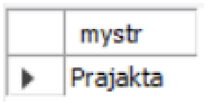
Assignment 4:

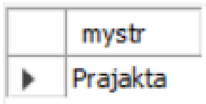
Aim: To implement the stored procedures and functions

1. Write a procedure to print your name N-number of times

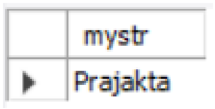
```
DELIMITER //
create procedure printMyName(IN mystr varchar(50),IN i INT)
begin
    declare counter int;
    set counter=1;
    while counter<=i  do
       select mystr;
   set counter=counter+1;
   end while;
end
//
DELIMITER ;
call printMyName("Prajakta",5);
```



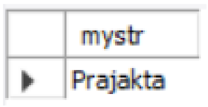
	mystr
▶	Prajakta



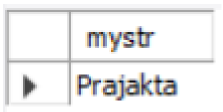
	mystr
▶	Prajakta



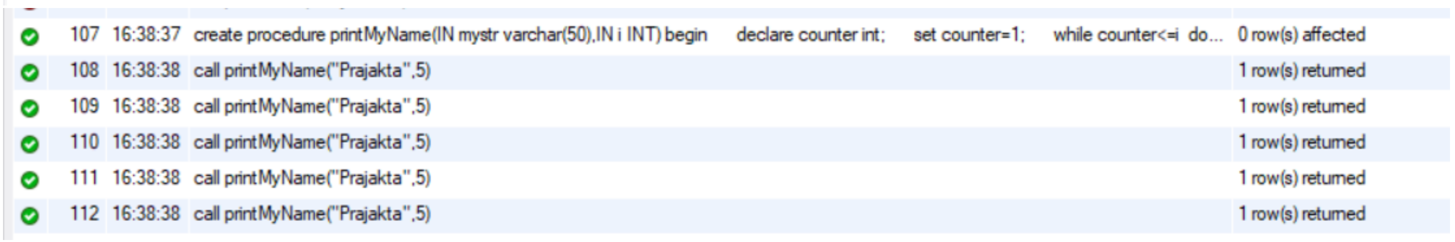
	mystr
▶	Prajakta



	mystr
▶	Prajakta



	mystr
▶	Prajakta

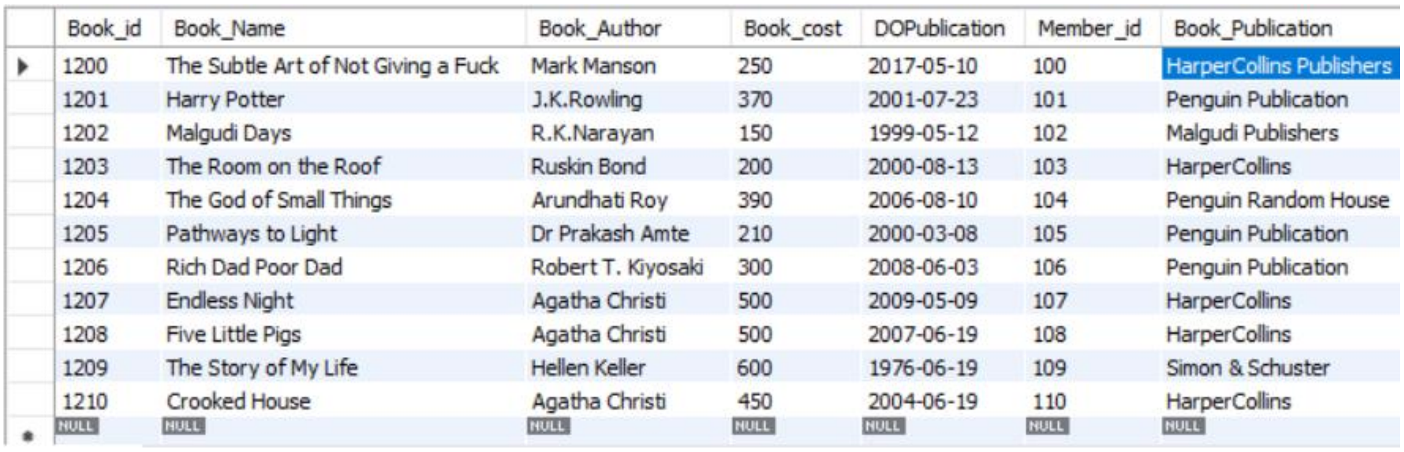


```
107  16:38:37  create procedure printMyName(IN mystr varchar(50),IN i INT) begin    declare counter int;    set counter=1;    while counter<=i  do...  0 row(s) affected
108  16:38:38  call printMyName("Prajakta",5)                                                                                          1 row(s) returned
109  16:38:38  call printMyName("Prajakta",5)                                                                                          1 row(s) returned
110  16:38:38  call printMyName("Prajakta",5)                                                                                          1 row(s) returned
111  16:38:38  call printMyName("Prajakta",5)                                                                                          1 row(s) returned
112  16:38:38  call printMyName("Prajakta",5)                                                                                          1 row(s) returned
```

2. Write a procedure to update the cost of the given book.

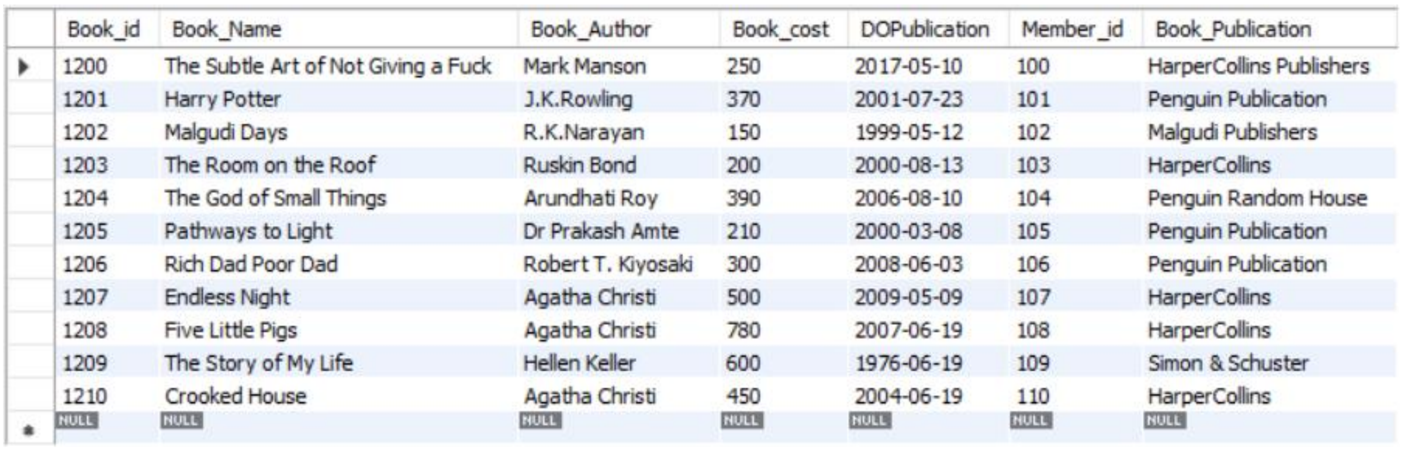
(bookid and newCost are input parameters). Use exception handling.

```
DELIMITER $$

CREATE PROCEDURE updateCosts(IN id INT,IN newCost INT)

begin

UPDATE Books

SET Book_cost=newCost where Book_id=id;

End $$

DELIMITER ;

SELECT * FROM Books;
```



Book_id	Book_Name	Book_Author	Book_cost	DOPublication	Member_id	Book_Publication
1200	The Subtle Art of Not Giving a Fuck	Mark Manson	250	2017-05-10	100	HarperCollins Publishers
1201	Harry Potter	J.K.Rowling	370	2001-07-23	101	Penguin Publication
1202	Malgudi Days	R.K.Narayan	150	1999-05-12	102	Malgudi Publishers
1203	The Room on the Roof	Ruskin Bond	200	2000-08-13	103	HarperCollins
1204	The God of Small Things	Arundhati Roy	390	2006-08-10	104	Penguin Random House
1205	Pathways to Light	Dr Prakash Amte	210	2000-03-08	105	Penguin Publication
1206	Rich Dad Poor Dad	Robert T. Kiyosaki	300	2008-06-03	106	Penguin Publication
1207	Endless Night	Agatha Christi	500	2009-05-09	107	HarperCollins
1208	Five Little Pigs	Agatha Christi	500	2007-06-19	108	HarperCollins
1209	The Story of My Life	Hellen Keller	600	1976-06-19	109	Simon & Schuster
1210	Crooked House	Agatha Christi	450	2004-06-19	110	HarperCollins
NULL	NULL	NULL	NULL	NULL	NULL	NULL

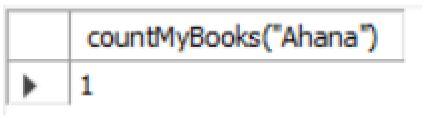
```
call updateCosts(1208,780);

SELECT * FROM Books;
```



Book_id	Book_Name	Book_Author	Book_cost	DOPublication	Member_id	Book_Publication
1200	The Subtle Art of Not Giving a Fuck	Mark Manson	250	2017-05-10	100	HarperCollins Publishers
1201	Harry Potter	J.K.Rowling	370	2001-07-23	101	Penguin Publication
1202	Malgudi Days	R.K.Narayan	150	1999-05-12	102	Malgudi Publishers
1203	The Room on the Roof	Ruskin Bond	200	2000-08-13	103	HarperCollins
1204	The God of Small Things	Arundhati Roy	390	2006-08-10	104	Penguin Random House
1205	Pathways to Light	Dr Prakash Amte	210	2000-03-08	105	Penguin Publication
1206	Rich Dad Poor Dad	Robert T. Kiyosaki	300	2008-06-03	106	Penguin Publication
1207	Endless Night	Agatha Christi	500	2009-05-09	107	HarperCollins
1208	Five Little Pigs	Agatha Christi	780	2007-06-19	108	HarperCollins
1209	The Story of My Life	Hellen Keller	600	1976-06-19	109	Simon & Schuster
1210	Crooked House	Agatha Christi	450	2004-06-19	110	HarperCollins
NULL	NULL	NULL	NULL	NULL	NULL	NULL

3. Write a function which accepts the Member_Name and returns the number of books issued by him/ her.

```
DELIMITER $$
create function countMyBooks(MemName VARCHAR(40)) returns int
READS SQL DATA
BEGIN
  DECLARE cnt int default 0;
  SELECT COUNT(*) into cnt FROM issues WHERE Member_id=(SELECT Member_id FROM
Members where Member_Name=MemName);
  return cnt;
END
$$
DELIMITER ;
SELECT countMyBooks("Ahana");
```



```
countMyBooks("Ahana")
1
```

4. Consider the following tables:

Books(BNo, Bname, Publisher, cost, DOP, status)

Issues(Mem_ID, BNo, Issue_Dt, Return_Dt, fine)

Write a procedure Book_Return which accepts the mem_ID, BNo and Issue_Dt as input parameters and performs the book-return operation by resetting the status of the book and calculating the fine. Use exception handling.

The procedure should calculate a fine as follows:

Check the number of days (from date of issue),

for days< 15, Rs. 5 per day.

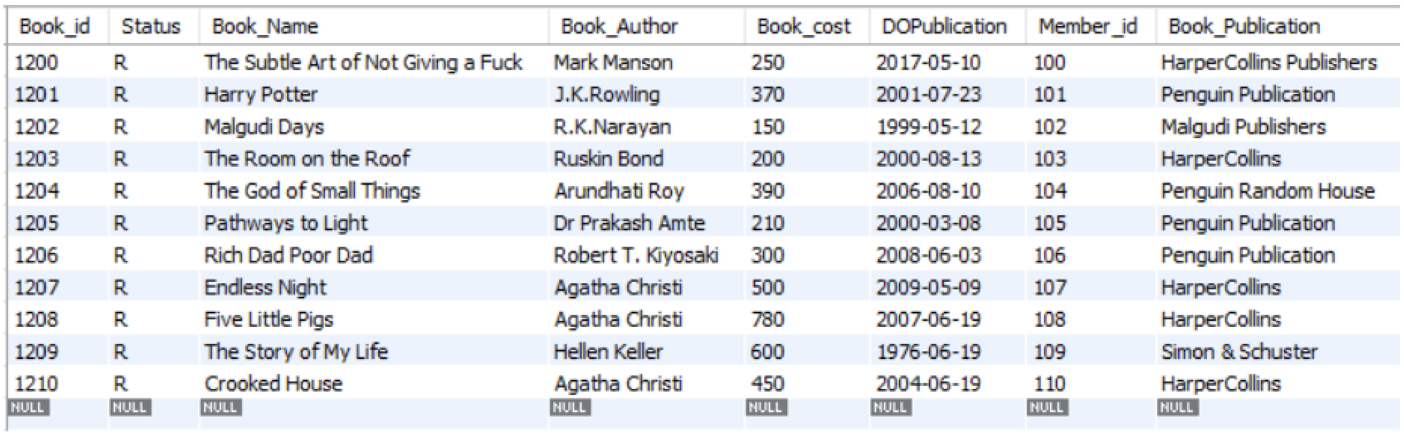
- If days are between 15 to 30 then fine amount will be Rs 15 per day.

- If no. of days>30, per day fine will be Rs 50 per day

After submitting the book, status will change from I to R.

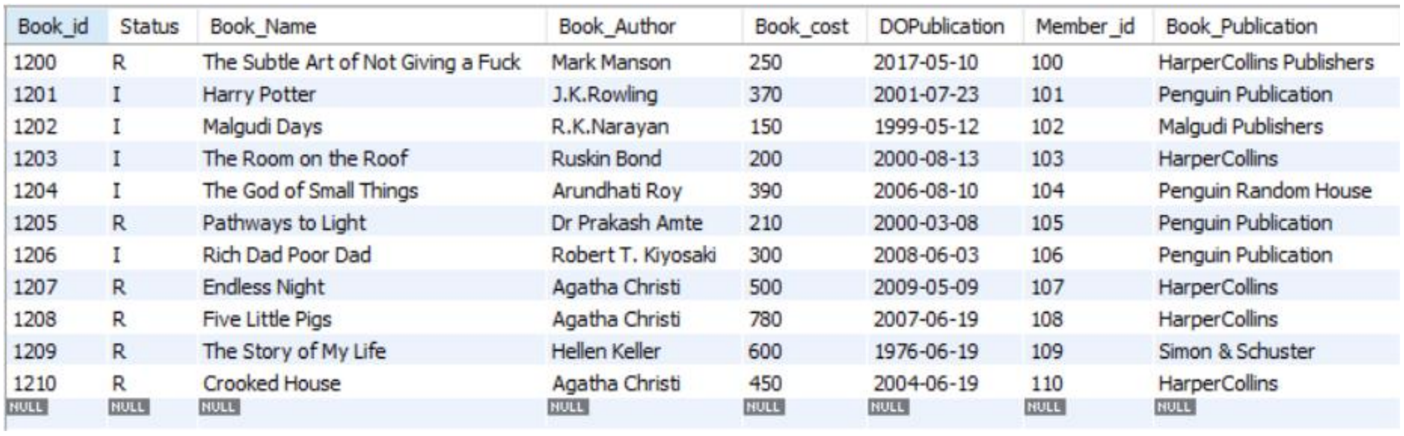
Updating issues and books tables for executing procedure

```
ALTER TABLE Books

ADD COLUMN Status VARCHAR(2) DEFAULT 'R' AFTER Book_id;
```



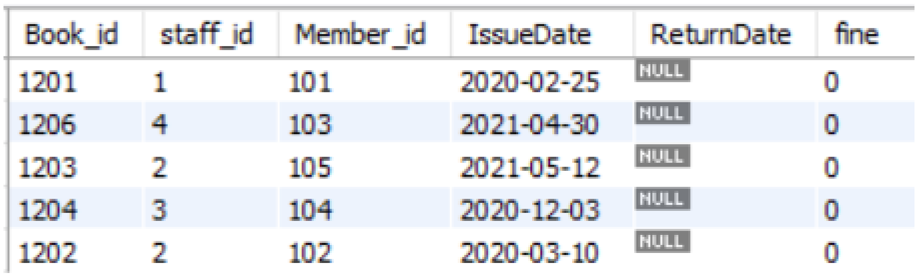
Book_id	Status	Book_Name	Book_Author	Book_cost	DOPublication	Member_id	Book_Publication
1200	R	The Subtle Art of Not Giving a Fuck	Mark Manson	250	2017-05-10	100	HarperCollins Publishers
1201	R	Harry Potter	J.K.Rowling	370	2001-07-23	101	Penguin Publication
1202	R	Malgudi Days	R.K.Narayan	150	1999-05-12	102	Malgudi Publishers
1203	R	The Room on the Roof	Ruskin Bond	200	2000-08-13	103	HarperCollins
1204	R	The God of Small Things	Arundhati Roy	390	2006-08-10	104	Penguin Random House
1205	R	Pathways to Light	Dr Prakash Amte	210	2000-03-08	105	Penguin Publication
1206	R	Rich Dad Poor Dad	Robert T. Kiyosaki	300	2008-06-03	106	Penguin Publication
1207	R	Endless Night	Agatha Christi	500	2009-05-09	107	HarperCollins
1208	R	Five Little Pigs	Agatha Christi	780	2007-06-19	108	HarperCollins
1209	R	The Story of My Life	Hellen Keller	600	1976-06-19	109	Simon & Schuster
1210	R	Crooked House	Agatha Christi	450	2004-06-19	110	HarperCollins
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
UPDATE Books SET Status='I' WHERE Book_id in (1201,1203,1206,1202,1204);

select * from Books;
```



Book_id	Status	Book_Name	Book_Author	Book_cost	DOPublication	Member_id	Book_Publication
1200	R	The Subtle Art of Not Giving a Fuck	Mark Manson	250	2017-05-10	100	HarperCollins Publishers
1201	I	Harry Potter	J.K.Rowling	370	2001-07-23	101	Penguin Publication
1202	I	Malgudi Days	R.K.Narayan	150	1999-05-12	102	Malgudi Publishers
1203	I	The Room on the Roof	Ruskin Bond	200	2000-08-13	103	HarperCollins
1204	I	The God of Small Things	Arundhati Roy	390	2006-08-10	104	Penguin Random House
1205	R	Pathways to Light	Dr Prakash Amte	210	2000-03-08	105	Penguin Publication
1206	I	Rich Dad Poor Dad	Robert T. Kiyosaki	300	2008-06-03	106	Penguin Publication
1207	R	Endless Night	Agatha Christi	500	2009-05-09	107	HarperCollins
1208	R	Five Little Pigs	Agatha Christi	780	2007-06-19	108	HarperCollins
1209	R	The Story of My Life	Hellen Keller	600	1976-06-19	109	Simon & Schuster
1210	R	Crooked House	Agatha Christi	450	2004-06-19	110	HarperCollins
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

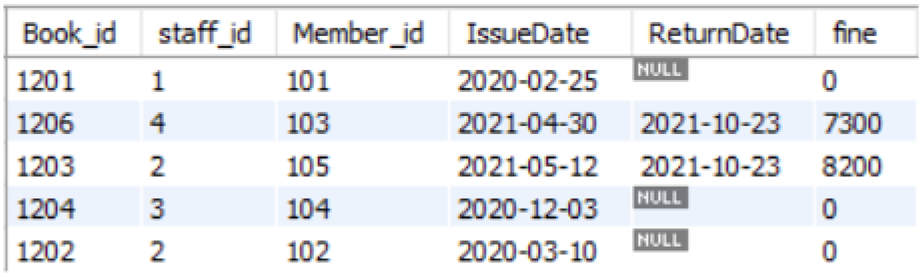
```
UPDATE issues

SET ReturnDate=NULL;

UPDATE issues

set fine=0;

SELECT * FROM issues;
```



Book_id	staff_id	Member_id	IssueDate	ReturnDate	fine
1201	1	101	2020-02-25	NULL	0
1206	4	103	2021-04-30	NULL	0
1203	2	105	2021-05-12	NULL	0
1204	3	104	2020-12-03	NULL	0
1202	2	102	2020-03-10	NULL	0

```
DELIMITER $$
create Procedure Book_Returns(IN M_id INT,IN B_id INT,IN IssueDate DATE)
BEGIN
   DECLARE fineAmount INT default 0;
   DECLARE ddiff INT;
   SELECT datediff(Curdate(),IssueDate) into ddiff;

   IF ddiff>30 THEN
   set fineAmount=50*ddiff;
   END IF;

   IF ddiff>=15 and ddiff<=30 THEN
   set fineAmount=15*ddiff;
   END IF;

   IF ddiff<15 THEN
   set fineAmount=5*ddiff;
   END IF;

   UPDATE Issues
   SET fine=fineAmount, ReturnDate=curdate()
   WHERE Book_id=B_id AND Member_id=M_id;

  UPDATE Books set status="R" WHERE Book_id=B_id;
END
$$
DELIMITER ;
call Book_Returns(1203,105,'2021-05-12');
call Book_Returns(1206,103,'2021-04-30');
```

SELECT * FROM Issues;



Book_id	staff_id	Member_id	IssueDate	ReturnDate	fine
1201	1	101	2020-02-25	NULL	0
1206	4	103	2021-04-30	2021-10-23	7300
1203	2	105	2021-05-12	2021-10-23	8200
1204	3	104	2020-12-03	NULL	0
1202	2	102	2020-03-10	NULL	0

5. Create table Result(RNo, Sname,tot_marks, class)

Populate the table with total marks out of 1500. Put NULL in the class field.

Write a stored procedure to fill the class of each student as follows:

class = Distinction if marks<1500 and marks>990

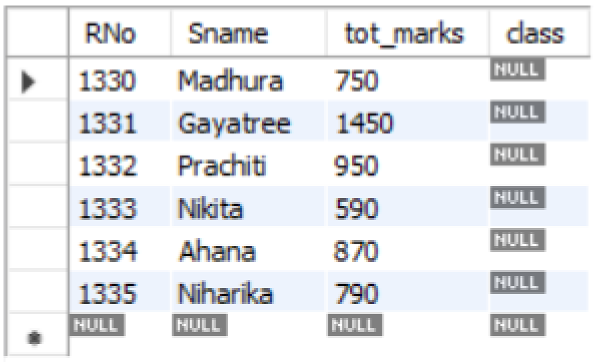
class = First Class if marks<990 and marks>900

class = Higher second class if marks<900 and marks>825

class = Second class if marks<825 and marks>600

If marks<600 then class = Fail

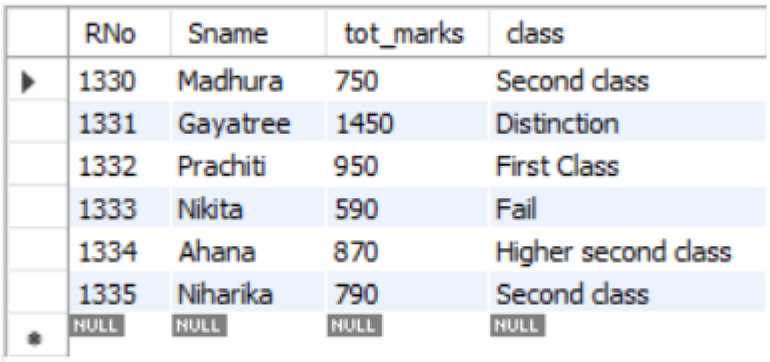
```
CREATE table Result(
RNo INT PRIMARY KEY,
Sname VARCHAR(50),
tot_marks INT,
class VARCHAR(35) DEFAULT NULL
);
INSERT INTO Result(RNo,Sname,tot_marks,class)
VALUES (1330,"Madhura",750,NULL),
    (1331,"Gayatree",1450,NULL),
    (1332,"Prachiti",950,NULL),
    (1333,"Nikita",590,NULL),
    (1334,"Ahana",870,NULL),
    (1335,"Niharika",790,NULL);
SELECT * FROM Result;
```



RNo	Sname	tot_marks	class
1330	Madhura	750	NULL
1331	Gayatree	1450	NULL
1332	Prachiti	950	NULL
1333	Nikita	590	NULL
1334	Ahana	870	NULL
1335	Niharika	790	NULL
NULL	NULL	NULL	NULL

```
DELIMITER $$
create procedure assignClasses()
begin
    DECLARE the_class varchar(35);
    DECLARE the_rno INT;
    DECLARE the_marks INT;
    declare finished boolean default false;

    DECLARE cur CURSOR FOR SELECT tot_marks,RNo FROM Result;
    declare continue handler for not found SET finished=true;

    open cur;
       the_loop:LOOP
           FETCH cur INTO the_marks,the_rno;

           if finished=true then
           leave the_loop;
           end if;

           if the_marks<600 THEN
           SET the_class="Fail";
           END if;
```

```
        if the_marks>=600 AND the_marks<=825 THEN
        SET the_class="Second class";
        END if;

        if the_marks>825 AND the_marks<=900 THEN
        SET the_class="Higher second class";
        END if;

        if the_marks>900 AND the_marks<=990 THEN
        SET the_class="First Class";
        END if;

        if the_marks>990 AND the_marks<=1500 THEN
        SET the_class="Distinction";
        END if;

        UPDATE Result SET class=the_class WHERE the_rno=RNo;
    end LOOP;
    close cur;
end $$
DELIMITER ;

call assignClasses();
SELECT * FROM Result;
```



RNo	Sname	tot_marks	class
1330	Madhura	750	Second class
1331	Gayatree	1450	Distinction
1332	Prachiti	950	First Class
1333	Nikita	590	Fail
1334	Ahana	870	Higher second class
1335	Niharika	790	Second class
NULL	NULL	NULL	NULL