**SQL Queries**

1. **Create Student Database**

create database Student;

use Student;

1. **Create the following table under the Student Database:**
   1. **StudentBasicInformation**
      1. **Columns**
         1. **StudentName**
         2. **StudentSurname**
         3. **StudentRollNo**
         4. **StudentAddress**
         5. **Add more three basic columns of the name of your own**
   2. **StudentAdmissionPaymentDetails**
      1. **Columns**
         1. **StudentRollNo**
         2. **AmountPaid**
         3. **AmountBalance**
         4. **Add more four basic columns of the name of your own**
   3. **StudentSubjectInformation**
      1. **Columns**
         1. **SubjectOpted**
         2. **StudentRollNo**
         3. **SubjectTotalMarks**
         4. **SubjectObtainedMarks**
         5. **StudentMarksPercentage**
         6. **Add more one columns of the name of your own**
   4. **SubjectScholarshipInformation**
      1. **Columns**
         1. **StudentRollNo**
         2. **ScholarshipName**
         3. **ScholarshipDescription**
         4. **ScholarshipAmount**
         5. **ScholarshipCategory**
         6. **Add more two columns of the name of your own**

Create table StudentBasicInformation(

StudentName varchar(255) NOT NULL,

StudentSurName varchar(255),

StudentRollNo int primary key,

StudentAddress varchar(255) NOT NULL,

StudentContactNo character(10) NOT NULL,

StudentEmail varchar(50),

StudentClass varchar(6));

describe StudentBasicInformation;

Create table StudentAdmissionPaymentDetails(

StudentPaymentId varchar(20) primary key,

StudentRollNo int NOT NULL,

AmountPaid int NOT NULL,

AmmountBalanace int NOT NULL,

PreviousPaymentId varchar(20) NOT NULL,

PaymentType varchar(10) not null,

AdmissionType varchar(20) not null,

FOREIGN KEY (StudentRollNo) REFERENCES StudentBasicInformation(StudentRollNo)

);

describe StudentAdmissionPaymentDetails;

Create table StudentSubjectInformation(

SubjectOpted varchar(20) not null,

StudentRollNo int not null,

StudentTotalMarks int not null,

StudentObtainedMarks int not null,

StudentMarksPercentage float(5),

SubjectCategory varchar(20) not null,

primary key (SubjectOpted,StudentRollNo),

Foreign key (StudentRollNo) references StudentBasicInformation(StudentRollNo)

);

describe StudentSubjectInformation;

Create table SubjectScholarshipInformation(

StudentRollNo int not null,

ScholarshipName varchar(255) not null,

ScholarshipDescription varchar(255) not null,

ScholarshipAmount float(10),

ScholarshipCategory varchar(255),

ScholarshipStatus varchar(10) not null,

ScholarshipVerficationLevel varchar(255) not null,

primary key (StudentRollNo, ScholarshipName),

Foreign key (StudentRollNo) references StudentBasicInformation(StudentRollNo)

);

describe SubjectScholarshipInformation;

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

Insert into StudentBasicInformation

(StudentName, StudentSurName,StudentRollNo, StudentAddress, StudentContactNo, StudentEmail, StudentClass)

values

('Nikhil','C',45678,'#748/8 18th main SriNagar Bangalore','8088631797','nik@gmail.com','XII-C'),

('Bharath','Kumar',12589,'#18 11th main GiriNagar Bangalore','7896654197','bhar@gmail.com','XII-A'),

('Arun','Patel',74123,'#12 7th main HanumanthNagar Delhi','7412331797','arun@gmail.com','X-B'),

('Mahesh','Kumar',78963,'#11/2 7th main HanumanthNagar Delhi','7414531797','mahesh@gmail.com','VI-C'),

('AChjit','Patel',75321,'#85/2 8th main HanumanthNagar Nagpur','5214531797','abhi@gmail.com','x-B'),

('Suresh','Kumar',75369,'#112/6 8th main RTNagar Nagpur','9614531797','suresh@gmail.com','XII-B'),

('Ram','Patel',95147,'#78 9th main RTNagar Nagpur','9687453177','ram@gmail.com','VII-B'),

('Sham','Kumar',95159,'#11 9th main RTNagar Indore','9784531797','sham@gmail.com','IX-A'),

('Naveen','Yadav',78988,'#741/1 9th main JPNagar Indore','9784531797','naveen@gmail.com','IX-B'),

('Abhishek','Kumar',78787,'#120/8 11th main JPNagar Indore','8532131797','abhishek@gmail.com','XII-A'),

('Suma','Yadav',12121,'#5 11th main JPNagar chennai','8532131797','suma@gmail.com','V-A'),

('Roopa','Yadav',12547,'#101 11th main JPNagar chennai','8534531797','roopa@gmail.com','XI-C'),

('pavithra','sharma',47854,'#789 7th main RTNagar Delhi','9545431797','pavi@gmail.com','X-C');

insert into StudentAdmissionPaymentDetails

(StudentPaymentId, StudentRollNo, AmountPaid, AmmountBalanace, PreviousPaymentId, PaymentType, AdmissionType)

values

('pay56168' , 12121 , 16000 , 21000 ,'pay61957','cash','CET' ),

('pay30168', 12547 , 9000 , 14000 ,'pay16947' ,'cash','CET'),

('pay29653' , 12589 , 22000 , 21000 ,'pay94669' ,'Cheque','management'),

('pay80209' , 45678 , 30000 , 3000 ,'pay96823' ,'Online','management'),

('pay78220' , 47854 , 11000 , 17000 ,'pay38864' ,'Online','Comed-K'),

('pay80187' , 74123 , 17000 , 7000 ,'pay54707' ,'Cheque','management'),

('pay11067' , 75321 , 30000 , 18000 ,'pay98983' ,'Cheque','management'),

('pay61087' , 75369 , 10000 , 27000 ,'pay58627' ,'Online','JEE'),

('pay64451' , 78787 , 17000 , 21000 ,'pay46628' ,'Online','JEE'),

('pay35720' , 78963 , 22000 , 2000 ,'pay78477' ,'cash','JEE'),

('pay11771' , 78988 , 20000 , 29000 ,'pay30843' ,'cash','JEE'),

('pay85659' , 95147 , 21000 , 4000 ,'pay14804' ,'cash','CET'),

('pay71612' , 95159 , 20000 , 22000 ,'pay12208' ,'cash','CET');

insert into StudentSubjectInformation

(SubjectOpted, StudentRollNo, StudentTotalMarks, StudentObtainedMarks, StudentMarksPercentage,SubjectCategory)

values

( 'JavaScript' ,12121, 90 , 81 ,NULL, 'develop' ),

( 'React' ,12547, 100 , 89 ,NULL, 'develop' ),

( 'React',12589, 90 , 81 ,NULL, 'logic' ),

( 'RDBMS' ,45678, 50 , 37 ,NULL, 'develop' ),

( 'JavaScript' ,47854, 100 , 99 ,NULL, 'coding' ),

( 'Git' ,74123, 50 , 48 ,NULL, 'design' ),

( 'Java' ,75321, 80 , 78 ,NULL, 'develop' ),

( 'RDBMS' ,75369, 75 , 70 ,NULL, 'develop' ),

( 'Angular' ,78787, 125 , 95 ,NULL, 'develop' ),

( 'Git' ,78963, 50 , 22 ,NULL, 'develop' ),

( 'Java' ,78988, 90 , 81 ,NULL, 'logic' ),

( 'RDBMS' ,95147, 100 , 97 ,NULL, 'design' ),

( 'Git' ,95159, 90 , 86 ,NULL, 'design' ),

( 'Angular' ,45678, 100 , 100 ,NULL, 'develop'),

( 'Angular' ,47854, 90 , 86 ,NULL, 'design' ),

('Angular',12121, 80 , 79 ,NULL, 'logic'),

('Git',12547, 80 , 74 ,NULL, 'develop'),

('NoSQL',12589, 75 , 73 ,NULL, 'develop'),

('RDBMS',78787, 75 , 69 ,NULL, 'design'),

('Git',12121, 75 , 65 ,NULL, 'coding'),

('NoSQL',95159, 50 , 26 ,NULL, 'coding'),

('NoSQL',47854, 90 , 83 ,NULL, 'coding'),

('Java',74123, 100 , 83 ,NULL, 'coding'),

('JavaScript',95147, 90 , 83 ,NULL, 'develop'),

('Angular',12547, 80 , 71 ,NULL, 'logic');

insert into SubjectScholarshipInformation

(StudentRollNo, ScholarshipName, ScholarshipDescription, ScholarshipAmount, ScholarshipCategory, ScholarshipStatus , ScholarshipVerficationLevel)

values

(12121,'SwarnaJayanti Fellowship','special assistance and support to a selected number of young scientists',2500,NULL,'Pending','District'),

(12547,'Karnata EPASS','Scholarship for Backward Class students',6000,NULL,'Rejected','District'),

(12589,'Karnata EPASS','Scholarship for Backward Class students',4000,NULL,'Processing','College'),

(45678,'Karnata EPASS','Scholarship for Backward Class students',7000,NULL,'Granted','District'),

(47854,'National Fellowship','Scholarship for Higher Education of ST Students',7500,NULL,'Granted','Taluk'),

(74123,'AICTE-Saksham','Ministry of Human Resource Development – AICTE',5000,NULL,'Rejected','Ministry'),

(75321,'Karnata EPASS','Scholarship for Backward Class students',4000,NULL,'Granted','College'),

(75369,'Karnata EPASS','Scholarship for Backward Class students',4000,NULL,'Processing','Taluk'),

(78787,'AICTE-Saksham','Ministry of Human Resource Development – AICTE',5000,NULL,'Processing','Ministry'),

(78963,'ISHAN UDAY','Special Scholarship Scheme for North Eastern Region',9000,NULL,'Granted','Taluk'),

(78988,'Karnata EPASS','Scholarship for Backward Class students',4000,NULL,'Granted','Taluk');

select \* from StudentBasicInformation;

select \* from StudentAdmissionPaymentDetails;

select \* from StudentSubjectInformation;

select \* from SubjectScholarshipInformation;

1. **Snap of the all the tables once the insertion is completed : Added SQL\_Query\_output folder**
2. **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.**
3. **Snap of the all the tables post updation**

update StudentBasicInformation set StudentContactNo='8088789821' where StudentRollNo=12589;

update StudentBasicInformation set StudentEmail='Abhishek.Kumar@gmail.com' where StudentRollNo=78787;

update StudentAdmissionPaymentDetails set PaymentType='UPI' where StudentPaymentId='pay64451';

update studentSubjectInformation set SubjectCategory='testing' where StudentRollNo=12547 and SubjectOpted='Angular';

update SubjectScholarshipInformation set ScholarshipVerficationLevel = 'District' where StudentRollNo=78988;

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

SELECT BasicInfo.StudentRollNo, BasicInfo.StudentName, BasicInfo.StudentEmail, BasicInfo.StudentClass,

SchoInfo.StudentRollNo, SchoInfo.ScholarshipName,SchoInfo.ScholarshipAmount

from StudentBasicInformation as BasicInfo

join SubjectScholarshipInformation as SchoInfo on

BasicInfo.StudentRollNo = SchoInfo.StudentRollNo

where ScholarshipAmount > 5000 and ScholarshipStatus='Granted';

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

SELECT StudentRollNo, StudentName, StudentEmail, StudentClass

from StudentBasicInformation

where StudentRollNo not in (

Select StudentRollNo

from SubjectScholarshipInformation

where

ScholarshipStatus = 'Granted');

1. **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 UpdatePercentage()

BEGIN

UPDATE StudentSubjectInformation

set

StudentSubjectInformation.StudentMarksPercentage =

( StudentSubjectInformation.StudentObtainedMarks / StudentSubjectInformation.StudentTotalMarks ) \* 100;

END //

CALL UpdatePercentage(); //

select \* from StudentSubjectInformation;

1. **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 UpdateCategory()

BEGIN

UPDATE SubjectScholarshipInformation as SchoInfo

JOIN StudentSubjectInformation as SubInfo

on

SubInfo.StudentRollNo = SchoInfo.StudentRollNo

set SchoInfo.ScholarshipCategory =

CASE WHEN SubInfo.StudentMarksPercentage > 90 THEN "Special facility"

WHEN SubInfo.StudentMarksPercentage <= 90 AND SubInfo.StudentMarksPercentage >80 THEN "Fee+hostel fee Concession"

WHEN SubInfo.StudentMarksPercentage <= 80 AND SubInfo.StudentMarksPercentage >70 THEN "Fee concession"

ELSE "Hostel concession"

END;

END //

CALL UpdateCategory(); //

select \* from SubjectScholarshipInformation; //

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

Create view BalanceAmountToBePaid as

SELECT BasicInfo.StudentRollNo, BasicInfo.StudentName, BasicInfo.StudentEmail, BasicInfo.StudentClass, Admission.AmmountBalanace

from

StudentBasicInformation as BasicInfo

join StudentAdmissionPaymentDetails as Admission

on BasicInfo.StudentRollNo = Admission.StudentRollNo; //

select \* from BalanceAmountToBePaid; //

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

Select StudentRollNo, StudentName, StudentEmail, StudentClass

from StudentBasicInformation

where StudentBasicInformation.StudentRollNo not in (

Select StudentRollNo

from SubjectScholarshipInformation);

1. **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 AmountToBePaid\_ByParameter(IN StudentRollNumber INT)

BEGIN

SELECT Admisssion.AmmountBalanace

from StudentAdmissionPaymentDetails as Admisssion

WHERE Admisssion.StudentRollNo = StudentRollNumber;

END $$

CALL AmountToBePaid\_ByParameter(95147); $$

1. **Retrieve the top five student details as per the StudentMarks Percentage values (use subqueries)**

Select Top5Students.StudentRollNo, Top5Students.StudentName, Top5Students.StudentEmail, Top5Students.StudentClass, Top5Students.percentage

from (

Select BasicInfo.StudentRollNo, BasicInfo.StudentName, BasicInfo.StudentEmail,

BasicInfo.StudentClass, SubInfo.StudentMarksPercentage as percentage

from StudentBasicInformation as BasicInfo

join StudentSubjectInformation as SubInfo

on

BasicInfo.StudentRollNo = SubInfo.StudentRollNo

order by SubInfo.StudentMarksPercentage desc)

as Top5Students

limit 0,5; $$

1. **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)**

/\*Compare percentage of each student with classmate by each subject\*/

SELECT table\_1.StudentRollNo,table\_1.SubjectOpted,table\_1.StudentMarksPercentage,

table\_2.StudentRollNo,table\_2.SubjectOpted,table\_2.StudentMarksPercentage

FROM StudentSubjectInformation as table\_1

CROSS JOIN StudentSubjectInformation as table\_2

WHERE NOT table\_1.StudentRollNo = table\_2.StudentRollNo

ORDER BY table\_1.StudentRollNo ASC; $$

/\*finding scholarship status of student whose scholarship is not Granted\*/

SELECT BasicInfo.StudentRollNo, BasicInfo.StudentName, SchoInfo.ScholarshipName,SchoInfo.ScholarshipAmount, SchoInfo.ScholarshipStatus

from StudentBasicInformation as BasicInfo

join SubjectScholarshipInformation as SchoInfo on

BasicInfo.StudentRollNo = SchoInfo.StudentRollNo

where ScholarshipStatus <>'Granted'; $$

/\* find students who paid fees but did not apply for scholarship\*/

SELECT Admission.StudentRollNo, Admission.StudentpaymentId

FROM StudentAdmissionPaymentDetails as Admission

LEFT OUTER JOIN SubjectScholarshipInformation as SchoInfo

ON Admission.StudentRollNo = SchoInfo.StudentRollNo

WHERE SchoInfo.StudentRollNo is NULL;

1. **Answered in the pdf along with Question 20**
2. **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 FrequencyTable.Category\_name, max(FrequencyTable.Count) as Category\_count from (

Select SchoInfo.ScholarshipCategory as Category\_Name, Count(SchoInfo.ScholarshipCategory) as Count

from SubjectScholarshipInformation as SchoInfo

group by SchoInfo.ScholarshipCategory) as FrequencyTable; $$

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

Select SchoInfo.ScholarshipCategory as Category\_Name, Count(SchoInfo.ScholarshipCategory) as Count

from SubjectScholarshipInformation as SchoInfo

group by SchoInfo.ScholarshipCategory; $$

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

update SubjectScholarshipInformation set ScholarshipAmount = 2000 where StudentRollNo=78963; $$

update SubjectScholarshipInformation set ScholarshipAmount = 7500 where StudentRollNo=45678; $$

Select BasicInfo.StudentRollNo, BasicInfo.StudentName, SubInfo.StudentMarksPercentage

from StudentBasicInformation as BasicInfo

JOIN StudentSubjectInformation as SubInfo

on

BasicInfo.StudentRollNo = SubInfo.StudentRollNo

where BasicInfo.StudentRollNo in

(Select ssss.StudentRollNo

from StudentSubjectInformation as ssss

where ssss.StudentRollNo in

( Select SchoInfo.StudentRollNo

from SubjectScholarshipInformation as SchoInfo

where SchoInfo.ScholarshipAmount in

( Select max(SubjectScholarshipInformation.ScholarshipAmount) from SubjectScholarshipInformation )

and

SchoInfo.ScholarshipStatus = 'Granted'

)

) order by SubInfo.StudentMarksPercentage desc; $$

1. **Answered in the pdf along with Question 16**