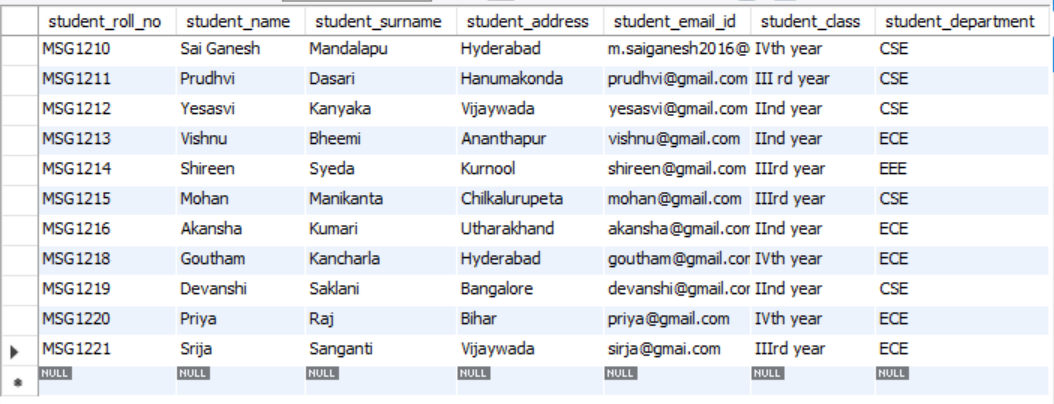
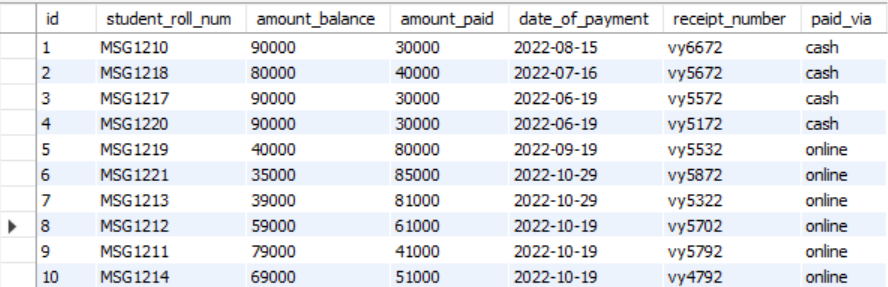
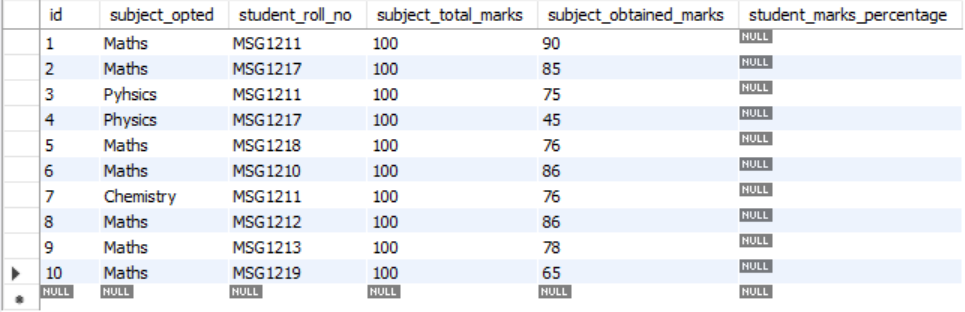
**Student Basic Information Table:**



**Student Admission Details:**



**Student Subject Information:**

****

**Student Scholarship Information:**

****

**5) Update any 5 entries**

UPDATE `student`.`student\_scholarship\_information` SET `scholarship\_description` = '3 year scholarship' WHERE (`id` = '8');

UPDATE `student`.`student\_scholarship\_information` SET `scholarship\_name` = 'SS12', `scholarship\_description` = '2 year scholarship' WHERE (`id` = '9');

UPDATE `student`.`student\_scholarship\_information` SET `scholarship\_description` = '1 year scholarship' WHERE (`id` = '1');

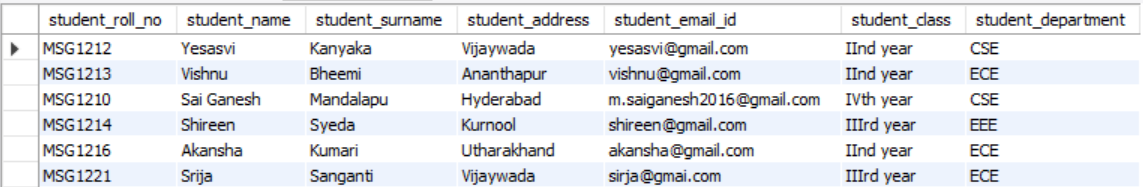
UPDATE `student`.`student\_scholarship\_information` SET `scholarship\_by` = 'Sri Ganesh' WHERE (`id` = '3');

UPDATE `student`.`student\_scholarship\_information` SET `scholarship\_name` = 'KhelVidhya' WHERE (`id` = '11');



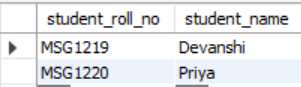
**7) Select the student details records who has received the scholarship more than 5000Rs**

select \* from student.student\_basic\_information where student\_basic\_information.student\_roll\_no in (select student\_roll\_number from student.student\_scholarship\_information where student\_scholarship\_information.scholarship\_amount>5000);

****

**8) Select the students who opted for scholarship but has not got the scholarship**

select student\_roll\_no,student\_name from student.student\_basic\_information where student\_roll\_no not in (select student\_roll\_number from student.student\_scholarship\_information);



**9)** Fill in data for the percentage column i.e. StudentMarksPercentage in the table StudentSubjectInformation by creating and using the stored procedure created

DELIMITER $$

CREATE

PROCEDURE `student`.`mark\_percentage`()

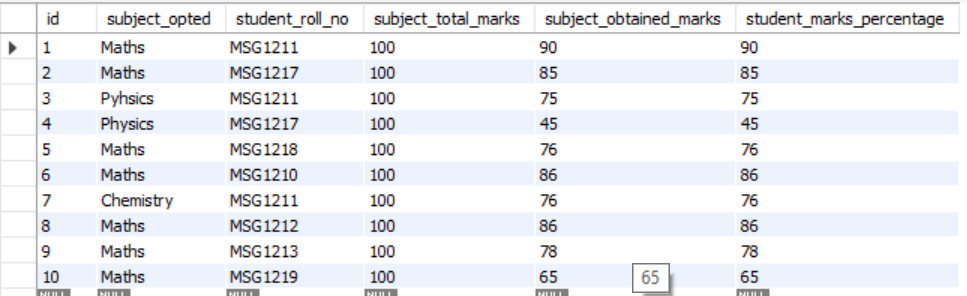
BEGIN

update student\_subject\_information set student\_marks\_percentage = (subject\_obtained\_marks/subject\_total\_marks)\*100;

END $$

DELIMITER ;

Call mark\_percentage();



10) Decide the category of the scholarship depending upon the marks/percentage obtained by the student and likewise update the ScholarshipCategory column, create a stored procedure in order to handle this operation

DELIMITER $$

CREATE

PROCEDURE `student`.`set\_category`()

BEGIN

update student\_scholarship\_information set scholarship\_category= "Merrit" where student\_roll\_number in (select student\_roll\_no from student\_subject\_information where student\_marks\_percentage > 80 );

END $$

DELIMITER ;

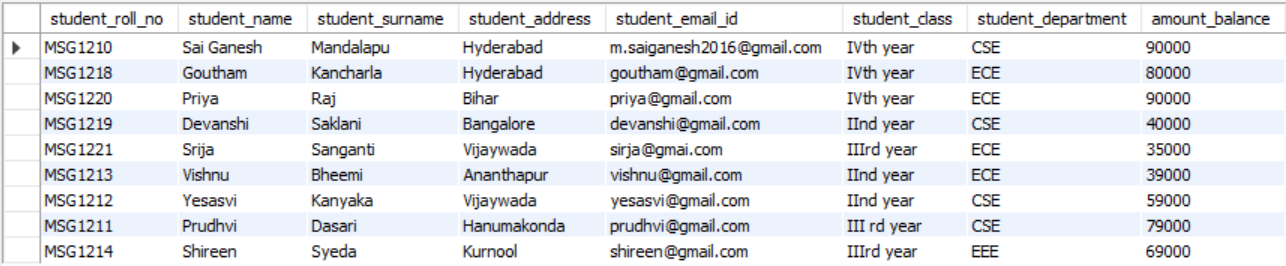
Call set\_category;

11) Create the View which shows balance amount to be paid by the student along with the student detailed information (use join)

create view `student`.`show\_balance`

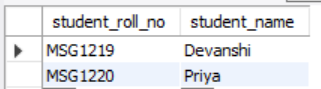
as

select student\_roll\_no, student\_name, student\_surname, student\_address, student\_email\_id, student\_class, student\_department,amount\_balance from student\_basic\_information join student\_admission\_payment\_details on student\_roll\_no=student\_roll\_num;



12) Get the details of the students who haven’t got any scholarship (use joins/subqueries)

SELECT student\_roll\_no,student\_name FROM student.student\_basic\_information where student\_roll\_no not in (select student\_roll\_number from student.student\_scholarship\_information);



13) Create Stored Procedure which will be return the amount balance to be paid by the student as per the student roll number passed through the stored procedure as the input

DELIMITER $$

drop procedure if exists `student`.`balance`;

CREATE

Procedure

`student`.`balance` (in student\_number varchar(40))

begin

select student\_roll\_no ,student\_name,amount\_balance from student\_admission\_payment\_details join student\_basic\_information on student\_roll\_num=student\_roll\_no where student\_roll\_no=student\_number;

end $$

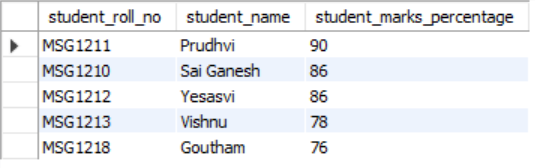
DELIMITER ;

call balance("MSG1210");



14) Retrieve the top five student details as per the StudentMarksPercentage values (use subqueries)

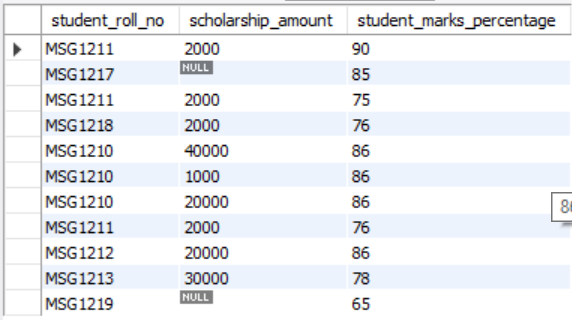
select s.student\_roll\_no,s.student\_name , d.student\_marks\_percentage from student.student\_basic\_information as s, student\_subject\_information as d where s.student\_roll\_no=d.student\_roll\_no and d.student\_roll\_no in (select k.student\_roll\_no from student.student\_subject\_information as k) order by d.student\_marks\_percentage desc limit 5;



15) Try to use all the three types of join learned today in a relevant way, and explain the same why you thought of using that particular join for your selected scenarios (try to cover relevant and real time scenarios for all the three studied joins)

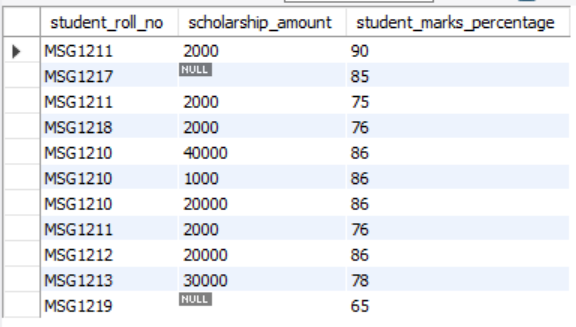
Scenario 1: *Students who have student\_marks\_percentage more than 60 but still not having any scholarship amount.*

select s.student\_roll\_no , d.scholarship\_amount, s.student\_marks\_percentage from student\_subject\_information as s left outer join student\_scholarship\_information as d on s.student\_roll\_no=d.student\_roll\_number where s.student\_marks\_percentage>60;



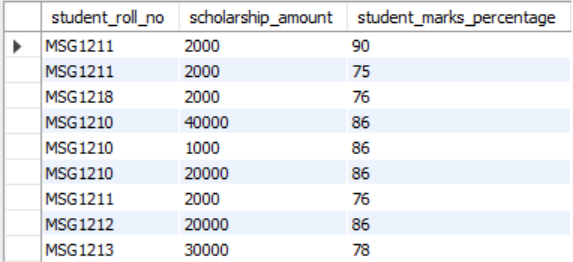
Scenario 2: *Students who have scholarship but not enough student marks percentage*

select s.student\_roll\_no , d.scholarship\_amount, s.student\_marks\_percentage from student\_scholarship\_information as d right outer join student\_subject\_information as s on s.student\_roll\_no=d.student\_roll\_number where s.student\_marks\_percentage>60;



Scenario 3: Students who have both scholarship and marks percentage more than 60%

select s.student\_roll\_no , d.scholarship\_amount, s.student\_marks\_percentage from student\_scholarship\_information as d inner join student\_subject\_information as s on s.student\_roll\_no=d.student\_roll\_number where s.student\_marks\_percentage>60;



16) Mention the differences between the delete, drop and truncate commands

Drop and Truncate are DDL commands which change the entire structure of the table.

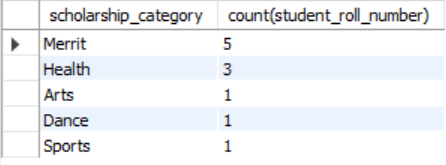
Truncate deletes all the records but not the structure of the table. It just removes all the data from the table.

Drop deletes all the data with the structure of the table. Data is completely lost and can’t be rolled back.

Delete can be used to remove the data with or without a condition or clause. Deletion of single or multiple rows depends on the query condition.

17) Get the count of the Scholarship category which is highly been availed by the students, i.e. get the count of the total number of students corresponding to the each scholarships category

select scholarship\_category,count(student\_roll\_number) from student.student\_scholarship\_information group by scholarship\_category order by count(student\_roll\_number) desc;



18) Along with the assignment no. 17 try to retrieve the maximum used scholarship category



19) Retrieve the percentage of the students along with students detailed information who has scored the highest percentage along with availing the maximum scholarship amount

select s.student\_roll\_no, s.student\_name, s.student\_email\_id, sc.scholarship\_amount, sub.student\_marks\_percentage from

student\_basic\_information as s join student\_scholarship\_information as sc join student\_subject\_information as sub on s.student\_roll\_no=sc.student\_roll\_number and sc.student\_roll\_number=sub.student\_roll\_no where student\_marks\_percentage in (select max(student\_marks\_percentage) from student\_subject\_information) and scholarship\_amount in (select max(scholarship\_amount) from student\_subject\_information);



20) Difference between the Triggers, Stored Procedures, Views and Functions

Trigger is invoked whenever an event occurs. They are programs which automatically get executed when some condition is met.

Stored procedure is a collection of statements that are stored inside a database. It has a name, parameters and set of statements that performs specific tasks.

View is a virtual table where data is not stored permanently. It is created for security purposes and for querying data by joining tables to save time.

Function is the same as a procedure, the only difference is that function can return value whereas a procedure cannot.