

**Government College of Engineering, Jalgaon**  
**(An Autonomous Institute of Government of Maharashtra)**

<b>Name :</b> Pratiksha Rajendra Badgujar	<b>Semester :</b> V	<b>PRN :</b> 2241003
<b>Class :</b> T. Y. B.Tech Computer	<b>Academic Year :</b> 2024-25	<b>Subject :</b> CO307U DBMS Lab
<b>Course Teacher : Mr. Vinit Kakde</b>		
<b>Date of Performance :</b>	<b>Date of Completion :</b>	

**Practical no. 10**

**Aim:**

- a) Write a PL/SQL block to find the maximum number from given three numbers.
- b) Write a PL/SQL program for swapping 2 numbers.
- c) Calculate the net salary and year salary if da is 30% of basic, bra is 10% of basic and pf is 7% if basic salary is less than 8000, pris 10% if basic sal between 8000 to 160000.

**Theory:**

**What is PL/SQL in DBMS?**

PL/SQL is a procedural language extension to the Structured Query Language. It is popular for building database-driven applications. Furthermore, it is a programming language that creates and manages tables, views, procedures, functions, packages, and triggers. It can only be used when you want to execute it in an Oracle database.

Now as we have a clear understanding of the meaning of PL SQL. Let's dig more into this database.

**PL SQL Architecture**

The PL/SQL architecture comprises three main components. These are:

- **PL/SQL Block:** It is the primary input that has the PL/SQL code. It has three different sections to logically divide the code:
  - The declarative section declares the purpose.
  - The execution section processes statements.
  - The exception handling section handles errors.

All the PL/SQL units are considered PL/SQL blocks. Some types of units are function, library, package specification, type, anonymous block, etc. These contain SQL instructions that help in interacting with the database server.

- **PL/SQL Engine:** It is mainly responsible for executing PL/SQL code on the server. This engine separates PL/SQL units and SQL parts in the input and handles the PL/SQL units. The SQL part is sent to the database server.
- **Database Server:** It is an essential component of the PL/SQL architecture that stores the data and has the SQL executor to parse and execute input SQL statements. PL/SQL engine uses the SQL from PL/SQL units to interact with the database server.

### Advantages of PL/SQL

The following are a few advantages of using PL/SQL:

- **High Performance:** PL SQL supports bulk processing, which means it allows efficient processing of large amounts of data. Furthermore, it stores in a compiled form in the database, which makes it faster to execute than interpreted languages like SQL.
- **Portability:** PL/SQL is a good choice for cross-platform applications because it can be simply ported between various platforms and operating systems.
- **SQL Seamless Integration:** PL SQL is an extension of SQL; therefore, it seamlessly integrates with SQL to allow easy data manipulation.
- **Modular Programming:** Through the use of packages, it supports modular programming. Hence, it allows better code organization and maintenance.
- **Exception Handling:** Because of its inbuilt exception handling, it allows the detection and handling of errors in a controlled way.
- **Security:** To control access to the database and ensure data integrity, PL/SQL includes security features, such as privileges and roles.

### Features of PL/SQL

PL/SQL has several key features. Some of these features are:

- **Block Structure:** PL SQL code is structured into blocks that include declarations, executable statements, and exception handlers.
- **Data Types:** Numbers, characters, dates, Booleans, and user-defined types are among the data types supported by PL/SQL.
- **Procedural Capabilities:** PL/SQL allows you to write procedural code using constructs, such as loops, conditional statements, and exception handling.

- **Stored Procedures and Functions:** It permits the creation of reusable code in the form of stored procedures and functions.
- **Cursors:** It provides cursors that enable retrieval and row-by-row processing of data from a database.
- **Exception Handling:** It offers extensive error-handling mechanisms that enable you to detect and handle errors during program execution.
- **Packages:** It permits the grouping of related procedures, functions, and variables into a package, making code easier to manage and maintain.
- **Triggers:** The triggers allow you to automatically execute code in response to database events like insert, update, or delete operations.
- **Dynamic SQL:** It allows you to construct SQL statements dynamically at runtime. This enables you to create more flexible and potent applications.
- **Object-Oriented Features:** It allows you to write more complex and modular code by supporting object-oriented programming concepts like encapsulation, inheritance, and polymorphism.

a) Code:

```

DECLARE
a number;
b number;
c number;
begin
a:=&a;
b:=&b;
c:=&c;
if (a>b and a>c)
then
dbms_output.put_line('a is maximum' || a);
elsif (b>a and b>c)
then
dbms_output.put_line('b is maximum' || b);
else
dbms_output.put_line('c is maximum' || c);

```

end if;

END;

/

Output:

STDIN

462

92

62

Output:

Enter value for a: old 6: a:=&a;

new 6: a:=462;

Enter value for b: old 7: b:=&b;

new 7: b:=92;

Enter value for c: old 8: c:=&c;

new 8: c:=62;

a is maximum 462

b) Code:

Declare

a number;

b number;

begin

```

a:=&a;

b:=&b;

dbms_output.put_line('Before swapping a='||a||' and b='||b);

a:=a+b;

b:=a-b;

a:=a-b;

dbms_output.put_line('After swapping a= '||a||' and b='||b);

end;

/

```

#### Output:

STDIN

625  
62

---

#### Output:

```

Enter value for a: old    6: a:=&a;
new    6: a:=625;
Enter value for b: old    7: b:=&b;
new    7: b:=62;
Before swapping a=625 and b=62
After swapping a= 62 and b=625

```

c) Code:

```
DECLARE
ename varchar2(15);
basic number;
da number;
hra number;
pf number;
netsalary number;
years salary number;
begin
ename:=' &ename';
basic:=&basic;
da:=basic*(30/100);
hra:= basic*(10/100);
if (basic<8000)
then
pf:=basic*(8/100);
elsif(basic>=8000 and basic<=16000)
then
pf:=basic*(10/100);
end if;
netsalary:=basic+da+hra-pf;
years salary:=netsalary*12;
dbms_output.put_line('Employee name: ' || ename);
dbms_output.put_line('Providend Fund: ' || pf);
dbms_output.put_line('Net salary: ' || netsalary);
dbms_output.put_line('Year salary: ' || years salary);
end;
/
```

Ouput:

STDIN

rohan  
~~~~~  
6000

---

Output:

```
Enter value for ename: old 10: ename:='&ename';
new 10: ename:='rohan';
Enter value for basic: old 11: basic:='&basic';
new 11: basic:=6000;
Employee name: rohan
Providend Fund: 480
Net salary: 7920
Year salary: 95040
```

Conclusion:

A PL/SQL block is defined by the keywords DECLARE, BEGIN, EXCEPTION, and END. These keywords divide the block into a declarative part, an executable part, and an exception-handling part. Only the executable part is required. A block can have a label.

Question and Answer:

Q1. What is PL/SQL?

PL/SQL is referred to as Procedural Language for SQL. It is used for decision-making, and iteration and is especially helpful for embracing the SQL statements. PL/SQL can run a series of queries by one block only using a single command statement.

Q2. What are the main features of PL SQL?

PL/SQL supports structured programming language and provides high integrity and productivity to programmers. It includes conditions and loops for procedural language elements.

Q3. What is the use of PL/SQL?

PL/SQL helps decrease the traffic between the users and applications by sending the block of statements. In real-world, PL/SQL is used in the syntactic analysis of XML, directory traversal, and password policy enforcement.

Q4. What do you mean by a package in PL SQL?

A package is a file that groups procedures, functions, variables, cursors, and other program elements into a single database item.

Q5. What do you know about exceptions and their types in PL/SQL?

An exception is any error condition that develops during the execution of the PL/SQL code block.

Two types of exceptions in PL/SQL are Pre-defined and User-defined.

- **Pre-defined-** They are used to manage the unknown conditions or errors that come in PL/SQL block series code.
- **User-defined-** They are used to manage the specific conditions or errors that come in PL/SQL block series code.