**Question 1:**

CREATE TABLE runners (

id INT PRIMARY KEY,

name VARCHAR(255)

);

INSERT INTO runners (id, name) VALUES

(1, 'John Doe'),

(2, 'Jane Doe'),

(3, 'Alice Jones'),

(4, 'Bobby Louis'),

(5, 'Lisa Romero');

CREATE TABLE races (

id INT PRIMARY KEY,

event VARCHAR(255),

winner\_id INT,

FOREIGN KEY (winner\_id) REFERENCES runners(id)

);

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

(1, '100 meter dash', 2),

(2, '500 meter dash', 3),

(3, 'cross-country', NULL);

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

The above query selects all data from the runners table whose id is not present in races table. This query will not return any rows. To fix the issue we have to remove not from the query:

**SELECT** \* **FROM** runners **WHERE** id **IN** (**SELECT** winner\_id **FROM** races)

**Question 2:**

create table test\_a(id numeric);

insert into test\_a(id) values

(10),

(20),

(30),

(40),

(50);

create table test\_b(id numeric);

insert into test\_b(id) values

(10),

(30),

(50);

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

**SELECT**  a.id **FROM** test\_a a

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

**WHERE** b.id IS **NULL**;

**Question: 3**

**Write a query 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.**

Create TABLE users (

user\_id INT PRIMARY KEY,

username VARCHAR(255)

);

INSERT INTO users (user\_id, username) VALUES

(1, 'John Doe'),

(2, 'Jane Don'),

(3, 'Alice Jones'),

(4, 'Lisa Romero');

CREATE TABLE training\_details (

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\_details (user\_training\_id, user\_id, training\_id, training\_date) VALUES

(1, 1, 1, '2015-08-02'),

(2, 1, 1, '2015-08-03'),

(3, 2, 1, '2015-08-02'),

(4, 2, 2, '2015-08-03'),

(5, 3, 1, '2015-08-04'),

(6, 3, 2, '2015-08-02'),

(7, 4, 3, '2015-08-04'),

(8, 4, 3, '2015-08-03'),

(9, 1, 1, '2015-08-02'),

(10, 1, 1, '2015-08-03'),

(11, 2, 1, '2015-08-02'),

(12, 2, 2, '2015-08-03'),

(13, 3, 1, '2015-08-04'),

(14, 3, 2, '2015-08-02');

**Question 4:**

CREATE TABLE Employees (

Emp\_Id INT PRIMARY KEY,

Emp\_name VARCHAR(255),

Salary INT,

Manager\_Id INT

);

INSERT INTO Employees (Emp\_Id, Emp\_name, Salary, Manager\_Id) VALUES

(10, 'Anil', 50000, 18),

(11, 'Vikas', 75000, 16),

(12, 'Nisha', 40000, 18),

(13, 'Nidhi', 60000, 17),

(14, 'Priya', 80000, 18),

(15, 'Mohit', 45000, 18),

(16, 'Rajesh', 90000, NULL),

(17, 'Raman', 55000, 16),

(18, 'Santosh', 65000, NULL);

SELECT e.Manager\_Id,

COALESCE(m.Emp\_name, 'Unknown') AS Manager,

AVG(e.Salary) AS Average\_Salary\_Under\_Manager

FROM Employees e

LEFT JOIN Employees m ON e.Manager\_Id = m.Emp\_Id

WHERE e.Manager\_Id IS NOT NULL

GROUP BY e.Manager\_Id, m.Emp\_name

ORDER BY e.Manager\_Id;