

Database Programming with PL/SQL

Creating PL/SQL Blocks

ORACLE®

ACADEMY

Objectives

This lesson covers the following objectives:

- Describe the structure of a PL/SQL block
- Identify the different types of PL/SQL blocks
- Identify PL/SQL programming environments
- Create and execute an anonymous PL/SQL block
- Output messages in PL/SQL

Purpose

When you put something into a box you intuitively know that the box has consistent properties. It has sides, a bottom, and a top that can be opened and closed.

In PL/SQL, you put your programming instructions into block structures that also have consistent properties.

Purpose (cont.)

Here you will learn the structure of a PL/SQL block and create one kind of block: an anonymous block.

After learning about the different environments into which you can develop your PL/SQL programs, you will also begin coding PL/SQL in the Application Express development environment.

PL/SQL Block Structure

A PL/SQL block consists of three sections.

Section	Description
Declarative (optional)	The declarative section begins with the keyword <code>DECLARE</code> and ends when your executable section starts.
Executable (mandatory)	The executable section begins with the keyword <code>BEGIN</code> and ends with <code>END</code> . Observe that <code>END</code> is terminated with a semicolon. The executable section of a PL/SQL block can include any number of nested PL/SQL blocks.
Exception handling (optional)	The exception section is nested within the executable section. This section begins with the keyword <code>EXCEPTION</code> .



PL/SQL Block Structure Sections

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

The PL/SQL Compiler

Every program written in a high-level programming language (C, Java, PL/SQL and so on) must be checked and translated into binary code (ones and zeros) before it can execute. The software that does this checking and translation is called a compiler.



The PL/SQL Compiler (cont.)

The PL/SQL compiler executes automatically when needed. It checks not only that every word is spelled correctly, but also that any referenced database objects (such as tables) exist, and that the user has the necessary privileges to access them.

Anonymous Blocks

Characteristics of anonymous blocks:

- Unnamed block
- Not stored in the database
- Declared inline at the point in an application where it is executed
- Compiled each time the application is executed
- Passed to the PL/SQL engine for execution at run time
- Cannot be invoked or called because it does not have a name and does not exist after it is executed

Anonymous Blocks (cont.)

```
[DECLARE]

BEGIN
    --statements

[EXCEPTION]

END;
```

Examples of Anonymous Blocks

No declaration or exception sections, execution only

```
BEGIN
  DBMS_OUTPUT.PUT_LINE('PL/SQL is easy!');
END;
```

Declaration and execution sections, but no exception section

```
DECLARE
  v_date DATE := SYSDATE;
BEGIN
  DBMS_OUTPUT.PUT_LINE(v_date);
END;
```

Examples of Anonymous Blocks (cont.)

Declaration and exception sections

```
DECLARE
  v_country_name VARCHAR2(40);
  v_region_id     NUMBER;
BEGIN
  SELECT country_name, region_id
    INTO v_country_name, v_region_id
    FROM countries WHERE country_id='CA';
  DBMS_OUTPUT.PUT_LINE ('The country name is: '
                        ||v_country_name||' and is located in '
                        ||v_region_id||'.') ;
EXCEPTION
  WHEN TOO_MANY_ROWS THEN
    DBMS_OUTPUT.PUT_LINE ('Your select statement retrieved
                          multiple rows. Consider using a cursor.');
```

END;

Subprograms

Subprograms:

- Are named PL/SQL blocks
- Are named PL/SQL blocks
- Are stored in the database
- Can be invoked whenever you want depending on your application

```
PROCEDURE name
IS
    --variable declaration(s)
BEGIN
    --statements

[EXCEPTION]

END;
```

```
FUNCTION name
RETURN datatype
    --variable declaration(s)
IS
BEGIN
    --statements
    RETURN value;

[EXCEPTION]

END;
```

Subprograms (cont.)

Subprograms:

- Can be declared as procedures or as functions
 - Procedure: Performs an action
 - Function: Computes and returns a value

Examples of Subprograms

Procedure to print the current date



```
CREATE 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;
```

Function to return the number of characters in a string

```
CREATE FUNCTION num_characters (p_string IN VARCHAR2)
  RETURN INTEGER IS
  v_num_characters INTEGER;
BEGIN
  SELECT LENGTH(p_string) INTO v_num_characters
    FROM DUAL;
  RETURN v_num_characters;
END;
```

Program Constructs

The following table outlines a variety of different PL/SQL program constructs that use the basic PL/SQL block. The constructs are available based on the environment in which they are executed.

	
Tools constructs	Database server constructs
Anonymous blocks	Stored procedures or functions
Application procedures or functions	Stored packages
Application packages	Database triggers
Application triggers	Object types
Object types	

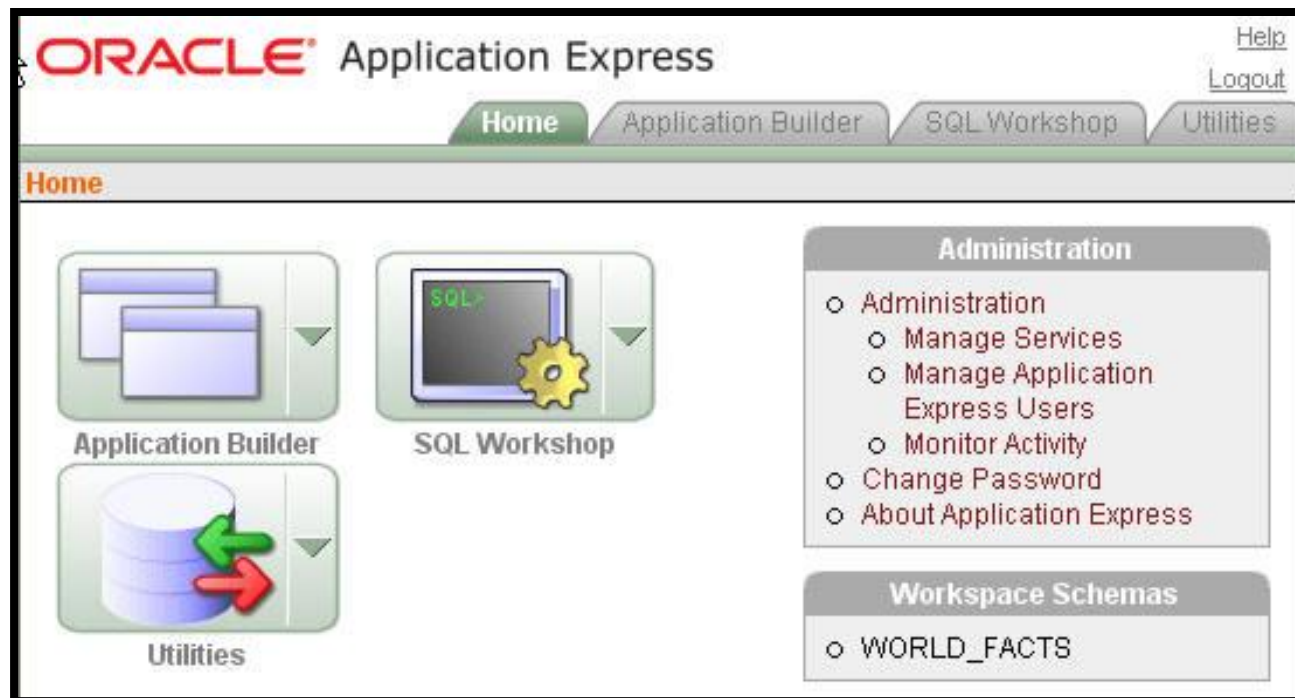
PL/SQL Programming Environments

There are many tools that provide an environment for developing PL/SQL. Oracle provides several tools you can use. Some of the Oracle development tools are:

ORACLE [®]	SQL*Workshop	A component of Application Express.
	SQL*Plus	A command-line application.
	SQL Developer	A Graphical User Interface (GUI) integrated development environments (IDE).
	JDeveloper	A Windows-based application.
	Application Express	A web-browser application.

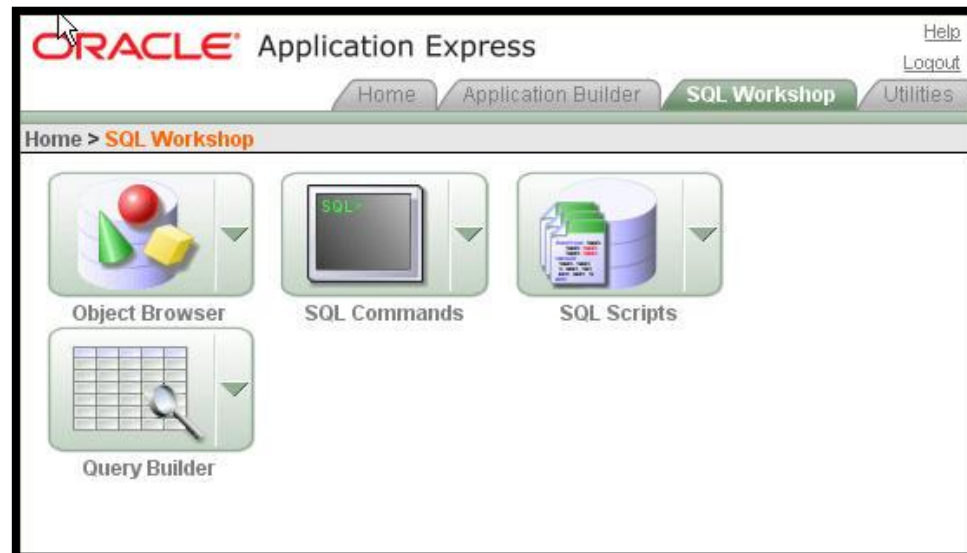
Oracle Application Express

Oracle Application Express is a browser-based web application environment that offers a SQL Workshop component.



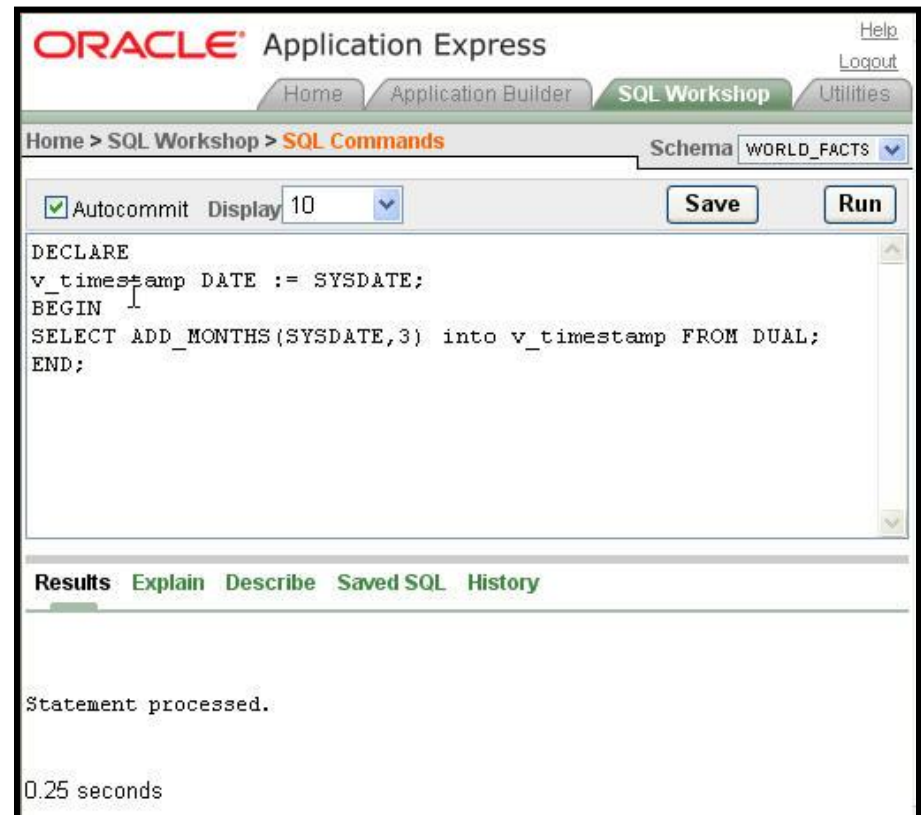
Developing with SQL Workshop

When you log in to Oracle Application Express and choose SQL Workshop, you can choose to use the SQL Commands option to use the SQL command-line editor, or you can choose the SQL Scripts option to work within the Script Editor.



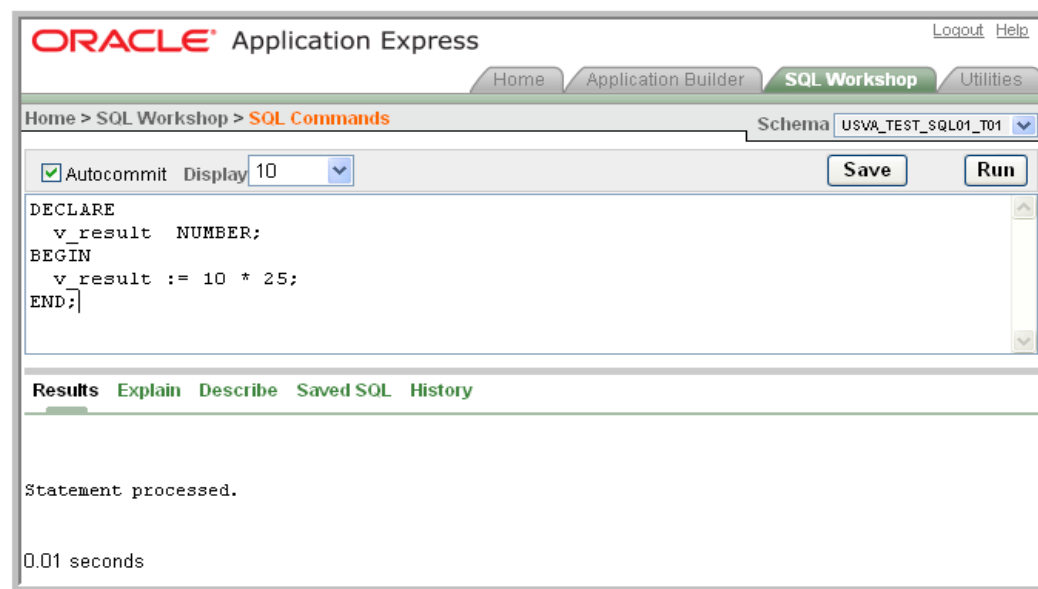
SQL Commands

You can use SQL Commands to enter and run a single SQL statement or a single PL/SQL block. A SQL script can contain one or more SQL statements and/or PL/SQL blocks. Use SQL Scripts to enter and run multi-statement scripts.



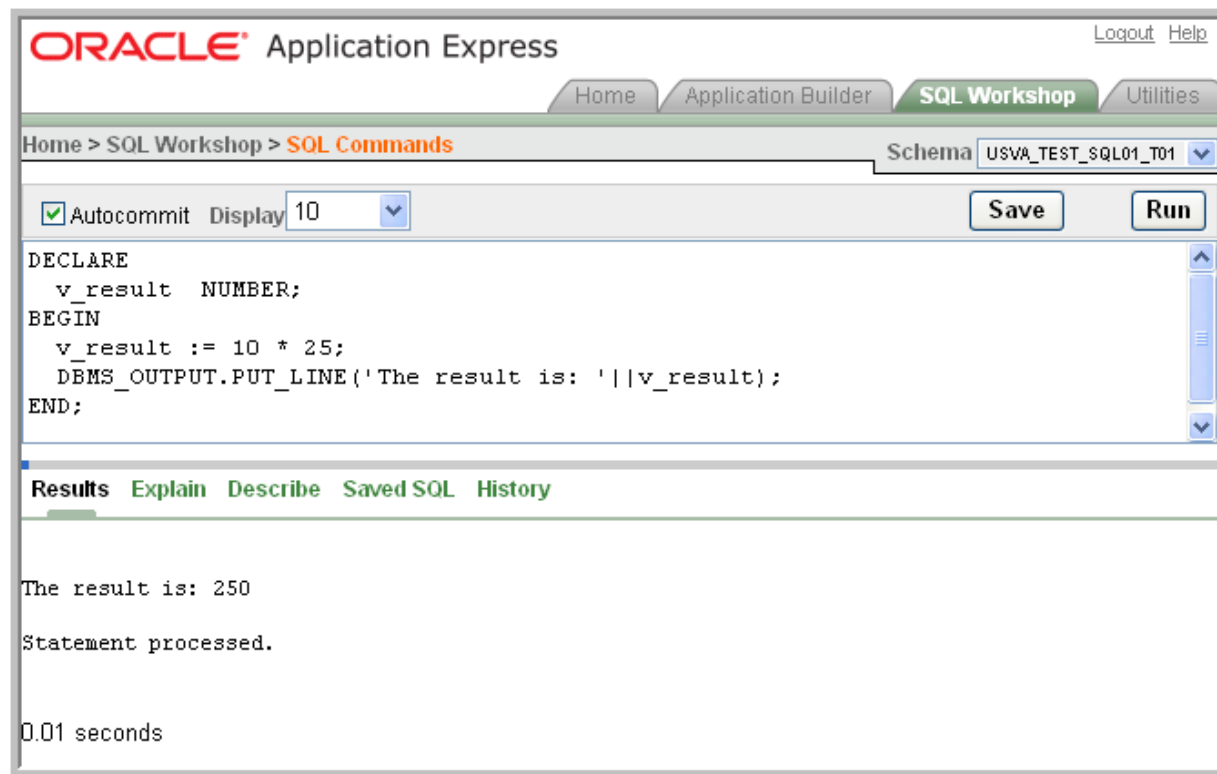
Using DBMS_OUTPUT.PUT_LINE Example

Look at this simple PL/SQL block and its output. How can you display the result?



Using DBMS_OUTPUT.PUT_LINE

Let's add a call to `DBMS_OUTPUT.PUT_LINE`. Now you can see the result!



Using DBMS_OUTPUT.PUT_LINE (cont.)

The `DBMS_OUTPUT.PUT_LINE` allows you to display results so that you can check that your block is working correctly. It allows you to display one character string at a time, although this can be concatenated.

```
DECLARE
    v_emp_count    NUMBER;
BEGIN
    DBMS_OUTPUT.PUT_LINE('PL/SQL is easy so far!');
    SELECT COUNT(*) INTO v_emp_count FROM employees;
    DBMS_OUTPUT.PUT_LINE('There are ' || v_emp_count || '
                          rows in the employees table');
END;
```

Terminology

Key terms used in this lesson included:

- Anonymous PL/SQL block
- Compiler
- Subprograms
- Procedures
- Functions

Summary

In this lesson, you should have learned how to:

- Describe the structure of a PL/SQL block
- Identify the different types of PL/SQL blocks
- Identify PL/SQL programming environments
- Create and execute an anonymous PL/SQL block
- Output messages in PL/SQL