Learn PL/SQL in 1 Day

By Krishna Rungta

Table Of Content

Chapter 1: What Is PL/SQL? Introduction & Architecture

What is PL/SQL?

PL/SQL is an extension of Structured Query Language (SQL) that is used in Oracle. Unlike SQL, PL/SQL allows the programmer to write code in a procedural format. Full form of PL/SQL is “Procedural Language extensions to SQL”.

It combines the data manipulation power of SQL with the processing power of procedural language to create super powerful SQL queries.

PL/SQL means instructing the compiler ‘what to do’ through SQL and ‘how to do’ through its procedural way.

Similar to other database languages, it gives more control to the programmers by the use of loops, conditions and object-oriented concepts.

Architecture of PL/SQL

The PL/SQL architecture mainly consists of following three components:

1. PL/SQL block
2. PL/SQL Engine
3. Database Server

PL/SQL block:

- This is the component which has the actual PL/SQL code. This
- consists of different sections to divide the code logically

(declarative section for declaring purpose, execution section for processing statements, exception handling section for handling errors)

- It also contains the SQL instruction that used to interact with the database server.
- All the PL/SQL units are treated as PL/SQL blocks, and this is the starting stage of the architecture which serves as the primary input.
- Following are the different type of PL/SQL units.
    - Anonymous Block Function
    - Library Procedure
    - Package Body
    - Package Specification Trigger
    - Type
    - Type Body

PL/SQL Engine

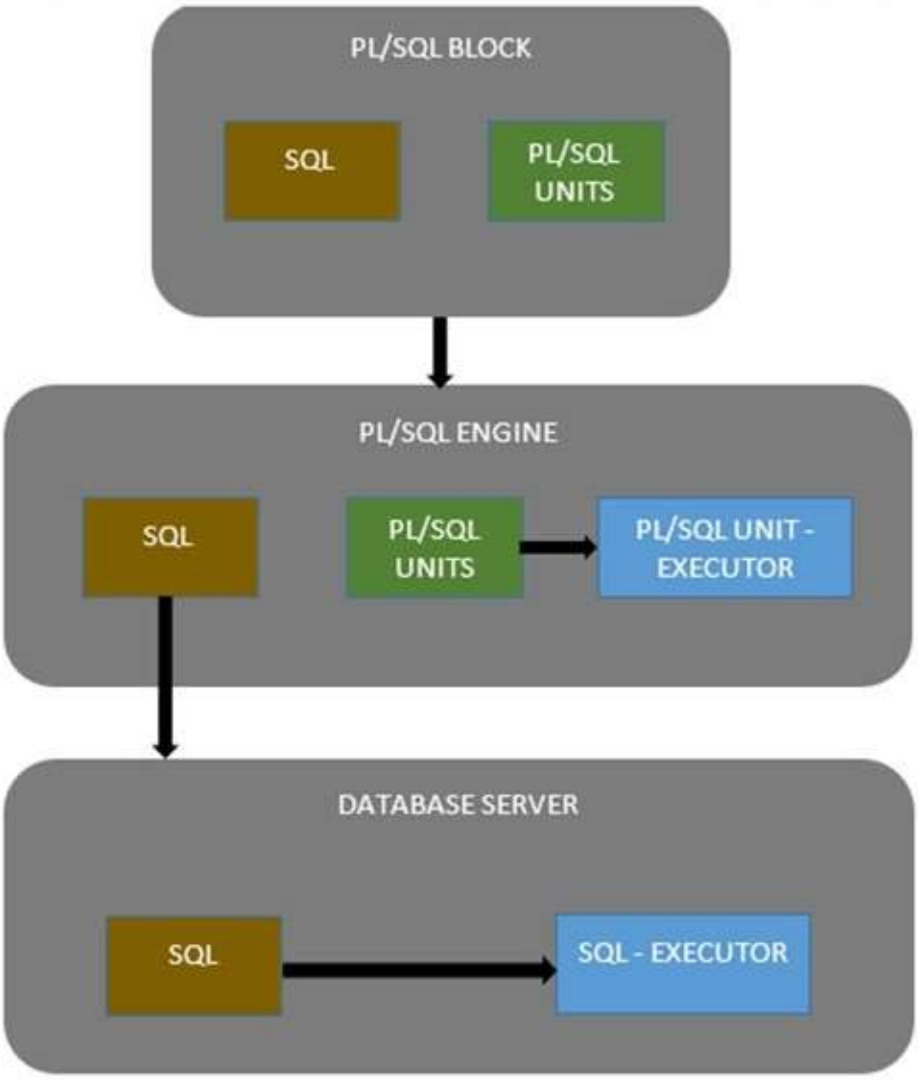
- PL/SQL engine is the component where the actual processing of the codes takes place.
- PL/SQL engine separates PL/SQL units and SQL part in the input (as shown in the image below).
- The separated PL/SQL units will be handled by the PL/SQL engine itself.
- The SQL part will be sent to database server where the actual interaction with database takes place.
- It can be installed in both database server and in the application server.

Database Server:

- This is the most important component of Pl/SQL unit which stores the data.
- The PL/SQL engine uses the SQL from PL/SQL units to interact with the database server.
- It consists of SQL executor which parses the input SQL statements and

execute the same.

Below is the pictorial representation of Architecture of PL/SQL.



PL/SQL Architecture Diagram

Advantage of Using PL/SQL

1. Better performance, as SQL is executed in bulk rather than a single statement
2. High Productivity
3. Tight integration with SQL
4. Full Portability
5. Tight Security
6. Support Object Oriented Programming concepts.

Chapter 2: SQL Vs PL/SQL Vs T-SQL: Key Differences

SQL is the standard language to query a database.

PL SQL basically stands for “Procedural Language extensions to SQL.” This is the extension of Structured Query Language (SQL) that is used in Oracle.

T-SQL basically stands for "Transact-SQL." This is the extension of Structured Query Language (SQL) that is used in Microsoft.

Difference between SQL and PL/SQL

SQL	PL/SQL
SQL is a single query that is used to perform DML and DDL operations.	PL/SQL is a block of codes that used to write the entire program blocks/ procedure/ function, etc.
It is declarative, that defines what need to be done, rather than how things need to be done.	PL/SQL is procedural that defines how the things needs to be done.
Execute as a single statement.	Execute as a whole block.
Mainly used to manipulate data.	Mainly used to create an application.
Interaction with a Database server.	No interaction with the database server.
Cannot contain PL/SQL code in it.	It is an extension of SQL, so that it can contain SQL inside it.

Difference Between T-SQL and PL- SQL

T-SQL	PL-SQL
T-SQL is a Microsoft product.	PL-SQL is developed by Oracle.
Full Form of TL SQL is Transact Structure Query language.	Full Form of PL SQL is Procedural Language Structural Query Language.
T-SQL gives a high degree of control to programmers.	It is a natural programming language that blends easily with the SQL

T-SQL performs best with Microsoft SQL server	PL-SQL performs best with Oracle database server.
It is easy and simple to understand.	PL-SQL is complex to understand.
T-SQL allows inserting multiples rows into a table using the BULK INSERT statement.	PL/SQL supports oops concepts like data encapsulation, function overloading, and information hiding.
SELECT INTO statement used in T-SQL	INSERT INTO statement must be used in PL/SQL
In T-SQL NOT EXISTS clause used along with SELECT statements.	In PL/SQL, there is a MINUS operator, which could be used with SELECT statements

Difference between SQL and T-SQL

SQL	T-SQL
SQL is a programming language which focuses on managing relational databases.	T-SQL is a procedural extension used by SQL Server.
This is used for controlling and manipulating data where large amounts of information are stored about products, clients, etc.	T-SQL has some features that are not available in SQL. Like procedural programming elements and a local variable to provide more flexible control of how the application flows.
SQL queries submitted individually to the database server.	T-SQL writes a program in such a way that all commands are submitted to the server in a single go
The syntax was formalized for many commands; some of these are SELECT, INSERT, UPDATE, DELETE, CREATE, and DROP.	It also includes special functions like the converted date () and some other functions which are not part of the regular SQL.

Chapter 3: PL/ SQL Block: STRUCTURE, Syntax, ANONYMOUS Example

What is PL/SQL block?

In PL/SQL, the code is not executed in single line format, but it is always executed by grouping the code into a single element called Blocks. In this tutorial, you are going to learn about these blocks.

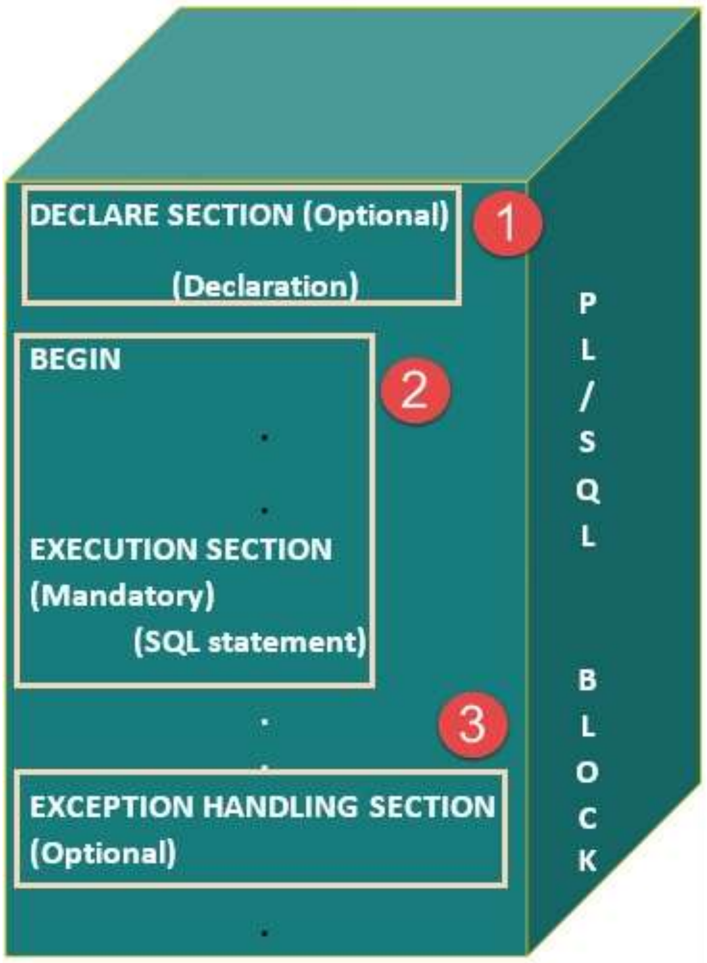
Blocks contain both PL/SQL as well as SQL instruction. All these instruction will be executed as a whole rather than executing a single instruction at a time.

Block Structure

PL/SQL blocks have a pre-defined structure in which the code is to be grouped. Below are different sections of PL/SQL blocks.

1. Declaration section
2. Execution section
3. Exception-Handling section

The below picture illustrates the different PL/SQL block and their section order.



Declaration Section

This is the first section of the PL/SQL blocks. This section is an optional part. This is the section in which the declaration of variables, cursors, exceptions, subprograms, pragma instructions and collections that are needed in the block will be declared. Below are few more characteristics of this part.

- This particular section is optional and can be skipped if no declarations are needed.
- This should be the first section in a PL/SQL block, if present. This
- section starts with the keyword ‘DECLARE’ for triggers and anonymous block. For other subprograms, this keyword will not be present. Instead, the part after the subprogram name definition marks the declaration section.
- This section should always be followed by execution section.

Execution Section

Execution part is the main and mandatory part which actually executes the code that is written inside it. Since the PL/SQL expects the executable statements from this block this cannot be an empty block, i.e., it should have at least one valid executable code line in it. Below are few more characteristics of this part.

- This can contain both PL/SQL code and SQL code.
- This can contain one or many blocks inside it as a nested block. This
- section starts with the keyword 'BEGIN'.
- This section should be followed either by 'END' or Exception-Handling section (if present)

Exception-Handling Section:

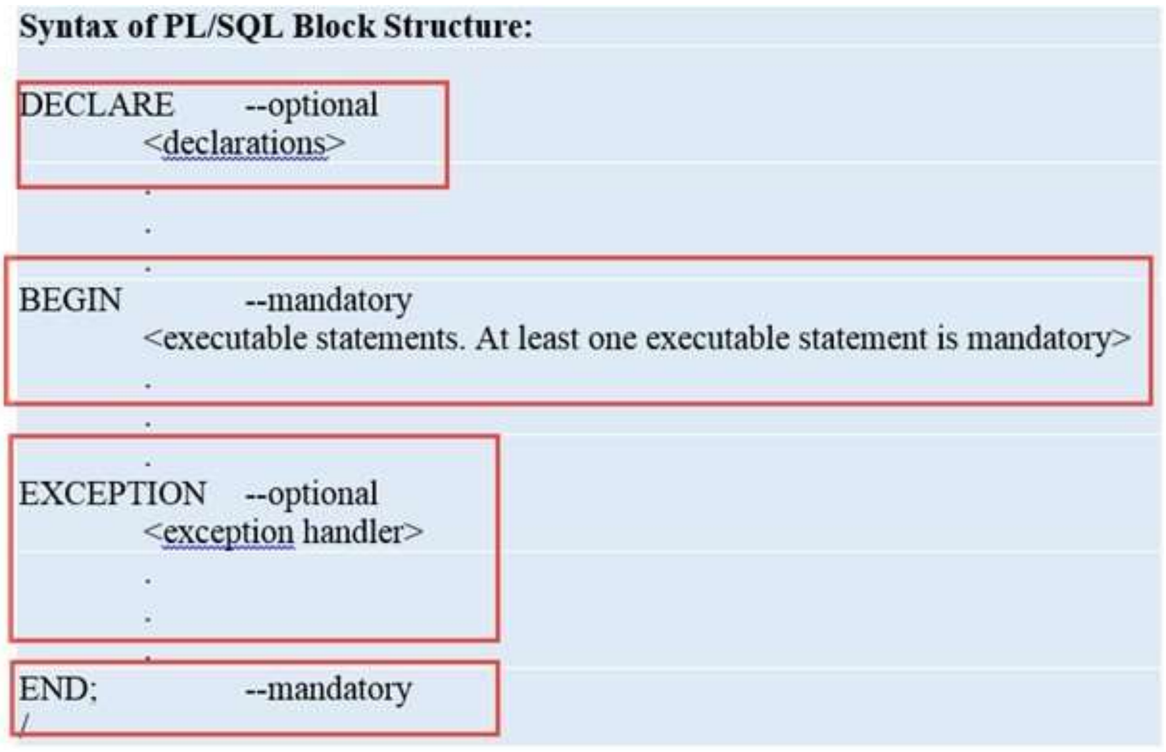
The exception is unavoidable in the program which occurs at run-time and to handle this Oracle has provided an Exception-handling section in blocks. This section can also contain PL/SQL statements. This is an optional section of the PL/SQL blocks.

- This is the section where the exception raised in the execution block is handled.
- This section is the last part of the PL/SQL block.
- Control from this section can never return to the execution block. This
- section starts with the keyword 'EXCEPTION'.
- This section should always be followed by the keyword 'END'.

The Keyword 'END' marks the end of PL/SQL block.

PL/SQL Block Syntax

Below is the syntax of the PL/SQL block structure.



Syntax of PL/SQL Block Structure:

```
DECLARE         --optional
        <declarations>
        .
        .
        .
BEGIN           --mandatory
        <executable statements. At least one executable statement is mandatory>
        .
        .
        .
EXCEPTION       --optional
        <exception handler>
        .
        .
        .
END;            --mandatory
/
```

```
DECLARE --optional
    <declarations>

BEGIN   --mandatory
    <executable statements. At least one executable statement is
mandatory>

EXCEPTION --optional
    <exception handles>

END;   --mandatory
/
```

Note: A block should always be followed by '/' which sends the information to the compiler about the end of the block.

Types of PL/SQL block

PL/SQL blocks are of mainly two types.

1. Anonymous blocks
2. Named Blocks

Anonymous blocks:

Anonymous blocks are PL/SQL blocks which do not have any names assigned to them. They need to be created and used in the same session because they will not be stored in the server as database objects.

Since they need not store in the database, they need no compilation steps. They are written and executed directly, and compilation and execution happen in a single process.

Below are few more characteristics of Anonymous blocks.

- These blocks don't have any reference name specified for them.
- These blocks start with the keyword 'DECLARE' or 'BEGIN'.
- Since these blocks do not have any reference name, these cannot be stored for later purpose. They shall be created and executed in the same session.
- They can call the other named blocks, but call to anonymous block is not possible as it is not having any reference.
- It can have nested block in it which can be named or anonymous. It can also be nested in any blocks.
- These blocks can have all three sections of the block, in which execution section is mandatory, the other two sections are optional.

Named blocks:

Named blocks have a specific and unique name for them. They are stored as the database objects in the server. Since they are available as database objects, they can be referred to or used as long as it is present on the server. The compilation process for named blocks happens separately while creating them as a database objects.

Below are few more characteristics of Named blocks.

- These blocks can be called from other blocks.
- The block structure is same as an anonymous block, except it will never start with the keyword 'DECLARE'. Instead, it will start with

the keyword ‘CREATE’ which instruct the compiler to create it as a database object.

- These blocks can be nested within other blocks. It can also contain nested blocks.
- Named blocks are basically of two types:

1. Procedure
2. Function

We will learn more about these named blocks in “Procedure” and “Function” topics in later tutorial.

Summary

After this tutorial, you should be aware of PL/SQL blocks and its types, different sections of blocks and their usages. The detailed description of the named PL/SQL blocks will be covered in the later tutorial.