

CPSC 368: Databases in Data Science
Practice Questions: Writing SQL

Consider the following relational schema for a video platform (e.g., Netflix).

1. Member(id, userNumber, username, startDate, membershipType)
 - A single membership can have many members. E.g., one person buys a membership and the whole family uses it. Each family member is identified with a userNumber.
2. Content(id, name, year, **distributionStudioID**, startAvailabilityDate, endAvailabilityDate)
 - distributionStudioID refers to the id attribute in DistributionStudio.
3. TVShows(**id**, season#, episode#, episodeName)
 - id refers to the same named column in Content
4. Movies(**id**, budget)
 - id refers to the same named column in Content
5. MovieStars(starID, firstName, lastName)
6. ActedIn(id, **starID**, characterName)
 - id refers to the same named column in Content
 - starID refers to the same named column in MovieStars
7. DistributionStudio(id, name, country)
8. Watches(id, userNumber, contentID, date)
 - id and userNumber refer to the same named columns in Member
 - contentID refers to the id attribute in Content
9. Tags(name)
10. PreferredTags(**id**, userNumber, **name**, dateAdded)
 - id and userNumber refer to the same named columns in Member
 - name refers to the same named column in Tags
11. ContentTags(**id**, **name**)
 - id refers to the same named column in Content
 - name refers to the same named column in Tags
12. Rating(id, userNumber, contentID, ratingValue)
 - id and userNumber refer to the same named columns in Member
 - contentID refers to the id attribute in Content
 - ratingValue is a value from 1 to 5

Part 1: Practicing IN/NOT IN

1. Find all the content that has never been viewed at all.
2. Find all the tags that are not preferred by the user with id = 12 and userNumber = 0.
3. Find all the distribution studios that have not released content in 2020.

Part 2: Practicing EXISTS/NOT EXISTS

1. Find all the actors (starID, firstName, lastName) who have not acted in a movie with "2" in the name.
2. Find all the actors (starID, firstName, lastName) who have acted in a TV show and a movie.

Part 3: Practicing Division (Method 2 – i.e., no EXCEPT)

1. Find the usernames of members who have watched all the content with a "comedy" tag.
2. Find the usernames of members who have watched all the content released in 2020.
3. Find the first and last name of movie stars who have acted in every TV show or movie distributed by "Orange Productions".

Answer Key

Part 1

1.

```
SELECT *
FROM Content
WHERE id NOT IN (SELECT DISTINCT contentID FROM Watches)
```
2.

```
SELECT *
FROM Tags t
WHERE t.name NOT IN (SELECT pt.name
                     FROM PreferredTags pt
                     WHERE pt.id = 12 AND pt.userNumber = 0)
```
3.

```
SELECT *
FROM DistributionStudio
WHERE id NOT IN (SELECT DISTINCT distributionStudioID
                FROM Content WHERE year = 2020)
```

Part 2

1.

```
SELECT ms.starID, ms.firstName, ms.lastName
FROM MovieStars ms, ActedIn ai
WHERE ms.starID = ai.starID AND NOT EXISTS (
    SELECT *
    FROM Movies m, Content c
    WHERE m.id = c.id AND c.name LIKE '%2%' AND ai.id = m.id)
```
2.

```
SELECT ms.starID, ms.firstName, ms.lastName
FROM MovieStar ms, ActedIn ai, Movies m
WHERE ms.starID = ai.starID AND m.id = ai.id AND EXISTS (
    SELECT *
    FROM TVShow ts, ActedIn ai1
    WHERE ts.id = a1.id AND a1.starID = ms.starID)
```

OR

```
SELECT ms.starID, ms.firstName, ms.lastName
FROM MovieStar ms, ActedIn ai
WHERE ms.starID = ai.starID AND EXISTS (SELECT *
                                       FROM Movies m, ActedIn ai1
                                       WHERE m.id = ai.id AND a1.starID = ms.starID)
```

INTERSECT

```
SELECT ms.starID, ms.firstName, ms.lastName
FROM MovieStar ms, ActedIn ai
WHERE ms.starID = ai.starID AND EXISTS (
    SELECT *
    FROM TVShows t, ActedIn ai1
    WHERE t.id = ai1.id AND a1.starID = ms.starID)
```

OR

```
SELECT ms.starID, ms.firstName, ms.lastName
FROM MovieStar ms, ActedIn a, Movies m
WHERE ms.id = a.starID AND a.id = m.id AND EXISTS (
    SELECT a1.starID
    FROM ActedIn a1, TVShows ts WHERE a1.id = ts.id AND a1.starID = a.starID)
```

Part 3

1.

```
SELECT username
FROM Member m
WHERE NOT EXISTS(
    SELECT ct.id
    FROM ContentTags ct
    WHERE ct.name = "comedy" AND NOT EXISTS(
        SELECT *
        FROM Watches w
        WHERE w.contentID = ct.id AND w.id = m.id AND w.userNumber =
            m.userNumber))
```

CPSC 368: Databases in Data Science
Practice Questions: Writing SQL

2.

```
SELECT username
FROM Member m
WHERE NOT EXISTS(
    SELECT c.id
    FROM Content c
    WHERE year = 2020 AND NOT EXISTS(
        SELECT *
        FROM Watches w
        WHERE c.id = w.contentID AND w.id = m.id AND w.userNumber =
            m.userNumber))
```
3.

```
SELECT ms.firstName, ms.lastName
FROM MovieStar ms
WHERE NOT EXISTS (
    SELECT c.id
    FROM Content c, DistributionStudio ds
    WHERE c.distributionStudioID = ds.id AND ds.name = "Orange Productions" AND
    NOT EXISTS (
        SELECT *
        FROM ActedIn ai
        WHERE ai.id = c.id AND ai.starID = ms.starID))
```