

```

14:
CREATE TABLE STUD (Roll INT PRIMARY KEY, Att INT, Status VARCHAR(2));
INSERT INTO STUD (Roll, Att, Status) VALUES (1, 80, 'ND'), (2, 70, 'ND'), (3, 60, 'ND');
CREATE TABLE ACCOUNT_MASTER (Account_No INT PRIMARY KEY, Balance DECIMAL(10, 2));
INSERT INTO ACCOUNT_MASTER (Account_No, Balance) VALUES (101, 5000.00), (102, 3000.00), (103, 10000.00);
DELIMITER $$

CREATE PROCEDURE CheckAttendance(IN roll_no INT)
BEGIN
    DECLARE v_attendance INT;
    DECLARE v_status VARCHAR(2);
    DECLARE not_found CONDITION FOR SQLSTATE '02000'; -- Condition for no data found
    DECLARE CONTINUE HANDLER FOR not_found SET v_attendance = NULL; -- Handle no data found
    -- Retrieve the attendance and status for the given roll number
    SELECT Att, Status INTO v_attendance, v_status
    FROM STUD
    WHERE Roll = roll_no;
    -- If no data is found, display a message
    IF v_attendance IS NULL THEN
        SELECT 'No student found with the provided roll number' AS message;
    ELSE
        -- Check if attendance is less than 75% and update status
        IF v_attendance < 75 THEN
            -- If attendance is less than 75%, set status as 'D' (Term not granted)
            UPDATE STUD
            SET Status = 'D'
            WHERE Roll = roll_no;
            SELECT 'Term not granted' AS message;
        ELSE
            -- If attendance is 75% or greater, set status as 'ND' (Term granted)
            UPDATE STUD
            SET Status = 'ND'
            WHERE Roll = roll_no;
            SELECT 'Term granted' AS message;
        END IF;
    END IF;
END $$
DELIMITER ;
CALL CheckAttendance(1); -- Replace 1 with the actual roll number you want to check.

```

```

DELIMITER $$
CREATE PROCEDURE HandleWithdrawal(IN acc_no INT, IN withdraw_amt DECIMAL(10, 2))
BEGIN
    DECLARE current_balance DECIMAL(10, 2);
    -- Declare a handler for when no data is found
    DECLARE CONTINUE HANDLER FOR NOT FOUND
        SET current_balance = NULL;
    -- Retrieve current balance for the account
    SELECT Balance INTO current_balance
    FROM ACCOUNT_MASTER
    WHERE Account_No = acc_no;
    -- If no account found, raise an error
    IF current_balance IS NULL THEN

```

```

        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Account not found';
    END IF;
    -- Check if the withdrawal amount is greater than the balance
    IF withdraw_amt > current_balance THEN
        -- If withdrawal is not possible, raise an exception
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Insufficient funds';
    ELSE
        -- Perform the withdrawal and update the balance
        UPDATE ACCOUNT_MASTER
        SET Balance = Balance - withdraw_amt
        WHERE Account_No = acc_no;
        -- Output the updated balance
        SELECT Balance FROM ACCOUNT_MASTER WHERE Account_No = acc_no;
    END IF;
END $$
DELIMITER ;
CALL HandleWithdrawal(101, 2000.00); -- Replace with actual account number and
withdrawal amount

```