



--

---

-- BEGIN Q3

```
SELECT videoID, title
FROM video
INNER JOIN cocreator ON video.id = cocreator.videoID
INNER JOIN content_creator ON content_creator.id = cocreator.creatorID
WHERE viewCount >= 10000000 AND screenName = 'TaylorSwiftOfficial';
```

-- END Q3

--

---

-- BEGIN Q4

```
SELECT id AS videoID, title, COUNT(*) AS linkedCount
FROM video
INNER JOIN annotation ON video.id = annotation.destinationVideoID
GROUP BY destinationVideoID
ORDER BY destinationVideoID DESC
LIMIT 1;
```

-- END Q4

--

---

-- BEGIN Q5

```
SELECT id AS videoID, uploaded AS uploadDatetime, COUNT(*) AS ratingCount
FROM video
INNER JOIN rating ON video.id = rating.videoID
GROUP BY videoID
HAVING COUNT(*) >= 3 AND id IN (
SELECT videoID
FROM video_hashtag
INNER JOIN hashtag ON hashtag.id = video_hashtag.hashtagID
WHERE tag LIKE '#memes');
```

-- END Q5

--

---

-- BEGIN Q6

```
SELECT username, realName, screenName
FROM user
INNER JOIN content_creator ON user.id = content_creator.linkedUser
WHERE reputation < 50 AND content_creator.id IN
(SELECT cocreator.creatorID
FROM cocreator
GROUP BY creatorID
HAVING COUNT(*) >= 3) AND content_creator.id IN (
```

```
SELECT cocreator.creatorID
FROM cocreator
INNER JOIN rating ON cocreator.videoID = rating.videoID
GROUP BY creatorID
HAVING COUNT(*) >= 6);
```

```
-- END Q6
--
```

---

```
-- BEGIN Q7
```

```
WITH PoliteComments AS (
    SELECT tag, COUNT(*) AS commentCount
    FROM rating
    INNER JOIN video_hashtag ON rating.videoID = video_hashtag.videoID
    INNER JOIN hashtag ON video_hashtag.hashtagID = hashtag.id
    WHERE comment LIKE '%thank you%' OR comment LIKE '%well done%'
    GROUP BY tag
)
SELECT tag, commentCount
FROM PoliteComments
WHERE commentCount = (SELECT MAX(commentCount) FROM PoliteComments);
```

```
-- END Q7
--
```

---

```
-- BEGIN Q8
```

```
SELECT tag AS hashtag, COUNT(*) AS totalAnnotationsAsDestination, SUM(duration) AS
totalDuration
FROM hashtag
INNER JOIN video_hashtag ON hashtag.id = hashtagID
INNER JOIN annotation ON videoID = destinationVideoID
GROUP BY hashtagID
HAVING totalAnnotationsAsDestination IN (SELECT * FROM (
SELECT DISTINCT(t.totalAnnotationsAsDestination)
FROM (SELECT hashtagID, COUNT(*) AS totalAnnotationsAsDestination
FROM hashtag
INNER JOIN video_hashtag ON hashtag.id = hashtagID
INNER JOIN annotation ON videoID = destinationvideoID
GROUP BY hashtagID
ORDER BY COUNT(*) DESC) AS t LIMIT 3 ) AS newTable)
ORDER BY COUNT(*) DESC;
```

```
-- END Q8
--
```

---

```
-- BEGIN Q9
```

```
SELECT DISTINCT(realName), screenName
FROM content_creator_hashtag
NATURAL JOIN cocreator
```

```
INNER JOIN content_creator ON id = cocreator.creatorID
WHERE hashtagID = 5 AND videoID IN (SELECT DISTINCT(videoID)
FROM cocreator
INNER JOIN content_creator ON creatorID = content_creator.id
INNER JOIN content_creator_hashtag ON cocreator.creatorID =
content_creator_hashtag.creatorID
INNER JOIN hashtag ON content_creator_hashtag.hashtagID = hashtag.id
WHERE tag = '#technology'
ORDER BY videoID)
ORDER BY realName ASC;
```

```
-- END Q9
--
```

---

```
-- BEGIN Q10
```

```
SELECT DISTINCT (content_creator.realName), content_creator.screenName
FROM content_creator
JOIN cocreator ON content_creator.id = cocreator.creatorID
JOIN video ON cocreator.videoID = video.id
WHERE content_creator.screenName != 'INFO20003Memes'
AND video.uploaded >= '2023-01-01'
AND cocreator.videoID IN (
    SELECT cocreator.videoID
    FROM cocreator
    JOIN content_creator AS contentcreator2 ON cocreator.creatorID =
contentcreator2.id
    WHERE contentcreator2.screenName = 'INFO20003Memes'
    AND cocreator.videoID IS NOT NULL
    AND video.uploaded >= '2023-01-01'
);
```

```
-- END Q10
--
```

---

```
-- END OF ASSIGNMENT Do not write below this line
```