# W02 Study: Materials

## Overview

Your assigned reading will be as follows:

* Chapter 3 (pgs. 44–110)

### **Reading Notes**

**Make sure you go over the following topics.**

**Block Structure** (pg. 44)

PL/SQL program units organize the code into blocks. A block without a name is known as an **anonymous block**. The anonymous block is the simplest unit in PL/SQL. On the other hand, **named blocks** are called **functions** and **procedures**.



**Declaration Block** (pg. 48)

It is optional and starts with the **DECLARE** keyword and ends with the **BEGIN** keyword for anonymous blocks. The declaration section allows you to define data types, structures, and variables. You often declare variables in the declaration section by giving them names, data types, and initial values.

**Execution Block** (pg. 44)

This is a mandatory section enclosed between the keywords **BEGIN** and **END**. When using the **EXCEPTION** block, it will end with the exception keyword. The execution section is the place where you put the execution code or business logic code. You can use both procedural and SQL statements inside the execution section.

**Exception Block** (pg. 49)

It is optional and starts with the **EXCEPTION** keyword and ends with the **END** keyword. The exception section is the place where you put the code to handle exceptions. You can either catch or handle exceptions in the exception section.

**Behavior of Variables in Blocks** (pg. 50)

Please review this section. It will be covered in more details in Chapter 4.

**Nested Anonymous Blocks** (pg. 55)

Make sure you understand the examples. The following is an example of nested blocks. Do not worry if you don’t understand the program, just look at the declare, begin, exception, and end keywords.



**Basic Scalar and Composite Data Types** (p. 63)

Covered in more detail in Chapter 4.

**Scalar** (pg. 63)

They hold only one thing at the time and are frequently labeled as primitives; these include numbers, strings, and timestamps.

**Composite** (pg. 68)

They hold collections of data.

**SQL Collections** (pg. 72)

Collections are programming structures that hold sets of variables; they fall into two categories: arrays and lists. Oracle8i Database forward provides three types of collections. Covered in more detail in Chapter 6.

1. **Associative arrays** are collection types that associates a unique key with a value of the index.
2. **Nested** **table** is one table placed inside of another.
3. **Varray** are fixed-size sequential collection of elements of the same type.

**Conditional Structures** (pg. 81)

Review these topics.

1. IF-THEN-ELSIF, and ELSE Statements (pg. 81)
2. Three-Valued Logic (pg. 81)
3. CASE Statement (pg. 82)
4. Iterative Structures (pg. 83)
5. For Loop Statements (pg. 84)
6. While Loop Statements (pg. 87)

**Exceptions** (pg. 92)

The exception block manages any exceptions that occur while running the execution block. Errors raised in the declaration block are thrown to and managed by the calling scope program.

Make sure to review these topics and their implementation.

1. **User-Defined Exceptions** (pg. 93)
2. **Dynamic User-Defined Exceptions** (pg. 94)

**Functions, Procedures, and Packages** (pg. 97)

PL/SQL stored programming units are typically functions, procedures, packages, and triggers. This functionality enables commonly required code to be written and tested once and then accessed by any application that requires the code. Covered in more detail in chapters 8 and 9.

1. **Functions** (pg. 97)
2. **Procedures** (pg. 99)
3. **Packages** (pg. 100)

**Database Triggers** (pg. 108)

Triggers are similar to stored procedures that run implicitly when an INSERT, UPDATE or DELETE statement is issued against the associated table or, in some cases, against a view or when database system actions occur. This topic is covered in more detail in Chapter 12.