

PL/SQL static code analysis: "FUNCTIONS" should not have "OUT" parameters

2-3 minutes

Functions with OUT parameters are complex to understand. Indeed, it is impossible to tell, just by looking at the function call, whether an argument is a input or output. Moreover, functions with OUT parameters cannot be called from SQL. It is better to either break such functions up into smaller ones, which each return a single value, or to return several values at once, by combining them in a collection, record, type, or table row.

Noncompliant Code Example

```
SET SERVEROUTPUT ON
```

```
CREATE TABLE employee(  
  firstName VARCHAR2(42),  
  name VARCHAR2(42),  
  phone VARCHAR2(42)  
);
```

```
INSERT INTO employee VALUES ('John', 'Smith', '+1');
```

```
DECLARE  
  firstName VARCHAR2(42);
```

```

name VARCHAR2(42);
phone VARCHAR2(42);

-- This DOES NOT return the employee name
FUNCTION getEmployeeInfos(firstName OUT VARCHAR2,
phone OUT VARCHAR2) RETURN VARCHAR2 AS -- Non-
Compliant, confusing
    name VARCHAR2(42);
BEGIN
    SELECT firstName, name, phone INTO firstName, name,
phone FROM employee;
    RETURN name;
END;
BEGIN
    name := getEmployeeInfos(firstName, phone);

    DBMS_OUTPUT.PUT_LINE('firstName: ' || firstName);
    DBMS_OUTPUT.PUT_LINE('name: ' || name);
    DBMS_OUTPUT.PUT_LINE('phone: ' || phone);
END;
/

```

Compliant Solution

```
SET SERVEROUTPUT ON
```

```

CREATE TABLE employee(
    firstName VARCHAR2(42),
    name VARCHAR2(42),
    phone VARCHAR2(42)
);

```

```
INSERT INTO employee VALUES ('John', 'Smith', '+1');
```

```
DECLARE
```

```
emp employee%ROWTYPE;
```

```
FUNCTION getEmployeeInfos RETURN  
employee%ROWTYPE AS -- Compliant
```

```
emp employee%ROWTYPE;
```

```
BEGIN
```

```
SELECT * INTO emp FROM employee;
```

```
RETURN emp;
```

```
END;
```

```
BEGIN
```

```
emp := getEmployeeInfos;
```

```
DBMS_OUTPUT.PUT_LINE('firstName: ' || emp.firstName);
```

```
DBMS_OUTPUT.PUT_LINE('name: ' || emp.name);
```

```
DBMS_OUTPUT.PUT_LINE('phone: ' || emp.phone);
```

```
END;
```

```
/
```

```
DROP TABLE employee;
```