

SQL Functions in PL/SQL: Examples

– Get the length of a string:

```
v_desc_size INTEGER(5);  
v_prod_description VARCHAR2(70) := 'You can use this product with your  
radios for higher frequency';
```

```
– get the length of the string in prod_description  
v_desc_size := LENGTH(v_prod_description);
```

– Get the number of months an employee has worked:

```
v_tenure := MONTHS_BETWEEN(CURRENT_DATE, v_hiredate);
```

ORACLE

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

SQL Functions in PL/SQL

SQL provides several predefined functions that can be used in SQL statements. Most of these functions (such as single-row number and character functions, data type conversion functions, and date and time-stamp functions) are valid in PL/SQL expressions.

The following functions are not available in procedural statements:

- DECODE

Group functions: AVG, MIN, MAX, COUNT, SUM, STDDEV, and VARIANCE

Group functions apply to groups of rows in a table and are, therefore, available only in SQL statements in a PL/SQL block. The functions mentioned here are only a subset of the complete list.

SQL Functions in PL/SQL: Examples

You can use SQL functions to manipulate data. These functions are grouped into the following categories:

- Number

- Character

Using Sequences in PL/SQL Expressions

Starting in 11g:

```
DECLARE
  v_new_id NUMBER;
BEGIN
  v_new_id := my_seq.NEXTVAL;
END;
/
```

Before 11g:

```
DECLARE
  v_new_id NUMBER;
BEGIN
  SELECT my_seq.NEXTVAL INTO v_new_id FROM Dual;
END;
/
```

ORACLE

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

Conversion

Date

Miscellaneous

Accessing Sequence Values

In Oracle Database 11g, you can use the NEXTVAL and CURRVAL pseudocolumns in any PL/SQL context, where an expression of the NUMBER data type may legally appear. Although the old style of using a SELECT statement to query a sequence is still valid, it is recommended that you do not use it.

Before Oracle Database 11g, you were forced to write a SQL statement in order to use a sequence object value in a PL/SQL subroutine. Typically, you would write a SELECT statement to reference the pseudocolumns of NEXTVAL and CURRVAL to obtain a sequence number. This method created a usability problem.

In Oracle Database 11g, the limitation of forcing you to write a SQL statement to retrieve a sequence value is eliminated. With the sequence enhancement feature:

- Sequence usability is improved

- The developer has to type less

- The resulting code is clearer