PL/SQL static code analysis: Pipelined functions should have at least one "PIPE ROW" statement and not return an expression (PLS-00633)

2 minutes

Pipelined functions offers the ability to create programmatically generated tables.

One of the benefits of such functions is that they reduce memory consumption as results are not all kept in memory before being returned.

Instead of relying on `RETURN`, `PIPE ROW` must be used to return the results, one row at a time.

Trying to return an expression from a pipelined function raises `PLS-00633: RETURN statement in a pipelined function cannot contain an expression`

Noncompliant Code Example

```
CREATE OR REPLACE TYPE myScalarType AS OBJECT
(
  dummy   VARCHAR2(42)
)
/
CREATE OR REPLACE TYPE myTableType AS TABLE OF
```

```
myScalarType;
/

CREATE OR REPLACE FUNCTION foo RETURN myTableType
PIPELINED AS  -- Noncompliant, should contain at least one
PIPE ROW
  result myTableType := myTableType();
BEGIN
  FOR i IN 1 .. 3 LOOP
    result.EXTEND;
    result(i) := myScalarType('Dummy ' || i);
  END LOOP;

  RETURN result;  -- Noncompliant, will raise PLS-00633
END;
/

SELECT * FROM TABLE(foo());

DROP FUNCTION foo;
DROP TYPE myTableType;
DROP TYPE myScalarType;
```

Compliant Solution

```
-- ...

CREATE OR REPLACE FUNCTION foo RETURN myTableType
PIPELINED AS
BEGIN
  FOR i IN 1 .. 3 LOOP
    PIPE ROW(myScalarType('Dummy ' || i));
```

```
  END LOOP;

  RETURN;
END;
/

-- ...
```