PL/SQL static code analysis: "EXCEPTION_INIT -20,NNN" calls should be centralized

2 minutes

Centralizing the definitions of custom exceptions comes with two major benefits:

- The duplication of the exceptions declarations and `PRAGMA EXCEPTION_INIT` is avoided
- The risk of associating multiple different exceptions to the same number is reduced

Noncompliant Code Example

```
SET SERVEROUTPUT ON

DECLARE
  user_not_found EXCEPTION;
  PRAGMA EXCEPTION_INIT(user_not_found, -20000); --
Noncompliant, user_not_found is bound to -20000
BEGIN
  NULL;
END;
/

DECLARE
```

```
  user_not_found EXCEPTION;
  PRAGMA EXCEPTION_INIT(user_not_found, -20000); --
Noncompliant, user_not_found is again bound to -20000,
duplication
BEGIN
  NULL;
END;
/

DECLARE
  wrong_password EXCEPTION;
  PRAGMA EXCEPTION_INIT(wrong_password, -20000); --
Noncompliant, wrong_password is bound to -20000, conflicting
with user_not_found
BEGIN
  NULL;
END;
/
```

Compliant Solution

```
SET SERVEROUTPUT ON

CREATE PACKAGE exceptions AS
 user_not_found EXCEPTION;
 wrong_password EXCEPTION;

 PRAGMA EXCEPTION_INIT(user_not_found, -20000); -- Non-
Compliant (flag as false-positive)
 PRAGMA EXCEPTION_INIT(wrong_password, -20001); --
Non-Compliant (flag as false-positive), conflicts are easier to
avoid
```

```
END;
/

DROP PACKAGE exceptions;
```