1. SELECT AVG(SCOST) AS AverageCost

FROM software

WHERE DEVELOPIN = 'PASCAL';

1. SELECT PNAME, DATEDIFF(YEAR, DOB, GETDATE()) AS Age

FROM programmer;

1. SELECT PNAME

FROM studies

WHERE COURSE = 'DAP';

1. SELECT PNAME, DOB

FROM programmer

WHERE MONTH(DOB) = 1;

1. SELECT MAX(SOLD)

FROM software;

1. SELECT MIN(COURSE\_FEE)

FROM studies;

1. SELECT COUNT(\*)

FROM studies

WHERE COURSE = 'PGDCA';

1. SELECT SUM(SCOST \* SOLD) AS TotalRevenue

FROM software

WHERE DEVELOPIN = 'C';

1. SELECT \*

FROM software

WHERE PNAME = 'RAMESH';

1. SELECT pname

FROM studies

WHERE INSTITUTE = 'SABHARI'

1. SELECT (SCOST \* SOLD) AS TotalRevenue

FROM software

WHERE (SCOST \* SOLD) > 2000

1. SELECT \*

FROM software

WHERE SOLD \* SCOST >= DCOST;

1. SELECT MAX(DCOST)

FROM software

WHERE DEVELOPIN = 'BASIC';

1. SELECT COUNT(\*)

FROM software

WHERE DEVELOPIN = 'DBASE';

1. SELECT pname

FROM studies

WHERE INSTITUTE = ' PRAGATHI'

1. SELECT COUNT(\*)

FROM studies

WHERE COURSE\_FEE BETWEEN 5000 AND 10000;

1. SELECT AVG(COURSE\_FEE)

FROM studies;

1. SELECT \*

FROM programmer

WHERE PROF1 = 'C' OR PROF2 = 'C';

1. SELECT COUNT(\*)

FROM programmer

WHERE PROF1 = 'COBOL' OR PROF2 = 'COBOL' OR PROF1 = 'PASCAL' OR PROF2 = 'PASCAL';

1. SELECT COUNT(\*)

FROM programmer

WHERE NOT (PROF1 = 'PASCAL' OR PROF2 = 'PASCAL') AND NOT (PROF1 = 'C' OR PROF2 = 'C');

1. SELECT MAX(DATEDIFF(YEAR, DOB, GETDATE()))

FROM programmer

WHERE GENDER = 'M'

1. SELECT AVG(DATEDIFF(YEAR, DOB, GETDATE()))

FROM programmer

WHERE GENDER = 'F';

1. SELECT PNAME, DATEDIFF(YEAR, DOJ, GETDATE()) AS Experience

FROM programmer

ORDER BY Experience DESC;

1. SELECT PNAME, DOB

FROM programmer

WHERE MONTH(DOB) = MONTH(GETDATE());

1. SELECT COUNT(\*)

FROM programmer

WHERE GENDER = 'F';

1. SELECT DISTINCT PROF1, PROF2

FROM programmer

WHERE GENDER = 'M';

1. SELECT AVG(SALARY)

FROM programmer;

1. SELECT COUNT(\*)

FROM programmer

WHERE SALARY BETWEEN 2000 AND 4000;

1. SELECT \*

FROM programmer

WHERE PROF1 NOT IN ('CLIPPER', 'COBOL', 'PASCAL') AND PROF2 NOT IN ('CLIPPER', 'COBOL', 'PASCAL');

1. SELECT PNAME, DCOST

FROM software

1. SELECT PNAME, SCOST \* SOLD

FROM software

1. SELECT PNAME, SOLD

FROM software

1. SELECT PNAME, DEVELOPIN, SUM(SCOST)

FROM software

GROUP BY PNAME, DEVELOPIN;

1. SELECT DEVELOPIN,

AVG(DCOST),

AVG(SCOST),

AVG(SCOST) / NULLIF(AVG(SOLD), 0)

FROM software

GROUP BY DEVELOPIN;

1. SELECT PNAME,

MAX(DCOST),

MIN(DCOST)

FROM software

GROUP BY PNAME;

1. SELECT INSTITUTE,

COUNT(\*) AS NumCourses,

AVG(COURSE\_FEE) AS AvgCostPerCourse

FROM studies

GROUP BY INSTITUTE;

1. SELECT INSTITUTE,COUNT(PNAME)

FROM studies

GROUP BY INSTITUTE

1. SELECT PNAME, GENDER

FROM programmer;

1. SELECT PNAME, TITLE

FROM software;

1. SELECT DEVELOPIN, COUNT(\*)

FROM software

WHERE DEVELOPIN NOT IN ('C', 'CPP')

GROUP BY DEVELOPIN;

1. SELECT DEVELOPIN, COUNT(\*)

FROM software

WHERE DCOST < 1000

GROUP BY DEVELOPIN;

1. SELECT TITLE, AVG(SCOST - DCOST)

FROM software

GROUP BY TITLE;

1. SELECT SCOST, DCOST, (DCOST-SCOST)

FROM software

where (DCOST-SCOST) < 0

1. SELECT MAX(SALARY),

MIN(SALARY),

AVG(SALARY)

FROM programmer

WHERE SALARY > 2000;

1. SELECT TOP 1 PNAME, salary

FROM programmer

WHERE PROF1 = 'C' OR PROF2 = 'C'

ORDER BY SALARY DESC;

1. SELECT TOP 1 PNAME, salary

FROM programmer

WHERE GENDER = 'F' AND (PROF1 = 'COBOL' OR PROF2 = 'COBOL')

ORDER BY SALARY DESC;

1. SELECT ProgrammingLanguage, MAX(MaxSalary) AS MaxSalary

FROM (

SELECT PROF1 AS ProgrammingLanguage, MAX(SALARY) AS MaxSalary

FROM programmer

GROUP BY PROF1

UNION

SELECT PROF2 AS ProgrammingLanguage, MAX(SALARY) AS MaxSalary

FROM programmer

GROUP BY PROF2

) AS Subquery

GROUP BY ProgrammingLanguage;

1. SELECT TOP 1 PNAME

FROM programmer

ORDER BY DOJ ASC;

1. SELECT TOP 1 PNAME

FROM programmer

WHERE GENDER = 'M' AND (PROF1 = 'PASCAL' OR PROF2 = 'PASCAL')

ORDER BY DOJ DESC;

1. SELECT COALESCE(P1.ProgrammingLanguage, P2.ProgrammingLanguage) AS ProgrammingLanguage

FROM (

SELECT PROF1 AS ProgrammingLanguage, COUNT(\*) AS NumProgrammers

FROM programmer

GROUP BY PROF1

) AS P1

FULL OUTER JOIN (

SELECT PROF2 AS ProgrammingLanguage, COUNT(\*) AS NumProgrammers

FROM programmer

GROUP BY PROF2

) AS P2 ON P1.ProgrammingLanguage = P2.ProgrammingLanguage

WHERE (P1.NumProgrammers = 1 AND P2.NumProgrammers IS NULL)

OR (P1.NumProgrammers IS NULL AND P2.NumProgrammers = 1);

1. SELECT p.PNAME

FROM programmer p

JOIN (

SELECT COALESCE(P1.ProgrammingLanguage, P2.ProgrammingLanguage) AS ProgrammingLanguage

FROM (

SELECT PROF1 AS ProgrammingLanguage, COUNT(\*) AS NumProgrammers

FROM programmer

GROUP BY PROF1

) AS P1

FULL OUTER JOIN (

SELECT PROF2 AS ProgrammingLanguage, COUNT(\*) AS NumProgrammers

FROM programmer

GROUP BY PROF2

) AS P2 ON P1.ProgrammingLanguage = P2.ProgrammingLanguage

WHERE (P1.NumProgrammers = 1 AND P2.NumProgrammers IS NULL)

OR (P1.NumProgrammers IS NULL AND P2.NumProgrammers = 1)

) AS Subquery ON p.PROF1 = Subquery.ProgrammingLanguage OR p.PROF2 = Subquery.ProgrammingLanguage;

1. SELECT TOP 1 PNAME

FROM programmer

WHERE (PROF1 = 'DBASE' OR PROF2 = 'DBASE')

ORDER BY DOB DESC;

1. SELECT PNAME

FROM programmer

WHERE GENDER = 'F'

AND SALARY > 3000

AND PNAME NOT IN (

SELECT PNAME

FROM programmer

WHERE PROF1 IN ('C', 'C++', 'Oracle', 'dBase')

OR PROF2 IN ('C', 'C++', 'Oracle', 'dBase')

);

1. SELECT INSTITUTE

FROM studies

GROUP BY INSTITUTE

HAVING COUNT(\*) = (

SELECT MAX(StudentCount)

FROM (

SELECT INSTITUTE, COUNT(\*) AS StudentCount

FROM studies

GROUP BY INSTITUTE

) AS InstituteCounts

);

1. SELECT COURSE, COURSE\_FEE

FROM studies

WHERE COURSE\_FEE = (

SELECT MAX(COURSE\_FEE)

FROM studies

);

1. SELECT COURSE, COUNT(\*) as student

FROM studies

GROUP BY COURSE

ORDER BY student DESC

OFFSET 0 ROWS FETCH FIRST 2 ROWS ONLY

1. SELECT INSTITUTE, COURSE

FROM studies

WHERE COURSE\_FEE = (SELECT MAX(COURSE\_FEE) FROM studies);

1. SELECT INSTITUTE, COURSE

FROM studies

WHERE COURSE\_FEE < (SELECT AVG(COURSE\_FEE) FROM studies);

1. SELECT COURSE

FROM studies

WHERE COURSE\_FEE BETWEEN (SELECT AVG(COURSE\_FEE) - 1000 FROM studies) AND (SELECT AVG(COURSE\_FEE) + 1000 FROM studies);

1. SELECT TITLE

FROM software

WHERE DCOST = (SELECT MAX(DCOST) FROM software);

1. select course

from studies

group by course

having count(\*) < (

SELECT ROUND(AVG(CAST(course\_count AS FLOAT)), 2) AS average\_count

FROM (

SELECT course, COUNT(\*) AS course\_count

FROM studies

GROUP BY course

) AS subquery

)

1. SELECT TITLE

FROM software

WHERE SCOST = (SELECT MIN(SCOST) FROM software);

1. SELECT PNAME FROM software WHERE SOLD = (SELECT MIN(SOLD) FROM software)
2. SELECT DEVELOPIN

FROM software

WHERE SCOST \* sold = (SELECT MAX(SCOST \* sold) FROM software);

1. SELECT SOLD

FROM software

WHERE DCOST - SCOST = (SELECT MIN(DCOST - SCOST) FROM software);

1. SELECT TITLE

FROM software

WHERE DEVELOPIN = 'PASCAL'

ORDER BY DCOST DESC

offset 0 rows fetch first 1 rows only;

1. SELECT DEVELOPIN

FROM software

GROUP BY DEVELOPIN

ORDER BY COUNT(\*) DESC

offset 0 rows fetch first 1 rows only;

1. SELECT PNAME, COUNT(\*)

FROM SOFTWARE

GROUP BY PNAME

ORDER BY COUNT(\*) DESC

offset 0 rows fetch first 1 rows only;

1. SELECT PNAME

FROM SOFTWARE

WHERE DCOST = (

SELECT MAX(DCOST)

FROM software)

1. SELECT TITLE

FROM software

WHERE SOLD < (SELECT AVG(SOLD) FROM software);

1. select pname

from software

where (sold\*scost) \* 2 > dcost

1. SELECT PNAME, DEVELOPIN, MIN(DCOST)

FROM software

GROUP BY PNAME, DEVELOPIN;

1. SELECT

PNAME,

(SELECT DEVELOPIN

FROM software s2

WHERE s2.PNAME = s1.PNAME

ORDER BY SOLD DESC

offset 0 rows fetch first 1 rows only) AS LANGUAGE\_HIGHEST\_SELLING,

(SELECT DEVELOPIN

FROM software s2

WHERE s2.PNAME = s1.PNAME

ORDER BY SOLD ASC

offset 0 rows fetch first 1 rows only) AS LANGUAGE\_LOWEST\_SELLING

FROM

(SELECT DISTINCT PNAME FROM software) AS s1

1. SELECT MAX(DOB)

FROM programmer

WHERE GENDER = 'M' AND YEAR(DOB) = 1965

1. SELECT PNAME

FROM programmer

WHERE GENDER = 'F' AND DOB = (SELECT MAX(DOB) FROM programmer WHERE GENDER = 'F' AND YEAR(DOJ) = 1992);

1. SELECT YEAR(DOB), COUNT(\*)

FROM programmer

GROUP BY YEAR(DOB)

ORDER BY COUNT(\*) DESC

offset 0 rows fetch first 1 rows only;

1. SELECT MONTH(DOJ), COUNT(\*)

FROM programmer

GROUP BY MONTH(DOJ)

ORDER BY COUNT(\*) DESC

offset 0 rows fetch first 1 rows only;

1. SELECT PROF1 AS LANGUAGE, COUNT(\*) AS PROF\_COUNT

FROM programmer

GROUP BY PROF1

UNION

SELECT PROF2 AS LANGUAGE, COUNT(\*) AS PROF\_COUNT

FROM programmer

GROUP BY PROF2

ORDER BY PROF\_COUNT DESC

offset 0 rows fetch first 1 rows only;

1. SELECT PNAME

FROM programmer

WHERE GENDER = 'M' AND SALARY < (SELECT AVG(SALARY) FROM programmer WHERE GENDER = 'F');

1. SELECT \*

FROM programmer

WHERE GENDER = 'F' AND SALARY > (SELECT MAX(SALARY) FROM programmer);

1. SELECT prof.PROF, COUNT(\*) AS NumProgrammers

FROM (

SELECT PROF1 AS PROF FROM programmer

UNION ALL

SELECT PROF2 AS PROF FROM programmer

) prof

GROUP BY prof.PROF

ORDER BY NumProgrammers DESC

offset 0 rows fetch first 1 rows only

1. SELECT \*

FROM programmer

WHERE SALARY IN (

SELECT SALARY

FROM programmer

GROUP BY SALARY

HAVING COUNT(\*) > 1

);

1. SELECT s.\*

FROM software s

INNER JOIN programmer p ON s.PNAME = p.PNAME

WHERE p.GENDER = 'M' AND p.SALARY > 3000;

1. SELECT \*

FROM software

WHERE DEVELOPIN = 'PASCAL'

AND PNAME IN (SELECT PNAME FROM programmer WHERE GENDER = 'F');

1. SELECT \*

FROM programmer

WHERE year(DOJ) < 1990;

1. SELECT s.\*

FROM software s

JOIN programmer p ON s.PNAME = p.PNAME

JOIN studies st ON p.PNAME = st.PNAME

WHERE p.GENDER = 'F' AND st.INSTITUTE = 'PRAGATHI' AND s.DEVELOPIN = 'C';

1. select count(software.TITLE), sum(software.SOLD), sum(software.SOLD \* software.SCOST)

from studies

inner join software on studies.PNAME = software.PNAME

group by INSTITUTE

1. SELECT s.PNAME

FROM software s

INNER JOIN programmer p ON s.PNAME = p.PNAME

WHERE s.DEVELOPIN = 'dbase'

AND p.GENDER = 'M'

AND p.PNAME IN (

SELECT sp.PNAME

FROM studies sp

WHERE sp.INSTITUTE = (

SELECT INSTITUTE

FROM studies

GROUP BY INSTITUTE

ORDER BY COUNT(PNAME) DESC

offset 0 rows fetch first 1 rows only

)

);

1. SELECT \*

FROM software

WHERE PNAME IN (

SELECT PNAME FROM programmer WHERE GENDER = 'M' AND year(DOB) < 1965

UNION

SELECT PNAME FROM programmer WHERE GENDER = 'F' AND year(DOB) > 1975

);

1. SELECT \*

FROM software

WHERE DEVELOPIN NOT IN (SELECT PROF1 FROM programmer UNION SELECT PROF2 FROM programmer);

1. SELECT s.\*

FROM software s

JOIN programmer p ON s.PNAME = p.PNAME

JOIN studies st ON p.PNAME = st.PNAME

WHERE p.GENDER = 'M' AND st.INSTITUTE = 'SABHARI';

1. SELECT PNAME

FROM programmer

WHERE PNAME NOT IN (SELECT DISTINCT PNAME FROM software);

1. SELECT SUM(s.DCOST)

FROM software s

JOIN programmer p ON s.PNAME = p.PNAME

JOIN studies st ON p.PNAME = st.PNAME

WHERE st.INSTITUTE = 'APPLE';

1. SELECT p1.PNAME, p2.PNAME

FROM programmer p1

JOIN programmer p2 ON p1.DOJ = p2.DOJ AND p1.PNAME < p2.PNAME

1. SELECT p1.PNAME, p2.PNAME

FROM programmer p1

JOIN programmer p2 ON p1.PROF2 = p2.PROF2 AND p1.PNAME < p2.PNAME

1. SELECT p.INSTITUTE, SUM(s.SCOST \* s.SOLD) AS TotalSalesValue

FROM studies p

LEFT JOIN software s ON p.PNAME = s.PNAME

GROUP BY p.INSTITUTE;

1. SELECT p.INSTITUTE

FROM software s

INNER JOIN studies p ON s.PNAME = p.PNAME

WHERE s.SCOST = (SELECT MAX(SCOST) FROM software)

1. SELECT DISTINCT p.PROF1

FROM programmer p

LEFT JOIN software s ON p.PNAME = s.PNAME

WHERE p.PROF1 NOT IN (SELECT DISTINCT DEVELOPIN FROM software)

UNION

SELECT DISTINCT p.PROF2

FROM programmer p

LEFT JOIN software s ON p.PNAME = s.PNAME

WHERE p.PROF2 NOT IN (SELECT DISTINCT DEVELOPIN FROM software);

1. SELECT p.SALARY, st.COURSE

FROM programmer p

JOIN software s ON p.PNAME = s.PNAME

JOIN studies st ON p.PNAME = st.PNAME

WHERE s.SOLD = (SELECT MAX(SOLD) FROM software);

1. SELECT AVG(p.SALARY)

FROM programmer p

JOIN software s ON p.PNAME = s.PNAME

WHERE (s.SOLD \* s.SCOST) > 50000;

1. SELECT COUNT(\*)

FROM software

WHERE PNAME IN (

SELECT PNAME

FROM studies

WHERE COURSE\_FEE = (

SELECT MIN(COURSE\_FEE)

FROM studies

)

);

1. SELECT s.PNAME, COUNT(\*) AS NumPackages, st.INSTITUTE

FROM software s

JOIN studies st ON s.PNAME = st.PNAME

WHERE s.DCOST = (SELECT MIN(DCOST) FROM software)

GROUP BY s.PNAME, st.INSTITUTE;

1. SELECT COUNT(\*) AS NumPackages

FROM software

WHERE PNAME IN (

SELECT s.PNAME

FROM software s

JOIN programmer p ON s.PNAME = p.PNAME

WHERE p.GENDER = 'F' AND p.SALARY > (SELECT MAX(SALARY) FROM programmer WHERE GENDER = 'M')

);

1. SELECT COUNT(\*) AS NumPackages

FROM software s

WHERE s.PNAME IN (

SELECT p.PNAME

FROM programmer p

JOIN studies st ON p.PNAME = st.PNAME

WHERE st.INSTITUTE = 'BDPS'

ORDER BY DATEDIFF(year, p.DOJ, GETDATE()) DESC

offset 0 rows fetch first 1 rows only

);

1. SELECT p.PNAME, st.INSTITUTE

FROM programmer p

LEFT JOIN studies st ON p.PNAME = st.PNAME;

1. SELECT prof.PROF,

COUNT(DISTINCT p.PNAME) AS NumProgrammers,

COUNT(DISTINCT s.TITLE) AS NumPackages

FROM (

SELECT PROF1 AS PROF FROM programmer

UNION

SELECT PROF2 AS PROF FROM programmer

) prof

LEFT JOIN programmer p ON prof.PROF = p.PROF1 OR prof.PROF = p.PROF2

LEFT JOIN software s ON p.PNAME = s.PNAME

GROUP BY prof.PROF;

1. SELECT p.PNAME, COUNT(s.TITLE) AS NumPackages

FROM programmer p

LEFT JOIN software s ON p.PNAME = s.PNAME

GROUP BY p.PNAME;