

Oracle 1.2 PL/SQL Developer

Baktagul Imasheva, senior-lecturer
b.imasheva@iitu.kz

Content

What is PL/SQL?

What is PL/SQL?

- It is a highly structured, readable, and accessible language
- It is a standard and portable language for Oracle development
- It is an embedded language It is a high-performance, highly integrated database language

Integration with SQL

```
1  DECLARE
2      l_book_count INTEGER;
3
4  BEGIN
5      SELECT COUNT(*)
6          INTO l_book_count
7          FROM books
8          WHERE author LIKE '%FEUERSTEIN, STEVEN%';
9
10     DBMS_OUTPUT.PUT_LINE (
11         'Steven has written (or co-written) ' ||
12         l_book_count ||
13         ' books. ');
14
15     -- Oh, and I changed my name, so...
16     UPDATE books
17         SET author = REPLACE (author, 'STEVEN', 'STEPHEN')
18         WHERE author LIKE '%FEUERSTEIN, STEVEN%';
19 END;
```

Difference in 12c

- Installation And Configurations
- Multi-tenancy (MSI) and Pluggable BI
- BI Application Archive(BAR)
- OBIEE Server level enhancements – In-memory analytics and Advanced analytics
- Data Mashups for Answers
- Visual Analyser
- Better Mobile Analytics

<https://debaatobiee.wordpress.com/tag/in-memory-analysis-obiee-12c/>

Running PL/SQL Code

- Use an IDE
- Run scripts in a command line environment

Using a command line

- *OS> sqlplus*
- *OS> sqlplus bob/swordfish*
- Running a SQL Statement *SELECT isbn, author, title FROM books;*
- Running a PL/SQL Statement

```
SQL> BEGIN
  2      DBMS_OUTPUT.PUT_LINE('Hey look, ma!');
  3  END;
  4  /
```

PL/SQL procedure successfully completed.

```
SQL>
```


Performing Essential PL/SQL Tasks

- Creating stored program

```
CREATE FUNCTION wordcount (str IN VARCHAR2)
RETURN PLS_INTEGER
AS
    declare local variables here
BEGIN
    implement algorithm here
END;
/
```

- Creation of file and run it **SQL> @wordcount.fun**
- To see source code **SQL> SET ECHO ON** **SQL>**
@wordcount.fun
- SQL*Plus' SHOW ERRORS command **SQL> SHO ERR**
- Showing Stored Programs **SQL> DESCRIBE word count**
- Managing grants and privileges **GRANT EXECUTE ON wordcount TO scott;**
REVOKE EXECUTE ON wordcount FROM scott;

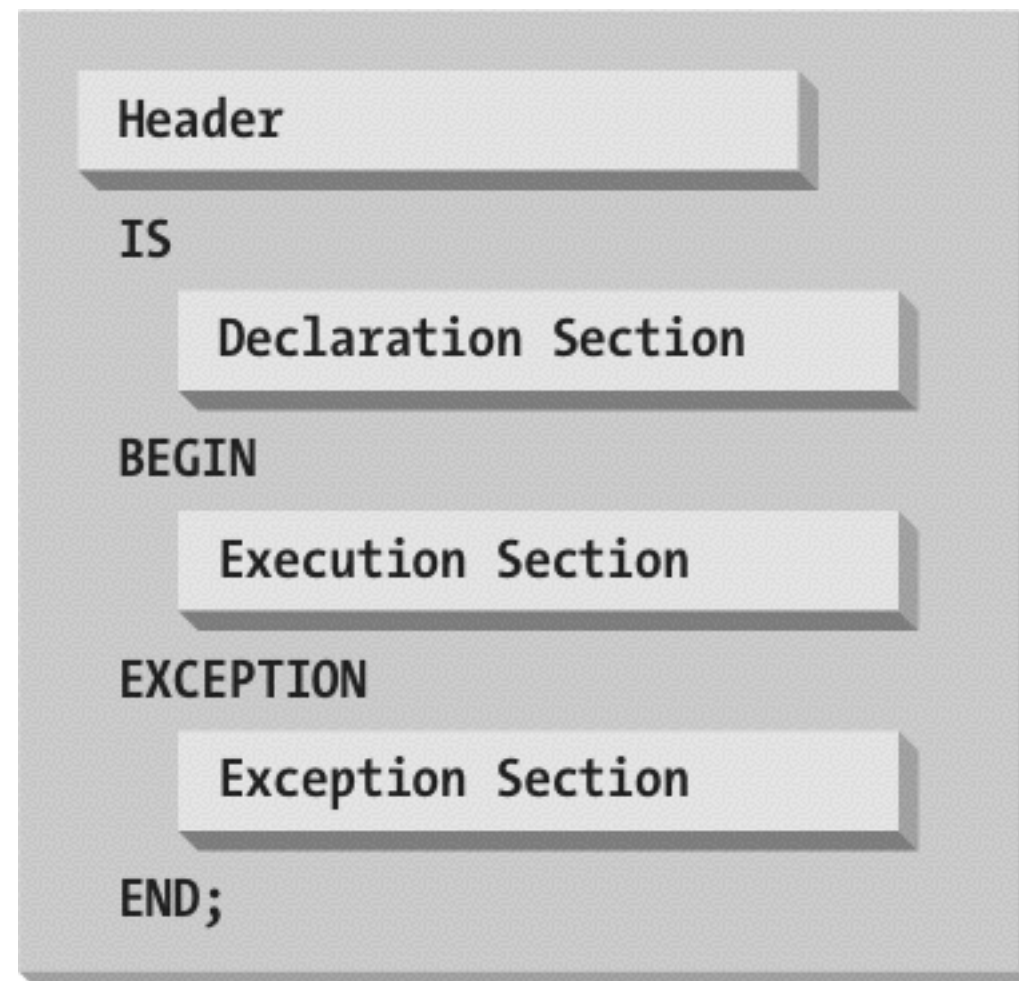
Editing Environments for PL/SQL

Product	Description
Toad	Offered by Quest Software, Toad is far and away the most popular PL/SQL IDE. It is used by hundreds of thousands of developers, in both its free and commercial versions. For more information, see http://www.quest.com/toad-for-oracle/ .
SQL Navigator	Also offered by Quest Software, SQL Navigator is also used by tens of thousands of developers who love the product's interface and productivity features. For more information, see http://www.quest.com/sql-navigator/ .
PL/SQL Developer	This product is sold by Allround Automations, and is a favorite of many PL/SQL developers. It is built around a plug-in architecture, so third parties can offer extensions to the base product. For more information, see http://www.allroundautomations.com/plsqldev.html .
SQL Developer	After years of little or no support for PL/SQL editing, Oracle Corporation created SQL Developer as a "fork" of the foundation JDeveloper tool. SQL Developer is free and increasingly robust. For more information, see http://www.oracle.com/technology/software/products/sql/index.html .

Calling PL/SQL from Other Languages

- C, using Oracle's precompiler (Pro*C)
- Java, using JDBC
- Perl, using Perl DBI and DBD::Oracle
- PHP
- PL/SQL Server Pages

Language Fundamentals



- Block-structured languages

What is the difference?

```
PROCEDURE get_happy (ename_in IN VARCHAR2) *  
IS  
    l_hiredate DATE; *  
BEGIN  
    l_hiredate := SYSDATE - 2; *  
    INSERT INTO employee *  
        (emp_name, hiredate)  
    VALUES (ename_in, l_hiredate);  
EXCEPTION  
    WHEN DUP_VAL_IN_INDEX *  
    THEN *  
        DBMS_OUTPUT.PUT_LINE *  
            ('Cannot insert.');
```

END;

```
BEGIN *  
    DBMS_OUTPUT.PUT_LINE ('Hello world'); *  
END;
```

What kind of Block?

```
DECLARE
    CURSOR emp_cur IS ...;
BEGIN
    DECLARE
        total_sales NUMBER;
    BEGIN
        DECLARE
            l_hiredate DATE;
        BEGIN
            ...
        END;
    END;
END;
```


Scope

```
PACKAGE scope_demo
IS
    g_global    NUMBER;

    PROCEDURE set_global (number_in IN NUMBER);
END scope_demo;

PACKAGE BODY scope_demo
IS
    PROCEDURE set_global (number_in IN NUMBER)
    IS
        l_salary  NUMBER := 10000;
        l_count   PLS_INTEGER;
    BEGIN

        <<local block>>
        DECLARE
            l_inner    NUMBER;
        BEGIN
            SELECT COUNT (*)
            INTO l_count
            FROM employees
            WHERE department_id = l_inner AND salary > l_salary;
        END local_block;

        g_global := number_in;
    END set_global;
END scope_demo;
```

Qualify all References

```
PACKAGE BODY scope_demo
IS
  PROCEDURE set_global (number_in IN NUMBER)
  IS
    l_salary  NUMBER := 10000;
    l_count   PLS_INTEGER;
  BEGIN

    <<local_block>>
    DECLARE
      l_inner  PLS_INTEGER;
    BEGIN
      SELECT COUNT (*)
      INTO set_global.l_count
      FROM employees e
      WHERE e.department_id = local_block.l_inner
      AND e.salary > set_global.l_salary;
    END local_block;

    scope_demo.g_global := set_global.number_in;
  END set_global;
END scope_demo;
```


Try to qualify correctly

```

PACKAGE BODY scope_demo
IS
    PROCEDURE set_global (number_in IN NUMBER)
    IS
        l_salary  NUMBER := 10000;
        l_count    PLS_INTEGER;
    BEGIN

        <<local_block>>
        DECLARE
            l_inner  PLS_INTEGER;
        BEGIN
            SELECT COUNT (*)
            INTO set_global.l_count
            FROM employees e
            WHERE e.department_id = local_block.l_inner
            AND e.salary > set_global.l_salary;
        END local_block;

        scope_demo.g_global := set_global.number_in;
    END set_global;
END scope_demo;

```

```
SELECT COUNT (*)
  INTO l_count
  FROM employees
 WHERE department_id = 1 inner AND salary > 1 salary;
```

Visibility

- whether you can refer to it using only its name, or whether you need to attach a prefix in front of it.

```
PACKAGE company_pkg
```

```
IS
```

```
    last_company_id NUMBER; ...
```

```
END company_pkg;
```

```
IF new_company_id = company_pkg.last_company_id THEN
```

```
PROCEDURE calc_totals
IS
    salary NUMBER;
BEGIN
    ...
    DECLARE
        salary NUMBER;
    BEGIN
        salary := calc_totals.salary;
    END;
    ...
END;
```