Page No.
Date

Assignment - 05

* Title: PL/SQL block

* Date of Completion: 09/09/2020

* Date of Submission: 30/09/2020

* Problem Statement:
  Write a PL/SQL block of code for the following requirements

Schema:

Customes (Cust_id , Name, DateofPayment, NameofScheme, Status)

Fine (cust_id, Date, Amt)

1. Accept Cust_id and name of scheme from user.
2. Check number of days (from date of payment) if days are between 15 to 30 then fine amount will Rs 5 per day.
3. If no of days >30, per day fine will be Rs 50 & for days less than 30, Rs 5 per day.
4. After payment, status will change from N to P.
5. If the condition of fine is true, then details will be stored into Fine table.

* Learning Objective:
  - To understand PL/SQL block code requirements
  - To understand exception handling
  - To understand basic structure of PL/SQL block
  - To apply control structure.

Teacher's Sign.: _____

Page No.
Date

* Learning Outcome: Students will be able to implement
  - PL/SQL block, user-defined and pre-defined exception handling.
  - Control structure using PL/SQL.

* S/W And H/W Requirements:
  Windows 10 (64 bit), MySQL, i5 processor, keyboard, mouse.

* Theory:
* ORACLE:

- PL/SQL - it stands for Procedural language / Structured Query language. PL/SQL offers a set of procedural commands (IF statements, loops, assignments), organized with blocks that complement and extend the reach of SQL.

- Blocks in PL/SQL:
  A block in PL/SQL is defined by the keyword DECLARE, BEGIN, EXCEPTION, AND END which breakup the block into three sections:
  1. Declarative: statements that declare variables, constants & other code elements
  2. Executables: statements that are run when the block is executed
  3. Exception handling: a specially structured section you can use to catch up or trap any exceptions

Teacher's Sign: _____

Page No.
Date

Exception Handling:

- handles exception
- using this, we can test the code & avoid it from exiting abruptly.
- Exception message consists of three parts
  i) Type of exception
  ii) An Error code
  iii) message

Syntax:

```
DECLARE
        Declaration section
BEGIN
        Exception section and execution section
EXCEPTION
    WHEN exname1 THEN
            - error handling statements
    WHEN exnamed THEN
            - error handling statements
    WHEN Others THEN
            - error handling statements
END
```

- Types of Exception:

There are 3 types of exceptions:

i) Named System exceptions:

- System exceptions are automatically raised by Oracle, when a program violates a RDBM rule.

ex: NO_DATA_FOUND and ZERO_DIVIDE

Teacher's Sign.:

ii) Unnamed System exceptions:
- Oracle does not provide a name for these
- They have a code & associated message
- We can assign name to these using a Pragma called EXCEPTION_INIT.

Syntax:

```
DECLARE
    exception_name  EXCEPTION;
    PRAGMA
    EXCEPTION_INIT (exception_name, Err.code);
    BEGIN
    Execution section
    EXCEPTION
    WHEN exception_name THEN
        handle exception
END;
```

iii) User-defined Exception:
- We can define exceptions explicitly based on business rules.
- They should be declared and raised explicitly in the PL/SQL block

- CONTROL SEQUENCE:
  - The selection structure tests a condition, then executes one sequence of statements instead of other depending on whether the condition is true or false.

ex: Using the IF-THEN: Statement.

Teacher's Sign.: _____

Syntax:

```
DECLARE
   ,declaration
BEGIN
  IF condition THEN
        execution statements
  END IF;
  END;
```

° MySQL

- Stored procedures:
  1. procedure is a subroutine like a sub-program in a regular computing language

- Creation of procedure:
  Syntax:

```
create procedure procedure-name (parameter)
begin
declare
        - declaration statements
        - execution statements
end;
```

- Dropping a Procedure:
  Syntax:

```
drop procedure procedure_names
```

- Parameters:
  - a procedure can have no parameters.

Teacher's Sign.: _____

- or it can have any number of in, out, inout parameters

IN - when you use this mode you must pass the parameters value when you call the stored procedure.

OUT - the value of out parameters can change inside the procedure & its returned to program calling the procedure

INOUT - you can pass initial value, change it and return the new value of this parameter

- Control structure:

The if-else control structure can be used to make decisions inside the execution block/section

Syntax:

```
If condition then
        - execution statements
else if condition then
        - execution statements
else
        - execution statements
end if;
```

- ERROR HANDLING:

It helps to handle errors inside a stored procedure but continuing or exiting the current code block's execution and issuing a meaningful message.

Teacher's Sign.:

Page No.
Date

Syntax:

DECLARE action HANDLER for condition_value
statement;
SET message as 'message';

action =>

continue - the execution of enclosing code block (BEGIN ---- END) continues

Exit - the execution of enclosing code block terminates

CALLING a PROCEDURE:

Syntax:

call procedure_name (parameters);

* TEST CASES:

	INPUT	EXPECTED OUTPUT	RESULT
i)	call proc1(2,'fixed_deposit')	1, 2020-08-05, 1950	Success
ii)	call proc1(7,'recurring deposit')	customer has already paid	Success.

* conclusion:-

- Thus, we learnt about PL/SQL block
- We learnt about exception handling in PL/SQL block
- We implemented stored procedure in MySQL.
- We implemented error handling inside stored procedures in MySQL.

Teacher's Sign.: _____