**SQL**

**Question 1**

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.

**Solution**

This query is used for extracting record from runner table where id not present in winner id records

Here is code

use dev;

CREATE TABLE runner (

id INT ,

name VARCHAR(255)

);

CREATE TABLE race (

id INT,

event VARCHAR(255),

winner\_id INT

);

INSERT INTO runner (id, name) VALUES

(1, 'Runner1'),

(2, 'Runner2'),

(3, 'Runner3'),

(4, 'Runner4'),

(5, 'Runner5');

select \* from runners;

INSERT INTO race (id, event, winner\_id) VALUES

(1, 'Race1', 2),

(2, 'Race2', 3),

(3, 'Race3', 2),

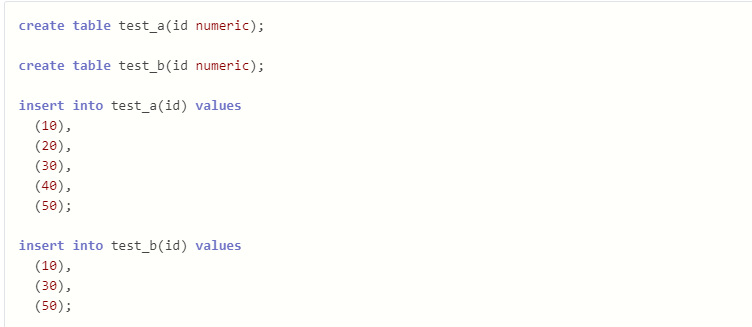
(4, 'Race4', null);

select \* from race;

SELECT \* FROM runner WHERE id NOT IN (SELECT winner\_id FROM race);

**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.

**Solution :-**

**Code**

SELECT a.id

FROM test\_a a

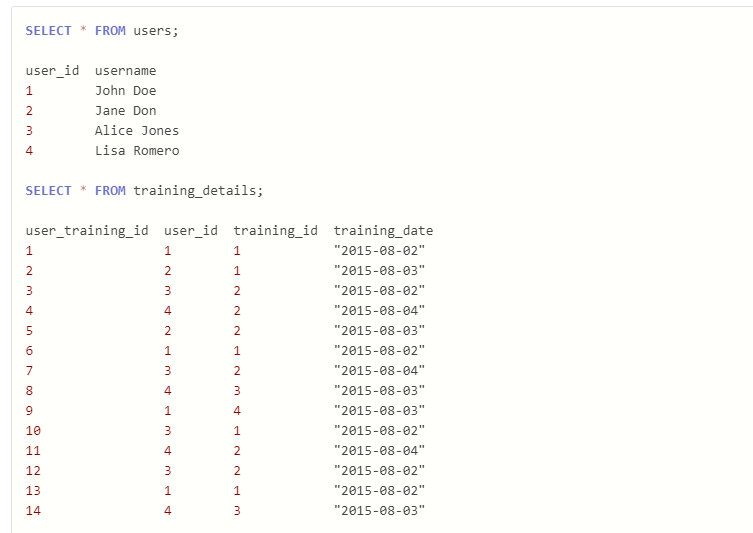
LEFT JOIN test\_b b ON a.id = b.id

WHERE b.id IS NULL;

In this code for extracting id which are not present in test\_b so in our code i am using left join which is used to extract all the data from left side and common record from right table

**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.**

**Solution 3**

Here is a code

CREATE TABLE users (

user\_id INT PRIMARY KEY,

user\_name VARCHAR(255)

);

INSERT INTO users (user\_id, user\_name) VALUES

(1, 'John'),

(2, 'John Dea'),

(3, 'Alisa'),

(4, 'Lisa');

CREATE TABLE training\_detail (

user\_training\_id INT PRIMARY KEY,

user\_id INT,

training\_id INT,

training\_date DATE,

FOREIGN KEY (user\_id) REFERENCES users(user\_id)

);

INSERT INTO training\_detail (user\_training\_id, user\_id, training\_id, training\_date) VALUES

(1, 1, 101, '2024-03-10'),

(2, 2, 102, '2024-03-10'),

(3, 3, 101, '2024-03-11'),

(4, 4, 103, '2024-03-12'),

(5, 1, 101, '2024-03-10'),

(6, 2, 102, '2024-03-10');

SELECT

u.user\_id,

u.user\_name,

td.training\_id,

COUNT(\*) AS lesson\_count,

MAX(td.training\_date) AS most\_recent\_date

FROM

users u

JOIN

training\_detail td ON u.user\_id = td.user\_id

GROUP BY

u.user\_id, u.user\_name, td.training\_id, td.training\_date

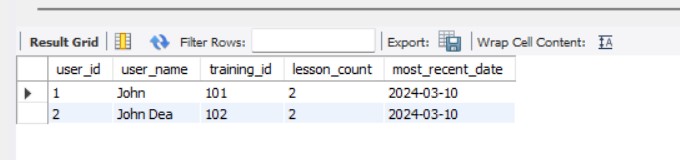
HAVING

lesson\_count > 1

ORDER BY

most\_recent\_date DESC;

**Output:-**

****

Here in this code we are fetching a records which the lesson count is more than 1 in this we used a join on user\_id from both the tables

Returns a

User\_id

User\_name :- from user table only the those where count of lesson is greater than 1

Training\_id :-

Lesson\_count

recent\_date

**Question 4**

**Soution 4 :-**

Code ➖

CREATE TABLE Employee (

Emp\_id INT PRIMARY KEY,

Emp\_name VARCHAR(255) NOT NULL,

Salary DECIMAL(10,2) NOT NULL,

Manager\_id INT

);

INSERT INTO Employee (Emp\_id, Emp\_name, Salary, Manager\_id)

VALUES(10, 'anil', 50000.00, 18),

(11, 'vikas', 75000.00, 16),

(12, 'nisha', 40000.00, 18),

(13, 'Nidhi', 60000.00, 17),

(14, 'priya', 80000.00, 18),

(15, 'mohit', 45000.00, 18),

(16, 'rajesh', 90000.00, null),

(17, 'raman', 55000.00, 16),

(18, 'santosh', 65000.00, 17)

;

select \* from employee;

SELECT E2.EMP\_ID, E2.EMP\_NAME, AVG(E1.SALARY)

FROM Employee E1

INNER JOIN Employee E2

ON E1.MANAGER\_ID = E2.EMP\_ID

GROUP BY E2.EMP\_ID, E2.EMP\_NAME;

Here in this query we are using inner join fetching record from same table on manager\_id column and emp\_id we just grouped record