

Created by Ahsan Arif

Function Examples of PL/SQL

We are going to create a function that parses a string and returns a number if the string being passed is a number otherwise it returns NULL.

```
CREATE OR REPLACE FUNCTION try_parse(  
    iv_number IN VARCHAR2)  
    RETURN NUMBER IS  
BEGIN  
    RETURN TO_NUMBER(iv_number);  
    EXCEPTION  
        WHEN OTHERS THEN  
            RETURN NULL;  
END;
```

The input parameter is `iv_number` that is a `varchar2` type. We can pass any string to the function `try_parse()`. We use built-in function `to_number` to convert a string into a number. If any exception occurs, the function will return `NULL` in the exception section of the function block.

Calling PL/SQL Function

The PL/SQL function returns a value so you can use it on the right hand side of an assignment or in a `SQL SELECT` statement.

Let's create an anonymous block to use the function `try_parse()` above.

```
SET SERVEROUTPUT ON SIZE 1000000;  
DECLARE  
    n_x NUMBER;  
    n_y NUMBER;  
    n_z NUMBER;  
BEGIN  
    n_x := try_parse('574');  
    n_y := try_parse('12.21');  
    n_z := try_parse('abcd');  
  
    DBMS_OUTPUT.PUT_LINE(n_x);  
    DBMS_OUTPUT.PUT_LINE(n_y);  
    DBMS_OUTPUT.PUT_LINE(n_z);  
END;  
/
```

Or you can also use the function `try_parse` in the SQL `SELECT` statement as follows:

1. `SELECT try_parse('1234') FROM dual;`
- 2.
3. `SELECT try_parse('Abc') FROM dual;`