--select \* from student

--select \* from department

--select \* from subjects

select \* from [dbo].[StudentSubjects]

select \* from [dbo].[StudentDepartment]

--დავამატოთ შეზღუდვა StudentSubjects Reg\_Date გავხადოთ დეფოლთად getdate()

alter table [dbo].[StudentSubjects]

add

constraint df\_regdate default getdate() for [reg\_date]

===პროცედურები

--შევქმნათ სტუდენტის ცხრილში ჩანაწერის დამატები პროცედურა--

if object\_id('InsertStudent','P')

is not null

Drop procedure InsertStudent

Go

create procedure InsertStudent

@StudentName nvarchar(30),

@StudentLastName nvarchar(30),

@DateofBirth date,

@personalId char(11),

@Gender nvarchar(20),

@city nvarchar(50),

@email varchar(30)

As

Begin

insert into student values

(

@StudentName,@StudentLastName,@DateofBirth,@personalId,@Gender,@city,@email

)

End

---შევხედოთ პროცედურის შექმნის კოდს

exec sp\_helptext 'InsertStudent'

--ჩავსვათ სტუდენტის ცხრილში ჩანაწერი პროცედურით

exec InsertStudent N'ელენე',N'ბექაური', '2/2/2020','11223344557',N'მდედრობითი',

N'ბათუმი','xxxx@gmail.com'

-- შევქმნათ StudentSubjects ცხრილში მონაცემის შეტანის პროცედურა

create procedure RegStudentSubjects

@StudentID int,

@SubjectId int

As

Begin

insert into StudentSubjects(StudentId, SubjectId)

values

(@StudentId, @SubjectId)

End

Go

--გავაკეთოთ ცხრილში ჩანაწერი პროცედურით

exec RegStudentSubjects 10, 400

-- შვქმნათ პროცედურა, რომელსაც გადავცემთ StudentId-ს და დაგვიბრუნებს სრულ

--ინფორმაციას Student ცხრილიდან

Create procedure StudentInfo

@StudentId int

As

Begin

Select \* from Student

Where StudentID=@studentId

End

exec StudentInfo 1

--დავწეროთ მოთხოვნა, რომელიც დააბრუნებს საგნებს და მასზე დარეგისტრირებულ სტუდენტების რაოდენობას.

--მხოლოდ ის საგნები რომლებზედაც სტუდენტები არიან დარეგისტრირებულები.

select stsb.SubjectId, SubjectName,count(StudentId) as Sttotal

from StudentSubjects stsb left join subjects sb

on stsb. SubjectId=sb.Subjectid

Group by stsb.SubjectId, SubjectName

--დავწეროთ მოთხოვნა, რომელიც დააბრუნებს სრულად საგნებს და მასზე დარეგისტრირებულ სტუდენტების რაოდენობას.

--ის საგნებიც, რომლებზედაც სტუდენტები არ არიან დარეგისტრირებულები.

select sb.SubjectId, SubjectName,count(StudentId) as Sttotal

from StudentSubjects stsb right join subjects sb

on stsb. SubjectId=sb.Subjectid

where sb.SubjectId=400

Group by sb.SubjectId, SubjectName

--შევქმნათ პროცედურა, რომელსაც გადავცემთ საგნის კოდს და

--გამოიტანს მასზე დარეგისტრირებული სტუდენტების რაოდენობას.

create procedure SubjectStudentTotal

@SubjectId int

As

Begin

select sb.SubjectId,SubjectName,count(StudentId) as Sttotal

from StudentSubjects stsb right join subjects sb

on stsb. SubjectId=sb.Subjectid

where sb.SubjectId=@subjectID

Group by sb.SubjectId, SubjectName

End

exec SubjectStudentTotal 800

--შევქმნათ პროცედურა, რომელიც მხოლოდ იმ საგნებს დააბრუნებს,

--რომლებზედაც დარეგისტრირებული სტუდენტების რაოდენობა >2-ზე (საგანი დახურულია)

Create procedure ClosedSubjects

as

Begin

select stsb.SubjectId, SubjectName, count(StudentId) as StTotal

From StudentSubjects stsb left join Subjects sb

on Stsb.SubjectId=Sb.Subjectid

Group by stsb.SubjectId, SubjectName

having count(StudentId)>2

End

exec ClosedSubjects

--განვსაზღვროთ საგნისათვის აქტივაციის სტატუსი:

select sb.SubjectId, SubjectName,count(StudentId),

case

when count(StudentId)>2 then N'დახურულია'

Else N'აქტიურია'

End N'აქტივცია'

From StudentSubjects stsb right join Subjects sb

on Stsb.SubjectId=Sb.Subjectid

Group by sb.SubjectId, SubjectName

--შევქმნათ პროცედურა, რომელიც სტუდენტის კოდის გადაცემის შემთხვევაში

--დაითვლის მის მიერ არჩეული საგნების კრეიტების ჯამს.

create procedure SumCredits

@StudentId int

As

Begin

select St.StudentId, sum(Subjectcredit) as total

from Student st left join StudentSubjects stsb on st.studentid=stsb.studentid

left join subjects sb on stsb.subjectid=sb.subjectid

where St.StudentId=@StudentId

group by St.StudentId

End

exec SumCredits 5

--შევქმნათ პროცედურა, რომელიც სტუდენტის კოდის, ძველი და ახალი საგნის კოდეის

--გადაცემის შემთხვეაში, მოახდენს სტუდენტისატვის საგნის განახლებას.--

Create procedure updateStudentSubject

@StudentId int,

@oldSubjectid int,

@NewSubjectID int

as

Begin

update studentSubjects

Set SubjectId=@newsubjectId

where studentId=@studentId

and Subjectid=@oldSubjectId

End

exec updateStudentSubject 7,200,100

exec updateStudentSubject 7,500

select \* from StudentSubjects

update studentSubjects

Set SubjectId=100 where studentId=7