# Assignment No. 5

- Title : PL/SQL Block Implementation

- Problem statement : Write a PL/SQL block of code for the following requirements -

  Schema :

  1) Borrower (Roll_no, Name, Date of Issue, Name of Book, Status)
  2) Fine (Roll_no, Date, Amt)

  - Accept roll_no and 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. 5 per day.
  - If no. of days >30, per day fine will be Rs. 50 per day and for days less than 30, Rs 5 per day.
  - After subnetting the book, status will change from I to R
  - If condition of fine is true, then details will be stored in to fine table.

- Objective : To

  1) Understand the control structure
  2) Understand exception handling.

- Outcomes: We will be able to

  1) Understand and write PL/SQL code
  2) Implement exception handing
  3) Apply control struntyre and understand basic structure of PL/SQL block.

- S/W and H/W Requirements :
  MySQL, PC with the configuration as latest version, of 64 bit OS e.g. Fedora, Windows 10, 8 GB RAM.

- Theory:
  PL/SQL :
  PL/SQL standers for Procedural language / Structure Query language. PL/SQL offers a set of procedural commands organized within blocks (that complement and extend SQL)
  PL/SQL is a block structured language. A PL/SQL block defined by the keywords DECLARE, BEGIN, EXCEPTION and END which breaks them into 3 sections.

1) DECLARADTIVE : statements that declare variables, constants other code elements.
2) EXECUTABLE : Statement that run when the block is executed
3) Exception: A specifically structured section used for handling any exception

For example,

```
BEGIN
    DBMS_OUTPUT. put_line ("Hello World");
END;
```

Features of PL/SQL
1) Tightly integrated with SQL
2) Offers Extensive data types
3) Supports objects oriented programming.

प्र.क्र.
Q. No.

4) Offers numerous data types

ADVANTAGES OF PL/SQL

1) Provides high security level
2) Provides access to predefined SQL packages.
3) Applications written in PL/SQL and fully portable.

EXCEPTION HANDLING:

PL/SQL provides a feature to handle the exceptions which occur in a PL/SQL Block known as exception handing. Using exception handling, we can test the code and avoid it from exiting abruptly. When an exception occurs a message which explains its cause is received.

PL/SQL Exception message consists of three parts

1) Type of Exception
2) An Error code
3) A message

STRUCTURE of EXCEPTION HANDLING

```
DECLARE
        Declaration section
    BEGIN
        Exception section
    EXCEPTION
        WHEN  E-name  THEN
            - Error handling statements
        WHEN  OTHERS  THEN
            - Error handling statements      END;
```

When an exception is raised, oracle searhes for an appropriate exception hander in exception section. For example, in the previous example, iferror raised is 'ex_name1', then the error is handled according to statements under it. Since, it is not possible to determine all the possible run time errors during testing for the code. Only one exception can be raised in a Block and the control does not return to the Execution Section after the error is handled.

Types of Exception

There are 3 types of Exceptions.

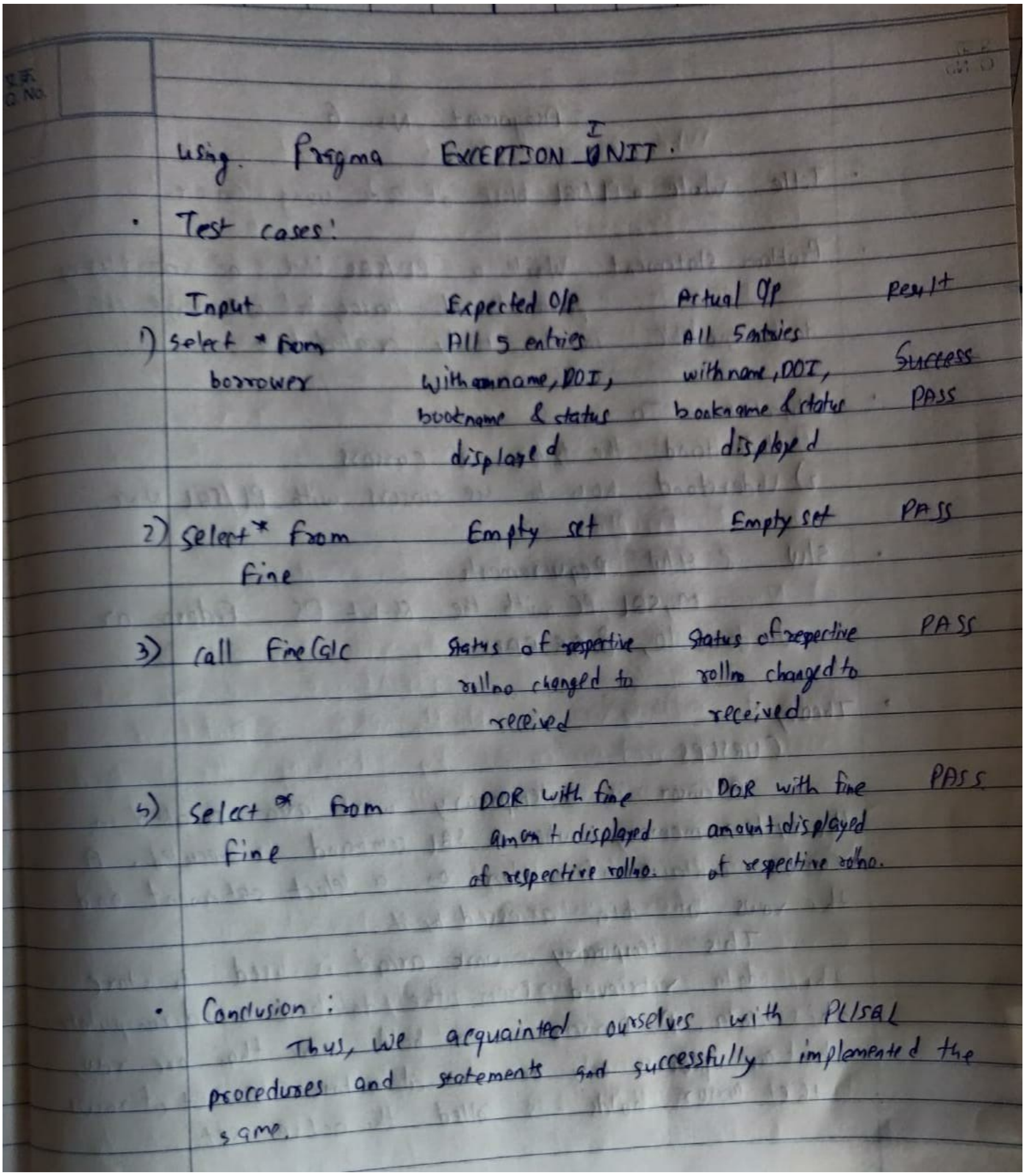
a) Named System Exceptions:

These are automatically raised by oracle, when a program violates a RDBMS rule. These are not declared explicitly raised implicitly when a predefined oracle error occurs and are caught by referencing the standard name within an exception-handling routine.

b) Unnamed System Exception:

Those system exception for which oracle does not provide a name is known as unnamed system exception. These exception do not occur frequently - these exceptions have a code and an associated messge.

They are raised implicitly. If they are not handled in WHEN others they must be handled explicitly To handle the exception explicitly, they must be declared



using Pragma EXCEPTION_I~~U~~NIT.

- Test cases:

| Input | Expected O/P | Actual O/P | Result |
|---|---|---|---|
| 1) select * from borrower | All 5 entries with name, DOI, bookname & status displayed | All 5 entries with name, DOI, bookname & status displayed | Success PASS |
| 2) Select * from fine | Empty set | Empty set | PASS |
| 3) call FineCalc | Status of respective rollno changed to received | Status of respective rollno changed to received | PASS |
| 4) Select * from fine | DOR with fine amount displayed of respective rollno. | DOR with fine amount displayed of respective rollno. | PASS |

- Conclusion:

Thus, we acquainted ourselves with PL/SQL procedures and statements and successfully implemented the same.