K. Abhishek
TCOA76

DBMSL

# Assignment No. : 5

## Aim

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 :
i) Borrower (Rollin, Name, DateofIssue, NameofBook, Status)
ii) Fine (Roll_no, Date, Amt)
a) Accept Roll_no & name of book from user
b) 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.
c) If no. of days >30, per day fine will be Rs. 50 per day & for days less than 30, Rs. 5 per day.
d) After Submitting the book, status will change from I to R
e) If condition of fine is true, the details will be stored into fine table.

## Objective

Learn the concept of PL/SQL.

## Theory

### Introduction to PL/SQL

The development of database applications typically requires language constructs similar to those that can be found in programming languages such as C, C++ or Pascal. These constructs are necessary in order to implement complex data structures and algorithms. A major restriction of the database language SQL, however, is that many tasks cannot be accomplished by using only the provided language elements.

PL/SQL (Procedural Language/SQL) is a procedural extension of Oracle-SQL that offers language constructs similar to those in imperative programming languages.

PL/SQL allows users and designers to develop complex database applications that requires the usage of control structures and procedural elements such as procedures, functions and modules.
PL/SQL blocks that specify procedures and functions can be grouped into packages. A package is similar to a module and has an interface and an implementation part. Another important feature of PL/SQL is that it offers a mechanism to process query related results in a tuple-oriented way.

Major goals of PL/SQL are to:
- Increase the expressiveness of SQL
- Process query results in a tuple-oriented way
- Optimize combined SQL statements
- Develop modular database application programs.
- Reuse program code
- Reduce the cost for maintaining and changing apps.

## Structure of PL/SQL block

The basic unit of code in any PL/SQL program is a block. All PL/SQL programs are composed of blocks. These blocks can be written sequentially.

```
DECLARE
    Declaration section
```

```
BEGIN
    Executable section
EXCEPTION
    Exception handling section
END;
```

i) Declaration section

PL/SQL variables, types, cursors and local subprograms are defined here.

ii) Executable section

Procedural and SQL statements are written here. This is the main section of the block. This section is required.

iii) Exception handling section

Error handling code is written here. This section is optional whether it is defined within body or outside body of program.

Conditional statements & loops used in PL/SQL

Conditional statements check the validity of a condition and accordingly execute a set of statements.

- IF... THEN
- IF... THEN... ELSE
- IF... THEN... ELSIF

Iterative constructs

Iterative constructs are used to execute a set of statements respectively.

- Simple loop
- While loop
- For loop

## Exceptions

Exceptions are errors or warnings in PL/SQL program. PL/SQL implements error handling using exceptions and exception handler.

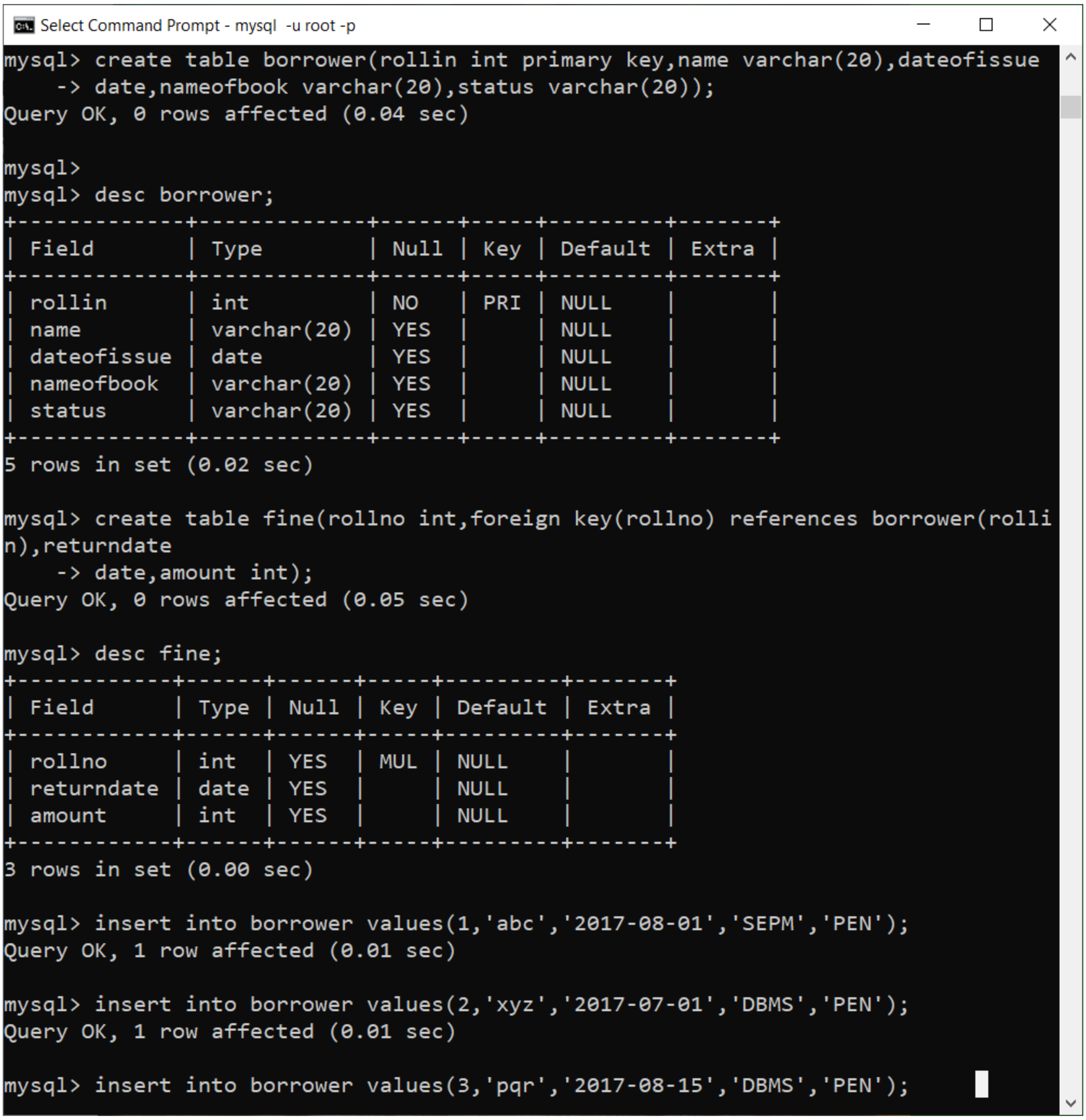
Types : i) Predefined exception ii) User defined exceptions

Syntax :

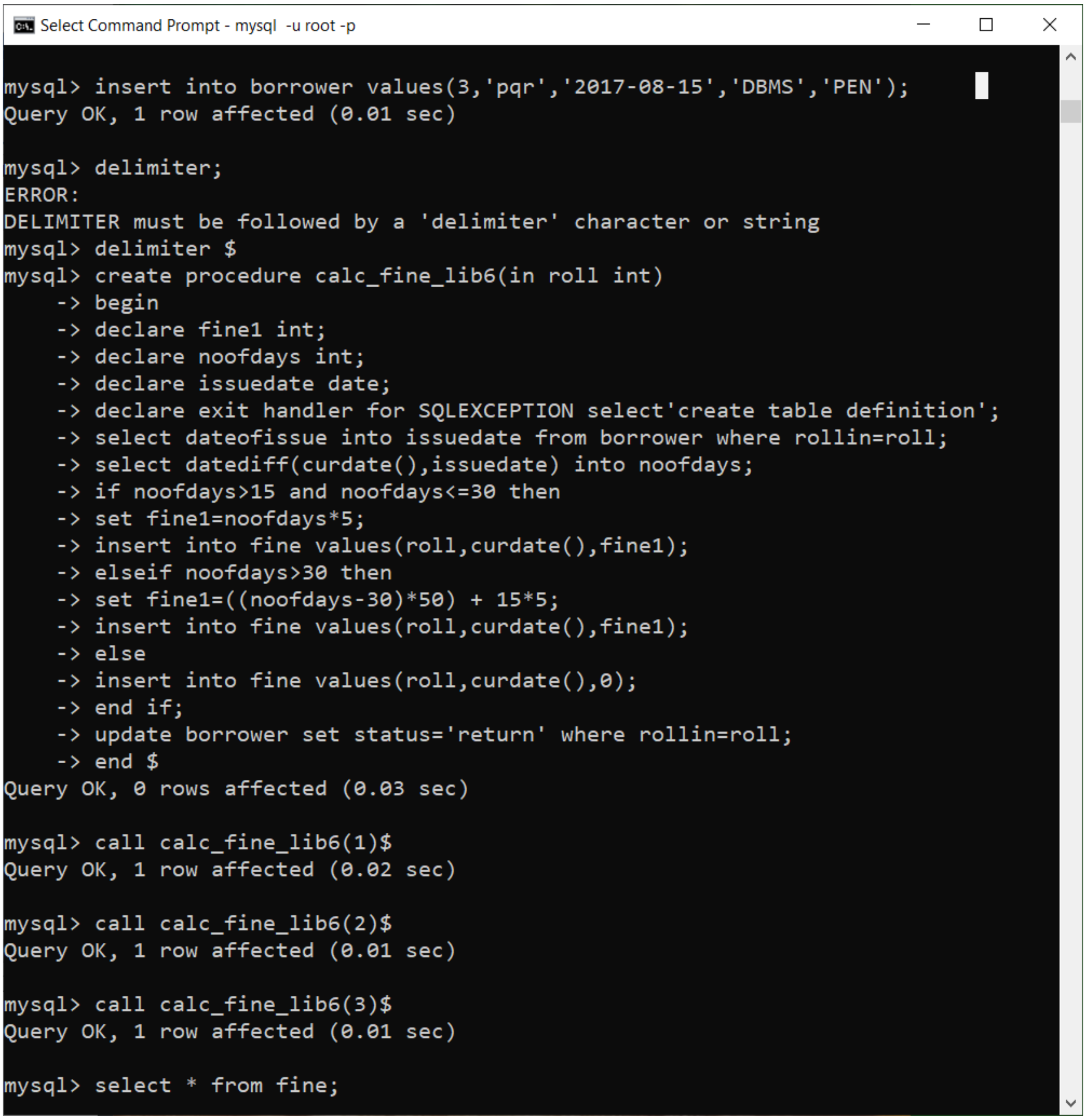
```
DECLARE
      <Exception Name> Exception;
BEGIN
      RAISE <Exception Name >
EXCEPTION
      WHEN < Exception Name > THEN
      < Action>
END;
```

## Conclusion

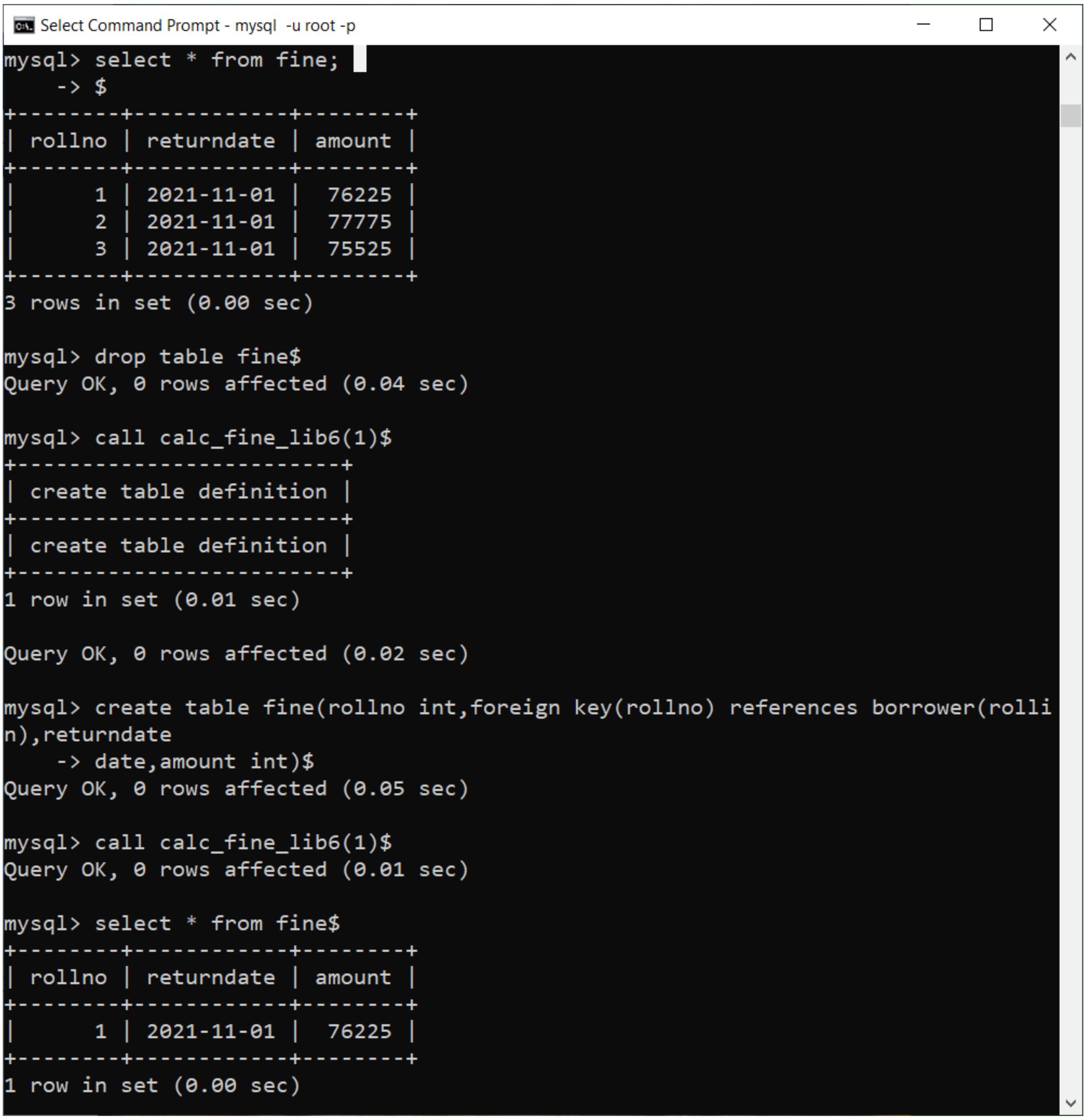
We learned control structure, exceptions and various other concepts of PL/SQL



```
mysql> create table borrower(rollin int primary key,name varchar(20),dateofissue
    -> date,nameofbook varchar(20),status varchar(20));
Query OK, 0 rows affected (0.04 sec)

mysql>
mysql> desc borrower;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| rollin      | int         | NO   | PRI | NULL    |       |
| name        | varchar(20) | YES  |     | NULL    |       |
| dateofissue | date        | YES  |     | NULL    |       |
| nameofbook  | varchar(20) | YES  |     | NULL    |       |
| status      | varchar(20) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
5 rows in set (0.02 sec)

mysql> create table fine(rollno int,foreign key(rollno) references borrower(rolli
n),returndate
    -> date,amount int);
Query OK, 0 rows affected (0.05 sec)

mysql> desc fine;
+------------+------+------+-----+---------+-------+
| Field      | Type | Null | Key | Default | Extra |
+------------+------+------+-----+---------+-------+
| rollno     | int  | YES  | MUL | NULL    |       |
| returndate | date | YES  |     | NULL    |       |
| amount     | int  | YES  |     | NULL    |       |
+------------+------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> insert into borrower values(1,'abc','2017-08-01','SEPM','PEN');
Query OK, 1 row affected (0.01 sec)

mysql> insert into borrower values(2,'xyz','2017-07-01','DBMS','PEN');
Query OK, 1 row affected (0.01 sec)

mysql> insert into borrower values(3,'pqr','2017-08-15','DBMS','PEN');
```



```
mysql> insert into borrower values(3,'pqr','2017-08-15','DBMS','PEN');
Query OK, 1 row affected (0.01 sec)

mysql> delimiter;
ERROR:
DELIMITER must be followed by a 'delimiter' character or string
mysql> delimiter $
mysql> create procedure calc_fine_lib6(in roll int)
    -> begin
    -> declare fine1 int;
    -> declare noofdays int;
    -> declare issuedate date;
    -> declare exit handler for SQLEXCEPTION select'create table definition';
    -> select dateofissue into issuedate from borrower where rollin=roll;
    -> select datediff(curdate(),issuedate) into noofdays;
    -> if noofdays>15 and noofdays<=30 then
    -> set fine1=noofdays*5;
    -> insert into fine values(roll,curdate(),fine1);
    -> elseif noofdays>30 then
    -> set fine1=((noofdays-30)*50) + 15*5;
    -> insert into fine values(roll,curdate(),fine1);
    -> else
    -> insert into fine values(roll,curdate(),0);
    -> end if;
    -> update borrower set status='return' where rollin=roll;
    -> end $
Query OK, 0 rows affected (0.03 sec)

mysql> call calc_fine_lib6(1)$
Query OK, 1 row affected (0.02 sec)

mysql> call calc_fine_lib6(2)$
Query OK, 1 row affected (0.01 sec)

mysql> call calc_fine_lib6(3)$
Query OK, 1 row affected (0.01 sec)

mysql> select * from fine;
```



```
mysql> select * from fine;
    -> $
+--------+------------+--------+
| rollno | returndate | amount |
+--------+------------+--------+
|      1 | 2021-11-01 |  76225 |
|      2 | 2021-11-01 |  77775 |
|      3 | 2021-11-01 |  75525 |
+--------+------------+--------+
3 rows in set (0.00 sec)

mysql> drop table fine$
Query OK, 0 rows affected (0.04 sec)

mysql> call calc_fine_lib6(1)$
+-------------------------+
| create table definition |
+-------------------------+
| create table definition |
+-------------------------+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.02 sec)

mysql> create table fine(rollno int,foreign key(rollno) references borrower(rolli
n),returndate
    -> date,amount int)$
Query OK, 0 rows affected (0.05 sec)

mysql> call calc_fine_lib6(1)$
Query OK, 0 rows affected (0.01 sec)

mysql> select * from fine$
+--------+------------+--------+
| rollno | returndate | amount |
+--------+------------+--------+
|      1 | 2021-11-01 |  76225 |
+--------+------------+--------+
1 row in set (0.00 sec)
```