**Database Management System Laboratory**

**GROUP A  
Experiment No: 4**

Name: Onasvee Banarse

TE COMPUTER SHIFT 1

Roll no: 09

**Aim**: Write a PL/SQL block to calculate fine for a library book by accessing borrower information from the database.

**Problem Statement**:

Unnamed PL/SQL code block: Use of Control structure and Exception handling is mandatory. Write a PL/SQL block of code for the following requirements:-

Schema:

1. Borrower(Rollin, Name, DateofIssue, NameofBook, Status)

2. Fine(Roll\_no,Date,Amt)

• Accept roll\_no & name of book from user.

• Check the number of days (from date of issue), if days are between 15 to 30 then fine amount will be Rs 5per day.

• If no. of days>30, per day fine will be Rs 50 per day & for days less than 30, Rs. 5 per day.

• After submitting the book, status will change from I to R.

• If condition of fine is true, then details will be stored into fine table.

**Objective:**

1. To learn and understand PL/SQL in Oracle.

**Requirement:** Any OS, MySQL, Workbench, Oracle Workbench (Online).

**Theory:**

PL/SQL stands for Procedural Language extension of SQL. PL/SQL is a combination of SQL along with the procedural features of programming languages. It was developed by Oracle Corporation in the early 90’s to enhance the capabilities of SQL.

**Architecture of PL/SQL:**

The PL/SQL architecture mainly consists of following 3 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 with 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.

**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.

**Basic Difference between SQL and PL/SQL:**

In this section, we will discuss some differences between SQL and PL/SQL.

| **Key** | **SQL** | **PL/SQL** |
| --- | --- | --- |
| Definition | SQL, is Structural Query Language for database. | PL/SQL is a programming language using SQL for a database. |
| Variables | SQL has no variables. | PL/SQL has variables, data types etc. |
| Control Structures | SQL has no FOR loop, if control and similar structures. | PL/SQL has FOR loop, while loop, if controls and other similar structures. |
| Operations | SQL can execute a single operation at a time. | PL/SQL can perform multiple operation at a time. |
| Language Type | SQL is a declarative language. | PL/SQL is a procedural language. |
| Embedded | SQL can be embedded in a PL/SQL block. | PL/SQL can also be embedded in SQL code. |
| Interaction | SQL directly interacts with database server. | PL/SQL does not directly interact with database server. |
| Orientation | SQL is data-oriented language. | PL/SQL is application-oriented language. |
| Objective | SQL is used to write queries, create, and execute DDL and DML statements. | PL/SQL is used to write program blocks, functions, procedures, triggers, and packages. |

**PL/SQL Block Structure:**

Basic Syntax of PL/SQL which is a block-structured language; this means that the PL/SQL programs are divided and written in logical blocks of code.

A PL/SQL block has up to four different sections, only one of which is mandatory:

* Header

Used only for named blocks. The header determines the way the named block or program must be called. Optional.

* Declaration section

Identifies variables, cursors, and subblocks that are referenced in the execution and exception sections. Optional.

* Execution section

Statements the PL/SQL runtime engine will execute at runtime. Mandatory.

* Exception section

Handles exceptions to normal processing (warnings and error conditions). Optional.

Every PL/SQL statement ends with a semicolon (;). PL/SQL blocks can be nested within other PL/SQL blocks using BEGIN and END. Following is the basic structure of a PL/SQL block −

DECLARE

<declarations section>

BEGIN

<executable command(s)>

EXCEPTION

<exception handling>

END;

The 'Hello World' Example

DECLARE

message varchar2(20):= 'Hello, World!';

BEGIN

dbms\_output.put\_line(message);

END;

/

The end; line signals the end of the PL/SQL block.

**PL/SQL Placeholders**

Placeholders are temporary storage area. PL/SQL Placeholders can be any of Variables, Constants and Records. Oracle defines placeholders to store data temporarily, which are used to manipulate data during the execution of a PL SQL block.

**Define PL/SQL Placeholders**: Depending on the kind of data you want to store, you can define placeholders with a name and a datatype. Few of the datatypes used to define placeholders are as given below. Number (n,m) , Char (n) , Varchar2 (n) , Date , Long , Long raw, Raw, Blob, Clob, Nclob, Bfile.

**PL/SQL Variables:** These are placeholders that store the values that can change through the PL/SQL Block.

**General Syntax to declare a variable is**

variable\_name datatype [NOT NULL := value ];

* variable\_name is the name of the variable.
* datatype is a valid PL/SQL datatype.
* NOT NULL is an optional specification on the variable.
* value or DEFAULT valueis also an optional specification, where you can initialize a variable.
* Each variable declaration is a separate statement and must be terminated by a semicolon.

For example, if you want to store the current salary of an employee, you can use a variable.

DECLARE

salary number (6);

The below example declares two variables, one of which is a not null.

DECLARE

salary number(4);

dept varchar2(10) NOT NULL := “HR Dept”;

The below example declares two variables, one of which is a not null.

DECLARE

salary number(4);

dept varchar2(10) NOT NULL := “HR Dept”;

The below example declares two variables, one of which is a not null.

DECLARE

salary number(4);

dept varchar2(10) NOT NULL := “HR Dept”;

**SQL Queries:**

**Conclusion:**

Thus we have successfully implemented PL/SQL block to retrieve fine for issued library book by reading borrower information from the database.