
```
-- BEGIN Q3
```

```
SELECT ConferenceID, COUNT(PaperID) AS PapersCount
FROM Conference
NATURAL JOIN SubmittedPaper
GROUP BY ConferenceID
HAVING COUNT(PaperID)>=5;
```

```
-- END Q3
--
```

```
-- BEGIN Q4
```

```
SELECT ResearcherEmail, COUNT(DISTINCT Name) AS DistinctConferencesCount
FROM Attended
NATURAL JOIN Conference
GROUP BY ResearcherEmail
-- Check for researchers who have the same distinct conferences count as the
highest count
HAVING COUNT(DISTINCT Name) = (
    SELECT COUNT(DISTINCT Name) AS MaxDistinctConferencesCount
    FROM Attended
    NATURAL JOIN Conference
    GROUP BY ResearcherEmail
    ORDER BY COUNT(DISTINCT Name) DESC
    LIMIT 1
);
```

```
-- END Q4
--
```

```
-- BEGIN Q5
```

```
SELECT ResearcherEmail
FROM Attended
NATURAL JOIN Conference
-- Check for researchers who attended before 2024 and didn't attend after 2024
WHERE StartDate < '2024-01-01' AND ResearcherEmail NOT IN(
    SELECT ResearcherEmail
    FROM Attended
    NATURAL JOIN Conference
    WHERE StartDate > '2024-01-01'
)
GROUP BY ResearcherEmail;
```

```
-- END Q5
--
```

```
-- BEGIN Q6
```

```

SELECT ResearcherEmail, COUNT(RejectedPaper.PaperID) AS RejectedPapersCount,
      COUNT(AcceptedPaper.PaperID) AS AcceptedPapersCount
FROM PaperAuthor
LEFT JOIN RejectedPaper ON RejectedPaper.PaperID = PaperAuthor.PaperID
LEFT JOIN AcceptedPaper ON AcceptedPaper.PaperID = PaperAuthor.PaperID
GROUP BY ResearcherEmail
/* Check if researchers have at least as many rejected papers as their accepted
papers,
and they have at least one accepted paper */
HAVING COUNT(AcceptedPaper.PaperID) >= 1
      AND COUNT(RejectedPaper.PaperID) >= COUNT(AcceptedPaper.PaperID);

```

```

-- END Q6
--

```

```

-- BEGIN Q7

```

```

SELECT SessionID, ROUND(TIMESTAMPDIFF(MINUTE, StartDateTime,
      EndDateTime)/COUNT(PaperID))
      AS MinutesPerPresentation
FROM Session
NATURAL JOIN Presentation
GROUP BY SessionID;

```

```

-- END Q7
--

```

```

-- BEGIN Q8

```

```

SELECT ResearcherEmail, COUNT(DISTINCT PaperID) AS presentedPaperCount
FROM Presentation
GROUP BY ResearcherEmail
-- Check if a researcher has presented all of their unique accepted papers at least
once
HAVING COUNT(DISTINCT PaperID) >= (
      SELECT COUNT(DISTINCT PaperID) AS AcceptedPaperCount
      FROM PaperAuthor
      NATURAL JOIN AcceptedPaper
      WHERE ResearcherEmail = Presentation.ResearcherEmail
);

```

```

-- END Q8
--

```

```

-- BEGIN Q9

```

```

SELECT InfluencerResearcherEmail, InfluencedResearcherEmail, MIN(Depth)
FROM (
      -- Check for all relationships of Depth = 1
      SELECT SupervisorEmail AS InfluencerResearcherEmail,
            SuperviseeEmail AS InfluencedResearcherEmail, 1 AS Depth
      FROM Researcher
      INNER JOIN Supervisor ON SuperviseeEmail = Email

```

```

UNION ALL
-- Check for all relationships of Depth = 2
SELECT s1.SupervisorEmail AS InfluencerResearcherEmail,
       s2.SuperviseeEmail AS InfluencedResearcherEmail, 2 AS Depth
FROM Researcher
INNER JOIN Supervisor s1 ON s1.SuperviseeEmail = Email
INNER JOIN Supervisor s2 ON s1.SuperviseeEmail = s2.SupervisorEmail
UNION ALL
-- Check for all relationships of Depth = 3
SELECT D2Supervisor AS InfluencerResearcherEmail,
       Supervisor.SuperviseeEmail AS InfluencedResearcherEmail, 3 AS Depth
FROM Supervisor
INNER JOIN (
    SELECT s1.SupervisorEmail AS D2Supervisor, s2.SuperviseeEmail AS
D2Supervisee
    FROM Researcher
    INNER JOIN Supervisor s1 ON s1.SuperviseeEmail = Email
    INNER JOIN Supervisor s2 ON s1.SuperviseeEmail = s2.SupervisorEmail
) AS Depth2 ON Depth2.D2Supervisee = Supervisor.SupervisorEmail
) AS InfluenceRelationships
-- Remove duplicate Influencer/Influenced relationships that are not minimum depth
GROUP BY InfluencerResearcherEmail, InfluencedResearcherEmail;

-- END Q9
--

```

```

-- BEGIN Q10

SELECT NextCon.Name AS ConferenceName, PrevCon.OccurrenceNumber AS
PrevOccurrenceNumber,
       PrevAttendanceCount, NextCon.OccurrenceNumber AS NextOccurrenceNumber,
NextAttendanceCount
FROM(
    SELECT Name, OccurrenceNumber, COUNT(ResearcherEmail) AS NextAttendanceCount
    FROM Conference
    NATURAL JOIN Attended
    GROUP BY ConferenceID
) AS NextCon
-- Find the consecutive conferences with the same name
INNER JOIN(
    SELECT Name, OccurrenceNumber, COUNT(ResearcherEmail) AS PrevAttendanceCount
    FROM Conference
    NATURAL JOIN Attended
    GROUP BY ConferenceID
) AS PrevCon ON NextCon.Name = PrevCon.Name
AND NextCon.OccurrenceNumber = PrevCon.OccurrenceNumber + 1
-- Check for a drop in attendance
WHERE PrevAttendanceCount > NextAttendanceCount;

-- END Q10
--

```

```

-- END OF ASSIGNMENT Do not write below this line

```