Write a query that prints a list of employee names (i.e.: the *name* attribute) for employees in **Employee** having a salary greater than  $2000 per month who have been employees for less than  10 months. Sort your result by ascending *employee\_id*.

**Input Format**

The **Employee** table containing employee data for a company is described as follows:

**Input Format**

The **Employee** table containing employee data for a company is described as follows:

A table of numbers with black text

Description automatically generated

Solutions

SELECT name FROM Employee WHERE salary > 2000 AND months < 1

You are given a table, *Projects*, containing three columns: *Task\_ID*, *Start\_Date* and *End\_Date*. It is guaranteed that the difference between the *End\_Date* and the *Start\_Date* is equal to *1* day for each row in the table.

If the *End\_Date* of the tasks are consecutive, then they are part of the same project. Samantha is interested in finding the total number of different projects completed.

Write a query to output the start and end dates of projects listed by the number of days it took to complete the project in ascending order. If there is more than one project that have the same number of completion days, then order by the start date of the project.

A screenshot of a computer

Description automatically generated

Solutions-

SELECT A.Start\_Date, min(B.End\_Date) from

(SELECT Start\_Date FROM Projects where

Start\_Date not IN(SELECT End\_Date FROM Projects)) AS A

INNER JOIN

(SELECT End\_Date From Projects Where End\_Date NOT IN

(SELECT Start\_Date FROM Projects)) AS B

WHERE A.Start\_Date < B.End\_Date

GROUP BY Start\_Date

Order BY MIN(B.End\_Date) - A.Start\_Date ASC, A.Start\_Date

Slution 2 –

WITH ConsecutiveProjects AS (

SELECT

Task\_ID,

Start\_Date,

End\_Date,

DATEADD(DAY, -ROW\_NUMBER() OVER (ORDER BY Start\_Date), Start\_Date) AS Project\_Group

FROM

Projects

),

ProjectDetails AS (

SELECT

MIN(Start\_Date) AS Project\_Start,

MAX(End\_Date) AS Project\_End,

DATEDIFF(DAY, MIN(Start\_Date), MAX(End\_Date)) + 1 AS Project\_Duration

FROM

ConsecutiveProjects

GROUP BY

Project\_Group

)

SELECT

Project\_Start,

Project\_End,

Project\_Duration

FROM

ProjectDetails

ORDER BY

Project\_Duration ASC,

Project\_Start ASC;

QUESTION 3

You are given three tables: Students, Friends and Packages. Students contains two columns: ID and Name. Friends contains two columns: ID and Friend\_ID (ID of the ONLY best friend). Packages contains two columns: ID and Salary (offered salary in $ thousands per month).

Write a query to output the names of those students whose best friends got offered a higher salary than them. Names must be ordered by the salary amount offered to the best friends. It is guaranteed that no two students got same salary offer.

A screenshot of a computer

Description automatically generated

A screenshot of a social media account

Description automatically generatedA screenshot of a computer

Description automatically generated

Solutions –

SELECT A.Name

From

Students AS A JOIN Friends AS F ON A.ID = F.ID

JOIN Packages AS P ON A.ID = P.ID

join Packages AS P2 ON F.Friend\_ID = P2.ID

where P2.Salary > P.Salary

order by P2.Salary;

You are given a table, *Functions*, containing two columns: *X*and *Y*.

A screenshot of a computer

Description automatically generated

Two pairs *(X1, Y1)* and *(X2, Y2)* are said to be *symmetric* *pairs* if *X1 = Y2* and *X2 = Y1*.

Write a query to output all such *symmetric* *pairs* in ascending order by the value of *X*. List the rows such that *X1 ≤ Y1*.

**Sample Input**

A screenshot of a computer

Description automatically generated

**Sample Output**

20 20

20 21

22 23

Solutions-

SELECT DISTINCT

a.X AS X1,

a.Y AS Y1

FROM Functions a

JOIN Functions b

ON a.X = b.Y AND a.Y = b.X

WHERE a.X <= a.Y

ORDER BY X1, Y1;

RIGHT---

SELECT DISTINCT f1.x,f1.y

FROM functions f1

inner join functions f2

ON f1.x IN f2.y

AND f1.y IN f2.x

WHERE f1.x < f1.y

OR f1.x IN (SELECT x

FROM functions

WHERE x = y

GROUP BY x,y

HAVING Count(\*) > 1)

ORDER BY f1.x;