no

Examination system Dictionary

Data Dictionary

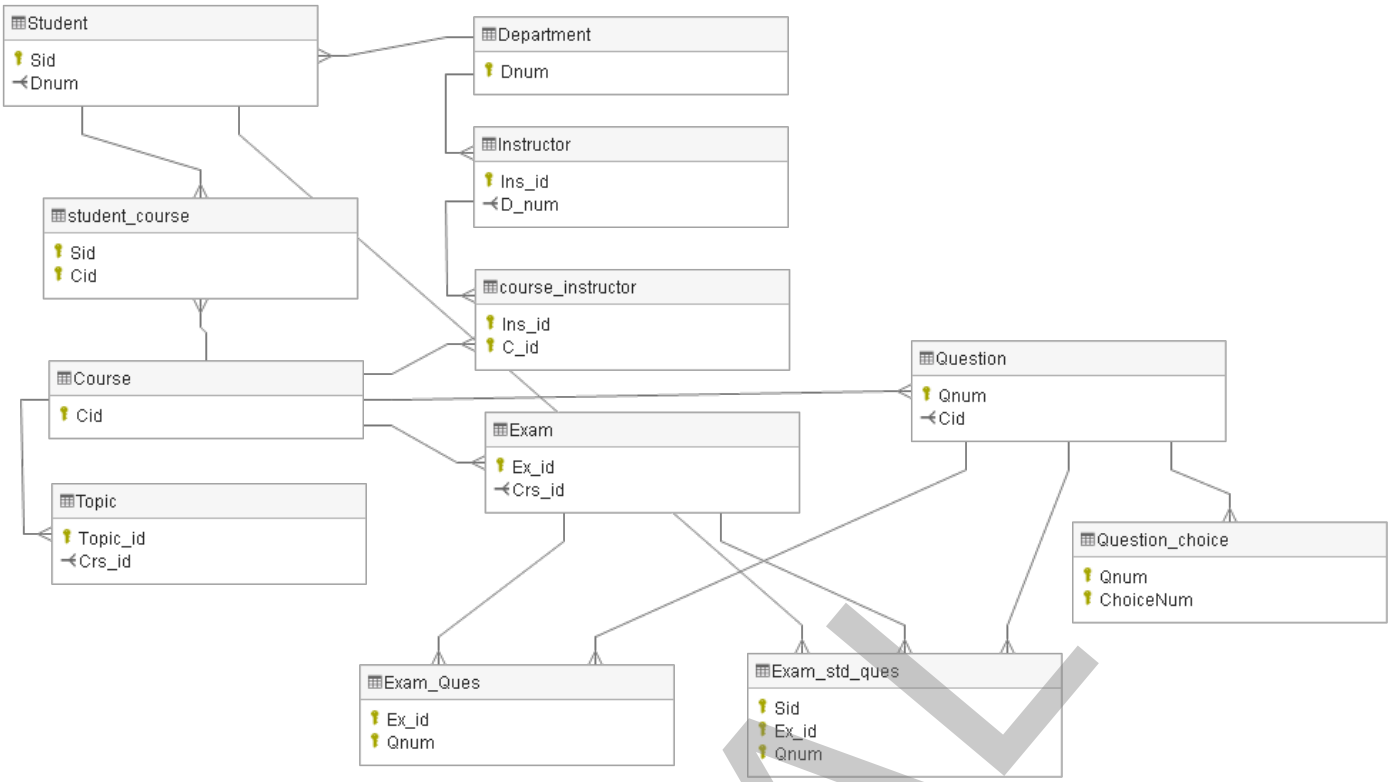
1/20/2021

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

1. Digaram



this is diagram show the relation between all tables in database

2. Other

2.1. Tables

2.1.1. Table: Course

Columns

		Name	Data type	Description / Attributes
		Cid	int	
		Cname	varchar(50)	**Nullable**

Linked from

	Table	Join	Title / Name / Description
	course_instructor	**Course**Cid = course_instructorC_id	FK_course_instructor_Course
	Exam	**Course**Cid = ExamCrs_id	FK_Exam_Course
	Question	**Course**Cid = QuestionCid	FK_Question_Course
	student_course	**Course**Cid = student_courseCid	FK_student_course_Course
	Topic	**Course**Cid = TopicCrs_id	FK_Topic_Course1

Unique keys

		Name / Description
	Cid	PK_Course

2.1.2. Table: course_instructor

Columns

		Name	Data type	Description / Attributes
		Ins_id	int	**References**: Instructor
		C_id	int	**References**: Course

Links to

	Table	Join	Title / Name / Description
	Course	**course_instructor**C_id = CourseCid	FK_course_instructor_Course
	Instructor	**course_instructor**Ins_id = InstructorIns_id	FK_course_instructor_Instructor

Unique keys

		Name / Description
	Ins_id, C_id	PK_course_instructor

2.1.3. Table: Department

Columns

		Name	Data type	Description / Attributes
		Dnum	int	

		Name	Data type	Description / Attributes
		Dname	varchar(50)	**Nullable**

Linked from

	Table	Join	Title / Name / Description
	Instructor	**Department**Dnum = InstructorD_num	FK_Instructor_Department
	Student	**Department**Dnum = StudentDnum	FK_Student_Department

Unique keys

		Name / Description
	Dnum	PK_De

2.1.4. Table: Exam

Columns

		Name	Data type	Description / Attributes
		Ex_id	int	
		NoOfMCQ	int	**Nullable**
		NoOfTF	int	**Nullable**
		Crs_id	int	**Nullable** **References**: Course

Links to

	Table	Join	Title / Name / Description
	Course	**Exam**Crs_id = CourseCid	FK_Exam_Course

Linked from

	Table	Join	Title / Name / Description
	Exam_Ques	**Exam**Ex_id = Exam_QuesEx_id	FK_Exam_Ques_Exam
	Exam_std_ques	**Exam**Ex_id = Exam_std_quesEx_id	FK_Exam_std_ques_Exam

Unique keys

		Name / Description
	Ex_id	PK_Exam

2.1.5. Table: Exam_Ques

Columns

		Name	Data type	Description / Attributes
		Ex_id	int	**References**: Exam
		Qnum	int	**References**: Question

Links to

	Table	Join	Title / Name / Description
	Exam	**Exam_Ques**Ex_id = ExamEx_id	FK_Exam_Ques_Exam

	Table	Join	Title / Name / Description
⟜	Question	**Exam_Ques**Qnum = QuestionQnum	FK_Exam_Ques_Question

Unique keys

		Name / Description
🔑	Ex_id, Qnum	PK_Exam_Ques

2.1.6. Table: Exam_std_ques

Columns

		Name	Data type	Description / Attributes
▤	🔑	Sid	int	**References**: Student
▤	🔑	Ex_id	int	**References**: Exam
▤	🔑	Qnum	int	**References**: Question
▤		std_Ans	varchar(300)	**Nullable**
▤		std_Ques_grade	int	**Nullable**

Links to

	Table	Join	Title / Name / Description
⟜	Exam	**Exam_std_ques**Ex_id = ExamEx_id	FK_Exam_std_ques_Exam
⟜	Question	**Exam_std_ques**Qnum = QuestionQnum	FK_Exam_std_ques_Question
⟜	Student	**Exam_std_ques**Sid = StudentSid	FK_Exam_std_ques_Student

Unique keys

		Name / Description
🔑	Sid, Ex_id, Qnum	PK_Exam_std_ques

2.1.7. Table: Instructor

Columns

		Name	Data type	Description / Attributes
▤	🔑	Ins_id	int	
▤		ins_name	varchar(50)	**Nullable**
▤		D_num	int	**Nullable** **References**: Department

Links to

	Table	Join	Title / Name / Description
⟜	Department	**Instructor**D_num = DepartmentDnum	FK_Instructor_Department

Linked from

	Table	Join	Title / Name / Description
⟝	course_instructor	**Instructor**Ins_id = course_instructorIns_id	FK_course_instructor_Instructor

Unique keys

		Name / Description
	Ins_id	PK_Instructor

2.1.8. Table: Question

Columns

		Name	Data type	Description / Attributes
		Qnum	int	
		Qtype	varchar(5)	**Nullable**
		Qhead	varchar(300)	**Nullable**
		Model_Ans	varchar(300)	**Nullable**
		fullmark_grade	int	**Nullable**
		Cid	int	**Nullable** **References**: Course

Links to

	Table	Join	Title / Name / Description
	Course	**Question**Cid = CourseCid	FK_Question_Course

Linked from

	Table	Join	Title / Name / Description
	Exam_Ques	**Question**Qnum = Exam_QuesQnum	FK_Exam_Ques_Question
	Exam_std_ques	**Question**Qnum = Exam_std_quesQnum	FK_Exam_std_ques_Question
	Question_choice	**Question**Qnum = Question_choiceQnum	FK_Question_choice_Question

Unique keys

		Name / Description
	Qnum	PK_Question

2.1.9. Table: Question_choice

Columns

		Name	Data type	Description / Attributes
		Qnum	int	**References**: Question
		ChoiceNum	int	
		ChoiceBody	varchar(300)	**Nullable**

Links to

	Table	Join	Title / Name / Description
	Question	**Question_choice**Qnum = QuestionQnum	FK_Question_choice_Question

Unique keys

		Name / Description
	Qnum, ChoiceNum	PK_Question_choice

2.1.10. Table: Student

Columns

		Name	Data type	Description / Attributes
		Sid	int	
		Sname	varchar(50)	**Nullable**
		Email	varchar(50)	**Nullable**
		Dnum	int	**Nullable** **References**: Department
		password	varchar(50)	**Nullable**

Links to

	Table	Join	Title / Name / Description
	Department	**Student**Dnum = DepartmentDnum	FK_Student_Department

Linked from

	Table	Join	Title / Name / Description
	Exam_std_ques	**Student**Sid = Exam_std_quesSid	FK_Exam_std_ques_Student
	student_course	**Student**Sid = student_courseSid	FK_student_course_Student

Unique keys

		Name / Description
	Sid	PK_Student

2.1.11. Table: student_course

Columns

		Name	Data type	Description / Attributes
		Sid	int	**References**: Student
		Cid	int	**References**: Course
		crs_grade	int	**Nullable** **Default**: 0

Links to

	Table	Join	Title / Name / Description
	Course	**student_course**Cid = CourseCid	FK_student_course_Course
	Student	**student_course**Sid = StudentSid	FK_student_course_Student

Unique keys

		Name / Description
	Sid, Cid	PK_student_course

2.1.12. Table: Topic

Columns

		Name	Data type	Description / Attributes
		Topic_id	int	
		Topic_name	varchar(50)	**Nullable**
		Crs_id	int	**References**: Course

Links to

	Table	Join	Title / Name / Description
	Course	**Topic**Crs_id = CourseCid	FK_Topic_Course1

Unique keys

		Name / Description
	Topic_id	PK_Topic

2.2. Procedures

2.2.1. Procedure: deletecourse

this is responsable about delete Course From Course table

Input/Output

	Name	Data type	Description
@	cid	int	Couse id

Uses

	Name
	deletecourse
	Course

2.2.2. Procedure: deleteCoursesInstuctor

this is responsable about delete Course From Course_Instructor table

Input/Output

	Name	Data type	Description
@	Ins_Id	int	instructor id
@	Cid	int	course id

Uses

	Name
	deleteCoursesInstuctor
	course_instructor

2.2.3. Procedure: deleteDepartment

this is responsable about delete Department From Department table

Input/Output

	Name	Data type	Description
→@	DeptNum	int	Department Number

Uses

	Name
deleteDepartment	
Department	

2.2.4. Procedure: deleteInstructor

this is responsable about delete Instructor From Instructor table

Input/Output

	Name	Data type	Description
→@	ins_id	int	instructor id

Uses

	Name
deleteInstructor	
Instructor	

2.2.5. Procedure: deleteQuesChoice

this is responsable about delete Question Choice From Question_Choice table

Input/Output

	Name	Data type	Description
→@	QuesNum	int	Question number
→@	ChoiceNum	int	Choice Number

Uses

	Name
deleteQuesChoice	
Question_choice	

Script

```
create Proc deleteQuesChoice @QuesNum int, @ChoiceNum int
as
begin try
if  exists (select Qnum,ChoiceNum from Question_choice where Qnum=@QuesNum and ChoiceNum=@ChoiceNum )
          begin
                    delete from Question_choice
                    where Qnum=@QuesNum and ChoiceNum=@ChoiceNum
          end
          else
                    select 'Question num or choice num not found'
end try
begin catch
          Select 'error'
end catch
```

2.2.6. Procedure: deleteQuestion

this is responsable about delete question From Question table

Input/Output

	Name	Data type	Description
→@	Qnum	int	Question Number

Uses

	Name
deleteQuestion	
Question	

2.2.7. Procedure: deleteStdCrs

this is responsable about delete Course for Student From Student_Course table

Input/Output

	Name	Data type	Description
→@	std_id	int	Student Id
→@	course_id	int	Course Id

Uses

	Name
deleteStdCrs	
student_course	

Script

```
create Proc deleteStdCrs @std_id int,@course_id int
as
begin try
if  exists (select Sid,Cid from student_course where Sid=@std_id and Cid=@course_id)
          begin
                    delete from student_course
                    where  Sid=@std_id and Cid=@course_id
          end
          else
                    select 'Student Id is not found'
end try
begin catch
          Select 'error'
end catch
```

2.2.8. Procedure: deleteTopic

this is responsable about delete Topic From Topic table

Input/Output

	Name	Data type	Description
→@	TopicId	int	Topic Id

Uses

	Name
deleteTopic	
Topic	

Script

```
CREATE Proc deleteTopic @TopicId int
as
begin try
if  exists (select Topic_id from Topic where Topic_id=@TopicId )
           begin
                      delete from Topic
                      where Topic_id=@TopicId
           end
           else
                      select 'Topic Id not found'
end try
begin catch
           Select 'error'
end catch
```

2.2.9. Procedure: ExamAnswers

Input/Output

	Name	Data type	Description
→@	std_Id	int	Student Id
→@	examId	int	Exam Id
→@	Qnum	int	Question Number
→@	Answer	varchar(300)	Answrer

Uses

	Name
ExamAnswers	
Exam	
Exam_std_ques	
Question	
Student	

Script

```
create proc ExamAnswers @std_Id int,@examId int,@Qnum int,@Answer varchar(300)
as
          if exists(select Sid from Student where Sid=@std_Id)
          begin
          if exists(select Ex_id from Exam where Ex_id=@examId)
          begin
          if exists(select Qnum from Question where Qnum=@Qnum)
          begin
          insert into Exam_std_ques(Sid,Ex_id,Qnum,std_Ans)
          values(@std_Id,@examId,@Qnum,@Answer)
          end
          else
          select 'Question num not found'
          end
          else
          select 'Exam id not found'
          end
          else
          select 'student id not found'
```

2.2.10. Procedure: ExamCorrection

Input/Output

	Name	Data type	Description
→@	stdID	int	Student ID
→@	examId	int	Exam ID

Uses

	Name
ExamCorrection	
Exam	
Exam_std_ques	
Question	
student_course	

Script

```
CREATE proc ExamCorrection @stdID int, @examId int
as
declare c1 cursor
for select esq.Ex_id,esq.Sid,esq.Qnum,esq.std_Ans,q.Model_Ans
from Exam_std_ques esq inner join Question q
on Q.Qnum=esQ.Qnum
where esq.Ex_id=@examId and Sid=@stdID

declare @Exam_ID int,@StudId int,@quesNum int ,@Student_Ans varchar(300),@Model_Ans varchar(300)
open c1
fetch c1 into @Exam_ID,@studId,@quesNum,@Student_Ans,@Model_Ans
while @@FETCH_STATUS=0
          begin
                    declare @x int
                    SELECT SOUNDEX(@Student_Ans) ,SOUNDEX(@Model_Ans)
                    select @x=DIFFERENCE(@Student_Ans,@Model_Ans);
                    if @x=4
                    begin
                              update Exam_std_ques
                              set std_Ques_grade =(select fullmark_grade from Question where Qnum=@quesNum)
                              where Ex_id=@Exam_ID and Sid=@StudId and Qnum=@quesNum
                    end
                    else
                    begin
                    update Exam_std_ques
                    set std_Ques_grade=0
                    where Ex_id=@Exam_ID and Sid=@StudId and Qnum=@quesNum
                    end
      fetch c1 into @Exam_ID,@studId,@quesNum,@Student_Ans,@Model_Ans
   end
close c1
deallocate c1

update student_course
set crs_grade=(select sum(std_Ques_grade)
          from Exam_std_ques
          where Ex_id=@examId and Sid=@stdID)
where Sid=@stdID and Cid =(select Crs_id
                                        from Exam
                                        where Ex_id = @examId )
```

2.2.11. Procedure: GenerateExam

Input/Output

	Name	Data type	Description
→@	ExamId	int	Exam Id
→@	courseName	varchar(50)	Course Name
→@	NoOfMCQ	int	Number Of Multi Choice Question
→@	NoOfTF	int	Number Of True Or False

Uses

	Name
	GenerateExam
	Course
	Exam
	Exam_Ques
	question

Script

```
CREATE proc GenerateExam @ExamId int ,@courseName varchar(50),@NoOfMCQ int ,@NoOfTF int
as
           if exists (select Ex_id from Exam where Ex_id=@ExamId)
           select 'there is already this ID '
           else
           begin
           if @NoOfTF+@NoOfMCQ =10
           begin

           if exists(select Cid from Course where Cname=@courseName)
           begin
           declare @Cid int
           select @Cid=Cid
           from Course
           where Cname=@courseName

           insert into Exam (Ex_id,NoOfMCQ,NoOfTF,Crs_id)
           values(@ExamId,@NoOfMCQ,@NoOfTF,@Cid)



           insert into Exam_Ques(Qnum,Ex_id)
           select top(@NoOfMCQ)Qnum,@ExamId
           from question
           where Qtype='MCQ' and Cid=@Cid
           order by newid()

           insert into Exam_Ques(Qnum,Ex_id)
           select top(@NoOfTF)Qnum,@ExamId
           from question
           where Qtype='T/F' and Cid=@Cid
           order by newid()
           end
           else
           select 'This course is not found'
           end
           else
           select 'No of questions must be 10'
           end
```

2.2.12. Procedure: get_ques

Input/Output

	Name	Data type	Description
→@	exam_num	int	Exam ID

Uses

	Name
get_ques	
Exam_Ques	
Question	

Script

```
CREATE proc get_ques @exam_num int
as
select Question.Qhead as Questions from  Question
inner join Exam_Ques on  Question.Qnum =Exam_Ques.Qnum
 where Exam_Ques.Ex_id = @exam_num
```

2.2.13. Procedure: Getall

Input/Output

	Name	Data type	Description
→@	st_id	int	Student Id
→@	crs_id	int	Course Id

Uses

	Name
Getall	
student_course	

Script

```
create Proc Getall @st_id int, @crs_id int
as
          select *
          from student_course
          where Sid = @st_id and Cid=@crs_id
```

2.2.14. Procedure: GetStd

Input/Output

	Name	Data type	Description
@	st_id	int	Student ID

Uses

	Name
GetStd	
Student	

Script

```
create Proc GetStd @st_id int
as
          select *
          from Student
          where Sid = @st_id
```

2.2.15. Procedure: getStdInfo

Input/Output

	Name	Data type	Description
@	DepartmentNo	int	Department Number

Uses

	Name
getStdInfo	
Student	

Script

```
create proc getStdInfo @DepartmentNo int
as
          select Sid as [Student ID],
           Sname as [Student Name],
           Email as [Student Email],
           Dnum as [Department No]
          from Student
          where Dnum=@DepartmentNo
```

2.2.16. Procedure: getStudentGrade

Input/Output

	Name	Data type	Description
→@	student_Id	int	Student ID

Uses

	Name
	getStudentGrade
	Course
	Student
	student_course

2.2.17. Procedure: GetTopics

Input/Output

	Name	Data type	Description
→@	CourseID	int	Course ID

Uses

	Name
	GetTopics
	Topic

Script

```
create proc GetTopics @CourseID int
as
          select Topic_id as [Topic Id],
                    Topic_name as [Topic Name],
                    Crs_id as [course Id]
                    from Topic
                    where Crs_id=@CourseID
```

2.2.18. Procedure: insertcourse

Input/Output

	Name	Data type	Description
→@	cid	int	Course ID
→@	cname	varchar(50)	Course Name

Uses

	Name
	insertcourse
	Course

2.2.19. Procedure: InsertCoursesInstuctor

Input/Output

	Name	Data type	Description
@	Ins_Id	int	Instructor ID
@	CourseId	int	Course ID

Uses

Name
InsertCoursesInstuctor
Course
course_instructor

2.2.20. Procedure: insertDepartment

Input/Output

	Name	Data type	Description
@	DeptNum	int	Department Num
@	DeptName	varchar(50)	Department Name

Uses

Name
insertDepartment
Department

2.2.21. Procedure: insertInstructor

Input/Output

	Name	Data type	Description
@	ins_id	int	Instructor Id
@	ins_name	varchar(50)	Instructor Name
@	D_num	int	Deparment Number

Uses

Name
insertInstructor
Instructor

2.2.22. Procedure: insertQuestion

Input/Output

	Name	Data type	Description
@	Qnum	int	Question Number
@	QType	varchar(5)	Question Type
@	Qhead	varchar(300)	Question Head
@	Model_Ans	varchar(50)	Model Answer

	Name	Data type	Description
@	fullmark_grade	int	FullMark_Grade
@	Cid	int	Course Id

Uses

	Name
	insertQuestion
	Question

2.2.23. Procedure: insertQuestionChoice

Input/Output

	Name	Data type	Description
@	QuesNum	int	Question Number
@	choiceNum	int	Choice Number
@	chiceBody	varchar(300)	Choice Body

Uses

	Name
	insertQuestionChoice
	Question_choice

Script

```
create Proc insertQuestionChoice  @QuesNum int,@choiceNum int,@chiceBody varchar(300)
as
begin try
if not exists (select Qnum,ChoiceNum from Question_choice where Qnum=@QuesNum and ChoiceNum=@QuesNum  )
           begin
           insert into Question_choice
           values (@QuesNum,@choiceNum,@chiceBody)
           end
           else
                     select 'Duplicate Topic Id'
end try
begin catch
           Select 'error'
end catch
```

2.2.24. Procedure: insertStdCrs

Input/Output

	Name	Data type	Description
@	st_id	int	Student ID
@	Crs_id	int	Course ID
@	grade	int	Final Grade

Uses

	Name
	insertStdCrs
	student_course

Script

```
create proc insertStdCrs @st_id int,@Crs_id int,@grade int
as
begin try
		if not exists (select Sid from student_course where Sid=@st_id )
begin
		insert into student_course
				values (@st_id,@Crs_id,@grade)
		end
		else
				select 'Duplicate Student Id'
end try
begin catch
		Select 'error'
end catch
```

2.2.25. Procedure: insertStudent

Input/Output

	Name	Data type	Description
→@	st_id	int	Student Id
→@	st_name	varchar(30)	Student Name
→@	st_Email	varchar(40)	Student Email
→@	Dno	int	Deparment Number
→@	Pass	varchar(50)	Password

Uses

	Name
	insertStudent
	Student

Script

```
CREATE proc insertStudent @st_id int,@st_name varchar(30),@st_Email varchar(40),@Dno int,@Pass varchar(50)
as
begin try
		if not exists (select Sid from Student where Sid=@st_id )
begin
		insert into Student
				values (@st_id,@st_name,@st_Email,@Pass,@Dno)
		end
		else
				select 'Duplicate Student Id'
end try
begin catch
		Select 'error'
end catch
```

2.2.26. Procedure: insertTopic

Input/Output

	Name	Data type	Description
→@	TopicId	int	Topic Id
→@	TopicName	varchar(50)	Topic Name
→@	courseId	int	Course ID

Uses

	Name
	insertTopic
	Topic

Script

```
CREATE Proc insertTopic @TopicId int,@TopicName varchar(50),@courseId int
as
begin try
if not exists (select Topic_id from Topic where Topic_id=@TopicId )
            begin
            insert into Topic
            values (@TopicId,@TopicName,@courseId)
            end
            else
                        select 'Duplicate Topic Id'
end try
begin catch
            Select 'error'
end catch
```

2.2.27. Procedure: Report3

Input/Output

	Name	Data type	Description
→@	instructorId	int	Instructor ID

Uses

	Name
	Report3
	Course
	course_instructor
	student_course

Script

```
CREATE proc Report3  @instructorId int
as
            select c.Cname AS [course name] ,count(sc.Sid) [No of student]
            from Course c inner join course_instructor CI
            on c.Cid = CI.C_id
            inner join student_course sc
            on sc.Cid=c.Cid
            where CI.Ins_id= @instructorId
            group by c.Cname
```

2.2.28. Procedure: selectcourse

Input/Output

	Name	Data type	Description
→@	cid	int	Course ID

Uses

	Name
	selectcourse
	Course

2.2.29. Procedure: selectCoursesInstuctor

Input/Output

	Name	Data type	Description
→@	Ins_Id	int	Instructor ID

Uses

	Name
selectCoursesInstuctor	
Course	
course_instructor	

2.2.30. Procedure: selectDepartment

Input/Output

	Name	Data type	Description
→@	Dnum	int	Department Number

Uses

	Name
selectDepartment	
Department	

2.2.31. Procedure: selectInstructor

Input/Output

	Name	Data type	Description
→@	ins_id	int	Instructor Id

Uses

	Name
selectInstructor	
Instructor	

2.2.32. Procedure: selectQuestion

Input/Output

	Name	Data type	Description
→@	QNum	int	Question Number

Uses

	Name
selectQuestion	
Question	

Script

```
create Proc selectQuestion @QNum int
as
          select *
          from Question
          where Qnum = @QNum
```

2.2.33. Procedure: selectQuestionChoice

Input/Output

	Name	Data type	Description
→@	QuesNum	int	Question Number
→@	choiceNum	int	Choice Number

Uses

	Name
selectQuestionChoice	
Question_choice	

Script

```
create Proc selectQuestionChoice @QuesNum int,@choiceNum int
as
          select *
          from Question_choice
          where Qnum = @QuesNum and ChoiceNum=@choiceNum
```

2.2.34. Procedure: selectTopic

Input/Output

	Name	Data type	Description
→@	TopicId	int	Topic ID

Uses

	Name
selectTopic	
Topic	

Script

```
create Proc selectTopic @TopicId int
as
          select *
          from Topic
          where Topic_id = @TopicId
```

2.2.35. Procedure: StDelete

Input/Output

	Name	Data type	Description
→@	st_id	int	Student ID

Uses

	Name
StDelete	
Student	

Script

```
create Proc StDelete @st_id int
as
begin try
if  exists (select Sid from Student where Sid=@st_id )
            begin
                        delete from Student
                        where Sid=@st_id
            end
            else
                        select 'Student Id is not found'
end try
begin catch
            Select 'error'
end catch
```

2.2.36. Procedure: student_ans

Input/Output

	Name	Data type	Description
@	exam_num	int	Exam Id
@	std_id	int	Student Id

Uses

	Name
student_ans	
Exam_std_ques	
Question	

Script

```
CREATE proc student_ans @exam_num int , @std_id int
as
select Question.Qhead as Question ,Exam_std_ques.std_Ans as answer from  Question
inner join Exam_std_ques on  Question.Qnum = Exam_std_ques.Qnum
 where Exam_std_ques.Ex_id = @exam_num and Exam_std_ques.Sid =@std_id
```

2.2.37. Procedure: Stupdate

Input/Output

	Name	Data type	Description
@	st_id	int	Student Id
@	st_name	varchar(30)	Student Name
@	st_Email	varchar(40)	Student Email
@	Pass	varchar(50)	Password
@	Dno	int	Department Number

Uses

	Name
	Stupdate
	Student

Script

```
create Proc Stupdate @st_id int,@st_name varchar(30),@st_Email varchar(40),@Pass varchar(50),@Dno int
as
begin try
if  exists (select Sid from Student where Sid=@st_id )
           begin
           update Student
            set Sname=@st_name,Email=@st_Email,password=@Pass,Dnum=@Dno
            where Sid=@st_id
           end
           else
                     select 'Student Id is not found'
end try
begin catch
           Select 'error'
end catch
```

2.2.38. Procedure: updatecourse

Input/Output

	Name	Data type	Description
@	cid	int	Course Id
@	cname	varchar(50)	Course Name

Uses

	Name
	updatecourse
	Course

2.2.39. Procedure: updateCoursesInstuctor

Input/Output

	Name	Data type	Description
@	Ins_Id	int	Instructor ID
@	CourseId	int	Old Course ID
@	newCourceID	int	New Course Id

Uses

	Name
	updateCoursesInstuctor
	Course
	course_instructor

2.2.40. Procedure: updateDepartment

Input/Output

	Name	Data type	Description
→@	DeptNum	int	Department Number
→@	DeptName	varchar(50)	Department Name

Uses

	Name
	updateDepartment
	Department

2.2.41. Procedure: updateInstructor

Input/Output

	Name	Data type	Description
→@	ins_id	int	Insructor ID
→@	ins_name	varchar(50)	Instructor Name
→@	D_num	int	Department Number

Uses

	Name
	updateInstructor
	Instructor

2.2.42. Procedure: updateQuestion

Input/Output

	Name	Data type	Description
→@	Qnum	int	Question Number
→@	QType	varchar(5)	Question Typr
→@	Qhead	varchar(300)	Question head
→@	Model_Ans	varchar(50)	Model Answer
→@	fullmark_grade	int	FullMark Grade
→@	Cid	int	Course ID

Uses

	Name
	updateQuestion
	Question

2.2.43. Procedure: updateQuestionChoice

Input/Output

	Name	Data type	Description
→@	QuesNum	int	Question Number

	Name	Data type	Description
→@	choiceNum	int	Choice Number
→@	chiceBody	varchar(300)	Choice Body

Uses

	Name
updateQuestionChoice	
Question_choice	

Script

```
create Proc updateQuestionChoice  @QuesNum int,@choiceNum int,@chiceBody varchar(300)
as
begin try
if  exists (select Qnum,ChoiceNum from Question_choice where Qnum=@QuesNum and ChoiceNum=@QuesNum)
            begin
            update Question_choice
             set ChoiceBody=@chiceBody
             where Qnum=@QuesNum and ChoiceNum=@QuesNum
            end
            else
                        select 'Question num or choice num not found'
end try
begin catch
            Select 'error'
end catch
```

2.2.44. Procedure: updateStdCrs

Input/Output

	Name	Data type	Description
→@	st_id	int	Student ID
→@	Crs_id	int	Course ID
→@	grade	int	Grade

Uses

	Name
updateStdCrs	
student_course	

Script

```
create Proc updateStdCrs @st_id int,@Crs_id int,@grade int
as
begin try
if  exists (select Sid,Cid from student_course where Sid=@st_id )
            begin
            update student_course
             set Sid=@st_id,Cid=@Crs_id,Crs_grade=@grade
             where Sid=@st_id and Cid=@Crs_id
            end
            else
                        select 'Student Id is not found'
end try
begin catch
            Select 'error'
end catch
```

2.2.45. Procedure: updateTopic

Input/Output

	Name	Data type	Description
→@	TopicId	int	Topic ID
→@	TopicName	varchar(50)	Topic Name
→@	courseId	int	Course Id

Uses

	Name
updateTopic	
Topic	

Script

```
create Proc updateTopic @TopicId int,@TopicName varchar(50),@courseId int
as
begin try
if  exists (select Topic_id from Topic where Topic_id=@TopicId )
           begin
           update Topic
            set Topic_name=@TopicName,Crs_id=@courseId
            where Topic_id=@TopicId
           end
           else
                     select 'Topic Id not found'
end try
begin catch
           Select 'error'
end catch
```