Hexaware TECHNOLOGIES
YOUR SUCCESS IS OUR FOCUS

www.hexaware.com

PL/SQL

Date : 01/12/2014

© Hexaware Technologies. All rights reserved.

Session Objective

Need for PL/SQL

Types of PL/SQL blocks

Variables and initialization

Various data types

Identify the benefits of using the %TYPE attribute

Cursors

Use simple loops and cursor FOR loops to fetch data

© Hexaware Technologies. All rights reserved. | www.hexaware.com

What is PL/SQL

- PL/SQL:
  - Stands for Procedural Language extension to SQL
  - Is Oracle Corporation's standard data access language for relational databases
  - Seamlessly integrates procedural constructs with SQL

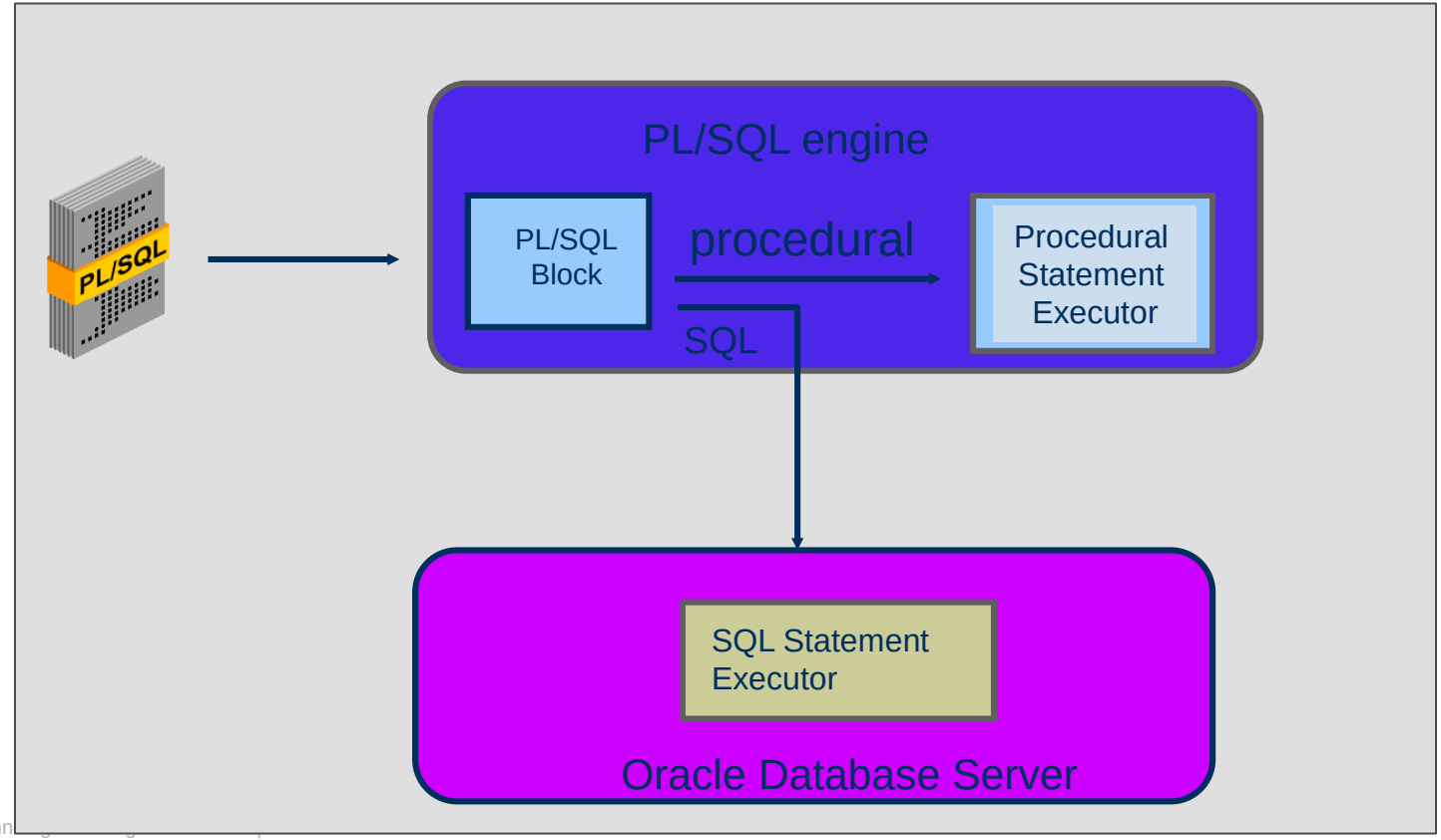
© Hexaware Technologies. All rights reserved. | www.hexaware.com

About PL/SQl

- PL/SQL:
  - Provides a block structure for executable units of code. Maintenance of code is made easier with such a well-defined structure.
  - Provides procedural constructs such as:
    - Variables, constants, and types
    - Control structures such as conditional statements and loops
    - Reusable program units that are written once and executed many times

© Hexaware Technologies. All rights reserved. | www.hexaware.com

PL/SQL Environment



© Hexaware Techn

Pl/SQL Block Structure

- `DECLARE` (optional)
  - Variables, cursors, user-defined exceptions
- `BEGIN` (mandatory)
  - SQL statements
  - PL/SQL statements
- `EXCEPTION` (optional)
  - Actions to perform when errors occur
- `END`; (mandatory)

© Hexaware Technologies. All rights reserved. | www.hexaware.com

Block Types

Block Types

Anonymous

```
[DECLARE]


BEGIN
  --statements

[EXCEPTION]

END;
```

Procedure

```
PROCEDURE name
IS

BEGIN
  --statements

[EXCEPTION]

END;
```

Function

```
FUNCTION name
RETURN datatype
IS
BEGIN
  --statements
  RETURN value;
[EXCEPTION]

END;
```

© Hexaware Technologies. All rights reserved. | www.hexaware.com

Identifiers

- Identifiers are used for:
  - Naming a variable
  - Providing conventions for variable names
    - Must start with a letter
    - Can include letters or numbers
    - Can include special characters (such as dollar sign, underscore, and pound sign)
    - Must limit the length to 30 characters
    - Must not be reserved words

© Hexaware Technologies. All rights reserved. | www.hexaware.com

Declaring and Initializing PL/SQL Variables

```
DECLARE
  emp_hiredate      DATE;
  emp_deptno     NUMBER(2) NOT NULL := 10;
  location          VARCHAR2(13) := 'Atlanta';
  c_comm            CONSTANT NUMBER := 1400;
```

© Hexaware Technologies. All rights reserved. | www.hexaware.com

Declaring and Initializing PL/SQL Variables

```
SET SERVEROUTPUT ON
DECLARE
  Myname VARCHAR2(20);
BEGIN
  DBMS_OUTPUT.PUT_LINE('My name is: '||Myname);
  Myname := 'John';
  DBMS_OUTPUT.PUT_LINE('My name is: '||Myname);
END;
/
```

```
SET SERVEROUTPUT ON
DECLARE
  Myname VARCHAR2(20):= 'John';
BEGIN
  Myname := 'Steven';
  DBMS_OUTPUT.PUT_LINE('My name is: '||Myname);
END;
/
```

© Hexaware Technologies. All rights reserved. | www.hexaware.com

Types of Variables

- PL/SQL variables:
  - Scalar
  - Composite
  - Reference
  - Large object (LOB)

- Non-PL/SQL variables: Bind variables

© Hexaware Technologies. All rights reserved. | www.hexaware.com

Base Scalar Data Types

- CHAR [(maximum_length)]
- VARCHAR2 (maximum_length)
- LONG
- LONG RAW
- NUMBER [(precision, scale)]
- BINARY_INTEGER
- PLS_INTEGER
- BOOLEAN
- BINARY_FLOAT
- BINARY_DOUBLE

© Hexaware Technologies. All rights reserved. | www.hexaware.com

Data Type	Description
CHAR [(*maximum_length*)]	Base type for fixed-length character data up to 32,767 bytes. If you do not specify a *maximum_length*, the default length is set to 1.
VARCHAR2 (*maximum_length*)	Base type for variable-length character data up to 32,767 bytes. There is no default size for VARCHAR2 variables and constants.
LONG	Base type for variable-length character data up to 32,760 bytes. Use the LONG data type to store variable-length character strings. You can insert any LONG value into a LONG database column because the maximum width of a LONG column is 2**31 bytes. However, you cannot retrieve a value longer than 32760 bytes from a LONG column into a LONG variable.
LONG RAW	Base type for binary data and byte strings up to 32,760 bytes. LONG RAW data is not interpreted by PL/SQL.
NUMBER [(*precision, scale*)]	Number having precision *p* and scale *s*. The precision *p* can range from 1 to 38. The scale *s* can range from -84 to 127.

© Hexaware Technologies. All rights reserved. | www.hexaware.com

PL/SQL Data types Contd

Data Type	Description
BINARY_INTEGER	Base type for integers between -2,147,483,647 and 2,147,483,647.
PLS_INTEGER	Base type for signed integers between -2,147,483,647 and 2,147,483,647. PLS_INTEGER values require less storage and are faster than NUMBER and BINARY_INTEGER values.
BOOLEAN	Base type that stores one of three possible values used for logical calculations: TRUE, FALSE, or NULL.

© Hexaware Technologies. All rights reserved. | www.hexaware.com

%TYPE Attribute

- The %TYPE attribute
  - Is used to declare a variable according to:
    - A database column definition
    - Another declared variable
  - Is prefixed with:
    - The database table and column
    - The name of the declared variable

© Hexaware Technologies. All rights reserved. | www.hexaware.com

Bind Variables and Host Variables

- Bind variables are:
  - Created in the environment
  - Also called *host* variables
  - Created with the `VARIABLE` keyword
  - Used in SQL statements and PL/SQL blocks
  - Accessed even after the PL/SQL block is executed
  - Referenced with a preceding colon

© Hexaware Technologies. All rights reserved. | www.hexaware.com

Substitution Variables

Are used to get user input at run time
Are referenced within a PL/SQL block with a preceding ampersand
Are used to avoid hard-coding values that can be obtained at run time

```
VARIABLE emp_salary NUMBER
SET AUTOPRINT ON
DECLARE
  empno NUMBER(6):=&empno;
BEGIN
  SELECT salary  INTO :emp_salary
  FROM  employees WHERE employee_id = empno;
END;
/
```

© Hexaware Technologies. All rights reserved. | www.hexaware.com

Composite Data Types

TRUE	23-DEC-98	EMPLOYEE	

PL/SQL table structure

1	SMITH
2	JONES
3	NANCY
4	TIM

VARCHAR2

PLS_INTEGER

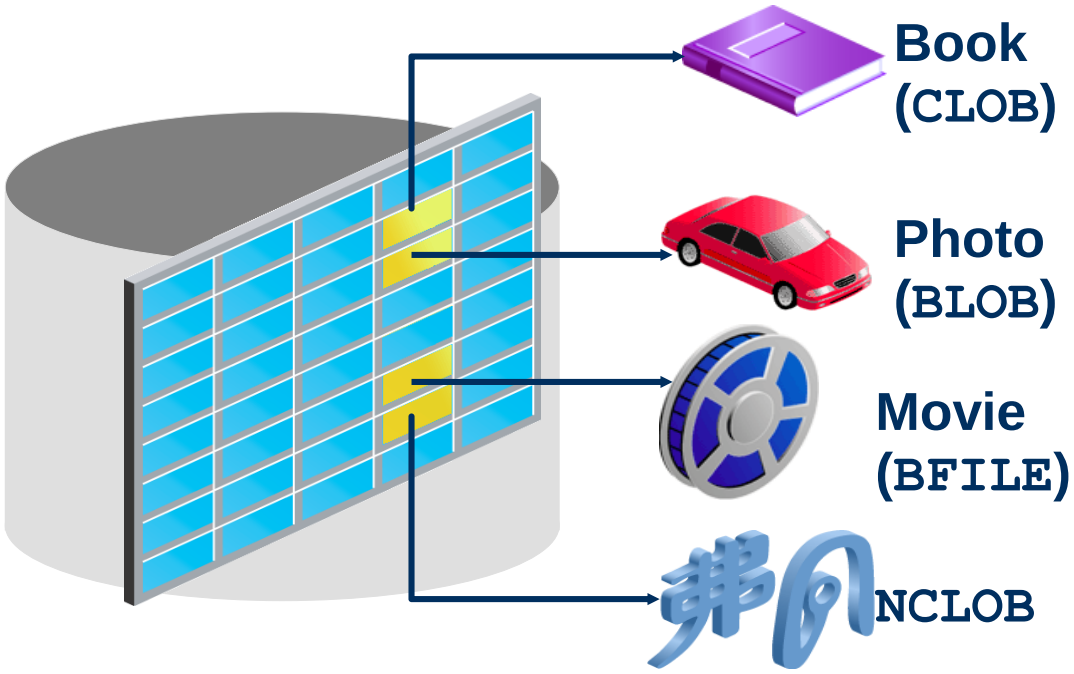
1	5000
2	2345
3	12
4	3456

NUMBER

PLS_INTEGER

© Hexaware Technologies. All rights reserved. | www.hexaware.com

LOB Data Type



© Hexaware Technologies. All rights reserved. | www.hexaware.com

Composite Data Types

Can hold multiple values (unlike scalar types)

Are of two types:

- PL/SQL records
- PL/SQL collections
  - `INDEX BY` tables or associative arrays
  - Nested table
  - `VARRAY`

© Hexaware Technologies. All rights reserved. | www.hexaware.com

PL/SQL Records

- Must contain one or more components (called *fields*) of any scalar, `RECORD`, or `INDEX BY` table data type
- Are similar to structures in most 3GL languages (including C and C++)
- Are user defined and can be a subset of a row in a table
- Treat a collection of fields as a logical unit
- Are convenient for fetching a row of data from a table for processing

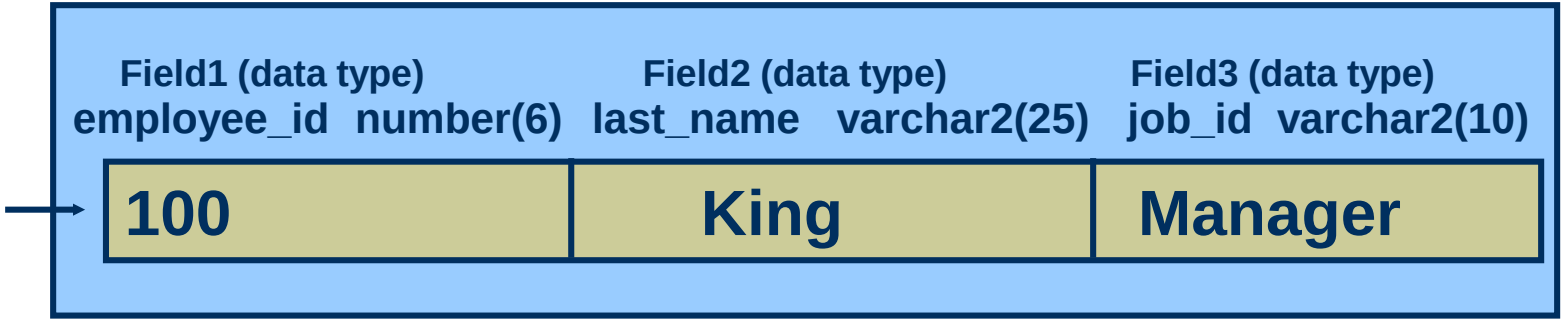
© Hexaware Technologies. All rights reserved. | www.hexaware.com

Creating a PL/SQL Record

- DECLARE
- TYPE emp_record_type IS RECORD(employee_id NUMBER(6) NOT NULL:= 100,
- last_name employees.last_name%TYPE,job_id employees.job_id%TYPE); emp_record emp_record_type;

© Hexaware Technologies. All rights reserved. | www.hexaware.com

PL/SQL Record Structure



© Hexaware Technologies. All rights reserved. | www.hexaware.com

%ROWTYPE Attribute

- Declare a variable according to a collection of columns in a database table or view.
- Prefix `%ROWTYPE` with the database table or view.
- Fields in the record take their names and data types from the columns of the table or view.

© Hexaware Technologies. All rights reserved. | www.hexaware.com

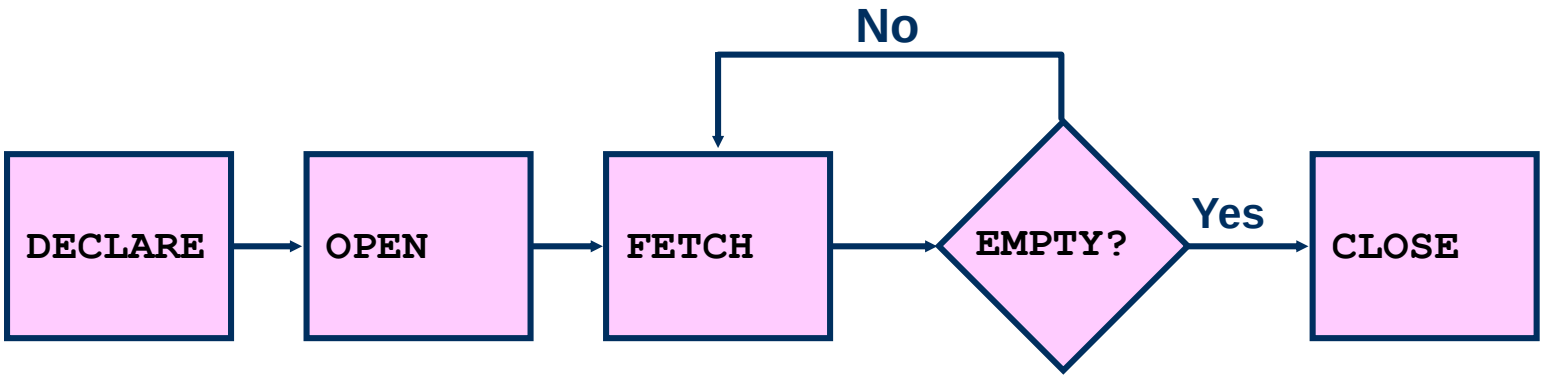
Cursors

•Every SQL statement executed by the Oracle server has an associated individual cursor:

- Implicit cursors: Declared and managed by PL/SQL for all DML and PL/SQL SELECT statements
- Explicit cursors: Declared and managed by the programmer

© Hexaware Technologies. All rights reserved. | www.hexaware.com

Controlling Explicit Cursors

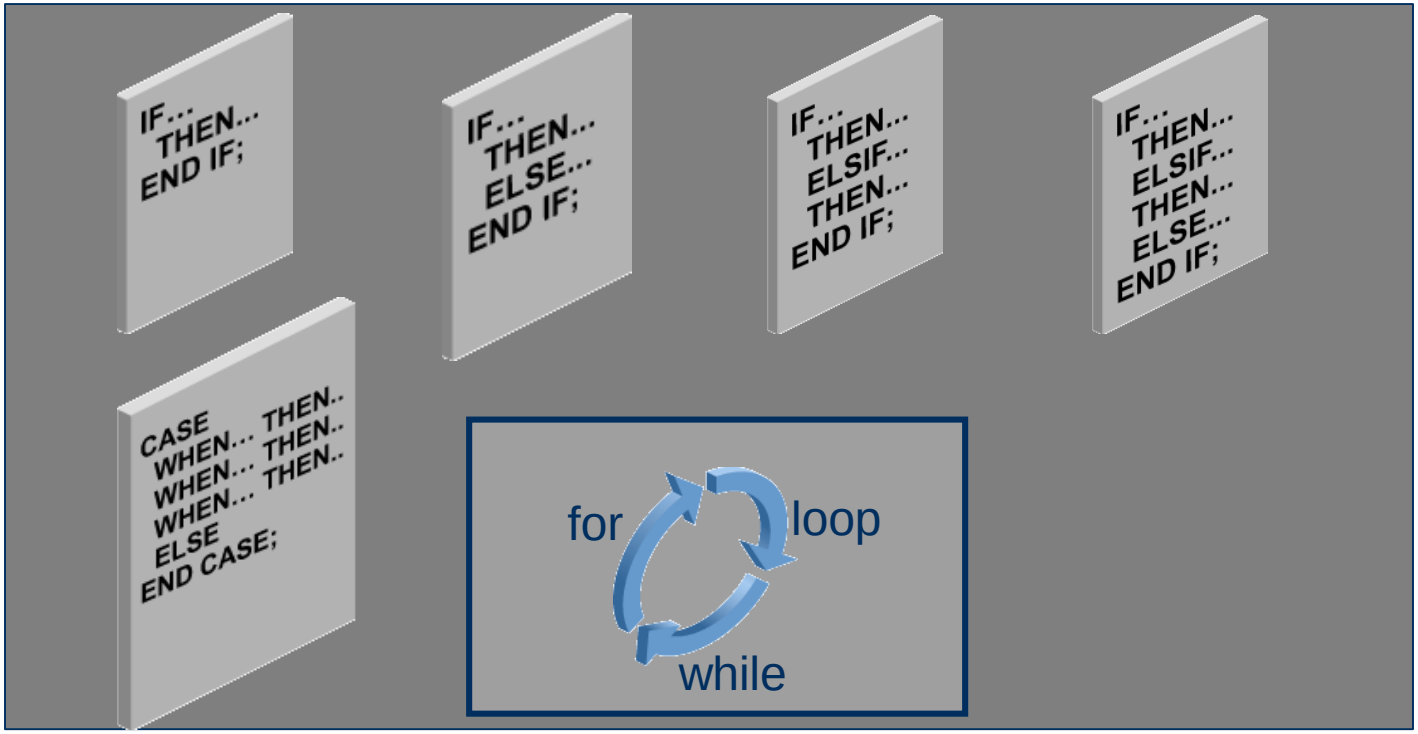


- Create a named SQL area.
- Identify the active set.
- Load the current row into variables.
- Test for existing rows.
- Return to `FETCH` if rows are found.
- Release the active set.

© Hexaware Technologies. All rights reserved. | www.hexaware.com

© Hexaware Technologies. All rights reserved. | www.hexaware.com

Controlling Flow of Execution



© Hexaware Technologies. All rights reserved. | www.hexaware.com

Iterative Control: `LOOP` Statements

- Loops repeat a statement or sequence of statements multiple times.
- There are three loop types:
  - Basic loop
  - `FOR` loop
  - `WHILE` loop

© Hexaware Technologies. All rights reserved. | www.hexaware.com

```
DECLARE
myage number:=31;
BEGIN
IF myage < 11
THEN
DBMS_OUTPUT.PUT_LINE(' I am a child ');
ELSE
DBMS_OUTPUT.PUT_LINE(' I am not a child ');
END IF;
END;
/
```

```
DECLARE
   grade CHAR(1) := UPPER('&grade');
   appraisal VARCHAR2(20);
BEGIN
   appraisal :=
    CASE grade
      WHEN 'A' THEN 'Excellent'
      WHEN 'B' THEN 'Very Good'
      WHEN 'C' THEN 'Good'
      ELSE 'No such grade'
      END;
DBMS_OUTPUT.PUT_LINE ('Grade: '|| grade || '
                      Appraisal ' || appraisal);
END;
/
```

```
declare
  n_num number := 1;
begin
   loop
   dbms_output.put(n_num||', ');
    n_num := n_num + 1;
    exit when n_num > 5;
 end loop;
  dbms_output.put_line('Final: '||n_num);
end;
/
```

```
DECLARE
salary number;
BEGIN
dbms_output.put_line('Enter the emp no');

SELECT sal into salary from emp where empno=&empno ;
WHILE salary<= 4000
LOOP
  salary := salary * 31;
dbms_output.put_line(salary);
END LOOP;
end;
/
```

© Hexaware Technologies. All rights reserved. | www.hexaware.com

```
DECLARE
  step  PLS_INTEGER := 2;
BEGIN
  FOR i IN 1..3 LOOP
    DBMS_OUTPUT.PUT_LINE (i*step);
  END LOOP;
END;
/
```

© Hexaware Technologies. All rights reserved. | www.hexaware.com

Thank you

© 2014 Hexaware Technologies Limited. All rights reserved. For internal circulation only. Neither this publication nor any part of it may be reproduced, stored in a retrieval system or transmitted in any form or in any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of Hexaware Technologies Limited. Published by Corporate Marketing & Communications