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 stands for Procedural Language extensions to the Structured Query Language (SQL). It 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.

In this tutorial, we'll give you a great understanding of PL/SQL to proceed with the Oracle database and other advanced RDBMS concepts.

Purpose of PL/SQL

The purpose of PL/SQL is to merge database commands with procedural programming language. It offers more complete programming solutions for building critical applications that operate on the Oracle database.

Features of PL/SQL

PL/SQL has the following features –

- PL/SQL is tightly integrated with SQL.
- It offers extensive error checking mechanisms.
- It supports numerous data types for flexible data handling.
- It includes a variety of programming structures, such as loops and conditionals.
- It supports structured programming through functions and procedures.
- It supports object-oriented programming, enabling more complex data handling and manipulation.
- It supports the web application development and server pages.

Why to learn PL/SQL?

Learning PL/SQL is an essential skill for persons who are interested in databases and other advanced RDBMS technologies. PL/SQL offers various benefits, making it an essential skill for database developers –

- **Ease of Use**: PL/SQL is straightforward to write and read, featuring block-structured syntax which simplifies programming and debugging.
- **Portability**: Programs written in PL/SQL are fully portable across different Oracle databases, ensuring consistency and ease of migration.
- **Tight SQL Integration**: PL/SQL is tightly integrated with SQL, allowing for efficient querying, transforming, and updating of data within a database.
- **High Performance**: It reduces network traffic by sending entire blocks of statements to the database at once, thus improving performance.
- **Security**: It includes robust security features to protect database integrity.

- **Object-Oriented Support**: It supports object-oriented programming, and allows you to define object types that can be used in object-oriented designs.

PL/SQL Block Structured

PL/SQL follows a block-structured approach, dividing programs into logical blocks of code. Each block consists of three main sections –

- **Declarations**: This section, starting with the keyword DECLARE, is optional and used for defining variables, cursors, subprograms, and other elements required within the block.
- **Executable Commands**: Enclosed between the keywords BEGIN and END, this mandatory section contains executable PL/SQL statements. It must include at least one executable line of code, even if it's just a NULL command indicating no action.
- **Exception Handling**: This starts with the keyword EXCEPTION. This optional section deals with handling errors in the program through defined exceptions.

PL/SQL statements are terminated with a semicolon(;). Additionally, blocks can be nested within each other using BEGIN and END keywords.

Applications of PL/SQL

PL/SQL is widely used in various applications, including –

- **Database Security**: It implements robust security measures within the database.
- **XML Management**: XML documents can be generated and managed within the database.
- **Linking Databases to Web Pages**: Integrates databases with web applications.
- **Automation**: Automating database administration tasks for efficient management.

Who should learn PL/SQL?

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 a great understanding of 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 to learn PL/SQL

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

PL/SQL Jobs and Opportunities

Proficiency in PL/SQL opens up various career opportunities, such as –

- Oracle PL/SQL Programmer
- PL/SQL Developer
- Database Developer
- Data Analyst
- Database Testers
- Data Scientist
- ETL Developer
- Database Migration Expert
- Cloud Database Expert etc

By mastering PL/SQL, you can increase your career opportunities in database management and development, as well as in creating secure and scalable applications.

Frequently Asked Questions about PL/SQL

There are some very Frequently Asked Questions(FAQs) about PL/SQL. This section tries to answer them briefly.

What are PL/SQL Records?

PL/SQL records are data structures designed to hold multiple data items of different types. They consist of various fields, much like a row in a database table.

Comparison of SQL and PL/SQL

SQL (Structured Query Language) is a standard language used for creating, manipulating, and retrieving data from relational databases. SQL is mainly used to write queries, as well as create and execute DDL (Data Definition Language) and DML (Data Manipulation Language) statements.

Whereas, PL/SQL (Procedural Language/SQL) is an extension of SQL. It adds procedural capabilities to SQL, enabling the creation of more complex and powerful database applications. PL/SQL supports variables, data types, and control structures such as loops and conditionals, which SQL does not. This makes PL/SQL more efficient for writing program blocks, functions, procedures, triggers, and packages.

How do you handle exceptions in PL/SQL?

When an exception is raised in PL/SQL, the current PL/SQL block stops its regular execution and transfers control to the exception section. The exception is then handled by an exception handler within the current PL/SQL block or passed to the enclosing block if not handled locally.

To manage an exception after it is raised, you need to write an exception handler for it. This handler should be placed after all executable statements in your PL/SQL block but before the END statement. The EXCEPTION keyword indicates the start of the exception-handling section.

Which is the best place to learn PL/SQL?

The best place to learn PL/SQL is through our comprehensive and user-friendly tutorial. Our PL/SQL tutorial provides an excellent starting point for understanding database programming with PL/SQL. You can explore our simple and effective learning materials at your own pace.

What do you understand by PL/SQL table?

A PL/SQL table, also known as an associative array or index-by table, is a collection of key-value pairs where each key is a unique index used to access the corresponding value. This table functions similarly to a relational table, storing data in rows and columns. Each column represents a different attribute or value, while each row contains individual records with values for all the columns.

How to learn PL/SQL?

Following are some tips to learn PL/SQL –

- The first and most crucial step is to decide to learn PL/SQL and stay committed to your goal.
- Install the necessary tools like SQL*Plus or Oracle SQL Developer on your computer.
- Start with our PL/SQL tutorial and progress step by step from the basics.
- Read more articles, watch online courses, or buy a book on PL/SQL to deepen your understanding.
- Apply what you've learned by developing small projects that incorporate PL/SQL and other technologies.

Which software is used for PL/SQL programming?

SQL*Plus and PL/SQL Developer are commonly used for writing and executing PL/SQL code.

What are the various Types of Parameters in PL/SQL?

In PL/SQL, there are two types of parameters –

- **Actual Parameters**: Actual parameters are the values or expressions provided in the parameter list when calling a module. In actual parameters, only the variable is mentioned, not the data types.
- **Formal Parameters**: Formal parameters are the names declared in the parameter list of a module's header.

Why should you learn PL/SQL?

Learning PL/SQL is crucial for a person who is interested in databases and advanced RDBMS technologies. It offers numerous benefits, making it an essential skill for database developers. With PL/SQL, you can efficiently manipulate and manage data within Oracle databases, which increases your ability to work with large datasets and complex queries.

Copyright & Disclaimer

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

PL/SQL Tutorial

1. 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 calls to database can also be made from external programming languages.
- PL/SQL's general syntax is based on ADA and Pascal programming languages.
- 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 standalone 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.

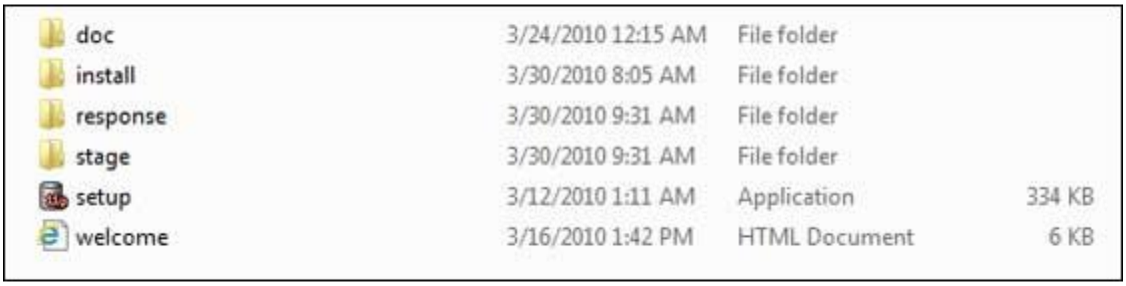
To run PL/SQL programs, you should have the Oracle RDBMS Server installed in your machine. This will take care of the execution of the SQL commands. The most recent version of Oracle RDBMS is 11g. You can download a trial version of Oracle 11g from the following link –

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**

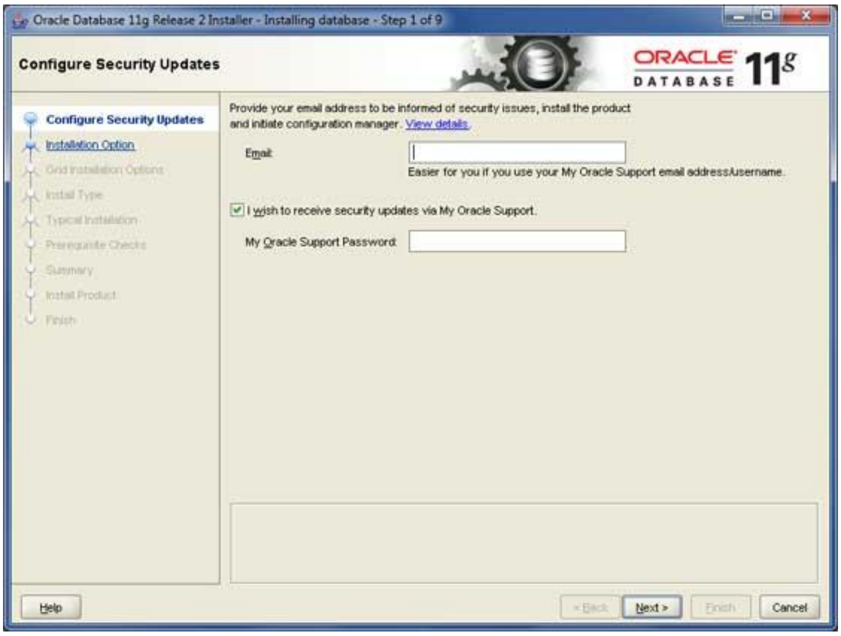
After downloading the above two files, you will need to unzip them in a single directory **database** and under that you will find the following sub-directories –



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.



Oracle Database 11g Release 2 Installer - Installing database - Step 1 of 9

Configure Security Updates

Provide your email address to be informed of security issues, install the product and initiate configuration manager. View details.

Email:

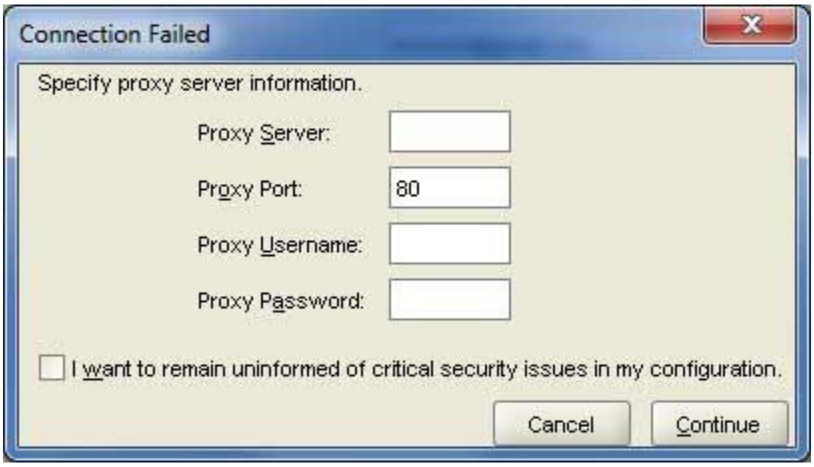
Easier for you if you use your My Oracle Support email address/username.

☑ I wish to receive security updates via My Oracle Support.

My Oracle Support Password:

Step 2

You will be directed to the following scre.n; Uncheck the checkbox and click the **Continue** button to proceed.



Connection Failed

Specify proxy server information.

Proxy Server:

Proxy Port: 80

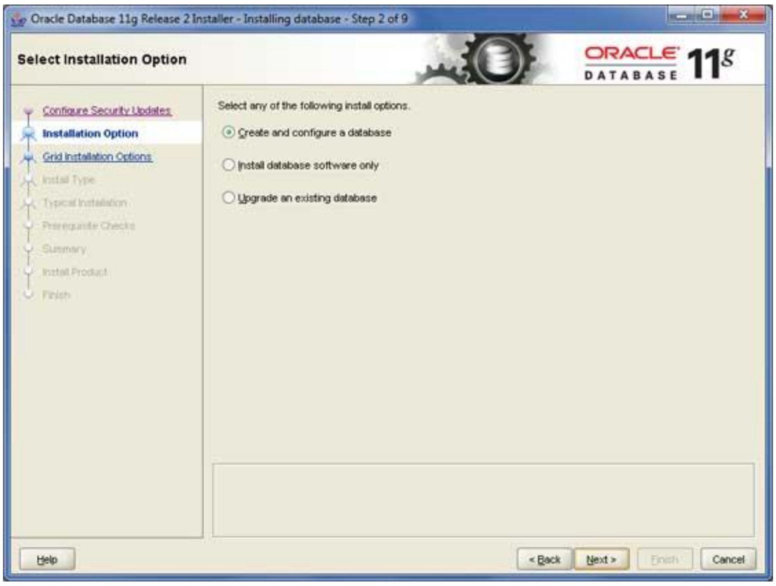
Proxy Username:

Proxy Password:

□ I want to remain uninformed of critical security issues in my configuration.

Step 3

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



Oracle Database 11g Release 2 Installer - Installing database - Step 2 of 9

Select Installation Option

Select any of the following install options.

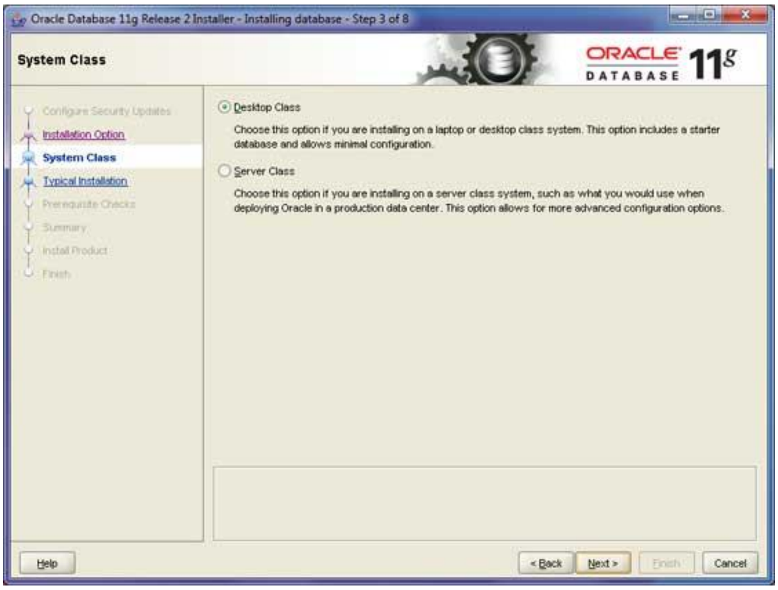
● Create and configure a database

○ Install database software only

○ Upgrade an existing database

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.



Oracle Database 11g Release 2 Installer - Installing database - Step 3 of 8

System Class

● Desktop Class

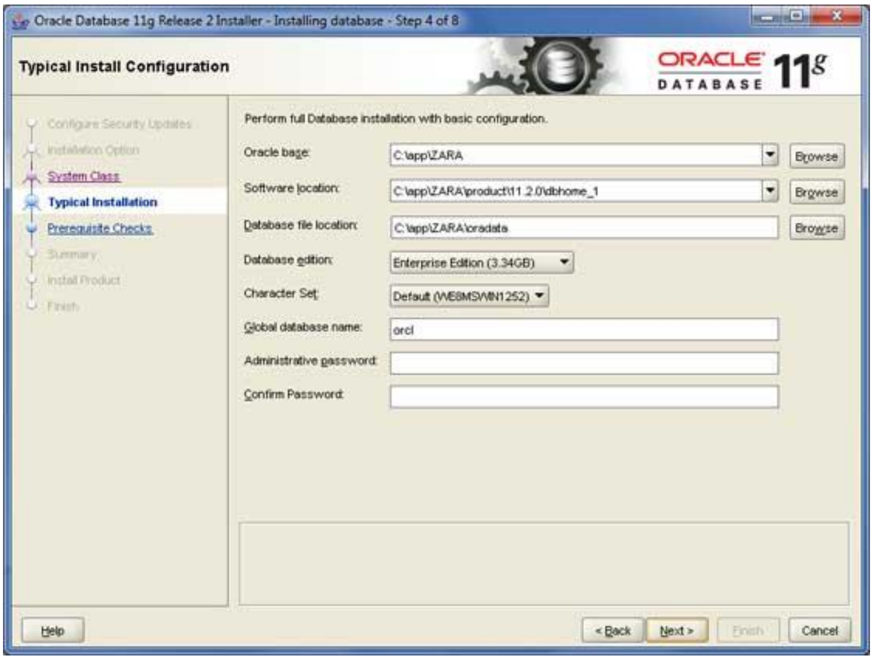
Choose this option if you are installing on a laptop or desktop class system. This option includes a starter database and allows minimal configuration.

○ Server Class

Choose this option if you are installing on a server class system, such as what you would use when deploying Oracle in a production data center. This option allows for more advanced configuration options.

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.



Perform full Database installation with basic configuration.

Oracle base: C:\app\ZARA

Software location: C:\app\ZARA\product\11.2.0\dbhome_1

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

Database edition: Enterprise Edition (3.34GB)

Character Set: Default (WE8MSWIN1252)

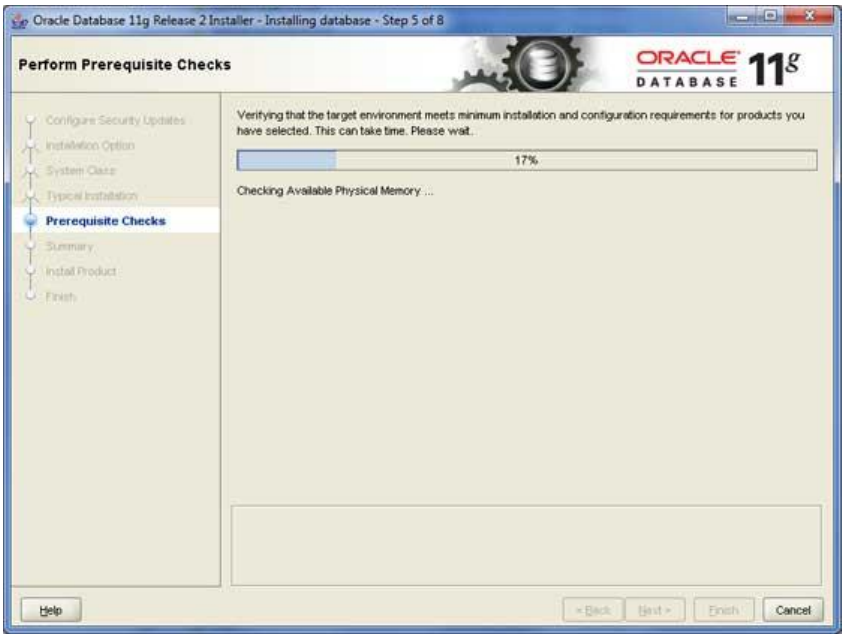
Global database name: orcl

Administrative password:

Confirm Password:

Step 6

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



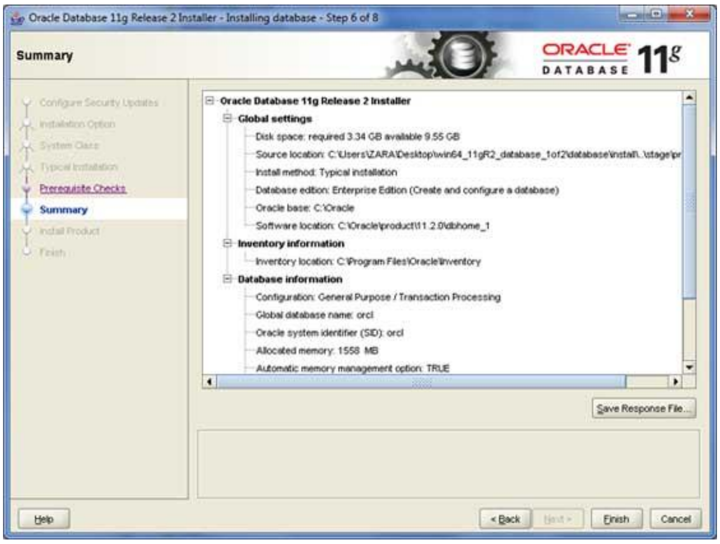
Verifying that the target environment meets minimum installation and configuration requirements for products you have selected. This can take time. Please wait.

17%

Checking Available Physical Memory ...

Step 7

Click the **Finish** button start the actual server installation.



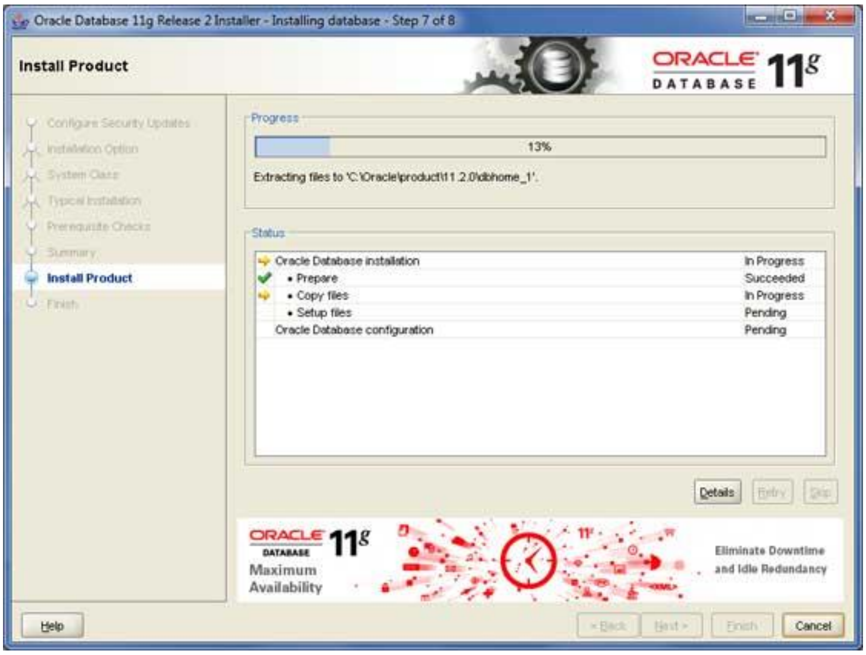
Oracle Database 11g Release 2 Installer - Installing database - Step 6 of 8

Summary

```
Oracle Database 11g Release 2 Installer
  Global settings
    Disk space: required 3.34 GB available 9.55 GB
    Source location: C:\Users\ZARA\Desktop\win64_11gR2_database_1of2\database\install\..\stage\pr
    Install method: Typical installation
    Database edition: Enterprise Edition (Create and configure a database)
    Oracle base: C:\Oracle
    Software location: C:\Oracle\product\11.2.0\dbhome_1
  Inventory information
    Inventory location: C:\Program Files\Oracle\Inventory
  Database information
    Configuration: General Purpose / Transaction Processing
    Global database name: orcl
    Oracle system identifier (SID): orcl
    Allocated memory: 1558 MB
    Automatic memory management option: TRUE
```

Step 8

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



Oracle Database 11g Release 2 Installer - Installing database - Step 7 of 8

Install Product

Progress

13%

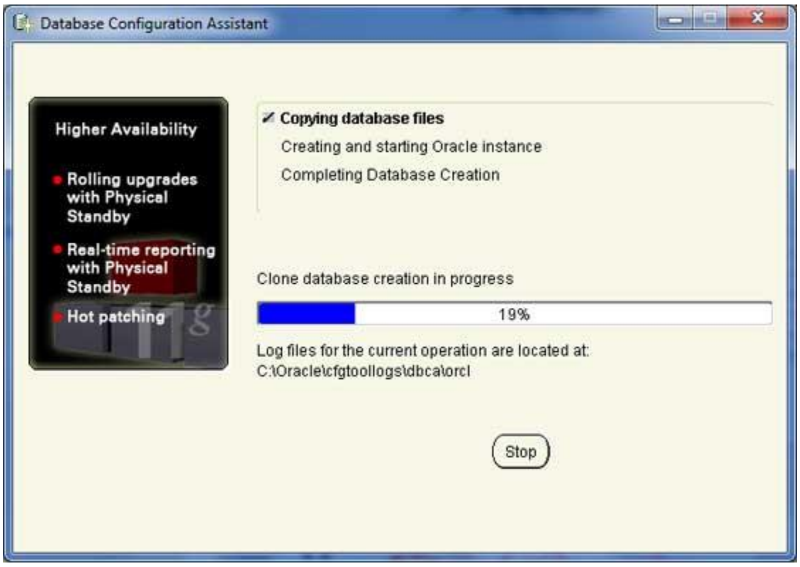
Extracting files to 'C:\Oracle\product\11.2.0\dbhome_1'.

Status

Oracle Database installation	In Progress
• Prepare	Succeeded
• Copy files	In Progress
• Setup files	Pending
Oracle Database configuration	Pending

Step 9

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


Database Configuration Assistant
Higher Availability
Rolling upgrades with Physical Standby
Real-time reporting with Physical Standby
Hot patching
Stop

✓ Copying database files
Creating and starting Oracle instance
Completing Database Creation

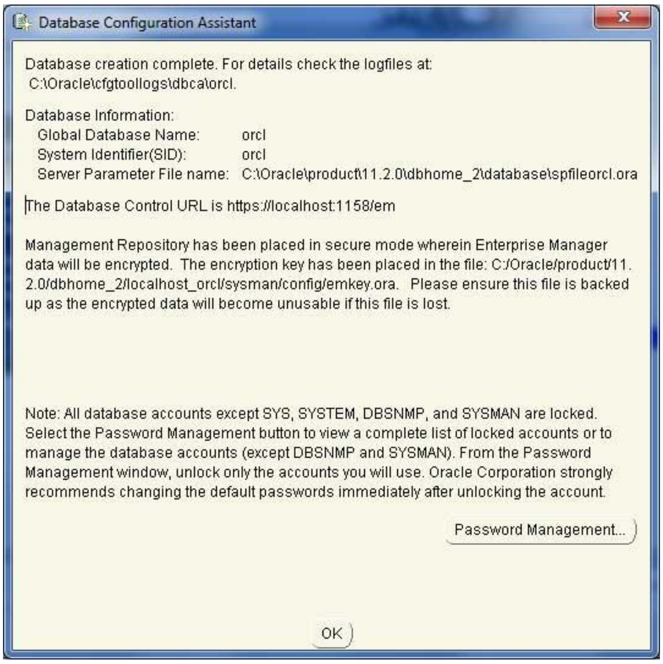
Clone database creation in progress

19%

Log files for the current operation are located at:
C:\Oracle\cfgtoollogs\dbca\orcl

Step 10

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



Database creation complete. For details check the logfiles at:
C:\Oracle\cfgtoollogs\dbca\orcl.

```
Database Information:
   Global Database Name:       orcl
   System Identifier(SID):     orcl
   Server Parameter File name: C:\Oracle\product\11.2.0\dbhome_2\database\spfileorcl.ora
```

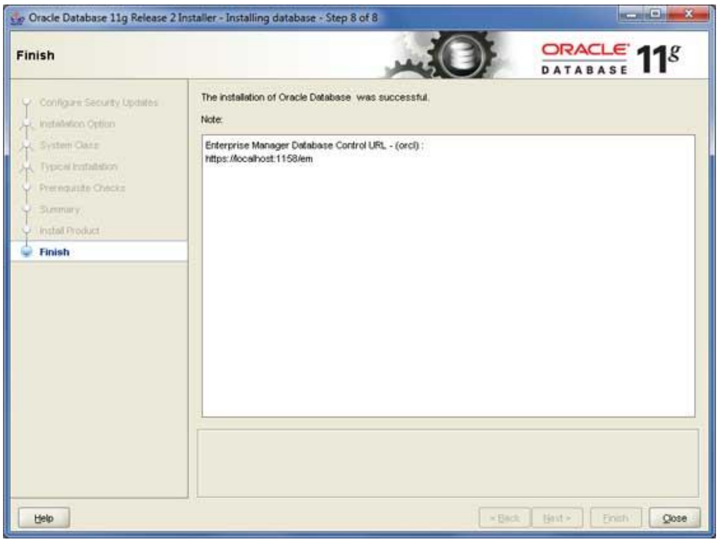
The Database Control URL is https://localhost:1158/em

Management Repository has been placed in secure mode wherein Enterprise Manager data will be encrypted. The encryption key has been placed in the file: C:/Oracle/product/11.2.0/dbhome_2/localhost_orcl/sysman/config/emkey.ora. Please ensure this file is backed up as the encrypted data will become unusable if this file is lost.

Note: All database accounts except SYS, SYSTEM, DBSNMP, and SYSMAN are locked. Select the Password Management button to view a complete list of locked accounts or to manage the database accounts (except DBSNMP and SYSMAN). From the Password Management window, unlock only the accounts you will use. Oracle Corporation strongly recommends changing the default passwords immediately after unlocking the account.

Step 11

Upon installation, you will have the following final window.



The installation of Oracle Database was successful.

Note:

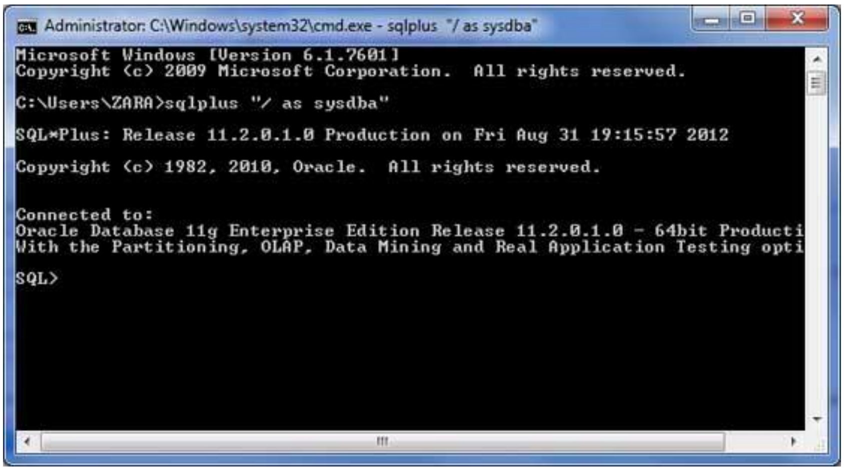
Enterprise Manager Database Control URL - (orcl) :
https://localhost:1158/em

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 –

S.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.