

DBMS Assignment A5

```
mysql> create database assign5;  
Query OK, 1 row affected (0.01 sec)
```

```
mysql> use assign5;  
Database changed  
mysql> create table Borrower(Rollin int primary key,Name varchar(10),DateofIssue  
varchar(10),NameofBook varchar(10),Status varchar(10));  
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> create table Fine(Rollin int,Days int,amt int,foreign key(Rollin)references  
Borrower(Rollin));  
Query OK, 0 rows affected (0.09 sec)
```

```
mysql> insert into Borrower values(1,'Dhruvil','2020-06-01','COA','I');  
Query OK, 1 row affected (0.02 sec)
```

```
mysql> insert into Borrower values(2,'Soham','2020-06-04','DSA','I');  
Query OK, 1 row affected (0.01 sec)
```

```
mysql> insert into Borrower values(3,'Gaurav','2020-07-17','ADS','I');  
Query OK, 1 row affected (0.01 sec)
```

```
mysql> insert into Borrower values(4,'Aabha','2020-09-18','CN','I');  
Query OK, 1 row affected (0.01 sec)
```

```
mysql> select * from Borrower;
```

Rollin	Name	DateofIssue	NameofBook	Status
1	Dhruvil	2020-06-01	COA	I
2	Soham	2020-09-18	CN	I
3	Gaurav	2020-07-17	ADS	I
4	Aabha	2020-06-04	DSA	I

```
4 rows in set (0.01 sec)
```

```
mysql> select * from Fine;  
Empty set (0.02 sec)
```

```
mysql> delimiter $$
```

```
mysql> create procedure proc_lib1(In roll Integer,In book_name varchar(20))
```

```
-> begin
```

```
-> declare no_of_days int;
```

```
-> declare fine_amount int;
```

```
-> declare issue_date date;
```

```
-> set fine_amount:=0;
```

```
-> select DateofIssue into issue_date from Borrower where Rollin=roll and NameofBook =  
book_name;
```

```

-> select issue_date;
-> select current_date-issue_date into no_of_days from Borrower where Rollin=roll;
-> select no_of_days;
-> if no_of_days>=15 and no_of_days<=30
-> then
-> set fine_amount:=(no_of_days-15)*5;
-> elseif no_of_days>30 then set fine_amount:=(no_of_days-30)*50 + 75;
-> end if;
-> update Borrower set status="R" where Rollin=roll;
-> if fine_amount>=0 then
-> insert into Fine values(roll,no_of_days,fine_amount);
-> elseif fine_amount<0 then
-> begin
-> DECLARE EXIT HANDLER FOR SQLEXCEPTION SELECT 'Table not found';
-> SELECT * from borrower;
-> end;
-> end if;
-> end;
-> $$

```

Query OK, 0 rows affected (0.03 sec)

```
mysql> call proc_lib1(3,'ADS');$$
```

```

+-----+
| issue_date |
+-----+
| 2020-07-17 |
+-----+

```

1 row in set (0.05 sec)

```

+-----+
| no_of_days |
+-----+
|          146 |
+-----+

```

1 row in set (0.06 sec)

Query OK, 1 row affected (0.07 sec)

```
mysql> select * from Fine;
```

```
-> ^C
```

```
mysql> select * from Fine;$$
```

```

+-----+-----+-----+
| Rollin | Days | amt   |
+-----+-----+-----+
|        3 | 146 | 5875 |
+-----+-----+-----+

```

1 row in set (0.00 sec)

```
mysql> select * from Borrower;$$
```

```

+-----+-----+-----+-----+-----+
| Rollin | Name   | DateofIssue | NameofBook | Status |
+-----+-----+-----+-----+-----+

```

	1	Dhruvil	2020-06-01	COA	I	
	2	Soham	2020-09-18	CN	I	
	3	Gaurav	2020-07-17	ADS	R	
	4	Aabha	2020-06-04	DSA	I	

4 rows in set (0.00 sec)

MES College of Engineering Pune-01

Department of Computer Engineering

Name of Student: Dhruvil Shah	Class: TE Comp 1
Semester/Year: 5th/2020	Roll No: 047
Date of Performance:	Date of Submission:
Examined By:	Experiment No: Part A-05

GROUP: A ASSIGNMENT NO: 05

AIM: Write a PL/SQL block of code using Control structure and Exception handling.

PROBLEM STATEMENT:

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.

OBJECTIVES:

- To learn the concept of procedural language.
- To learn Control structure and Exception handling.

PRE - REQUISITES:

Interactive SQL commands, PL/SQL programming, use of oracle 11g database Editor.

APPARATUS:

- Operating System recommended: 64-bit Open source Linux or its derivative
- Front End :- Oracle Editor
- Back end: Oracle 11g

Syntax for Control Structure and Exception Handling.

IF-THEN Statement

```
IF condition THEN
    sequence_of_statements
END IF;
```

IF-THEN-ELSE Statement

```
IF condition THEN
    sequence_of_statements1
ELSE
    sequence_of_statements2
END IF;
```

IF-THEN-ELSIF Statement

```
IF condition1 THEN
    sequence_of_statements1
ELSIF condition2 THEN
    sequence_of_statements2
ELSE
    sequence_of_statements3
END IF;
```

LOOP

```
LOOP
    sequence_of_statements
END LOOP;
```

EXIT-WHEN

```
LOOP
    ....
    EXIT WHEN condition -- exit loop if condition is true
    ...
END LOOP;
```

WHILE-LOOP

```
WHILE condition LOOP
    sequence_of_statements
END LOOP;
```

FOR-LOOP

```
FOR counter IN [REVERSE] lower_bound..higher_bound LOOP
    sequence_of_statements
END LOOP;
```

GOTO Statement

```
BEGIN
    ...
    GOTO insert_row;
    ...
    <<insert_row>>
    INSERT INTO emp VALUES ...
END;
```

Syntax for Exception Handling

An exception is an error condition during a program execution. PL/SQL supports programmers to catch such conditions using EXCEPTION block in the program and an appropriate action is taken against the error condition. There are two types of exceptions –

- System-defined exceptions
- User-defined exceptions

```
DECLARE
    <declarations section>
BEGIN
    <executable command(s)>
EXCEPTION
    <exception handling goes here >
    WHEN exception1 THEN
        exception1-handling-statements
    WHEN exception2 THEN
        exception2-handling-statements

    .....
    WHEN others THEN
        exception3-handling-statements
END;
```

Raising Exceptions

Exceptions are raised by the database server automatically whenever there is any internal database error, but exceptions can be raised explicitly by the programmer by using the command RAISE. Following is the simple syntax for raising an exception –

```
DECLARE
    exception_name EXCEPTION;
BEGIN
    IF condition THEN
        RAISE exception_name;
    END IF;
EXCEPTION
    WHEN exception_name THEN
        statement;
END;
```

CONCLUSION:

QUESTIONS:

1. What are the advantages of PLSQL over SQL
2. List Different Pre-defined Exceptions.

Q1 What are the advantages of PLSQL over SQL?

Ans PLSQL is a procedure language which is applicable on SQL, using PLSQL we can write multiple statements in a single execution of code block. When we are working with SQL we can only fire or execute one query/statement at a time and also lacks the capability of logically grouping multiple database operations. In PLSQL data variables are available. Control structures are available in PLSQL.

Q2 List different Pre-defined exceptions.

Ans The pre-defined exceptions in PLSQL are:

- 1] ACCESS_INTO_NULL
- 2] CASE_NOT_FOUND
- 3] COLLECTION_IS_NULL
- 4] DUP_VAL_ON_INDEX
- 5] INVALID_CURSOR
- 6] INVALID_NUMBER
- 7] LOGIN_DENIED.
- 8] NO_DATA_FOUND
- 9] NOT_LOGGED_ON
- 10] PROGRAM_ERROR
- 11] SELF_IS_NULL
- 12] STORAGE_ERROR
- 13] VALUE_ERROR
- 14] ZERO_DIVIDE