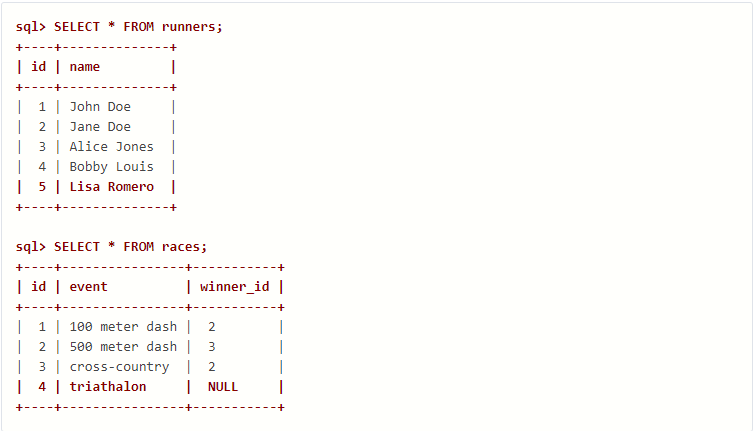
**SQL:**

**Question: 1**

**Given the following tables:**

****

**What will be the result of the query below?**

**SELECT \* FROM runners WHERE id NOT IN (SELECT winner\_id FROM races)**

**Explain your answer and also provide an alternative version of this query that will avoid the issue that it exposes.**

Answer:

SELECT \* FROM runners WHERE id NOT IN (SELECT winner\_id FROM races)

This query selects all records from the 'runners' table where the 'id' of the runner is not found in the 'winner\_id' column of the 'races' table.

This query has an issue, 'winner\_id' column in the 'races' table contains NULL values.

**the alternative version using LEFT JOIN of this query that will avoid the issue:**

**```**

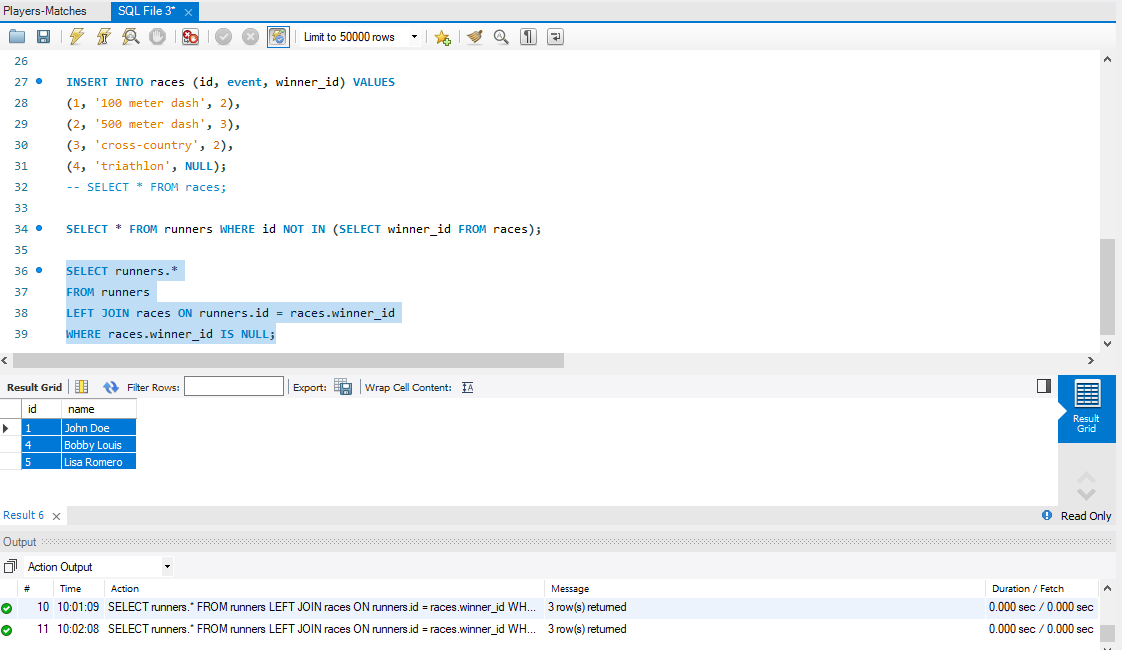
**SELECT runners.\***

**FROM runners LEFT JOIN races**

**ON runners.id = races.winner\_id**

**WHERE races.winner\_id IS NULL;**

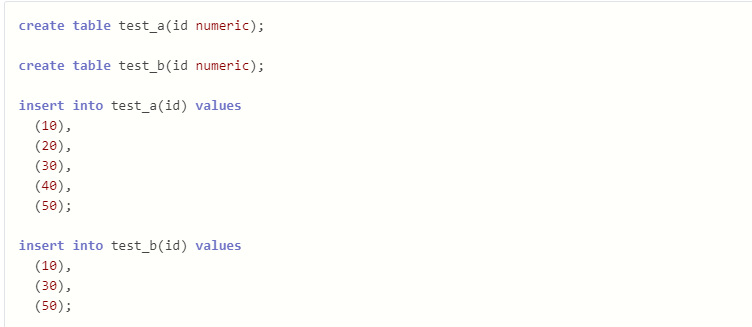
**```**

****

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Question: 2**

Given two tables created as follows



Write a query to fetch values in table test\_a that are and not in test\_b without using the NOT keyword.

Answer:

```

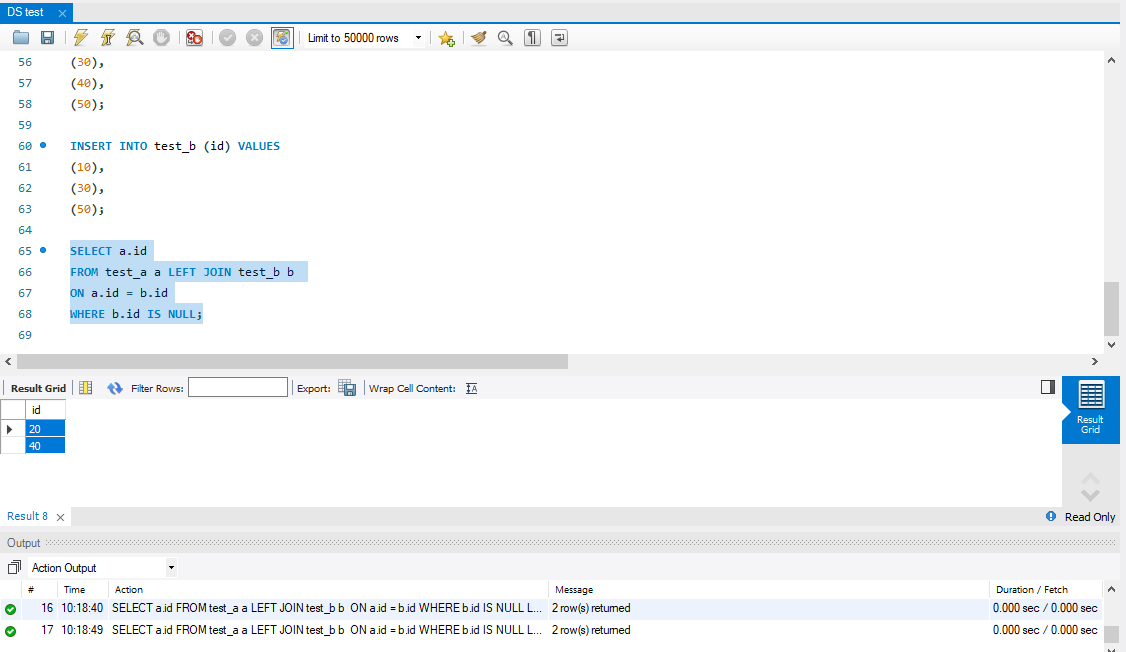
SELECT a.id

FROM test\_a a

LEFT JOIN test\_b b ON a.id = b.id

WHERE b.id IS NULL;

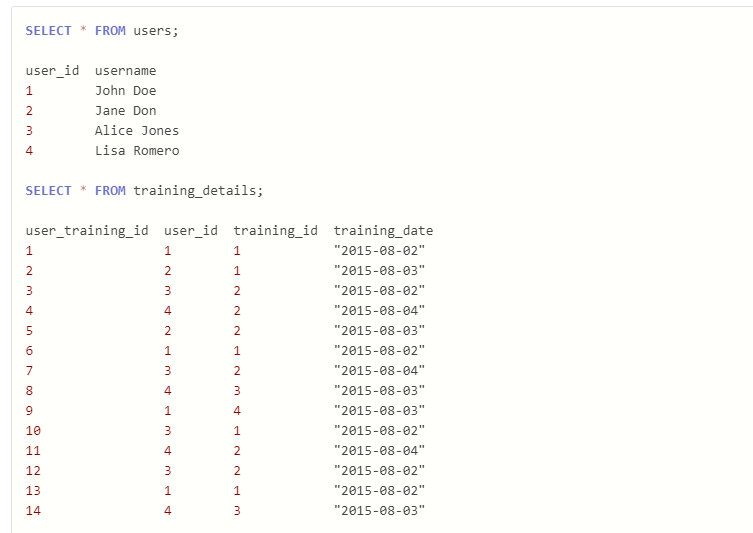
```



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Question: 3**

Given the following tables:



Write a query to to get the list of users who took the a training lesson more than once in the same day, grouped by user and training lesson, each ordered from the most recent lesson date to oldest date.

Answer:

```

SELECT u.user\_id, u.username, td.training\_id, td.training\_date

FROM users u

JOIN training\_details td ON u.user\_id = td.user\_id

WHERE td.training\_date IN (

SELECT training\_date

FROM training\_details

WHERE user\_id = u.user\_id AND training\_id = td.training\_id

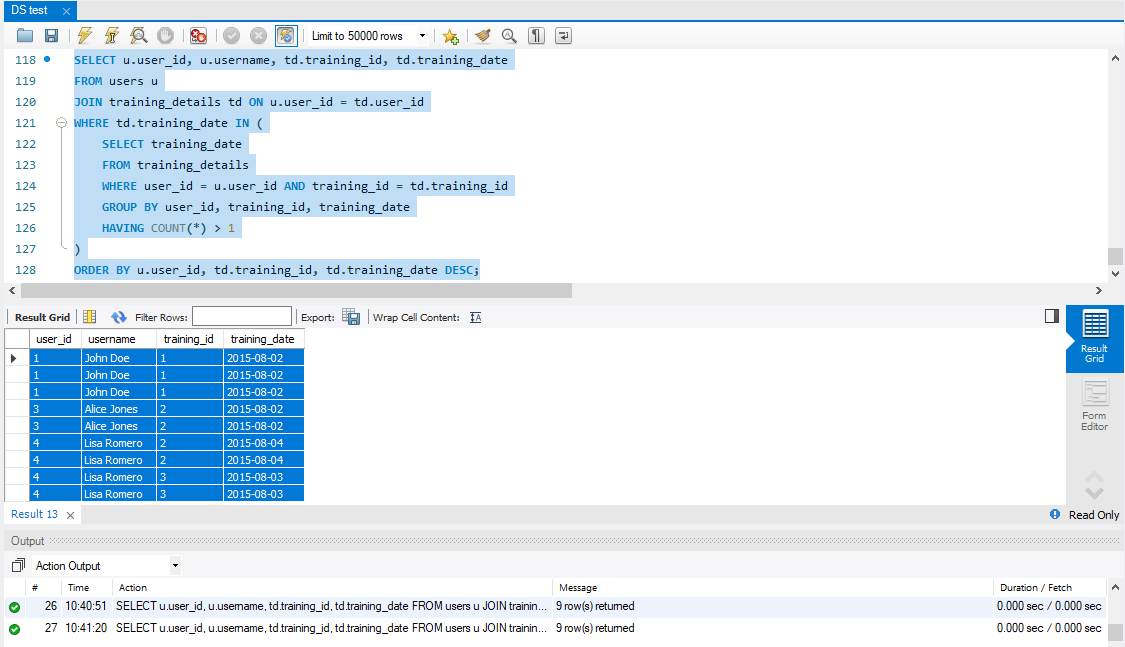
GROUP BY user\_id, training\_id, training\_date

HAVING COUNT(\*) > 1

)

ORDER BY u.user\_id, td.training\_id, td.training\_date DESC;

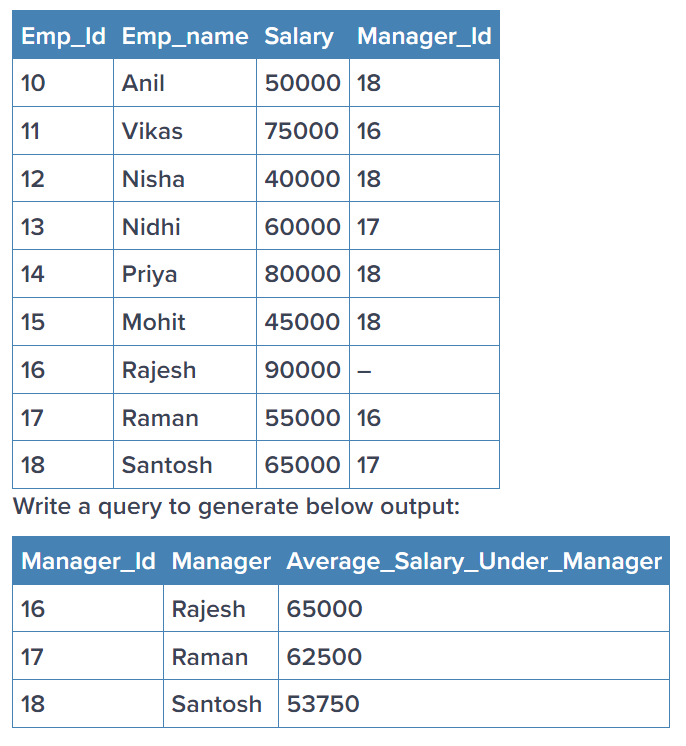
```



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Question: 4**

Consider the Employee table below.



Answer:

```

WITH ManagerData AS (

SELECT

e.Manager\_Id AS Manager\_Id,

m.Emp\_Name AS Manager\_Name,

e.Salary,

AVG(e.Salary) OVER (PARTITION BY e.Manager\_Id) AS Average\_Salary\_Under\_Manager

FROM Employe e INNER JOIN Employe m ON e.Manager\_Id = m.Emp\_Id

)

SELECT

DISTINCT Manager\_Id AS Manager\_Id,

Manager\_Name AS Manager\_Name,

Average\_Salary\_Under\_Manager AS Average\_Salary\_Under\_Manager

FROM ManagerData

WHERE Manager\_Id IS NOT NULL;

```

