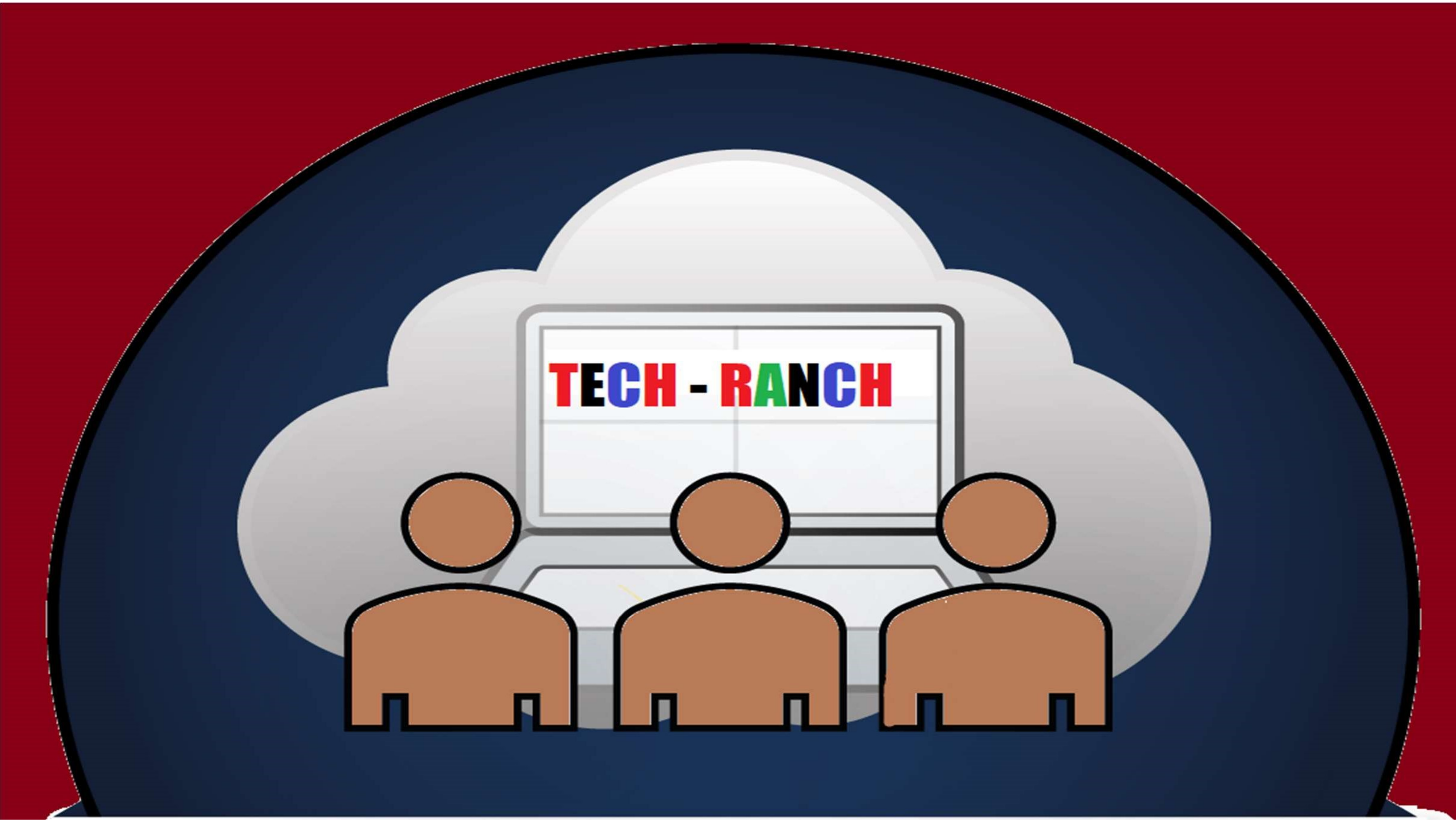

TECH - RANCH

PL/SQL Tutorial

(Procedural - Structured Query Language)

By

Anjali singh

Let’s make coding fun!

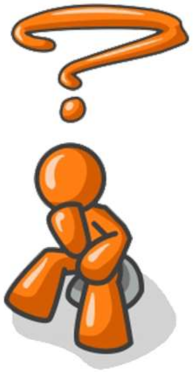
Chapter 6

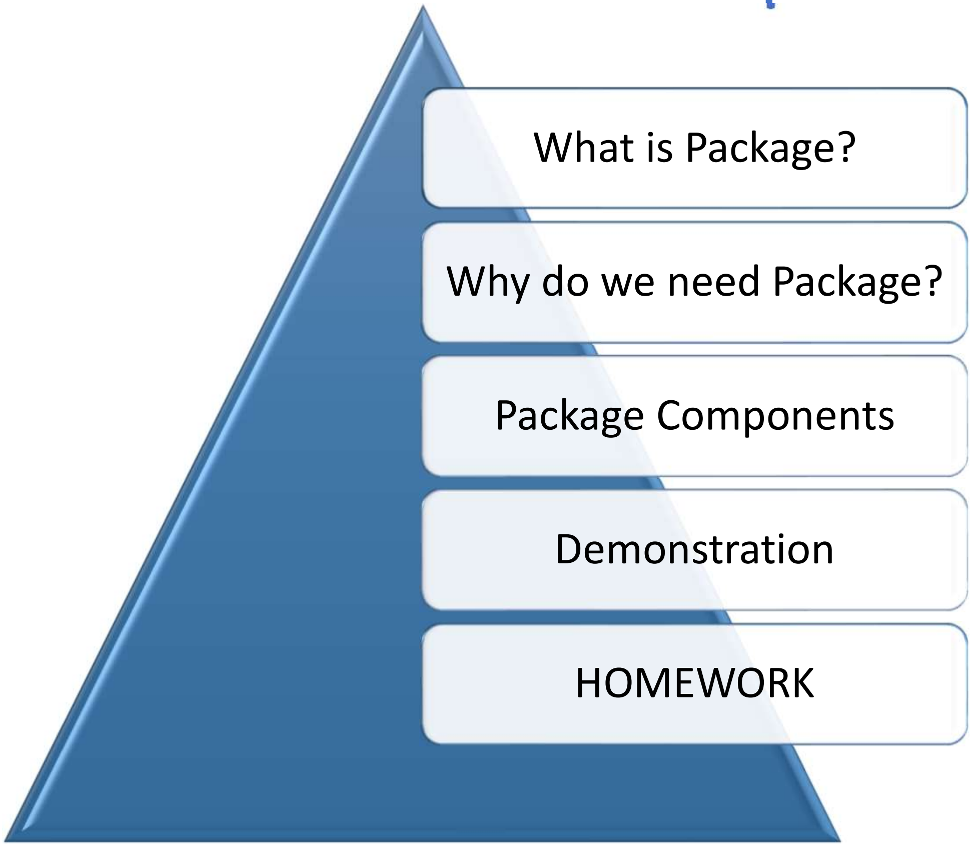
PL/SQL Tutorial

Packages & Demonstration

Let’s make coding fun!

What do we learn in this Chapter?



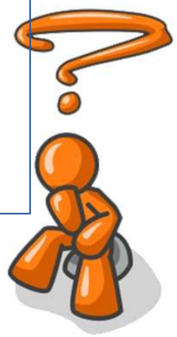


- What is Package?
- Why do we need Package?
- Package Components
- Demonstration
- HOMEWORK

What is Package?

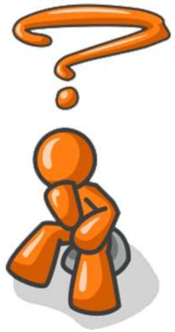
- Generic
- Encapsulated
- Re-usable code
- Standalone sub program

A package is an Oracle object, which holds other objects within it. Object commonly help within a package are procedure, functions, variables, constants, cursors and exceptions.

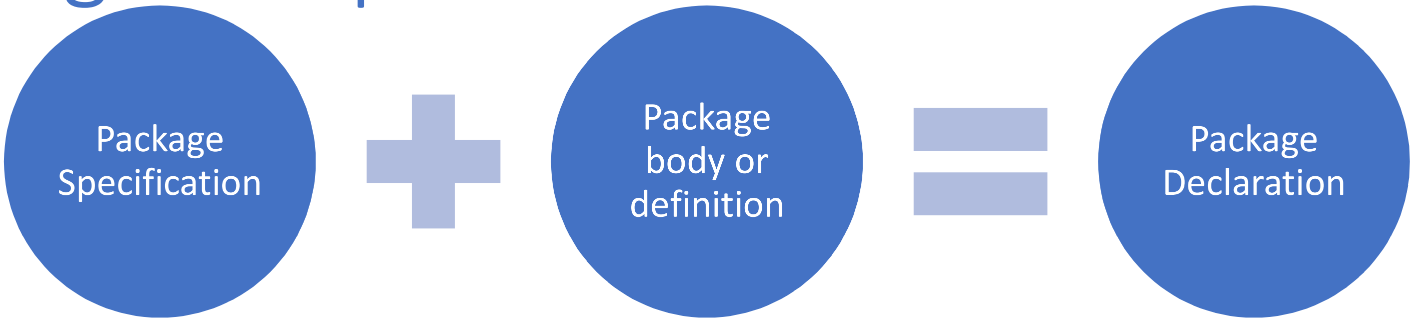


Why do we need Package?

1. Granting privileges efficiently
2. Enable overloading procedure and functions
3. Enables organization of commercial applications into efficient modules. Improve performance by loading multiple objects into memory at once. Therefore, subsequent calls to related subprograms in package require no Input/Output
4. Promote code reuse through the use of libraries that contain stored procedures and function, thereby reducing redundant coding



Package Components

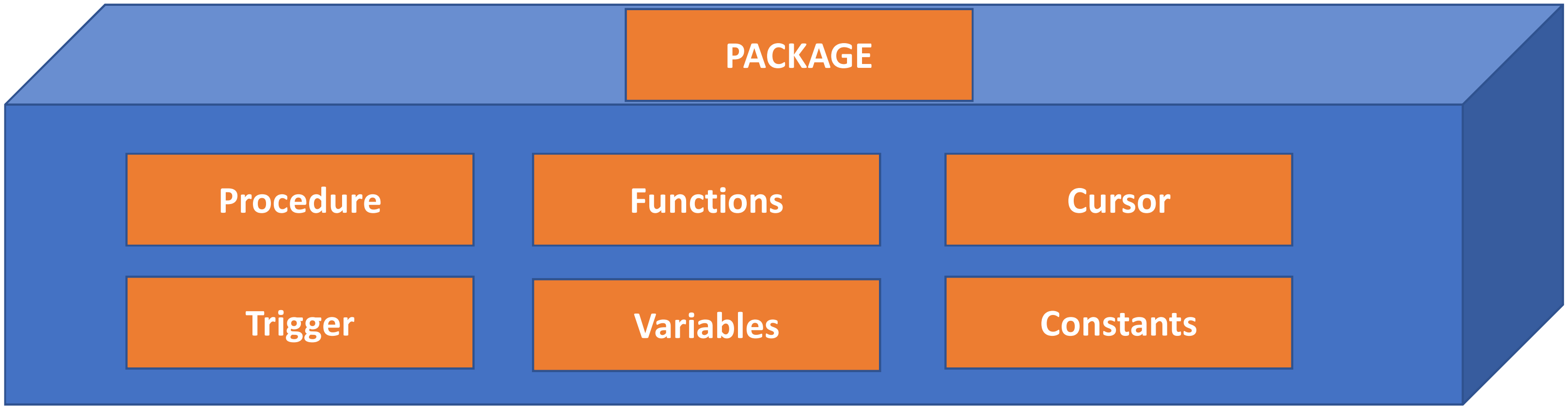

Package Specification

+

Package body or definition

=

Package Declaration


PACKAGE

Procedure

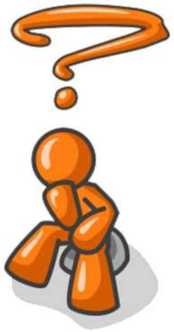
Functions

Cursor

Trigger

Variables

Constants



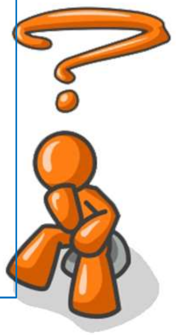
Package Components

Package Specification

Package body or definition

The specification is the interface to the package. It just **DECLARES** the types, variables, constants, exceptions, cursors, and subprograms that can be referenced from outside the package. In other words, it contains all information about the content of the package, but excludes the code for the subprograms.
All objects placed in the specification are called **public** objects. Any subprogram not in the package specification but coded in the package body is called a **private** object.

The package body has the codes for various methods declared in the package specification and other private declarations, which are hidden from the code outside the package.



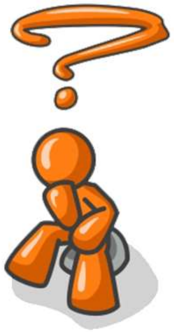
Package Syntax

PACKAGE SPECIFICATION

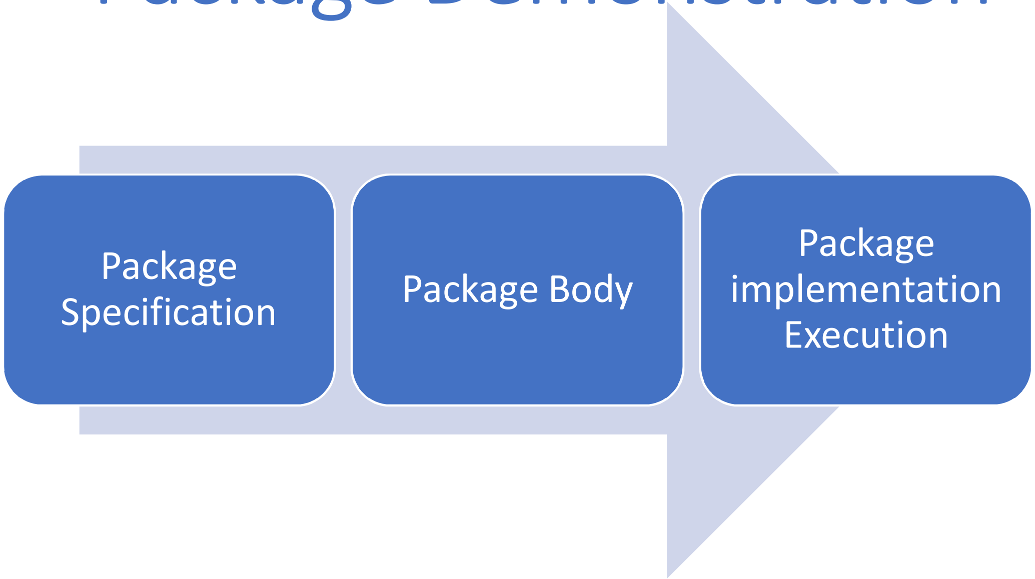
CREATE PACKAGE <Package_name> AS
<declaration of database object>
PROCEDURE <Procedure_name_Def>;
FUNCTION <function_name_Def>;
VARIABLES;
END PACKAGE_NAME;

PACKAGE BODY

CREATE OR REPLACE PACKAGE BODY
<package_name> AS
PROCEDURE Procedure_name_def IS
BEGIN
SQL STATEMENT;
PL/SQL STATEMENTS;
END Procedure_name;
END Package_name;



Package Demonstration



Package Specification

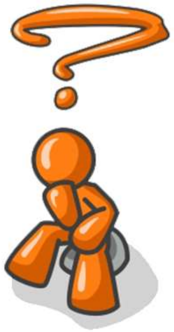
Package Body

Package implementation Execution

PACKAGE_EXAMPLE
PROCEDURE ADD_VALUES
PROCEDURE DELETE_VALUES

TABLEA

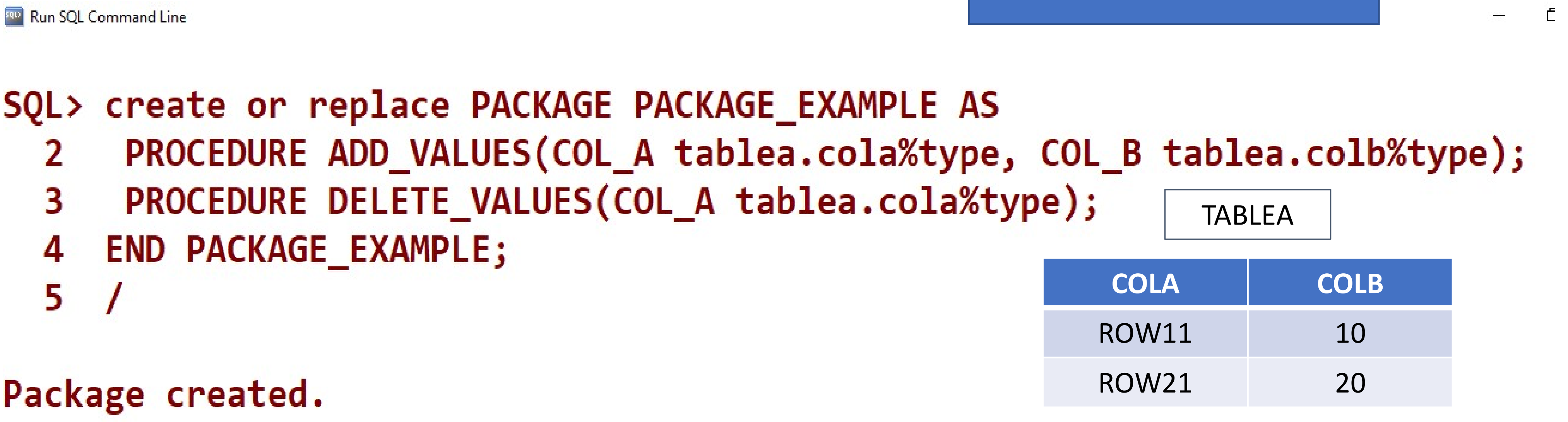
COLA	COLB
ROW11	10
ROW21	20



Package Demonstration

Package Specification

PACKAGE_EXAMPLE
PROCEDURE ADD_VALUES
PROCEDURE DELETE_VALUES

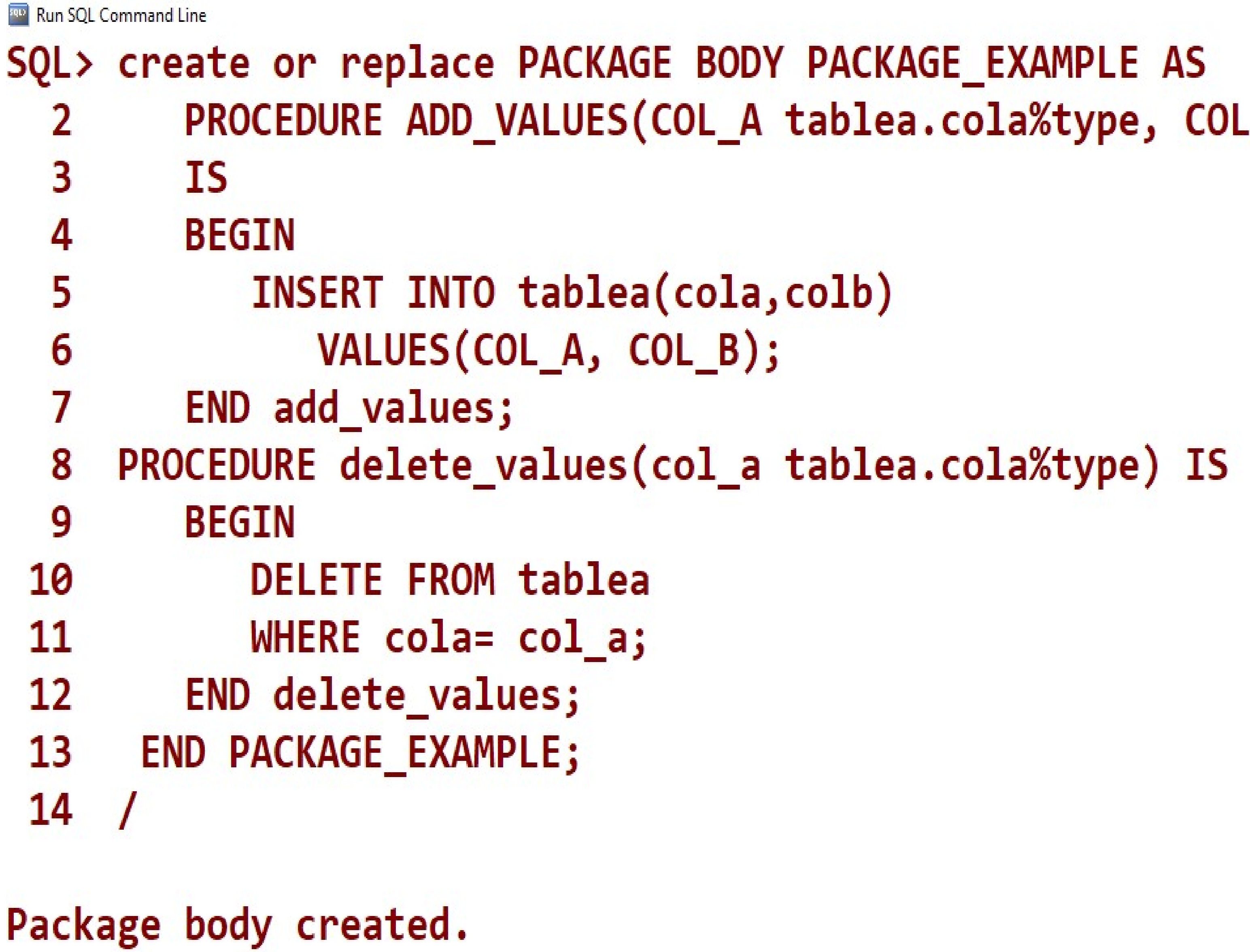


```
SQL> create or replace PACKAGE PACKAGE_EXAMPLE AS
  2   PROCEDURE ADD_VALUES(COL_A tablea.cola%type, COL_B tablea.colb%type);
  3   PROCEDURE DELETE_VALUES(COL_A tablea.cola%type);
  4  END PACKAGE_EXAMPLE;
  5  /

Package created.
```

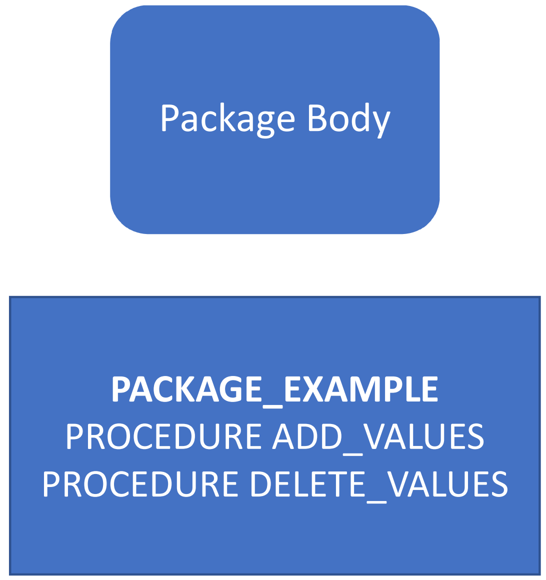
TABLEA

COLA	COLB
ROW11	10
ROW21	20



```
SQL> create or replace PACKAGE BODY PACKAGE_EXAMPLE AS
       PROCEDURE ADD_VALUES(COL_A tablea.cola%type, COL_B tablea.colb%type)
       IS
       BEGIN
          INSERT INTO tablea(cola,colb)
             VALUES(COL_A, COL_B);
       END add_values;
     PROCEDURE delete_values(col_a tablea.cola%type) IS
       BEGIN
          DELETE FROM tablea
          WHERE cola= col_a;
       END delete_values;
      END PACKAGE_EXAMPLE;
     /

Package body created.
```

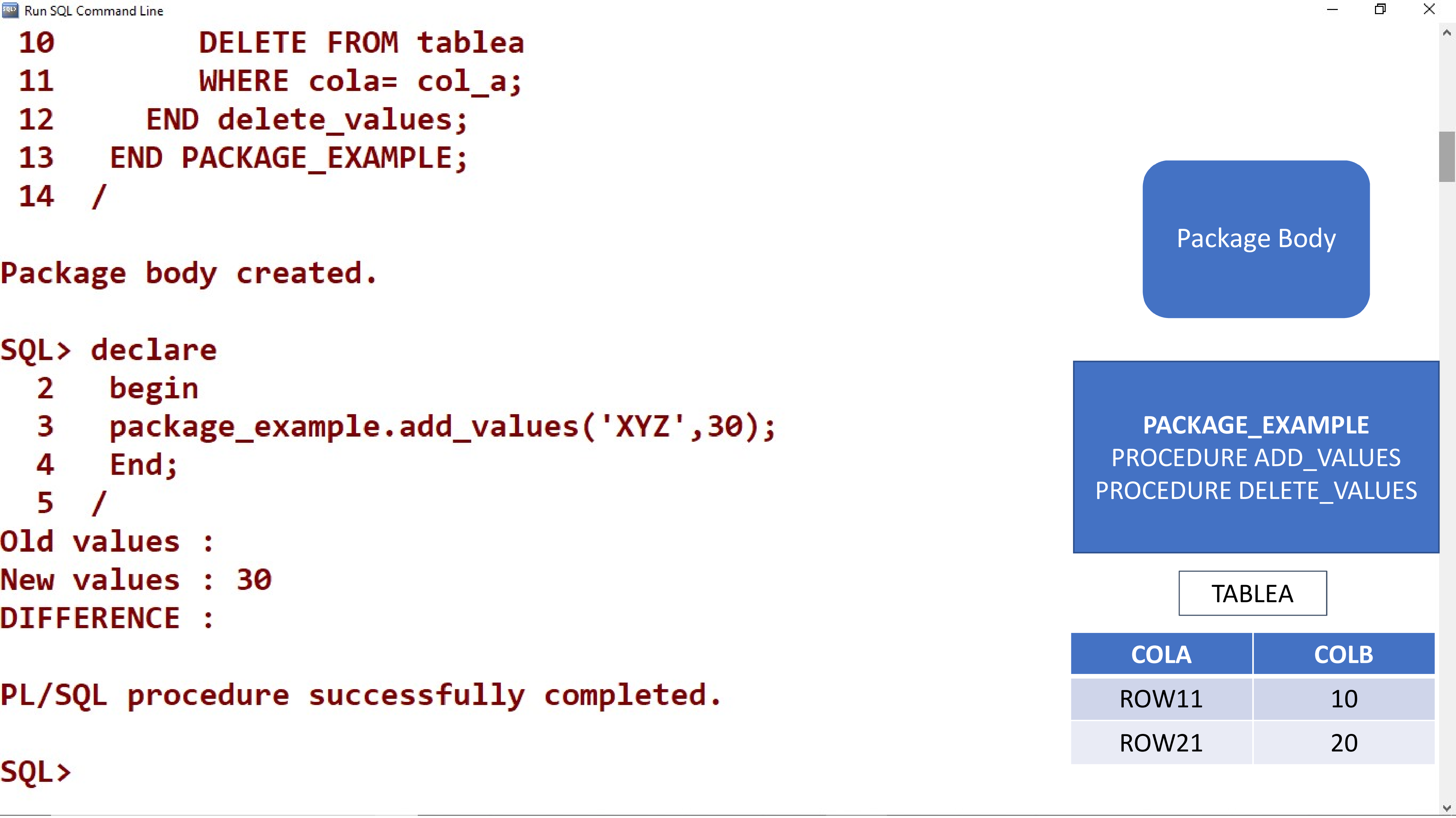


Package Body

PACKAGE_EXAMPLE
PROCEDURE ADD_VALUES
PROCEDURE DELETE_VALUES

TABLEA

COLA	COLB
ROW11	10
ROW21	20



```
      DELETE FROM tablea
      WHERE cola= col_a;
   END delete_values;
 END PACKAGE_EXAMPLE;
/

Package body created.

SQL> declare
 begin
 package_example.add_values('XYZ',30);
 End;
/
Old values :
New values : 30
DIFFERENCE :

PL/SQL procedure successfully completed.

SQL>
```

Package Body

PACKAGE_EXAMPLE
PROCEDURE ADD_VALUES
PROCEDURE DELETE_VALUES

TABLEA

COLA	COLB
ROW11	10
ROW21	20

IMPORTANT FACTS

PL/SQL code outside the package reference objects from within the package using dot notation , declared in package specification called PUBLIC

Objects that are declared inside the package body, their use is restricted to within that package only called PRIVATE

Multiple procedures that are declared with the same name are called Overloaded Procedures.

Multiple functions that are declared with the same name are called Overloaded Functions

Practice Project

Online Banking System

Account_No	Acct_holder	Balance	Eligibility
ACCT001	Alex	120000	NO
ACCT002	Mark	340000	YES
ACCT003	John	245000	YES
ACCT005	Felix	150000	NO

HOME WORK!

Instructions

1. Create a package and define function which helps to perform update on credit eligibility condition when balance is more than 220000 for an account holder

For more resource , assistance and questions , please contact techranch2019@gmail.com

Let's make coding fun!

Conclusion

PL/SQL

Stored Procedure

Functions

Trigger

Cursors

Package Demonstration

Interview & Certifications Tips

Let’s make coding fun!

SUBSCRIBE

Let’s make coding fun!

SUBSCRIBE