

INTRODUCTION TO PL/SQL

OBJECTIVES

- After completing this lesson, you should be able to do the following:
 - Explain the need for PL/SQL
 - Explain the benefits of PL/SQL
 - Identify the different types of PL/SQL blocks
 - Use *iSQL*Plus* as a development environment for PL/SQL
 - Output messages in PL/SQL

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

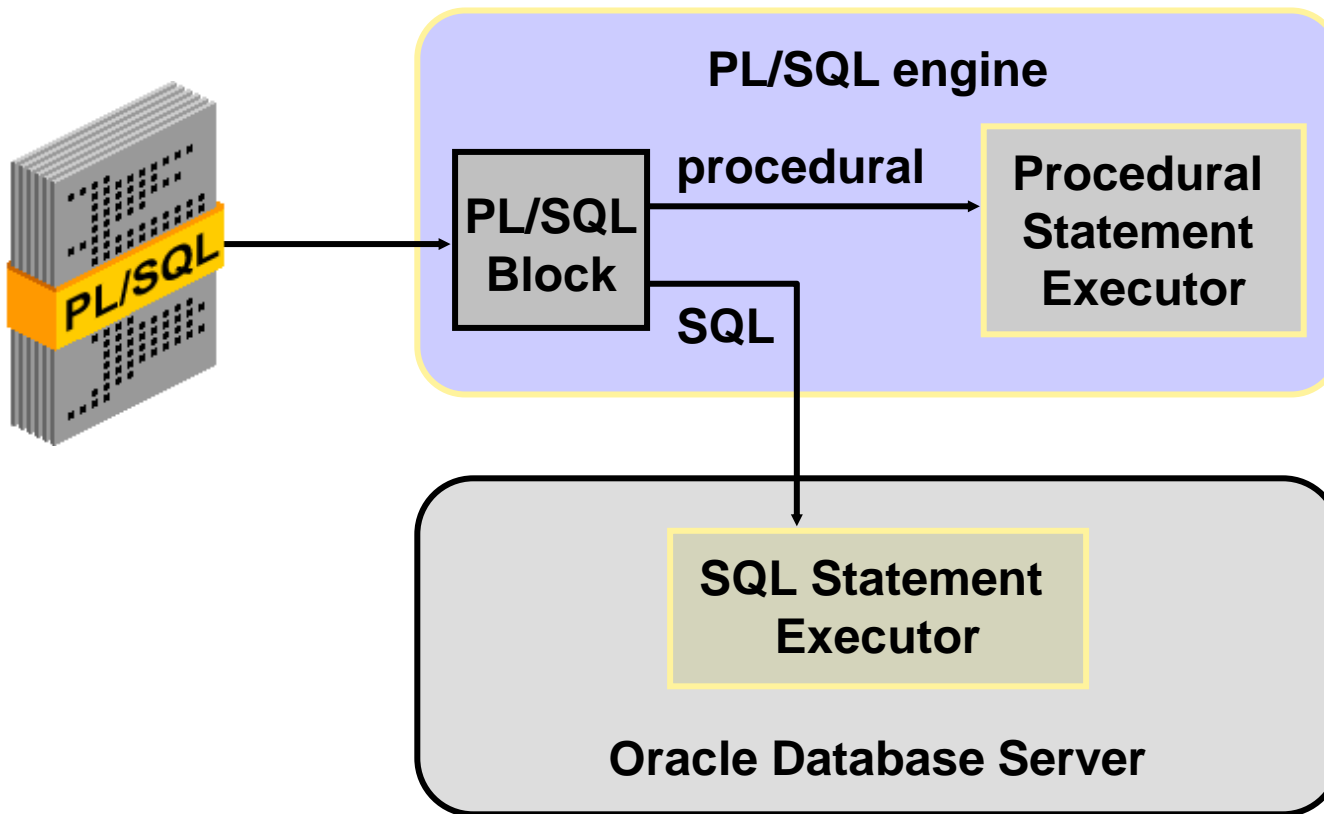


ABOUT PL/SQL

- PL/SQL:

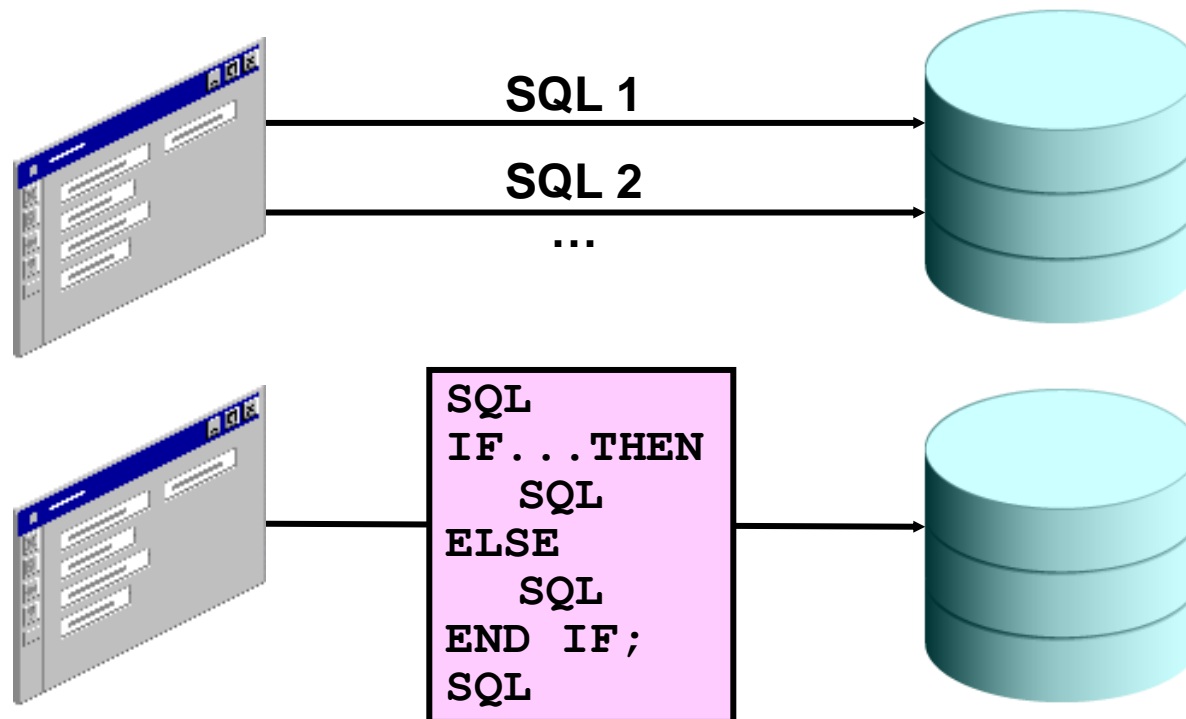
- Provides a block structure for executable units of code. Maintenance of code is made easier with such a well-defined structure.
- Provides procedural constructs such as:
 - Variables, constants, and types
 - Control structures such as conditional statements and loops
 - Reusable program units that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

PL/SQL ENVIRONMENT



BENEFITS OF PL/SQL

- Integration of procedural constructs with SQL
- Improved performance

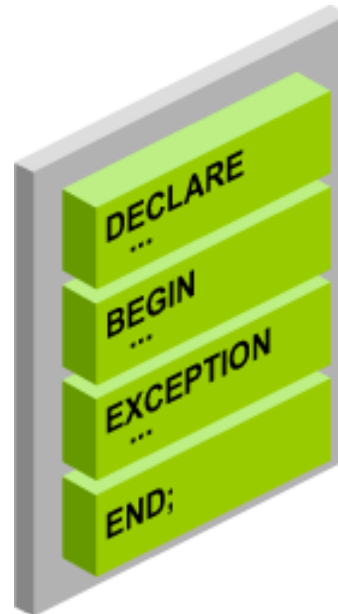


BENEFITS OF PL/SQL

- Modularized program development
- Integration with Oracle tools
- Portability
- Exception handling

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)



BLOCK TYPES



Anonymous

```
[DECLARE]

BEGIN
    --statements

[EXCEPTION]

END;
```

Function

```
PROCEDURE name
IS

BEGIN
    --statements

[EXCEPTION]

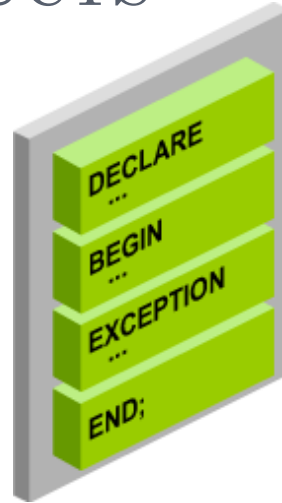
END;
```

Procedure

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

END;
```

PROGRAM CONSTRUCTS



Tools Constructs

Anonymous blocks

Application procedures
or functions

Application packages

Application triggers

Object types

Database Server Constructs

Anonymous blocks

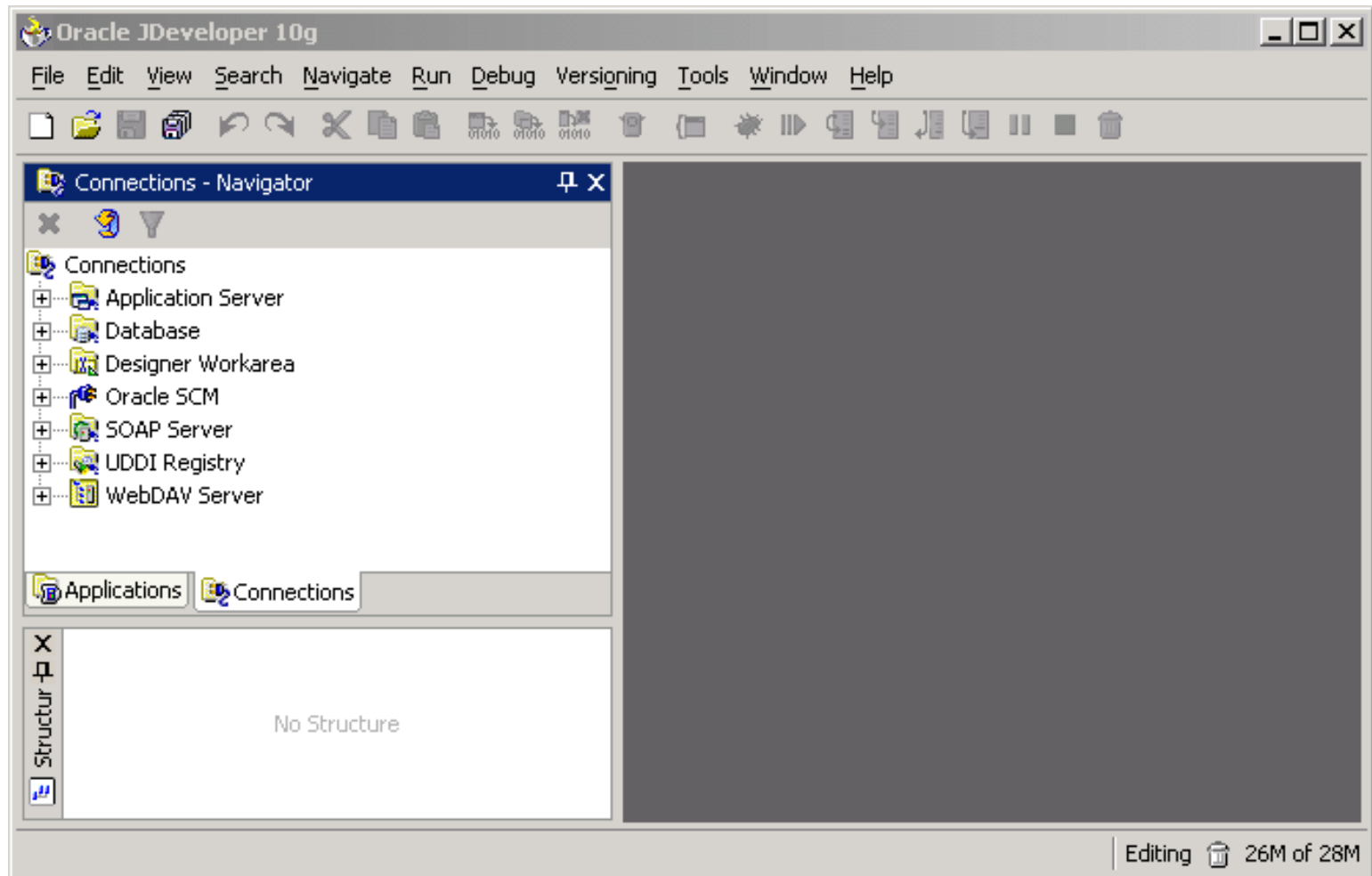
Stored procedures or
functions

Stored packages

Database triggers

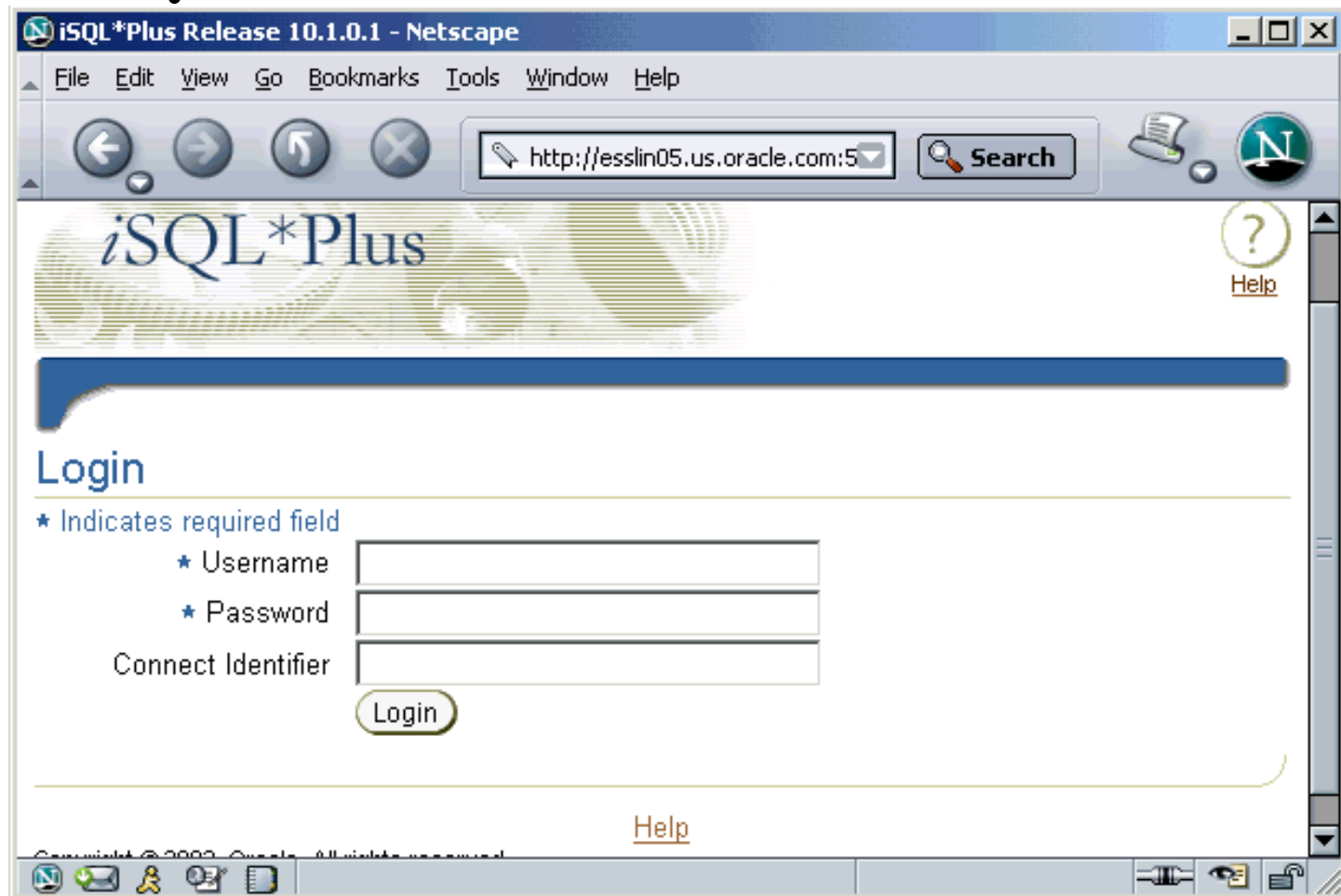
Object types

PL/SQL PROGRAMMING ENVIRONMENTS

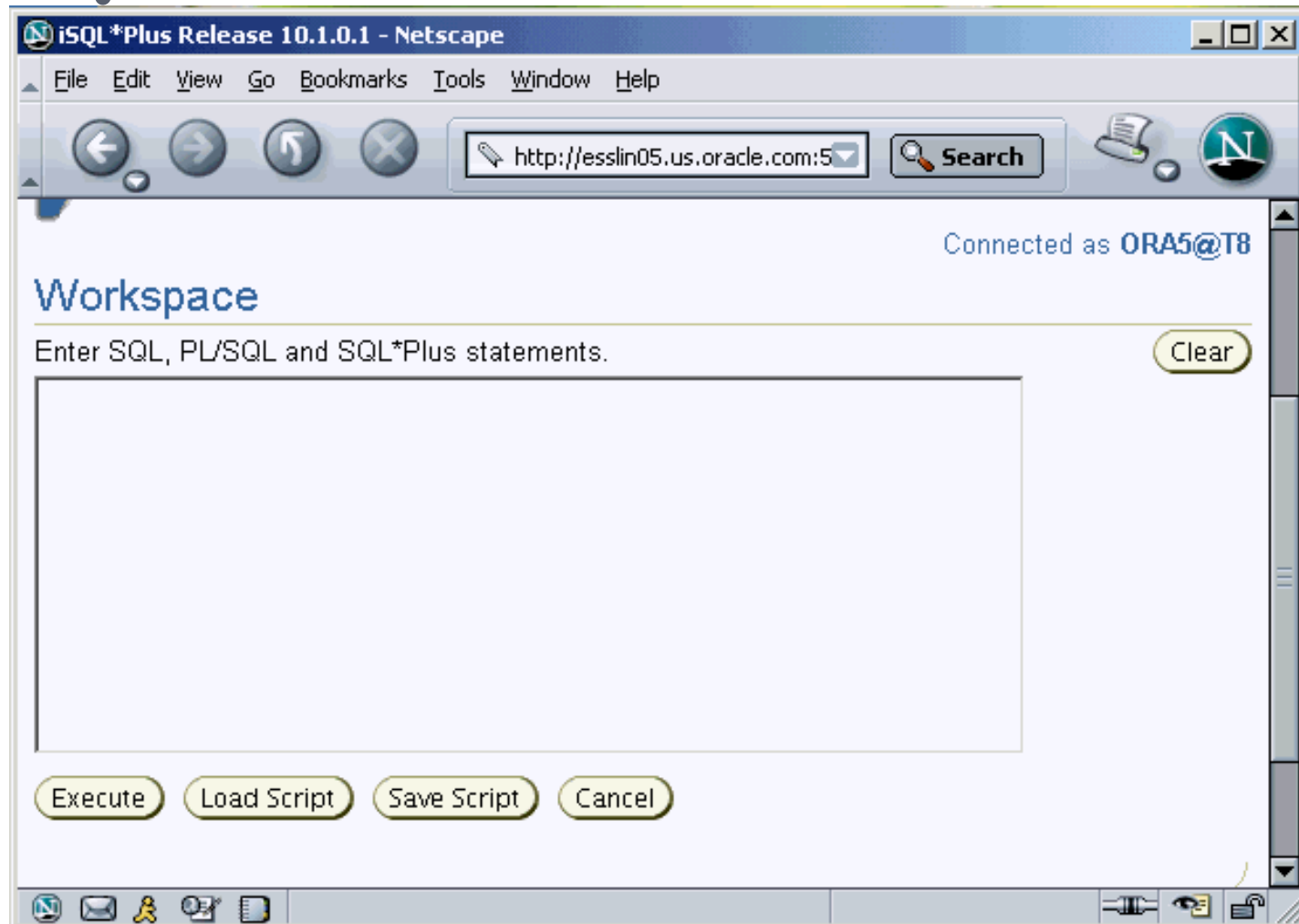


PL/SQL PROGRAMMING ENVIRONMENTS

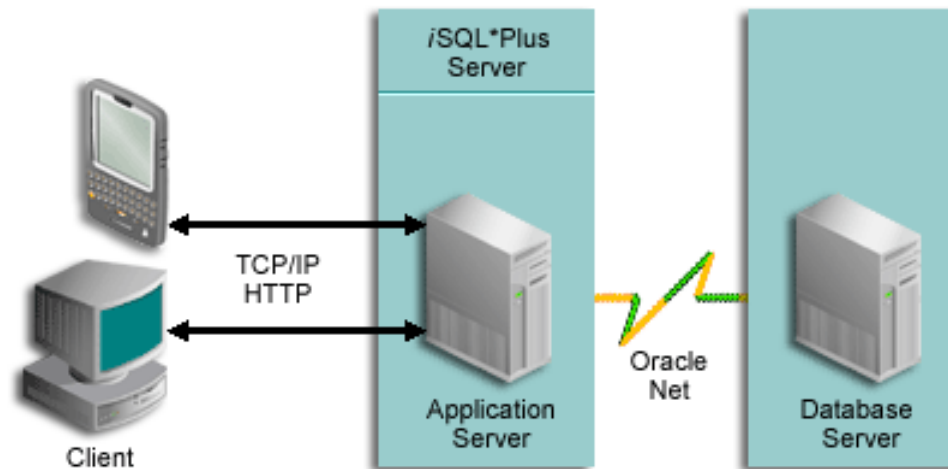
○ iSQL*Plus



PL/SQL PROGRAMMING ENVIRONMENTS

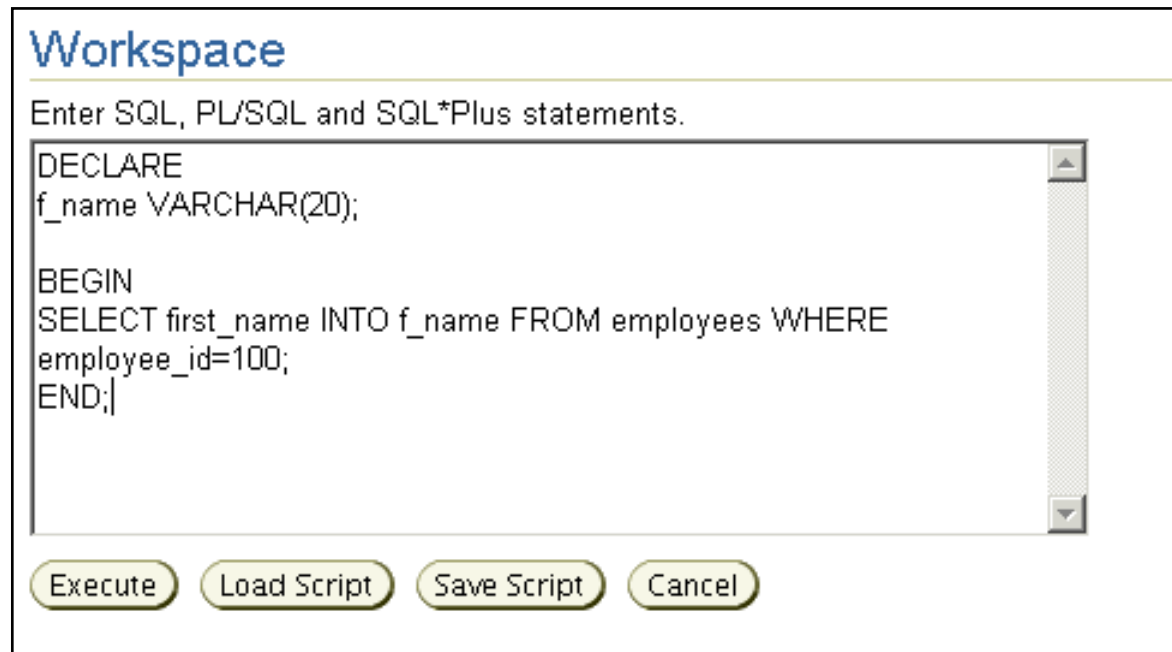


iSQL*PLUS ARCHITECTURE



CREATE AN ANONYMOUS BLOCK

- Type the anonymous block in the *iSQL*Plus* workspace:



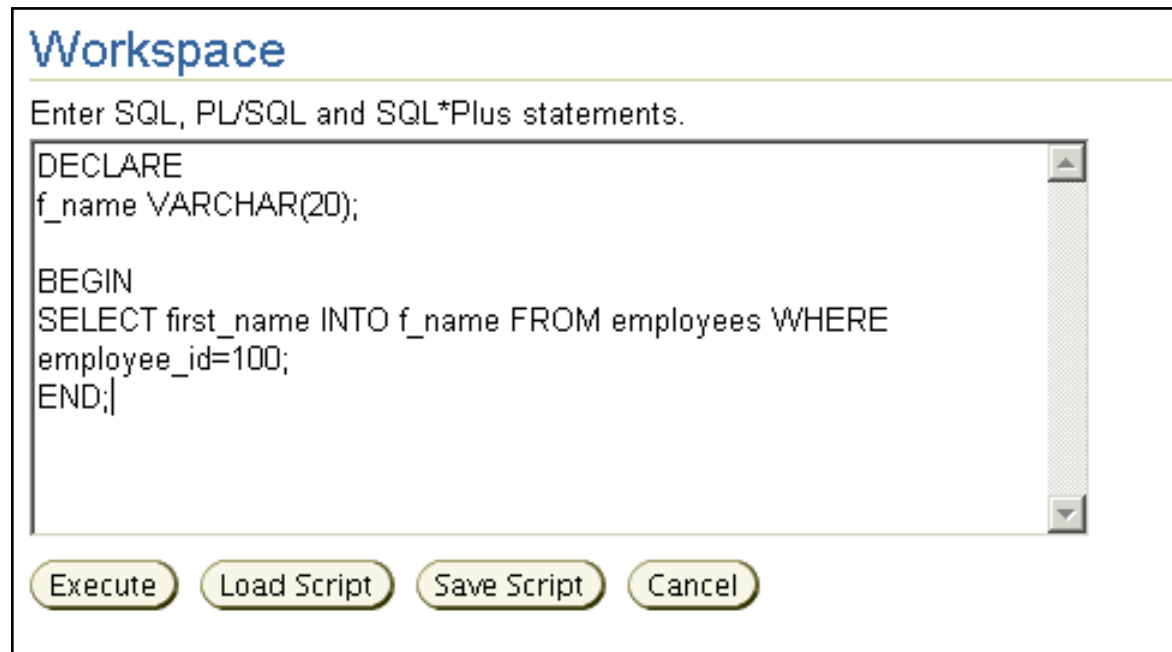
The screenshot shows the 'Workspace' window in iSQL*Plus. It has a title bar 'Workspace' and a subtitle 'Enter SQL, PL/SQL and SQL*Plus statements.' Below this is a text area containing the following PL/SQL code:

```
DECLARE  
f_name VARCHAR(20);  
  
BEGIN  
SELECT first_name INTO f_name FROM employees WHERE  
employee_id=100;  
END;
```

At the bottom of the window are four buttons: 'Execute', 'Load Script', 'Save Script', and 'Cancel'.

EXECUTE AN ANONYMOUS BLOCK

- Click the Execute button to execute the anonymous block:



The screenshot shows the 'Workspace' window in Oracle SQL Developer. It contains a text area for entering SQL, PL/SQL, and SQL*Plus statements. The code entered is an anonymous PL/SQL block that declares a variable and performs a SELECT statement. Below the text area are four buttons: 'Execute', 'Load Script', 'Save Script', and 'Cancel'.

Workspace

Enter SQL, PL/SQL and SQL*Plus statements.

```
DECLARE
f_name VARCHAR(20);

BEGIN
SELECT first_name INTO f_name FROM employees WHERE
employee_id=100;
END;
```

Execute Load Script Save Script Cancel

PL\SQL procedure successfully completed.

TEST THE OUTPUT OF A PL/SQL BLOCK

- Enable output in *iSQL*Plus* with the following command:
SET SERVEROUTPUT ON
- Use a predefined Oracle package and its procedure:
 - DBMS_OUTPUT.PUT_LINE

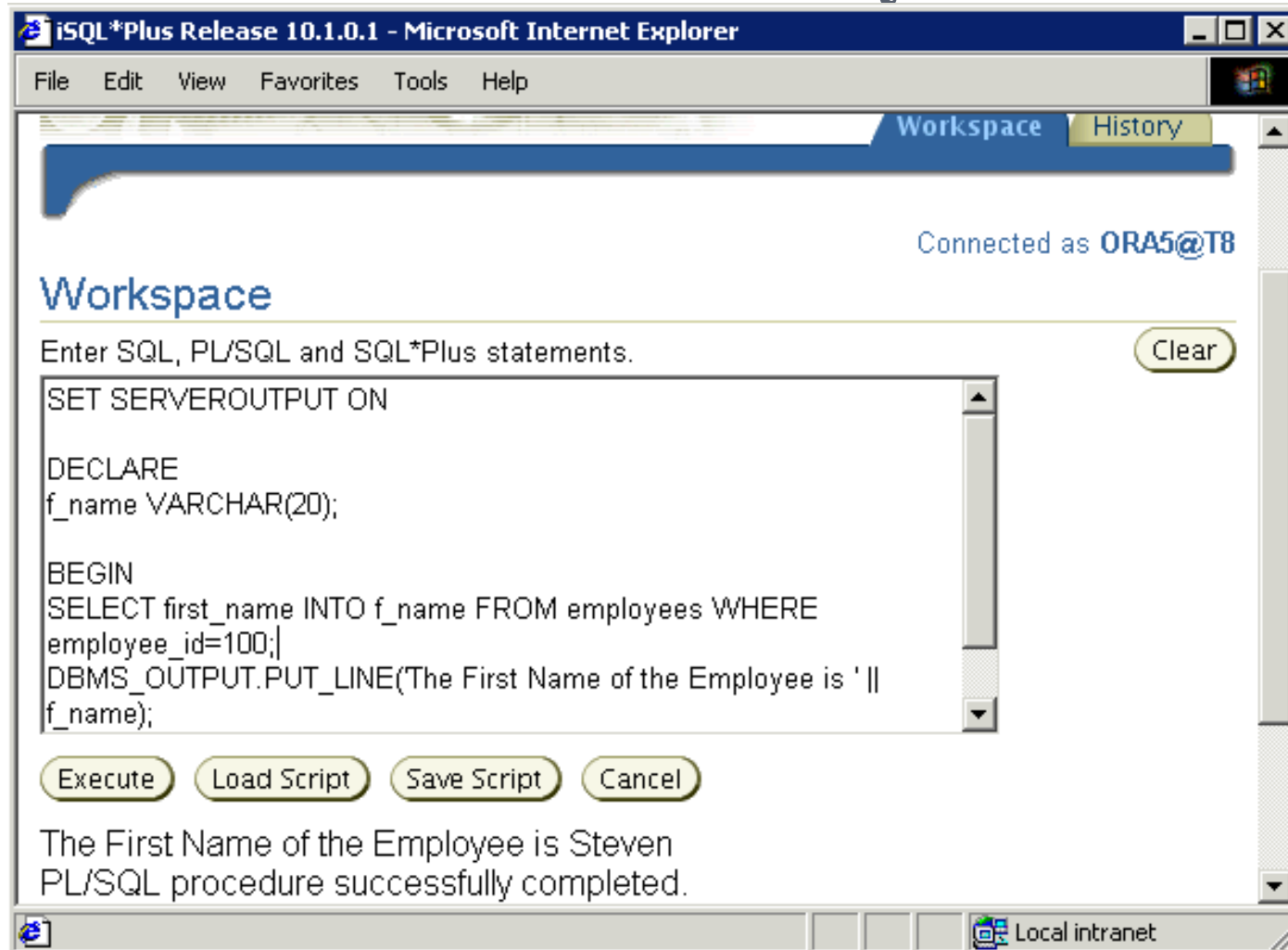
```
SET SERVEROUTPUT ON
```

```
...
```

```
DBMS_OUTPUT.PUT_LINE(' The First Name of the  
Employee is ' || f_name);
```

```
...
```

TEST THE OUTPUT OF A PL/SQL BLOCK



SUMMARY

- In this lesson, you should have learned how to:
 - Integrate SQL statements with PL/SQL program constructs
 - Identify the benefits of PL/SQL
 - Differentiate different PL/SQL block types
 - Use *iSQL*Plus* as the programming environment for PL/SQL
 - Output messages in PL/SQL

PRACTICE 1: OVERVIEW

- This practice covers the following topics:
 - Identifying which PL/SQL blocks execute successfully
 - Creating and executing a simple PL/SQL block