ExaminationDB@DESKTOP-1FDPDQQ

Data Dictionary

2023-01-18

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 relationship

 User-defined many to one relationship

 One to many relationship

 User-defined one to many relationship

 Many to many relationship

 User-defined many to many relationship

 One to one relationship

 User-defined one to one relationship

 Input

 Output

 Input/Output

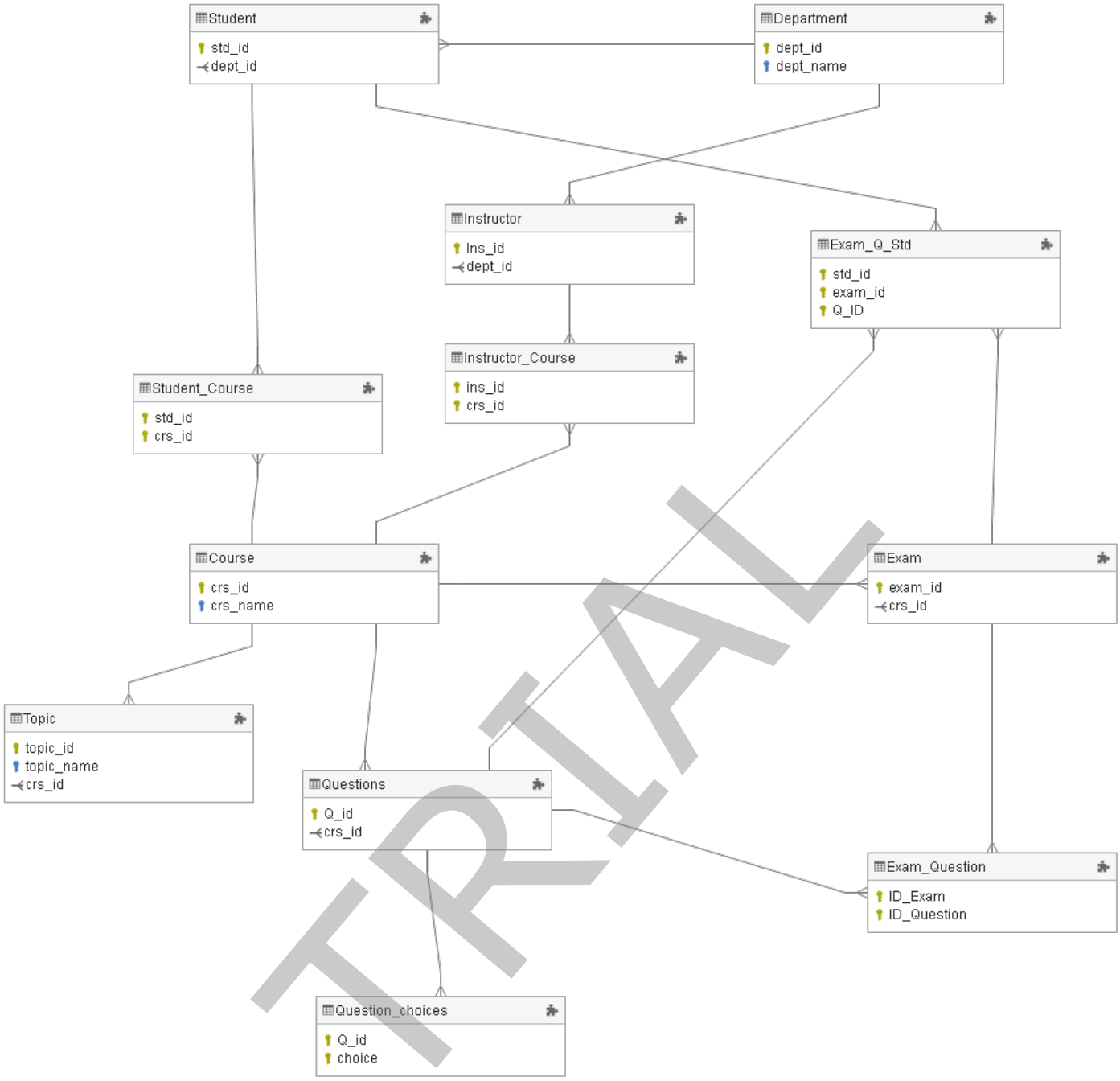
 Uses dependency

 User-defined uses dependency

 Used by dependency

 User-defined used by dependency

1. New subject area



1.1. Tables

1.1.1. Table: Course

Course table contains a number of courses that allows students to register for a course.

Columns

Name	Data type	Description / Attributes
crs_id	int	Primary key for courses records
crs_name	nvarchar(50)	Unique key for courses name **Nullable**
crs_duration	int	Course duration **Nullable**
course_grade	int	Course grade **Nullable** **Default**: 100

Linked from

Table	Join	Title / Name / Description
Exam	**Course**crs_id = Examcrs_id	FK_Exam_Course
Instructor_Course	**Course**crs_id = Instructor_Coursecrs_id	FK__Instructo__crs_i__5535A963
Questions	**Course**crs_id = Questionscrs_id	FK__Questions__crs_i__5812160E
Student_Course	**Course**crs_id = Student_Coursecrs_id	FK__Student_C__crs_i__5BE2A6F2
Topic	**Course**crs_id = Topiccrs_id	FK__Topic__crs_id__5BE2A6F2

Unique keys

Columns	Name / Description
crs_id	PK__Course__ECAF5375A258C6E2
crs_name	UQ__Course__775BF427EC09378F

Used By

Name
Course
Exam
Instructor_Course
Questions
Student_Course
Topic

1.1.2. Table: Department

Department houses instructors and the enrolled students.

Columns

Name	Data type	Description / Attributes
dept_id	int	Primary key which classifies each department with its number
dept_name	nvarchar(50)	
dept_location	nvarchar(50)	**Nullable**

Linked from

Table	Join	Title / Name / Description
Instructor	**Department**dept_id = Instructordept_id	FK_dept_id
Student	**Department**dept_id = Studentdept_id	FK__Student__dept_id__59063A47

Unique keys

Columns	Name / Description
dept_id	PK__Departme__DCA659742684E0B5
dept_name	UQ__Departme__C7D39AE10105417A

Used By

Name
Department
Instructor
Student

1.1.3. Table: Exam

Columns

Name	Data type	Description / Attributes
exam_id	int	**Identity / Auto increment**
exam_duration	int	**Nullable**
crs_id	int	**Nullable** **References**: Course

Links to

Table	Join	Title / Name / Description
Course	**Exam**crs_id = Coursecrs_id	FK_Exam_Course

Linked from

Table	Join	Title / Name / Description
Exam_Q_Std	**Exam**exam_id = Exam_Q_Stdexam_id	FK__Exam_Q_St__exam___5165187F
Exam_Question	**Exam**exam_id = Exam_QuestionID_Exam	FK_Exam_Question_Exam

Unique keys

Columns	Name / Description
exam_id	PK__Exam__9C8C7BE9BADCFE69

Uses

Name
Exam
Course

Used By

Name
Exam
Exam_Q_Std
Exam_Question

1.1.4. Table: Exam_Q_Std

Columns

Name	Data type	Description / Attributes
std_id	int	**References**: Student
exam_id	int	**References**: Exam
Q_ID	int	**References**: Questions
std_answer	nvarchar(10)	**Nullable**

Links to

Table	Join	Title / Name / Description
Exam	**Exam_Q_Std**exam_id = Examexam_id	FK__Exam_Q_St__exam___5165187F
Questions	**Exam_Q_Std**Q_ID = QuestionsQ_id	FK__Exam_Q_Std__Q_ID__534D60F1
Student	**Exam_Q_Std**std_id = Studentstd_id	FK__Exam_Q_St__std_i__52593CB8

Unique keys

Columns	Name / Description
std_id, exam_id, Q_ID	PK__Exam_Q_S__AD3E7E33BE8F1301

Uses

Name
Exam_Q_Std
Exam
Questions
Student

1.1.5. Table: Exam_Question

Columns

Name	Data type	Description / Attributes
ID_Exam	int	**References**: Exam
ID_Question	int	**References**: Questions

Links to

Table	Join	Title / Name / Description
Exam	**Exam_Question**ID_Exam = Examexam_id	FK_Exam_Question_Exam
Questions	**Exam_Question**ID_Question = QuestionsQ_id	FK_Exam_Question_Questions

Unique keys

Columns	Name / Description
ID_Exam, ID_Question	PK_Exam_Question

Uses

Name
Exam_Question
Exam
Questions

1.1.6. Table: Instructor

Columns

Name	Data type	Description / Attributes
Ins_id	int	
Ins_name	nvarchar(50)	**Nullable**
salary	int	**Nullable**
dept_id	int	**Nullable** **References**: Department

Links to

Table	Join	Title / Name / Description
Department	**Instructor**dept_id = Departmentdept_id	FK_dept_id

Linked from

Table	Join	Title / Name / Description
Instructor_Course	**Instructor**Ins_id = Instructor_Courseins_id	FK__Instructo__ins_i__5629CD9C

Unique keys

Columns	Name / Description
Ins_id	PK__Instruct__1520C1E5C37A90A8

Uses

Name
Instructor
Department

Used By

Name
Instructor
Instructor_Course

1.1.7. Table: Instructor_Course

Columns

Name	Data type	Description / Attributes
ins_id	int	**References**: Instructor
crs_id	int	**References**: Course

Links to

Table	Join	Title / Name / Description
Course	**Instructor_Course**crs_id = Coursecrs_id	FK__Instructo__crs_i__5535A963
Instructor	**Instructor_Course**ins_id = InstructorIns_id	FK__Instructo__ins_i__5629CD9C

Unique keys

Columns	Name / Description
ins_id, crs_id	PK__Instruct__D27DD8173EC7BA6B

Uses

Name
Instructor_Course
Course
Instructor

1.1.8. Table: Question_choices

Columns

Name	Data type	Description / Attributes
Q_id	int	**References**: Questions
choice_id	int	**Identity / Auto increment**
choice	nvarchar(200)	

Links to

Table	Join	Title / Name / Description
Questions	**Question_choices**Q_id = QuestionsQ_id	FK__Question_c__Q_id__571DF1D5

Unique keys

Columns	Name / Description
Q_id, choice	PK__Question__E38E2236C061134F

Uses

Name
Question_choices
Questions

1.1.9. Table: Questions

Columns

Name	Data type	Description / Attributes
Q_id	int	**Identity / Auto increment**
Question	nvarchar(500)	**Nullable**
model_answer	nvarchar(10)	**Nullable**
Q_type	int	
Q_grade	int	**Nullable**
crs_id	int	**Nullable** **References**: Course

Links to

Table	Join	Title / Name / Description
Course	**Questions**crs_id = Coursecrs_id	FK__Questions__crs_i__5812160E

Linked from

Table	Join	Title / Name / Description
Exam_Q_Std	**Questions**Q_id = Exam_Q_StdQ_ID	FK__Exam_Q_Std__Q_ID__534D60F1
Exam_Question	**Questions**Q_id = Exam_QuestionID_Question	FK_Exam_Question_Questions
Question_choices	**Questions**Q_id = Question_choicesQ_id	FK__Question_c__Q_id__571DF1D5

Unique keys

Columns	Name / Description
Q_id	PK__Question__F4FD2B661ACD5C0A

Uses

Name
Questions
Course

Used By

Name
Questions
Exam_Q_Std
Exam_Question
Question_choices

1.1.10. Table: Student

Columns

		Name	Data type	Description / Attributes
		std_id	int	
		std_fname	nvarchar(50)	**Nullable**
		std_lname	nvarchar(50)	**Nullable**
		std_age	int	**Nullable**
		std_phone	nvarchar(50)	**Nullable**
		dept_id	int	**Nullable** **References**: Department

Links to

	Table	Join	Title / Name / Description
	Department	**Student**dept_id = Departmentdept_id	FK__Student__dept_id__59063A47

Linked from

	Table	Join	Title / Name / Description
	Exam_Q_Std	**Student**std_id = Exam_Q_Stdstd_id	FK__Exam_Q_St__std_i__52593CB8
	Student_Course	**Student**std_id = Student_Coursestd_id	FK__Student_C__std_i__5AEE82B9

Unique keys

	Columns	Name / Description
	std_id	PK__Student__0B0245BAE6446AEC

Uses

Name
Student
Department

Used By

Name
Student
Exam_Q_Std
Student_Course

1.1.11. Table: Student_Course

Columns

Name	Data type	Description / Attributes
std_id	int	**References**: Student
crs_id	int	**References**: Course
grade_overall	int	**Nullable**

Links to

Table	Join	Title / Name / Description
Course	**Student_Course**crs_id = Coursecrs_id	FK__Student_C__crs_i__5BE2A6F2
Student	**Student_Course**std_id = Studentstd_id	FK__Student_C__std_i__5AEE82B9

Unique keys

Columns	Name / Description
std_id, crs_id	PK__Student___45C8B08D5D2D4D25

Uses

Name
Student_Course
Course
Student

1.1.12. Table: Topic

Columns

Name		Data type	Description / Attributes
topic_id		int	
topic_name		nvarchar(50)	**Nullable**
crs_id		int	**Nullable** **References**: Course

Links to

Table	Join	Title / Name / Description
Course	**Topic**crs_id = Coursecrs_id	FK__Topic__crs_id__5BE2A6F2

Unique keys

Columns	Name / Description
topic_id	PK__Topic__D5DAA3E974BC54EF
topic_name	UQ__Topic__54BAE5EC835B11D7

Uses

Name
Topic
Course

2. Other

2.1. Procedures

2.1.1. Procedure: assign_answers

Input/Output

	Name	Data type	Description
→@	std_id	int	
→@	exam_id	int	
→@	q1	nvarchar(10)	
→@	q2	nvarchar(10)	
→@	q3	nvarchar(10)	
→@	q4	nvarchar(10)	
→@	q5	nvarchar(10)	
→@	q6	nvarchar(10)	
→@	q7	nvarchar(10)	
→@	q8	nvarchar(10)	
→@	q9	nvarchar(10)	
→@	q10	nvarchar(10)	

Script

```
create proc [dbo].[assign_answers] @std_id int,@exam_id int,@q1 nvarchar(10),@q2 nvarchar(10),@q3 nvarchar(10),@q4
nvarchar(10),@q5 nvarchar(10),
                                                                    @q6 nvarchar(10),@q7 nvarchar(10),@q8
nvarchar(10),@q9 nvarchar(10),@q10 nvarchar(10)
As
begin
            declare @t table(id int,std_ans nvarchar(10))
            INSERT INTO @t
            VALUES
            (1,@q1),(2,@q2),(3,@q3),(4,@q4),(5,@q5),(6,@q6),(7,@q7),(8,@q8),(9,@q9),(10,@q10)

            insert into Exam_Q_Std
            select @std_id,@exam_id,E.ID_Question,null
            from Exam_Question E
            where E.ID_Exam = @exam_id

            declare Mc1 cursor
            for select E.std_answer from Exam_Q_Std E
            where E.std_id=@std_id and E.exam_id=@exam_id
            for update

            declare @ans int
            declare @increment int = 1
            open Mc1
            declare @temp nvarchar(10)
            fetch Mc1 into @ans
            while @@FETCH_STATUS=0
                        begin
                                    select @temp=std_ans
                                    from @t T
                                    where T.id=@increment

                                    update Exam_Q_Std
                                    set std_answer=@temp
                                    where current of Mc1

                                    set @increment +=1

                                    fetch Mc1 into @ans
                        end
            close Mc1
            deallocate Mc1
end
```

2.1.2. Procedure: Correction_Exam

Input/Output

	Name	Data type	Description
→@	std_id	int	
→@	ExamId	int	

Script

```
CREATE proc [dbo].[Correction_Exam]  @std_id int,@ExamId int
as
begin
           declare @crsId int,@degree int
           select @crsId=E.crs_id
           from Exam E
           where E.exam_id=@ExamId

           select @degree=count(*)
           from Exam_Q_Std E inner join Questions Q
           on E.std_answer=Q.model_answer and Q.Q_id = E.Q_ID

           --set @degree *=10

           insert into Student_Course
           values(@std_id,@crsId,@degree*10)

end
```

2.1.3. Procedure: Course_Delete_SP

Input/Output

	Name	Data type	Description
→@	crs_id	int	

Script

```
create proc [dbo].[Course_Delete_SP] @crs_id int
    AS
           if @crs_id in (select crs_id from Course)
           begin
    DELETE FROM Course WHERE crs_id = @crs_id
           end
           else
           select'this id is not exist'
```

2.1.4. Procedure: Course_Insert_SP

Input/Output

	Name	Data type	Description
→@	crs_id	int	
→@	crs_name	varchar(20)	
→@	crs_duration	int	
→@	course_grade	int	

Script

```
create proc [dbo].[Course_Insert_SP] @crs_id int , @crs_name varchar(20), @crs_duration int , @course_grade int
as
          if @crs_id in (select crs_id from Course) or  @crs_name in (select crs_name from Course)
                    BEGIN
                    print 'this course id or name is exists already'
                    END
                    else
                    insert into Course values ( @crs_id , @crs_name , @crs_duration  , @course_grade )
```

2.1.5. Procedure: Course_Select_SP

Input/Output

	Name	Data type	Description
→@	crs_id	int	

Script

```
create proc [dbo].[Course_Select_SP] @crs_id int=null
    as
           if @crs_id in (select crs_id from Course)
           begin
           select * from Course
           where crs_id =@crs_id
           end
           else if @crs_id  is null
            begin
           select * from Course
           end
           else
           select ' this id is not exist'
```

2.1.6. Procedure: Course_Update_SP

Input/Output

	Name	Data type	Description
→@	crs_id	int	
→@	crs_name	varchar(20)	
→@	crs_duration	int	
→@	course_grade	int	

Script

```
create proc [dbo].[Course_Update_SP] @crs_id int , @crs_name varchar(20), @crs_duration int , @course_grade int
     as

   if @crs_id not in (select crs_id  from Course)
   begin
   select 'this id is not exist'
   end
   else
   begin
   update Course
   set  crs_name=@crs_name,crs_duration=@crs_duration,course_grade=@course_grade
   where crs_id=@crs_id and crs_name !=@crs_name
   end
```

2.1.7. Procedure: Depart_delete_SP

Input/Output

	Name	Data type	Description
→@	dept_id	int	

Script

```
create proc [dbo].[Depart_delete_SP] @dept_id int
  as
           if @dept_id in (select dept_id  from Department)
           begin
    DELETE FROM Department WHERE dept_id = @dept_id
           end
           else
           select'this id is not exist'
```

2.1.8. Procedure: Depart_Insert_SP

Input/Output

	Name	Data type	Description
→@	dept_id	int	
→@	dept_name	varchar(50)	
→@	dept_location	varchar(50)	

Script

```
create proc [dbo].[Depart_Insert_SP] @dept_id int, @dept_name varchar(50) ,
 @dept_location varchar(50)
 as
  if(@dept_id is not null)
  begin
            insert into Department
            values(@dept_id ,@dept_name,@dept_location )
  end
  else
  begin
            select 'Not Exist or you did not entered the id'
  end
```

2.1.9. Procedure: Depart_select_SP

Input/Output

	Name	Data type	Description
→@	dept_id	int	

Script

```
create proc [dbo].[Depart_select_SP] @dept_id int=null
  as
  if exists (select dept_id from Department where dept_id = @dept_id)
  begin
           select * from Department
           where  dept_id = @dept_id
  end
  else if @dept_id is null
  select * from Department
  else
  select 'Id not Found'
```

2.1.10. Procedure: Depart_Update_SP

Input/Output

	Name	Data type	Description
→@	dept_id	int	
→@	dept_name	varchar(50)	
→@	dept_location	varchar(50)	

Script

```
create proc [dbo].[Depart_Update_SP] @dept_id int, @dept_name varchar(50) ,
 @dept_location varchar(50)
 as
 if @dept_id in (select dept_id from Department)
 update Department
 set dept_name=@dept_name , dept_location=@dept_location
 where dept_id=@dept_id
 else
 select 'id not found'
```

2.1.11. Procedure: Exam_Delete_SP

Input/Output

	Name	Data type	Description
→@	Exam_id	int	

Script

```
create Proc [dbo].[Exam_Delete_SP] @Exam_id int
AS
          DELETE FROM Exam WHERE exam_id = @Exam_id
```

2.1.12. Procedure: Exam_Insert_SP

Input/Output

	Name	Data type	Description
→@	duration	int	
→@	crsID	int	

Script

```
CREATE procedure [dbo].[Exam_Insert_SP]  @duration int,@crsID int
as
          insert into Exam(exam_duration,crs_id)
          values(@duration,@crsID)
```

2.1.13. Procedure: Exam_Q_Std_Delete_SP

Input/Output

	Name	Data type	Description
→@	std_id	int	
→@	Exam_id	int	
→@	Question_id	int	

Script

```
create Proc [dbo].[Exam_Q_Std_Delete_SP] @std_id int, @Exam_id int , @Question_id int
AS
          DELETE FROM Exam_Q_Std
          WHERE std_id = @std_id and exam_id = @Exam_id and Q_ID = @Question_id
```

2.1.14. Procedure: Exam_Q_Std_Insert_SP

Input/Output

	Name	Data type	Description
→@	std_id	int	
→@	Exam_id	int	
→@	Question_id	int	
→@	answer	nvarchar(200)	

Script

```
create procedure [dbo].[Exam_Q_Std_Insert_SP]  @std_id int, @Exam_id int , @Question_id int ,@answer nvarchar(200)=null
as
            if exists(select S.std_id from Student S where S.std_id=@std_id) and
                        exists(select E.exam_id from Exam E where E.exam_id=@Exam_id) and
                         exists(select Q_id from Questions Q where Q_id=@Question_id)
            begin
                        if not exists(select E.std_id,E.exam_id,E.Q_ID from Exam_Q_Std E
                                                                  where E.std_id=@std_id and E.exam_id=@Exam_id and
E.Q_ID=@Question_id)
                        begin
                                                insert into Exam_Q_Std
                                                values(@std_id,@Exam_id,@Question_id,@answer)
                        end
                        else
                                    select 'This answer already exsist'
            end
            else
                        select 'Error'
```

2.1.15. Procedure: Exam_Q_Std_Select_SP

Input/Output

	Name	Data type	Description
→@	std_id	int	
→@	Exam_id	int	
→@	Question_id	int	

Script

```
CREATE proc [dbo].[Exam_Q_Std_Select_SP] @std_id int, @Exam_id int , @Question_id int
as
          if exists(select E.std_id,E.exam_id,E.Q_ID from Exam_Q_Std E
                                                              where E.std_id=@std_id and E.exam_id=@Exam_id and
E.Q_ID=@Question_id)
                    begin
                              select E.std_id,E.exam_id,E.Q_ID,E.std_answer
                              from Exam_Q_Std E
                              where E.std_id=@std_id and E.exam_id=@Exam_id and E.Q_ID=@Question_id
                    end
          else
                    select 'Not Valid Data'
```

2.1.16. Procedure: Exam_Q_Std_update_sp

Input/Output

	Name	Data type	Description
→@	std_id	int	
→@	Exam_id	int	
→@	Question_id	int	
→@	answer	nvarchar(200)	

Script

```
create proc [dbo].[Exam_Q_Std_update_sp] @std_id int, @Exam_id int , @Question_id int ,@answer nvarchar(200)=null
as
            declare @fi_answer nvarchar(200) = (select E.std_answer from Exam_Q_Std E
            where E.std_id=@std_id and E.exam_id=@Exam_id and E.Q_ID=@Question_id)

           if exists(select E.std_id,E.exam_id,E.Q_ID from Exam_Q_Std E
                                                                  where E.std_id=@std_id and E.exam_id=@Exam_id and
E.Q_ID=@Question_id)
                        begin
                                    set @fi_answer = (coalesce(@answer,@fi_answer))
                                    update Exam_Q_Std
                                    set std_answer = @fi_answer where std_id=@std_id and exam_id=@Exam_id and Q_ID=@Question_id
                        end
           else
                        select 'Not Found To Update'
```

2.1.17. Procedure: Exam_Question_delete_SP

Input/Output

	Name	Data type	Description
→@	Exam_id	int	
→@	Question	int	

Script

```
create proc [dbo].[Exam_Question_delete_SP] @Exam_id int , @Question int
as
if exists (select E.ID_Exam,E.ID_Question from Exam_Question E where E.ID_Exam=@Exam_id and E.ID_Question=@Question)
begin
delete from Exam_Question
end
else
select 'this id is not found'
```

2.1.18. Procedure: Exam_Question_Insert_SP

Input/Output

	Name	Data type	Description
→@	Exam_id	int	
→@	Question	int	

Script

```
CREATE proc [dbo].[Exam_Question_Insert_SP] @Exam_id int , @Question int
as
if exists ( select E.exam_id  from Exam E where E.exam_id=@Exam_id)
and exists (select Q.Q_id  from Questions Q where Q.Q_id=@Question)
begin
insert into Exam_Question values ( @Exam_id,@Question)
end
else
select 'Error in Input Data'
```

2.1.19. Procedure: Exam_Question_select_SP

Input/Output

	Name	Data type	Description
→@	Exam_id	int	

Script

```
create proc [dbo].[Exam_Question_select_SP] @Exam_id int = null
as
if exists (select E.ID_Exam from Exam_Question E where E.ID_Exam=@Exam_id)
select * from Exam_Question E where E.ID_Exam = @Exam_id

else if @Exam_id is null
select * from Exam_Question
```

2.1.20. Procedure: Exam_Select_SP

Input/Output

	Name	Data type	Description
→@	Exam_id	int	

Script

```
CREATE proc [dbo].[Exam_Select_SP] @Exam_id int=0
    as
          if @Exam_id <> 0
                    begin
                              if exists(select E.exam_id from Exam E where exam_id=@Exam_id)
                              begin
                                        select *
                                        from Exam E
                                        where exam_id=@Exam_id
                              end
                              else
                                        select 'Not Exist'
                    end
          else
          begin
                    select * from Exam
          end
```

2.1.21. Procedure: Exam_update_sp

Input/Output

	Name	Data type	Description
→@	Exam_id	int	
→@	duration	int	
→@	crs_id	int	

Script

```
CREATE proc [dbo].[Exam_update_sp] @Exam_id int, @duration int=null,@crs_id int = null
as
begin
            declare @fi_duration int = (select exam_duration from Exam where exam_id=@Exam_id)
            declare @fi_crs_id int = (select crs_id from Exam where exam_id=@Exam_id)

            if exists(select E.exam_id from Exam E where exam_id=@Exam_id)
            begin
                        set @fi_duration = (coalesce(@duration,@fi_duration))
                        set @fi_crs_id = (coalesce(@crs_id,@fi_crs_id))

                        update Exam
                        set exam_duration = @fi_duration,
                        crs_id = @fi_crs_id
                        where exam_id = @Exam_id
            end
            else
            print 'Exam does not exist'
end
```

2.1.22. Procedure: Generate_Exam

Input/Output

	Name	Data type	Description
→@	duration	int	
→@	crs_id	int	
→@	numOf_trueFalse	int	
→@	numOf_multiChoice	int	

Script

```
CREATE proc [dbo].[Generate_Exam]  @duration int=null,@crs_id int,@numOf_trueFalse int,@numOf_multiChoice int
    as
begin
           execute Exam_Insert_SP @duration,@crs_id
           declare @newExamId int = @@identity


           INSERT INTO Exam_Question
           select top(@numOf_trueFalse) @newExamId,Q.Q_id
           from Questions Q
           where Q.Q_type=2 and Q.crs_id=@crs_id
           order by NEWID()


           INSERT INTO Exam_Question
           select top(@numOf_multiChoice) @newExamId,Q.Q_id
           from Questions Q
           where Q.Q_type=1 and Q.crs_id=@crs_id
           order by NEWID()

end
```

2.1.23. Procedure: Instructor_Course_delete_SP

Input/Output

	Name	Data type	Description
→@	ins_id	int	
→@	crs_id	int	

Script

```
--delete
create proc [dbo].[Instructor_Course_delete_SP] @ins_id int,@crs_id int
as
if exists (select I.ins_id,I.crs_id from Instructor_Course I where ins_id=@ins_id and I.crs_id=@crs_id)
begin
delete from Instructor_Course
where ins_id=@ins_id
end
else
select 'this id is not found'
```

2.1.24. Procedure: Instructor_Course_Insert_SP

Input/Output

	Name	Data type	Description
→@	ins_id	int	
→@	crs_id	int	

Script

```
create proc [dbo].[Instructor_Course_Insert_SP] @ins_id int , @crs_id int
as
if exists ( select ins_id  from Instructor where ins_id=@ins_id)
and exists (select crs_id  from Course where crs_id=@crs_id)
begin
insert into Instructor_Course values ( @ins_id,@crs_id)
end
else
select 'the instructor id or the course id is not found'
```

2.1.25. Procedure: Instructor_Course_select_SP

Input/Output

	Name	Data type	Description
→@	ins_id	int	

Script

```
create proc [dbo].[Instructor_Course_select_SP] @ins_id int = null
as
if exists (select ins_id from Instructor_Course where ins_id=@ins_id)
select * from Instructor_Course where ins_id=@ins_id

else if @ins_id is null
select * from Instructor_Course

else
select'this is is not found'
```

2.1.26. Procedure: Instructor_Course_update_SP

Input/Output

	Name	Data type	Description
→@	ins_id	int	
→@	crs_id	int	

Script

```
create proc [dbo].[Instructor_Course_update_SP] @ins_id int , @crs_id int=null
as
declare @old_crs_id int = (select crs_id from Instructor_Course where ins_id=@ins_id )
if exists ( select ins_id from Instructor_Course where ins_id=@ins_id)
and @crs_id not in (select crs_id from Instructor_Course )
begin
update Instructor_Course
set crs_id= ISNULL(@crs_id,@old_crs_id )
end
else
select 'this id is not found'
```

2.1.27. Procedure: Instructor_Delete_SP

Input/Output

	Name	Data type	Description
→@	Ins_id	int	

Script

```
create proc [dbo].[Instructor_Delete_SP] @Ins_id int=null
as
if exists (select Ins_id from Instructor where Ins_id=@Ins_id)
delete  from Instructor where Ins_id=@Ins_id

else if @Ins_id is null
select 'please insert an ID'

else
select'this ID is not exist'
```

2.1.28. Procedure: Instructor_Insert_SP

Input/Output

	Name	Data type	Description
→@	Ins_id	int	
→@	Ins_name	varchar(20)	
→@	salary	int	
→@	dept_id	int	

Script

```
create proc [dbo].[Instructor_Insert_SP] @Ins_id int , @Ins_name varchar(20)  ,
@salary int ,@dept_id int
as
if exists (select dept_id from Department where dept_id=@dept_id)
begin
if exists (select Ins_id from Instructor where Ins_id=@Ins_id)
or exists (select Ins_name from Instructor where Ins_name=@Ins_name)
begin
select 'this id or name is already exist'
end
else
begin
insert into Instructor values ( @Ins_id,@Ins_name,@salary,@dept_id)
end
end
else
select 'this department id is not exist'
```

2.1.29. Procedure: Instructor_Select_SP

Input/Output

	Name	Data type	Description
→@	Ins_id	int	

Script

```
create proc [dbo].[Instructor_Select_SP] @Ins_id int=null
as
if exists(select Ins_id from Instructor where Ins_id=@Ins_id)
begin
select * from Instructor where Ins_id=@Ins_id
end
else if @Ins_id is null
select* from Instructor
else
select 'this id is not exits'
```

2.1.30. Procedure: Instructor_Update_SP

Input/Output

	Name	Data type	Description
→@	Ins_id	int	
→@	Ins_name	varchar(20)	
→@	salary	int	
→@	dept_id	int	

Script

```
create proc [dbo].[Instructor_Update_SP] @Ins_id int , @Ins_name varchar(20)  ,
@salary int ,@dept_id int
as
declare
@y varchar(20)=(select Ins_name from Instructor where Ins_id=@Ins_id) ,
@z int=(select salary from Instructor where Ins_id=@Ins_id) ,
@a int =(select dept_id from Instructor where Ins_id=@Ins_id)

if exists (select Ins_id from Instructor where Ins_id=@Ins_id)
begin
if exists (select D.dept_id from Department D where D.dept_id=@dept_id)
begin
update Instructor
set Ins_name= ISNULL(@Ins_name, @y) ,
salary=ISNULL (@salary,@z) ,
dept_id=ISNULL (@dept_id,@a)
where Ins_id=@Ins_id
end
else
          select 'Incorrect Department'
end
else
select 'this id does not exist'
```

2.1.31. Procedure: Question_Chocies_Delete_SP

Input/Output

	Name	Data type	Description
→@	Q_id	int	
→@	choice_id	int	

Script

```
create Proc [dbo].[Question_Chocies_Delete_SP]  @Q_id int,@choice_id int
AS
          DELETE FROM Question_choices WHERE Q_id = @Q_id and choice_id=@choice_id
```

2.1.32. Procedure: Question_Choices_Insert_SP

Input/Output

	Name	Data type	Description
→@	Q_id	int	
→@	choice	nvarchar(200)	

Script

```
create procedure [dbo].[Question_Choices_Insert_SP]  @Q_id int ,@choice nvarchar(200)
as
           insert into Question_choices(Q_id,choice)
           values(@Q_id,@choice)
```

2.1.33. Procedure: Question_Choices_Select_SP

Input/Output

	Name	Data type	Description
→@	Q_id	int	

Script

```
create proc [dbo].[Question_Choices_Select_SP]  @Q_id int=0
    as
          if @Q_id <> 0
                    begin
                              if exists(select Q.Q_id from Question_choices Q where Q_id=@Q_id)
                              begin
                                        select *
                                        from Question_choices Q
                                        where Q_id=@Q_id
                              end
                              else
                                        select 'Not Exist'
                    end
          else
          begin
                    select * from Question_choices
          end
```

2.1.34. Procedure: Question_Choices_Update_SP

Input/Output

	Name	Data type	Description
→@	Q_id	int	
→@	choice_id	int	
→@	choice	nvarchar(200)	

Script

```
create proc [dbo].[Question_Choices_Update_SP] @Q_id int, @choice_id int ,@choice nvarchar(200)=null
as
begin
            declare @fi_Choise nvarchar(200) = (select choice from Question_choices where Q_id=@Q_id and
choice_id=@choice_id)

            if exists(select Q.Q_id from Question_choices Q where Q_id=@Q_id and choice_id=@choice_id)
            begin
                        set @fi_Choise = (coalesce(@choice,@fi_Choise))

                        update Question_choices
                        set choice = @fi_Choise

                        where Q_id = @Q_id and choice_id=@choice_id
            end
            else
            print 'Question Choise does not exist'
end
```

2.1.35. Procedure: Question_Delete_SP

Input/Output

	Name	Data type	Description
→@	Q_id	int	

Script

```
create Proc [dbo].[Question_Delete_SP]  @Q_id int
AS
          DELETE FROM Questions WHERE Q_id = @Q_id
```

2.1.36. Procedure: Question_Insert_SP

Input/Output

	Name	Data type	Description
→@	Question	nvarchar(50)	
→@	model_answer	nvarchar(200)	
→@	Q_type	int	
→@	Q_grade	int	
→@	crs_id	int	

Script

```
CREATE procedure [dbo].[Question_Insert_SP]  @Question nvarchar(50),@model_answer nvarchar(200),
                                    @Q_type int,@Q_grade int ,@crs_id int
as
          insert into Questions(Question,model_answer,Q_type,Q_grade,crs_id)
          values(@Question,@model_answer,@Q_type,@Q_grade,@crs_id)
```

2.1.37. Procedure: Question_Select_SP

Input/Output

	Name	Data type	Description
→@	Q_id	int	

Script

```
CREATE proc [dbo].[Question_Select_SP]  @Q_id int=0
    as
          if @Q_id <> 0
                    begin
                              if exists(select Q.Q_id from Questions Q where Q_id=@Q_id)
                              begin
                                        select *
                                        from Questions Q
                                        where Q_id=@Q_id
                              end
                              else
                                        select 'Not Exist'
                    end
          else
          begin
                    select * from Questions
          end
```

2.1.38. Procedure: Question_Update_SP

Input/Output

	Name	Data type	Description
→@	Q_id	int	
→@	Question	nvarchar(50)	
→@	model_answer	nvarchar(200)	
→@	Q_type	int	
→@	Q_grade	int	
→@	crs_id	int	

Script

```
CREATE proc [dbo].[Question_Update_SP] @Q_id int, @Question nvarchar(50)=null,@model_answer nvarchar(200)=null,
                                @Q_type int=null,@Q_grade int=null ,@crs_id int=null
as
begin
            declare @fi_Question nvarchar(50) = (select Question from Questions where Q_id=@Q_id)
            declare @fi_model_answer nvarchar(200) = (select model_answer from Questions where Q_id=@Q_id)
            declare @fi_Q_type int = (select Q_type from Questions where Q_id=@Q_id)
            declare @fi_Q_grade int = (select Q_grade from Questions where Q_id=@Q_id)
            declare @fi_crs_id int = (select crs_id from Questions where Q_id=@Q_id)
            if exists(select Q.Q_id from Questions Q where Q_id=@Q_id)
            begin
                        set @fi_Question = (coalesce(@Question,@fi_Question))
                        set @fi_model_answer = (coalesce(@model_answer,@fi_model_answer))
                        set @fi_Q_type = (coalesce(@Q_type,@fi_Q_type))
                        set @fi_Q_grade = (coalesce(@Q_grade,@fi_Q_grade))
                        set @fi_crs_id = (coalesce(@crs_id,@fi_crs_id))
                        update Questions
                        set Question = @fi_Question
                        , model_answer = @fi_model_answer
                        , Q_type = @fi_Q_type
                        , Q_grade = @fi_Q_grade
                        , crs_id = @fi_crs_id
                        where Q_id = @Q_id
            end
            else
            print 'Question does not exist'
end
```

2.1.39. Procedure: report_1_sp

Input/Output

	Name	Data type	Description
→@	dept_id	int	

Script

```
create proc [dbo].[report_1_sp] @dept_id int
as
begin
if @dept_id in (select dept_id from Department)
   begin
                        select s.* ,d.dept_name
                        from student s, department d
                        where s.dept_id=d.dept_id and s.dept_id=@dept_id
                        order by dept_name
           end
           else
           print 'you enter wrong data'
end
```

2.1.40. Procedure: report_2_sp

Input/Output

	Name	Data type	Description
→@	std_id	int	

Script

```
CREATE proc [dbo].[report_2_sp] @std_id int
as
begin
if @std_id in (select std_id from Student)
        begin
                                    select s.std_id ,s.std_fname+' '+s.std_lname 'full_name' ,c.crs_name ,sc.grade_overall
                                    from Student s , student_course sc,Course c
                                    where s.std_id =sc.std_id and sc.crs_id=c.crs_id  and s.std_id=@std_id
                        end
else
print'you enter wrong data'
end
```

2.1.41. Procedure: report_3_sp

Input/Output

	Name	Data type	Description
→@	ins_id	int	

Script

```
create proc [dbo].[report_3_sp] @ins_id int
as
begin
if @ins_id in (select Ins_id from Instructor)
        begin

                                                select i.Ins_id, i.Ins_name ,c.crs_name ,count(sc.crs_id) std_#
                                                from Instructor i , Instructor_Course ic ,Course c ,student_course sc
                                                where i.Ins_id=ic.ins_id and ic.crs_id =c.crs_id and i.Ins_id=@ins_id and
ic.crs_id=sc.crs_id
                                                group by i.Ins_id, i.Ins_name ,c.crs_name
                        end
else
print'you enter wrong data'
end
```

2.1.42. Procedure: report_4_sp

Input/Output

	Name	Data type	Description
→@	crs_id	int	

Script

```
create proc [dbo].[report_4_sp] @crs_id int
as
begin
if @crs_id in (select crs_id from Course)
begin
                    select c.crs_id,c.crs_name,t.topic_name
                    from Course c , Topic t
                    where c.crs_id = t.crs_id and c.crs_id=@crs_id
end
else
print 'you enter wrong data'
end
```

2.1.43. Procedure: report_5_sp

Input/Output

	Name	Data type	Description
→@	exam_id	int	

Script

```
create proc [dbo].[report_5_sp] @exam_id int
as
begin
if @exam_id in (select exam_id from exam_Q_std) and  @exam_id in (select exam_id from Exam)
begin
                      select e.exam_id ,e.Q_ID ,q.Question ,QE.choice
                      from exam_Q_std e
                      inner join Questions q
                      on e.Q_ID =q.Q_id and e.exam_id=@exam_id
                      inner join Question_choices QE
                      on q.Q_id=QE.Q_id
end
else
print 'I will show the exam after student answer it for security reasons'
end
```

2.1.44. Procedure: Student_course_Delete_SP

Input/Output

	Name	Data type	Description
→@	Student_id	int	

Script

```
create PROC [dbo].[Student_course_Delete_SP] @Student_id int
    AS
           DELETE FROM Student_Course WHERE std_id = @Student_id
```

2.1.45. Procedure: Student_course_Insert_SP

Input/Output

	Name	Data type	Description
→@	Student_id	int	
→@	course_id	int	
→@	grade	int	

Script

```
create proc [dbo].[Student_course_Insert_SP] @Student_id int,@course_id int,@grade int

as

           if  exists (select s.std_id from Student s where s.std_id=@student_id)
                      and exists(select c.crs_id from Course c where c.crs_id=@course_id)
                      BEGIN
                      insert into Student_Course(std_id,crs_id,grade_overall)
                      values(@Student_id,@course_id,@grade)
                      END
           else
           select 'Please Cheak Your Student ID'
```

2.1.46. Procedure: Student_course_Select_SP

Input/Output

	Name	Data type	Description
→@	Student_id	int	

Script

```
create proc [dbo].[Student_course_Select_SP] @Student_id int=null
    as
           if @Student_id is null
                     select * from Student_Course
           else
                     select * from Student_Course where std_id=@Student_id
```

2.1.47. Procedure: Student_course_Update_SP

Input/Output

	Name	Data type	Description
→@	Student_id	int	
→@	crs_id	int	
→@	grad_overall	int	

Script

```
create proc [dbo].[Student_course_Update_SP] @Student_id int,@crs_id int ,@grad_overall int=null
as

begin
     declare @grade int = (select C.grade_overall from Student_Course C where C.std_id=@Student_id and C.crs_id = @crs_id)

            if exists(select std_id,@crs_id from Student_Course Q where std_id=@Student_id and Q.crs_id=@crs_id)
            begin
                      set @grade = (coalesce(@grad_overall,@grade))

                      update Student_Course
                      set grade_overall =@grade
            end
            else
            print 'Student does not exist'
end
```

2.1.48. Procedure: Student_Delete_SP

Input/Output

	Name	Data type	Description
→@	Student_id	int	

Script

```
create PROC [dbo].[Student_Delete_SP] @Student_id int
    AS
           DELETE FROM Student WHERE std_id = @Student_id
```

2.1.49. Procedure: Student_Insert_SP

Input/Output

	Name	Data type	Description
→@	Student_id	int	
→@	student_fname	nvarchar(50)	
→@	student_lname	nvarchar(50)	
→@	Student_age	int	
→@	student_phone	nvarchar(50)	
→@	dept_id	int	

Script

```
CREATE proc [dbo].[Student_Insert_SP] @Student_id int,  @student_fname nvarchar(50) ,@student_lname nvarchar(50),
 @Student_age int, @student_phone nvarchar(50),@dept_id int

as

           if not exists (select std_id from Student s where s.std_id=@student_id)
                     BEGIN
                     insert into Student(std_id,std_fname,std_lname,std_age,std_phone,dept_id)
                     values(@Student_id,@student_fname,@student_lname
                      ,@Student_age, @student_phone,@dept_id)
                     END
           else
           select 'Please Cheak Your Student ID'
```

2.1.50. Procedure: Student_Select_SP

Input/Output

	Name	Data type	Description
→@	Student_id	int	

Script

```
create proc [dbo].[Student_Select_SP] @Student_id int=null
    as
           if @Student_id is null
                     select * from Student
           else
                     select * from Student where std_id=@Student_id
```

2.1.51. Procedure: Student_Update_SP

Input/Output

	Name	Data type	Description
→@	Student_id	int	
→@	student_phone	nvarchar(50)	
→@	std_age	int	
→@	f_name	nvarchar(50)	
→@	l_name	nvarchar(50)	
→@	dept	int	

Script

```
CREATE proc [dbo].[Student_Update_SP] @Student_id int,@student_phone nvarchar(50)=null,@std_age int=null,@f_name
nvarchar(50)=null,
                                @l_name nvarchar(50)=null,@dept int=null
as
begin
            declare @student_phone2 nvarchar(50) = (select std_phone from Student where std_id=@Student_id)
      declare @fname nvarchar(50) = (select std_fname from Student where std_id=@Student_id)
      declare @lname nvarchar(50) = (select std_lname from Student where std_id=@Student_id)
      declare @age nvarchar(50) = (select std_age from Student where std_id=@Student_id)
      declare @dpt nvarchar(50) = (select dept_id from Student where std_id=@Student_id)

            if exists(select std_id from Student Q where std_id=@Student_id)
            begin
                        set @student_phone2 = (coalesce(@student_phone,@student_phone2))
                        set @fname = (coalesce(@f_name,@fname))
                        set @lname = (coalesce(@l_name,@lname))
                        set @age = (coalesce(@std_age,@age))
                        set @dpt = (coalesce(@dept,@dpt))

                        update Student
                        set std_phone =@student_phone2 ,
                        std_fname =@fname ,
                        std_lname =@lname ,
                        std_age =@age ,
                        dept_id =@dpt
            end
            else
            print 'Student does not exist'
end
```