

# **Online Examination System**

Under the Supervision of Eng.Rami Nagi

## **Team Members**

Reem Sami

Jessica Safwat

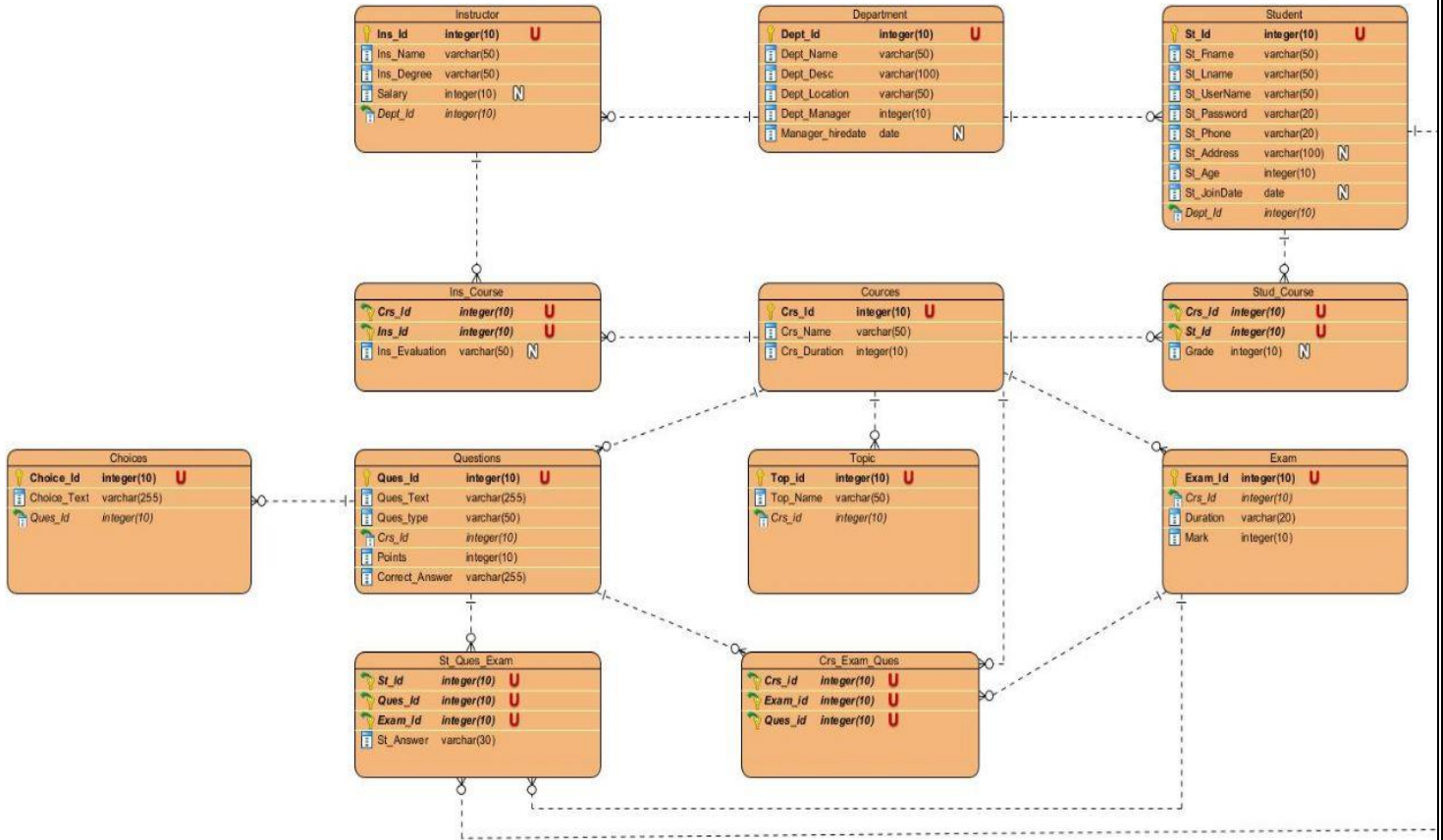
Tasbeeh Gamal

Claudine Bahgat

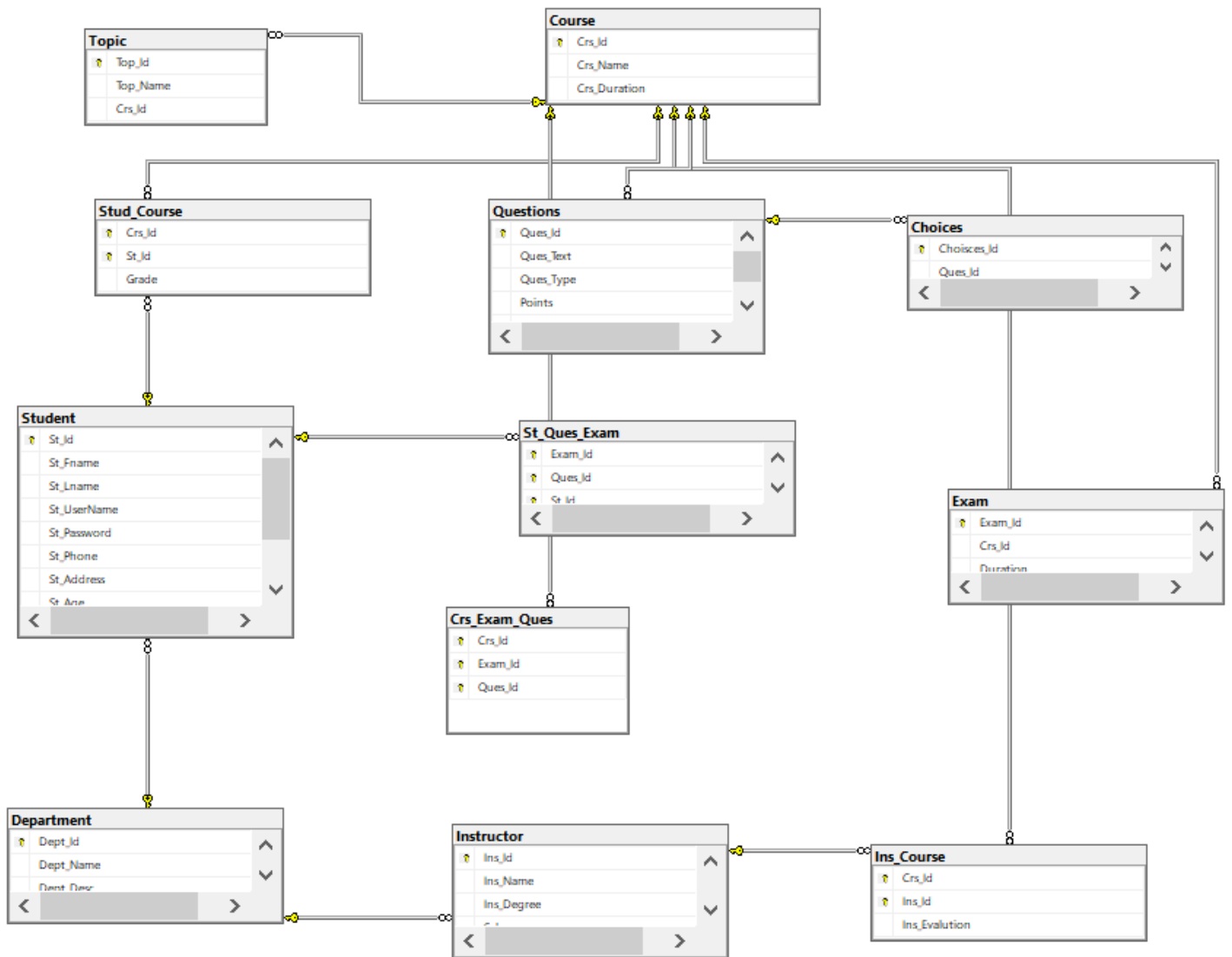
Nada Albasuony

Mennatallah Elsayed

# ERD



## Diagram



## Tables

### Topic:

	Top_Id	Top_Name	Crs_Id
▶	1	LinQ	1
	2	EF	1
	3	Agile	5
*	NULL	NULL	NULL

### Student:

	St_Id	St_Fname	St_Lname	St_UserNa...	St_Password	St_Phone	St_Address	St_Age	St_JoinDate	Dept_Id
▶	1	Nada	Al-Basuony	Nada_B	ITI_1234	+20101839...	Tanta	23	2022-10-09	10
	2	Tasbeeh	Gamal	Tasbeeh_G	ITI_7777	+20122649...	Alex	23	2022-10-09	10
	3	Mennatallah	Elsayed	Menna_E	ITI_6565	+20197549...	Alex	24	2022-10-09	10
	4	Jessica	Safwat	Jessica_S	ITI_2792	+20154019...	Alex	24	2022-10-09	10
	5	Reem	Samy	Reem_S	ITI_4321	+20154019...	Tanta	23	2022-10-09	10
	6	Claudine	Bahgat	Claudine_B	ITI_5656	+20127726...	Alex	23	2022-10-09	10
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

### Exam:

	Exam_Id	Crs_Id	Duration	Mark
▶	1	1	2H	60
	2	2	2H	60
	3	3	2H	60
	4	4	2H	60
	5	5	2H	60
*	NULL	NULL	NULL	NULL

### Question:

	Ques_Id	Ques_Text	Ques_Type	Points	Crs_Id	Correct_A
▶	1	What does t...	MCQ	6	3	Hypertext
	2	What is the ...	MCQ	6	3	h2
	3	Which tag d...	MCQ	6	3	 
	4	In HTML, th...	MCQ	6	3	not case se
	5	What is the ...	MCQ	6	3	<ol type =
	6	Comments ...	T/F	6	3	False
	7	<h1> tag is ...	T/F	6	3	True
	8	<p> tag in ...	T/F	6	3	False
	9	The "div" ta...	T/F	6	3	True
	10	Each choice...	T/F	6	3	True
	11	The HTML a...	MCQ	6	4	<style>
	12	Which of th...	MCQ	6	4	padding
	13	Which of th...	MCQ	6	4	text-transf
	14	Which of th...	MCQ	6	4	div~p
	15	The CSS pro...	MCQ	6	4	border-rac
	16	Which of th...	MCQ	6	4	:not select
	17	Which of th...	MCQ	6	4	text-overfl
	18	The correct ...	MCQ	6	4	text-decor
	19	The CSS pro...	MCQ	6	4	font-stretc
	20	Which if the...	MCQ	6	4	calc() func
	21	The link ele...	T/F	6	4	False
	22	Specifying a...	T/F	6	4	True
	23	To specify t...	T/F	6	4	False
	24	The CSS pro...	T/F	6	4	False
	25	The span el...	T/F	6	4	True
	26	The CSS pro...	T/F	6	4	False
	27	The CSS pro...	T/F	6	4	True
	28	The negativ...	T/F	6	4	False
	29	To display t...	T/F	6	4	False

**Instructor:**

	Ins_Id	Ins_Name	Ins_Degree	Salary	Dept_Id
►	100	Ghada Qad...	Manager	12000.0000	10
	101	Mahmoud ...	Head	15000.0000	10
	102	Marwa Ibra...	Manager	12000.0000	20
	103	Noha salah	Manager	12000.0000	30
	104	Noha Amer	Manager	12000.0000	40
	105	Iman Benda...	Specialist	10000.0000	10
	106	Rami Nagi	Specialist	12000.0000	10
	107	Rehab Faro...	Specialist	12000.0000	10
	108	Sara Ahmed	Specialist	10000.0000	10
*	NULL	NULL	NULL	NULL	NULL

**Department:**

	Dept_Id	Dept_Name	Dept_Desc	Dept_Locat...
►	10	PD	Profeession...	Alex
	20	UX/UI	UX & UI De...	Cairo
	30	Open Source	Linux Open ...	Alex
	40	DB	Data Base	Alex
*	NULL	NULL	NULL	NULL

### Crs\_Exam\_Ques:

	Crs_Id	Exam_Id	Ques_Id
	3	3	1
	3	3	2
▶	3	3	3
	3	3	4
	3	3	5
	3	3	7
	3	3	8
	3	3	9
*	NULL	NULL	NULL

### Course:

	Crs_Id	Crs_Name	Crs_Duration
▶	1	C#	12
	2	Sql Server	6
	3	Html5	3
	4	CSS3	3
	5	SW	8
	6	DB	4
	7	Soft Skills	6
	8	English	33
	9	Adobe Illust...	9
*	NULL	NULL	NULL

## Choices:

	Choisces_Id	Ques_Id	Choice1	Choice2	Choice3	Choice4
	12	17	text-shadow	text-stroke	text-overflow	text-decora...
	13	18	text-decora...	text-decora...	text-decora...	text-decora...
	14	19	font-stretch ...	font-weight...	font-transfo...	font-variant...
	15	20	calc() functi...	calculator() ...	calculate() f...	cal() function
	16	31	?:	is	as	*
	17	32	Value para...	Reference p...	Output para...	None of the...
	18	33	By using the...	Additional ...	Only one pa...	All of the a...
	19	34	A program ...	The catch k...	Both of the ...	None of the...
	20	35	8	4	2	1
	21	36	.NET Frame...	Java Virtual ...	Both of the ...	None of the...
	22	37	Web apps	Desktop ap...	Mobiles apps	All of the a...
	23	38	double -> fl...	double -> fl...	float -> dou...	float -> dou...
	24	39	Write() writ...	Write() writ...	Write() writ...	Both can be...
	25	40	for(;;)	for(;;);	for(;;)	for(1;1;1)
	26	41	int arr[2][3] ...	int arr[2,3] ...	int [,] arr = ...	int [,] arr = ...
	27	42	Static Const...	Private Cons...	Body Constr...	Parameteriz...
	28	55	Image orien...	Text, files co...	Data in the ...	All of the a...
	29	56	Image	Text	Table	Graph
	30	57	SELECT	EXTRACT	OPEN	GET
	31	58	Minimum D...	High Level ...	Single-user ...	Support ACL...
	32	59	Hyper data	Tera data	Meta data	Relations
	33	60	Domain rel...	Tuple relati...	Relational a...	Query langu...
	34	61	Dept_id sho...	Group by cl...	Avg(salary) ...	None
	35	69	Re-engineer...	Reverse eng...	Software re...	Science and...
	36	70	Project sche...	Detailed sc...	Macroscopi...	None of the...
	37	71	Cost	Effort appli...	Efficiency	All of the a...
	38	72	Developme...	Maintenanc...	Design	Analysis
	39	73	Iterative En...	RAD	Spiral	Waterfall



### St\_Ques\_Exam:

	Column Name	Data Type	Allow Nulls
PK	Exam_Id	int	<input type="checkbox"/>
FK	Ques_Id	int	<input type="checkbox"/>
FK	St_Id	int	<input type="checkbox"/>
	St_Answer	nvarchar(30)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

### Ins\_Course:

	Column Name	Data Type	Allow Nulls
PK	Crs_Id	int	<input type="checkbox"/>
FK	Ins_Id	int	<input type="checkbox"/>
	Ins_Evaluation	nvarchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

### Stud\_Course:

	Column Name	Data Type	Allow Nulls
PK	Crs_Id	int	<input type="checkbox"/>
FK	St_Id	int	<input type="checkbox"/>
	Grade	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

## Procedures

```
--student table
--1)insert
create procedure Insertstudent @St_Fname nvarchar(50),@St_Lname nvarchar(50),
@St_UserName nvarchar(50),@St_Password nvarchar(20),@St_Phone nvarchar(20),
@St_Address nvarchar(100),@St_Age int,@Dept_Id int
as
begin
If EXISTS(select @Dept_Id from Department where Dept_Id=@Dept_Id)
insert into Student (St_Fname,St_Lname,St_UserName,St_Password,St_Phone,St_Address,St_Age,Dept_Id)
values(@St_Fname,@St_Lname,@St_UserName,@St_Password,@St_Phone,@St_Address,@St_Age,@Dept_Id)
else
select 'Student cannot be Inserted'
End
--execute Insertstudent 'ahmed','ramy','ahmedramy','123','0000','alex',10,10

--student table
--2)Update

create procedure Updatestudent @St_Id int,@St_Fname nvarchar(50),@St_Lname nvarchar(50),
@St_UserName nvarchar(50),@St_Password nvarchar(20),@St_Phone nvarchar(20),
@St_Address nvarchar(100),@St_Age int,@Dept_Id int
as
begin
If EXISTS(select St_Id from Student where St_Id=@St_Id)
begin
If EXISTS(select @Dept_Id from Department where Dept_Id=@Dept_Id)
update Student set St_Fname=@St_Fname,St_Lname=@St_Lname, St_UserName=@St_UserName,St_Password=@St_Password,
St_Phone=@St_Phone,St_Address=@St_Address,St_Age=@St_Age ,Dept_Id=@Dept_Id where St_Id=@St_Id
else
select 'The is no Department with this ID'
End
else
select 'Student cannot be Updated'
end

--execute Updatestudent 10,'mennat','Elsayed','Mennat','123','0000','alex',10,10

--student table
--3)Delete
create procedure Deletestudent @St_Id int
as
begin
If EXISTS(select St_Id from Student where St_Id=@St_Id)
Delete from Student where St_Id=@St_Id
else
select 'Student cannot be Deleted'
end

--execute Deletestudent 5

*****
~

--Department table
--1)insert
create procedure insertDepartment @Dept_Id int,@Dept_Name nvarchar(50),@Dept_Desc nvarchar(50)
,@Dept_Location nvarchar(50)
as
begin
If NOT EXISTS (SELECT Dept_Id FROM Department WHERE Dept_Id = @Dept_Id)
INSERT INTO Department(Dept_Id, Dept_Name, Dept_Desc , Dept_Location )
VALUES (@Dept_Id,@Dept_Name , @Dept_Desc , @Dept_Location);
else
select 'Department cannot be Inserted'
end
```

```

--execute insertDepartment 90,'professional web','web track','alex'

--Department table
--2)Update

create procedure UpdateDepartment @Dept_Id int,@Dept_Name nvarchar(50),@Dept_Desc nvarchar(50)
,@Dept_Location nvarchar(50)
as
Begin
If EXISTS(select @Dept_Id from Department where Dept_Id=@Dept_Id)
update Department set Dept_Name=@Dept_Name ,Dept_Desc=@Dept_Desc , Dept_Location=@Dept_Location
where Dept_Id=@Dept_Id
else
select 'There is no Department with this ID'
End

--execute UpdateDepartment 30,'os','os track','cairo'

--Department table
--3)Delete
create procedure DeleteDepartment @Dept_Id int
as
begin
If EXISTS(select Dept_Id from Department where Dept_Id=@Dept_Id)
Delete from Department where Dept_Id=@Dept_Id
else
select 'Department cannot be Deleted'
end

--execute DeleteDepartment 90

--1)insert Instructor

create procedure InsertInstructor @Ins_Id int , @Ins_Name nvarchar(50),@Ins_Degree nvarchar(50),@Salary money, @Dept_Id int
as
Begin
If not EXISTS(select Ins_Id from Instructor where Ins_Id=@Ins_Id)
begin
If EXISTS(select Dept_Id from Department where Dept_Id=@Dept_Id)
INSERT INTO Instructor(Ins_Id, Ins_Name, Ins_Degree, Salary, Dept_Id )
VALUES (@Ins_Id,@Ins_Name, @Ins_Degree ,@Salary , @Dept_Id);
else
select 'Department not existed'
end
else
select 'Instructor cannot be inserted'
End

--execute InsertInstructor 119,'karem','PR',200000,10

--Instructor table
--2)Update Instructor

create procedure UpdateInstructor @Ins_Id int , @Ins_Name nvarchar(50),@Ins_Degree nvarchar(50),@Salary money, @Dept_Id int
as
Begin
If EXISTS(select Ins_Id from Instructor where Ins_Id=@Ins_Id)
begin
If EXISTS(select Dept_Id from Department where Dept_Id=@Dept_Id)
update Instructor set Ins_Name=@Ins_Name, Ins_Degree=@Ins_Degree , Salary=@Salary ,Dept_Id =@Dept_Id where Ins_Id=@Ins_Id
else
select 'Department not Existed'
end
else

```

```

select 'Student cannot be Updated'
end

--execute UpdateInstructor 1100,'sarah','pr','2000',10

--Instructor table
--3)Delete Instructor

create procedure DeleteInstructor @Ins_Id int
as
begin
If EXISTS(select Ins_Id from Instructor where Ins_Id=@Ins_Id)
Delete from Instructor where Ins_Id=@Ins_Id
else
select 'Instructor cannot be Deleted'
end

--execute DeleteInstructor 119
*****

--Ins-Course table
--1)Insert Ins-Course

create procedure insert_Ins_Course @Crs_Id int,@Ins_Id int
as
begin
If Exists (select Crs_Id , Ins_Id from Course, Instructor where Crs_Id=@Crs_Id and Ins_Id=@Ins_Id)
begin
if not Exists( select * from Ins_Course where Crs_Id=@Crs_Id and Ins_Id=@Ins_Id )
INSERT INTO Ins_Course(Crs_Id, Ins_Id )
VALUES (@Crs_Id,@Ins_Id);
else
select 'Not Inserted'
end

else
select 'Instructor or course not existed'
end

--execute insert_Ins_Course 9,108
--execute insert_Ins_Course 100,102

--Ins-Course table
--2)Update Ins-Course

--Ins-Course table
--3)Delete Ins-Course

--std-Course table
--1)Insert std-Course

create procedure insert_std_Course @Crs_Id int,@St_Id int
as
begin
If Exists (select Crs_Id , St_Id from Course, Student where Crs_Id=@Crs_Id and St_Id=@St_Id)
begin
if not Exists( select * from Stud_Course where Crs_Id=@Crs_Id and St_Id=@St_Id )
INSERT INTO Stud_Course(Crs_Id, St_Id )
VALUES (@Crs_Id,@St_Id);
else
select 'Not Inserted'
end
else
select 'Student or course not existed'
end

```

```

        Select 'Already exist'
    End
    Go
    --course update
    create proc course_update @crsId int, @crsName nvarchar(50), @crsDuration int
    as
    Begin
    IF EXISTS (SELECT Crs_Id FROM Course where Crs_Id = @crsId)
        UPDATE Course SET Crs_Id = @crsId, Crs_Name = @crsName, Crs_Duration = @crsDuration WHERE Crs_Id = @crsId
    ELSE
        select 'not exist'
    End
    Go
    --course delete
    create proc course_delete @crsId int
    as
    Begin
    IF EXISTS (SELECT Crs_Id FROM Course where Crs_Id = @crsId)
        DELETE from Course where Crs_Id = @crsId
    ELSE
        select 'not exist'
    End
    Go
    ----- topic -----
    --insert topic
    create proc topic_insert @topicId int, @topicName nvarchar(10), @crsId int
    as
    Begin
    IF not EXISTS (SELECT Top_Id FROM Topic WHERE Top_Id = @topicId)
        INSERT INTO Topic
        VALUES (@topicId, @topicName, @crsId)
    ELSE
        Select 'Already exist'

```

```

--std-Course table
--2)Update Ins-Course

--std-Course table
--3)Delete Ins-Course

--Crs_Exam_Ques table
--1)Insert Crs_Exam_Ques

create procedure insert_Crs_Exam_Ques @Crs_Id int,@Exam_Id int ,@Ques_Id int
as
begin
If Exists (select c.Crs_Id ,Exam_Id,Ques_Id from Course as c ,Exam ,Questions as q where c.Crs_Id=@Crs_Id and Exam_Id=@Exam_Id and Ques_Id=@Ques_Id)
begin
if not Exists( select Crs_Id,Exam_Id,Ques_Id from Crs_Exam_Ques where Crs_Id=@Crs_Id and Exam_Id=@Exam_Id and Ques_Id=@Ques_Id )
INSERT INTO Crs_Exam_Ques (Crs_Id, Exam_Id,Ques_Id )
VALUES (@Crs_Id,@Exam_Id,@Ques_Id);
else
select 'Not Inserted'
end
else
select ' Crs_Exam_Ques not existed'
end
Go
----- COURSE TABLE -----
--insert course
create proc course_insert @crsId int, @crsName nvarchar(50), @crsDuration int
as
Begin
IF not EXISTS (SELECT Crs_Id FROM Course WHERE Crs_Name = @crsName)
    INSERT INTO Course
    VALUES (@crsId, @crsName, @crsDuration)
ELSE

```

```

End
Go
-- topic update
create proc topic_update @topicId int, @topicName nvarchar(10), @crsId int
as
Begin
IF EXISTS (SELECT Top_Id FROM Topic WHERE Top_Id = @topicId)
Update Topic set Top_Id = @topicId, Top_Name = @topicName, Crs_Id = @crsId where Top_Id = @topicId
ELSE
Select 'Not exist'
End
Go
-- topic delete
create proc topic_delete @topicId int
as
Begin
IF EXISTS (SELECT Top_Id FROM Topic WHERE Top_Id = @topicId)
Delete from Topic where Top_Id = @topicId
ELSE
Select 'Not exist'
End
Go
----- Questions -----
--insert question
Go
Create proc ques_Choices_insert @quesId int, @quesText varchar(255), @quesType varchar(50), @crsId int, @points int, @correctAnswer varchar(255), @choice
as
Begin
IF Not EXISTS (SELECT Ques_Id FROM Questions WHERE Ques_Id = @quesId)
Begin
IF (@quesType = 'T/F')
INSERT INTO Questions
VALUES (@quesId, @quesText, @quesType, @crsId, @points, @correctAnswer)

```

```

Else
Begin
INSERT INTO Questions
VALUES (@quesId, @quesText, @quesType, @crsId, @points, @correctAnswer)
INSERT INTO Choices
VALUES (@quesId, @choice1, @choice2, @choice3, @choice4)
End
End
ELSE
Select 'Already Exist'
End
Go
-- update question
create proc ques_Choices_update @quesId int, @quesText varchar(255), @quesType varchar(50), @crsId int, @points int, @correctAnswer varchar(255), @choic
as
Begin
IF EXISTS (SELECT Ques_Id FROM Questions WHERE Ques_Id = @quesId)
Begin
IF (@quesType = 'T/F')
UPDATE Questions
SET Ques_Text = @quesText, Ques_Type = @quesType, Crs_Id = @crsId, Points = @points, Correct_Answer = @correctAnswer
where Ques_Id = @quesId
Else
Begin
UPDATE Questions
SET Ques_Text = @quesText, Ques_Type = @quesType, Crs_Id = @crsId, Points = @points, Correct_Answer = @correctAnswer
where Ques_Id = @quesId
UPDATE Choices
SET Choice1 = @choice1, Choice2 = @Choice2, Choice3 = @choice3, Choice4 = @choice4
where Ques_Id = @quesId
End
End
ELSE

```

```

        Select 'Not Exist'
    End
Go
--delete question
create proc ques_delete @quesId int
as
Begin
    IF EXISTS (SELECT Ques_Id FROM Questions WHERE Ques_Id = @quesId)
        Delete from Questions where Ques_Id = @quesId
    ELSE
        Select 'Not exist'
    End
Go

----- Exam -----
--exam insert
create proc Exam_insert @crsId int, @Duration nvarchar(20), @mark int
as
Begin
    INSERT INTO Exam
    VALUES (@crsId, @Duration, @mark)
End
Go
--exam update
create proc exam_update @examId int, @crsId int, @Duration nvarchar(20), @mark int
as
Begin
    IF EXISTS (SELECT Exam_Id FROM Exam WHERE Exam_Id = @examId)
        UPDATE Exam
        SET Crs_Id = @crsId, Duration = @Duration, Mark = @mark
        where Exam_Id = @examId
    ELSE
        Select 'Not Exist'
    End
End

```

```

--exam delete
create proc exam_delete @examId int
as
Begin
    IF EXISTS (SELECT Exam_Id FROM Exam WHERE Exam_Id = @examId)
        Delete from Exam where Exam_Id = @examId
    ELSE
        Select 'Not exist'
    End
GO

----- ST_Ques_Exam -----
--insert (procedure Answers)
create proc ST_Ques_Exam_insert @examId int, @quesId int, @stdId int, @stdAnswer varchar(255)
as
begin
    IF EXISTS (SELECT Exam_Id,Ques_Id,St_Id FROM Exam, Questions, Student WHERE Exam_Id=@examId and Ques_Id = @quesId and St_Id = @stdId)
        begin
            if not Exists( select * from St_Ques_Exam where Exam_Id=@examId and Ques_Id = @quesId and St_Id = @stdId )
                INSERT INTO St_Ques_Exam
                VALUES (@examId, @quesId,@stdId,@stdAnswer)
            else
                select 'Not Inserted'
            end
        else
            select ' Crs_Exam_Ques not existed'
        end
    end
Go

----- exam generation -----
create procedure ExamGeneration @Crs_Id int ,@MCQ_No int,@T_F_No int
as
begin
    if Exists (select Crs_Id from Course where Crs_Id=@Crs_Id)
begin |

```

```

declare @Ex_id int
insert into Exam (Crs_Id)VALUES(@Crs_Id)
select @Ex_id=Exam_Id from Exam where Crs_Id=@Crs_Id

insert into Crs_Exam_Ques (Crs_Id,Exam_Id,Ques_Id)
SELECT TOP (@MCQ_No) Crs_Id,@Ex_id,Ques_Id FROM Questions WHERE Crs_Id = @Crs_Id and Ques_Type = 'MCQ' ORDER BY NEWID();

insert into Crs_Exam_Ques (Crs_Id, Exam_Id,Ques_Id)
SELECT TOP (@T_F_No) Crs_Id,@Ex_id,Ques_Id FROM Questions WHERE Crs_Id = @Crs_Id and Ques_Type = 'T/F' ORDER BY NEWID();
end
else
select 'No course with this ID'
end
--execute ExamGeneration 1,3,7
Go

----- Correction Procedure -----
create proc Exam_Correction @stdId int, @examid int
as
Begin
IF EXISTS (SELECT st_Id, Exam_Id FROM St_Ques_Exam WHERE St_Id = @stdId AND Exam_Id = @examid)
Begin
Declare c1 cursor for
SELECT Ques_Id ,St_Answer FROM St_Ques_Exam WHERE St_Id = @stdId AND Exam_Id = @examid FOR READ ONLY
Declare @quesId int , @stdAnswer varchar(255), @grade int = 0
open c1
FETCH c1 INTO @quesId, @stdAnswer
WHILE @@FETCH_STATUS = 0
Begin
IF @stdAnswer = (SELECT Correct_Answer FROM Questions WHERE Ques_Id = @quesId)
SET @grade += 6
FETCH c1 INTO @quesId, @stdAnswer

```

---

```

End
CLOSE c1
DEALLOCATE c1
Declare @crsId int
SELECT @crsId = (SELECT Crs_Id From Crs_Exam_Ques WHERE Ques_Id = @quesId)
UPDATE Stud_Course SET Grade = @grade WHERE St_Id = @stdId AND Crs_Id = @crsId
End
GO
ques_insert 100, 'hhhhh', 'mcq', 1, 60 , 'a'
Go
ques_update 100, 'mmmmmm', 'T\F', 1, 60 , 'T'
Go
course_insert 10 , 'ado', 20
Go
course_delete 10
Go
insert into Topic values(1,'Linq',1);
insert into Topic values(2,'EF',1);
insert into Topic values(3,'Agile',5);
Go
topic_update 4, 'ado.Net', 10
topic_update 1, 'Linq', 10
Go
topic_insert 4, 'ado.Net', 1
Go
topic_delete 4
Go
ques_Choices_insert 121, 'mmm?', 'MCQ', 1,60,'a is answer','a','b','c','d'
Go
ques_Choices_update 120, 'gggg?', 'T/F', 1,60,'False'
Go
ques_Choices_update 121, 'mmYmm?', 'MCQ', 1,60,'bb is answer','aa','bb','cc','dd'

```



## Report Procedures

```

/*****Reports*****/
-----One-----
create procedure GetSetudentByDepartment
@Dept_Id int
as
begin
select * from Student where Dept_Id = @Dept_Id
end
--GetSetudentByDepartment @Dept_Id = 10

-----Two-----
alter PROCEDURE GetStudentGrades
@StudentID INT
AS
BEGIN
SELECT cc.Crs_Name,
        FLOOR((sc.Grade*0.01)*c.Mark) AS Grade_Percentage
FROM Stud_Course AS sc
JOIN Exam AS c ON sc.Crs_Id = c.Crs_Id
JOIN Course AS cc ON cc.Crs_Id =sc.Crs_Id
WHERE sc.St_Id = @StudentID
END
-- GetStudentGrades @StudentID = 1

-----Three-----
CREATE PROCEDURE GetInstructorCourses
@InstructorID INT
AS
BEGIN
SELECT c.Crs_Name, COUNT(sc.St_Id) AS NumberOfStudents
FROM Course AS c
JOIN Stud_Course AS sc ON c.Crs_Id = sc.Crs_Id
join Ins_Course as v on v.Crs_Id= sc.Crs_Id
WHERE v.Ins_Id = @InstructorID
GROUP BY c.Crs_Name
END
--GetInstructorCourses @InstructorID = 106

-----Four-----
alter proc TopicReport
@CourseID int
as
select Crs_Id,Top_Name
from Topic
where Crs_Id = @CourseID
--TopicReport @CourseID = 5

-----Five-----
alter proc QuesExamChoiceReport
@ExamNumber int
as
select q.Ques_Text, c.Choice1, c.Choice2, c.Choice3, c.Choice4
from Questions q, Choices c, Exam e, Crs_Exam_Ques ceq
where e.Exam_Id = ceq.Exam_Id
and q.Ques_Id = ceq.Ques_Id

```

```

    and q.Ques_Id = ceq.Ques_Id
    and e.Exam_Id = @ExamNumber
    and q.Ques_Id = c.Ques_Id
--QuesExamChoiceReport @ExamNumber =1

-----
--ExamGeneration 3,9,3
-----Six-----
create proc StudExamQuesReport
    @ExamNumber int
    ,@StudentID int
as
select q.Ques_Text , sqe.St_Answer
from Questions q, Exam e, St_Ques_Exam sqe
where q.Ques_Id = sqe.Ques_Id
    and e.Exam_id = sqe.Exam_Id
    and e.Exam_Id = @ExamNumber
    and sqe.St_Id = @StudentID
    StudExamQuesReport 1,2

--ST_Ques_Exam_insert 2,1,2,'Hypertext Markup Language'

```