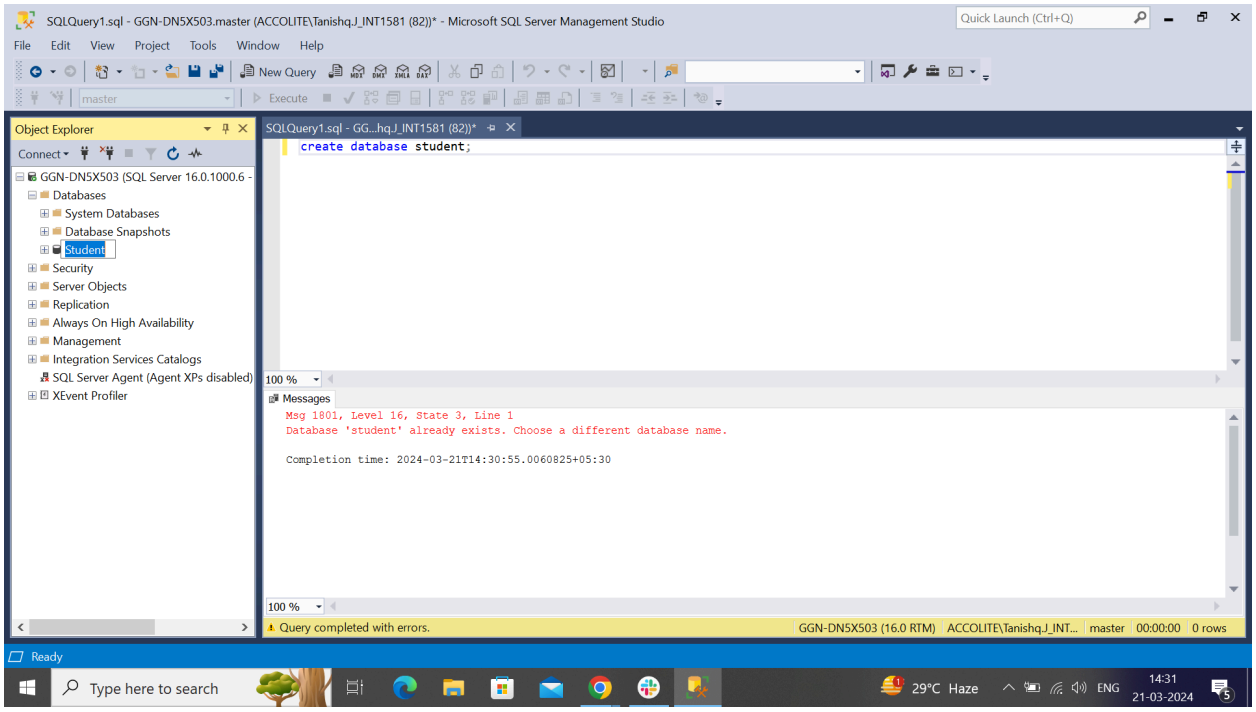
# SQL Assignment

1. Create a Database named Student.

   **Commands:**

   ```
   Create database student;
   Use student;
   ```

   **Snapshot:**




2. Create the following table under the Student Database:
   a. StudentBasicInformation

      Columns

      1. StudentName
      2. StudentSurname
      3. StudentRollNo
      4. StudentAddress
      5. Add more two basic columns of the name of your own

**Commands:**

```
create table studentBasicInformation(
StudentName varchar(25),
StudentSurname varchar(25),
StudentRollNo int,
StudentAddress varchar(255),
StudentFatherName varchar(25),
StudentMotherName varchar(25)
);
```

b. StudentAdmissionPaymentDetails

Columns

1. StudentRollNo
2. AmountPaid
3. AmountBalance
4. Add more three basic columns of the name of your own

**Commands:**

```
create table StudentAdmissionPaymentDetails(
StudentRollNo int,
AmoutnPaid int,
AmountBalance int,
FatherIncome int,
MotherIncome int,
FamilyIncome int
);
```

c. StudentSubjectInformation

Columns

1. SubjectOpted
2. StudentRollNo
3. SubjectTotalMarks
4. SubjectObtainedMarks
5. StudentMarksPercentage
6. Add more one columns of the name of your own

**Commands:**

```
Create table StudentSubjectInformation(
SubjectOpted varchar(25),
StudentRollNo int,
StudentTotalMarks int,
SubjectObtainedMarks int,
StudentMarksPercentage int,
SubjectCode int );
```

d. SubjectScholarshipInformation

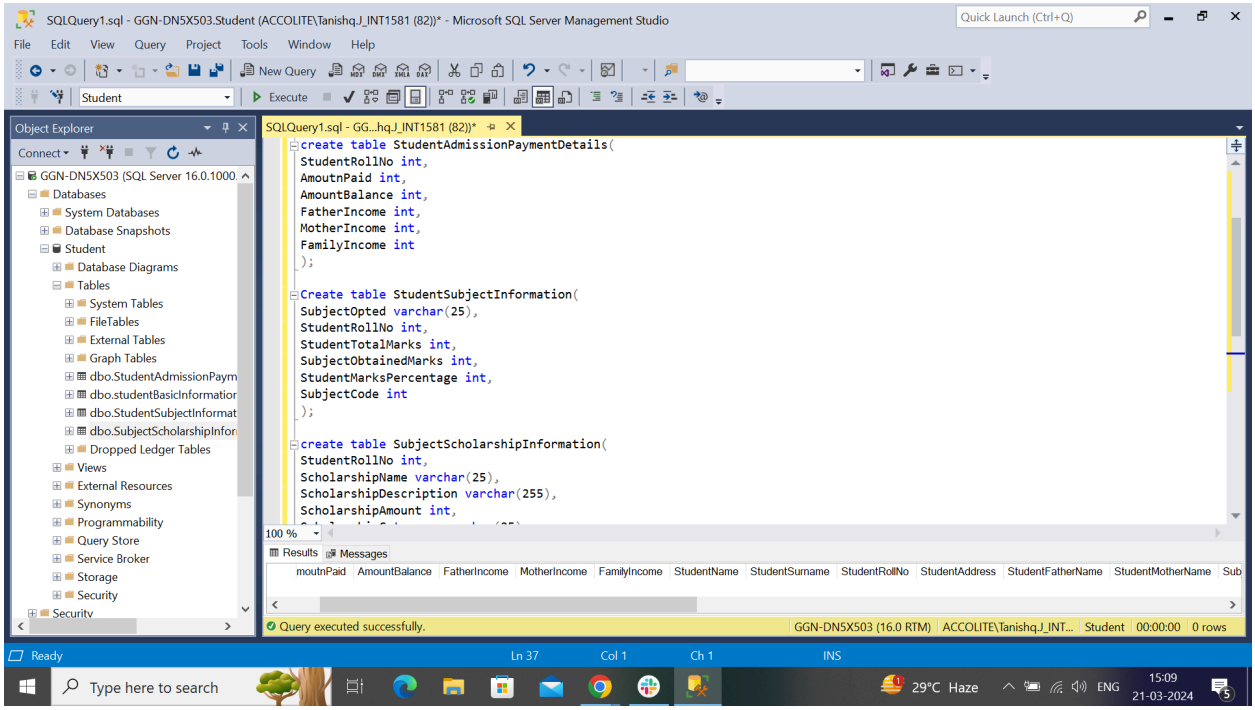
Columns

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

**Commands:**

```
create table SubjectScholarshipInformation(
StudentRollNo int,
ScholarshipName varchar(255),
ScholarshipDescription varchar(255),
ScholarshipAmount int,
ScholarshipCategory varchar(255),
PreviousScholarshipAmount int,
PreviousScholarshipName varchar(255)
);
```

**Snapshot:**



3. Insert at least 10 or more records in each and every table created and add snapshots of all the tables once the insertion is completed.

**Commands:**

```
INSERT INTO studentBasicInformation (StudentName, StudentSurname, StudentRollNo, StudentAddress, StudentFatherName, StudentMotherName)
VALUES
  ('Aarav', 'Sharma', 101, '123 Main St', 'Rajesh Sharma', 'Priya Sharma'),
  ('Isha', 'Verma', 102, '456 Elm Ave', 'Anil Verma', 'Sunita Verma'),
  ('Rahul', 'Patel', 103, '789 Oak Rd', 'Nitin Patel', 'Pooja Patel'),
  ('Ananya', 'Singh', 104, '567 Pine Ln', 'Vikas Singh', 'Ritu Singh'),
  ('Arjun', 'Gupta', 105, '890 Maple Ct', 'Rakesh Gupta', 'Anita Gupta'),
  ('Kavya', 'Joshi', 106, '432 Birch Dr', 'Amit Joshi', 'Sneha Joshi'),
  ('Aditya', 'Kumar', 107, '654 Cedar Blvd', 'Sanjay Kumar', 'Meera Kumar'),
  ('Neha', 'Rajput', 108, '987 Willow Ave', 'Rajendra Rajput', 'Sarita Rajput'),
  ('Aryan', 'Yadav', 109, '321 Spruce St', 'Ramesh Yadav', 'Savita Yadav'),
  ('Sneha', 'Choudhary', 110, '234 Fir Rd', 'Vijay Choudhary', 'Poonam Choudhary');

INSERT INTO StudentAdmissionPaymentDetails (StudentRollNo, AmoutnPaid, AmountBalance, FatherIncome, MotherIncome, FamilyIncome)
VALUES
  (101, 50000, 20000, 800000, 600000, 1400000),
  (102, 60000, 15000, 700000, 550000, 1250000),
  (103, 55000, 25000, 750000, 650000, 1400000),
  (104, 48000, 32000, 600000, 500000, 1100000),
  (105, 52000, 28000, 900000, 700000, 1600000),
  (106, 45000, 35000, 550000, 450000, 1000000),
  (107, 58000, 22000, 720000, 580000, 1300000),
  (108, 51000, 29000, 680000, 520000, 1200000),
  (109, 49000, 31000, 620000, 480000, 1100000),
  (110, 54000, 26000, 800000, 620000, 1420000);

INSERT INTO StudentSubjectInformation (SubjectOpted, StudentRollNo, StudentTotalMarks, SubjectObtainedMarks, StudentMarksPercentage, SubjectCode)
VALUES
  ('Mathematics', 101, 100, 85, 85, 101),
  ('Physics', 102, 100, 92, 92, 102),
  ('Chemistry', 103, 100, 78, 78, 103),
  ('English', 104, 100, 88, 88, 104),
  ('History', 105, 100, 70, 70, 105),
  ('Biology', 106, 100, 95, 95, 106),
  ('Computer Science', 107, 100, 80, 80, 107),
  ('Geography', 108, 100, 75, 75, 108),
```

```
('Economics', 109, 100, 90, 90, 109),
('Physical Education', 110, 100, 98, 98, 110);

INSERT INTO SubjectScholarshipInformation (StudentRollNo, ScholarshipName, ScholarshipDescription, ScholarshipAmount, ScholarshipCategory, PreviousScholarshipAmount, PreviousScholarshipName) VALUES
(101, 'Merit Scholarship', 'Awarded to top-performing students based on academic excellence.', 10000, 'Academic', 8000, 'Previous Merit Scholarship'),
(102, 'SC/ST Scholarship', 'Reserved for students from Scheduled Castes (SC) or Scheduled Tribes (ST) backgrounds.', 5000, 'Social Welfare', 4000, 'Previous SC/ST Scholarship'),
(103, 'Minority Scholarship', 'For students belonging to minority communities (e.g., Muslims, Christians, Sikhs, etc.).', 7500, 'Minority Affairs', 6000, 'Previous Minority Scholarship'),
(104, 'Sports Scholarship', 'Awarded to outstanding athletes representing the school or state.', 12000, 'Sports', 10000, 'Previous Sports Scholarship'),
(105, 'OBC Scholarship', 'Reserved for students from Other Backward Classes (OBC) backgrounds.', 6000, 'Social Welfare', 5000, 'Previous OBC Scholarship'),
(106, 'Disability Scholarship', 'For differently-abled students (physically challenged, visually impaired, etc.).', 8000, 'Special Needs', 7000, 'Previous Disability Scholarship'),
(107, 'Rural Scholarship', 'Aimed at students from rural areas to promote education.', 4500, 'Rural Development', 3500, 'Previous Rural Scholarship'),
(108, 'Cultural Scholarship', 'Given to students excelling in cultural activities (music, dance, arts, etc.).', 9000, 'Cultural', 7500, 'Previous Cultural Scholarship'),
(109, 'NTSE Scholarship', 'Awarded to students who clear the National Talent Search Examination (NTSE).', 15000, 'Academic', 12000, 'Previous NTSE Scholarship'),
(110, 'Single Girl Child Scholarship', 'Exclusively for single girl children in the family.', 5500, 'Gender Empowerment', 4000, 'Previous Single Girl Child Scholarship');
```

**Snapshots:**

| | StudentName | StudentSurname | StudentRollNo | StudentAddress | StudentFatherName | StudentMotherName |
|---|---|---|---|---|---|---|
| 1 | Aarav | Sharma | 101 | 123 Main St | Rajesh Sharma | Priya Sharma |
| 2 | Isha | Verma | 102 | 456 Elm Ave | Anil Verma | Sunita Verma |
| 3 | Rahul | Patel | 103 | 789 Oak Rd | Nitin Patel | Pooja Patel |
| 4 | Ananya | Singh | 104 | 567 Pine Ln | Vikas Singh | Ritu Singh |
| 5 | Arjun | Gupta | 105 | 890 Maple Ct | Rakesh Gupta | Anita Gupta |
| 6 | Kavya | Joshi | 106 | 432 Birch Dr | Amit Joshi | Sneha Joshi |
| 7 | Aditya | Kumar | 107 | 654 Cedar Blvd | Sanjay Kumar | Meera Kumar |
| 8 | Neha | Rajput | 108 | 987 Willow Ave | Rajendra Rajput | Sarita Rajput |
| 9 | Aryan | Yadav | 109 | 321 Spruce St | Ramesh Yadav | Savita Yadav |
| 10 | Sneha | Choudhary | 110 | 234 Fir Rd | Vijay Choudhary | Poonam Choudhary |

| | StudentRollNo | AmoutnPaid | AmountBalance | FatherIncome | MotherIncome | FamilyIncome |
|---|---|---|---|---|---|---|
| 1 | 101 | 50000 | 20000 | 800000 | 600000 | 1400000 |
| 2 | 102 | 60000 | 15000 | 700000 | 550000 | 1250000 |
| 3 | 103 | 55000 | 25000 | 750000 | 650000 | 1400000 |
| 4 | 104 | 48000 | 32000 | 600000 | 500000 | 1100000 |
| 5 | 105 | 52000 | 28000 | 900000 | 700000 | 1600000 |
| 6 | 106 | 45000 | 35000 | 550000 | 450000 | 1000000 |
| 7 | 107 | 58000 | 22000 | 720000 | 580000 | 1300000 |
| 8 | 108 | 51000 | 29000 | 680000 | 520000 | 1200000 |
| 9 | 109 | 49000 | 31000 | 620000 | 480000 | 1100000 |
| 10 | 110 | 54000 | 26000 | 800000 | 620000 | 1420000 |

| | SubjectOpted | StudentRollNo | StudentTotalMarks | SubjectObtainedMarks | StudentMarksPercentage | SubjectCode |
|---|---|---|---|---|---|---|
| 1 | Mathematics | 101 | 100 | 85 | 85 | 101 |
| 2 | Physics | 102 | 100 | 92 | 92 | 102 |
| 3 | Chemistry | 103 | 100 | 78 | 78 | 103 |
| 4 | English | 104 | 100 | 88 | 88 | 104 |
| 5 | History | 105 | 100 | 70 | 70 | 105 |
| 6 | Biology | 106 | 100 | 95 | 95 | 106 |
| 7 | Computer S... | 107 | 100 | 80 | 80 | 107 |
| 8 | Geography | 108 | 100 | 75 | 75 | 108 |
| 9 | Economics | 109 | 100 | 90 | 90 | 109 |
| 10 | Physical Ed... | 110 | 100 | 98 | 98 | 110 |

| | StudentRollNo | ScholarshipName | ScholarshipDescription | ScholarshipAmount | ScholarshipCategory | PreviousScholarshipAmount | PreviousScholarshipName |
|---|---|---|---|---|---|---|---|
| 1 | 101 | Merit Scholarship | Awarded to top-performing students based on acade... | 10000 | Academic | 8000 | Previous Merit Scholarship |
| 2 | 102 | SC/ST Scholarship | Reserved for students from Scheduled Castes (SC) ... | 5000 | Social Welfare | 4000 | Previous SC/ST Scholarship |
| 3 | 103 | Minority Scholarship | For students belonging to minority communities (e.g., ... | 7500 | Minority Affairs | 6000 | Previous Minority Scholarship |
| 4 | 104 | Sports Scholarship | Awarded to outstanding athletes representing the sc... | 12000 | Sports | 10000 | Previous Sports Scholarship |
| 5 | 105 | OBC Scholarship | Reserved for students from Other Backward Classes ... | 6000 | Social Welfare | 5000 | Previous OBC Scholarship |
| 6 | 106 | Disability Scholarship | For differently-abled students (physically challenged, ... | 8000 | Special Needs | 7000 | Previous Disability Scholarship |
| 7 | 107 | Rural Scholarship | Aimed at students from rural areas to promote educa... | 4500 | Rural Development | 3500 | Previous Rural Scholarship |
| 8 | 108 | Cultural Scholarship | Given to students excelling in cultural activities (music... | 9000 | Cultural | 7500 | Previous Cultural Scholarship |
| 9 | 109 | NTSE Scholarship | Awarded to students who clear the National Talent S... | 15000 | Academic | 12000 | Previous NTSE Scholarship |
| 10 | 110 | Single Girl Child Scholarship | Exclusively for single girl children in the family. | 5500 | Gender Empowerment | 4000 | Previous Single Girl Child Scholarship |

4. 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 and add snapshots of all the tables post update.

   **Commands:**

```
UPDATE StudentAdmissionPaymentDetails
SET
 AmoutnPaid = 50500,
   AmountBalance = 20400,
   FatherIncome = 806000,
   MotherIncome = 60800,
   FamilyIncome = 140200
WHERE StudentRollNo = 101;

UPDATE StudentAdmissionPaymentDetails
SET
   AmoutnPaid = 60200,
   AmountBalance = 15400,
   FatherIncome = 730000,
   MotherIncome = 555000,
   FamilyIncome = 1258000
WHERE StudentRollNo = 102;

UPDATE StudentAdmissionPaymentDetails
SET
   AmoutnPaid = 55200,
   AmountBalance = 25006,
   FatherIncome = 750400,
   MotherIncome = 650600,
   FamilyIncome = 1400300
WHERE StudentRollNo = 103;

UPDATE StudentAdmissionPaymentDetails
SET
   AmoutnPaid = 48000,
   AmountBalance = 32030,
   FatherIncome = 600000,
   MotherIncome = 500500,
   FamilyIncome = 1100000
WHERE StudentRollNo = 104;

UPDATE StudentAdmissionPaymentDetails
SET
   AmoutnPaid = 52000,
   AmountBalance = 28000,
```

```
    FatherIncome = 900030,
    MotherIncome = 700200,
    FamilyIncome = 1640000
WHERE StudentRollNo = 105;
```

**Snapshots:**

**Unmodified table:**

| | StudentRollNo | AmoutnPaid | AmountBalance | FatherIncome | MotherIncome | FamilyIncome |
|---|---|---|---|---|---|---|
| 1 | 101 | 50000 | 20000 | 800000 | 600000 | 1400000 |
| 2 | 102 | 60000 | 15000 | 700000 | 550000 | 1250000 |
| 3 | 103 | 55000 | 25000 | 750000 | 650000 | 1400000 |
| 4 | 104 | 48000 | 32000 | 600000 | 500000 | 1100000 |
| 5 | 105 | 52000 | 28000 | 900000 | 700000 | 1600000 |
| 6 | 106 | 45000 | 35000 | 550000 | 450000 | 1000000 |
| 7 | 107 | 58000 | 22000 | 720000 | 580000 | 1300000 |
| 8 | 108 | 51000 | 29000 | 680000 | 520000 | 1200000 |
| 9 | 109 | 49000 | 31000 | 620000 | 480000 | 1100000 |
| 10 | 110 | 54000 | 26000 | 800000 | 620000 | 1420000 |

**Modified table:**

| | StudentRollNo | AmoutnPaid | AmountBalance | FatherIncome | MotherIncome | FamilyIncome |
|---|---|---|---|---|---|---|
| 1 | 101 | 50500 | 20400 | 806000 | 60800 | 140200 |
| 2 | 102 | 60200 | 15400 | 730000 | 555000 | 1258000 |
| 3 | 103 | 55200 | 25006 | 750400 | 650600 | 1400300 |
| 4 | 104 | 48000 | 32030 | 600000 | 500500 | 1100000 |
| 5 | 105 | 52000 | 28000 | 900030 | 700200 | 1640000 |
| 6 | 106 | 45000 | 35000 | 550000 | 450000 | 1000000 |
| 7 | 107 | 58000 | 22000 | 720000 | 580000 | 1300000 |
| 8 | 108 | 51000 | 29000 | 680000 | 520000 | 1200000 |
| 9 | 109 | 49000 | 31000 | 620000 | 480000 | 1100000 |
| 10 | 110 | 54000 | 26000 | 800000 | 620000 | 1420000 |

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

**Commands:**

```
select *from studentBasicInformation sbi
JOIN SubjectScholarshipInformation ssi ON sbi.StudentRollNo=ssi.StudentRollNo
WHERE ssi.ScholarshipAmount>5000;
```

**Snapshots:**

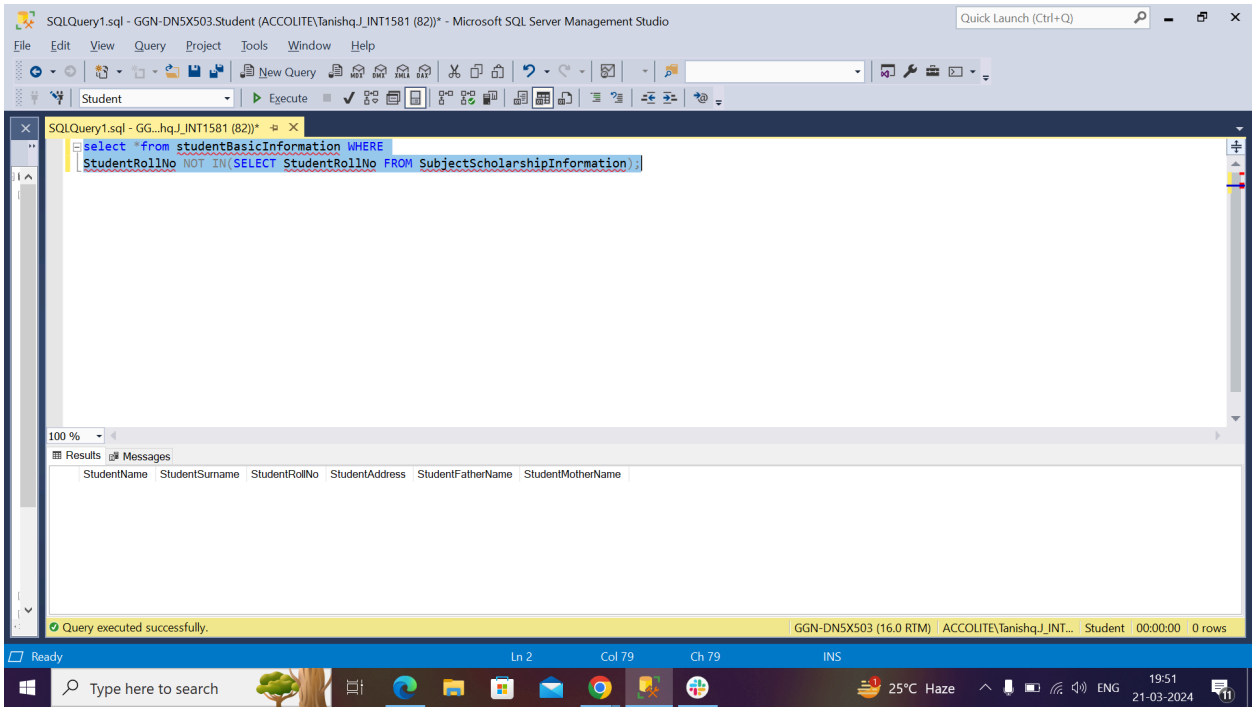
| | StudentName | StudentSurname | StudentRollNo | StudentAddress | StudentFatherName | StudentMotherName | StudentRollNo | ScholarshipName | ScholarshipDescription | ScholarshipAmount |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Aarav | Sharma | 101 | 123 Main St | Rajesh Sharma | Priya Sharma | 101 | Merit Scholarship | Awarded to top-performing students based on acade... | 10000 |
| 2 | Rahul | Patel | 103 | 789 Oak Rd | Nitin Patel | Pooja Patel | 103 | Minority Scholarship | For students belonging to minority communities (e.g., ... | 7500 |
| 3 | Ananya | Singh | 104 | 567 Pine Ln | Vikas Singh | Ritu Singh | 104 | Sports Scholarship | Awarded to outstanding athletes representing the sc... | 12000 |
| 4 | Arjun | Gupta | 105 | 890 Maple Ct | Rakesh Gupta | Anita Gupta | 105 | OBC Scholarship | Reserved for students from Other Backward Classes ... | 6000 |
| 5 | Kavya | Joshi | 106 | 432 Birch Dr | Amit Joshi | Sneha Joshi | 106 | Disability Scholarship | For differently-abled students (physically challenged, ... | 8000 |
| 6 | Neha | Rajput | 108 | 987 Willow Ave | Rajendra Rajput | Sarita Rajput | 108 | Cultural Scholarship | Given to students excelling in cultural activities (music... | 9000 |
| 7 | Aryan | Yadav | 109 | 321 Spruce St | Ramesh Yadav | Savita Yadav | 109 | NTSE Scholarship | Awarded to students who clear the National Talent S... | 15000 |
| 8 | Sneha | Choudhary | 110 | 234 Fir Rd | Vijay Choudhary | Poonam Choudhary | 110 | Single Girl Child Scholarship | Exclusively for single girl children in the family. | 5500 |

6. Select the students who opted for scholarship but did not get the scholarship.

**Commands:**

```
select *from studentBasicInformation WHERE
StudentRollNo NOT IN(SELECT StudentRollNo FROM
SubjectScholarshipInformation);
```

**Snapshots:**



7. Fill in data for the percentage column i.e. StudentMarksPercentage in the table 32StudentSubjectInformation by creating and using the stored procedure created.

**Commands:**

```
GO
CREATE PROCEDURE
CalculatePercent AS
BEGIN
UPDATE StudentSubjectInformation
SET StudentMarksPercentage=(SubjectObtainedMarks*100)/StudentTotalMarks;
END;
EXEC CalculatePercent
Select *from StudentSubjectInformation
```

**Snapshots:**

| | SubjectOpted | StudentRollNo | StudentTotalMarks | SubjectObtainedMarks | StudentMarksPercentage | SubjectCode |
|---|---|---|---|---|---|---|
| 1 | Mathematics | 101 | 100 | 85 | 85 | 101 |
| 2 | Physics | 102 | 100 | 92 | 92 | 102 |
| 3 | Chemistry | 103 | 100 | 78 | 78 | 103 |
| 4 | English | 104 | 100 | 88 | 88 | 104 |
| 5 | History | 105 | 100 | 70 | 70 | 105 |
| 6 | Biology | 106 | 100 | 95 | 95 | 106 |
| 7 | Computer Science | 107 | 100 | 80 | 80 | 107 |
| 8 | Geography | 108 | 100 | 75 | 75 | 108 |
| 9 | Economics | 109 | 100 | 90 | 90 | 109 |
| 10 | Physical Education | 110 | 100 | 98 | 98 | 110 |

8. 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.

**Commands:**

```
GO
CREATE PROCEDURE
UpdateScholarshipCategoryOnPercentageBasis AS
BEGIN
        UPDATE SubjectScholarshipInformation
        SET ScholarshipCategory = CASE
                WHEN (SELECT StudentMarksPercentage FROM
StudentSubjectInformation WHERE StudentSubjectInformation.StudentRollNo =
SubjectScholarshipInformation.StudentRollNo) >= 90 THEN 'Category A'
                WHEN (SELECT StudentMarksPercentage FROM
StudentSubjectInformation WHERE StudentSubjectInformation.StudentRollNo =
SubjectScholarshipInformation.StudentRollNo) >= 80 THEN 'Category B'
                WHEN (SELECT StudentMarksPercentage FROM
StudentSubjectInformation WHERE StudentSubjectInformation.StudentRollNo =
SubjectScholarshipInformation.StudentRollNo) >= 70 THEN 'Category C'
                ELSE 'Category D'
```

```
        END4
END;
EXEC UpdateScholarshipCategoryOnPercentageBasis;
```

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

**Commands:**

```
CREATE VIEW StudentBalanceView AS
SELECT
sb.StudentName,sb.StudentSurname,sb.StudentRollNo,sb.StudentAddre
ss,spd.AmountBalance
FROM StudentBasicInformation sb
JOIN StudentAdmissionPaymentDetails spd ON sb.StudentRollNo =
spd.StudentRollNo;
```

**Snapshots:**

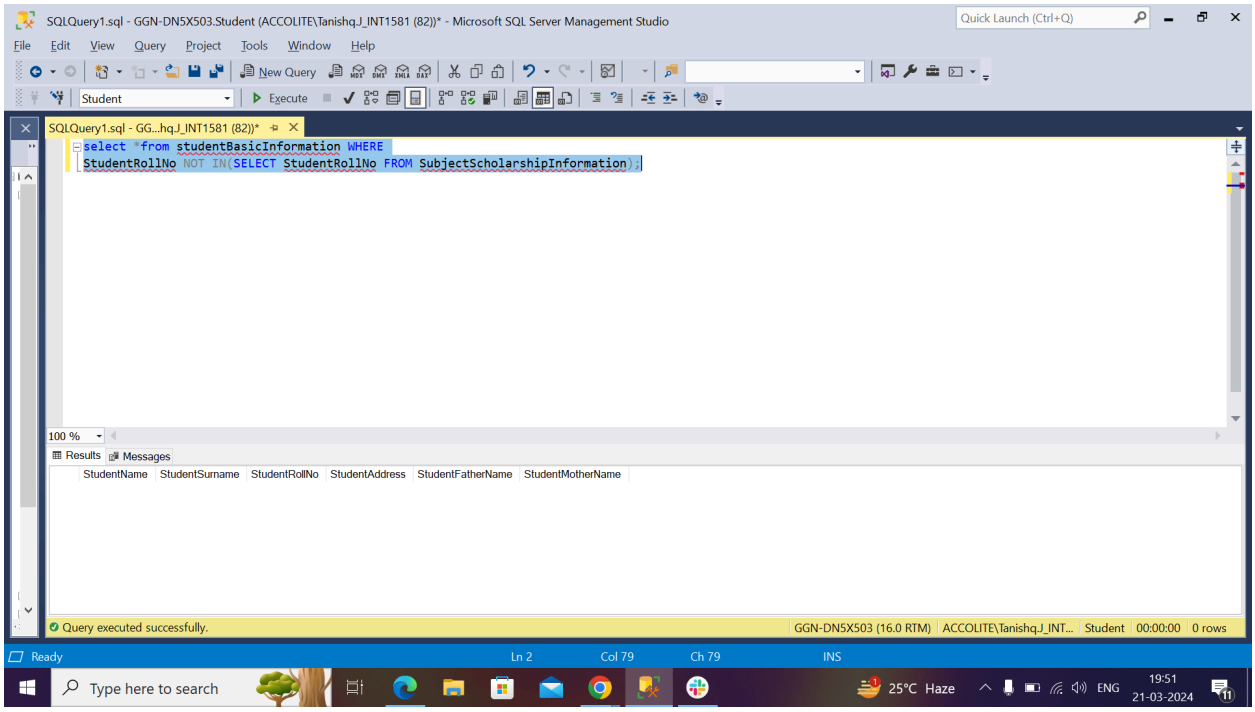
| | StudentName | StudentSurname | StudentRollNo | StudentAddress | AmountBalance |
|---|---|---|---|---|---|
| 1 | Aarav | Sharma | 101 | 123 Main St | 20400 |
| 2 | Isha | Verma | 102 | 456 Elm Ave | 15400 |
| 3 | Rahul | Patel | 103 | 789 Oak Rd | 25006 |
| 4 | Ananya | Singh | 104 | 567 Pine Ln | 32030 |
| 5 | Arjun | Gupta | 105 | 890 Maple Ct | 28000 |
| 6 | Kavya | Joshi | 106 | 432 Birch Dr | 35000 |
| 7 | Aditya | Kumar | 107 | 654 Cedar Blvd | 22000 |
| 8 | Neha | Rajput | 108 | 987 Willow Ave | 29000 |
| 9 | Aryan | Yadav | 109 | 321 Spruce St | 31000 |
| 10 | Sneha | Choudhary | 110 | 234 Fir Rd | 26000 |

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

**Commands:**

```
select *from studentBasicInformation WHERE
StudentRollNo NOT IN(SELECT StudentRollNo FROM
SubjectScholarshipInformation);
```

**Snapshots:**




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

**Commands:**

```
GO
CREATE PROCEDURE GetStudentBalance
@RollNo INT
AS
BEGIN
SELECT AmountBalance
FROM StudentAdmissionPaymentDetails
WHERE StudentRollNo = @RollNo;
END;
EXEC GetStudentBalance @RollNo = 425
EXEC GetStudentBalance @RollNo = 423
EXEC GetStudentBalance @RollNo = 428
```

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

**Commands:**

```
SELECT *
FROM StudentBasicInformation
WHERE StudentRollNo IN (SELECT TOP 5 StudentRollNo FROM
StudentSubjectInformation ORDER BY StudentMarksPercentage DESC);
```

**Snapshots:**

| | StudentName | StudentSurname | StudentRollNo | StudentAddress | StudentFatherName | StudentMotherName |
|---|---|---|---|---|---|---|
| 1 | Isha | Verma | 102 | 456 Elm Ave | Anil Verma | Sunita Verma |
| 2 | Ananya | Singh | 104 | 567 Pine Ln | Vikas Singh | Ritu Singh |
| 3 | Kavya | Joshi | 106 | 432 Birch Dr | Amit Joshi | Sneha Joshi |
| 4 | Aryan | Yadav | 109 | 321 Spruce St | Ramesh Yadav | Savita Yadav |
| 5 | Sneha | Choudhary | 110 | 234 Fir Rd | Vijay Choudhary | Poonam Choudhary |

13. Write a query using Group By to retrieve roll numbers on the basis of ScholarshipName.
**Commands:**

```
SELECT ScholarshipName, COUNT(StudentRollNo) AS
NumberOfStudents
FROM SubjectScholarshipInformation
GROUP BY ScholarshipName;
```

**Snapshots:**

| | ScholarshipName | NumberOfStudents |
|---|---|---|
| 1 | Cultural Scholarship | 1 |
| 2 | Disability Scholarship | 1 |
| 3 | Merit Scholarship | 1 |
| 4 | Minority Scholarship | 1 |
| 5 | NTSE Scholarship | 1 |
| 6 | OBC Scholarship | 1 |
| 7 | Rural Scholarship | 1 |
| 8 | SC/ST Scholarship | 1 |
| 9 | Single Girl Child Scholarship | 1 |
| 10 | Sports Scholarship | 1 |

14. Try to use the three types of join - INNER, LEFT, RIGHT 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)
**Commands:**
**INNER JOIN** : Retrieves records that have matching values in both tables.

```
SELECT sb.StudentName, ssi.ScholarshipName
FROM StudentBasicInformation sb
```

INNER JOIN SubjectScholarshipInformation ssi ON sb.StudentRollNo = ssi.StudentRollNo;

| | StudentName | ScholarshipName |
|---|---|---|
| 1 | Aarav | Merit Scholarship |
| 2 | Isha | SC/ST Scholarship |
| 3 | Rahul | Minority Scholarship |
| 4 | Ananya | Sports Scholarship |
| 5 | Arjun | OBC Scholarship |
| 6 | Kavya | Disability Scholarship |
| 7 | Aditya | Rural Scholarship |
| 8 | Neha | Cultural Scholarship |
| 9 | Aryan | NTSE Scholarship |
| 10 | Sneha | Single Girl Child Scholarship |

**LEFT JOIN** : Retrieves all record from left table and the matched records from the right table.

```
SELECT sb.*, ssi.ScholarshipName
FROM StudentBasicInformation sb
LEFT JOIN SubjectScholarshipInformation ssi ON sb.StudentRollNo =
ssi.StudentRollNo;
```

| | StudentName | StudentSurname | StudentRollNo | StudentAddress | StudentFatherName | StudentMotherName | ScholarshipName |
|---|---|---|---|---|---|---|---|
| 1 | Aarav | Sharma | 101 | 123 Main St | Rajesh Sharma | Priya Sharma | Merit Scholarship |
| 2 | Isha | Verma | 102 | 456 Elm Ave | Anil Verma | Sunita Verma | SC/ST Scholarship |
| 3 | Rahul | Patel | 103 | 789 Oak Rd | Nitin Patel | Pooja Patel | Minority Scholarship |
| 4 | Ananya | Singh | 104 | 567 Pine Ln | Vikas Singh | Ritu Singh | Sports Scholarship |
| 5 | Arjun | Gupta | 105 | 890 Maple Ct | Rakesh Gupta | Anita Gupta | OBC Scholarship |
| 6 | Kavya | Joshi | 106 | 432 Birch Dr | Amit Joshi | Sneha Joshi | Disability Scholarship |
| 7 | Aditya | Kumar | 107 | 654 Cedar Blvd | Sanjay Kumar | Meera Kumar | Rural Scholarship |
| 8 | Neha | Rajput | 108 | 987 Willow Ave | Rajendra Rajput | Sarita Rajput | Cultural Scholarship |
| 9 | Aryan | Yadav | 109 | 321 Spruce St | Ramesh Yadav | Savita Yadav | NTSE Scholarship |
| 10 | Sneha | Choudhary | 110 | 234 Fir Rd | Vijay Choudhary | Poonam Choudhary | Single Girl Child Scholarship |

**RIGHT JOIN** : Retrieves all record from righttable and the matched records from the left table

```
SELECT sb.StudentName, ssi.*
FROM StudentBasicInformation sb
RIGHT JOIN SubjectScholarshipInformation ssi ON sb.StudentRollNo
= ssi.StudentRollNo;
```

| | StudentName | StudentRollNo | ScholarshipName | ScholarshipDescription | ScholarshipAmount | ScholarshipCategory | PreviousScholarshipAmount |
|---|---|---|---|---|---|---|---|
| 1 | Aarav | 101 | Merit Scholarship | Awarded to top-performing students based on acade... | 10000 | Academic | 8000 |
| 2 | Isha | 102 | SC/ST Scholarship | Reserved for students from Scheduled Castes (SC) ... | 5000 | Social Welfare | 4000 |
| 3 | Rahul | 103 | Minority Scholarship | For students belonging to minority communities (e.g., ... | 7500 | Minority Affairs | 6000 |
| 4 | Ananya | 104 | Sports Scholarship | Awarded to outstanding athletes representing the sc... | 12000 | Sports | 10000 |
| 5 | Arjun | 105 | OBC Scholarship | Reserved for students from Other Backward Classes ... | 6000 | Social Welfare | 5000 |
| 6 | Kavya | 106 | Disability Scholarship | For differently-abled students (physically challenged, ... | 8000 | Special Needs | 7000 |
| 7 | Aditya | 107 | Rural Scholarship | Aimed at students from rural areas to promote educa... | 4500 | Rural Development | 3500 |
| 8 | Neha | 108 | Cultural Scholarship | Given to students excelling in cultural activities (music... | 9000 | Cultural | 7500 |
| 9 | Aryan | 109 | NTSE Scholarship | Awarded to students who clear the National Talent S... | 15000 | Academic | 12000 |
| 10 | Sneha | 110 | Single Girl Child Scholarship | Exclusively for single girl children in the family. | 5500 | Gender Empowerment | 4000 |

15. Mention the differences between delete, drop and truncate commands with examples.
**Ans:**

**DELETE:**

a. **Purpose**: Used to remove specific rows from a table.
b. **Language**: Data Manipulation Language (DML).
c. **Usage**:
    i. To delete all rows: `DELETE FROM <table_name>;`
    ii. To delete rows based on a condition: `DELETE FROM <table_name> WHERE <condition>;`
d. **Rollback**: Can be rolled back using the `ROLLBACK` command.
e. **Efficiency**: Less efficient for large tables due to manual specification of conditions.

**DROP:**

f. **Purpose**: Removes the entire table along with its structure.
g. **Language**: Data Definition Language (DDL).
h. **Usage**: To drop a table: `DROP TABLE <table_name>;`
i. **Rollback**: Cannot be rolled back (auto-commits).
j. **Efficiency**: Depends on the size of the object being dropped.

**TRUNCATE:**

k. **Purpose**: Deletes all rows from a table, leaving the table structure intact.
l. **Language**: Data Definition Language (DDL).
m. **Usage**: To truncate a table: `TRUNCATE TABLE <table_name>;`
n. **Rollback**: Cannot be rolled back.

o. **Efficiency**: More efficient for large tables as it removes all data in one step.

16. Get the count of the Scholarship category which is highly availed by the students, i.e. get the count of the total number of students corresponding to each scholarships category.
    **Commands:**

```
SELECT ScholarshipCategory, COUNT(StudentRollNo) AS
CountofStudents
FROM SubjectScholarshipInformation
GROUP BY ScholarshipCategory
ORDER BY CountofStudents DESC;
```

    **Snapshots:**

| | ScholarshipCategory | CountofStudents |
|---|---|---|
| 1 | Academic | 2 |
| 2 | Social Welfare | 2 |
| 3 | Special Needs | 1 |
| 4 | Sports | 1 |
| 5 | Cultural | 1 |
| 6 | Gender Empowerment | 1 |
| 7 | Minority Affairs | 1 |
| 8 | Rural Development | 1 |

17. Along with question no.16 try to retrieve the maximum used scholarship category.
    **Commands:**

```
SELECT TOP 1 ScholarshipCategory, COUNT(StudentRollNo) AS
CountofStudents
FROM SubjectScholarshipInformation
GROUP BY ScholarshipCategory
ORDER BY CountofStudents DESC;
```

    **Snapshots:**

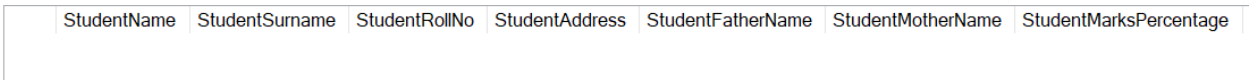
| | ScholarshipCategory | CountofStudents |
|---|---|---|
| 1 | Academic | 2 |

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

**Commands:**

```
SELECT sb.*, ssi.StudentMarksPercentage
FROM StudentBasicInformation sb
JOIN StudentSubjectInformation ssi ON sb.StudentRollNo =
ssi.StudentRollNo
WHERE ssi.StudentMarksPercentage = (SELECT
MAX(StudentMarksPercentage) FROM StudentSubjectInformation)
AND sb.StudentRollNo IN (SELECT StudentRollNo FROM
SubjectScholarshipInformation WHERE ScholarshipAmount = (SELECT
MAX (ScholarshipAmount) FROM SubjectScholarshipInformation));
```

**Snapshots:**



| StudentName | StudentSurname | StudentRollNo | StudentAddress | StudentFatherName | StudentMotherName | StudentMarksPercentage |
|---|---|---|---|---|---|---|

19. Difference between the Triggers, Stored Procedures, Views and Functions with examples.

**Ans:**

**<u>Stored Procedures</u>**

- A stored procedure is a collection of SQL statements written to perform specific tasks.
- It promotes code reusability and saves development time.

Example:-

```
CREATE PROCEDURE CalculateTotalSales
AS
BEGIN
 SELECT SUM(SalesAmount) AS TotalSales FROM SalesTable;
END;
```

**<u>Triggers</u>**

A trigger is a special kind of procedure that automatically executes when specific events (e.g., INSERT, UPDATE, DELETE) occur in a table.

Example:-

```
CREATE TRIGGER UpdateStock
ON Products
AFTER INSERT
AS
BEGIN
UPDATE StockTable
SET Quantity = Quantity - inserted.Quantity
WHERE ProductID = inserted.ProductID;
END;
```

**Views**

- A view is a virtual table created by querying one or more base tables.
- It simplifies complex queries and provides a customized view of data.

Example:-

```
CREATE VIEW HighRevenueProducts AS
SELECT ProductName, UnitPrice
FROM Products
WHERE UnitPrice > 100;
```

**Functions**

- A function is a reusable piece of code that performs a specific task and can return a value.
- It can be called explicitly.

Example:-

```
CREATE FUNCTION CalculateDiscount(@TotalAmount DECIMAL)
RETURNS DECIMAL
AS
BEGIN
 RETURN @TotalAmount * 0.1; -- 10% discount
END;
```

20. Difference between clustered and non-clustered indexes.

**Ans:**

**Clustered Index:**

- A clustered index defines the physical order in which data is stored within a table.
- Uniqueness: There can be only one clustered index per table.
- Primary Key: When you define a primary key constraint on a column, it automatically creates a clustered index on that column.
- Storage: The data pages are stored in the leaf nodes of the clustered index.
- Order: Rows are physically ordered based on the indexed column.

**Non-Clustered Index**:

- A non-clustered index stores data at one location and index entries at another.
- Storage: Data pages are not stored in the leaf nodes of the non-clustered index.
- Multiple Indexes: You can have multiple non-clustered indexes per table.
- Lookup: Non-clustered indexes involve an extra lookup step to find the actual data.

**Thank you. All The Best. Enjoy The Assignment.**