PL/SQL static code analysis: Native SQL joins should be used

2-3 minutes

SQL is an extremely powerful and hard to master language. It may be tempting to emulate SQL joins in PL/SQL using nested cursor loops, but those are not optimized by Oracle at all. In fact, they lead to numerous context switches between the SQL and PL/SQL engines, and those switches have a highly negative impact on performance. It is therefore much better to replace nested PL/SQL cursor loops with native SQL joins.

Noncompliant Code Example

```
SET SERVEROUTPUT ON

CREATE TABLE countriesTable(
  countryName VARCHAR2(42)
);

CREATE TABLE citiesTable(
  cityName VARCHAR2(42)
);

INSERT INTO countriesTable VALUES('India');
INSERT INTO countriesTable VALUES('Switzerland');
INSERT INTO countriesTable VALUES('United States');
```

```
INSERT INTO citiesTable VALUES('Berne');
INSERT INTO citiesTable VALUES('Delhi');
INSERT INTO citiesTable VALUES('Bangalore');
INSERT INTO citiesTable VALUES('New York');

BEGIN
  FOR countryRecord IN (SELECT countryName FROM
countriesTable) LOOP
    FOR cityRecord IN (SELECT cityName FROM citiesTable)
LOOP -- Non-Compliant
      DBMS_OUTPUT.PUT_LINE('Country: ' ll
countryRecord.countryName ll ', City: ' ll cityRecord.cityName);
    END LOOP;
  END LOOP;
END;
/

DROP TABLE citiesTable;

DROP TABLE countriesTable;
```

Compliant Solution

```
SET SERVEROUTPUT ON

CREATE TABLE countriesTable(
  countryName VARCHAR2(42)
);

CREATE TABLE citiesTable(
  cityName VARCHAR2(42)
```

```
);

INSERT INTO countriesTable VALUES('India');
INSERT INTO countriesTable VALUES('Switzerland');
INSERT INTO countriesTable VALUES('United States');

INSERT INTO citiesTable VALUES('Berne');
INSERT INTO citiesTable VALUES('Delhi');
INSERT INTO citiesTable VALUES('Bangalore');
INSERT INTO citiesTable VALUES('New York');

BEGIN
  FOR myRecord IN (SELECT * FROM countriesTable CROSS
JOIN citiesTable) LOOP -- Compliant
    DBMS_OUTPUT.PUT_LINE('Country: ' ||
myRecord.countryName || ', City: ' || myRecord.cityName);
  END LOOP;
END;
/

DROP TABLE citiesTable;

DROP TABLE countriesTable;
```