Azhar

Examination System report

Data Dictionary

1/5/2021

Table of contents

Legend

- Primary key
- Primary key disabled
- User-defined primary key
- Unique key
- Unique key disabled
- User-defined unique key
- Active trigger
- Disabled trigger
- Many to one relation
- User-defined many to one relation
- One to many relation
- User-defined one to many relation
- Many to many relation
- User-defined many to many relation
- One to one relation
- User-defined one to one relation
- Input
- Output
- Input/Output
- Uses dependency
- User-defined uses dependency
- Used by dependency
- User-defined used by dependency

Examination System

This an Examination system which generates an exam randomly for each student and correct the student answer according to amodel answer then calculate the final grade fot the student .

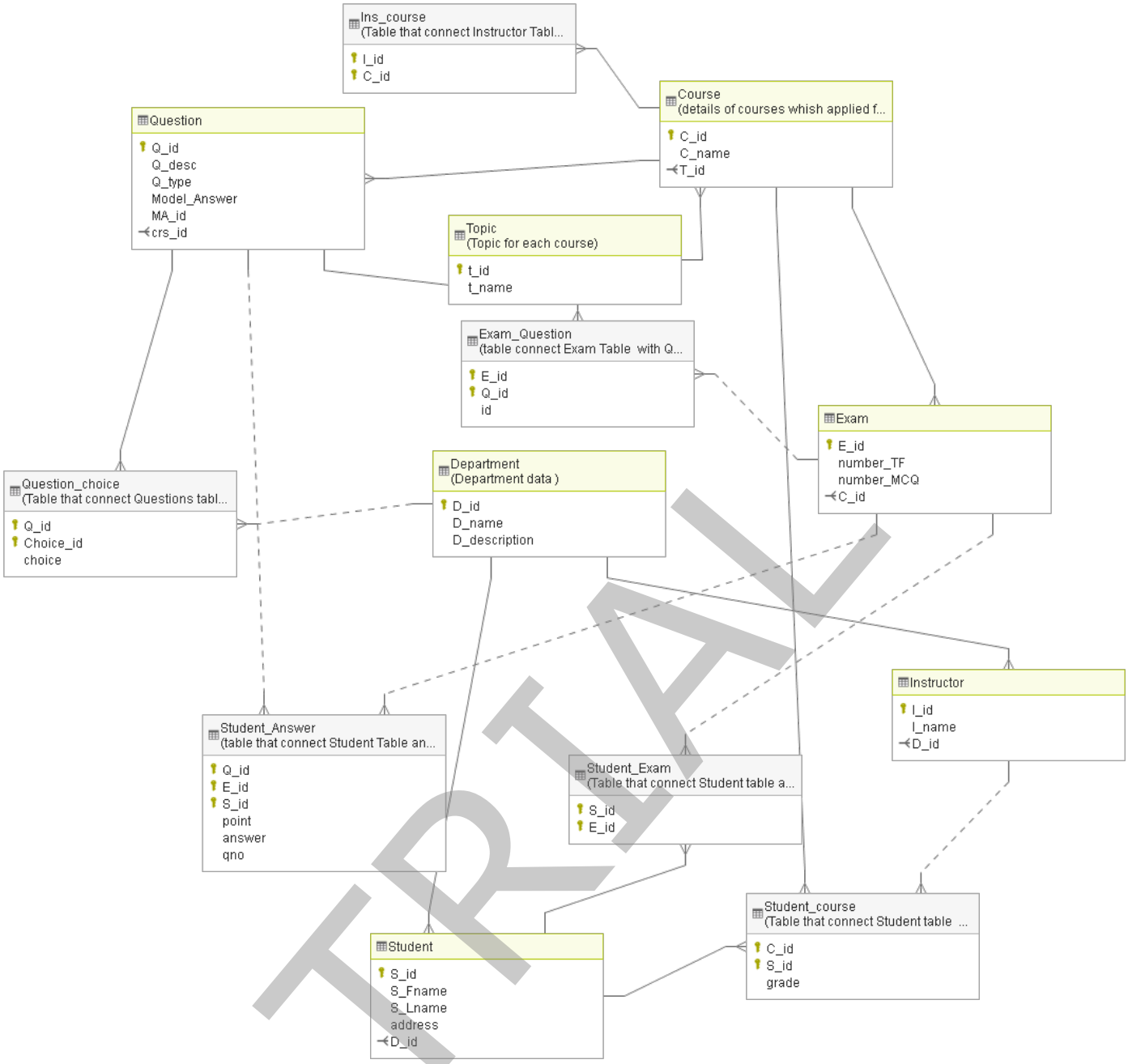
this system contain several featuers...

1) like generating two types of questions (true&false - MCQ)
2) each exam consist of 10 questions that would be distriputed randomly as aTrue&false or MCQ Question
3) system calculate student point according to it's answer either it is true or false then add the result to the final grade
4) we applied different reports according to student ,course,instructore,exam and topic that would help mangers to improve
education process and now more insights about his academy
5) generate dashboard that show an analysis of student's social media accounts

now let's go in details and show our tools and platforms ....

1) we used sql server 2016 to create database and build stored prosedures and rund different quieries
2) visual studio to implement our database into adesktop application using wondows forms with as a smart User Interface
3) Power BI Desktop & Power BI report Service to publish pur reports and build a smart dashboard that show snalysis of student's
social media accounts
4) SSRS to build other reports

1. System_Schema



gfxgfCVBJNL;,YFVBKM,

2. Other

2.1. Tables

2.1.1. Table: Course (details of courses whish applied for students)

Status: Active

Columns

		Name	Data type	Description / Attributes
		C_id	int	unique id for each course
		C_name	nvarchar(50)	course name
		T_id	int	unique Topic id for each course **References**: Topic

Links to

	Table	Join	Title / Name / Description
	Topic (Topic for each course)	**Course**T_id = Topict_id	FK_Course_Topic

Linked from

	Table	Join	Title / Name / Description
	Exam	**Course**C_id = ExamC_id	FK_Exam_Course
	Ins_course (Table that connect Instructor Table with Course Table)	**Course**C_id = Ins_courseC_id	FK_Ins_course_Course
	Question	**Course**C_id = Questioncrs_id	FK_Question_Course
	Student_course (Table that connect Student table and course table)	**Course**C_id = Student_courseC_id	FK_Student_course_Course

Unique keys

		Name / Description
	C_id	PK_Course

Uses

Name
Course (details of courses whish applied for students)
Topic (Topic for each course)

Used By

Name
Course (details of courses whish applied for students)
Exam
Ins_course (Table that connect Instructor Table with Course Table)
Question
Student_course (Table that connect Student table and course table)

2.1.2. Table: Department (Department data)

each department contain several courses,students,instructores

Columns

		Name	Data type	Description / Attributes
		D_id	int	Department ID
		D_name	nvarchar(50)	Department Name
		D_description	nvarchar(MAX)	Department Details

Linked from

	Table	Join	Title / Name / Description
	Instructor	**Department**D_id = InstructorD_id	FK_Instructor_Department
	Question_choice (Table that connect Questions table and Choices for each Question)	**Department**D_id = Question_choiceQ_id	fk_Department_Question_choice
	Student	**Department**D_id = StudentD_id	FK_Student_Department

Unique keys

		Name / Description
	D_id	PK_Department

Used By

Name
Department (Department data)
Instructor
Question_choice (Table that connect Questions table and Choices for each Question)
Student

2.1.3. Table: Exam

Columns

		Name	Data type	Description / Attributes
		E_id	int	Exam ID **Identity / Auto increment**
		number_TF	int	Number of True and False Questions
		number_MCQ	int	Number of MCQ questions
		C_id	int	Course ID **References**: Course

Links to

	Table	Join	Title / Name / Description
	Course (details of courses whish applied for students)	**Exam**C_id = CourseC_id	FK_Exam_Course

Linked from

	Table	Join	Title / Name / Description
	Exam_Question (table connect Exam Table with Question Table)	**Exam**E_id = Exam_QuestionE_id	fk_Exam_Exam_Question
	Student_Answer (table that connect Student Table and Answer Table)	**Exam**E_id = Student_AnswerE_id	fk_Exam_Student_Answer
	Student_Exam (Table that connect Student table and Exam table)	**Exam**E_id = Student_ExamE_id	fk_Exam_Student_Exam

Unique keys

		Name / Description
	E_id	PK_Exam

Uses

	Name
Exam	
	Course (details of courses whish applied for students)

Used By

	Name
Exam	
	correction (comparing student answer with the correct answer)
	generate (generate exam for each student)
	st_ans (save student answers for each question)
	Exam_Question (table connect Exam Table with Question Table)
	Student_Answer (table that connect Student Table and Answer Table)
	Student_Exam (Table that connect Student table and Exam table)

2.1.4. Table: Exam_Question (table connect Exam Table with Question Table)

*_two tables are connected with the exam id and question id .
*_as it is possible to repeate a question among many exams ,we set aunique composit Key .

Columns

		Name	Data type	Description / Attributes
		E_id	int	Exam ID **References**: Exam
		Q_id	int	Question ID **References**: Question
		id	int	**Identity / Auto increment**

Links to

	Table	Join	Title / Name / Description
	Exam	**Exam_Question**E_id = ExamE_id	fk_Exam_Exam_Question
	Question	**Exam_Question**Q_id = QuestionQ_id	FK_Exam_Question_Question

Unique keys

		Name / Description
	E_id, Q_id	PK_Exam_Question

Uses

	Name
	Exam_Question (table connect Exam Table with Question Table)
	Exam
	Question

Used By

	Name
	Exam_Question (table connect Exam Table with Question Table)
	answers (Stored Prosedure That take 12 parameters as inputs)
	generate (generate exam for each student)
	st_ans (save student answers for each question)

2.1.5. Table: Ins_course (Table that connect Instructor Table with Course Table)

each instructor can teach one or more course and course may have one or more instructore , so we connectedthe two tables with a composit key in separated table to be unique and select them easly

Columns

		Name	Data type	Description / Attributes
		I_id	int	Instructore ID
		C_id	int	Course ID **References**: Course

Links to

	Table	Join	Title / Name / Description
	Course (details of courses whish applied for students)	**Ins_course**C_id = CourseC_id	FK_Ins_course_Course

Unique keys

		Name / Description
	I_id, C_id	PK_Ins_course

Uses

	Name
	Ins_course (Table that connect Instructor Table with Course Table)
	Course (details of courses whish applied for students)

2.1.6. Table: Instructor

Columns

		Name	Data type	Description / Attributes
		I_id	int	Instructor ID
		I_name	nvarchar(50)	Instructore Name
		D_id	int	Department ID **References**: Department

Links to

	Table	Join	Title / Name / Description
	Department (Department data)	**Instructor**D_id = DepartmentD_id	FK_Instructor_Department

Linked from

	Table	Join	Title / Name / Description
	Student_course (Table that connect Student table and course table)	**Instructor**I_id = Student_courseS_id	fk_Instructor_Student_course

Unique keys

		Name / Description
	I_id	PK_Instructor

Uses

	Name
	Instructor
	Department (Department data)

Used By

	Name
	Instructor
	Student_course (Table that connect Student table and course table)

2.1.7. Table: Question

Columns

		Name	Data type	Description / Attributes
		Q_id	int	Question ID **Identity / Auto increment**
		Q_desc	nvarchar(MAX)	Question Text
		Q_type	nvarchar(50)	Question Type (TF || MCQ)
		Model_Answer	nvarchar(50)	Correct Answer Text
		MA_id	int	Correct Answer ID
		crs_id	int	Course ID **Nullable** **References**: Course

Links to

	Table	Join	Title / Name / Description
	Course (details of courses whish applied for students)	**Question**crs_id = CourseC_id	FK_Question_Course

Linked from

	Table	Join	Title / Name / Description
	Exam_Question (table connect Exam Table with Question Table)	**Question**Q_id = Exam_QuestionQ_id	FK_Exam_Question_Question
	Question_choice (Table that connect Questions table and Choices for each Question)	**Question**Q_id = Question_choiceQ_id	FK_Question_choice_Question
	Student_Answer (table that connect Student Table and Answer Table)	**Question**Q_id = Student_AnswerQ_id	fk_Question_Student_Answer

Unique keys

		Name / Description
	Q_id	PK_Question

Uses

	Name
	Question
	Course (details of courses whish applied for students)

Used By

	Name
	Question
	correction (comparing student answer with the correct answer)
	generate (generate exam for each student)
	Exam_Question (table connect Exam Table with Question Table)
	Question_choice (Table that connect Questions table and Choices for each Question)
	Student_Answer (table that connect Student Table and Answer Table)

2.1.8. Table: Question_choice (Table that connect Questions table and Choices for each Question)

each Multiple choice question has 4 choices , so we needed to connet each question with each choice in aseparate row to be unique and compare it with the model answer .

Columns

		Name	Data type	Description / Attributes
		Q_id	int	Question ID **References**: Department, Question
		Choice_id	int	Choice ID
		choice	varchar(MAX)	Choice as a Text **Nullable**

Links to

	Table	Join	Title / Name / Description
	Department (Department data)	**Question_choice**Q_id = DepartmentD_id	fk_Department_Question_choice
	Question	**Question_choice**Q_id = QuestionQ_id	FK_Question_choice_Question

Unique keys

		Name / Description
	Q_id, Choice_id	PK_Question_choice

Uses

	Name
	Question_choice (Table that connect Questions table and Choices for each Question)
	Department (Department data)
	Question

2.1.9. Table: Student

Columns

		Name	Data type	Description / Attributes
		S_id	int	Student ID
		S_Fname	nvarchar(50)	Student First Name
		S_Lname	nvarchar(50)	Student last Name
		address	nvarchar(50)	Student Address
		D_id	int	Department ID **References**: Department

Links to

	Table	Join	Title / Name / Description
	Department (Department data)	**Student**D_id = DepartmentD_id	FK_Student_Department

Linked from

	Table	Join	Title / Name / Description
	Student_course (Table that connect Student table and course table)	**Student**S_id = Student_courseS_id	FK_Student_course_Student
	Student_Exam (Table that connect Student table and Exam table)	**Student**S_id = Student_ExamS_id	FK_Student_Exam_Student

Unique keys

		Name / Description
	S_id	PK_Student

Uses

	Name
	Student
	Department (Department data)

Used By

	Name
	Student
	st_ans (save student answers for each question)
	Student_course (Table that connect Student table and course table)
	Student_Exam (Table that connect Student table and Exam table)

2.1.10. Table: Student_Answer (table that connect Student Table and Answer Table)

we connected three keys as acomposite key to be unique as questions may be repeated with many exams and student may take more than one exam

Columns

		Name	Data type	Description / Attributes
		Q_id	int	Question ID **References**: Question
		E_id	int	Exam ID **References**: Exam
		S_id	int	Student ID
		point	int	Student score for each question **Nullable**
		answer	nvarchar(50)	Student Answer for each Question **Nullable**
		qno	int	

Links to

	Table	Join	Title / Name / Description
	Exam	**Student_Answer**E_id = ExamE_id	fk_Exam_Student_Answer
	Question	**Student_Answer**Q_id = QuestionQ_id	fk_Question_Student_Answer

Unique keys

		Name / Description
	Q_id, E_id, S_id	PK_Student_Answer

Uses

	Name
	Student_Answer (table that connect Student Table and Answer Table)
	Exam
	Question

Used By

	Name
	Student_Answer (table that connect Student Table and Answer Table)
	answers (Stored Prosedure That take 12 parameters as inputs)
	correction (comparing student answer with the correct answer)
	st_ans (save student answers for each question)

2.1.11. Table: Student_course (Table that connect Student table and course table)

Columns

		Name	Data type	Description / Attributes
		C_id	int	Course ID **References**: Course
		S_id	int	Student ID **References**: Instructor, Student
		grade	int	Final Score for each student **Nullable**

Links to

	Table	Join	Title / Name / Description
	Course (details of courses whish applied for students)	**Student_course**C_id = CourseC_id	FK_Student_course_Course
	Instructor	**Student_course**S_id = InstructorI_id	fk_Instructor_Student_course
	Student	**Student_course**S_id = StudentS_id	FK_Student_course_Student

Unique keys

		Name / Description
	C_id, S_id	PK_Student_course

Uses

	Name
	Student_course (Table that connect Student table and course table)
	Course (details of courses whish applied for students)
	Instructor
	Student

Used By

	Name
	Student_course (Table that connect Student table and course table)
	correction (comparing student answer with the correct answer)
	st_ans (save student answers for each question)

2.1.12. Table: Student_Exam (Table that connect Student table and Exam table)

Columns

		Name	Data type	Description / Attributes
		S_id	int	Student ID **References**: Student
		E_id	int	Exam ID **References**: Exam

Links to

	Table	Join	Title / Name / Description
	Exam	**Student_Exam**E_id = ExamE_id	fk_Exam_Student_Exam
	Student	**Student_Exam**S_id = StudentS_id	FK_Student_Exam_Student

Unique keys

		Name / Description
	S_id, E_id	PK_Student_Exam

Uses

	Name
	Student_Exam (Table that connect Student table and Exam table)
	Exam
	Student

Used By

	Name
	Student_Exam (Table that connect Student table and Exam table)
	st_ans (save student answers for each question)

2.1.13. Table: Topic (Topic for each course)

Columns

		Name	Data type	Description / Attributes
		t_id	int	Topic ID
		t_name	nvarchar(50)	Topic Name

Linked from

	Table	Join	Title / Name / Description
	Course (details of courses whish applied for students)	**Topic**t_id = CourseT_id	FK_Course_Topic

Unique keys

		Name / Description
	t_id	PK_Topic

Used By

	Name
	Topic (Topic for each course)
	Course (details of courses whish applied for students)

2.2. Procedures

2.2.1. Procedure: answers (Stored Prosedure That take 12 parameters as inputs)

save student answers for 10 questions in Student_Answer table
in addition to saving question id and student id

Input/Output

	Name	Data type	Description
@	st_id	int	
@	exm	int	
@	a1	int	
@	a2	int	
@	a3	int	
@	a4	int	
@	a5	int	
@	a6	int	
@	a7	int	
@	a8	int	
@	a9	int	
@	a10	int	

Uses

	Name
	answers (Stored Prosedure That take 12 parameters as inputs)
	Exam_Question (table connect Exam Table with Question Table)
	Student_Answer (table that connect Student Table and Answer Table)

Script

```
create proc answers @st_id int,@exm int,@a1 int,@a2 int, @a3 int, @a4 int, @a5 int, @a6 int, @a7 int, @a8 int, @a9 int, @a10
int
as
declare @t table(id int,ans int)
insert into @t
values
(1,@a1),
(2,@a2),
(3,@a3),
(4,@a4),
(5,@a5),
(6,@a6),
(7,@a7),
(8,@a8),
(9,@a9),
(10,@a10)
insert into Student_Answer(Q_id,E_id,S_id,answer,qno)
select eq.Q_id,eq.e_id,@st_id,ans,eq.id
from Exam_Question eq ,@t t
where eq.id=t.id and eq.E_id=@exm
order by t.id
```

2.2.2. Procedure: correction (comparing student answer with the correct answer)

it takes two inputs as student id and exam id , then compare the saved answers with the correct answers in student answer

Input/Output

	Name	Data type	Description
→@	st_id	int	
→@	exam_id	int	

Uses

	Name
⚙	correction (comparing student answer with the correct answer)
	Exam
	Question
	Student_Answer (table that connect Student Table and Answer Table)
	Student_course (Table that connect Student table and course table)

Script

```
CREATE proc correction @st_id int,@exam_id int
as
declare c1 Cursor
for select sa.Q_id,E_id,S_id,answer,point,ma_id
    from Student_Answer sa inner join Question q on q.Q_id=sa.Q_id
    where E_id=@exam_id and s_id=@st_id
for update
declare @q_id int,@e_id int,@s_id int,@answer int,@pnt int,@ma int
open c1
fetch c1 into @q_id,@e_id,@s_id,@answer,@pnt,@ma
while @@FETCH_STATUS=0
           begin
                      if @answer=@ma
                      update Student_Answer set point = 1 where current of c1
                      else
                      update Student_Answer set point = 0 where current of c1
                      fetch c1 into @q_id,@e_id,@s_id,@answer,@pnt,@ma
           end
close c1
deallocate c1

declare @crs_id int
select @crs_id=c_id from Exam where e_id=@exam_id

update Student_course
           set grade= (select ((sum(point)*100)/10 )   from Student_Answer)
           where S_id=@st_id and C_id=@crs_id
```

2.2.3. Procedure: generate (generate exam for each student)

exam is generated according to number of True & false and MCQ questions , then save the questions in Question table and connect it with exam and exam_question table

Input/Output

	Name	Data type	Description
@	crs	int	
@	mcq	int	
@	t_f	int	

Uses

	Name
	generate (generate exam for each student)
	exam
	Exam_Question (table connect Exam Table with Question Table)
	question

Script

```
CREATE proc generate @crs int,@mcq int, @t_f int
as
if @mcq+@t_f=10
begin

insert into exam(number_TF,number_MCQ,C_id) values(@t_f,@mcq,@crs)
declare @x int
set @x= @@IDENTITY
insert into Exam_Question(Q_id,E_id)
select top(@mcq) Q_id,@x  from question
where Q_type='MCQ' and crs_id=@crs
order by newid()

insert into Exam_Question(Q_id,E_id)
select top(@t_f) Q_id,@x from question
where Q_type='TF' and crs_id=@crs
order by newid()
dbcc checkident(Exam_Question,reseed,0)
end
else
select 'wrong input'
```

2.2.4. Procedure: st_ans (save student answers for each question)

it takes the 10 answers,student id , exam id , question id as inputs and save them in in all related tables , like Student ,Exam , Question

Input/Output

	Name	Data type	Description
@	st_id	int	
@	exam_id	int	
@	q1	int	
@	q2	int	
@	q3	int	
@	q4	int	
@	q5	int	
@	q6	int	
@	q7	int	
@	q8	int	
@	q9	int	
@	q10	int	

Uses

Name
st_ans (save student answers for each question)
Exam
exam_question (table connect Exam Table with Question Table)
Student
Student_Answer (table that connect Student Table and Answer Table)
Student_course (Table that connect Student table and course table)
Student_Exam (Table that connect Student table and Exam table)

Script

```
CREATE proc st_ans @st_id int,@exam_id int,@q1 int,@q2 int,@q3 int,@q4 int
                                                              ,@q5 int,@q6 int,@q7 int,@q8 int,@q9 int,@q10 int
as
declare @ans table (answers int)
insert into @ans values(@q1),(@q2),(@q3),(@q4),(@q5),(@q6),(@q7),(@q8),(@q9),(@q10)

insert into Student_Exam values (@st_id,@exam_id)
declare @crs_id int
select @crs_id=c_id from Exam where e_id=@exam_id

insert into Student_course(C_id,S_id) values (@crs_id,@st_id)

insert into Student_Answer(Q_id,E_id,S_id)
select Q_id,E_id,S_id
from exam_question,Student
where E_id=@exam_id and s_id=@st_id

declare c1 Cursor
for select Q_id,E_id,S_id,answer
    from Student_Answer
    where E_id=@exam_id and s_id=@st_id
for update
declare @q_id int,@e_id int,@s_id int,@answer int
open c1
fetch c1 into @q_id,@e_id,@s_id,@answer
while @@FETCH_STATUS=0
          begin
                    update Student_Answer set answer = (select top(1) answers from @ans) where current of c1
                    delete top(1) from @ans
                    fetch c1 into @q_id,@e_id,@s_id,@answer
          end
close c1
deallocate c1
```