1. **What does**UNION**do? What is the difference between**UNION**and**UNION ALL**?**
2. UNION statement will return all values of SELECT statements combined into one column. This makes viewing SELECT of multiple tables easily fit into one return.

UNION ALL will do the same, except it will show duplicate values. Previously UNION would only show duplicate values once, whereas UNION ALL will show the value exactly how many times it appears in the tables

Given the following tables:

sql> SELECT \* FROM runners;

+----+--------------+

| id | name |

+----+--------------+

| 1 | John Doe |

| 2 | Jane Doe |

| 3 | Alice Jones |

| 4 | Bobby Louis |

| 5 | Lisa Romero |

+----+--------------+

sql> SELECT \* FROM races;

+----+----------------+-----------+

| id | event | winner\_id |

+----+----------------+-----------+

| 1 | 100 meter dash | 2 |

| 2 | 500 meter dash | 3 |

| 3 | cross-country | 2 |

| 4 | triathalon | NULL |

+----+----------------+-----------+

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.

This query is trying to show the runners who have not won any races. It's looking for rows that don't have an ID in the winner\_id column from the races table. The NOT IN function does not handle NULL by default, and since we have a NULL value it messes up the entire query.

**Revision**:

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

**Revision2**: SELECT runners.\*

FROM runners

LEFT JOIN races ON runners.id = races.winner\_id

WHERE races.id IS NULL;

1. What are the NVL and the NVL2 functions in SQL? How do they differ?

These are OracleDB functions and they are used to handle NULL values.

- NVL is used to return NULL with a default value we have specified.

- NVL2 is used to return one value if an expression is not NULL and another value if it is NULL.

- The main differences in the coding would be below

SELECT NVL(column,'default\_value') FROM table

SELECT NVL2(column,'not\_null\_value', 'null\_value') FROM table

1. What is the difference between single-row functions and multiple-row functions?

Single row functions are used on a single row of data at a time. These are most commonly used as Arithmetic or Date functions.

- Round, To\_Date, Month etc.

Multiple row functions are used on multiple rows to provide a single result. Most commonly used with aggregate functions.

- Sum, Avg, Count

- Group by, Having

1. What is the group by clause used for?

Group by is used to combine rows that have shared values in specific columns. We do this to use aggregate functions like max,min,avg, etc.

1. Given following Tables:

**SELECT** \* **FROM** users;

user\_id username

1 John Doe

2 Jane Don

3 Alice Jones

4 Lisa Romero

**SELECT** \* **FROM** training\_details;

user\_training\_id user\_id training\_id training\_date

1 1 1 "2015-08-02"

2 2 1 "2015-08-03"

3 3 2 "2015-08-02"

4 4 2 "2015-08-04"

5 2 2 "2015-08-03"

6 1 1 "2015-08-02"

7 3 2 "2015-08-04"

8 4 3 "2015-08-03"

9 1 4 "2015-08-03"

10 3 1 "2015-08-02"

11 4 2 "2015-08-04"

12 3 2 "2015-08-02"

13 1 1 "2015-08-02"

14 4 3 "2015-08-03"

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.

SELECT training\_details.user\_id, training\_details.training\_id as training\_id, MIN(users.username) as name, MAX(training\_details.training\_date) as latest\_train\_date

FROM training\_details

JOIN users ON

training\_details.user\_id = users.user\_id

GROUP BY training\_details.user\_id,training\_details.training\_id

HAVING COUNT(training\_date) >1

ORDER BY latest\_train\_date DESC