



PL/SQL

pl/sql programming

tutorialspoint

SIMPLY EASY LEARNING

www.tutorialspoint.com



<https://www.facebook.com/tutorialspointindia>



<https://twitter.com/tutorialspoint>

About the Tutorial

PL/SQL is a combination of SQL along with the procedural features of programming languages. It was developed by Oracle Corporation in the early 90's to enhance the capabilities of SQL.

PL/SQL is one of three key programming languages embedded in the Oracle Database, along with SQL itself and Java.

This tutorial will give you great understanding on PL/SQL to proceed with Oracle database and other advanced RDBMS concepts.

Audience

This tutorial is designed for Software Professionals, who are willing to learn PL/SQL Programming Language in simple and easy steps. This tutorial will give you great understanding on PL/SQL Programming concepts, and after completing this tutorial, you will be at an intermediate level of expertise from where you can take yourself to a higher level of expertise.

Prerequisites

Before proceeding with this tutorial, you should have a basic understanding of software basic concepts like what is database, source code, text editor and execution of programs, etc. If you already have an understanding on SQL and other computer programming language, then it will be an added advantage to proceed.

Copyright & Disclaimer

© Copyright 2018 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

About the Tutorial.....	i
Audience	i
Prerequisites	i
Copyright & Disclaimer.....	i
Table of Contents	iii
1. PL/SQL — OVERVIEW	1
Features of PL/SQL.....	1
Advantages of PL/SQL	1
2. PL/SQL — ENVIRONMENT SETUP	3
Text Editor.....	14
3. PL/SQL — BASIC SYNTAX	15
4. PL/SQL — DATA TYPES.....	19
PL/SQL Scalar Data Types and Subtypes	19
PL/SQL Numeric Data Types and Subtypes	20
PL/SQL Character Data Types and Subtypes	21
PL/SQL Boolean Data Types	22
PL/SQL Datetime and Interval Types	22
PL/SQL Large Object (LOB) Data Types	23
PL/SQL User-Defined Subtypes.....	24
NULLs in PL/SQL	25
5. PL/SQL — VARIABLES	26
Variable Declaration in PL/SQL.....	26
Initializing Variables in PL/SQL	27
Variable Scope in PL/SQL	28

	Assigning SQL Query Results to PL/SQL Variables	29
6.	PL/SQL — CONSTANTS AND LITERALS	31
	Declaring a Constant	31
	The PL/SQL Literals.....	32
7.	PL/SQL — OPERATORS	34
	Arithmetic Operators	34
	Relational Operators	36
	[Comparison Operators]	39
	Logical Operators	44
	PL/SQL Operator Precedence	46
8.	PL/SQL — CONDITIONS	49
	IF-THEN Statement.....	50
	IF-THEN-ELSE Statement.....	53
	IF-THEN-ELSIF Statement.....	55
	CASE Statement	56
	Searched CASE Statement	58
	Nested IF-THEN-ELSE Statements.....	60
9.	PL/SQL — LOOPS	62
	Basic Loop Statement.....	63
	WHILE LOOP Statement	65
	FOR LOOP Statement	66
	Reverse FOR LOOP Statement	68
	Nested Loops	69
	Labeling a PL/SQL Loop	71
	The Loop Control Statements.....	72

EXIT Statement	73
The EXIT WHEN Statement.....	75
CONTINUE Statement.....	77
GOTO Statement.....	80
10. PL/SQL — STRINGS	83
Declaring String Variables	83
PL/SQL String Functions and Operators.....	84
11. PL/SQL — ARRAYS	89
Creating a Varray Type	89
12. PL/SQL — PROCEDURES	94
Parts of a PL/SQL Subprogram.....	94
Creating a Procedure.....	95
Executing a Standalone Procedure	96
Deleting a Standalone Procedure	97
Parameter Modes in PL/SQL Subprograms.....	98
Methods for Passing Parameters.....	100
13. PL/SQL — FUNCTIONS.....	103
Creating a Function	103
Calling a Function	104
PL/SQL Recursive Functions	106
14. PL/SQL — CURSORS.....	108
Implicit Cursors	108
Explicit Cursors.....	110
Declaring the Cursor.....	112
Opening the Cursor	112

Fetching the Cursor	112
Closing the Cursor	112
15. PL/SQL — RECORDS.....	114
Table-Based Records	114
Cursor-Based Records	115
User-Defined Records.....	116
16. PL/SQL — EXCEPTIONS	120
Syntax for Exception Handling.....	120
Raising Exceptions.....	121
User-defined Exceptions	122
Pre-defined Exceptions	123
17. PL/SQL — TRIGGERS.....	126
Creating Triggers	126
Triggering a Trigger	129
18. PL/SQL — PACKAGES.....	130
Package Specification	130
Package Body	130
Using the Package Elements.....	131
19. PL/SQL — COLLECTIONS.....	138
Index-By Table	139
Nested Tables.....	141
Collection Methods	144
Collection Exceptions	145

20. PL/SQL — TRANSACTIONS	147
Starting and Ending a Transaction	147
Committing a Transaction	147
Rolling Back Transactions	148
Automatic Transaction Control	149
21. PL/SQL — DATE & TIME.....	150
Field Values for Datetime and Interval Data Types.....	150
The Datetime Data Types and Functions	151
The Interval Data Types and Functions.....	155
22. PL/SQL — DBMS OUTPUT.....	157
DBMS_OUTPUT Subprograms	157
23. PL/SQL — OBJECT-ORIENTED	160
Instantiating an Object.....	161
Member Methods	161
Using Map method.....	162
Using Order method.....	164
Inheritance for PL/SQL Objects	166
Abstract Objects in PL/SQL.....	168

1. PL/SQL — Overview

The PL/SQL programming language was developed by Oracle Corporation in the late 1980s as procedural extension language for SQL and the Oracle relational database. Following are certain notable facts about PL/SQL:

- PL/SQL is a completely portable, high-performance transaction-processing language.
- PL/SQL provides a built-in, interpreted and OS independent programming environment.
- PL/SQL can also directly be called from the command-line **SQL*Plus interface**.
- Direct call can also be made from external programming language calls to database.
- PL/SQL's general syntax is based on that of ADA and Pascal programming language.
- Apart from Oracle, PL/SQL is available in **TimesTen in-memory database** and **IBM DB2**.

Features of PL/SQL

PL/SQL has the following features:

- PL/SQL is tightly integrated with SQL.
- It offers extensive error checking.
- It offers numerous data types.
- It offers a variety of programming structures.
- It supports structured programming through functions and procedures.
- It supports object-oriented programming.
- It supports the development of web applications and server pages.

Advantages of PL/SQL

PL/SQL has the following advantages:

- SQL is the standard database language and PL/SQL is strongly integrated with SQL. PL/SQL supports both static and dynamic SQL. Static SQL supports DML operations and transaction control from PL/SQL block. In Dynamic SQL, SQL allows embedding DDL statements in PL/SQL blocks.

- PL/SQL allows sending an entire block of statements to the database at one time. This reduces network traffic and provides high performance for the applications.
- PL/SQL gives high productivity to programmers as it can query, transform, and update data in a database.
- PL/SQL saves time on design and debugging by strong features, such as exception handling, encapsulation, data hiding, and object-oriented data types.
- Applications written in PL/SQL are fully portable.
- PL/SQL provides high security level.
- PL/SQL provides access to predefined SQL packages.
- PL/SQL provides support for Object-Oriented Programming.
- PL/SQL provides support for developing Web Applications and Server Pages.

2. PL/SQL — Environment Setup

In this chapter, we will discuss the Environment Setup of PL/SQL. PL/SQL is not a stand-alone programming language; it is a tool within the Oracle programming environment. **SQL* Plus** is an interactive tool that allows you to type SQL and PL/SQL statements at the command prompt. These commands are then sent to the database for processing. Once the statements are processed, the results are sent back and displayed on screen.

To run PL/SQL programs, you should have the Oracle RDBMS Server installed in your machine. This will take care of the execution of the SQL commands. The most recent version of Oracle RDBMS is 11g. You can download a trial version of Oracle 11g from the following link:

[Download Oracle 11g Express Edition](#)

You will have to download either the 32-bit or the 64-bit version of the installation as per your operating system. Usually there are two files. We have downloaded the 64-bit version. You will also use similar steps on your operating system, does not matter if it is Linux or Solaris.

- **win64_11gR2_database_1of2.zip**
- **win64_11gR2_database_2of2.zip**

After downloading the above two files, you will need to unzip them in a single directory **database** and under that you will find the following sub-directories:

 doc	3/24/2010 12:15 AM	File folder	
 install	3/30/2010 8:05 AM	File folder	
 response	3/30/2010 9:31 AM	File folder	
 stage	3/30/2010 9:31 AM	File folder	
 setup	3/12/2010 1:11 AM	Application	334 KB
 welcome	3/16/2010 1:42 PM	HTML Document	6 KB

Step 1

Let us now launch the Oracle Database Installer using the setup file. Following is the first screen. You can provide your email ID and check the checkbox as shown in the following screenshot. Click the **Next** button.



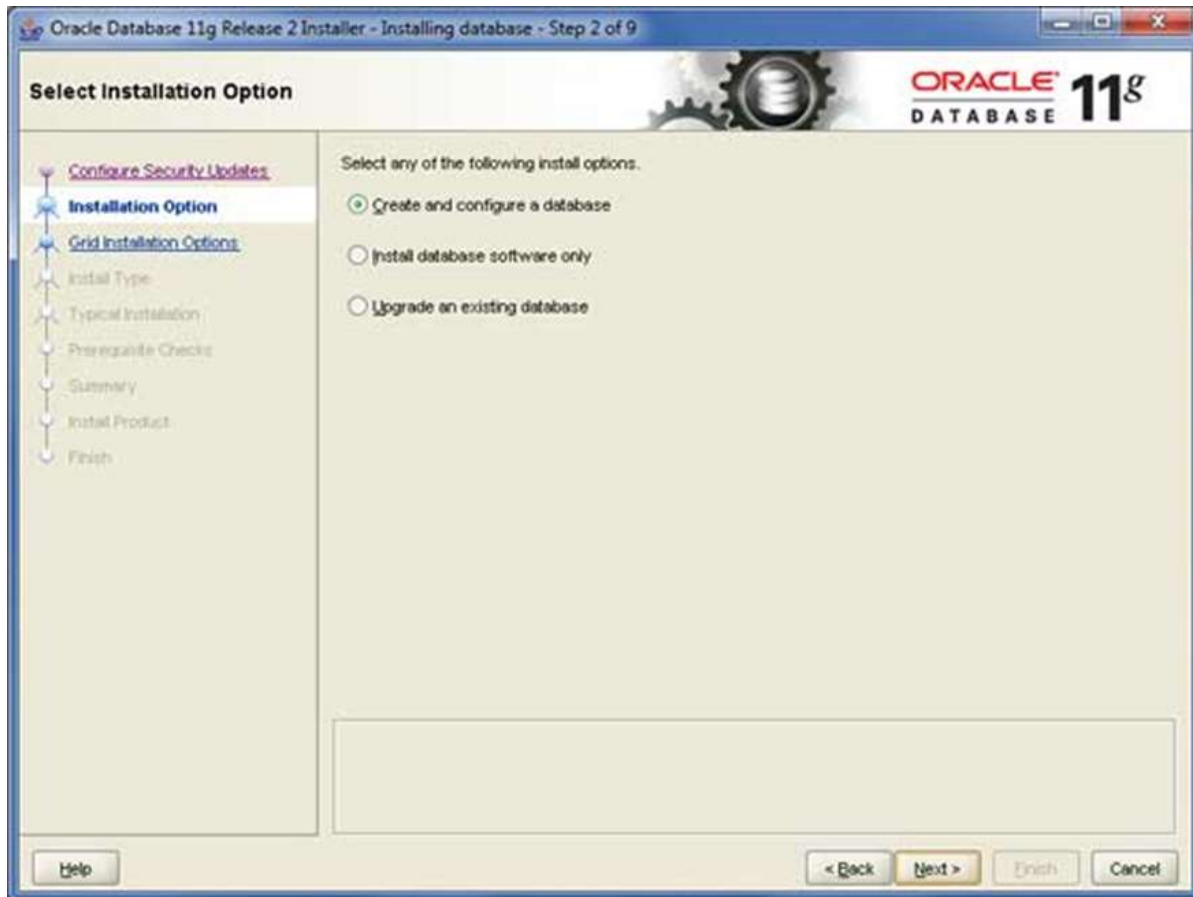
Step 2

You will be directed to the following screen; uncheck the checkbox and click the **Continue** button to proceed.



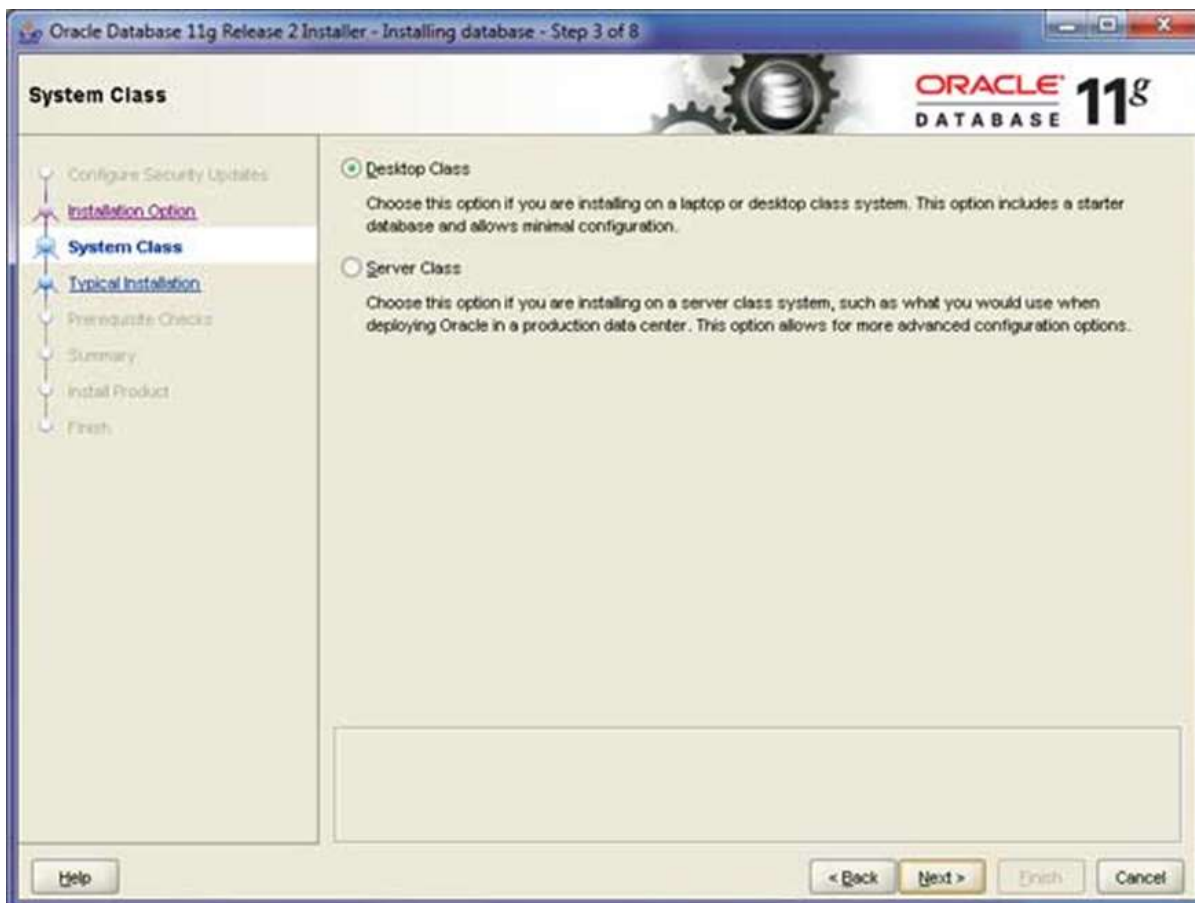
Step 3

Just select the first option **Create and Configure Database** using the radio button and click the **Next** button to proceed.



Step 4

We assume you are installing Oracle for the basic purpose of learning and that you are installing it on your PC or Laptop. Thus, select the **Desktop Class** option and click the **Next** button to proceed.



Step 5

Provide a location, where you will install the Oracle Server. Just modify the **Oracle Base** and the other locations will set automatically. You will also have to provide a password; this will be used by the system DBA. Once you provide the required information, click the **Next** button to proceed.

Oracle Database 11g Release 2 Installer - Installing database - Step 4 of 8

Typical Install Configuration

Perform full Database installation with basic configuration.

Oracle base: C:\app\ZARA

Software location: C:\app\ZARA\product11.2.0\sdhome_1

Database file location: C:\app\ZARA\oradata

Database edition: Enterprise Edition (3.34GB)

Character Set: Default (WE8MSWIN1252)

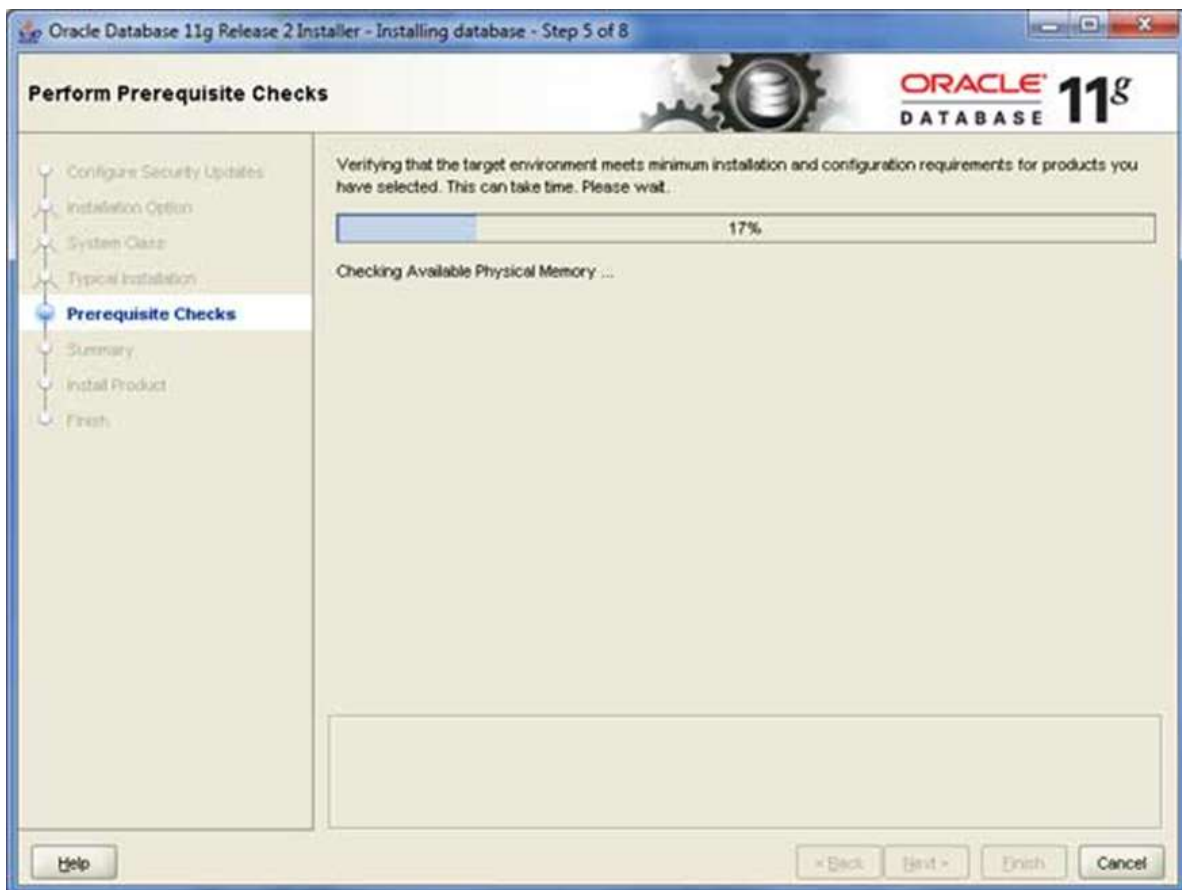
Global database name: orcl

Administrative password:

Confirm Password:

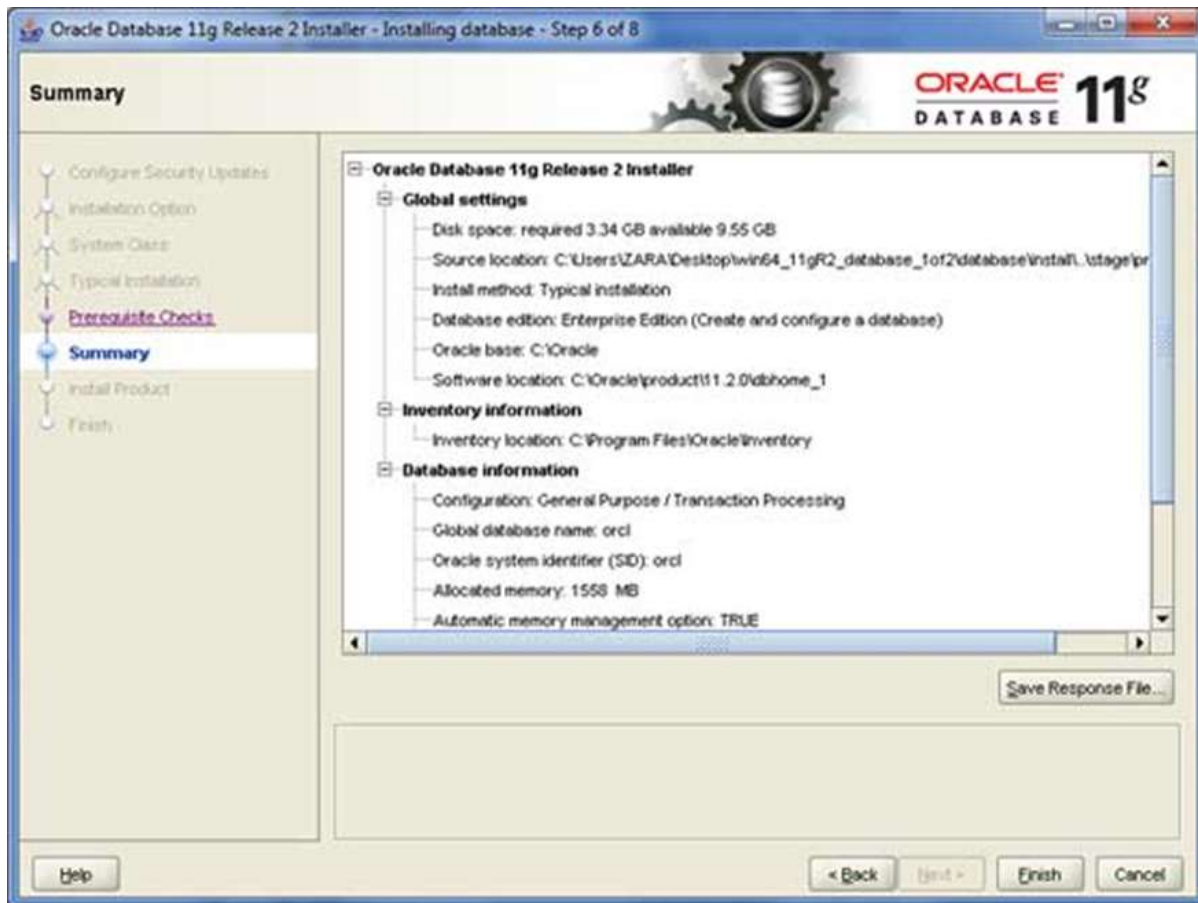
Step 6

Again, click the **Next** button to proceed.



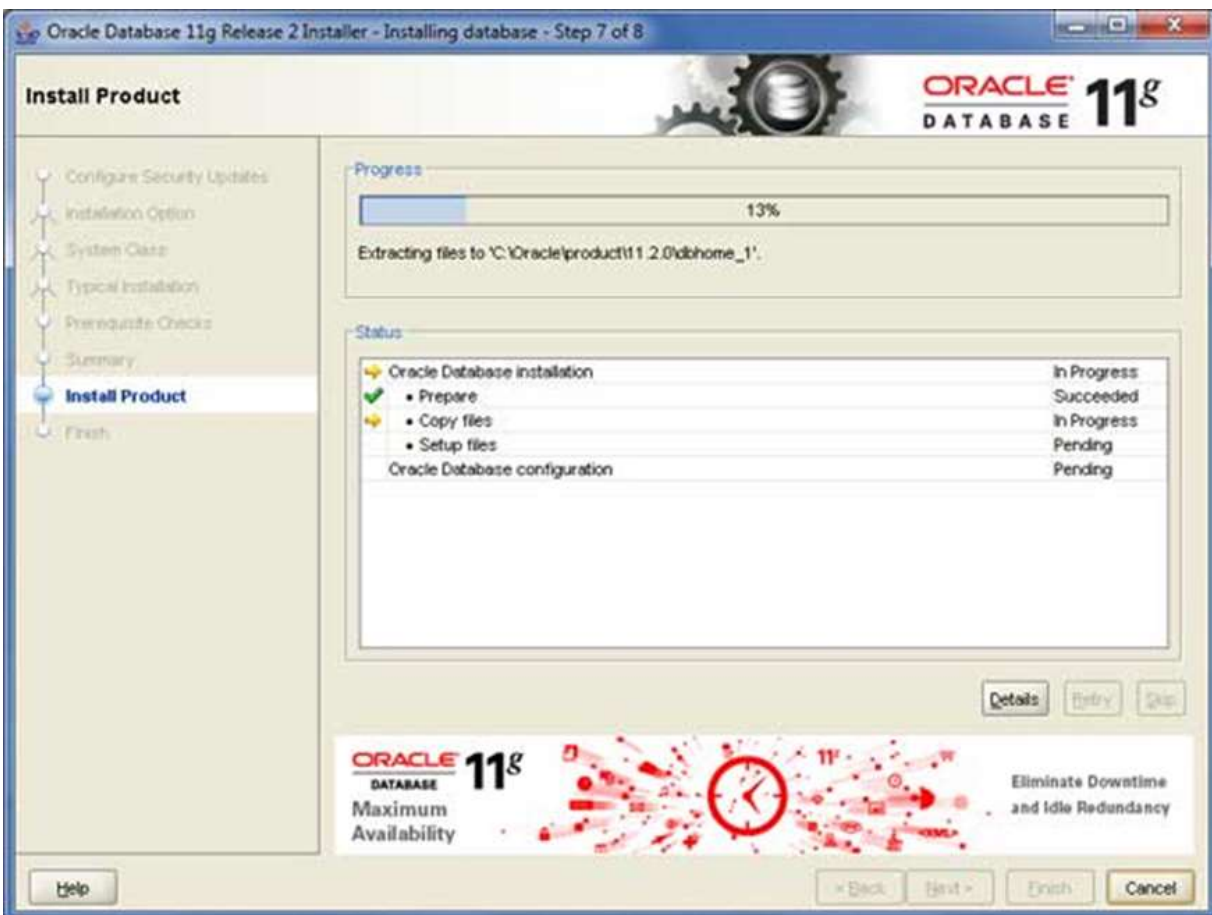
Step 7

Click the **Finish** button to proceed; this will start the actual server installation.



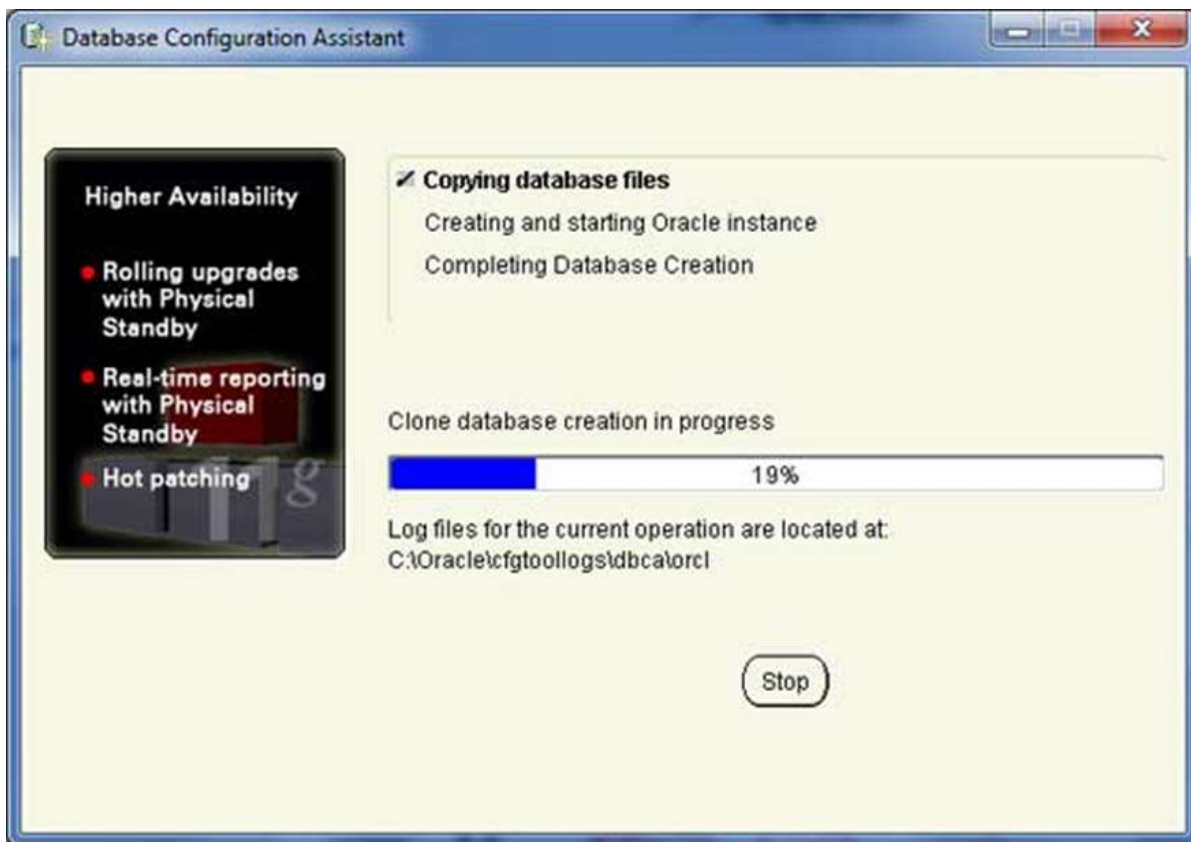
Step 8

This will take a few moments, until Oracle starts performing the required configuration.



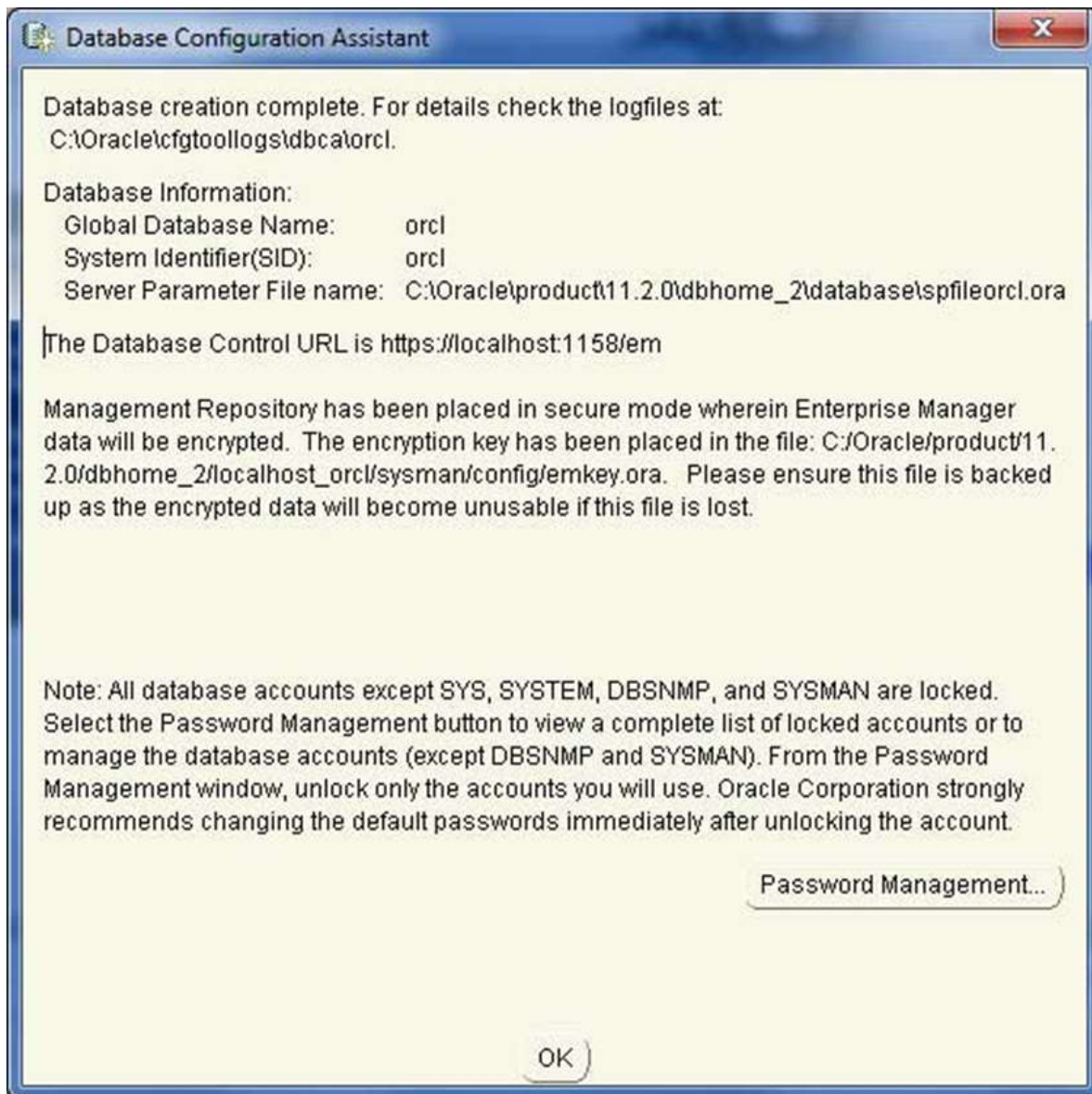
Step 9

Here, Oracle installation will copy the required configuration files. This should take a moment:



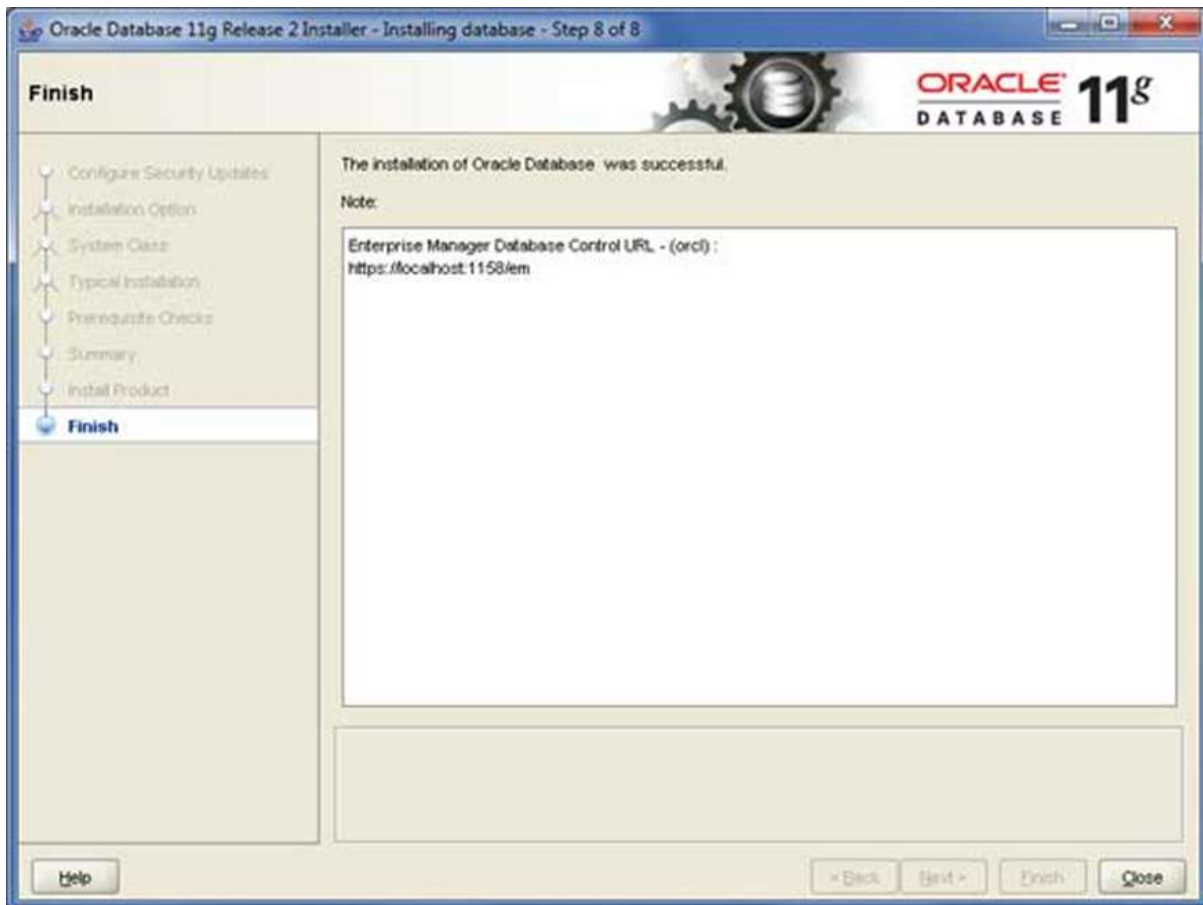
Step 10

Once the database files are copied, you will have the following dialogue box. Just click the **OK** button and come out.



Step 11

Upon installation, you will have the following final window.




Final Step

It is now time to verify your installation. At the command prompt, use the following command if you are using Windows:

```
sqlplus "/ as sysdba"
```

You should have the SQL prompt where you will write your PL/SQL commands and scripts:



```
Administrator: C:\Windows\system32\cmd.exe - sqlplus "/ as sysdba"
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\ZARA>sqlplus "/ as sysdba"

SQL*Plus: Release 11.2.0.1.0 Production on Fri Aug 31 19:15:57 2012
Copyright (c) 1982, 2010, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Producti
With the Partitioning, OLAP, Data Mining and Real Application Testing opti
SQL>
```

Text Editor

Running large programs from the command prompt may land you in inadvertently losing some of the work. It is always recommended to use the command files. To use the command files:

- Type your code in a text editor, like **Notepad**, **Notepad+**, or **EditPlus**, etc.
- Save the file with the **.sql** extension in the home directory.
- Launch the **SQL*Plus command prompt** from the directory where you created your PL/SQL file.
- Type **@file_name** at the SQL*Plus command prompt to execute your program.

If you are not using a file to execute the PL/SQL scripts, then simply copy your PL/SQL code and right-click on the black window that displays the SQL prompt; use the **paste** option to paste the complete code at the command prompt. Finally, just press **Enter** to execute the code, if it is not already executed.

3. PL/SQL — Basic Syntax

In this chapter, we will discuss the Basic Syntax of PL/SQL which is a **block-structured** language; this means that the PL/SQL programs are divided and written in logical blocks of code. Each block consists of three sub-parts:

Sr. No.	Sections & Description
1	Declarations This section starts with the keyword DECLARE . It is an optional section and defines all variables, cursors, subprograms, and other elements to be used in the program.
2	Executable Commands This section is enclosed between the keywords BEGIN and END and it is a mandatory section. It consists of the executable PL/SQL statements of the program. It should have at least one executable line of code, which may be just a NULL command to indicate that nothing should be executed.
3	Exception Handling This section starts with the keyword EXCEPTION . This optional section contains exception(s) that handle errors in the program.

Every PL/SQL statement ends with a semicolon (**;**). PL/SQL blocks can be nested within other PL/SQL blocks using **BEGIN** and **END**. Following is the basic structure of a PL/SQL block:

```
DECLARE
    <declarations section>
BEGIN
    <executable command(s)>
EXCEPTION
    <exception handling>
END;
```

The 'Hello World' Example

```

DECLARE
    message varchar2(20):= 'Hello, World!';
BEGIN
    dbms_output.put_line(message);
END;
/

```

The **end;** line signals the end of the PL/SQL block. To run the code from the SQL command line, you may need to type **/** at the beginning of the first blank line after the last line of the code. When the above code is executed at the SQL prompt, it produces the following result:

```

Hello World

PL/SQL procedure successfully completed.

```

The PL/SQL Identifiers

PL/SQL identifiers are constants, variables, exceptions, procedures, cursors, and reserved words. The identifiers consist of a letter optionally followed by more letters, numerals, dollar signs, underscores, and number signs and should not exceed 30 characters.

By default, **identifiers are not case-sensitive**. So you can use **integer** or **INTEGER** to represent a numeric value. You cannot use a reserved keyword as an identifier.

The PL/SQL Delimiters

A delimiter is a symbol with a special meaning. Following is the list of delimiters in PL/SQL:

Delimiter	Description
+, -, *, /	Addition, subtraction/negation, multiplication, division
%	Attribute indicator
'	Character string delimiter

.	Component selector
(,)	Expression or list delimiter
:	Host variable indicator
,	Item separator
"	Quoted identifier delimiter
=	Relational operator
@	Remote access indicator
;	Statement terminator
:=	Assignment operator
=>	Association operator
	Concatenation operator
**	Exponentiation operator
<<, >>	Label delimiter (begin and end)
/*, */	Multi-line comment delimiter (begin and end)
--	Single-line comment indicator
..	Range operator
<, >, <=, >=	Relational operators

<>, !=, ~=, ^=	Different versions of NOT EQUAL
----------------	---------------------------------

The PL/SQL Comments

Program comments are explanatory statements that can be included in the PL/SQL code that you write and helps anyone reading its source code. All programming languages allow some form of comments.

The PL/SQL supports single-line and multi-line comments. All characters available inside any comment are ignored by the PL/SQL compiler. The PL/SQL single-line comments start with the delimiter -- (double hyphen) and multi-line comments are enclosed by /* and */.

```

DECLARE
    -- variable declaration
    message varchar2(20):= 'Hello, World!';
BEGIN
    /*
    * PL/SQL executable statement(s)
    */
    dbms_output.put_line(message);
END;
/

```

When the above code is executed at the SQL prompt, it produces the following result:

```

Hello World

PL/SQL procedure successfully completed.

```

PL/SQL Program Units

A PL/SQL unit is any one of the following:

- PL/SQL block
- Function
- Package
- Package body

- Procedure
- Trigger
- Type
- Type body

Each of these units will be discussed in the following chapters.

End of ebook preview
If you liked what you saw...
Buy it from our store @ <https://store.tutorialspoint.com>