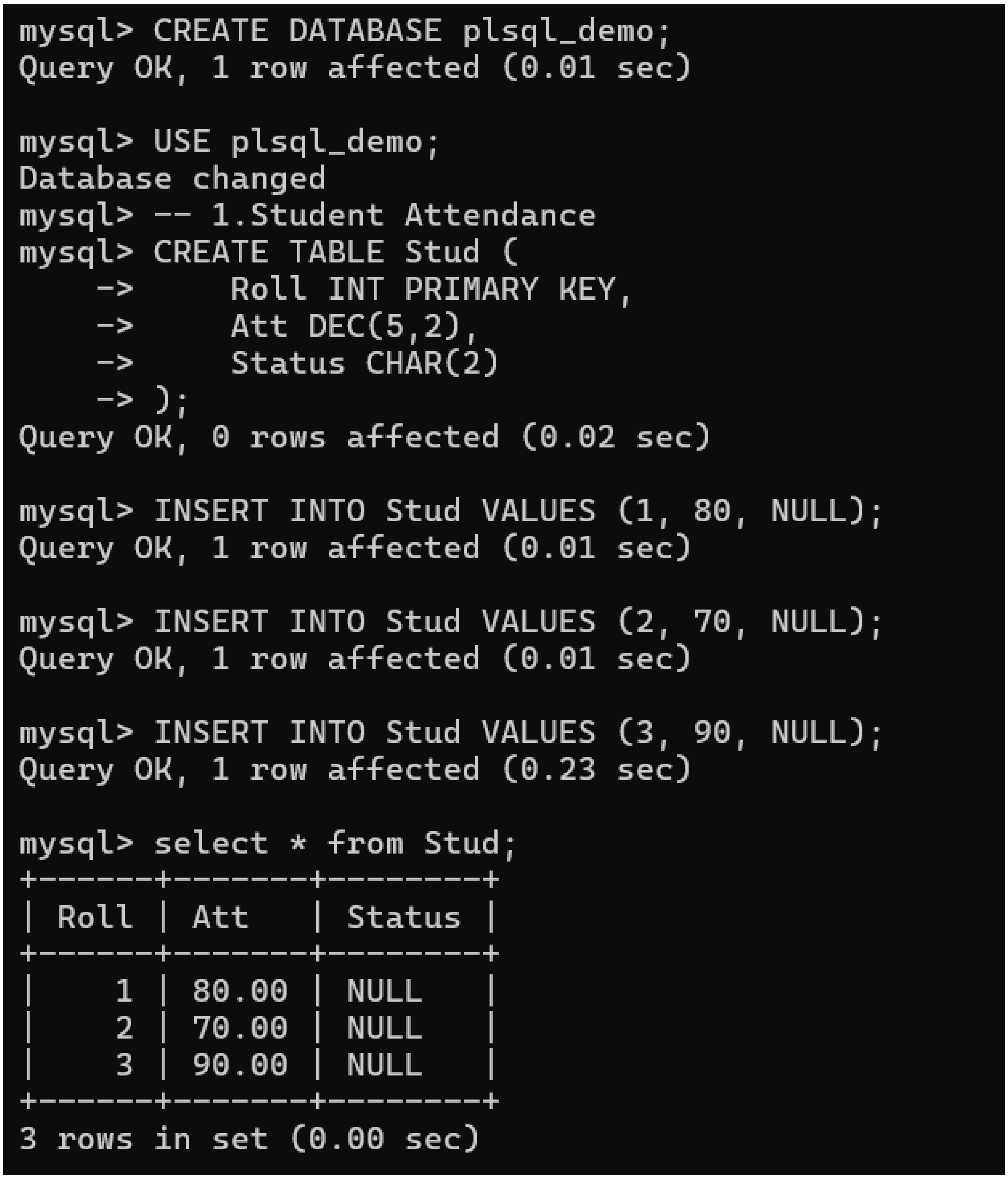
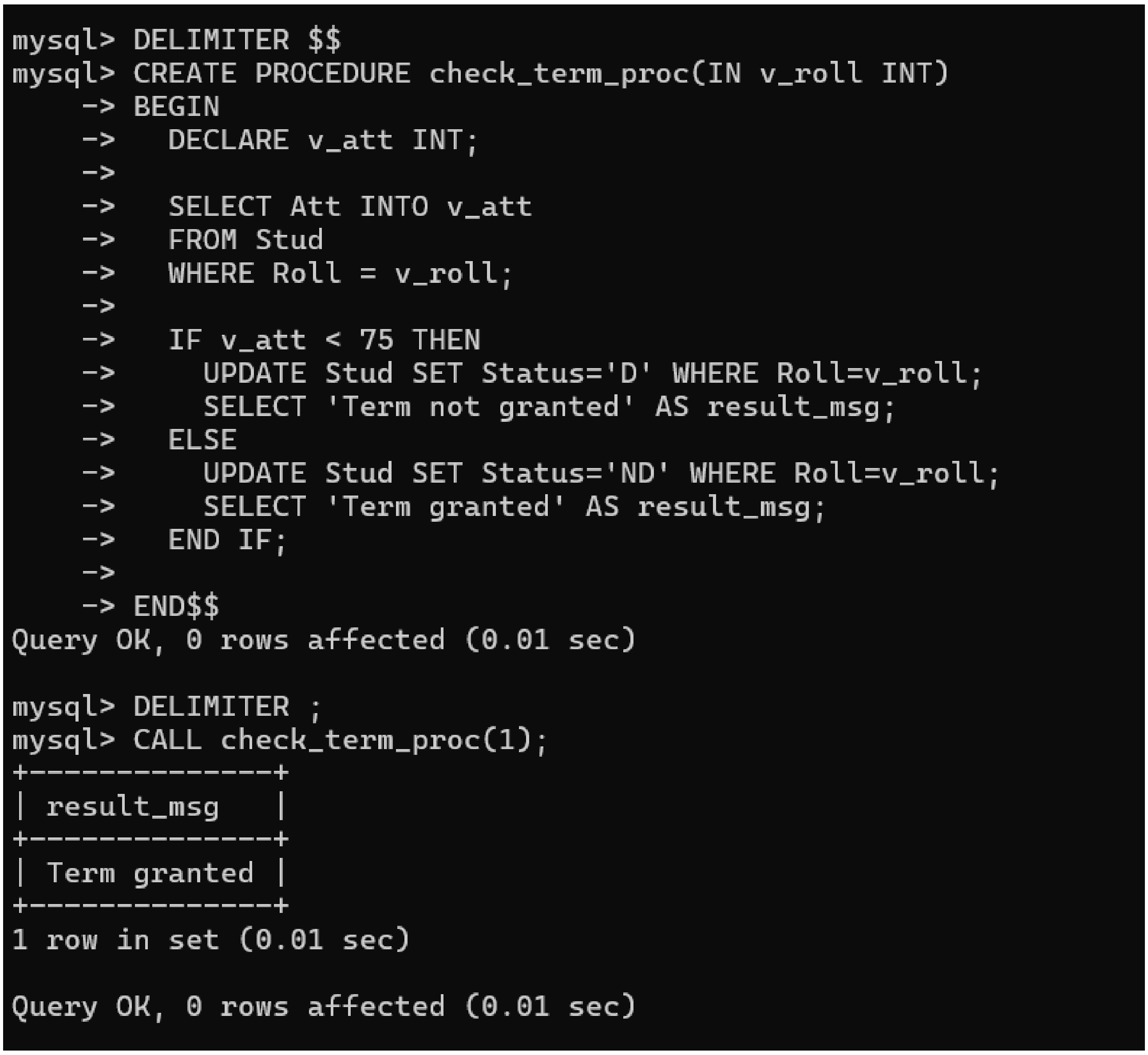
Name: Sudhir Singh
Class: TE - A
Batch: A3
Roll No: 23152
Subject: SL-I (A)

# Lab Assignment 4

1. Consider table Stud(Roll, Att,Status)
Write a PL/SQL block for following requirement and handle the exceptions.
Roll no. of student will be entered by user. Attendance of roll no. entered by user will be checked in Stud table. If attendance is less than 75% then display the message “Term not granted” and set the status in stud table as “D”. Otherwise display message “Term granted” and set the status in stud table as “ND”.

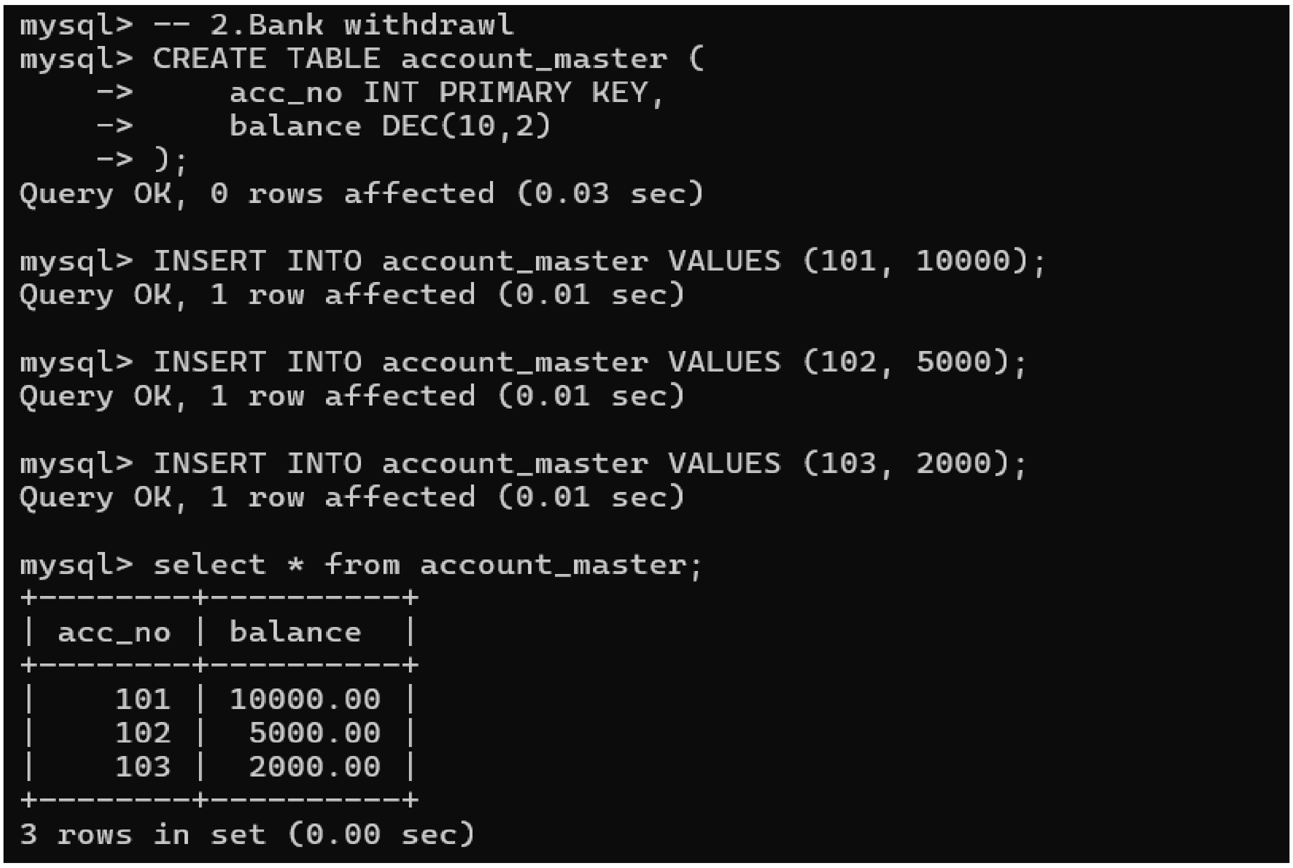


```
mysql> CREATE DATABASE plsql_demo;
Query OK, 1 row affected (0.01 sec)

mysql> USE plsql_demo;
Database changed
mysql> -- 1.Student Attendance
mysql> CREATE TABLE Stud (
    ->     Roll INT PRIMARY KEY,
    ->     Att DEC(5,2),
    ->     Status CHAR(2)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> INSERT INTO Stud VALUES (1, 80, NULL);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Stud VALUES (2, 70, NULL);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Stud VALUES (3, 90, NULL);
Query OK, 1 row affected (0.23 sec)

mysql> select * from Stud;
+------+-------+--------+
| Roll | Att   | Status |
+------+-------+--------+
|    1 | 80.00 | NULL   |
|    2 | 70.00 | NULL   |
|    3 | 90.00 | NULL   |
+------+-------+--------+
3 rows in set (0.00 sec)
```

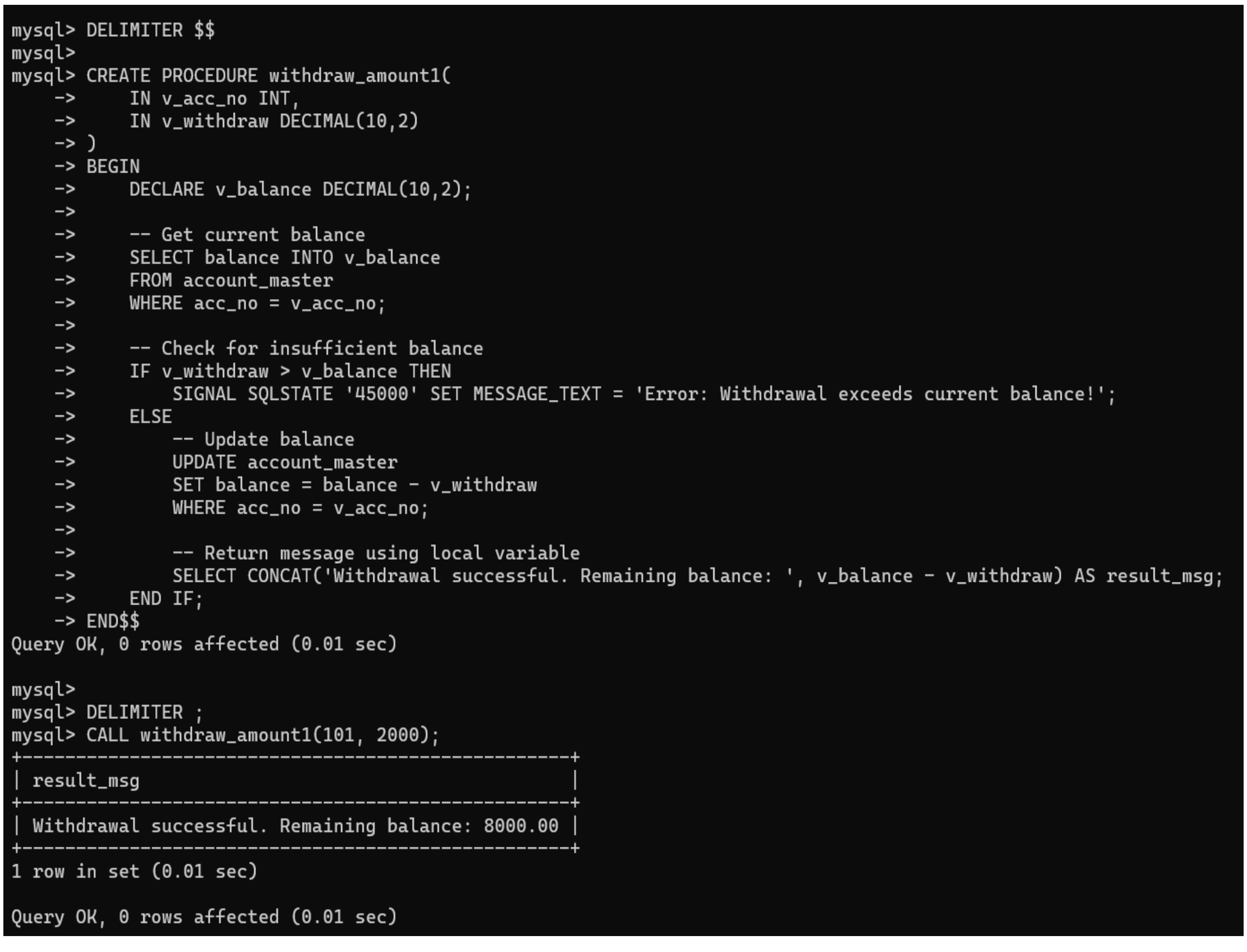


```
mysql> DELIMITER $$
mysql> CREATE PROCEDURE check_term_proc(IN v_roll INT)
    -> BEGIN
    ->   DECLARE v_att INT;
    ->
    ->   SELECT Att INTO v_att
    ->   FROM Stud
    ->   WHERE Roll = v_roll;
    ->
    ->   IF v_att < 75 THEN
    ->     UPDATE Stud SET Status='D' WHERE Roll=v_roll;
    ->     SELECT 'Term not granted' AS result_msg;
    ->   ELSE
    ->     UPDATE Stud SET Status='ND' WHERE Roll=v_roll;
    ->     SELECT 'Term granted' AS result_msg;
    ->   END IF;
    ->
    -> END$$
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql> CALL check_term_proc(1);
+--------------+
| result_msg   |
+--------------+
| Term granted |
+--------------+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)
```

2. Write a PL/SQL block for following requirement using user defined exception handling.
The account_master table records the current balance for an account, which is updated whenever, any deposits or withdrawals takes place. If the withdrawal attempted is more than the current balance held in the account. The user defined exception is raised, displaying an appropriate message. Write a PL/SQL block for above requirement using user defined exception handling.

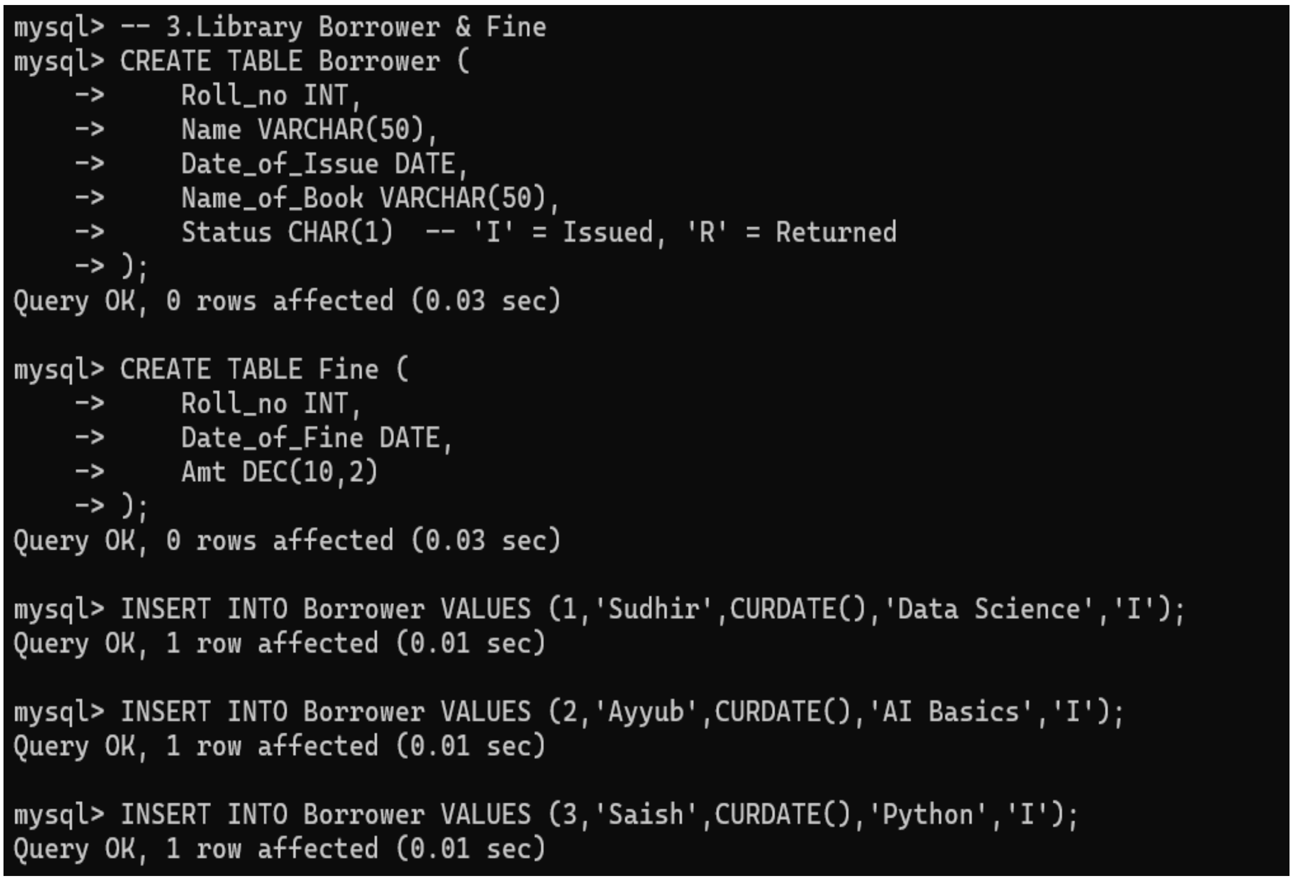


```
mysql> -- 2.Bank withdrawl
mysql> CREATE TABLE account_master (
    ->     acc_no INT PRIMARY KEY,
    ->     balance DEC(10,2)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> INSERT INTO account_master VALUES (101, 10000);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO account_master VALUES (102, 5000);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO account_master VALUES (103, 2000);
Query OK, 1 row affected (0.01 sec)

mysql> select * from account_master;
+--------+----------+
| acc_no | balance  |
+--------+----------+
|    101 | 10000.00 |
|    102 |  5000.00 |
|    103 |  2000.00 |
+--------+----------+
3 rows in set (0.00 sec)
```

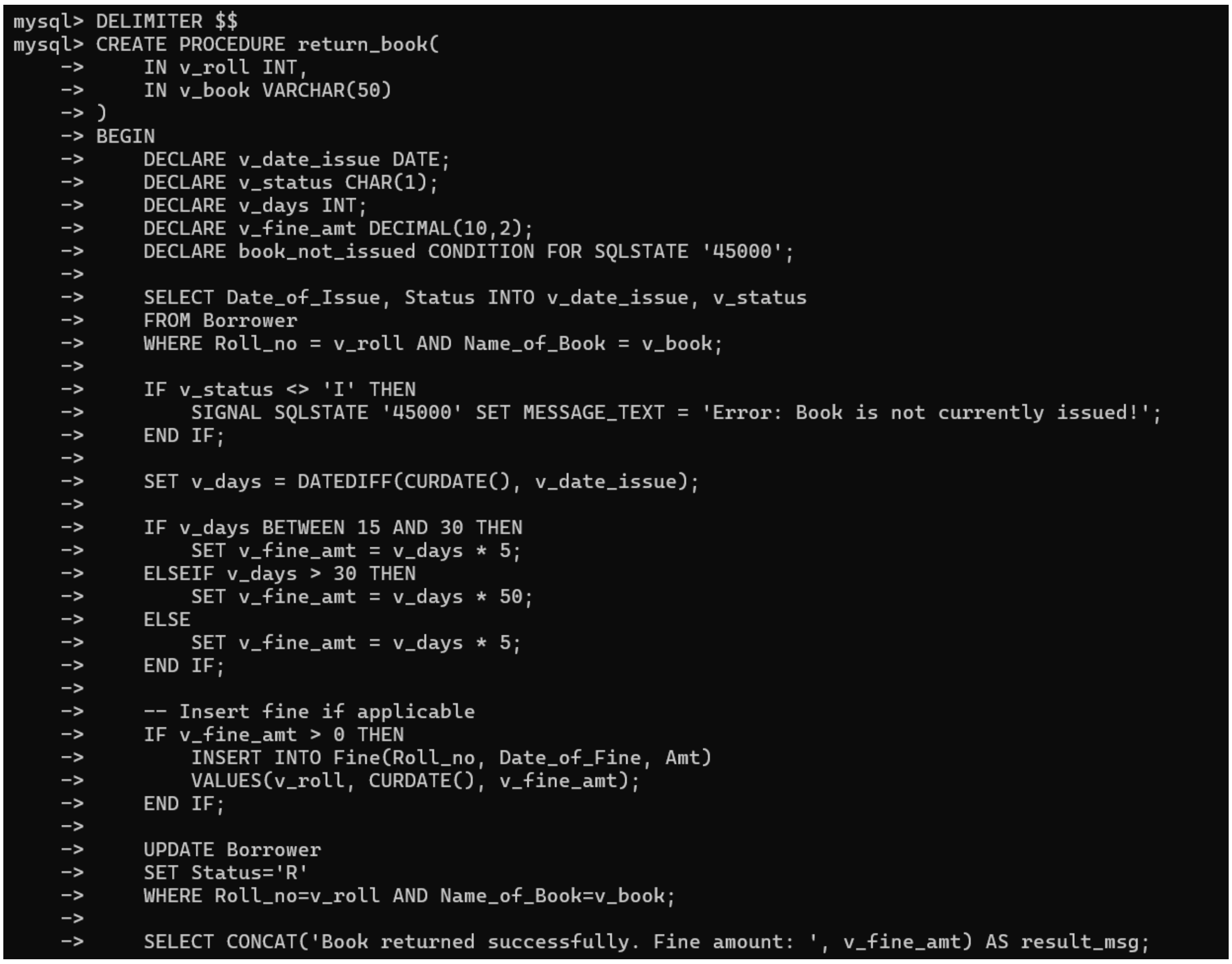


```
mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE withdraw_amount1(
    ->     IN v_acc_no INT,
    ->     IN v_withdraw DECIMAL(10,2)
    -> )
    -> BEGIN
    ->     DECLARE v_balance DECIMAL(10,2);
    ->
    ->     -- Get current balance
    ->     SELECT balance INTO v_balance
    ->     FROM account_master
    ->     WHERE acc_no = v_acc_no;
    ->
    ->     -- Check for insufficient balance
    ->     IF v_withdraw > v_balance THEN
    ->         SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Error: Withdrawal exceeds current balance!';
    ->     ELSE
    ->         -- Update balance
    ->         UPDATE account_master
    ->         SET balance = balance - v_withdraw
    ->         WHERE acc_no = v_acc_no;
    ->
    ->         -- Return message using local variable
    ->         SELECT CONCAT('Withdrawal successful. Remaining balance: ', v_balance - v_withdraw) AS result_msg;
    ->     END IF;
    -> END$$
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql> CALL withdraw_amount1(101, 2000);
+-----------------------------------------------+
| result_msg                                    |
+-----------------------------------------------+
| Withdrawal successful. Remaining balance: 8000.00 |
+-----------------------------------------------+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)
```

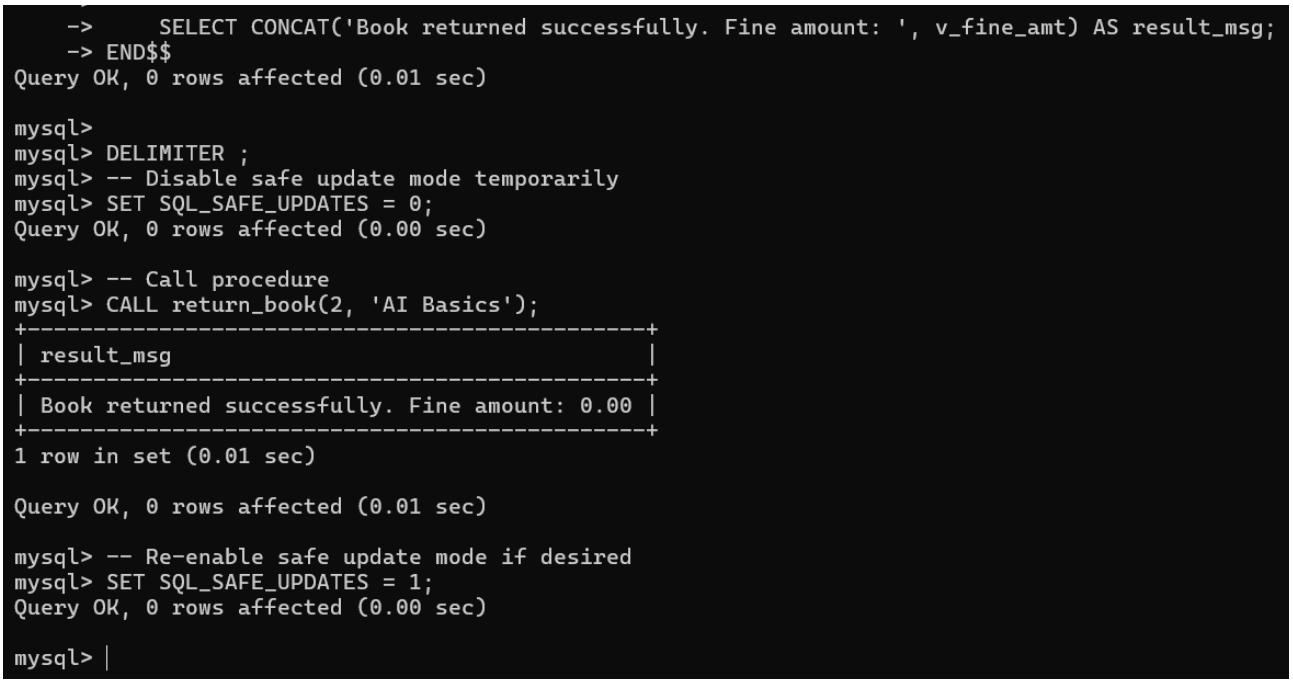
3.
1. Borrower(Roll_no, Name, Date_of_Issue, Name_of_Book, Status)
2. Fine(Roll_no, Date, Amt)
• Accept Roll_no & Name of Book from user.
• Check the number of days (from date of issue), if days are between 15 to 30 then fine amount will be Rs 5per day.
• If no. of days>30, fine will be Rs 50 per day & 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.
Also handles the exception by named exception handler or user define exception handler.



```
mysql> -- 3.Library Borrower & Fine
mysql> CREATE TABLE Borrower (
    ->     Roll_no INT,
    ->     Name VARCHAR(50),
    ->     Date_of_Issue DATE,
    ->     Name_of_Book VARCHAR(50),
    ->     Status CHAR(1)  -- 'I' = Issued, 'R' = Returned
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> CREATE TABLE Fine (
    ->     Roll_no INT,
    ->     Date_of_Fine DATE,
    ->     Amt DEC(10,2)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> INSERT INTO Borrower VALUES (1,'Sudhir',CURDATE(),'Data Science','I');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Borrower VALUES (2,'Ayyub',CURDATE(),'AI Basics','I');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Borrower VALUES (3,'Saish',CURDATE(),'Python','I');
Query OK, 1 row affected (0.01 sec)
```



```
mysql> DELIMITER $$
mysql> CREATE PROCEDURE return_book(
    ->     IN v_roll INT,
    ->     IN v_book VARCHAR(50)
    -> )
    -> BEGIN
    ->     DECLARE v_date_issue DATE;
    ->     DECLARE v_status CHAR(1);
    ->     DECLARE v_days INT;
    ->     DECLARE v_fine_amt DECIMAL(10,2);
    ->     DECLARE book_not_issued CONDITION FOR SQLSTATE '45000';
    ->
    ->     SELECT Date_of_Issue, Status INTO v_date_issue, v_status
    ->     FROM Borrower
    ->     WHERE Roll_no = v_roll AND Name_of_Book = v_book;
    ->
    ->     IF v_status <> 'I' THEN
    ->         SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Error: Book is not currently issued!';
    ->     END IF;
    ->
    ->     SET v_days = DATEDIFF(CURDATE(), v_date_issue);
    ->
    ->     IF v_days BETWEEN 15 AND 30 THEN
    ->         SET v_fine_amt = v_days * 5;
    ->     ELSEIF v_days > 30 THEN
    ->         SET v_fine_amt = v_days * 50;
    ->     ELSE
    ->         SET v_fine_amt = v_days * 5;
    ->     END IF;
    ->
    ->     -- Insert fine if applicable
    ->     IF v_fine_amt > 0 THEN
    ->         INSERT INTO Fine(Roll_no, Date_of_Fine, Amt)
    ->         VALUES(v_roll, CURDATE(), v_fine_amt);
    ->     END IF;
    ->
    ->     UPDATE Borrower
    ->     SET Status='R'
    ->     WHERE Roll_no=v_roll AND Name_of_Book=v_book;
    ->
    ->     SELECT CONCAT('Book returned successfully. Fine amount: ', v_fine_amt) AS result_msg;
```



```
    ->     SELECT CONCAT('Book returned successfully. Fine amount: ', v_fine_amt) AS result_msg;
    -> END$$
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql> -- Disable safe update mode temporarily
mysql> SET SQL_SAFE_UPDATES = 0;
Query OK, 0 rows affected (0.00 sec)

mysql> -- Call procedure
mysql> CALL return_book(2, 'AI Basics');
+-------------------------------------------+
| result_msg                                |
+-------------------------------------------+
| Book returned successfully. Fine amount: 0.00 |
+-------------------------------------------+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

mysql> -- Re-enable safe update mode if desired
mysql> SET SQL_SAFE_UPDATES = 1;
Query OK, 0 rows affected (0.00 sec)

mysql> |
```