

Introduction to PL/SQL

Introduction to PL/SQL

- Procedural Language extension for SQL
- Oracle Proprietary
- 3GL Capabilities
- Integration of SQL
- Portable within Oracle data bases
- Callable from any client

Structure of PL/SQL

- PL/SQL is Block Structured

A block is the basic unit from which all PL/SQL programs are built. A block can be named (functions and procedures) or anonymous

- Sections of block
 - 1- Header Section
 - 2- Declaration Section
 - 3- Executable Section
 - 4- Exception Section

Structure of PL/SQL

HEADER

Type and Name of block

DECLARE

Variables; Constants; Cursors;

BEGIN

PL/SQL and SQL Statements

EXCEPTION

Exception handlers

END;

Structure of PL/SQL

DECLARE

 a number;

 text1 varchar2(20);

 text2 varchar2(20) := "HI";

BEGIN

END;

Important Data Types in PL/SQL include
NUMBER, INTEGER, CHAR, VARCHAR2, DATE
etc

to_date('02-05-2007','dd-mm-yyyy') { Converts
String to Date}

Structure of PL/SQL

- Data Types for specific columns

Variable_name Table_name.Column_name%type;

This syntax defines a variable of the type of the referenced column on the referenced table

PL/SQL Control Structure

- PL/SQL has a number of control structures which includes:
 - Conditional controls
 - Iterative or loop controls.
 - Exception or error controls
- It is these controls, used singly or together, that allow the PL/SQL developer to direct the flow of execution through the program.

PL/SQL Control Structure

- **Conditional Controls**

IF....THEN....END IF;

IF....THEN...ELSE....END IF;

IF....THEN...ELSIF....THEN....ELSE....END IF;

PL/SQL Control Structure

- LOOP
 ...SQL Statements...
 EXIT;
END LOOP;
- WHILE loops
- WHILE condition LOOP
 ...SQL Statements...
END LOOP;
- FOR loops
- FOR <variable(numeric)> IN [REVERSE]
 <lowerbound>..<>upperbound> LOOP END LOOP;

PL/SQL Control Structure

- **Cursor**

```
DECLARE
    name varchar2(20);
    Cursor c1 is
    select t.name
    from table t
    where date is not null;
BEGIN
    OPEN c1;
    LOOP
        FETCH c1 into name;
        exit when c1%NOTFOUND;
    END LOOP;
    CLOSE c1;
END;
```

Debugging

- show error
- `DBMS_OUTPUT.PUT_LINE(' .. ');`

Execution

- How to execute a function in PL/SQL?

```
Var issue_flag number;
```

```
exec :issue_flag:=fun_name(arg1,arg2,. . . .);
```

```
PRINT :issue_flag;
```

- How to execute a procedure in PL/SQL?

```
Exec procedure_name(arg1,arg2,. . . .);
```

THANK YOU

References

- [1] http://www.csee.umbc.edu/help/oracle8/server.815/a67842/01_oview.htm#740
- [2] <http://en.wikipedia.org/wiki/PL/SQL>
- [3] http://www.skillbuilders.com/Tutorials-V2/ora9i_Intro_to_PLSQL/ora9i_Intro_to_PLSQL.cfm