1.Using this procedure, we can determine the students are from the same area.

Parameter: city\_name

drop procedure if exists citywise\_students;

delimiter $$

create procedure citywise\_students

(

in city varchar(25)

)

begin

set @student\_city = city;

set @regex = concat( @student\_city, '$');

select \*

from student

where address regexp @regex;

end $$

delimiter ;

1. In this procedure we will find that on a particular date which are those students which are present in both half.

Parameter attendance\_date

drop procedure if exists attendance\_record;

delimiter $$

create procedure attendance\_record

(

in attendance\_date varchar(25)

)

begin

set @at\_date = attendance\_date;

set @regex = concat(@at\_date, '%');

select s.name, s.rollno

from student s

join attendance\_student ast

using (rollno)

where ast.first = 'Present' and ast.second = 'Present' and ast.Date like @regex;

end $$

delimiter ;

3.Using this procedure, we can determine the students’ name and roll no. Which are absent in particular half

Parameter: attendance\_date,half(particular)

drop procedure if exists attendance\_absent;

delimiter $$

create procedure attendance\_absent

(

in attendance\_date varchar(25),

in half varchar(10)

)

begin

set @at\_date = attendance\_date;

set @regex = concat(@at\_date, '%');

set @period = half;

IF @period = 'first' THEN

select s.name, s.rollno

from student s

join attendance\_student ast

using (rollno)

where ast.Date like @regex and ast.first = 'absent';

ELSE

select s.name, s.rollno

from student s

join attendance\_student ast

using (rollno)

where ast.Date like @regex and ast.second = 'absent';

END IF;

end $$

delimiter ;

1. Using this procedure, we can determine the teachers’ name and Emp\_id who is on leave on particular date

Parameter: attendance\_date,attendance(leave)

drop procedure if exists attendance\_teacher;

delimiter $$

create procedure attendance\_teacher

(

in attendance\_date varchar(25),

in attendence varchar(10)

)

begin

set @at\_date = attendance\_date;

set @regex = concat(@at\_date, '%');

select t.emp\_id, t.name

from teacher t

join attendance\_teacher att

using (emp\_id)

where att.Date like @regex and att.first = 'leave' and att.second = 'leave' ;

end $$

delimiter ;

5.Using this procedure, we can delete the particular student record from the database.

Parameter: roll\_no

DROP PROCEDURE IF EXISTS operation\_delete;

DELIMITER $$

CREATE PROCEDURE operation\_delete (IN roll\_no INT)

BEGIN

DELETE FROM student

WHERE rollno = roll\_no;

SELECT \* FROM student;

END $$

DELIMITER ;

6.Using this procedure, we will find students’ name and its total marks of all the subjects

Parameter: roll\_no

delimiter $$

create procedure calculate\_marks

(

in roll\_no varchar(10)

)

begin

select s.rollno, s.name, (marks1 + marks2 + marks3 + marks4 + marks5) as total

from student s

join marks m

using (rollno)

where m.rollno = roll\_no;

end $$

delimiter ;

7.Using this procedure, we will first find percentage of any particular student and based on that we will give remarks to them

Parameter: roll\_no

drop procedure if exists remarks;

DELIMITER $$

CREATE PROCEDURE remarks

(

IN roll\_no varchar(10)

)

BEGIN

set @percentage = (select (marks1 + marks2 + marks3 + marks4 + marks5)/5 from marks where rollno = roll\_no);

select s.rollno, s.name, @percentage as percentage,

case

when @percentage > 90 then

"Excellent"

when @percentage > 80 and @percentage <= 90 then

"Very Good"

when @percentage > 70 and @percentage <= 80 then

"Good"

when @percentage > 60 and @percentage <= 70 then

"Moderate"

when @percentage > 50 and @percentage <= 60 then

"Fair"

when @percentage > 40 and @percentage <= 50 then

"Satisfactory"

when @percentage > 30 and @percentage <= 40 then

"Grace Pass"

else

"Fail"

end as Remark

from student s

join marks m

using (rollno)

where m.rollno = roll\_no;

end $$

DELIMITER ;

1. Using this procedure, we can determine that which student should get how much scholership based on theiraverage of class 10 and class 12th result

Parameter: roll\_no

drop procedure if exists scholarship\_detail;

DELIMITER $$

CREATE PROCEDURE scholarship\_detail

(

IN roll\_no varchar(10)

)

BEGIN

set @Average = (select (class\_x + class\_xii)/2 from student where rollno = roll\_no);

select s.rollno, s.name, @Average as Average,

case

when @Average > 90 then "90% Scholarship"

when @Average > 85 then "75% Scholarship"

when @Average > 80 then "60% Scholarship"

when @Average > 75 then "40% Scholarship"

when @Average > 70 then "20% Scholarship"

else "Sorry! No Scholarship can be funded"

end as Scholarship

from student s

where rollno = roll\_no;

end $$

DELIMITER ;

9)for inserting student record

drop procedure if exists insert\_student;

DELIMITER $$

CREATE PROCEDURE insert\_student

(

IN name varchar(20),

IN fathers\_name varchar(20),

IN age varchar(5),

IN dob varchar(20),

IN address varchar(30),

IN phone varchar(10),

IN email varchar(30),

IN class\_x varchar(10),

IN class\_xii varchar(10),

IN aadhar varchar(20),

IN rollno varchar(15),

IN course varchar(10),

IN branch varchar(20)

)

BEGIN

insert into student

values(name, fathers\_name, age, dob, address, phone, email, class\_x, class\_xii, aadhar, rollno, course, branch);

end $$

DELIMITER ;

10)For inserting teacher record

drop procedure if exists insert\_teacher;

DELIMITER $$

CREATE PROCEDURE insert\_teacher

(

IN name varchar(20),

IN fathers\_name varchar(20),

IN age varchar(5),

IN dob varchar(20),

IN address varchar(30),

IN phone varchar(10),

IN email varchar(30),

IN class\_x varchar(10),

IN class\_xii varchar(10),

IN aadhar varchar(20),

IN course varchar(15),

IN emp\_id varchar(10),

IN dept varchar(20)

)

BEGIN

insert into teacher

values(name, fathers\_name, age, dob, address, phone, email, class\_x, class\_xii, aadhar, course, emp\_id, dept);

end $$

DELIMITER ;

**Triggers**

1. It Update student detail in whole database for same record

delimiter $$

create trigger student\_after\_update

before update on student

for each row

begin

Update fee f

set f.name = new.name

where rollno = new.rollno;

end $$

delimiter ;

update student

set name = 'dixit'

where rollno = '15331902'

2)To check mobile length of mobile number

delimiter $$

drop trigger if exists student\_before\_update;

create trigger student\_before\_update

before update on student

for each row

begin

DECLARE msg VARCHAR(100);

DECLARE numLength INT;

SET numLength = (SELECT CHAR\_LENGTH(NEW.Phone));

IF (numLength > 10) THEN

set msg = concat('MyTriggerError: Trying to update invalid phone number: ',cast(new.phone as char));

SIGNAL SQLSTATE '45000' set message\_text = msg;

END IF;

end $$

delimiter ;

3)To delete a student record from whole table

delimiter $$

create trigger student\_after\_delete

after delete on student

for each row

begin

delete from fee f

where f.rollno = old.rollno;

delete from attendance\_student ast

where ast.rollno = old.rollno;

end $$

delimiter ;

4) To delete a teacher record from whole table

delimiter $$

create trigger teacher\_after\_delete

after delete on teacher

for each row

begin

delete from attendance\_teacher

where emp\_id = old.emp\_id;

end $$

delimiter ;