Examination System

1- Database name => Examination_System2

2- The created files:

ExamSys_mainA.mdf

ExamSys2A.ndf

ExamSys3A.ndf

ExamSys4A.ndf

ExamSys_logA.ldf

3- The created group files:

SecondryExamSys

the schema

exs

the tables

exs.Students

exs.Instructor

exs.Course

exs.Exam

exs.QuestionBank

exs.Inst_Courses

exs.Course_Students

exs.Exam_Students

dbo.studentAnswers

Programmability in the system

Stored procedures:

[exs].[addQuestionToTheBank2]

Take parameters

```
(@questionText nvarchar(max),@QuestionType nvarchar(20),@degree
int,@CorrectAnswer nvarchar(max),@CourId int,@InstId int)
```

To insert new question with it’s details to the table

```
[exs].[QuestionBank]
```

And ckeck if there is exception happened

[exs].[UpdateQuestion]

Take parameters

```
(@quesId int,@questionText nvarchar(max),@degree
int,@CorrectAnswer nvarchar(max),@CourId int,@InstId int)
```

To update a spesific question with it’s id and it’s instructor by it’s instructor only.

To upadate the table `[exs].[QuestionBank]`

[exs].[deleteQuestion]

Take parameters

```
(@quesId int,@CourId int,@InstId int)
```

To delete a spesific question with it’s id ,it’s istructor and it’s course by it’s instructor only.

To delete the question from the table
`[exs].[QuestionBank]`

[exs].[addExam]

Take parameters

```
(@examType nvarchar(20),@totalDegree int,@sTime time,@eTime
time,@examDate date,@year int,@CourId int,@InstId int)
```

To insert a new exam to the table [exs].[Exam]

With it’s details

[exs].[putQuestionsInExam]

Take parameters

```
(@InstId int,@examId int,@examDegree int)
```

To make the system choose questions and put them in the the specific exam

And add every question degree to a avariable

And check to make the sum of question degrees not to exceed the Exam degree .

The system will choose from the questions of the instructor only.

[exs].[putQuestionsInExamManually]

Take parameters

```
(@quesId int,@InstId int,@examId int)
```

The instructor will choose a question by it’s id

And put it in a specific Exam

The instructor can’t add a question to the exam if he didn’t add the question to the question pool before . he can add it’s questions only.

User defined functions:

[dbo].[studentInExam]

Take parameters

```
(@studentId int, @examId int)
```

And return a new table

```
returns @StuDegrees table
         (
          stuId int  default 1 ,
          examId int ,
          quesId int primary key identity(1,1),
          stuAnswer nvarchar(20),
          correctAnswer nvarchar(20),
          quesDegree int,
          stuQuesDegree int

         )
```

This function take student id and exam id

and display all details when student take exam

it will display take exam id and question degree and

the correct answer from table [exs].[QuestionBank] by using Exam id.

The function will go on loop to update the values of

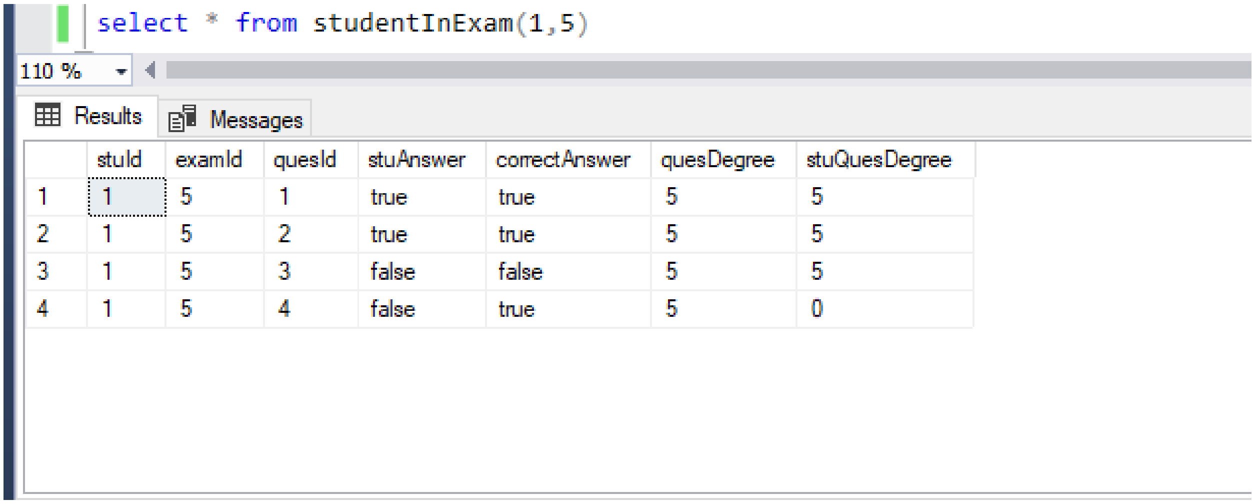
[stuId] the function will take it from the parameters and
[stuAnswer] the function will loop in another function

[dbo].[studentAnswersFun] to get every question answer from the table [dbo].[studentAnswers]

The function will compare [stuAnswer] with [correctAnswer]

And will put the degree of every question in [stuQuesDegree]

Finally the function will return a new table with all the information



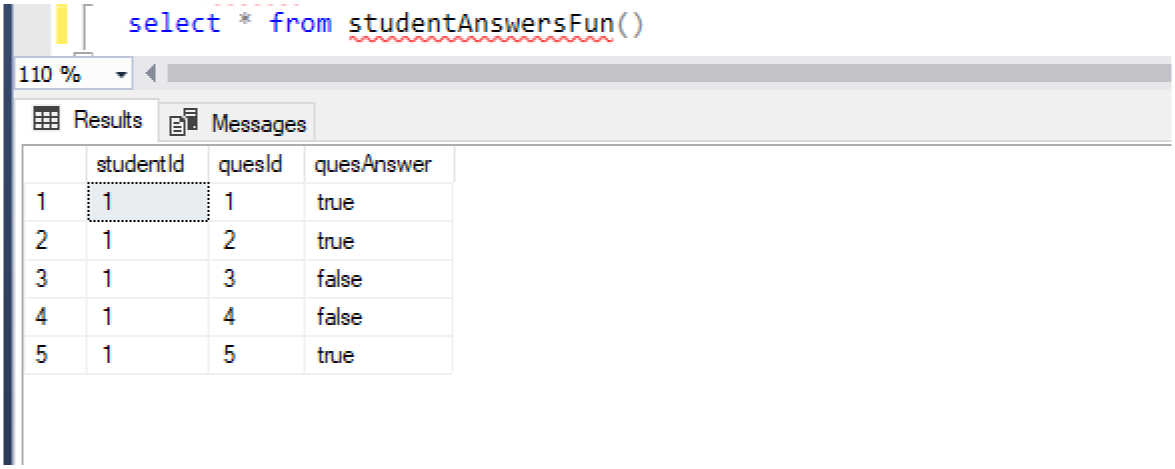
```
select * from studentInExam(1,5)
```

	stuId	examId	quesId	stuAnswer	correctAnswer	quesDegree	stuQuesDegree
1	1	5	1	true	true	5	5
2	1	5	2	true	true	5	5
3	1	5	3	false	false	5	5
4	1	5	4	false	true	5	0

[dbo].[studentAnswersFun]

```
 studentAnswersFun()
 returns table
 as
return  (select * from [dbo].[studentAnswers])
```

this fun to return the table of `[dbo].[studentAnswers]`



```
select * from studentAnswersFun()
```

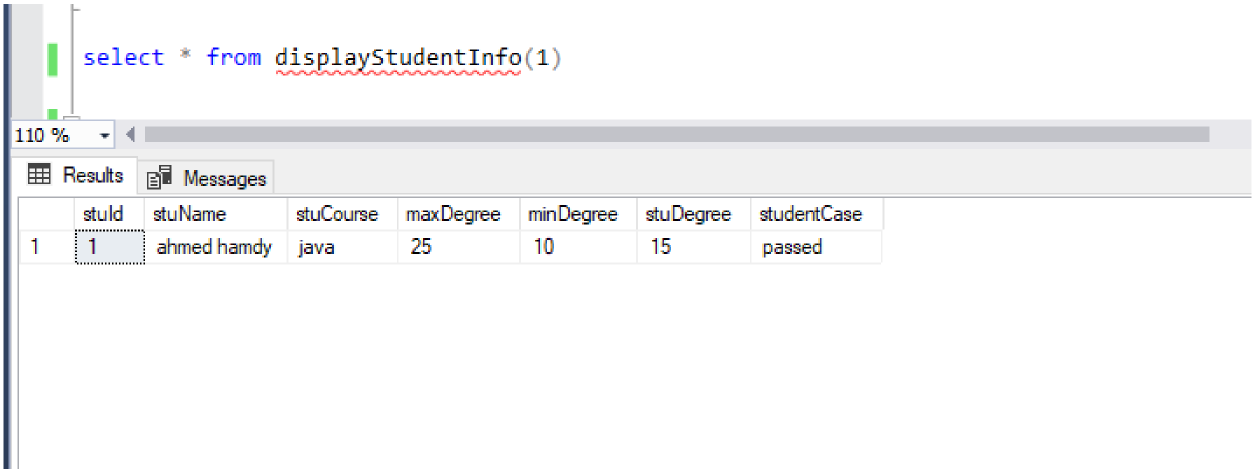
	studentId	quesId	quesAnswer
1	1	1	true
2	1	2	true
3	1	3	false
4	1	4	false
5	1	5	true

[dbo].[displayStudentInfo]

```
displayStudentInfo(@studentId int)
returns @studentInfo table
    (
     stuId int  default 1 ,
     stuName nvarchar(max),
     stuCourse nvarchar(20),
     maxDegree int,
     minDegree int,
     stuDegree int,
     studentCase nvarchar(20)
    )
```

This function to take the student id and display all information about the student that related to it’s exam and it’s case (passed or failed).

The function check if the student degree bigger than the min degree of the exam the system will put in [studentCase] passed else will put fail



```
select * from displayStudentInfo(1)
```

```
   stuId  stuName      stuCourse  maxDegree  minDegree  stuDegree  studentCase
1  1      ahmed hamdy  java       25         10         15         passed
```

Views:

```
create view allCourses
as
( select * from [exs].[Course]
  )
```

To display all courses from table [exs].[Course]