Exception Handling Using Procedure

1.Creating Database:-

```
show databases;
create database Pract_4;
use Pract_4;
```

2.Creating Tables:-

```
create table Borrower(Roll_no int primary key, Name varchar(50), Date_of_Issue date, Name_of_Book varchar(100), Status char(1));
create table Fine(Roll_no int, Date date, Amt decimal(10,2));
```

3.Inserting Data Into Table Borrower:-

```
insert into Borrower values (101, 'sakshi', '2025-10-01', 'Data Science', 'I'), (102,'Pragati', '2025-09-20', 'AI Basics', 'I'), (103,'Mahima', '2025-10-10', 'Python Programming', 'I');
```

4.Procedure Code:-

```
DELIMITER $$
CREATE PROCEDURE ReturnBook(IN p_Roll_no INT, IN p_Name_of_Book VARCHAR(100))
  BEGIN
    DECLARE v_Date_of_Issue DATE;
    DECLARE v_Days INT;
    DECLARE v_Fine DECIMAL(10,2);
    DECLARE book_not_found CONDITION FOR SQLSTATE '45000';
    DECLARE CONTINUE HANDLER FOR book_not_found
    BEGIN
      SELECT 'Error: Book not found or already returned.' AS Message;
    END;
    SELECT Date_of_Issue INTO v_Date_of_Issue
    FROM Borrower
    WHERE Roll_no = p_Roll_no AND Name_of_Book = p_Name_of_Book AND Status='I';
    IF v_Date_of_Issue IS NULL THEN
      SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Book not found or already returned';
    END IF;
    SET v_Days = DATEDIFF(CURDATE(), v_Date_of_Issue);
```

```
        IF v_Days BETWEEN 15 AND 30 THEN
          SET v_Fine = v_Days * 5;
        ELSEIF v_Days > 30 THEN
          SET v_Fine = (30 * 5) + ((v_Days - 30) * 50);
        ELSE
          SET v_Fine = 0;
        END IF;
        UPDATE Borrower
        SET Status='R'
        WHERE Roll_no = p_Roll_no AND Name_of_Book = p_Name_of_Book;
        IF v_Fine > 0 THEN
          INSERT INTO Fine (Roll_no, Date, Amt)
          VALUES (p_Roll_no, CURDATE(), v_Fine);
        END IF;
        SELECT p_Roll_no AS Roll_no, p_Name_of_Book AS Book, v_Days AS Days_Borrowed, v_Fine AS
Fine_Amount;
   END$$


 DELIMITER ;
```

5. Calling Procedure For ID-101:-

```
CALL ReturnBook(101, 'Data Science');
```

6.Checking Borrower Table is Updated or Not:-

```
select * from Borrower;
```

7.Checking Fine Table Is Updated or Not:-

```
select * from Fine;
```

8.Calling Procedure For ID-102:-

```
CALL ReturnBook(102, 'AI Basics');
```

9.Checking Exception Handling:-

```
CALL ReturnBook(104, 'Psycology');
```