

Database Systems

Laboratory 9

PL/SQL

PL/SQL Basics

Advantages of PL/SQL

Structure of PL/SQL Block

Character Sets, Comments,
Special operators

Variables, Constant,
BOOLEAN Datatype,
%TYPE

User Message

Conditional Logic

CASE Statement

Looping Structure

Scope of variable in
nested block

DEFINE Statement

Chittaranjan Pradhan
School of Computer Engineering,
KIIT University

1 PL/SQL Basics

Advantages of PL/SQL

Structure of PL/SQL Block

Character Sets, Comments, Special operators

Variables, Constant, BOOLEAN Datatype, %TYPE

User Message

2 Conditional Logic

CASE Statement

3 Looping Structure

4 Scope of variable in nested block

5 DEFINE Statement

PL/SQL Basics

Advantages of PL/SQL

Structure of PL/SQL Block

Character Sets, Comments,
Special operators

Variables, Constant,
BOOLEAN Datatype,
%TYPE

User Message

Conditional Logic

CASE Statement

Looping Structure

Scope of variable in
nested block

DEFINE Statement

PL/SQL Basics

- It is the procedural language extension of SQL
- It is a block - structured language that enables developers to combine the power of SQL with procedural statements
- It bridges the gap between database technology and procedural programming languages
- PL/SQL also possesses features of object oriented languages like: Data encapsulation, Error handling, Data hiding and Object oriented programming

PL/SQL Basics

Advantages of PL/SQL

Structure of PL/SQL Block

Character Sets, Comments,
Special operators

Variables, Constant,
BOOLEAN Datatype,
%TYPE

User Message

Conditional Logic

CASE Statement

Looping Structure

Scope of variable in
nested block

DEFINE Statement

Advantages of PL/SQL

- It not only supports SQL data manipulation, but also provides facilities of conditional checking, branching & looping
- It sends an entire block of statements to the oracle engine at one time. Thus, reduces network traffic
- Permits dealing with errors
- It allows declaration & use of variables in blocks of code
- Applications written in PL/SQL are portable to any computer hardware and operating system

PL/SQL Basics

Advantages of PL/SQL

Structure of PL/SQL Block

Character Sets, Comments, Special operators

Variables, Constant, BOOLEAN Datatype, %TYPE

User Message

Conditional Logic

CASE Statement

Looping Structure

Scope of variable in nested block

DEFINE Statement

Structure of PL/SQL Block

Structure of PL/SQL Block

- DECLARE: declaration and initialization of variables and constants
- BEGIN: implementation of actual programming logic
- EXCEPTION: deals with runtime errors
- END: end of PL/SQL block

DECLARE

Declarations of variables and constants

BEGIN

PL/SQL executable statements

EXCEPTION

PL/SQL codes to handle runtime errors

END;

Character Sets, Comments, Special operators

Character Sets

- **Letters:** A...Z, and a...z
- **Numbers:** 0...9
- **Symbols:** ~ ! @ # \$ % ^ & * () - + { [] } | : ; « , > . ? / etc

Comments

- **Single line:** - -
- **Multi line:** /* ... */

Special operators

- **Assignment operator:** :=
- **Comparison operator:** =
- **Concatenation symbol:** ||
- **Exponents:** **

PL/SQL Basics

Advantages of PL/SQL

Structure of PL/SQL Block

Character Sets, Comments, Special operators

Variables, Constant, BOOLEAN Datatype, %TYPE

User Message

Conditional Logic

CASE Statement

Looping Structure

Scope of variable in nested block

DEFINE Statement

Variables, Constant, BOOLEAN Datatype, %TYPE

Variables

Variable name must begin with a character. Variable can hold at most 30 characters

Varname datatype[size];

Ex: Input_no NUMBER(3);

Constant

Varname CONSTANT datatype[size]:=value;

Ex: Input_no NUMBER(3) :=16;

Ex: Input_no CONSTANT NUMBER(3) :=16;

BOOLEAN Datatype

Logical data type, either TRUE or FALSE or NULL only

%TYPE

Describes the data type of the table column

identifier tablename.columnname%TYPE;

Ex: name Employee.empname%TYPE;

PL/SQL Basics

Advantages of PL/SQL

Structure of PL/SQL Block

Character Sets, Comments, Special operators

Variables, Constant, BOOLEAN Datatype, %TYPE

User Message

Conditional Logic

CASE Statement

Looping Structure

Scope of variable in nested block

DEFINE Statement

PL/SQL Basics

Advantages of PL/SQL

Structure of PL/SQL Block

Character Sets, Comments,
Special operators

Variables, Constant,
BOOLEAN Datatype,
%TYPE

User Message

Conditional Logic

CASE Statement

Looping Structure

Scope of variable in
nested block

DEFINE Statement

User Message

Oracle has a built-in package called DBMS_OUTPUT with the procedure PUT_LINE to print

DBMS_OUTPUT.PUT_LINE (message);

Message is generally of varchar

Before using PUT_LINE(), you should switch ON the output buffer as:

SET SERVEROUTPUT ON;

Simple IF

```
IF condition THEN  
    Statements  
END IF;
```

Simple IF-ELSE

```
IF condition THEN  
    Statements  
ELSE  
    Statements  
END IF;
```

PL/SQL Basics

- Advantages of PL/SQL
- Structure of PL/SQL Block
- Character Sets, Comments, Special operators
- Variables, Constant, BOOLEAN Datatype, %TYPE
- User Message

Conditional Logic

- CASE Statement

Looping Structure

- Scope of variable in nested block

- DEFINE Statement

Conditional Logic...

IF-ELSE Ladder

```
IF condition THEN
    Statements
ELSIF condition THEN
    Statements
ELSE
    Statements
END IF;
```

Nested IF-ELSE

```
IF condition THEN
    IF condition THEN
        Statements
    END IF
ELSE
    Statements
END IF;
```

PL/SQL Basics

- Advantages of PL/SQL
- Structure of PL/SQL Block
- Character Sets, Comments, Special operators
- Variables, Constant, BOOLEAN Datatype, %TYPE
- User Message

Conditional Logic

- CASE Statement

Looping Structure

- Scope of variable in nested block

DEFINE Statement

PL/SQL Basics

- Advantages of PL/SQL
- Structure of PL/SQL Block
- Character Sets, Comments, Special operators
- Variables, Constant, BOOLEAN Datatype, %TYPE
- User Message

Conditional Logic

- CASE Statement

Looping Structure

- Scope of variable in nested block

DEFINE Statement

Input a number and check whether it is even or odd

```
DECLARE
    n number(5);
BEGIN
    n: =&n;
    IF(MOD (n,2)=0) THEN
        DBMS_OUTPUT.PUT_LINE('even');
    ELSE
        DBMS_OUTPUT.PUT_LINE('odd');
    END IF;
END;
```

Conditional Logic...

Write a PL/SQL block that will accept a client_no from the user and adds 500 rupees to the bal_due column if the bal_due column has a value less than 5000 rupees. The process is fired on the Client_Master table

```
DECLARE
    bal Client_Master.bal_due%TYPE;
    cno Client_Master.client_no%TYPE;
    addn number(4) := 500;

BEGIN
    cno := &cno;
    SELECT bal_due INTO bal FROM Client_Master
        WHERE client_no=cno;
    IF bal<5000 THEN
        UPDATE Client_Master SET bal_due=bal_due+addn
            WHERE client_no=cno;
    END IF;
END;
```

PL/SQL Basics

Advantages of PL/SQL

Structure of PL/SQL Block

Character Sets, Comments,
Special operators

Variables, Constant,
BOOLEAN Datatype,
%TYPE

User Message

Conditional Logic

CASE Statement

Looping Structure

Scope of variable in
nested block

DEFINE Statement

PL/SQL Basics

Advantages of PL/SQL

Structure of PL/SQL Block

Character Sets, Comments,
Special operators

Variables, Constant,
BOOLEAN Datatype,
%TYPE

User Message

Conditional Logic

CASE Statement

Looping Structure

Scope of variable in
nested block

DEFINE Statement

CASE Statement

```
CASE selector  
  WHEN  $exp_1$  THEN  $Statement_1$   
  WHEN  $exp_2$  THEN  $Statement_2$   
  ...  
  WHEN  $exp_n$  THEN  $Statement_n$   
  ELSE  $Statement_{n+1}$   
END CASE;
```

CASE Statement...

Input a 1-digit number and display the name

```
DECLARE
    vdigit NUMBER(1);
BEGIN
    vdigit: =&vdigit;
    CASE vdigit
        WHEN 0 THEN DBMS_OUTPUT.PUT_LINE ('ZERO');
        WHEN 1 THEN DBMS_OUTPUT.PUT_LINE ('ONE');
        WHEN 2 THEN DBMS_OUTPUT.PUT_LINE ('TWO');
        WHEN 3 THEN DBMS_OUTPUT.PUT_LINE ('THREE');
        WHEN 4 THEN DBMS_OUTPUT.PUT_LINE ('FOUR');
        WHEN 5 THEN DBMS_OUTPUT.PUT_LINE ('FIVE');
        WHEN 6 THEN DBMS_OUTPUT.PUT_LINE ('SIX');
        WHEN 7 THEN DBMS_OUTPUT.PUT_LINE ('SEVEN');
        WHEN 8 THEN DBMS_OUTPUT.PUT_LINE ('EIGHT');
        WHEN 9 THEN DBMS_OUTPUT.PUT_LINE ('NINE');
        ELSE DBMS_OUTPUT.PUT_LINE (vdigit||' is wrong input');
    END CASE;
END;
```

PL/SQL Basics

Advantages of PL/SQL

Structure of PL/SQL Block

Character Sets, Comments,
Special operators

Variables, Constant,
BOOLEAN Datatype,
%TYPE

User Message

Conditional Logic

CASE Statement

Looping Structure

Scope of variable in
nested block

DEFINE Statement

Looping Structure

Basic Loop/Simple LOOP

```
LOOP
    Statements
END LOOP;
```

The EXIT statement provides a way to stop the iterative loop:

- EXIT WHEN $i > 10$;
- IF $i > 10$ THEN EXIT;
END IF;

WHILE LOOP

```
WHILE condition
LOOP
    Statements
END LOOP;
```

PL/SQL Basics

Advantages of PL/SQL

Structure of PL/SQL Block

Character Sets, Comments,
Special operators

Variables, Constant,
BOOLEAN Datatype,
%TYPE

User Message

Conditional Logic

CASE Statement

Looping Structure

Scope of variable in
nested block

DEFINE Statement

PL/SQL Basics

Advantages of PL/SQL

Structure of PL/SQL Block

Character Sets, Comments,
Special operators

Variables, Constant,
BOOLEAN Datatype,
%TYPE

User Message

Conditional Logic

CASE Statement

Looping Structure

Scope of variable in
nested block

DEFINE Statement

FOR LOOP

```
FOR variable IN [REVERSE] low.. high  
LOOP  
    Statements  
END LOOP;
```


Scope of variable in nested block

Scope of variable in nested block

PL/SQL statements can be nested wherever an executable statement is allowed. A nested block acts like a statement and executable statements can be broken into smaller blocks

```
DECLARE
    VA NUMBER(4): =10;
    VB NUMBER(4): =11;
BEGIN
    DECLARE
        VA NUMBER(4): =100;
        VB NUMBER(4): = 111;
    BEGIN
        DBMS_OUTPUT.PUT_LINE(VA);
        DBMS_OUTPUT.PUT_LINE(VB);
    END;
    DBMS_OUTPUT.PUT_LINE('.....');
    DBMS_OUTPUT.PUT_LINE(VA);
    DBMS_OUTPUT.PUT_LINE(VB);
END;
```

PL/SQL Basics

Advantages of PL/SQL

Structure of PL/SQL Block

Character Sets, Comments,
Special operators

Variables, Constant,
BOOLEAN Datatype,
%TYPE

User Message

Conditional Logic

CASE Statement

Looping Structure

Scope of variable in nested block

DEFINE Statement

DEFINE Statement

It is used to enter different sets of values by using & symbol on substitution variables

It is done by setting the DEFINE ON and OFF

At the entry of each value, execute:

SET DEFINE OFF;

Later execute:

SET DEFINE ON;

PL/SQL Basics

Advantages of PL/SQL

Structure of PL/SQL Block

Character Sets, Comments,
Special operators

Variables, Constant,
BOOLEAN Datatype,
%TYPE

User Message

Conditional Logic

CASE Statement

Looping Structure

Scope of variable in
nested block

DEFINE Statement