and LOWEST selling package.**

1. **Who is the OLDEST female programmer WHO joined in 1992?**

SELECT name FROM Programmer

WHERE DATEDIFF(CURRENT\_TIMESTAMP, dob) IN (SELECT MAX(DATEDIFF(CURRENT\_TIMESTAMP, dob))

FROM Programmer WHERE YEAR(dob) = 1992 AND sex = 'f');

1. **In WHICH year where the MOST NUMBER of programmer born?**

WITH result AS (

SELECT YEAR(dob) AS 'years', COUNT(name) AS 'counts'

FROM Programmer

GROUP BY YEAR(dob)

)

SELECT years

FROM result

WHERE counts IN (SELECT MAX(counts) FROM result);

1. **In WHICH month did MOST NUMBRER of programmer join?**

WITH resut AS (

SELECT MONTH(doj) AS 'months', COUNT(name) AS 'counts'

FROM Programmer

GROUP BY MONTH(doj)

)

SELECT months

FROM result

WHERE counts IN (SELECT MAX(counts) FROM result);

1. **In WHICH language are MOST of the programmer's proficient?**

SELECT Prof1 AS Language, COUNT(\*)

AS ProgrammerCount FROM Programmer GROUP BY Prof1

ORDER BY ProgrammerCount DESC LIMIT 1;

1. **Who are the male programmers earning BELOW the AVERAGE salary of female programmers?**

SELECT name FROM Programmer

WHERE salary < (SELECT AVG(salary)

FROM Programmer WHERE sex = 'f' AND sex = 'm';

# QUERY - IV

1. **Display the details of THOSE WHO are drawing the same salary.**

WITH repeats AS (

SELECT salary, COUNT(name) AS 'counts'

FROM Programmer

GROUP BY salary

HAVING COUNT(name) > 1

)

SELECT name, dob, doj, sex, prof1, prof2, salary

FROM Programmer

WHERE salary IN (SELECT salary FROM repeats);

1. **Display the details of software developed by male programmers earning MORE than 3000.**

SELECT p1.name, p1.title, p1.dev\_in, p1.scost, p1.dcost, p1.sold

FROM Software p1

LEFT JOIN Programmer p2

ON p1.name = p2.name

WHERE p2.salary > 3000;

1. **Display details of packages developed in PASCAL by female programmers.**

SELECT p1.name, p1.title, p1.dev\_in, p1.scost, p1.dcost, p1.sold

FROM Software p1

LEFT JOIN Programmer p2

ON p1.name = p2.name

WHERE p1.dev\_in = 'pascal' AND p2.sex = 'f';

1. **Display the details of the programmer WHO joined BEFORE 1990.**

SELECT name, dob, doj, sex, prof1, prof2, salary

FROM Programmer

WHERE YEAR(doj) < 1990;

1. **Display details of software developed in C by female programmers of PRAGATHI.**

WITH results AS (

SELECT p1.name, p1.title, p1.dev\_in, p1.scost, p1.dcost, p1.sold, p2.sex

FROM Software p1

LEFT JOIN Programmer p2

ON p1.name = p2.name

)

SELECT p1.name, p1.title, p1.dev\_in, p1.scost, p1.dcost, p1.sold, p1.sex

FROM results p1

LEFT JOIN Studies p2

ON p1.name = p2.name

WHERE p1.sex = 'f' AND p1.dev\_in = 'c' AND p2.splace = 'praghati';

1. **Display NUMBER of packages NUMBER of copies sold and sales value of EACH programmer Institute-wise.**
2. **Display details of software developed in DBASE by male programmers WHO belong to the institute on which MOST NUMBER OF programmer’s studies.**

WITH results AS (

SELECT p1.name, p1.title, p1.dev\_in, p1.scost, p1.dcost, p1.sold, p2.sex

FROM Software p1

LEFT JOIN Programmer p2

ON p1.name = p2.name

), num\_stu AS (

SELECT splace, COUNT(name) AS 'counts'

FROM Studies

GROUP BY splace

), place\_mx AS (

SELECT splace

FROM num\_stu

WHERE counts IN (SELECT MAX(counts) FROM num\_stu)

)

SELECT p1.name, p1.title, p1.dev\_in, p1.scost, p1.dcost, p1.sold

FROM results p1

LEFT JOIN Studies p2

ON p1.name = p2.name

WHERE p1.sex = 'f' AND p1.dev\_in = 'c' AND p2.splace IN (SELECT splace FROM place\_mx);

1. **Display the details of the software that was developed by male programmers born BEFORE 1965 and female programmers born AFTER 1975.**

SELECT p1.name, p1.title, p1.dev\_in, p1.scost, p1.dcost, p1.sold

FROM Software p1

LEFT JOIN Programmer p2

ON p1.name = p2.name

WHERE p2.sex = 'm' AND YEAR(dob) < 1965 OR p2.sex = 'f' AND YEAR(dob) < 1975;

1. **Display the details of the software that was developed in the language that is NOT the programmer’s first proficiency.**

SELECT p1.name, p1.title, p1.dev\_in, p1.scost, p1.dcost, p1.sold

FROM Software p1

LEFT JOIN Programmer p2

ON p1.name = p2.name

WHERE p1.dev\_in <> p2.prof1;

1. **Display details of software that was developed in the language which is NEITHER first NOR second proficiency of the programmer.**

SELECT p1.name, p1.title, p1.dev\_in, p1.scost, p1.dcost, p1.sold

FROM Software p1

LEFT JOIN Programmer p2

ON p1.name = p2.name

WHERE p1.dev\_in <> p2.prof1 AND p1.dev\_in <> p2.prof2;

1. **Display details of software developed by male students of SABHARI.**

WITH result AS (

SELECT p1.name, p1.title, p1.dev\_in, p1.scost, p1.dcost, p1.sold, p2.sex

FROM Software p1

LEFT JOIN Programmer p2

ON p1.name = p2.name

)

SELECT p1.name, p1.title, p1.dev\_in, p1.scost, p1.dcost, p1.sold, p2.splace

FROM result p1

LEFT JOIN Studies p2

ON p1.name = p2.name

WHERE p1.sex = 'm' AND p2.splace = 'sabhari';

1. **Display the names of programmers WHO HAVE NOT developed any package.**

SELECT p1.name, p1.dob, p1.doj, p1.sex, p1.prof1, p1.prof2, p1.salary

FROM Programmer p1

LEFT JOIN Software p2

ON p1.name = p2.name

WHERE p2.title IS NULL;

1. **What is the total cost of the software developed by the programmers by APPLE?**

SELECT SUM(dcost) AS 'Total cost'

FROM Software

WHERE dev\_in = 'apple';

1. **Who are the programmers WHO JOINED in the same day?**

WITH result AS (

SELECT COUNT(name) AS 'total', DATEDIFF(CURRENT\_TIMESTAMP, doj) AS 'days'

FROM Programmer

GROUP BY DATEDIFF(CURRENT\_TIMESTAMP, doj)

)

SELECT name FROM Programmer

WHERE DATEDIFF(CURRENT\_TIMESTAMP, doj) IN (SELECT days FROM result

WHERE total > 1);

1. **Who are the programmers WHO HAVE THE SAME PROF2?**

WITH repeats AS (

SELECT prof2, COUNT(name) AS 'counts'

FROM Programmer GROUP BY prof2

HAVING COUNT(name) > 1

)

SELECT name, dob, doj, sex, prof1, prof2, salary

FROM Programmer

WHERE prof2 IN (SELECT prof2 FROM repeats);

1. **Display the total sales values of software, institutes-wise.**

SELECT p2.splace, SUM(p1.dcost) AS 'Total cost'

FROM Software p1 LEFT JOIN Studies p2

ON p1.name = p2.name GROUP BY p2.splace;

1. **In which institutes did the person who developed the COSTLIEST package study?**

SELECT p1.splace, p2.name

FROM Studies p1 RIGHT JOIN Software p2

ON p1.name = p2.name

WHERE p2.dcost IN (SELECT MAX(dcost) FROM Software);

1. **Which language listed in prof1 and prof2 HAS NOT BEEN used to develop any package?**

WITH result AS (

SELECT prof1 AS 'pro', name FROM Programmer

UNION ALL

SELECT prof2, name FROM Programmer

)

SELECT DISTINCT pro FROM result

WHERE pro NOT IN (SELECT dev\_in FROM Software);

1. **How much does the person WHO developed the HIGHEST selling package earn and WHAT course did he/she undergo?**

WITH results AS (

SELECT p1.name, p1.title, p1.dev\_in, p1.scost, p1.dcost, p1.sold, p2.salary

FROM Software p1 LEFT JOIN Programmer p2

ON p1.name = p2.name

)

SELECT p1.salary, p2.splace FROM results p1

LEFT JOIN Studies p2 ON p1.name = p2.name

WHERE p1.dcost IN (SELECT MAX(dcost) FROM Software);

1. **How many months will it take for each programmer to recover the cost of the course underwent?**

SELECT p2.ccost/p1.salary AS 'Courses'

FROM Programmer p1 LEFT JOIN Studies p2

ON p1.name = p2.name

WHERE p2.course IS NOT NULL;

1. **Which is the COSTLIEST package developed by a person with under 3 year’s expenences?**

SELECT p1.title FROM Software p1

LEFT JOIN Programmer p2 ON p1.name = p2.name

WHERE DATEDIFF(CURRENT\_TIMESTAMP, p2.doj)/365.25 < 3 AND p1.dcost

IN (SELECT MAX(dcost) FROM Software);

1. **What is the AVERAGE salary for those WHOSE software's sales value is more than 50,000?**

SELECT AVG(p2.salary) AS 'Average'

FROM Software p1 LEFT JOIN Programmer p2

ON p1.name = p2.name

WHERE p1.sold\*p1.scost > 50000;

1. **How many packages were developed by the students WHO studied in the institute that Charge the LOWEST course fee?**

WITH result AS (

SELECT splace, MIN(ccost) AS 'mincost'

FROM Studies GROUP BY splace

)

SELECT COUNT(p1.title) AS 'Packages'

FROM Software p1 LEFT JOIN Studies p2

ON p1.name = p2.name

WHERE p2.ccost IN (SELECT MIN(mincost) FROM result);

1. **How many packages were developed by the person WHO developed the CHEAPEST package? Where did he\she study?**

WITH cour AS (

SELECT splace, MIN(ccost) AS 'mincost'

FROM Studies GROUP BY splace

)

SELECT COUNT(p1.title) AS 'Packages'

FROM Software p1 LEFT JOIN Studies p2

ON p1.name = p2.name

WHERE p1.dcost IN (SELECT MIN(dcost) FROM Software);

1. **How many packages were developed by female programmers earning MORE than the HIGHEST paid male programmer?**

SELECT p1.x=name, COUNT(p1.title) AS 'Pack'

FROM Software p1 LEFT JOIN Programmer p2

ON p1.name = p2.name WHERE p2.sex = 'f'

AND p2.salary > (SELECT MAX(salary) FROM Programmer WHERE sex = 'm')

GROUP BY p1.name;

1. **How many packages were developed by the MOST experienced programmers from BDPS?**

WITH result AS (

SELECT p1.xname, p1.title, p1.dev\_in, p1.scost, p1.dcost, p1.sold, p2.doj

FROM Software p1 LEFT JOIN Programmer p2

ON p1.name = p2.name

)

SELECT p1.name, COUNT(p1.title) AS 'pack'

FROM result p1

LEFT JOIN Studies p2 ON p1.name = p2.name

WHERE DATEDIFF(CURRENT\_TIMESTAMP, p1.doj) IN (SELECT MAX(DATEDIFF(CURRENT\_TIMESTAMP, doj)) FROM Programmer) AND p2.splace = 'bdps'

GROUP BY p1.name;

1. **List the programmers (from software table) and institutes they studied, including those WHO DIDN'T develop any package.**

WITH result AS (

SELECT p2.name, p1.title, p1.dev\_in, p1.scost, p1.dcost, p1.sold, p2.doj

FROM Software p1

RIGHT JOIN Programmer p2

ON p1.name = p2.name

)

SELECT DISTINCT p1.name, p2.splace

FROM result p1

LEFT JOIN Studies p2

ON p1.name = p2.name;

1. **List each profit with the number of programmers having that prof1 and the number of packages developed in that prof1.**

WITH result AS (

SELECT prof1, COUNT(name) AS 'counts'

FROM Programmer GROUP BY prof1), num\_pack AS (

SELECT dev\_in, COUNT(title) AS 'counts'

FROM SoftwareGROUP BY dev\_in)

SELECT p1.prof1, p1.counts AS 'programmers', p2.counts AS 'Packages'

FROM result p1

LEFT JOIN num\_pack p2 ON p1.prof1 = p2.dev\_in;

1. **List programmer names (from programmer table) and number of packages EACH developed.**

SELECT p1.name, COUNT(p2.title) AS 'Packages'

FROM Programmer p1 LEFT JOIN Software p2 ON p1.name = p2.name

GROUP BY p1.name;

1. **List all the details of programmers who have done a course at S.S.I.L.**

SELECT p1.name, p1.dob, p1.doj, p1.sex, p1.prof1, p1.prof2, p1.salary

FROM Programmer p1 LEFT JOIN Studies p2

ON p1.name = p2.name WHERE p2.splace = 'ssil';