

PL/SQL static code analysis: "UNION" should be used with caution

2 minutes

UNION is a convenient syntax to combine the results of two or more SQL statements because it helps you cut a complex problem into multiple simple SQL statements. But when it comes to execution, using UNION is debatable.

First, it may be possible to fuse two simple SQL statements into a bigger one that will run faster. Second, UNION is significantly less performant compared to UNION ALL because it removes duplicated entries and runs an internal DISTINCT to achieve this.

UNION ALL does not remove duplicates and returns all the results from the queries. It performs faster in most cases compared to UNION. Nevertheless, the quantity of data returned by UNION ALL can be significantly larger than with UNION. On a slow network, the performance gain of using UNION ALL instead of UNION can be negated by the time lost in the larger data transfer.

This rule raises an issue on each UNION. It's up to the developer to challenge its use and see if there is a better way to rewrite without UNION.

Noncompliant Code Example

```
-- case #1
SELECT EMAIL FROM EMPLOYEES WHERE COUNTRY =
'FR'
UNION
SELECT EMAIL FROM EMPLOYEES WHERE COUNTRY =
'CH'

-- case #2
-- if you care about not having duplicated entries, then UNION is
the good choice
SELECT EMAIL FROM EMPLOYEES
UNION
SELECT EMAIL FROM CUSTOMERS
```

Compliant Solution

```
-- case #1
SELECT EMAIL FROM EMPLOYEES WHERE COUNTRY =
'FR' OR COUNTRY = 'CH'

-- case #2
-- if you don't care about duplicated entries in the results of this
UNION, then UNION ALL should be preferred
SELECT EMAIL FROM EMPLOYEES
UNION ALL
SELECT EMAIL FROM CUSTOMERS
```