Submitted By : Navneet Kumar

Ans.1: Create Student Database

```
CREATE DATABASE STUDENT;
```

Ans.2: Creating tables

a. StudentBasicInformation

```
CREATE TABLE StudentBasicInformation (StudentRollNo int NOT NULL, StudentName varchar(255) NOT NULL, StudentSurname varchar(255), StudentAddress varchar(255), Age int, ScholarshipOpted varchar(255), Gender varchar(255), PRIMARY KEY (StudentRollNo) );
```

b. StudentAdmissionPaymentDetail

```
CREATE TABLE StudentAdmissionPaymentDetails(AmountPaid int, AmountBalance int, StudentRollNo int, AdmissionDate date, DueDate date, FOREIGN KEY (StudentRollNo) REFERENCES StudentBasicInformation(StudentRollNo));
```

c. **StudentSubjectInformation**

```
Create table StudentSubjectInformation ( StudentRollNo int, SubjectOpted varchar(255) NOT NULL, SubjectTotalMarks int, SubjectObtainedMarks int, StudentMarksPercentage int, MentorName varchar(255), FOREIGN KEY (StudentRollNo) REFERENCES StudentBasicInformation(StudentRollNo) );
```

d. SubjectScholarshipInformation

```
create table SubjectScholarshipInformation (ScholarshipName varchar(255), ScholarshipDescription varchar(2000), ScholarshipAmount int, ScholarshipCategory varchar(255), PRIMARY KEY (ScholarshipName) );
```

Ans.3 :
```
INSERT INTO StudentBasicInformation VALUES (1, 'John', 's', 'b.hk', 120, 'Yes', 'Male');
INSERT INTO StudentBasicInformation VALUES (2, 'Kaapil', 'J', '3B', 20, 'Yes', 'Male');
INSERT INTO StudentBasicInformation VALUES (3, 'Mahiat', 'Kart', '45 Helsinki', 20, 'Yes', 'Female');
INSERT INTO StudentBasicInformation VALUES (4, 'T', 'Erichsen', 'Stavanger', 20, 'Yes', 'Male');
INSERT INTO StudentBasicInformation VALUES (5, 'H', 'B', '15Sandnes', 19, 'No', 'Male');
INSERT INTO StudentBasicInformation VALUES (6, 'J', 'Y', 'H.Road', 19, 'Yes', 'Female');
INSERT INTO StudentBasicInformation VALUES (7, 'Nicholas', 'W', '1 Street', 20, 'No', 'Male');
INSERT INTO StudentBasicInformation VALUES (8, 'Nikom', 'B', 'Beach', 20, 'No', 'Male');
INSERT INTO StudentBasicInformation VALUES (9, 'David', 'K', '4 Garden', 20, 'Yes', 'Male');
INSERT INTO StudentBasicInformation VALUES (10, 'Hem', 'Johnson', 'Park Lane', 19, 'Yes', 'Male');
```

```
mysql> select * from StudentBasicInformation;
+---------------+-------------+----------------+----------------+------+------------------+--------+
| StudentRollNo | StudentName | StudentSurname | StudentAddress | Age  | ScholarshipOpted | Gender |
+---------------+-------------+----------------+----------------+------+------------------+--------+
|             1 | John        | s              | b.hk           |  120 | Yes              | Male   |
|             2 | Kaapil      | J              | 3B             |   20 | Yes              | Male   |
|             3 | Mahiat      | Kart           | 45 Helsinki    |   20 | Yes              | Female |
|             4 | T           | Erichsen       | Stavanger      |   20 | Yes              | Male   |
|             5 | H           | B              | 15Sandnes      |   19 | No               | Male   |
|             6 | J           | Y              | H.Road         |   19 | Yes              | Female |
|             7 | Nicholas    | W              | 1 Street       |   20 | No               | Male   |
|             8 | Nikom       | B              | Beach          |   20 | No               | Male   |
|             9 | David       | K              | 4 Garden       |   20 | Yes              | Male   |
|            10 | Hem         | Johnson        | Park Lane      |   19 | Yes              | Male   |
+---------------+-------------+----------------+----------------+------+------------------+--------+
```

```
INSERT INTO StudentAdmissionPaymentDetails VALUES (100, 900, 1, '2008-04-03', '2008-05-01');
INSERT INTO StudentAdmissionPaymentDetails VALUES (300, 700, 2, '2008-04-05', '2008-05-01');
INSERT INTO StudentAdmissionPaymentDetails VALUES (500, 500, 3, '2008-04-08', '2008-05-01');
INSERT INTO StudentAdmissionPaymentDetails VALUES (700, 300, 4, '2008-04-01', '2008-05-01');
INSERT INTO StudentAdmissionPaymentDetails VALUES (600, 400, 5, '2008-04-02', '2008-05-01');
INSERT INTO StudentAdmissionPaymentDetails  ALUES (400, 600, 6, '2008-04-12', '2008-05-01');
INSERT INTO StudentAdmissionPaymentDetails VALUES (100, 900, 7, '2008-04-14', '2008-05-01');
INSERT INTO StudentAdmissionPaymentDetails VALUES (150, 850, 8, '2008-04-09', '2008-05-01');
INSERT INTO StudentAdmissionPaymentDetails VALUES (250, 750, 9, '2008-04-08', '2008-05-01');
```

```
mysql> select * from StudentAdmissionPaymentDetails;
+------------+---------------+---------------+---------------+------------+
| AmountPaid | AmountBalance | StudentRollNo | AdmissionDate | DueDate    |
+------------+---------------+---------------+---------------+------------+
|        100 |           900 |             1 | 2008-04-03    | 2008-05-01 |
|        300 |           700 |             2 | 2008-04-05    | 2008-05-01 |
|        500 |           500 |             3 | 2008-04-08    | 2008-05-01 |
|        700 |           300 |             4 | 2008-04-01    | 2008-05-01 |
|        600 |           400 |             5 | 2008-04-02    | 2008-05-01 |
|        400 |           600 |             6 | 2008-04-12    | 2008-05-01 |
|        100 |           900 |             7 | 2008-04-14    | 2008-05-01 |
|        150 |           850 |             8 | 2008-04-09    | 2008-05-01 |
|        250 |           750 |             9 | 2008-04-08    | 2008-05-01 |
+------------+---------------+---------------+---------------+------------+
```

```
INSERT INTO StudentSubjectInformation (StudentRollNo, SubjectOpted, SubjectTotalMarks,
SubjectObtainedMarks, MentorName ) VALUES (1, 'English', 100, 76, 'Jessy');
INSERT INTO StudentSubjectInformation (StudentRollNo, SubjectOpted, SubjectTotalMarks,
SubjectObtainedMarks, MentorName ) VALUES (2, 'Computer Science', 100, 96, 'Maria');
INSERT INTO StudentSubjectInformation (StudentRollNo, SubjectOpted, SubjectTotalMarks,
SubjectObtainedMarks, MentorName ) VALUES (3, 'Social Science', 100, 80, 'Smith');
INSERT INTO StudentSubjectInformation (StudentRollNo, SubjectOpted, SubjectTotalMarks,
SubjectObtainedMarks, MentorName ) VALUES (4, 'History', 100, 60, 'Mriguez');
INSERT INTO StudentSubjectInformation (StudentRollNo, SubjectOpted, SubjectTotalMarks,
SubjectObtainedMarks, MentorName ) VALUES (5, 'Statistics', 100, 96, 'Madez');
INSERT INTO StudentSubjectInformation (StudentRollNo, SubjectOpted, SubjectTotalMarks,
SubjectObtainedMarks, MentorName ) VALUES (6, 'Calculas', 100, 68, 'Jouel');
INSERT INTO StudentSubjectInformation (StudentRollNo, SubjectOpted, SubjectTotalMarks,
SubjectObtainedMarks, MentorName ) VALUES (7, 'Geometry', 100, 89, 'Henry');
INSERT INTO StudentSubjectInformation (StudentRollNo, SubjectOpted, SubjectTotalMarks,
SubjectObtainedMarks, MentorName ) VALUES (8, 'English', 100, 46, 'Jessy');
INSERT INTO StudentSubjectInformation (StudentRollNo, SubjectOpted, SubjectTotalMarks,
SubjectObtainedMarks, MentorName ) VALUES (9, 'Geometry', 100, 76, 'Henry');
INSERT INTO StudentSubjectInformation (StudentRollNo, SubjectOpted, SubjectTotalMarks,
SubjectObtainedMarks, MentorName ) VALUES (10, 'Socila Science', 100, 46, 'Smith');
INSERT INTO StudentSubjectInformation (StudentRollNo, SubjectOpted, SubjectTotalMarks,
SubjectObtainedMarks, MentorName ) VALUES (11, 'Statistics', 100, 76, 'Hernandez');
```

```
mysql> select * from StudentSubjectInformation;
+---------------+------------------+------------------+--------------------+-----------------------+------------+
| StudentRollNo | SubjectOpted     | SubjectTotalMarks | SubjectObtainedMarks | StudentMarksPercentage | MentorName |
+---------------+------------------+------------------+--------------------+-----------------------+------------+
|             1 | English          |              100 |                 76 |                  NULL | Jessy      |
|             2 | Computer Science |              100 |                 96 |                  NULL | Maria      |
|             3 | Social Science   |              100 |                 80 |                  NULL | Smith      |
|             4 | History          |              100 |                 60 |                  NULL | Mriguez    |
|             5 | Statistics       |              100 |                 96 |                  NULL | Madez      |
|             6 | Calculas         |              100 |                 68 |                  NULL | Jouel      |
|             7 | Geometry         |              100 |                 89 |                  NULL | Henry      |
|             8 | English          |              100 |                 46 |                  NULL | Jessy      |
|             9 | Geometry         |              100 |                 76 |                  NULL | Henry      |
|            10 | Socila Science   |              100 |                 46 |                  NULL | Smith      |
+---------------+------------------+------------------+--------------------+-----------------------+------------+
10 rows in set (0.00 sec)
```

insert into SubjectScholarshipInformation values('Future', 'Scholarship more than 90%', 100, 'Excellent');
insert into SubjectScholarshipInformation values('Little Heart', 'Scholarship 80% marks and less than 90%', 90, 'Very Good');
insert into SubjectScholarshipInformation values('matter', 'Scholarship 70% marks and less than 80% in any subject', 80, 'Good');

```
mysql> select * from SubjectScholarshipInformation;
+-----------------+-----------------------------------------------------+-----------------+---------------------+
| ScholarshipName | ScholarshipDescription                              | ScholarshipAmount | ScholarshipCategory |
+-----------------+-----------------------------------------------------+-----------------+---------------------+
| Future          | Scholarship more than 90%                           |             100 | Excellent           |
| Little Heart    | Scholarship 80% marks and less than 90%             |              90 | Very Good           |
| matter          | Scholarship 70% marks and less than 80% in any subject |              80 | Good                |
+-----------------+-----------------------------------------------------+-----------------+---------------------+
3 rows in set (0.00 sec)
```

Ans.5 : **Q5. Update any 5 records of your choice in any table like update the StudentAddress with some other address content and likewise so on with any records of any table of your choice**

UPDATE StudentBasicInformation SET StudentName = 'Alfred' WHERE StudentRollNo = 1;
UPDATE StudentBasicInformation SET StudentName = 'Alfred' WHERE StudentRollNo = 1;
UPDATE StudentSubjectInformation SET SubjectOpted = 'English' WHERE StudentRollNo = 2;
UPDATE StudentSubjectInformation SET MentorName = 'Ralph Lauren' WHERE StudentRollNo = 4;
UPDATE StudentAdmissionPaymentDetails SET AmountPaid = 5000, AmountBalance=5000 WHERE StudentRollNo = 11;

Ans.6 :

```
mysql> select * from StudentSubjectInformation ;
+---------------+----------------+------------------+--------------------+-----------------------+--------------+
| StudentRollNo | SubjectOpted   | SubjectTotalMarks | SubjectObtainedMarks | StudentMarksPercentage | MentorName   |
+---------------+----------------+------------------+--------------------+-----------------------+--------------+
|             1 | English        |              100 |                 76 |                  NULL | Jessy        |
|             2 | English        |              100 |                 96 |                  NULL | Maria        |
|             3 | Social Science |              100 |                 80 |                  NULL | Smith        |
|             4 | History        |              100 |                 60 |                  NULL | Ralph Lauren |
|             5 | Statistics     |              100 |                 96 |                  NULL | Madez        |
|             6 | Calculas       |              100 |                 68 |                  NULL | Jouel        |
|             7 | Geometry       |              100 |                 89 |                  NULL | Henry        |
|             8 | English        |              100 |                 46 |                  NULL | Jessy        |
|             9 | Geometry       |              100 |                 76 |                  NULL | Henry        |
|            10 | Socila Science |              100 |                 46 |                  NULL | Smith        |
+---------------+----------------+------------------+--------------------+-----------------------+--------------+
```

Ans.7 : **Q7. Select the student details records who has received the scholarship more than 5000Rs/-**

select S.StudentName, S.StudentSurname, S.StudentRollNo from StudentBasicInformation as S inner join ScholarshipStudent on S.StudentRollNo = ScholarshipStudent.StudentRollNo inner join SubjectScholarshipInformation on SubjectScholarshipInformation.ScholarshipName = ScholarshipStudent.ScholarshipName where SubjectScholarshipInformation.scholarshipAmount > 50;

Q8. Select the students who opted for scholarship but has not got the scholarship

```
select S.StudentName, S.StudentSurname, S.StudentRollNo from StudentBasicInformation as S left
outer join ScholarshipStudent on S.StudentRollNo = ScholarshipStudent.StudentRollNo
where ScholarshipStudent.ScholarshipName = NULL and S.ScholarshipOpted = 'Yes';

Update StudentSubjectInformation set StudentMarksPercentage =
SubjectObtainedMarks*100) / SubjectTotalMarks where StudentRollNo = rollno;
END//
DELIMITER ;
CALL updatePercentage(1);
```

Q10. 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.

```
CREATE VIEW subjectbasic AS SELECT * FROM StudentBasicInformation natural join
StudentSubjectInformation;
Delimiter //
CREATE PROCEDURE updateScholarshipCategory(IN rollno int)
BEGIN
update ScholarshipStudent
set StudentRollNo = rollno and ScholarshipName = Case when rollno IN (SELECT StudentRollNo from
subjectbasic WHERE subjectbasic.StudentMarksPercentage between 71 and 80) THEN "I matter"
when rollno IN (SELECT StudentRollNo from subjectbasic WHERE
subjectbasic.StudentMarksPercentage between 81 and 90) THEN "Little Heart"
when rollno IN (SELECT StudentRollNo from subjectbasic WHERE
subjectbasic.StudentMarksPercentage between 91 and 100) THEN "Inspire Our Future"
else null
end;
END//
Delimiter ;
Call updateScholarshipCategory(1);
```

Q11. Create the View which shows balance amount to be paid by the student along with the student detailed information (use join)

```
CREATE VIEW balanceAmount AS SELECT StudentName, StudentSurname, AmountBalance
FROM StudentBasicInformation join StudentAdmissionPaymentDetails where
StudentBasicInformation.StudentRollNo = StudentAdmissionPaymentDetails.StudentRollNo;
SELECT * FROM balanceAmount;
```

Q12. Get the details of the students who haven't got any scholarship (use joins/subqueries)

```
select S.StudentRollNo from StudentBasicInformation as S left outer join ScholarshipStudent
on S.StudentRollNo = ScholarshipStudent.StudentRollNo where
ScholarshipStudent.ScholarshipName is NULL;
```

Q13. 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 //
CREATE PROCEDURE procedureBalanceAmount (IN rollno int, OUT balance INT)
BEGIN
SELECT AmountBalance
FROM StudentAdmissionPaymentDetails WHERE StudentRollNo = rollno;
```

END//
DELIMITER ;
CALL procedurebalanceAmount(1, @balance);

Q14. Retrieve the top five student details as per the StudentMarksPercentage values (use subqueries)

SELECT StudentRollNo, StudentName From StudentBasicInformation
WHERE StudentRollNo IN (SELECT StudentRollNo FROM StudentSubjectInformation ORDER BY SubjectObtainedMarks DESC LIMIT 5);

Q15. 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)

LEFT OUTER Join

select S.StudentRollNo from StudentBasicInformation as S left outer join ScholarshipStudent on S.StudentRollNo = ScholarshipStudent.StudentRollNo where ScholarshipStudent.ScholarshipName is NULL
SELECT S1.ScholarshipName, count(S1.ScholarshipName), S2.ScholarshipCategory
FROM ScholarshipStudent as S1, SubjectScholarshipInformation as S2
WHERE S1.ScholarshipName = S2.ScholarshipName
GROUP BY S1.ScholarshipName
ORDER BY count(S1.ScholarshipName) DESC;

RIGHT OUTER JOIN

select S.StudentRollNo
from left outer join ScholarshipStudent, StudentBasicInformation as S on
S.StudentRollNo = ScholarshipStudent.StudentRollNo
where ScholarshipStudent.ScholarshipName is NULL

Q16. Mention the differences between the delete, drop and truncate commands

DELETE	DROP	TRUNCATE
The DELETE statement in SQL is a Data Manipulation Language (DML) Command.	DROP statement is a Data Definition Language (DDL) Command	TRUNCATE command is a Data Definition Language (DDL)operation.
It is use to delete the one or more tuples of a table.	It is use to drop the whole table.	It is use to delete all the rows of a relation (table) in one go but not the table itself.
If used with WHERE clause, selected rows are deleted.	WHERE clause cannot be used.	WHERE clause cannot be used.
The structure or schema of the table is preserved.	The structure or schema of the table is not preserved.	The structure or schema of the table is preserved.
DELETE FROM Employees WHERE Emp_Id = 7;	DROP TABLE Employees;	TRUNCATE TABLE Employees;

Q17. 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 ScholarshipName, count(*) FROM ScholarshipStudent GROUP BY ScholarshipName;

Q18. Along with the assignment no. 17 try to retrieve the maximum used scholarship category
SELECT S1.ScholarshipName, count(S1.ScholarshipName), S2.ScholarshipCategory FROM ScholarshipStudent as S1, SubjectScholarshipInformation as S2 WHERE S1.ScholarshipName = S2.ScholarshipName GROUP BY S1.ScholarshipName ORDER BY count(S1.ScholarshipName) DESC;

Q19. 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 * FROM StudentBasicInformation as stu left JOIN ScholarshipStudent s ON stu.StudentRollNo = S.StudentRollNo left JOIN SubjectScholarshipInformation ON S.ScholarshipName = subjectScholarshipInformation.ScholarshipName WHERE Stu.StudentRollNo IN(SELECT StudentRollNo FROM StudentSubjectInformation WHERE StudentMarksPercentage = (SELECT max(StudentMarksPercentage) FROM StudentSubjectInformation));

Ans.20 :

TRIGGERS	STORED PROCEDURE	VIEWS	FUNCTIONS
trigger is a stored procedure that runs automatically when various events happen (eg: update, insert, delete)	Stored procedures are a pieces of the code in written in PL/SQL to do some specific task	A view is a virtual table based on the result-set of an SQL statement.	Functions are routines that accept parameters, perform an action, such as a complex calculation, and return the result of that action as a value.
Triggers cannot return values	Stored procedures can return values	It does not return a value but a table.	The return value can either be a single scalar value or a result set.