**SQL Answers**

**Q1 Ans:**  
The query SELECT \* FROM runners WHERE id NOT IN (SELECT winner\_id FROM races) will return all runners who have never won a race. It achieves this by using a subquery and the NOT IN operator.

Here's a breakdown of how it works:

1. The subquery SELECT winner\_id FROM races retrieves all the winner IDs from the races table.
2. The main query SELECT \* FROM runners WHERE id NOT IN (SELECT winner\_id FROM races) selects all runners from the runners table whose IDs are not present in the list of winner IDs retrieved from the subquery.

The issue with this approach is that it treats NULL values differently than expected. In SQL, NULL doesn't mean the same thing as an empty value. It indicates that a value is missing entirely. Because of this, if a race has no winner recorded (winner\_id is NULL), any runner who participated in that race will be excluded from the results even if they won other races.

Alternative query to avoid this issue:

SELECT \* FROM runners r LEFT JOIN races ra ON r.id = ra.winner\_id WHERE ra.winner\_id IS NOT NULL;

**Q2 Ans:**

select \* from test\_a except select \* from test\_b;

**Q3 Ans:**

SELECT u.user\_id, username, training\_id, training\_date, COUNT(\*) AS count

FROM users u

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

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

HAVING COUNT(\*) > 1

ORDER BY training\_date DESC;

**Q4 Ans**  
SELECT Manager\_Id, AVG(Salary) AS Manager\_Average\_Salary\_Under\_Manager

FROM Employee

GROUP BY Manager\_Id

ORDER BY Manager\_Id;