use triggers;

create table mymarks(sname varchar(20) ,marks int,status varchar(20));

insert into mymarks (sname,marks) values("ram",33),("kjkj",88),("kjkj",9);

select \* from mymarks;

delimiter @@

select \* from mymarks;

@@

create procedure update1(in mark1 int)

begin

declare m1 int;

set m1=mark1;

if m1>35 then

update mymarks set status='pass' where marks=m1;

else

update mymarks set status='fail' where marks=m1;

end if;

end;

@@

call update1(33);

@@

select \* from mymarks1;

@@

CREATE TABLE studentMarks (stud\_id SMALLINT(5) NOT NULL AUTO\_INCREMENT PRIMARY KEY, total\_marks INT, grade VARCHAR(5));

Query OK, 0 rows affected, 1 warning (0.05 sec)

mysql> INSERT INTO studentMarks(total\_marks, grade) VALUES(450, 'A'), (480, 'A+'), (490, 'A++'), (440, 'B+'),(400, 'C+'),(380,'C') ,(250, 'D'),(200,'E'),(100,'F'),(150,'F'),(220, 'E');

Query OK, 11 rows affected (0.01 sec)

Records: 11 Duplicates: 0 Warnings: 0

DELIMITER $$

mysql> CREATE PROCEDURE GetStudentData1()

-> BEGIN

-> SELECT \* FROM studentMarks;

-> END$$

Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;

mysql> call GetStudentData1();

+---------+-------------+-------+

| stud\_id | total\_marks | grade |

+---------+-------------+-------+

| 1 | 450 | A |

| 2 | 480 | A+ |

| 3 | 490 | A++ |

| 4 | 440 | B+ |

| 5 | 400 | C+ |

| 6 | 380 | C |

| 7 | 250 | D |

| 8 | 200 | E |

| 9 | 100 | F |

| 10 | 150 | F |

| 11 | 220 | E |

+---------+-------------+-------+

**Stored Procedure with IN Parameter:**

mysql> DELIMITER //

mysql> CREATE PROCEDURE spGetDetailsByStudentName(IN studentId INT)

-> BEGIN

-> SELECT \* FROM studentMarks where stud\_id = studentId;

-> END //

Query OK, 0 rows affected (0.03 sec)

mysql> DELIMITER ;

mysql> CALL spGetDetailsByStudentName(1);

+---------+-------------+-------+

| stud\_id | total\_marks | grade |

+---------+-------------+-------+

| 1 | 450 | A |

+---------+-------------+-------+

1 row in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

Creating Procedure with Output Parameters

DELIMITER //

CREATE PROCEDURE spGetAverageMarks(OUT average DECIMAL(5,2))

BEGIN

    SELECT AVG(total\_marks) INTO average FROM studentMarks;

END //

DELIMITER ;

mysql> CALL spGetAverageMarks(@average\_marks);

Query OK, 1 row affected, 1 warning (0.01 sec)

mysql> SELECT @average\_marks;

+----------------+

| @average\_marks |

+----------------+

| 323.64 |

+----------------+

1 row in set (0.00 sec)

#### Procedures With INOUT PARAMETERS

mysql> DELIMITER //

mysql> CREATE PROCEDURE spUpdateCounter(INOUT counter INT, IN increment INT)

-> BEGIN

-> SET counter = counter + increment;

-> END //

Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;

To pass the values to the IN parameter

mysql> SET @counter=10;

Query OK, 0 rows affected (0.00 sec)

mysql> CALL spUpdateCounter(@counter,3);

Query OK, 0 rows affected (0.00 sec)

mysql> SELECT @counter;

+----------+

| @counter |

+----------+

| 13 |

+----------+

1 row in set (0.00 sec)