

# Functions, Cursors and Procedures: A Case Study

Prof. Rukhsar Haji<sup>1</sup>, Prof. Ronica Raj<sup>2</sup>, Naushin Khan<sup>3</sup>, Farheen Rizvi<sup>4</sup>

<sup>1</sup>Professor, Dept. of Computer Engineering, Rizvi College of Engineering, Maharashtra, India

<sup>2</sup>Professor, Dept. of Computer Engineering, Rizvi College of Engineering, Maharashtra, India

<sup>3</sup>Student, Dept. of Computer Engineering, Rizvi College of Engineering, Maharashtra, India

<sup>4</sup>Student, Dept. of Computer Engineering, Rizvi College of Engineering, Maharashtra, India

**Abstract** - PL/SQL may be a Procedural Language extension of SQL. It overcomes the gap between database technology and procedural programming languages. PL/SQL could even be a mix of SQL in concomitance with the procedural features of programming languages. It was developed by Oracle Corporation within the early 90's to enhance the capabilities of SQL. A PL/SQL language code can be stored in the client system or in the database (server-side). The concepts like cursors, functions and stored procedures are often utilized in other database systems like Sybase, Microsoft SQL server etc., with some change in SQL syntax.

**Key Words:** functions, procedures, cursors, Oracle, PL/SQL

## 1. INTRODUCTION

In [3] PL/SQL is a block-structured language. Procedures, functions and anonymous blocks are the basic units that make up a PL/SQL program. These are logical blocks, which can contain any number of nested sub-blocks. Typically, each logical block is equivalent to a problem or subproblem to be solved.<sup>[5]</sup> Thus, PL/SQL takes the divide-and-conquer approach to problem solving called stepwise refinement.



Fig - 1: Block Structure of PL/SQL

As figure 1 shows, a PL/SQL block has three parts: a declarative part, an executable part, and an exception-handling part.

PL/SQL is a completely portable, high-performance transaction processing language that offers support for SQL. PL/SQL allows you to use SQL statements to control

Oracle data, declare constants and variables, define procedures and functions, and trap runtime errors. A variety of terms are included in SQL as its scope and application area is really vast. They are as follows:

**Functions:** Functions could be a standalone PL/SQL subprogram. Like PL/SQL procedure, functions have a singular name by which it may be referred. These are stored as PL/SQL database objects. Below are a number of the characteristics of functions. Functions are a standalone block that's mainly used for calculation purpose. Function use RETURN keyword to return the worth, and also the datatype of this is often defined at the time of creation. A Function should either return a worth or raise the exception, i.e. return is mandatory in functions.

**Procedures:** Procedures and Functions are the subprograms which may be created and saved within the database as database objects. They'll be called or referred inside the opposite blocks also. Procedures are standalone blocks of a program which will be stored within the database. Call to those procedures is made by relating their name, to execute the PL/SQL statements.

**Cursors:** To execute SQL statements, a piece area is employed by the Oracle engine for its internal processing and storing the knowledge. This work area is private to SQL's operations. The 'Cursor' is that the PL/SQL construct that enables the user to call the work area and access the stored information in it.

## 2. FUNCTIONS

In [5], A stored function (also called a user function or user-defined function) is referred to as a set of PL/SQL statements you can call by name. Stored functions are akin to procedures, except that a function returns a value to the environment in which it is called. User functions are often used as a part of a SQL expression. It can have nested blocks, or it is often defined and nested inside the opposite blocks or packages.

A function contains declaration part, execution part, exception handling part. The values are often passed into the function or fetched from the procedure through the parameters. These parameters should be included within the calling statement. Function also can return the worth through OUT parameters aside from using RETURN. Since this always returns the value, in calling statement it accompanies with assignment operator to populate the variables.

```
Syntax:
CREATE OR REPLACE FUNCTION
<procedure_name>
(
  <parameter1 IN/OUT <datatype>
  ..
  )
RETURN <datatype>
[ IS | AS ]
<declaration_part>
BEGIN
  <execution part>
EXCEPTION
  <exception handling part>
END;
```

Fig - 2: Syntax for function.

```
create database Companys
use Companys

Create Table E1
(E_id int primary key not null,
E_name varchar(10) not null,
Salary money)

insert into E1 values(1, 'RAM',45000)
insert into E1 values(2, 'SHYAM',56000)
insert into E1 values(3, 'Navneet',80000)
insert into E1 values(4, 'Ganga',30000)
insert into E1 values(5, 'John',87000)

Create procedure E
as
select E_name from E1
go
exec E
```

Fig - 4: Example for procedure

### 3. PROCEDURES

A procedure is another name for a PL/SQL block that can accept parameters (sometimes mentioned as arguments). Generally, a procedure is to perform some action. It can be invoked by using the procedure name in the execution section of another PL/SQL block. Procedure consists of a header, a declaration section, an executable section, and an optional exception-handling section.

A procedure is compiled and stored inside the database in form of a schema object. If you are using the procedures with Oracle Forms and Reports, they can be compiled within the Oracle Forms or Oracle Reports executables. Procedures promote reusability and maintainability. When validated, they can be used in any number of applications. If the requirements change, only the procedure needs to be updated.

```
CREATE [OR REPLACE] PROCEDURE procedure_name
[(parameter_name [IN | OUT | IN OUT] type [, ...])]
[IS | AS]
BEGIN
  <procedure_body >
END
procedure_name;
```

Fig - 3: Syntax for procedure.

### 4. CURSORS

To execute SQL statements, the Oracle engine uses a work space for its internal processing and storing the information. This work area is solely for SQL operations. The 'Cursor' is one kind of a PL/SQL block that allows the user to name the work space and access the information that it stores.

The data that the cursor stored is known as Active Data Set. However, conceptually the size of the cursor in memory is the size required to hold the number of rows in the Active Data Set.<sup>[9]</sup> The actual size, however, is determined by the oracle engines built in memory management capabilities and the amount of RAM available.

Unlike the SQL commands which operate on all the rows in the result set at one time, the major function of a cursor is to retrieve data, one row at a time, from a result set. Cursors come into picture when the user needs to update records in a singleton fashion or in a row by row manner, in a database table.

Oracle DBMS has yet another predefined area in the main memory Set, within which the cursors are opened. Hence the size of the cursor limits to the size of this pre-defined area.

```

DECLARE cursorName CURSOR FOR
select_statement;

OPEN cursorName;

FETCH cursorName INTO var_name;

CLOSE cursorName;
    
```

Fig - 5: Syntax for cursor.

```

create database CompanyS
use CompanyS
create Table E2
(E_id int primary key not null,
E_name varchar(10) not null,
Salary money)

insert into E2 values(1, 'RAM', 45000)
insert into E2 values(2, 'SHYAM', 56000)
insert into E2 values(3, 'Navneet', 80000)
insert into E2 values(4, 'Ganga', 30000)
insert into E2 values(5, 'John', 87000)

select * from E2

declare W cursor
for select E_id from E2
open W

Fetch W
    
```

Fig - 6: Example for cursor.

### 4.1 Cursor Actions

- 1) *Declare Cursor:* To declare a cursor you need to define the SQL statement that returns a result set.
- 2) *Open:* A Cursor is opened and populated as the SQL statement is defined by the cursor are executed.
- 3) *Fetch:* When the cursor is opened, rows can be fetched from the cursor one by one or in a block to perform data manipulation.
- 4) *Close:* After completing the data manipulation, close the cursor explicitly.
- 5) *Deallocate:* Finally, delete the cursor definition and release all the system resources associated with the cursor.

### 4.2 Cursor Types

Cursors are classified depending on the circumstances in which they are opened.

**Implicit Cursor:** If the Oracle engine opened a cursor for its internal processing it is known as an Implicit Cursor. It

is a predefined creation for the user by Oracle when a query is executed and is simpler to code.

**Explicit Cursor:** Another way to open a cursor for processing data is through a PL/SQL block, on demand. These are user-defined cursors that are known as Explicit Cursor

## 5. COMPARISON

Table - 1: Sample Table format

Functions	Procedures	Cursors
A subprogram that computes a value.	A subprogram that performs a particular action.	It is a pointer to the memory location on DB server.
Must return a single value.	May or may not return the value.	When we execute a query a relation is returned.
Invoke as a part of an expression.	Execute as a PL/SQL statement.	Used to retrieve and manipulate DB data in PL/SQL statement

## 6. ADVANTAGES

### 6.1 Cursors

- 1) Cursors help us perform row by row processing so we can do row wise validation or operations on each row.
- 2) Cursors provide the first few rows before the whole result set being assembled. In instance of no cursor, the entire result set needs to be delivered before any rows are displayed by the application, whereas cursor allows to achieve better response time.
- 3) If we make updates without using cursors in your application then sending separate SQL statements to the database server is required to apply the changes. This perhaps might cause the possibility of concurrency problems, if the result set has changed since it was queried by the client. This in turn, augments the possibility of lost updates. So using cursor, concurrency Control is attained better.
- 4) Cursors can be agile as compared to a while loop but at the cost of more overhead.

### 6.2 Functions and Procedures

Following are some benefits of stored procedure and function in PL/SQL:

1. Improves Database Performance:

Compilation is automatically done by oracle engine. While calling a procedure or function, the Oracle engine loads the compiled code into a memory area called System Global Area (SGA) due to which execution becomes faster.

2. Provides Reusability and avoids redundancy:

The same block of code for procedure or function can be called any number of times for working on multiple data. Due to which number of lines of code cannot be written repeatedly.

3. Maintains Integrity:

Integrity means accuracy. Use of procedure or function corroborates integrity because they are stored as database objects by the Oracle Engine.

4. Maintains Security:

Use of stored procedure or function helps in maintaining the security of the database as access to them and their usage can be controlled by granting access/permission to users while the permission to change or to edit or to manipulate the database may not be granted to users.

5. Saves Memory

Stored procedure or function has shared memory. Due to which it saves memory as a single copy of either a procedure or a function can be loaded for execution by any number of users who have access permission.

**7. CONCLUSIONS**

The design of PL/SQL done by Oracle’s Corporation bridged the gap between database technology and procedural programming languages. PL/SQL is a block-structure procedural programming language. PL/SQL supports the concept of cursor management. It provides two types of cursor, i.e. Implicit and Explicit. Cursors help us perform row by row processing, thus enabling to do row wise validation or operations on each row. Another feature provided by PL/SQL is to define functions and procedures. Use of these stored procedures or functions benefits in maintaining security and integrity. The same block of code for procedure or function can be called any number of times for working on multiple data. Along with allowing reusability and eschewing redundancy, functions and procedures also save memory as they have shared memory.

**REFERENCES**

[1] Anthony Molinaro, SQL Cookbook, 2nd ed., USA: O’reilly, 2005.  
 [2] J. Breckling, Ed., The Analysis of Directional Time Series: Applications to Wind Speed and Direction, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.  
 [3] Arshi Gouhar, “Database Management System,” IJESC Department of Computer Science & Engineering, vol. 7, Issue 5, May 2017.  
 [4] S. Abiteboul and S. Grumbach. A rule-based language with functions and sets. ACM Trans. on Database

Systems, 16(1):1–30, 1991. R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, “High-speed digital-to-RF converter,” U.S. Patent 5 668 842, Sept. 16, 1997.  
 [5] Tom Portfolio, John Russell, PL/SQL User’s Guide and Reference, 9th ed., Oracle Corporation, 2001.  
 [6] The IEEE website. [Online]. Available: <http://www.ieee.org/>  
 [7] The Tutorial link website. [Online] Available: <https://tutorialink.com/dbms/cursors.dbms>  
 [8] The PL/SQL link website [Online] Available: <https://plsql-tutorial.com/index.html>  
 [9] Dr. Laxmi Joshi, “Cursor Management in PL/SQL”, published by: SK publisher, vol.1, Issue 1, Nov. 2014.  
 [10] The Oracle link website [Online] Available: [https://docs.oracle.com/cd/E11882\\_01/server.112/e41084/statements\\_5011.htm#SQLRF01208](https://docs.oracle.com/cd/E11882_01/server.112/e41084/statements_5011.htm#SQLRF01208)

**BIOGRAPHIES**



Professor Rukhsar Haji is professor in Rizvi College of Engineering which is affiliated with Mumbai University. She has done Masters in Computer Engineering from Thadomal College. She have 6 years of teaching experience.



Ms. Ronica Raj is working as an Assistant Professor in Rizvi College of Engineering. She has completed Masters of Engineering in Computer Engineering from Thakur College of Engineering.



Ms. Naushin Khan is pursuing Bachelor of Engineering in Computer Engineering from Rizvi college of Engineering which is affiliated with the Mumbai University. She is aiming to pursue web page development for further studies.



Ms. Farheen Rizvi pursued Diploma in Computer Engineering from Government Polytechnic Mumbai which is an Autonomous Institute. She is currently pursuing Bachelor of Engineering in Computer Engineering from Rizvi College of Engineering which is affiliated with the Mumbai University.