DATABASE PROGRAMMING WITH PL SQL 1 / 2

ORACLE ACADEMY

Contenido

Section 1 – Fundamentals

1-1 Introduction to PL/SQL

Procedural Constructs	Programming language features such as reusable/callable program units, modular blocks, cursors, constants, variables, assignment statements, conditional control statements, and loops.
PL/SQL	Oracle Corporations standard procedural language for relational databases which allows basic program logic and control flow to be combined with SQL statements.

MySQL	ORACLE

set @prom = (select avg(salary) from employees);
select avg(salary) into @prom from employees;
select @prom from dual;
	define prom = (select avg(salary) from employees);
	select &prom promedio from dual;

set @stmt = 'SELECT employee_id, salary FROM employees'; execute IMMEDIATE @stmt;	DEFINE colname=salary ; SELECT employee_id, &colname FROM employees;

Procedural Language extension to SQL.

```
-- anonymous procedures
set serveroutput on
declare
  prom number;
begin
  select avg(salary) into prom from employees;
  dbms_output.put_line('promedio: ' || prom);
end;
/
```

The basic unit in a PL/SQL program is a block.

All PL/SQL programs consist of blocks as modules.

```
set serveroutput on;
declare
  numero number := 5;
  cadena varchar2(100);
begin
  for i in 1..10 loop
    cadena := numero ||
       ' x ' || i || ' = ' || (numero*i);
    dbms_output.put_line(cadena);
  end loop;
end;
```

```
CREATE OR REPLACE PROCEDURE tabla(numero NUMBER)
IS
cadena VARCHAR2(100);
BEGIN
    FOR i IN reverse 1..10 LOOP
      cadena := numero ||
         ' x ' || i || ' = ' || (numero*i);
      dbms_output.put_line(cadena);
    END LOOP;
END;
/
```

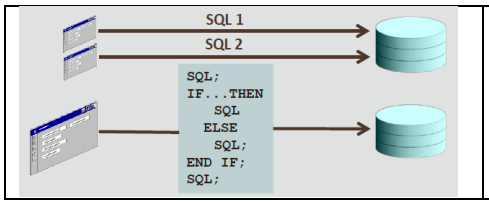
```
DECLARE
CURSOR cursor_employees IS SELECT * FROM employees;
BEGIN
    FOR c_emp in cursor_employees LOOP
        IF c_emp.job_id= 'SA_REP' AND c_emp.hire_date<='05-Feb-2005' THEN
            UPDATE employees SET job_id= 'SR_SA_REP'
            WHERE employee_id= c_emp.employee_id;
        ELSIF c_emp.job_id= 'MK_REP' AND c_emp.hire_date<= '05-Feb-2005' THEN
            UPDATE employees SET job_id= 'SR_MK_REP'
            WHERE employee_id= c_emp.employee_id;
        ELSIF c_emp.job_id = 'ST_CLERK' AND c_emp.hire_date<='05-Feb-2005' THEN
            UPDATE employees SET job_id= 'SR_ST_CLRK'
            WHERE employee_id= c_emp.employee_id;
        END IF;
    END LOOP;
END;
```

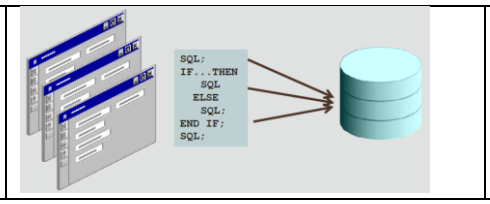
1-2 Benefits of PL/SQL

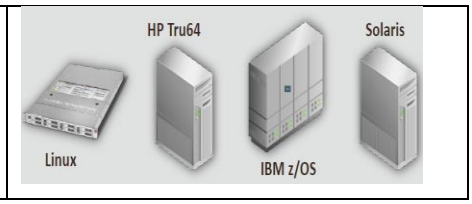
Portability	The ability for PL/SQL programs to run anywhere an Oracle server runs.
Blocks	The basic unit of PL/SQL programs - also known as modules.
Exceptions	An error that occurs in the database or in a user's program during runtime.

You can nest blocks inside other blocks to build powerful programs.
PL/SQL is integrated in Oracle tools, such as Oracle Forms Developer, Oracle Report Builder, and Application Express.
You can write portable program packages and create libraries that can be reused in different environments.
Exception Handling:

- If no data is found then…
- If too many rows are found then…
- If an invalid number is calculated then…







PL/SQL in Oracle Products

Oracle Product	PL/SQL
ORACLE DATABASE 11g	You can write PL/SQL code to manage application data or to manage the Oracle database itself. For example, you can write code for updating data (DML), creating data (DDL), generating reports, managing security, and so on.
ORACLE APPLICATION SERVER 10g	Using the Web Application Toolkit, you can create database-centric web applications written entirely or partially in PL/SQL.
ORACLE DEVELOPER SUITE 10g	Using Forms Builder and Reports Developer, Oracle's client-side developer tools, you can build database-centric web applications and reports that include PL/SQL.
ORACLE Application Express	Using a Web browser you can develop web applications that include PL/SQL.

1-3 Creating PL/SQL Blocks

Anonymous PL/SQL block	Unnamed blocks of code not stored in the database and do not exist after they are executed
Function	A program that computes and returns a single value
Subprograms	Named PL/SQL blocks that are stored in the database and can be declared as procedures or functions
Compiler	Software that checks and translates programs written in high-level programming languages into binary code to execute
Procedure	A program that performs an action, but does not have to return a value

Application Express	Browser-based, database-driven, application development environment.
SQL Workshop	A component of Application Express.
Application Builder	A component of Application Express.
SQL Developer	An IDE for database development and management.
JDeveloper	An IDE for Java-based development.
NetBeans	An IDE for Java, HTML5, PHP, and C++.

Anonymous Blocks	Procedure: Performs an action	Function: Computes and returns a value

[DECLARE]
BEGIN
  -- statements
[EXCEPTION]
END;
	PROCEDURE name
	IS
	--variable declarations
	BEGIN
	  --statements
	[EXCEPTION]
	END;
		FUNCTION name
		RETURN datatype
		--variable declaration(s)
		IS
		BEGIN
		  --statements
		  RETURN value;
		[EXCEPTION]
		END;

Section	**Description**	**Inclusion**
Declarative (DECLARE)	Contains declarations of all variables, constants, cursors, and user-defined exceptions that are referenced in the executable and exception sections.	Optional
Executable (BEGIN … END;)	Contains SQL statements to retrieve data from the database and PL/SQL statements to manipulate data in the block. Must contain at least one statement.	Mandatory
Exception (EXCEPTION)	Specifies the actions to perform when errors and abnormal conditions arise in the executable section.	Optional

EXCEPTIONS

```
set SERVEROUTPUT ON;
DECLARE
    v_first_name VARCHAR2(25);
    v_last_name  VARCHAR2(25);
BEGIN
    SELECT first_name, last_name
    INTO v_first_name, v_last_name
    FROM employees
    WHERE last_name= 'King';
    DBMS_OUTPUT.PUT_LINE ('The employee of the month is: '  ||
                          v_first_name|| ' ' || v_last_name|| '.');
EXCEPTION
    WHEN TOO_MANY_ROWS THEN
        DBMS_OUTPUT.PUT_LINE ('Your select statement retrieved multiple rows.
                              Consider using a cursor or changingthe search criteria.');
END;
/
```

```
CREATE OR REPLACE PROCEDURE print_date
IS
v_date VARCHAR2(30);
BEGIN
    SELECT TO_CHAR(SYSDATE,'Mon DD, YYYY') INTO v_date FROM DUAL;
    DBMS_OUTPUT.PUT_LINE(v_date);
END;
```

```
begin
    PRINT_DATE;
end;
/

call    print_date();
execute print_date();
```

```
CREATE OR REPLACE FUNCTION tomorrow(p_today IN DATE)
    RETURN DATE
IS
    v_tomorrow DATE;
BEGIN
    SELECT p_today+1 INTO v_tomorrow FROM DUAL;
    RETURN v_tomorrow;
END;
```

```
BEGIN
DBMS_OUTPUT.PUT_LINE(TOMORROW(SYSDATE));
END;
/

SELECT TOMORROW(SYSDATE) FROM DUAL;
```

```
CREATE OR REPLACE FUNCTION factorial(m number)
  RETURN number
IS
  r number default 1;
  n number := m;
BEGIN
   while (n > 0) loop
      r := r * n;
      n := n - 1;
   end loop;
   return r;
END;
/
```

```
select factorial(5) from dual;
```

Section 2 – Defining Variables and Datatypes

2-1 Using Variables in PL/SQL

Variables	Used for storage of data and manipulation of stored values.
Parameters	Values passed to a program by a user or by another program to customize the program.

***Identifier* [CONSTANT] *datatype* [NOT NULL] [:= *expr*| DEFAULT *expr*];**

```
set SERVEROUTPUT ON
DECLARE
/* Declaracion de
   Variables */
    v_counter INTEGER := 0;
    v_contador number(3) DEFAULT 0;
    v_name VARCHAR2(20) := 'John';
    v_date Date default SYSDATE;
    c_pi constant number(5,4) := 3.1416;
    v_activo BOOLEAN := True;
BEGIN
    v_counter:= v_counter + 1;
    -- SELECT SYSDATE INTO v_date FROM DUAL;
    DBMS_OUTPUT.PUT_LINE(v_counter || ' ' || v_name || ' ' || v_date);
END;
/
```

2-2 Recognizing PL/SQL Lexical Units

Literals	An explicit numeric, character string, date, or Boolean value that is not represented by an identifier. 'UPA' != 'Upa'
Delimiters	Symbols that have special meaning to an Oracle database.
Reserved words	Words that have special meaning to an Oracle database and cannot be used as identifiers.
Comments	Describe the purpose and use of each code segment and are ignored by PL/SQL.
Identifiers	A name, up to 30 characters in length, given to a PL/SQL object. **Not sensitive** May include $ (dollar sign), _ (underscore), or # (hashtag) . v**C**ounter$ = vcounter$
Lexical Units	Building blocks of any PL/SQL block and are sequences of characters including letters, digits, tabs, returns, and symbols.

Partial List of Reserved Words

ALL	CREATE	FROM	MODIFY	SELECT
ALTER	DATE	GROUP	NOT	SYNONYM
AND	DEFAULT	HAVING	NULL	SYSDATE
ANY	DELETE	IN	NUMBER	TABLE
AS	DESC	INDEX	OR	THEN
ASC	DISTINCT	INSERT	ORDER	UPDATE
BETWEEN	DROP	INTEGER	RENAME	VALUES
CHAR	ELSE	INTO	ROW	VARCHAR2
COLUMN	EXISTS	IS	ROWID	VIEW
COMMENT	FOR	LIKE	ROWNUM	WHERE

Delimiters

Symbol	Meaning	Symbol	Meaning
+	addition operator	<>	inequality operator
-	subtraction/negation operator	!=	inequality operator
*	multiplication operator	||	concatenation operator
/	division operator	--	single-line comment indicator
=	equality operator	/*	beginning comment delimiter
'	character string delimiter	*/	ending comment delimiter
;	statement terminator	**	exponent
		:=	assignment operator

2-3 Recognizing Data Types

Object	A schema object with a name, attributes, and methods.
Scalar	Hold a single value with no internal components.
Composite	Contain internal elements that are either scalar (record) or composite (record and table)
Reference	Hold values, called pointers, that point to a storage location.
LOB	Hold values, called locators, that specify the location of large objects (such as graphic images) that are stored out of line. (text, images, video, audio) up to 4GB
BFILE	Store large **binary files** outside of the database.
BLOB	Store large unstructured or structured **binary objects**.
CLOB	Store large blocks of **character** data in the database.
NCLOB	Store large blocks of single-byte or fixed width multi-byte NCHAR data in the database. National language character large object (NCLOB)

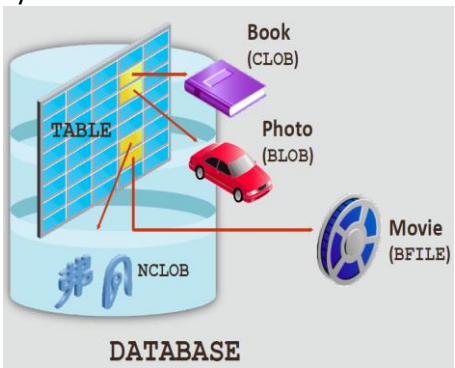
PL/SQL supports five categories of data type

Data Type	Description
Scalar	Holds a single value with no internal elements. Character (char [1..32767], varchar2, long(2GB) Number(number, pls_integer) Date(date, Timestamp) Boolean (True, False, Null)
Composite	Contains multiple internal elements that can be manipulated individually. Record(One row), Table, Varray RECORD v_emp_record employees%ROWTYPE; v_emp_record.first_name
Large Object (LOB)	Holds values called locators that specify the location of large objects (such as graphic images) that are stored out of line.
Reference	Holds values called pointers that point to a storage location.
Object	It is a schema object with a name, attributes, and methods. An object data type is similar to the class mechanism supported by C++ and Java.

LOB Data Type

CLOB, BLOB, and NCLOB data is stored in the database, either inside or outside of the row.

BFILE data is stored in operating system files outside the database.



2-4 Using Scalar Data Types

BOOLEAN	A datatype that stores one of the three possible values used for logical calculations: TRUE, FALSE, or NULL.
%TYPE	Attribute used to declare a variable according to another previously declared variable or database column. PL/SQL determines the data type and size of the variable.

***Identifier* table_name.column_name%TYPE;**
***Identifier identifier*%TYPE;**

```
set SERVEROUTPUT ON
DECLARE
    v_valid       BOOLEAN := True;
    v_id          employees.first_name%TYPE default 100;
    v_last_name   VARCHAR2(25);
    v_salary      employees.salary%TYPE;
    v_new_salary  v_salary%TYPE;
BEGIN
    select last_name, salary into v_last_name, v_salary
    from employees where employee_id = v_id;
    IF v_valid THEN
        DBMS_OUTPUT.PUT_LINE(v_Last_name || ' ' || (v_salary+1));
    ELSE
        DBMS_OUTPUT.PUT_LINE('Test is FALSE');
    END IF;
END;
/
```

2-5 Writing PL/SQL Executable Statements

Explicit conversion	Converts values from one data type to another by using built-in functions.
Implicit conversion	Converts data types dynamically if they are mixed in a statement.

Character Functions:			Number Functions:			Date Functions:	
ASCII	LENGTH	RPAD	ABS	EXP	ROUND	ADD_MONTHS	MONTHS_BETWEEN
CHR	LOWER	RTRIM	ACOS	LN	SIGN	CURRENT_DATE	ROUND
CONCAT	LPAD	SUBSTR	ASIN	LOG	SIN	CURRENT_TIMESTAMP	SYSDATE
INITCAP	LTRIM	TRIM	ATAN	MOD	TAN	LAST_DAY	TRUNC
INSTR	REPLACE	UPPER	COS	POWER	TRUNC		

Implicit Conversions (It's not recommended)	Explicit Conversions	
	TO_NUMBER()	ROWIDTONCHAR()
	TO_CHAR()	HEXTORAW()
	TO_CLOB()	RAWTOHEX()
	CHARTOROWID()	RAWTONHEX()
	ROWIDTOCHAR()	TO_DATE()

Operator	Operation
**	Exponentiation
+, -	Identity, negation
*, /	Multiplication, division
+, -, ||	Addition, subtraction, concatenation
=, <, >, <=, >=, <>, !=, ~=, ^=, IS NULL, LIKE, BETWEEN, IN	Comparison
NOT	Logical negation
AND	Conjunction
OR	Inclusion

Statements can continue over several lines:
v_quote := 'The only thing that we can know is that we know
nothing and that is the highest flight of human reason.';
 	
Numbers can be simple values or scientific notation: v_salary number := 2E4;
v_good_sal := v_sal BETWEEN 5000 AND 15000;
	DECLARE
	x VARCHAR2(20);
	BEGIN
	x := '123' + '456' ;
	DBMS_OUTPUT.PUT_LINE(x);
	END;

2-6 Nested Blocks and Variable Scope

2-7 Good Programming Practices

Section 3 – Using SQL in PL/SQL