**Procedure**

**🡪Insert Student Data After Checking All the Validation like email ID, Phone No.**

DELIMITER $$

create procedure StuData(IN first\_name varchar(10),IN middle\_name varchar(10),IN last\_name varchar(10),IN email\_id varchar(20),IN age decimal(2,0),IN Contact\_Number decimal(10,0),IN event\_name varchar(20),IN pg\_name varchar(20),IN re\_name varchar(20),IN University\_name varchar(30),IN degree varchar(10),IN branch varchar(10),IN sem decimal(1,0),IN gender varchar(10))

begin

DECLARE Counter varchar(4);

select count(s\_id) into Counter from student;

set counter = counter + 1;

IF email\_id like '%@gmail.com' THEN

IF length(contact\_number) <=> 10 THEN

IF (select count(University\_name) from University u where u.university\_name = University\_name) then

SELECT 'University Found' AS MESSAGE;

INSERT into student VALUES (first\_name,middle\_name,last\_name,email\_id,counter,age,contact\_number,event\_name,pg\_name,re\_name,University\_name,degree,branch,sem,gender);

SELECT 'YOUR DATA IS INSERTED' AS MESSAGE;

ELSE

SELECT 'Enter Valid Name' AS ERROR;

END IF;

ELSE

SELECT 'Your Mobile No. is not valid' AS ERROR;

END IF;

ELSE

SELECT 'YOUR Email-ID is not valid' AS ERROR;

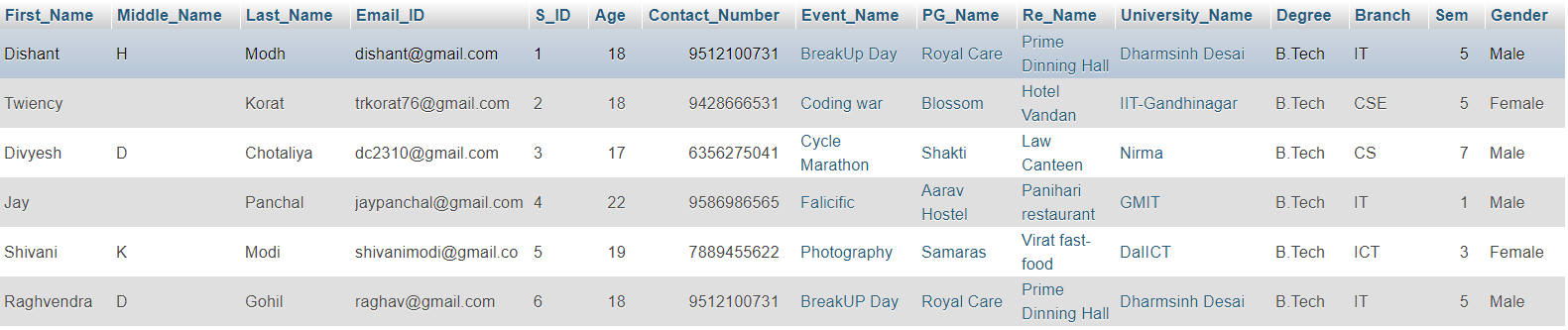
END IF;

END $$

DELIMITER ;

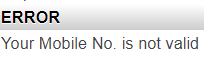
Call StuData('Raghvendra','D','Gohil','raghav@gmail.com',18,9512100731,'BreakUP Day','Royal Care','Prime Dinning Hall','Dharmsinh Desai','B.Tech','IT',5,'Male');

* In the Student one row is Inserted.



Call StuData('Raghvendra','D','Gohil','raghav@gmail.com',18,9512100,'BreakUP Day','Royal Care','Prime Dinning Hall','Dharmsinh Desai','B.Tech','IT',5,'Male');

🡪As from above query Number Format is not proper so, it will give ERROR.



**🡪Show the number of Student in particular University.**

**FUNCTION**

DELIMITER $$

CREATE FUNCTION no\_of\_student(University\_Name varchar(30)) RETURNS int DETERMINISTIC

BEGIN

DECLARE no int;

SELECT COUNT(S\_ID) INTO no FROM student s WHERE s.University\_Name=University\_Name;

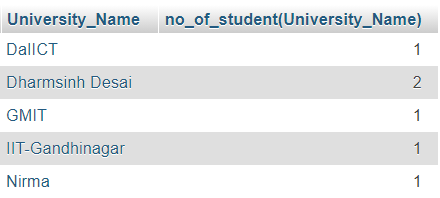
RETURN no;

END

$$

DELIMITER ;

SELECT University\_Name, no\_of\_student(University\_Name) FROM student GROUP BY University\_Name;



**🡪Show Name Of all Restaurant’s.**

**CURSOR**

DELIMITER $$

CREATE PROCEDURE r1 (INOUT Name\_List VARCHAR(1000))

BEGIN

DECLARE done INTEGER DEFAULT 0;

DECLARE r\_name VARCHAR(20) DEFAULT "";

DECLARE r\_cursor CURSOR FOR

SELECT Re\_Name FROM resturant;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

OPEN r\_cursor;

list:LOOP

FETCH r\_cursor INTO r\_name;

IF done = 1 THEN

LEAVE list;

END IF;

SET Name\_List = CONCAT(r\_name, "-->",Name\_List);

END LOOP list;

CLOSE r\_cursor;

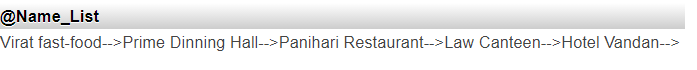
END$$

DELIMITER ;

SET @Name\_List="";

CALL r1(@Name\_List);

SELECT @Name\_List;



**Trigger**

**🡪Update Event name into Event table and Organize table using Trigger.**

DELIMITER $$

CREATE TRIGGER e1 AFTER UPDATE ON events FOR EACH ROW

BEGIN

DECLARE v1 VARCHAR(20);

SELECT Event\_Name INTO v1 FROM events WHERE Event\_Name=NEW.Event\_Name;

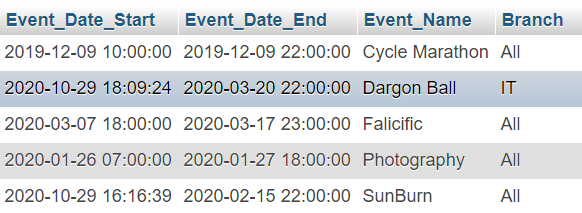
UPDATE `oraganize` SET `Event\_Name`=v1 WHERE Event\_Name=OLD.Event\_Name;

END $$

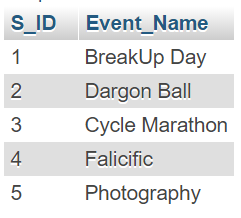
DELIMITER ;

UPDATE `events` SET `Event\_Name`="Dargon Ball" WHERE Event\_Name="Coding war";

* After Update on Event Table.



* Updated by Trigger on Organize table.



**🡪Delete Community Name from Community Table and Member table using Trigger.**

DELIMITER $$

CREATE TRIGGER c1 AFTER DELETE ON community FOR EACH ROW

BEGIN

DECLARE v1 VARCHAR(20);

SELECT Community\_Name INTO v1 FROM community WHERE Community\_Name\_Name=NEW.Community\_Name;

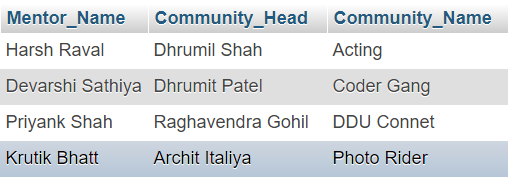
DELETE FROM `member` WHERE Community\_Name=OLD.Community\_Name;

END $$

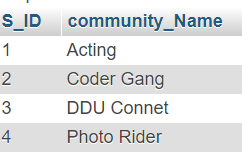
DELIMITER ;

DELETE FROM `community` WHERE Community\_Name="DSC";

* After Deleting “DSC” from community table



* Deleted “DSC” from Member table using DELETE Trigger



Select \* from `student` where `First\_Name` like "D%";

SELECT `First\_Name`, `Middle\_Name`, `Last\_Name`,`community\_Name` FROM `student` RIGHT JOIN `member` ON student.S\_ID=member.S\_ID WHERE 1

SELECT `First\_Name`,`Middle\_Name`,`Last\_Name` FROM `university` INNER JOIN `student` ON university.University\_Name=student.University\_Name WHERE `Location`="Nadiad"

SELECT University\_Name , COUNT(Courses\_Name) AS No\_Of\_Courses FROM courses GROUP BY University\_Name ORDER BY No\_Of\_Courses DESC;