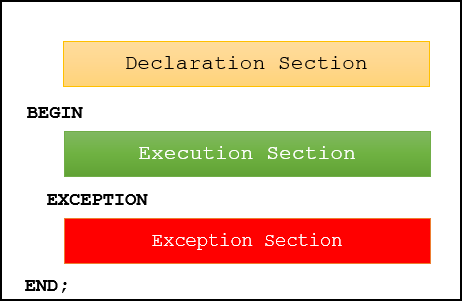
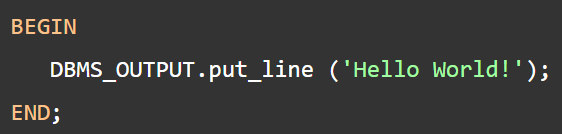
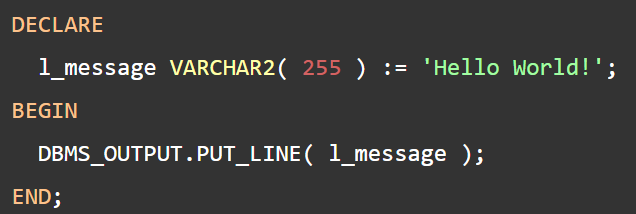
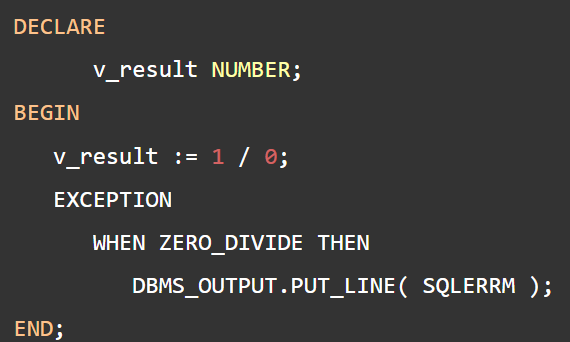
1. Introduction
   1. PL/SQL stands for “Procedural Language extensions to the Structured Query Language”.
2. PL/SQL anonymous block
   1. PL/SQL is a block-structured language whose code is organized into blocks.
   2. A PL/SQL block consists of three sections: declaration, executable, and exception-handling sections.
   3. In a block, the executable section is mandatory while the declaration and exception-handling sections are optional.
   4. A PL/SQL block has a name**. Functions or Procedures** is an example of a named block. A named block is stored into the Oracle Database server and can be reused later.
   5. A block without a name is **an anonymous block**. An anonymous block is not saved in the Oracle Database server, so it is just for one-time use. However, PL/SQL anonymous blocks can be useful for testing purposes.



* 1. Example



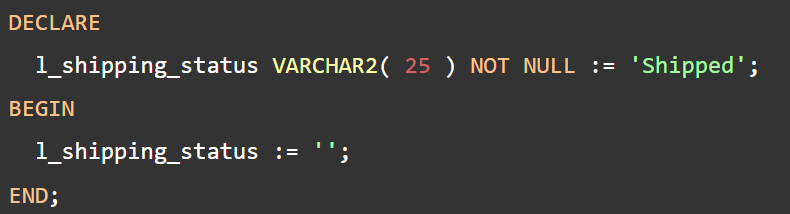




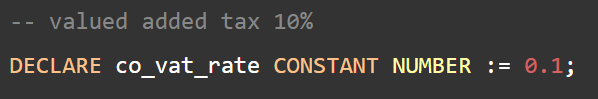
1. PL/SQL data types
   1. Each value in PL/SQL such as a constant, variable and parameter has a data type that determines the storage format, valid values, and allowed operations.
   2. PL/SQL has two kinds of data types: **scalar** and **composite**. The scalar types are types that store single values such as number, Boolean, character, and datetime whereas the composite types are types that store multiple values, for example, record and collection.
   3. PL/SQL divides the scalar data types into four families:
      1. Number
         1. The numeric data types represent real numbers, integers, and floating-point numbers.
         2. They are stored as NUMBER, IEEE floating-point storage types (BINARY\_FLOAT and BINARY\_DOUBLE), and PLS\_INTEGER.
      2. Boolean
         1. The BOOLEAN datatype has three data values: TRUE, FALSE, and NULL.
      3. Character
         1. CHAR(n) is a fixed-length character type whose length is from 1 to 32,767 bytes.
         2. VARCHAR2(n) is varying length character data from 1 to 32,767 bytes.
      4. Datetime
         1. The datetime data types represent dates, timestamp with or without time zone and intervals. PL/SQL datetime data types are DATE, TIMESTAMP.
2. PL/SQL Variables
   1. In PL/SQL, a variable is named storage location that stores a value of a particular data type.
   2. Syntax:



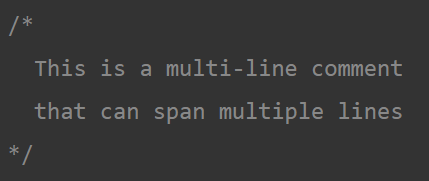
* 1. Example:



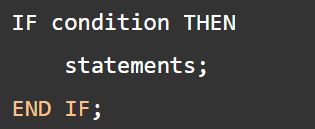
1. PL/SQL Comments
   1. PL/SQL comments allow you to describe the purpose of a line or a block of PL/SQL code.
   2. Single-line comments
      1. A single-line comment starts with a double hyphen ( --) that can appear anywhere on a line and extends to the end of the line.
      2. Example



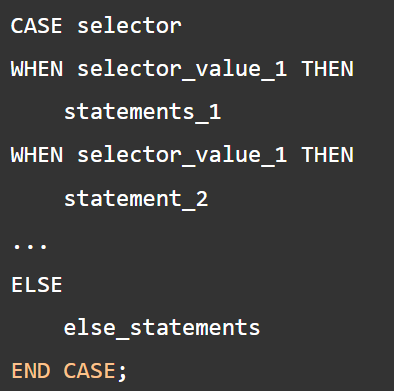
* 1. Multi-line comments
     1. A multi-line comment starts with a slash-asterisk ( /\* ) and ends with an asterisk-slash ( \*/ ), and can span multiple lines:



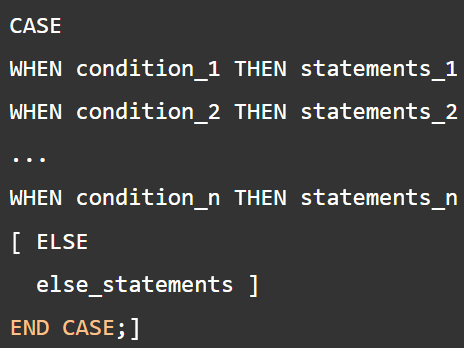
1. PL/SQL constants
   1. Unlike a variable, a constant holds a value that does not change throughout the execution of the program.
2. PL/SQL IF Statement
   1. The IF statement allows you to either execute or skip a sequence of statements, depending on a condition. The IF statement has the three forms:
      1. IF THEN
      2. IF THEN ELSE
      3. IF THEN ELSEIF



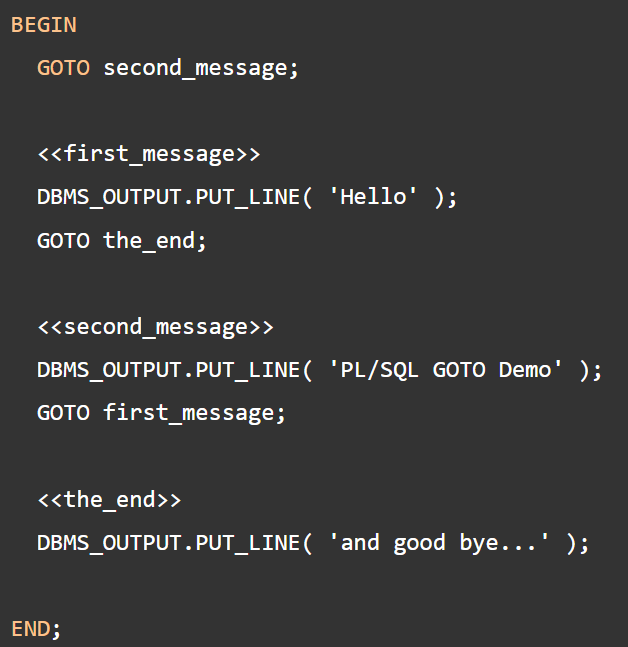
1. PL/SQL CASE Statement
   1. The CASE statement chooses one sequence of statements to execute out of many possible sequences.
   2. The CASE statement has two types: simple CASE statement and searched CASE statement.
   3. Simple CASE statement
      1. A simple CASE statement evaluates a single expression and compares the result with some values.



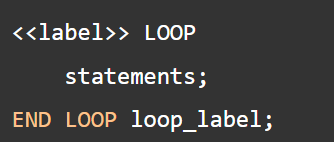
* 1. Searched CASE statement
     1. The searched CASE statement evaluates multiple Boolean expressions and executes the sequence of statements associated with the first condition that evaluates to TRUE.

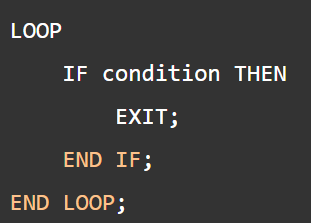


1. PL/SQL GOTO Statement
   1. The GOTO statement allows you to transfer control to a labeled block or statement.
   2. you cannot use a GOTO statement to transfer control into an IF, CASE or LOOP statement, the same for sub-block.

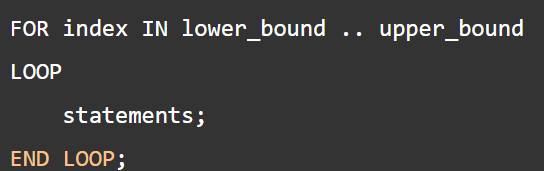


1. PL/SQL NULL Statement
   1. The NULL statement is a NULL keyword followed by a semicolon ( ;). The NULL statement does nothing except that it passes control to the next statement.
2. PL/SQL LOOP

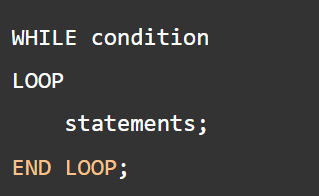




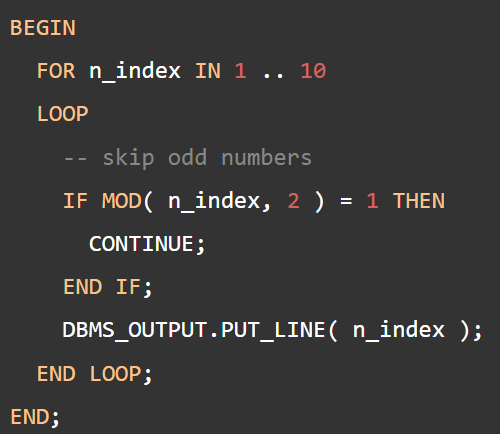
1. PL/SQL FOR LOOP
   1. PL/SQL FOR LOOP executes a sequence of statements a specified number of times.



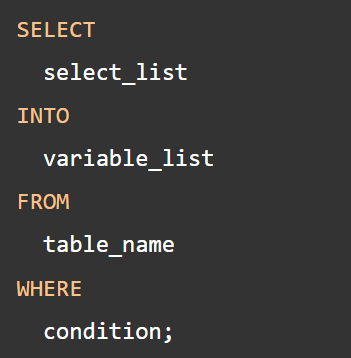
1. PL/SQL WHILE Loop



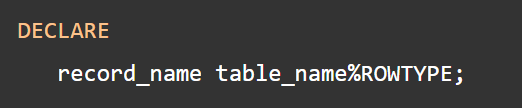
1. PL/SQL CONTINUE
   1. The CONTINUE statement allows you to exit the current loop iteration and immediately continue on to the next iteration of that loop.



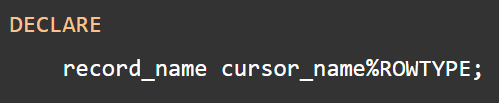
1. PL/SQL SELECT INTO
   1. PL/SQL SELECT INTO statement is the simplest and fastest way to fetch a single row from a table into variables.



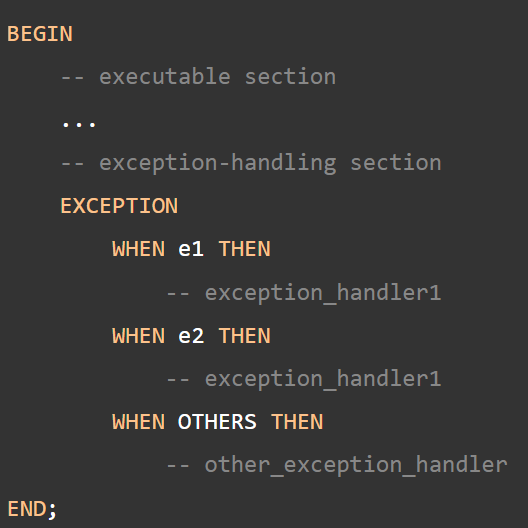
1. PL/SQL Record
   1. A PL/SQL record is a composite data structure which consists of multiple fields; each has its own value.
   2. Table-based record



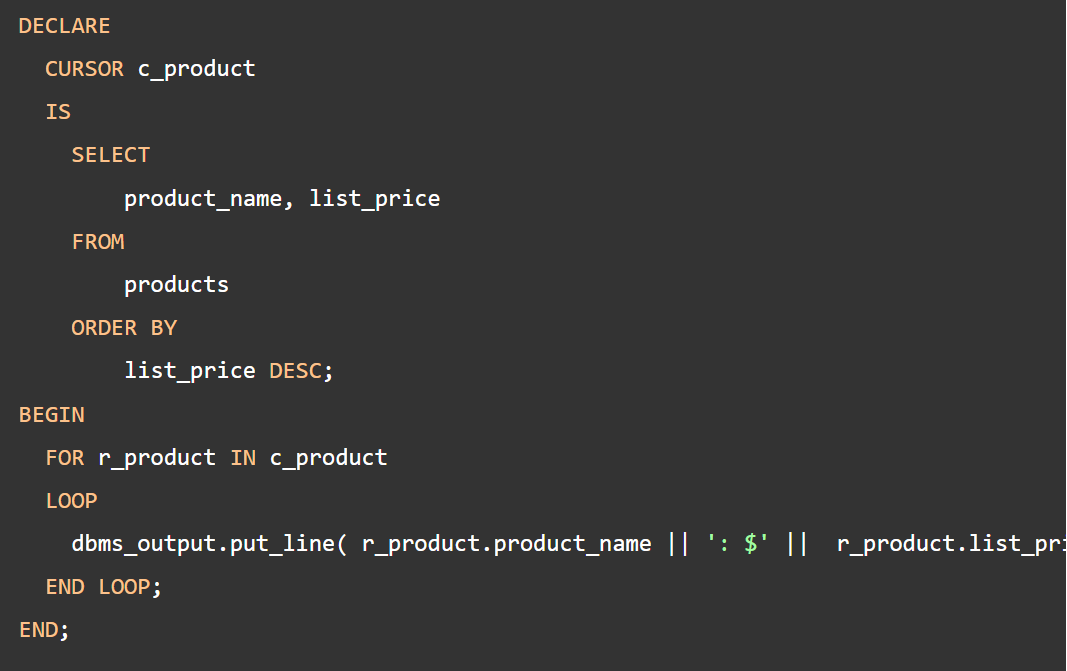
* 1. Cursor-based record



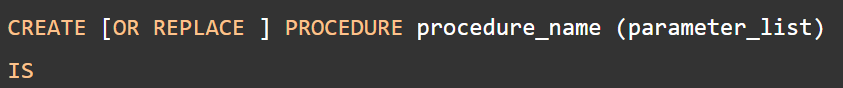
1. PL/SQL Exception
   1. PL/SQL treats all errors that occur in an anonymous block, procedure, or function as exceptions. The exceptions can have different causes such as coding mistakes, bugs, even hardware failures.



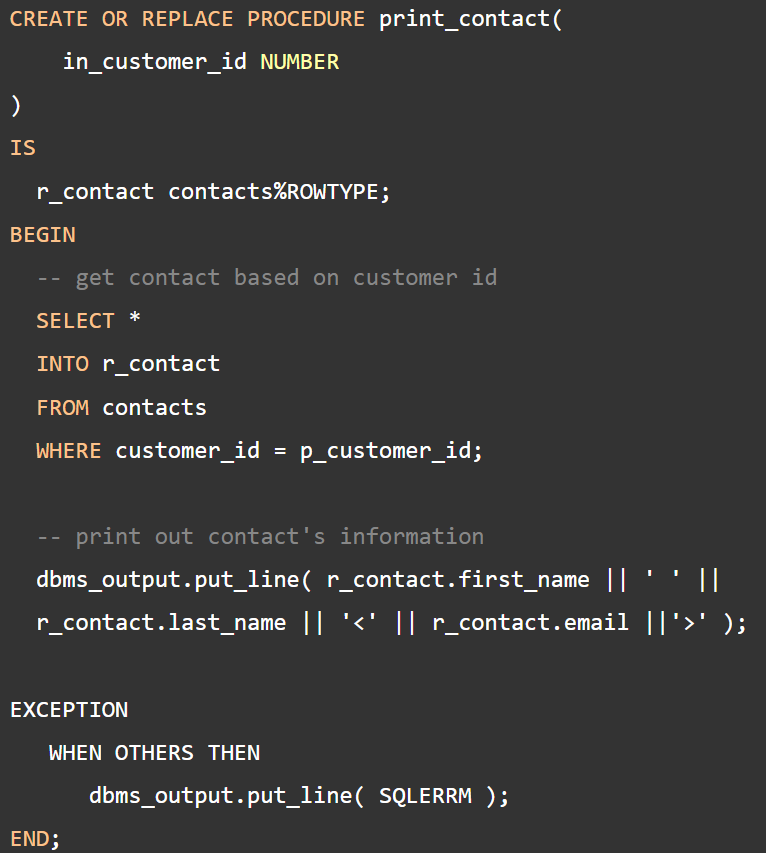
1. PL/SQL Raise Exceptions
   1. To raise an exception explicitly, you use the RAISE statement. The RAISE statement allows you to:
      1. Raise a user-defined exception.
      2. Raise an internally defined exception.
      3. Reraising the current exception.
2. PL/SQL Cursor
   1. A cursor is a pointer that points to a result of a query.
   2. PL/SQL has two types of cursors: implicit cursors and explicit cursors.
   3. Implicit cursors
      1. Whenever Oracle executes an SQL statement such as SELECT INTO, INSERT, UPDATE, and DELETE, it automatically creates an implicit cursor.
      2. Oracle internally manages the whole execution cycle of implicit cursors and reveals only the cursor’s information and statuses such as SQL%ROWCOUNT, SQL%ISOPEN, SQL%FOUND, and SQL%NOTFOUND.
   4. Explicit cursors
      1. An explicit cursor is an SELECT statement declared explicitly in the declaration section of the current block or a package specification.
      2. For an explicit cursor, you have control over its execution cycle from OPEN, FETCH, and CLOSE.
      3. Oracle defines an execution cycle that executes an SQL statement and associates a cursor with it.



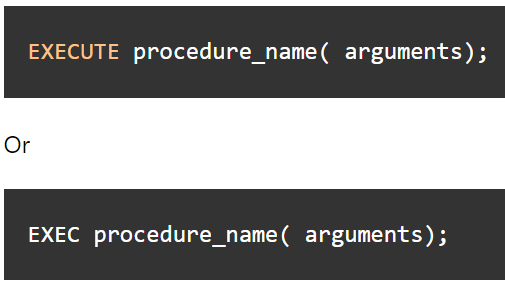
1. PL/SQL Procedure
   1. A PL/SQL procedure is a reusable unit that encapsulates specific business logic of the application.
   2. Technically speaking, a PL/SQL procedure is a named block stored as a schema object in the Oracle Database.



* 1. PL/SQL procedure header
     1. Each parameter can be in either IN, OUT, or INOUT mode. The parameter mode specifies whether a parameter can be read from or write to.
     2. An IN parameter is read-only. You can reference an IN parameter inside a procedure, but you cannot change its value. Oracle uses IN as the default mode. It means that if you don’t specify the mode for a parameter explicitly, Oracle will use the IN mode.
     3. An OUT parameter is writable. Typically, you set a returned value for the OUT parameter and return it to the calling program. Note that a procedure ignores the value that you supply for an OUT parameter.
     4. An INOUT parameter is both readable and writable. The procedure can read and modify it.
  2. PL/SQL procedure body
     1. Declarative part
     2. Executable part
     3. Exception-handling part



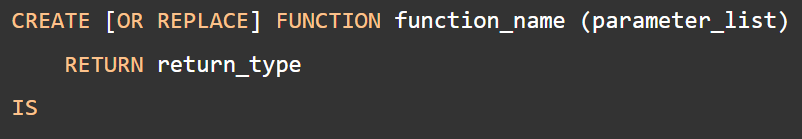
* 1. Executing a PL/SQL procedure



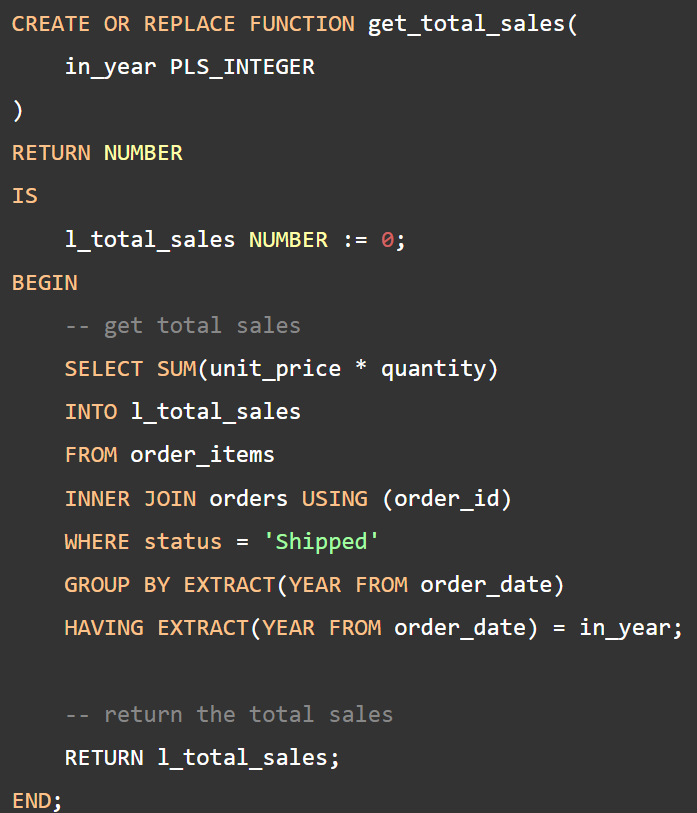
* 1. Removing a procedure



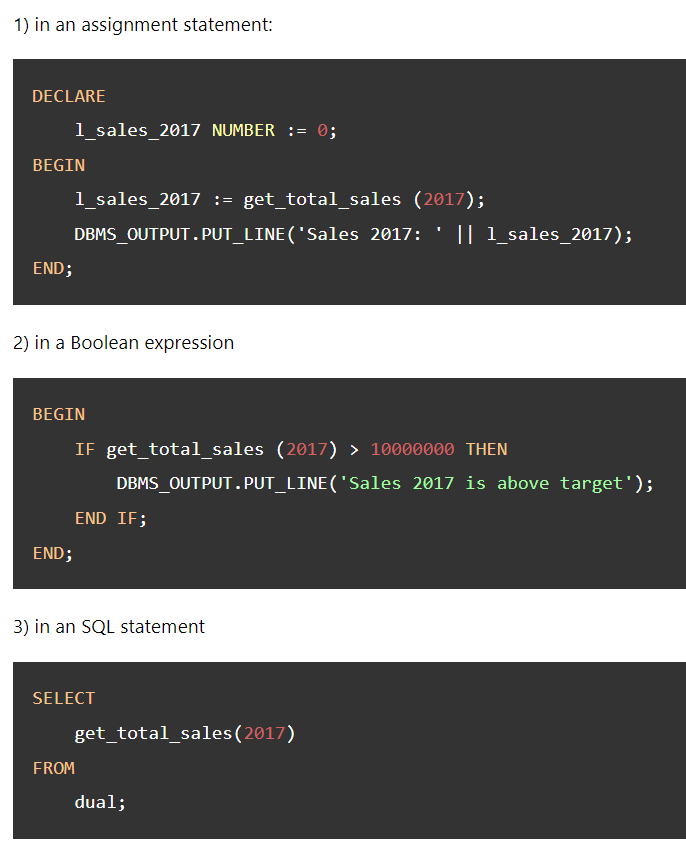
1. PL/SQL Function
   1. Creating a PL/SQL function
      1. A PL/SQL function is a reusable program unit stored as a schema object in the Oracle Database.



* 1. Example:



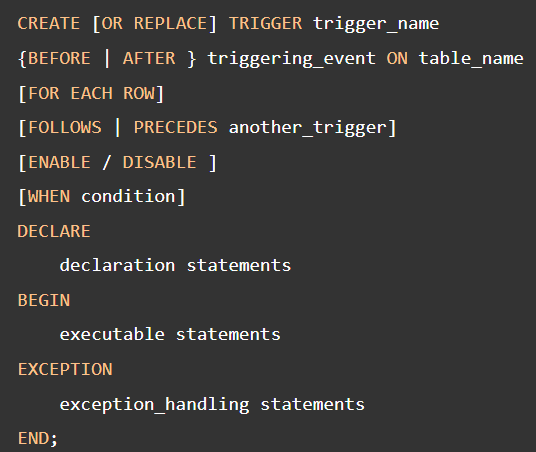
* 1. Calling a PL/SQL function
     1. You use a function anywhere that you use an expression of the same type.



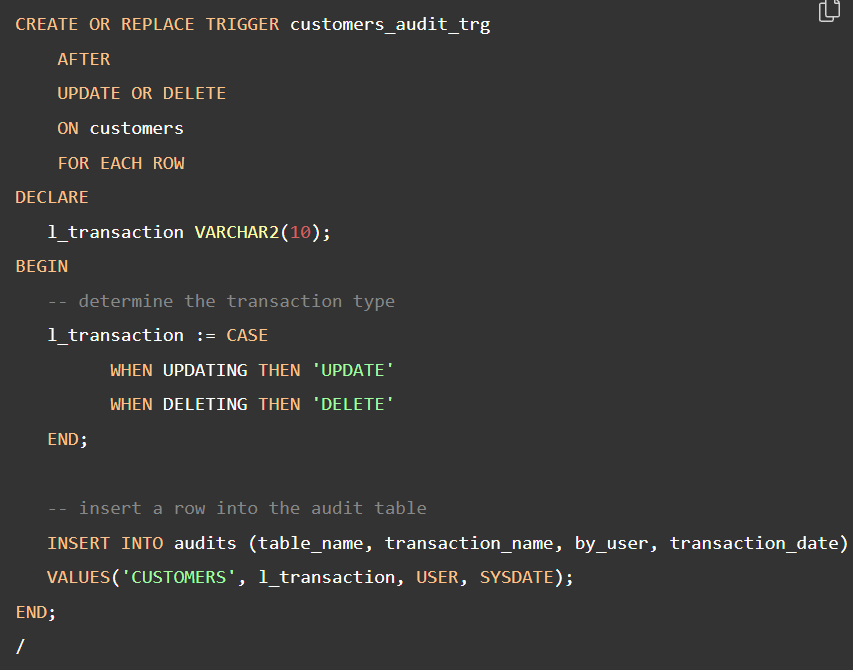
* 1. Removing a function
     1. The DROP FUNCTION deletes a function from the Oracle Database



1. PL/SQL Package
   1. In PL/SQL, a package is a schema object that contains definitions for a group of related functionalities.
   2. A package includes variables, constants, cursors, exceptions, procedures, functions, and subprograms. It is compiled and stored in the Oracle Database.
2. Oracle Trigger
   1. A trigger is a named PL/SQL block stored in the Oracle Database and executed automatically when a triggering event takes place.
   2. The event can be any of the following:
      1. A data manipulation language (DML) statement executed against a table e.g., INSERT, UPDATE, or DELETE. For example, if you define a trigger that fires before an INSERT statement on the customers table, the trigger will fire once before a new row is inserted into the customers table.
      2. A data definition language (DDL) statement executes e.g., CREATE or ALTER statement. These triggers are often used for auditing purposes to record changes of the schema.
      3. A system event such as startup or shutdown of the Oracle Database.
      4. A user event such as login or logout.
   3. How to create a trigger in Oracle



* 1. Example:



* 1. Oracle Disable Triggers



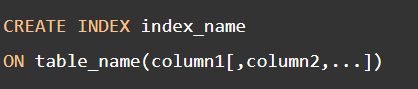
* 1. Disable all triggers of a table



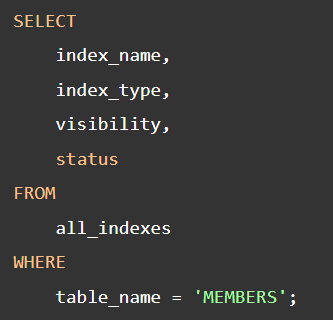
* 1. Oracle DROP TRIGGER



1. Oracle Index
   1. Oracle index is one of the effective tools for boost the query performance.
   2. Oracle Create Index



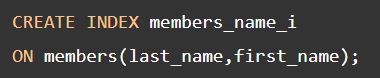
* 1. Check all index of table



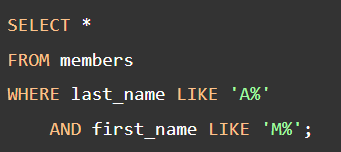
* 1. Removing an index



* 1. Creating an index on multiple columns



* 1. Query automatically uses index for querying



* 1. Oracle DROP INDEX



1. Important Query