Burst1_Q1

1:30

Are the following two queries returning the same set of results?

1

```
SELECT DISTINCT name, year
FROM movies
```

2

```
SELECT id FROM movies
```

A. Yes, because movies with the same name cannot be released in different years
B. Yes, because movies cannot have the same name
C. No, because movies with the same name can only be released in different years
D. <u>No, because movies with the same name can be released in the same year</u>

Burst1_Q2

1:30

Consider the following query

```
SELECT role AS part
FROM imdb.roles
WHERE actor_id=759315
      AND movie_id = 376108
```

How many tuples will be returned?

A. 1 tuple, because no actor can play multiple times in the same movie
B. 0 tuples, because the actor having id = 300 did not played in the movie
C. <u>More than 1 tuples, because the actor might have played multiple roles in the movie</u>
D. 1 tuple, because the actor played a single role in the movie

Burst1_Q3

1:30

How many of the following queries return the same result set?

1
```
SELECT *
FROM imdb.roles
WHERE role = "A"
```

2
```
SELECT *
FROM imdb.roles
WHERE role LIKE "A"
```

3
```
SELECT *
FROM imdb.roles
WHERE role LIKE "A_"
```

4
```
SELECT *
FROM imdb.roles
WHERE role LIKE "A%"
```

5
```
SELECT *
FROM imdb.roles
WHERE role LIKE "A%_"
```

A. Only 3) and 4)
B. Only 1) and 2)
C. 1) and 2) , 4) and 5)
D. All

Burst2_Q1

1:30

Are the following two queries returning the same set of results?

1
```
SELECT *
FROM imdb.movies
WHERE rank IS NOT NULL
```

2
```
SELECT *
FROM imdb.movies
WHERE rank > 0
```

A. <u>Yes, because no movie in the DB has rank = 0</u>
B. Yes, because all movies in the DB have a rank
C. No, because some movies have rank = NULL
D. No, because the two queries test different condition

Burst2_Q2

1:30

Which of the following SQL queries are guaranteed to always produce a result-set containing ALL the tuples from the Movies table of the IMDB database?

1

```
SELECT DISTINCT name, year
FROM imdb.movies
WHERE id IS NOT NULL
```

2

```
SELECT DISTINCT id, name
FROM imdb.movies
WHERE id > 0
```

3

```
SELECT DISTINCT  name,year,rank
FROM imdb.movies
```

4

```
SELECT *
FROM imdb.movies
WHERE name IS NOT NULL
```

Burst2_Q3

2:00

Write the SQL query that can support the following data need. [OPEN QUESTION]

Retrieve the surname of female actors that have a first name containing the character 'a' at least 6 times

Burst3_Q1

2:00

Which statement better describes the following query?

```
SELECT DISTINCT b.actor_id
FROM imdb.roles AS a,imdb.roles AS b,imdb.movies AS movies
WHERE a.movie_id = movies.id AND b.movie_id = movies.id
AND a.actor_id <> b.actor_id
AND a.actor_id = 393411
```

A. Retrieve the ID of actors that played in movies featuring actor “393411"
B. Retrieve the ID of actors that played the same role as actor "393411"
C. Retrieve the roles of actors that played multiple roles in movies featuring actor "393411"
D. Retrieve the ID of actor "393411" in movies played only by him/herself

Burst3_Q2

3:00

Consider the query from the previous exercise (1) and the following query. Are the two queries returning the same set of results?

```
SELECT DISTINCT b.actor_id
FROM imdb.roles AS a,imdb.roles AS b
 WHERE a.actor_id <> b.actor_id
   AND a.actor_id = 393411
```

A. Yes
B. No

Burst3_Q3

1:30

Why is the following query taking a long time to return results? [open question]

```
SELECT DISTINCT b.actor_id
FROM imdb.roles AS a,imdb.roles AS b
  WHERE a.actor_id <> b.actor_id
    AND a.actor_id = 393411
```

Burst3_Q4

0:30

[BONUS] Who is actor 393411?