Accolite University Feb Batch 1- SQL Assignment

- Meet Shah
(INT626)

1. Create Student Database

```
CREATE DATABASE Student;
```

2. Create the following table under the Student Database:
    a. StudentBasicInformation
        i. Columns
            1. StudentName
            2. StudentSurname
            3. StudentRollNo
            4. StudentAddress
            5. Add more three basic columns of the name of your own

```
CREATE TABLE StudentBasicInformation ( StudentName VARCHAR(255) NOT NULL, StudentSurname VARCHAR(255), StudentRollNo INT NOT NULL, StudentAddress VARCHAR(255), StudentAge INT, StudentGender CHAR(1), OptedForScholarship BOOLEAN, PRIMARY KEY(StudentRollNo));
```

    b. StudentAdmissionPaymentDetails
        i. Columns
            1. StudentRollNo
            2. AmountPaid
            3. AmountBalance
            4. Add more four basic columns of the name of your own

```
CREATE TABLE StudentAdmissionPaymentDetails ( StudentRollNo INT, AmountPaid INT, AmountBalance INT, PaymentId INT(6) NOT NULL, PaymentDate DATE, PaymentMode VARCHAR(255), PaymentBank VARCHAR(255), FOREIGN KEY (StudentRollNo) REFERENCES StudentBasicInformation(StudentRollNo));
```

    c. StudentSubjectInformation
        i. Columns
            1. SubjectOpted
            2. StudentRollNo
            3. SubjectTotalMarks
            4. SubjectObtainedMarks
            5. StudentMarksPercentage

6. Add more one columns of the name of your own

```
CREATE TABLE StudentSubjectInformation ( SubjectOpted VARCHAR(255), StudentRollNo INT, SubjectTotalMarks INT, SubjectObtainedMarks INT, StudentMarksPercentage INT(100),  SubjectTeacher VARCHAR(255), FOREIGN KEY (StudentRollNo) REFERENCES StudentBasicInformation(StudentRollNo));
```

d. SubjectScholarshipInformation

i. Columns

1. StudentRollNo
2. ScholarshipName
3. ScholarshipDescription
4. ScholarshipAmount
5. ScholarshipCategory
6. Add more two columns of the name of your own

```
CREATE TABLE SubjectScholarshipInformation ( StudentRollNo INT, ScholarshipName VARCHAR(255), ScholarshipDescription VARCHAR(255), ScholarshipAmount INT, ScholarshipCategory VARCHAR(255), ScholarshipSubject VARCHAR(255), ScholarshipDurationMonths INT, FOREIGN KEY (StudentRollNo) REFERENCES StudentBasicInformation(StudentRollNo));
```

3. Insert more than 10 records in each and every table created

```
INSERT INTO StudentBasicInformation VALUES ('Aman', 'Sharma', 1, 'Mumbai', 20, 'M', true);
INSERT INTO StudentBasicInformation VALUES ('Diksha', 'Kashyap', 2, 'Delhi', 19, 'F', true);
INSERT INTO StudentBasicInformation VALUES ('Fenil', 'Kumar', 3, 'Kanpur', 21, 'M', true);
INSERT INTO StudentBasicInformation VALUES ('Gayatri', 'S', 4, 'Chennai', 20, 'F', false);
INSERT INTO StudentBasicInformation VALUES ('Harsh', 'Agarwal', 5, 'Jaipur', 20, 'M', true);
INSERT INTO StudentBasicInformation VALUES ('Heena', 'Khan', 6, 'Agra', 21, 'F', true);
INSERT INTO StudentBasicInformation VALUES ('Jayant', 'Babu', 7, 'Patna', 20, 'M', true);
INSERT INTO StudentBasicInformation VALUES ('Karan', 'Patil', 8, 'Indore', 19, 'M', false);
INSERT INTO StudentBasicInformation VALUES ('Mandeep', 'Singh', 9, 'Chandigarh', 20, 'M', false);
INSERT INTO StudentBasicInformation VALUES ('Naina', 'Kapoor', 10, 'Bangalore', 21, 'F', true);
INSERT INTO StudentBasicInformation VALUES ('Naresh', 'Narayan', 11, 'Hyderabad', 20, 'M', true);
```

```
INSERT INTO StudentBasicInformation VALUES ('Payal', 'Shah', 12, 'Ahmedabad', 20, 'F', false);
INSERT INTO StudentBasicInformation VALUES ('Rohan', 'Gupta', 13, 'Lucknow', 19, 'M', true);
INSERT INTO StudentBasicInformation VALUES ('Shantanu', 'Mukherjee', 14, 'Kolkata', 21, 'M', true);
INSERT INTO StudentBasicInformation VALUES ('Terence', 'Lewis', 15, 'Goa', 20, 'M', true);
```

```
INSERT INTO StudentAdmissionPaymentDetails VALUES (1, 90000, 10000, 111111, '2021-01-01','Debit Card', 'ICICI');
INSERT INTO StudentAdmissionPaymentDetails VALUES (2, 85000, 15000, 111112, '2021-01-02','Debit Card', 'HDFC');
INSERT INTO StudentAdmissionPaymentDetails VALUES (3, 100000, 0, 111113, '2021-01-01','Credit Card', 'ICICI');
INSERT INTO StudentAdmissionPaymentDetails VALUES (4, 100000, 0, 111114, '2021-01-03','Internet Banking', 'AXIS');
INSERT INTO StudentAdmissionPaymentDetails VALUES (5, 70000, 30000, 111115, '2021-01-01','Credit Card', 'SBI');
INSERT INTO StudentAdmissionPaymentDetails VALUES (6, 90000, 10000, 111116, '2021-01-03','Debit Card', 'SBI');
INSERT INTO StudentAdmissionPaymentDetails VALUES (7, 100000, 0, 111117, '2021-01-04','Debit Card', 'HDFC');
INSERT INTO StudentAdmissionPaymentDetails VALUES (8, 100000, 0, 111118, '2021-01-02','Debit Card', 'AXIS');
INSERT INTO StudentAdmissionPaymentDetails VALUES (9, 100000, 0, 111119, '2021-01-01','Internet Banking', 'ICICI');
INSERT INTO StudentAdmissionPaymentDetails VALUES (10, 50000, 50000, 111120, '2021-01-04','Internet Banking', 'ICICI');
INSERT INTO StudentAdmissionPaymentDetails VALUES (11, 65000, 35000, 111121, '2021-01-01','Debit Card', 'SBI');
INSERT INTO StudentAdmissionPaymentDetails VALUES (12, 85000, 15000, 111122, '2021-01-03','Credit Card', 'HDFC');
INSERT INTO StudentAdmissionPaymentDetails VALUES (13, 90000, 10000, 111123, '2021-01-01','Debit Card', 'ICICI');
INSERT INTO StudentAdmissionPaymentDetails VALUES (14, 100000, 0, 111124, '2021-01-02','Credit Card', 'AXIS');
INSERT INTO StudentAdmissionPaymentDetails VALUES (15, 30000, 70000, 111125, '2021-01-02','Internet Banking', 'AXIS');
```

```
INSERT INTO StudentSubjectInformation (SubjectOpted, StudentRollNo, SubjectTotalMarks, SubjectObtainedMarks, SubjectTeacher) VALUES ('Java', 1, 100, 75, 'Mr. Verma');

INSERT INTO StudentSubjectInformation (SubjectOpted, StudentRollNo, SubjectTotalMarks, SubjectObtainedMarks, SubjectTeacher) VALUES ('Java', 2, 100, 88, 'Mr. Verma');

INSERT INTO StudentSubjectInformation (SubjectOpted, StudentRollNo, SubjectTotalMarks, SubjectObtainedMarks, SubjectTeacher) VALUES ('C', 3, 100, 98, 'Mrs. Lakshmi');

INSERT INTO StudentSubjectInformation (SubjectOpted, StudentRollNo, SubjectTotalMarks, SubjectObtainedMarks, SubjectTeacher) VALUES ('C++', 4, 100, 63, 'Mr. Roy');

INSERT INTO StudentSubjectInformation (SubjectOpted, StudentRollNo, SubjectTotalMarks, SubjectObtainedMarks, SubjectTeacher) VALUES ('C++', 5, 100, 70, 'Mr. Roy');

INSERT INTO StudentSubjectInformation (SubjectOpted, StudentRollNo, SubjectTotalMarks, SubjectObtainedMarks, SubjectTeacher) VALUES ('Python', 6, 100, 88, 'Mr. Das');

INSERT INTO StudentSubjectInformation (SubjectOpted, StudentRollNo, SubjectTotalMarks, SubjectObtainedMarks, SubjectTeacher) VALUES ('Python', 7, 100, 96, 'Mr. Das');

INSERT INTO StudentSubjectInformation (SubjectOpted, StudentRollNo, SubjectTotalMarks, SubjectObtainedMarks, SubjectTeacher) VALUES ('Python', 8, 100, 60, 'Mr. Das');

INSERT INTO StudentSubjectInformation (SubjectOpted, StudentRollNo, SubjectTotalMarks, SubjectObtainedMarks, SubjectTeacher) VALUES ('C', 9, 100, 79, 'Mrs. Lakshmi');

INSERT INTO StudentSubjectInformation (SubjectOpted, StudentRollNo, SubjectTotalMarks, SubjectObtainedMarks, SubjectTeacher) VALUES ('C++', 10, 100, 94, 'Mr. Roy');

INSERT INTO StudentSubjectInformation (SubjectOpted, StudentRollNo, SubjectTotalMarks, SubjectObtainedMarks, SubjectTeacher) VALUES ('C++', 11, 100, 86, 'Mr. Roy');

INSERT INTO StudentSubjectInformation (SubjectOpted, StudentRollNo, SubjectTotalMarks, SubjectObtainedMarks, SubjectTeacher) VALUES ('Java', 12, 100, 59, 'Mr. Verma');

INSERT INTO StudentSubjectInformation (SubjectOpted, StudentRollNo, SubjectTotalMarks, SubjectObtainedMarks, SubjectTeacher) VALUES ('C++', 13, 100, 81, 'Mr. Roy');

INSERT INTO StudentSubjectInformation (SubjectOpted, StudentRollNo, SubjectTotalMarks, SubjectObtainedMarks, SubjectTeacher) VALUES ('Java', 14, 100, 69, 'Mr. Verma');

INSERT INTO StudentSubjectInformation (SubjectOpted, StudentRollNo, SubjectTotalMarks, SubjectObtainedMarks, SubjectTeacher) VALUES ('Java', 15, 100, 87, 'Mr. Verma');
```

```
INSERT INTO SubjectScholarshipInformation VALUES (2, 'Academic', 'Scholarship based on academic performance', 4000, 'General', 'Java', 6);
INSERT INTO SubjectScholarshipInformation VALUES (3, 'Academic', 'Scholarship based on academic performance', 8000, 'General', 'C', 6);
INSERT INTO SubjectScholarshipInformation VALUES (6, 'Academic', 'Scholarship based on academic performance', 4000, 'General', 'Python', 6);
INSERT INTO SubjectScholarshipInformation VALUES (7, 'Academic', 'Scholarship based on academic performance', 8000, 'General', 'Python', 6);
INSERT INTO SubjectScholarshipInformation VALUES (10, 'Academic', 'Scholarship based on academic performance', 6000, 'General', 'C++', 6);
INSERT INTO SubjectScholarshipInformation VALUES (11, 'Academic', 'Scholarship based on academic performance', 4000, 'General', 'C++', 6);
INSERT INTO SubjectScholarshipInformation VALUES (15, 'Academic', 'Scholarship based on academic performance', 4000, 'General', 'Java', 6);
```

4. Snap of the all the tables once the insertion is completed

```
mysql> CREATE DATABASE Student;
Query OK, 1 row affected (0.54 sec)

mysql> USE Student;
Database changed
mysql> CREATE TABLE StudentBasicInformation ( StudentName VARCHAR(255) NOT NULL, StudentSurname VARCHAR(255), StudentRollNo INT NOT NULL, StudentAddress VARCHAR(255), S
tudentAge INT, StudentGender CHAR(1), OptedForScholarship BOOLEAN, PRIMARY KEY(StudentRollNo));
Query OK, 0 rows affected (3.25 sec)

mysql> CREATE TABLE StudentAdmissionPaymentDetails ( StudentRollNo INT, AmountPaid INT, AmountBalance INT, PaymentId INT(6) NOT NULL, PaymentDate DATE, PaymentMode VARC
HAR(255), PaymentBank VARCHAR(255), FOREIGN KEY (StudentRollNo) REFERENCES StudentBasicInformation(StudentRollNo));
Query OK, 0 rows affected, 1 warning (1.54 sec)

mysql> CREATE TABLE StudentSubjectInformation ( SubjectOpted VARCHAR(255), StudentRollNo INT, SubjectTotalMarks INT, SubjectObtainedMarks INT, StudentMarksPercentage IN
T(100), SubjectTeacher VARCHAR(255), FOREIGN KEY (StudentRollNo) REFERENCES StudentBasicInformation(StudentRollNo));
Query OK, 0 rows affected, 1 warning (1.38 sec)

mysql> CREATE TABLE SubjectScholarshipInformation ( StudentRollNo INT, ScholarshipName VARCHAR(255), ScholarshipDescription VARCHAR(255), ScholarshipAmount INT, Scholar
shipCategory VARCHAR(255),  ScholarshipSubject VARCHAR(255), ScholarshipDurationMonths INT, FOREIGN KEY (StudentRollNo) REFERENCES StudentBasicInformation(StudentRollNo
));
Query OK, 0 rows affected (1.37 sec)
```

```
mysql> SELECT * FROM StudentBasicInformation;
+-------------+----------------+---------------+----------------+------------+---------------+----------------------+
| StudentName | StudentSurname | StudentRollNo | StudentAddress | StudentAge | StudentGender | OptedForScholarship |
+-------------+----------------+---------------+----------------+------------+---------------+----------------------+
| Aman        | Sharma         |             1 | Mumbai         |         20 | M             |                    1 |
| Diksha      | Kashyap        |             2 | Delhi          |         19 | F             |                    1 |
| Fenil       | Kumar          |             3 | Kanpur         |         21 | M             |                    1 |
| Gayatri     | S              |             4 | Chennai        |         20 | F             |                    0 |
| Harsh       | Agarwal        |             5 | Jaipur         |         20 | M             |                    1 |
| Heena       | Khan           |             6 | Agra           |         21 | F             |                    1 |
| Jayant      | Babu           |             7 | Patna          |         20 | M             |                    1 |
| Karan       | Patil          |             8 | Indore         |         19 | M             |                    0 |
| Mandeep     | Singh          |             9 | Chandigarh     |         20 | M             |                    0 |
| Naina       | Kapoor         |            10 | Bangalore      |         21 | F             |                    1 |
| Naresh      | Narayan        |            11 | Hyderabad      |         20 | M             |                    1 |
| Payal       | Shah           |            12 | Ahmedabad      |         20 | F             |                    0 |
| Rohan       | Gupta          |            13 | Lucknow        |         19 | M             |                    1 |
| Shantanu    | Mukherjee      |            14 | Kolkata        |         21 | M             |                    1 |
| Terence     | Lewis          |            15 | Goa            |         20 | M             |                    1 |
+-------------+----------------+---------------+----------------+------------+---------------+----------------------+
15 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM StudentAdmissionPaymentDetails;
+---------------+------------+---------------+-----------+-------------+------------------+-------------+
| StudentRollNo | AmountPaid | AmountBalance | PaymentId | PaymentDate | PaymentMode      | PaymentBank |
+---------------+------------+---------------+-----------+-------------+------------------+-------------+
|             1 |      90000 |         10000 |    111111 | 2021-01-01  | Debit Card       | ICICI       |
|             2 |      85000 |         15000 |    111112 | 2021-01-02  | Debit Card       | HDFC        |
|             3 |     100000 |             0 |    111113 | 2021-01-01  | Credit Card      | ICICI       |
|             4 |     100000 |             0 |    111114 | 2021-01-03  | Internet Banking | AXIS        |
|             5 |      70000 |         30000 |    111115 | 2021-01-01  | Credit Card      | SBI         |
|             6 |      90000 |         10000 |    111116 | 2021-01-03  | Debit Card       | SBI         |
|             7 |     100000 |             0 |    111117 | 2021-01-04  | Debit Card       | HDFC        |
|             8 |     100000 |             0 |    111118 | 2021-01-02  | Debit Card       | AXIS        |
|             9 |     100000 |             0 |    111119 | 2021-01-01  | Internet Banking | ICICI       |
|            10 |      50000 |         50000 |    111120 | 2021-01-04  | Internet Banking | ICICI       |
|            11 |      65000 |         35000 |    111121 | 2021-01-01  | Debit Card       | SBI         |
|            12 |      85000 |         15000 |    111122 | 2021-01-03  | Credit Card      | HDFC        |
|            13 |      90000 |         10000 |    111123 | 2021-01-01  | Debit Card       | ICICI       |
|            14 |     100000 |             0 |    111124 | 2021-01-02  | Credit Card      | AXIS        |
|            15 |      30000 |         70000 |    111125 | 2021-01-02  | Internet Banking | AXIS        |
+---------------+------------+---------------+-----------+-------------+------------------+-------------+
15 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM StudentSubjectInformation;
+-------------+---------------+------------------+---------------------+------------------------+----------------+
| SubjectOpted | StudentRollNo | SubjectTotalMarks | SubjectObtainedMarks | StudentMarksPercentage | SubjectTeacher |
+-------------+---------------+------------------+---------------------+------------------------+----------------+
| Java        |             1 |              100 |                  75 |                   NULL | Mr. Verma      |
| Java        |             2 |              100 |                  88 |                   NULL | Mr. Verma      |
| C           |             3 |              100 |                  98 |                   NULL | Mrs. Lakshmi   |
| C++         |             4 |              100 |                  63 |                   NULL | Mr. Roy        |
| C++         |             5 |              100 |                  70 |                   NULL | Mr. Roy        |
| Python      |             6 |              100 |                  88 |                   NULL | Mr. Das        |
| Python      |             7 |              100 |                  96 |                   NULL | Mr. Das        |
| Python      |             8 |              100 |                  60 |                   NULL | Mr. Das        |
| C           |             9 |              100 |                  79 |                   NULL | Mrs. Lakshmi   |
| C++         |            10 |              100 |                  94 |                   NULL | Mr. Roy        |
| C++         |            11 |              100 |                  86 |                   NULL | Mr. Roy        |
| Java        |            12 |              100 |                  59 |                   NULL | Mr. Verma      |
| C++         |            13 |              100 |                  81 |                   NULL | Mr. Roy        |
| Java        |            14 |              100 |                  69 |                   NULL | Mr. Verma      |
| Java        |            15 |              100 |                  87 |                   NULL | Mr. Verma      |
+-------------+---------------+------------------+---------------------+------------------------+----------------+
15 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM SubjectScholarshipInformation;
+---------------+---------------+-----------------------------------------+------------------+-------------------+-------------------+-------------------------+
| StudentRollNo | ScholarshipName | ScholarshipDescription                | ScholarshipAmount | ScholarshipCategory | ScholarshipSubject | ScholarshipDurationMonths |
+---------------+---------------+-----------------------------------------+------------------+-------------------+-------------------+-------------------------+
|             2 | Academic      | Scholarship based on academic performance |             4000 | General           | Java              |                       6 |
|             3 | Academic      | Scholarship based on academic performance |             8000 | General           | C                 |                       6 |
|             6 | Academic      | Scholarship based on academic performance |             4000 | General           | Python            |                       6 |
|             7 | Academic      | Scholarship based on academic performance |             8000 | General           | Python            |                       6 |
|            10 | Academic      | Scholarship based on academic performance |             6000 | General           | C++               |                       6 |
|            11 | Academic      | Scholarship based on academic performance |             4000 | General           | C++               |                       6 |
|            15 | Academic      | Scholarship based on academic performance |             4000 | General           | Java              |                       6 |
+---------------+---------------+-----------------------------------------+------------------+-------------------+-------------------+-------------------------+
```

5. 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 StudentAddress = 'Pune' WHERE StudentRollNo = 7;

UPDATE StudentAdmissionPaymentDetails SET PaymentBank = 'SBI' WHERE StudentRollNo = 15;

UPDATE StudentSubjectInformation SET SubjectObtainedMarks = 62 WHERE StudentRollNo = 4;
```

```
UPDATE StudentAdmissionPaymentDetails SET AmountPaid = 10000,
AmountBalance = 0 WHERE StudentRollNo = 12;

UPDATE SubjectScholarshipInformation SET ScholarshipDurationMonths = 8
WHERE ScholarshipSubject = 'Java';
```

```
mysql> UPDATE StudentBasicInformation SET StudentAddress = 'Pune' WHERE StudentRollNo = 7;
Query OK, 1 row affected (0.22 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE StudentAdmissionPaymentDetails SET PaymentBank = 'SBI' WHERE StudentRollNo = 15;
Query OK, 1 row affected (0.27 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE StudentSubjectInformation SET SubjectObtainedMarks = 62 WHERE StudentRollNo = 4;
Query OK, 1 row affected (0.16 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE StudentAdmissionPaymentDetails SET AmountPaid = 10000, AmountBalance = 0 WHERE StudentRollNo = 12;
Query OK, 1 row affected (0.12 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE SubjectScholarshipInformation SET ScholarshipDurationMonths = 8 WHERE ScholarshipSubject = 'Java';
Query OK, 2 rows affected (0.10 sec)
Rows matched: 2  Changed: 2  Warnings: 0
```

6. Snap of the all the tables post updation

```
mysql> SELECT * FROM StudentBasicInformation;
+-------------+----------------+---------------+----------------+------------+---------------+----------------------+
| StudentName | StudentSurname | StudentRollNo | StudentAddress | StudentAge | StudentGender | OptedForScholarship |
+-------------+----------------+---------------+----------------+------------+---------------+----------------------+
| Aman        | Sharma         |             1 | Mumbai         |         20 | M             |                    1 |
| Diksha      | Kashyap        |             2 | Delhi          |         19 | F             |                    1 |
| Fenil       | Kumar          |             3 | Kanpur         |         21 | M             |                    1 |
| Gayatri     | S              |             4 | Chennai        |         20 | F             |                    0 |
| Harsh       | Agarwal        |             5 | Jaipur         |         20 | M             |                    1 |
| Heena       | Khan           |             6 | Agra           |         21 | F             |                    1 |
| Jayant      | Babu           |             7 | Pune           |         20 | M             |                    1 |
| Karan       | Patil          |             8 | Indore         |         19 | M             |                    0 |
| Mandeep     | Singh          |             9 | Chandigarh     |         20 | M             |                    0 |
| Naina       | Kapoor         |            10 | Bangalore      |         21 | F             |                    1 |
| Naresh      | Narayan        |            11 | Hyderabad      |         20 | M             |                    1 |
| Payal       | Shah           |            12 | Ahmedabad      |         20 | F             |                    0 |
| Rohan       | Gupta          |            13 | Lucknow        |         19 | M             |                    1 |
| Shantanu    | Mukherjee      |            14 | Kolkata        |         21 | M             |                    1 |
| Terence     | Lewis          |            15 | Goa            |         20 | M             |                    1 |
+-------------+----------------+---------------+----------------+------------+---------------+----------------------+
15 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM StudentAdmissionPaymentDetails;
+---------------+------------+---------------+-----------+-------------+------------------+-------------+
| StudentRollNo | AmountPaid | AmountBalance | PaymentId | PaymentDate | PaymentMode      | PaymentBank |
+---------------+------------+---------------+-----------+-------------+------------------+-------------+
|             1 |      90000 |         10000 |    111111 | 2021-01-01  | Debit Card       | ICICI       |
|             2 |      85000 |         15000 |    111112 | 2021-01-02  | Debit Card       | HDFC        |
|             3 |     100000 |             0 |    111113 | 2021-01-01  | Credit Card      | ICICI       |
|             4 |     100000 |             0 |    111114 | 2021-01-03  | Internet Banking | AXIS        |
|             5 |      70000 |         30000 |    111115 | 2021-01-01  | Credit Card      | SBI         |
|             6 |      90000 |         10000 |    111116 | 2021-01-03  | Debit Card       | SBI         |
|             7 |     100000 |             0 |    111117 | 2021-01-04  | Debit Card       | HDFC        |
|             8 |     100000 |             0 |    111118 | 2021-01-02  | Debit Card       | AXIS        |
|             9 |     100000 |             0 |    111119 | 2021-01-01  | Internet Banking | ICICI       |
|            10 |      50000 |         50000 |    111120 | 2021-01-04  | Internet Banking | ICICI       |
|            11 |      65000 |         35000 |    111121 | 2021-01-01  | Debit Card       | SBI         |
|            12 |      10000 |             0 |    111122 | 2021-01-03  | Credit Card      | HDFC        |
|            13 |      90000 |         10000 |    111123 | 2021-01-01  | Debit Card       | ICICI       |
|            14 |     100000 |             0 |    111124 | 2021-01-02  | Credit Card      | AXIS        |
|            15 |      30000 |         70000 |    111125 | 2021-01-02  | Internet Banking | SBI         |
+---------------+------------+---------------+-----------+-------------+------------------+-------------+
15 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM StudentSubjectInformation;
+--------------+---------------+------------------+----------------------+------------------------+----------------+
| SubjectOpted | StudentRollNo | SubjectTotalMarks | SubjectObtainedMarks | StudentMarksPercentage | SubjectTeacher |
+--------------+---------------+------------------+----------------------+------------------------+----------------+
| Java         |             1 |              100 |                   75 |                   NULL | Mr. Verma      |
| Java         |             2 |              100 |                   88 |                   NULL | Mr. Verma      |
| C            |             3 |              100 |                   98 |                   NULL | Mrs. Lakshmi   |
| C++          |             4 |              100 |                   62 |                   NULL | Mr. Roy        |
| C++          |             5 |              100 |                   70 |                   NULL | Mr. Roy        |
| Python       |             6 |              100 |                   88 |                   NULL | Mr. Das        |
| Python       |             7 |              100 |                   96 |                   NULL | Mr. Das        |
| Python       |             8 |              100 |                   60 |                   NULL | Mr. Das        |
| C            |             9 |              100 |                   79 |                   NULL | Mrs. Lakshmi   |
| C++          |            10 |              100 |                   94 |                   NULL | Mr. Roy        |
| C++          |            11 |              100 |                   86 |                   NULL | Mr. Roy        |
| Java         |            12 |              100 |                   59 |                   NULL | Mr. Verma      |
| C++          |            13 |              100 |                   81 |                   NULL | Mr. Roy        |
| Java         |            14 |              100 |                   69 |                   NULL | Mr. Verma      |
| Java         |            15 |              100 |                   87 |                   NULL | Mr. Verma      |
+--------------+---------------+------------------+----------------------+------------------------+----------------+
15 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM SubjectScholarshipInformation;
+---------------+-----------------+-----------------------------------------+------------------+--------------------+--------------------+-------------------------+
| StudentRollNo | ScholarshipName | ScholarshipDescription                  | ScholarshipAmount | ScholarshipCategory | ScholarshipSubject | ScholarshipDurationMonths |
+---------------+-----------------+-----------------------------------------+------------------+--------------------+--------------------+-------------------------+
|             2 | Academic        | Scholarship based on academic performance |             4000 | General            | Java               |                       8 |
|             3 | Academic        | Scholarship based on academic performance |             8000 | General            | C                  |                       6 |
|             6 | Academic        | Scholarship based on academic performance |             4000 | General            | Python             |                       6 |
|             7 | Academic        | Scholarship based on academic performance |             8000 | General            | Python             |                       6 |
|            10 | Academic        | Scholarship based on academic performance |             6000 | General            | C++                |                       6 |
|            11 | Academic        | Scholarship based on academic performance |             4000 | General            | C++                |                       6 |
|            15 | Academic        | Scholarship based on academic performance |             4000 | General            | Java               |                       8 |
+---------------+-----------------+-----------------------------------------+------------------+--------------------+--------------------+-------------------------+
```

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

```
SELECT SBI.StudentName, SBI.StudentSurname, SBI.StudentRollNo, SBI.StudentAddress, SBI.StudentAge, SBI.StudentGender, SSI.ScholarshipAmount FROM StudentBasicInformation SBI INNER JOIN SubjectScholarshipInformation SSI ON SBI.StudentRollNo = SSI.StudentRollNo where SSI.ScholarshipAmount > 5000;
```

```
mysql> SELECT SBI.StudentName, SBI.StudentSurname, SBI.StudentRollNo, SBI.StudentAddress, SBI.StudentAge, SBI.StudentGender, SSI.ScholarshipAmount FROM StudentBasicInfo
rmation SBI INNER JOIN SubjectScholarshipInformation SSI ON SBI.StudentRollNo = SSI.StudentRollNo where SSI.ScholarshipAmount > 5000;
+-------------+----------------+---------------+----------------+------------+---------------+------------------+
| StudentName | StudentSurname | StudentRollNo | StudentAddress | StudentAge | StudentGender | ScholarshipAmount |
+-------------+----------------+---------------+----------------+------------+---------------+------------------+
| Fenil       | Kumar          |             3 | Kanpur         |         21 | M             |             8000 |
| Jayant      | Babu           |             7 | Pune           |         20 | M             |             8000 |
| Naina       | Kapoor         |            10 | Bangalore      |         21 | F             |             6000 |
+-------------+----------------+---------------+----------------+------------+---------------+------------------+
3 rows in set (0.00 sec)
```

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

```
SELECT * FROM StudentBasicInformation WHERE OptedForScholarship = true AND StudentRollNo NOT IN (SELECT StudentRollNo FROM SubjectScholarshipInformation);
```

```
mysql> SELECT * FROM StudentBasicInformation WHERE OptedForScholarship = true AND StudentRollNo NOT IN (SELECT StudentRollNo FROM SubjectScholarshipInformation);
+-------------+---------------+--------------+----------------+------------+--------------+---------------------+
| StudentName | StudentSurname | StudentRollNo | StudentAddress | StudentAge | StudentGender | OptedForScholarship |
+-------------+---------------+--------------+----------------+------------+--------------+---------------------+
| Aman        | Sharma        |            1 | Mumbai         |         20 | M            |                   1 |
| Harsh       | Agarwal       |            5 | Jaipur         |         20 | M            |                   1 |
| Rohan       | Gupta         |           13 | Lucknow        |         19 | M            |                   1 |
| Shantanu    | Mukherjee     |           14 | Kolkata        |         21 | M            |                   1 |
+-------------+---------------+--------------+----------------+------------+--------------+---------------------+
4 rows in set (0.06 sec)
```

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 CalculatePercentage()
BEGIN
        UPDATE StudentSubjectInformation SET StudentMarksPercentage =
        SubjectObtainedMarks*100/SubjectTotalMarks;
END &&
DELIMITER ;
CALL CalculatePercentage();
```

```
mysql> DELIMITER &&
mysql> CREATE PROCEDURE CalculatePercentage()
    -> BEGIN
    -> UPDATE StudentSubjectInformation SET StudentMarksPercentage = SubjectObtainedMarks*100/SubjectTotalMarks;
    -> END &&
Query OK, 0 rows affected (0.32 sec)
```

```
mysql> CALL CalculatePercentage();
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * FROM StudentSubjectInformation;
+--------------+---------------+------------------+---------------------+------------------------+----------------+
| SubjectOpted | StudentRollNo | SubjectTotalMarks | SubjectObtainedMarks | StudentMarksPercentage | SubjectTeacher |
+--------------+---------------+------------------+---------------------+------------------------+----------------+
| Java         |             1 |              100 |                  75 |                     75 | Mr. Verma      |
| Java         |             2 |              100 |                  88 |                     88 | Mr. Verma      |
| C            |             3 |              100 |                  98 |                     98 | Mrs. Lakshmi   |
| C++          |             4 |              100 |                  62 |                     62 | Mr. Roy        |
| C++          |             5 |              100 |                  70 |                     70 | Mr. Roy        |
| Python       |             6 |              100 |                  88 |                     88 | Mr. Das        |
| Python       |             7 |              100 |                  96 |                     96 | Mr. Das        |
| Python       |             8 |              100 |                  60 |                     60 | Mr. Das        |
| C            |             9 |              100 |                  79 |                     79 | Mrs. Lakshmi   |
| C++          |            10 |              100 |                  94 |                     94 | Mr. Roy        |
| C++          |            11 |              100 |                  86 |                     86 | Mr. Roy        |
| Java         |            12 |              100 |                  59 |                     59 | Mr. Verma      |
| C++          |            13 |              100 |                  81 |                     81 | Mr. Roy        |
| Java         |            14 |              100 |                  69 |                     69 | Mr. Verma      |
| Java         |            15 |              100 |                  87 |                     87 | Mr. Verma      |
+--------------+---------------+------------------+---------------------+------------------------+----------------+
15 rows in set (0.00 sec)
```

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 FillCategory()
BEGIN
        UPDATE SubjectScholarshipInformation SET ScholarshipCategory = CASE
        WHEN StudentRollNo IN (SELECT StudentRollNo FROM
        StudentSubjectInformation WHERE StudentMarksPercentage >= 95) THEN
        "Category 1"
        WHEN StudentRollNo IN (SELECT StudentRollNo FROM
        StudentSubjectInformation WHERE StudentMarksPercentage >= 90 AND
        StudentMarksPercentage < 95) THEN "Category 2"
        WHEN StudentRollNo IN (SELECT StudentRollNo FROM
        StudentSubjectInformation WHERE StudentMarksPercentage >= 85 AND
        StudentMarksPercentage < 90) THEN "Category 3"
        ELSE "NULL"
        END;
END &&
DELIMITER ;
CALL FillCategory();
```

```
mysql> DELIMITER &&
mysql> CREATE PROCEDURE FillCategory()
    -> BEGIN
    -> UPDATE SubjectScholarshipInformation SET ScholarshipCategory = CASE
    -> WHEN StudentRollNo IN (SELECT StudentRollNo FROM StudentSubjectInformation WHERE StudentMarksPercentage >= 95) THEN "Category 1"
    -> WHEN StudentRollNo IN (SELECT StudentRollNo FROM StudentSubjectInformation WHERE StudentMarksPercentage >= 90 AND StudentMarksPercentage < 95) THEN "Category 2"
    -> WHEN StudentRollNo IN (SELECT StudentRollNo FROM StudentSubjectInformation WHERE StudentMarksPercentage >= 85 AND StudentMarksPercentage < 90) THEN "Category 3"
    -> ELSE "NULL"
    -> END;
    -> END &&
Query OK, 0 rows affected (0.14 sec)
```

```
mysql> CALL FillCategory();
Query OK, 7 rows affected (0.28 sec)

mysql> SELECT * FROM SubjectScholarshipInformation;
+---------------+-----------------+-----------------------------------------+-------------------+--------------------+--------------------+--------------------------+
| StudentRollNo | ScholarshipName | ScholarshipDescription                  | ScholarshipAmount | ScholarshipCategory | ScholarshipSubject | ScholarshipDurationMonths |
+---------------+-----------------+-----------------------------------------+-------------------+--------------------+--------------------+--------------------------+
|             2 | Academic        | Scholarship based on academic performance |              4000 | Category 3          | Java               |                        8 |
|             3 | Academic        | Scholarship based on academic performance |              8000 | Category 1          | C                  |                        6 |
|             6 | Academic        | Scholarship based on academic performance |              4000 | Category 3          | Python             |                        6 |
|             7 | Academic        | Scholarship based on academic performance |              8000 | Category 1          | Python             |                        6 |
|            10 | Academic        | Scholarship based on academic performance |              6000 | Category 2          | C++                |                        6 |
|            11 | Academic        | Scholarship based on academic performance |              4000 | Category 3          | C++                |                        6 |
|            15 | Academic        | Scholarship based on academic performance |              4000 | Category 3          | Java               |                        8 |
+---------------+-----------------+-----------------------------------------+-------------------+--------------------+--------------------+--------------------------+
7 rows in set (0.00 sec)
```

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

```
CREATE VIEW StudentBalance AS
SELECT SBI.StudentName, SBI.StudentSurname, SBI.StudentRollNo,
SBI.StudentAddress, SBI.StudentAge, SBI.StudentGender, SAPD.AmountBalance
FROM StudentBasicInformation SBI INNER JOIN
StudentAdmissionPaymentDetails SAPD ON SBI.StudentRollNo =
SAPD.StudentRollNo;
```

```
mysql> CREATE VIEW StudentBalance AS
    -> SELECT SBI.StudentName, SBI.StudentSurname, SBI.StudentRollNo, SBI.StudentAddress, SBI.StudentAge, SBI.StudentGender, SAPD.AmountBalance FROM StudentBasicInforma
tion SBI INNER JOIN StudentAdmissionPaymentDetails SAPD ON SBI.StudentRollNo = SAPD.StudentRollNo;
Query OK, 0 rows affected (1.11 sec)

mysql> SELECT * FROM StudentBalance;
+-------------+---------------+--------------+----------------+------------+---------------+---------------+
| StudentName | StudentSurname | StudentRollNo | StudentAddress | StudentAge | StudentGender | AmountBalance |
+-------------+---------------+--------------+----------------+------------+---------------+---------------+
| Aman        | Sharma        |            1 | Mumbai         |         20 | M             |         10000 |
| Diksha      | Kashyap       |            2 | Delhi          |         19 | F             |         15000 |
| Fenil       | Kumar         |            3 | Kanpur         |         21 | M             |             0 |
| Gayatri     | S             |            4 | Chennai        |         20 | F             |             0 |
| Harsh       | Agarwal       |            5 | Jaipur         |         20 | M             |         30000 |
| Heena       | Khan          |            6 | Agra           |         21 | F             |         10000 |
| Jayant      | Babu          |            7 | Pune           |         20 | M             |             0 |
| Karan       | Patil         |            8 | Indore         |         19 | M             |             0 |
| Mandeep     | Singh         |            9 | Chandigarh     |         20 | M             |             0 |
| Naina       | Kapoor        |           10 | Bangalore      |         21 | F             |         50000 |
| Naresh      | Narayan       |           11 | Hyderabad      |         20 | M             |         35000 |
| Payal       | Shah          |           12 | Ahmedabad      |         20 | F             |             0 |
| Rohan       | Gupta         |           13 | Lucknow        |         19 | M             |         10000 |
| Shantanu    | Mukherjee     |           14 | Kolkata        |         21 | M             |             0 |
| Terence     | Lewis         |           15 | Goa            |         20 | M             |         70000 |
+-------------+---------------+--------------+----------------+------------+---------------+---------------+
15 rows in set (0.15 sec)
```

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

```
SELECT * FROM StudentBasicInformation WHERE StudentRollNo NOT IN (
SELECT StudentRollNo FROM SubjectScholarshipInformation);
```

```
mysql> SELECT * FROM StudentBasicInformation WHERE StudentRollNo NOT IN (
    -> SELECT StudentRollNo FROM SubjectScholarshipInformation);
+-------------+---------------+--------------+----------------+------------+---------------+---------------------+
| StudentName | StudentSurname | StudentRollNo | StudentAddress | StudentAge | StudentGender | OptedForScholarship |
+-------------+---------------+--------------+----------------+------------+---------------+---------------------+
| Aman        | Sharma        |            1 | Mumbai         |         20 | M             |                   1 |
| Gayatri     | S             |            4 | Chennai        |         20 | F             |                   0 |
| Harsh       | Agarwal       |            5 | Jaipur         |         20 | M             |                   1 |
| Karan       | Patil         |            8 | Indore         |         19 | M             |                   0 |
| Mandeep     | Singh         |            9 | Chandigarh     |         20 | M             |                   0 |
| Payal       | Shah          |           12 | Ahmedabad      |         20 | F             |                   0 |
| Rohan       | Gupta         |           13 | Lucknow        |         19 | M             |                   1 |
| Shantanu    | Mukherjee     |           14 | Kolkata        |         21 | M             |                   1 |
+-------------+---------------+--------------+----------------+------------+---------------+---------------------+
8 rows in set (0.04 sec)
```

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 &&
CREATE PROCEDURE GetBalance(IN RollNo INT)
BEGIN
        SELECT AmountBalance FROM StudentAdmissionPaymentDetails WHERE
        StudentRollNo = RollNo;
END &&
DELIMITER ;
CALL GetBalance(8);
```

```
mysql> DELIMITER &&
mysql> CREATE PROCEDURE GetBalance(IN RollNo INT)
    -> BEGIN
    -> SELECT AmountBalance FROM StudentAdmissionPaymentDetails WHERE StudentRollNo = RollNo;
    -> END &&
Query OK, 0 rows affected (0.33 sec)

mysql> DELIMITER ;
mysql> CALL GetBalance(8);
+---------------+
| AmountBalance |
+---------------+
|             0 |
+---------------+
1 row in set (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

mysql> CALL GetBalance(11);
+---------------+
| AmountBalance |
+---------------+
|         35000 |
+---------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

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

```
SELECT SBI.*, SSI.SubjectObtainedMarks FROM StudentBasicInformation SBI
INNER JOIN (SELECT StudentRollNo, SubjectObtainedMarks FROM
StudentSubjectInformation ORDER BY SubjectObtainedMarks DESC LIMIT 5) SSI
ON SBI.StudentRollNo = SSI.StudentRollNo;
```

```
mysql> SELECT SBI.*, SSI.SubjectObtainedMarks FROM StudentBasicInformation SBI INNER JOIN (SELECT StudentRollNo, SubjectObtainedMarks FROM StudentSubjectInformation ORD
ER BY SubjectObtainedMarks DESC LIMIT 5) SSI ON SBI.StudentRollNo = SSI.StudentRollNo;
+-------------+---------------+--------------+----------------+------------+--------------+--------------------+----------------------+
| StudentName | StudentSurname | StudentRollNo | StudentAddress | StudentAge | StudentGender | OptedForScholarship | SubjectObtainedMarks |
+-------------+---------------+--------------+----------------+------------+--------------+--------------------+----------------------+
| Fenil       | Kumar         |            3 | Kanpur         |         21 | M            |                  1 |                   98 |
| Jayant      | Babu          |            7 | Pune           |         20 | M            |                  1 |                   96 |
| Naina       | Kapoor        |           10 | Bangalore      |         21 | F            |                  1 |                   94 |
| Diksha      | Kashyap       |            2 | Delhi          |         19 | F            |                  1 |                   88 |
| Heena       | Khan          |            6 | Agra           |         21 | F            |                  1 |                   88 |
+-------------+---------------+--------------+----------------+------------+--------------+--------------------+----------------------+
5 rows in set (0.00 sec)
```

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)

    Inner joins - We can find the details of all the students with their respective percentage. Inner Joins are useful when we need to get the data from both the tables using some condition.

```
SELECT SBI.StudentRollNo, SBI.StudentName, SSI.StudentMarksPercentage
FROM StudentBasicInformation SBI INNER JOIN StudentSubjectInformation SSI
ON SBI.StudentRollNo = SSI.StudentRollNo;
```

```
mysql> SELECT SBI.StudentRollNo, SBI.StudentName, SSI.StudentMarksPercentage FROM StudentBasicInformation SBI INNER JOIN StudentSubjectInformation SSI ON SBI.StudentRol
lNo = SSI.StudentRollNo;
+---------------+-------------+------------------------+
| StudentRollNo | StudentName | StudentMarksPercentage |
+---------------+-------------+------------------------+
|             1 | Aman        |                     75 |
|             2 | Diksha      |                     88 |
|             3 | Fenil       |                     98 |
|             4 | Gayatri     |                     62 |
|             5 | Harsh       |                     70 |
|             6 | Heena       |                     88 |
|             7 | Jayant      |                     96 |
|             8 | Karan       |                     60 |
|             9 | Mandeep     |                     79 |
|            10 | Naina       |                     94 |
|            11 | Naresh      |                     86 |
|            12 | Payal       |                     59 |
|            13 | Rohan       |                     81 |
|            14 | Shantanu    |                     69 |
|            15 | Terence     |                     87 |
+---------------+-------------+------------------------+
15 rows in set (0.02 sec)
```

Left Join - Left joins can be used when we need to have all the rows of the left table, regardless of the matching condition. It is useful when we need to see all the data as well as their category. If there is no category for data don't remove it like inner join rather show it with null.

```
SELECT SBI.StudentRollNo, SBI.StudentName, SSI.ScholarshipCategory FROM StudentBasicInformation SBI LEFT JOIN SubjectScholarshipInformation SSI ON SBI.StudentRollNo = SSI.StudentRollNo;
```

```
mysql> SELECT SBI.StudentRollNo, SBI.StudentName, SSI.ScholarshipCategory FROM StudentBasicInformation SBI LEFT JOIN SubjectScholarshipInformation SSI ON SBI.StudentRol
lNo = SSI.StudentRollNo;
+---------------+-------------+---------------------+
| StudentRollNo | StudentName | ScholarshipCategory |
+---------------+-------------+---------------------+
|             1 | Aman        | NULL                |
|             2 | Diksha      | Category 3          |
|             3 | Fenil       | Category 1          |
|             4 | Gayatri     | NULL                |
|             5 | Harsh       | NULL                |
|             6 | Heena       | Category 3          |
|             7 | Jayant      | Category 1          |
|             8 | Karan       | NULL                |
|             9 | Mandeep     | NULL                |
|            10 | Naina       | Category 2          |
|            11 | Naresh      | Category 3          |
|            12 | Payal       | NULL                |
|            13 | Rohan       | NULL                |
|            14 | Shantanu    | NULL                |
|            15 | Terence     | Category 3          |
+---------------+-------------+---------------------+
15 rows in set (0.03 sec)
```

Right join - Right and left joins are nearly the same, just that left join shows all the table rows of the left table, whereas right join shows all the values of the right table. The above query result will be exactly the same as the left join because we swapped the left and right position in the table.

```
SELECT SBI.StudentRollNo, SBI.StudentName, SSI.ScholarshipCategory FROM SubjectScholarshipInformation SSI RIGHT JOIN StudentBasicInformation SBI ON SBI.StudentRollNo = SSI.StudentRollNo;
```

```
mysql> SELECT SBI.StudentRollNo, SBI.StudentName, SSI.ScholarshipCategory FROM SubjectScholarshipInformation SSI RIGHT JOIN StudentBasicInformation SBI ON SBI.StudentRo
llNo = SSI.StudentRollNo;
+---------------+-------------+---------------------+
| StudentRollNo | StudentName | ScholarshipCategory |
+---------------+-------------+---------------------+
|             1 | Aman        | NULL                |
|             2 | Diksha      | Category 3          |
|             3 | Fenil       | Category 1          |
|             4 | Gayatri     | NULL                |
|             5 | Harsh       | NULL                |
|             6 | Heena       | Category 3          |
|             7 | Jayant      | Category 1          |
|             8 | Karan       | NULL                |
|             9 | Mandeep     | NULL                |
|            10 | Naina       | Category 2          |
|            11 | Naresh      | Category 3          |
|            12 | Payal       | NULL                |
|            13 | Rohan       | NULL                |
|            14 | Shantanu    | NULL                |
|            15 | Terence     | Category 3          |
+---------------+-------------+---------------------+
15 rows in set (0.00 sec)
```

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

DELETE	DROP	TRUNCATE
The DELETE statement in SQL is a DML Command.	DROP statement is a DDL Command.	TRUNCATE command is a DDL operation.
It is used to delete one or more tuples of a table.	It is used to drop the whole table along with its structure.	It is used to delete all the rows in the table but the structure of table still remains.
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 EMPLOYEE;	TRUNCATE TABLE EMPLOYEE;

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 ScholarshipCategory, COUNT(*) AS COUNT FROM
SubjectScholarshipInformation GROUP BY ScholarshipCategory;
```

```
mysql> SELECT ScholarshipCategory, COUNT(*) AS COUNT FROM SubjectScholarshipInformation GROUP BY ScholarshipCategory;
+--------------------+-------+
| ScholarshipCategory | COUNT |
+--------------------+-------+
| Category 3         |     4 |
| Category 1         |     2 |
| Category 2         |     1 |
+--------------------+-------+
3 rows in set (0.03 sec)
```

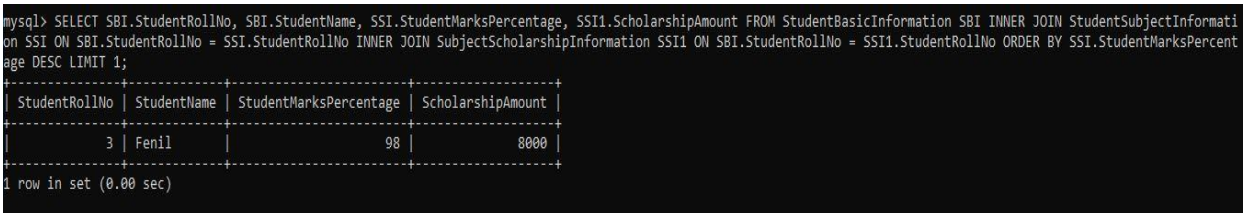
18. Along with the assignment no. 17 try to retrieve the maximum used scholarship category

```
SELECT ScholarshipCategory FROM SubjectScholarshipInformation GROUP BY
ScholarshipCategory LIMIT 1;
```

```
mysql> SELECT ScholarshipCategory FROM SubjectScholarshipInformation GROUP BY ScholarshipCategory LIMIT 1;
+--------------------+
| ScholarshipCategory |
+--------------------+
| Category 3         |
+--------------------+
1 row in set (0.03 sec)
```

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 SBI.StudentRollNo, SBI.StudentName, SSI.StudentMarksPercentage, SSI1.ScholarshipAmount FROM StudentBasicInformation SBI INNER JOIN StudentSubjectInformation SSI ON SBI.StudentRollNo = SSI.StudentRollNo INNER JOIN SubjectScholarshipInformation SSI1 ON SBI.StudentRollNo = SSI1.StudentRollNo ORDER BY SSI.StudentMarksPercentage DESC LIMIT 1;
```



```
mysql> SELECT SBI.StudentRollNo, SBI.StudentName, SSI.StudentMarksPercentage, SSI1.ScholarshipAmount FROM StudentBasicInformation SBI INNER JOIN StudentSubjectInformation SSI ON SBI.StudentRollNo = SSI.StudentRollNo INNER JOIN SubjectScholarshipInformation SSI1 ON SBI.StudentRollNo = SSI1.StudentRollNo ORDER BY SSI.StudentMarksPercentage DESC LIMIT 1;
+---------------+-------------+-----------------------+-------------------+
| StudentRollNo | StudentName | StudentMarksPercentage | ScholarshipAmount |
+---------------+-------------+-----------------------+-------------------+
|             3 | Fenil       |                    98 |              8000 |
+---------------+-------------+-----------------------+-------------------+
1 row in set (0.00 sec)
```

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

TRIGGERS	STORED PROCEDURES	VIEWS	FUNCTIONS
Triggers are stored procedures that runs automatically when various events haven (for eg: insert, update, delete)	Stored Procedures are a piece of SQL code meant to do some specific tasks.	A view is a virtual table based on the result set of an SQL query.	Functions are routines that accept parameters, perform an action and return the result of that action as a value (for eg: max(), avg(), count())
Triggers cannot return values.	They can return values.	It does not return a value instead it returns a table.	It can return a single scalar value or a result set.