OnlineExam@DESKTOP-KKGNO6I\SQLEXPRESS

Data Dictionary

2023-01-12

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

OnlineExam@DESKTOP-KKGNO6I\SQLEXPRESS

1. Tables

1.1. Table: Course

Columns

Name	Data type	Description / Attributes
Crs_ID	int	Identity / Auto increment
Crs_Name	varchar(50)	Nullable
Crs_Duration	int	Nullable

Linked from

Table	Join	Title / Name / Description
Ins_crs	**Course**Crs_ID = Ins_crsCrs_id	FK_Ins_crs_Course
Questions	**Course**Crs_ID = QuestionsCrs_id	FK_Questions_Course
std_crs	**Course**Crs_ID = std_crsCrs_id	FK_std_crs_Course
Topic	**Course**Crs_ID = Topiccourse_id	FK_Topic_Course

Unique keys

Columns	Name / Description
Crs_ID	PK_Course

Used By

Name
Course
Ins_crs
Questions
std_crs
Topic

1.2. Table: Department

Columns

Name	Data type	Description / Attributes
Dept_Id	int	**Identity / Auto increment**
Dept_name	varchar(50)	**Nullable**
Dept_Desc	varchar(50)	**Nullable**
Dept_Loaction	varchar(50)	**Nullable**
Dept_Manager	varchar(50)	**Nullable**
manager_HireDate	int	**Nullable**

Linked from

Table	Join	Title / Name / Description
Instructor	**Department**Dept_Id = InstructorDept_id	FK_Instructor_Department
Student	**Department**Dept_Id = StudentDept_Id	FK_Student_Department

Unique keys

Columns	Name / Description
Dept_Id	PK_Department

Used By

Name
Department
Instructor
Student

1.3. Table: Exam

Columns

Name			Data type	Description / Attributes
		Exam_id	int	**Identity / Auto increment**
		Exam_name	varchar(50)	**Nullable**
		Exam_date	date	**Nullable** **Default**: getdate()

Linked from

Table	Join	Title / Name / Description
Qest_Exam	**Exam**Exam_id = Qest_ExamExam_id	FK_Qest_Exam_Exam
Std_Ans	**Exam**Exam_id = Std_AnsExam_id	FK_Std_Ans_Exam

Unique keys

Columns	Name / Description
Exam_id	PK_Exam

Used By

Name
Exam
Qest_Exam
Std_Ans

1.4. Table: Ins_crs

Columns

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

Links to

Table	Join	Title / Name / Description
Course	**Ins_crs**Crs_id = CourseCrs_ID	FK_Ins_crs_Course
Instructor	**Ins_crs**Ins_id = InstructorIns_id	FK_Ins_crs_Instructor

Unique keys

Columns	Name / Description
Ins_id, Crs_id	PK_Ins_crs

Uses

Name
Ins_crs
Course
Instructor

1.5. Table: Instructor

Columns

Name	Data type	Description / Attributes
Ins_id	int	**Identity / Auto increment**
Ins_F_name	varchar(50)	**Nullable**
Ins_L_name	varchar(50)	**Nullable**
Ins_Degree	varchar(50)	**Nullable**
salary	float	**Nullable**
Dept_id	int	**Nullable** **References**: Department

Links to

Table	Join	Title / Name / Description
Department	**Instructor**Dept_id = DepartmentDept_Id	FK_Instructor_Department

Linked from

Table	Join	Title / Name / Description
Ins_crs	**Instructor**Ins_id = Ins_crsIns_id	FK_Ins_crs_Instructor

Unique keys

Columns	Name / Description
Ins_id	PK_Instructor

Uses

Name
Instructor
Department

Used By

Name
Instructor
Ins_crs

1.6. Table: Qest_Exam

Columns

		Name	Data type	Description / Attributes
		Exam_id	int	**References**: Exam
		Qest_id	int	**References**: Questions

Links to

	Table	Join	Title / Name / Description
	Exam	**Qest_Exam**Exam_id = ExamExam_id	FK_Qest_Exam_Exam
	Questions	**Qest_Exam**Qest_id = QuestionsQuest_id	FK_Qest_Exam_Questions

Unique keys

Columns	Name / Description
Qest_id, Exam_id	PK_Qest_Exam

Uses

Name
Qest_Exam
Exam
Questions

1.7. Table: Questions

Columns

Name	Data type	Description / Attributes
Quest_id	int	**Identity / Auto increment**
correct_ans	varchar(50)	**Nullable**
Crs_id	int	**Nullable** **References**: Course
question_body	varchar(200)	**Nullable**
Quest_type	char(1)	**Nullable**

Links to

Table	Join	Title / Name / Description
Course	**Questions**Crs_id = CourseCrs_ID	FK_Questions_Course

Linked from

Table	Join	Title / Name / Description
Qest_Exam	**Questions**Quest_id = Qest_ExamQest_id	FK_Qest_Exam_Questions
QuestMultiChoices	**Questions**Quest_id = QuestMultiChoicesQuest_id	FK_QuestMultiChoices_Questions
Std_Ans	**Questions**Quest_id = Std_AnsQuest_id	FK_Std_Ans_Questions

Unique keys

Columns	Name / Description
Quest_id	PK_Questions

Uses

Name
Questions
Course

Used By

Name
Questions
Qest_Exam
QuestMultiChoices
Std_Ans

1.8. Table: QuestMultiChoices

Columns

		Name	Data type	Description / Attributes
		Quest_id	int	**References**: Questions
		ChoiceNum	char(1)	
		ChoiceVal	nvarchar(200)	**Nullable**

Links to

	Table	Join	Title / Name / Description
	Questions	**QuestMultiChoices**Quest_id = QuestionsQuest_id	FK_QuestMultiChoices_Questions

Unique keys

	Columns	Name / Description
	Quest_id, ChoiceNum	PK_QuestMultiChoices

Uses

Name
QuestMultiChoices
Questions

1.9. Table: Std_Ans

Columns

Name	Data type	Description / Attributes
Std_id	int	**References**: Student
Exam_id	int	**References**: Exam
Quest_id	int	**References**: Questions
Std_Ans	varchar(50)	**Nullable**

Links to

Table	Join	Title / Name / Description
Exam	**Std_Ans**Exam_id = ExamExam_id	FK_Std_Ans_Exam
Questions	**Std_Ans**Quest_id = QuestionsQuest_id	FK_Std_Ans_Questions
Student	**Std_Ans**Std_id = StudentstdID	FK_Std_Ans_Student

Unique keys

Columns	Name / Description
Std_id, Exam_id, Quest_id	PK_Std_Ans

Uses

Name
Std_Ans
Exam
Questions
Student

1.10. Table: std_crs

Columns

		Name	Data type	Description / Attributes
		Std_id	int	**References**: Student
		Crs_id	int	**References**: Course
		Grades	float	**Nullable**

Links to

Table	Join	Title / Name / Description
Course	**std_crs**Crs_id = CourseCrs_ID	FK_std_crs_Course
Student	**std_crs**Std_id = StudentstdID	FK_std_crs_Student

Unique keys

Columns	Name / Description
Std_id, Crs_id	PK_std_crs

Uses

Name
std_crs
Course
Student

1.11. Table: Student

Columns

Name	Data type	Description / Attributes
stdID	int	**Identity / Auto increment**
std_f_name	varchar(50)	**Nullable**
std_l_name	varchar(50)	**Nullable**
std_address	varchar(50)	**Nullable**
std_age	int	**Nullable**
Dept_Id	int	**Nullable** **References**: Department

Links to

Table	Join	Title / Name / Description
Department	**Student**Dept_Id = DepartmentDept_Id	FK_Student_Department

Linked from

Table	Join	Title / Name / Description
Std_Ans	**Student**stdID = Std_AnsStd_id	FK_Std_Ans_Student
std_crs	**Student**stdID = std_crsStd_id	FK_std_crs_Student

Unique keys

Columns	Name / Description
stdID	PK_Student

Uses

Name
Student
Department

Used By

Name
Student
Std_Ans
std_crs

1.12. Table: Topic

Columns

Name	Data type	Description / Attributes
Topic_id	int	**Identity / Auto increment**
Topic_name	varchar(50)	**Nullable**
course_id	int	**Nullable** **References**: Course

Links to

Table	Join	Title / Name / Description
Course	**Topic**course_id = CourseCrs_ID	FK_Topic_Course

Unique keys

Columns	Name / Description
Topic_id	PK_Topic

Uses

Name
Topic
Course

2. Procedures

2.1. Procedure: calcGrade

Input/Output

	Name	Data type	Description
→@	stdid	int	
→@	examid	int	
@→	grades	float	

Script

```
create   proc calcGrade @stdid int,@examid int,@grades float out
as
          declare c1 cursor
          for select a.Std_Ans ,q.correct_ans
                    from Questions q, Std_Ans a
                    where q.Quest_id=a.Quest_id and a.Std_id=@stdid
          for read only     --no update  --no modification

          declare @std char,@corr char, @correct int=0,@total float=8;
          open c1
          fetch c1 into @std,@corr

          while @@FETCH_STATUS=0
                    begin
                              select @std as 'Student Answer',@corr as 'Correct Answer'
                              fetch c1 into @std,@corr
                              if @std=@corr
                                        set @correct+=1
                    end
          close c1
          deallocate c1
          set @gradeS =(@correct/@total)*100
```

2.2. Procedure: ExamAnswers

Input/Output

	Name	Data type	Description
→@	QuestArray	nvarchar(MAX)	
→@	AnsArray	nvarchar(8)	
→@	stdid	int	
→@	examid	int	

Script

```
create   proc ExamAnswers @QuestArray nvarchar(max), @AnsArray nvarchar(8), @stdid int, @examid int
as
          declare @counter int=1, @quest int,@ans char;

          --- split question array by ','
          declare @TempTable AS table (Id int identity, questID int)
          insert into @TempTable(questID)
          SELECT [Value]  FROM STRING_SPLIT(@QuestArray,',')

          while(@counter<9)
                    begin
                              select  @quest=questID from @TempTable where Id=@counter
                              select @ans=SUBSTRING(@AnsArray,@counter, 1);

                              exec InsertStudAns @stdid, @examid,@quest,@ans
                              set @counter+=1
                    end
```

2.3. Procedure: ExamCorrection

Input/Output

	Name	Data type	Description
→@	stdid	int	
→@	examid	int	
→@	crsid	int	

Script

```
create   proc ExamCorrection @stdid int,@examid int,@crsid int
as
          declare @grade float=0;
          exec calcGrade @stdid,@examid,@grade out
          select @grade as Grade
          exec updateGrade @stdid,@crsid,@grade
```

2.4. Procedure: GenerateExam

Input/Output

	Name	Data type	Description
→@	examname	varchar(50)	
→@	courseId	int	

Script

```
create   proc GenerateExam @examname varchar(50), @courseId int
as

           insert into Exam (Exam_name,Exam_date) values(@examname, GETDATE())
           declare @examid int =@@IDENTITY
           --- insert true and false
           exec QuestionGenerator 3,2,@examid,@courseId

    --- insert mcq
           exec QuestionGenerator 5,1,@examid,@courseId

-----------------------------
```

2.5. Procedure: InsertStudAns

Input/Output

	Name	Data type	Description
→@	stdif	int	
→@	examid	int	
→@	questid	int	
→@	ans	char(1)	

Script

```
create proc InsertStudAns @stdif int, @examid int, @questid int, @ans char
as
          insert into Std_Ans(Std_id,Exam_id,Quest_id,Std_Ans)
          values(@stdif,@examid,@questid,@ans)
```

2.6. Procedure: InstChoices

Input/Output

	Name	Data type	Description
→@	QuestionId	int	
→@	choiceNum	char(1)	
→@	choiceVal	nvarchar(200)	

Script

```
CREATE proc InstChoices @QuestionId int, @choiceNum char,@choiceVal nvarchar(200)
as
          insert into QuestMultiChoices values (@QuestionId,@choiceNum,@choiceVal)
```

2.7. Procedure: InstCourse

Input/Output

	Name	Data type	Description
→@	CourseName	varchar(50)	
→@	CrsDuration	int	

Script

```
create   proc InstCourse  @CourseName varchar(50), @CrsDuration int
as
          insert into Course (Crs_Name,Crs_Duration)
          values (@CourseName,@CrsDuration)
```

2.8. Procedure: InstDept

Input/Output

	Name	Data type	Description
→@	name	varchar(50)	
→@	Desc	varchar(50)	
→@	Location	varchar(50)	
→@	Manager	varchar(50)	
→@	Manger_HireDate	int	

Script

```
create proc InstDept @name varchar(50), @Desc varchar(50), @Location varchar(50),@Manager varchar(50),@Manger_HireDate int
as
          insert into Department (Dept_name,Dept_Desc,Dept_Loaction,Dept_Manager,manager_HireDate)
          values (@name,@Desc,@Location,@Manager,@Manger_HireDate)
```

2.9. Procedure: InstExam

Input/Output

	Name	Data type	Description
→@	ExamName	varchar(50)	

Script

```
create   proc InstExam @ExamName varchar(50)
as
          insert into Exam(Exam_name,Exam_date)
          values (@ExamName,GETDATE())
```

2.10. Procedure: InstInsc

Input/Output

	Name	Data type	Description
→@	Fname	varchar(50)	
→@	Lname	varchar(50)	
→@	Degree	varchar(50)	
→@	Salary	float	
→@	Dept_id	int	

Script

```
create proc InstInsc @Fname varchar(50), @Lname varchar(50), @Degree varchar(50),@Salary float,@Dept_id int
as
          insert into Instructor (Ins_F_name,Ins_L_name,Ins_Degree,salary,Dept_id)
          values (@Fname,@Lname,@Degree,@Salary,@Dept_id)
```

2.11. Procedure: InstQuestion

Input/Output

	Name	Data type	Description
→@	CorrectAns	varchar(50)	
→@	CourseID	int	
→@	QuestionBody	varchar(200)	

Script

```
create   proc InstQuestion @CorrectAns varchar(50),@CourseID int , @QuestionBody varchar(200)
as
          insert into Questions (correct_ans,Crs_id,question_body)
          values (@CorrectAns,@CourseID,@QuestionBody)
```

2.12. Procedure: InstStd

Input/Output

	Name	Data type	Description
→@	Fname	varchar(50)	
→@	Lname	varchar(50)	
→@	Adress	varchar(50)	
→@	age	int	
→@	Dept_id	int	

Script

```
create    proc InstStd @Fname varchar(50), @Lname varchar(50), @Adress varchar(50),@age int,@Dept_id int
as
          insert into Student(std_f_name,std_l_name,std_address,std_age,Dept_Id)
          values (@Fname,@Lname,@Adress,@age,@Dept_id)
```

2.13. Procedure: InstTopic

Input/Output

	Name	Data type	Description
→@	TopicName	varchar(50)	
→@	courseID	int	

Script

```
create   proc InstTopic  @TopicName varchar(50), @courseID int
as
          insert into Topic (Topic_name,course_id)
          values (@TopicName,@courseID)
```

2.14. Procedure: QuestionGenerator

Input/Output

	Name	Data type	Description
→@	numQuest	int	
→@	typeQuest	int	
→@	examid	int	
→@	courseId	int	

Script

```
create   proc QuestionGenerator @numQuest int,@typeQuest int, @examid int, @courseId int
as

           insert into Qest_Exam(Qest_id,Exam_id)
           select top(@numQuest) Quest_id , @examid
           from Questions q inner join Course c
           on q.Crs_id = c.Crs_ID
           where q.Quest_type=@typeQuest and  q.Crs_id=@courseId
           order by NEWID()
```

2.15. Procedure: REP1_StudentInformation

Input/Output

	Name	Data type	Description
→@	deptId	int	

Script

```
create   proc REP1_StudentInformation @deptId int
as
          exec select_StudentDept @deptId
```

2.16. Procedure: REP2_std_Grades

Input/Output

	Name	Data type	Description
→@	std_id	int	

Script

```
create   proc REP2_std_Grades @std_id int
as
          exec select_StudentGrades @std_id
```

2.17. Procedure: REP3_Instructor_Course

Input/Output

	Name	Data type	Description
→@	ins_id	int	

Script

```
create   proc REP3_Instructor_Course @ins_id int
as
          exec select_InstCourses @ins_id
```

2.18. Procedure: REP4_Course_Topic

Input/Output

	Name	Data type	Description
→@	crs_id	int	

Script

```
create   proc REP4_Course_Topic @crs_id int
as
          exec select_CourseTopics  @crs_id
```

2.19. Procedure: REP5_ExamChoices

Input/Output

	Name	Data type	Description
→@	Exam_id	int	

Script

```
create    proc REP5_ExamChoices @Exam_id int
as

           SELECT Questions.question_body, QuestMultiChoices.ChoiceVal
           FROM   Exam INNER JOIN
                                        Qest_Exam ON Exam.Exam_id = Qest_Exam.Exam_id INNER JOIN
                                        Questions ON Qest_Exam.Qest_id = Questions.Quest_id INNER JOIN
                                        QuestMultiChoices ON Questions.Quest_id = QuestMultiChoices.Quest_id


           where @Exam_id=Exam.Exam_id
           --true and false
           union all
           select * from Create_TrueAndFalse(@Exam_id)
```

2.20. Procedure: REP6_ExamStudAnswers

Input/Output

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

Script

```
create   proc  REP6_ExamStudAnswers @Exam_id int,@Stdid int
as
SELECT Std_Ans.Std_Ans, Questions.question_body
FROM   Questions INNER JOIN
            Std_Ans ON Questions.Quest_id = Std_Ans.Quest_id
where   @Exam_id =Std_Ans.Exam_id and @Stdid=Std_Ans.Std_id

------------------------------------------------------
```

2.21. Procedure: select_Course

Script

```
create Procedure select_Course
as
select * from Course

--execute select_Course

------- Select Course id ------
```

2.22. Procedure: select_Course_id

Script

```
create Procedure select_Course_id
as
select Crs_ID from Course

--execute select_Course_id

------------------------------------------- Department Procedure ------------------------------------------------------
-----Select * --------
```

2.23. Procedure: select_CourseTopics

Input/Output

	Name	Data type	Description
→@	crs_id	int	

Script

```
create proc select_CourseTopics @crs_id int
as
          select Topic.Topic_name from Topic , Course
          where @crs_id = Crs_ID and Topic.course_id = Course.Crs_ID
```

2.24. Procedure: select_department

Script

```
create Procedure select_department
as
select * from Department

--execute select_department

------- Select Department id ------
```

2.25. Procedure: select_department_id

Script

```
create Procedure select_department_id
as
select Dept_Id from Department

--execute select_department_id

-----------------------------------------------Exam Procedure ------------------------------------------------------
-----Select * --------
```

2.26. Procedure: select_exam

Script

```
create Procedure select_exam
as
select * from Exam

--execute select_exam

------- Select Exam id ------
```

2.27. Procedure: select_exam_id

Script

```
create Procedure select_exam_id
as
select Exam_Id from Exam

--execute select_exam_id

-----------------------------------------------Instructor Procedure ------------------------------------------------------
-----Select * --------
```

2.28. Procedure: select_Ins_crs

Script

```
create Procedure select_Ins_crs
as
select * from Ins_crs

--execute select_Ins_crs

----------------------------------------------Qest_Exam Procedure ------------------------------------------------------
-----Select * --------
```

2.29. Procedure: select_InstCourses

Input/Output

	Name	Data type	Description
→@	ins_id	int	

Script

```
create  proc select_InstCourses @ins_id int
           as
                      select Course.Crs_Name , COUNT(Std_id) as Number_of_Students from Ins_crs , Course , std_crs
                      where @ins_id =Ins_id and Ins_crs.Crs_id = Course.Crs_ID and std_crs.Crs_id = Course.Crs_ID
                      group by Crs_Name
```

2.30. Procedure: select_instructor

Script

```
create Procedure select_instructor
as
select * from Instructor

--execute select_instructor

------- Select Instructor id ------
```

2.31. Procedure: select_instructor_id

Script

```
create Procedure select_instructor_id
as
select Ins_id from Instructor

--execute select_instructor_id

-----------------------------------------------Ins_crs Procedure ----------------------------------------------------
-----Select * --------
```

2.32. Procedure: select_Qest_Exam

Script

```
create Procedure select_Qest_Exam
as
select * from Qest_Exam

--execute select_Qest_Exam

-----------------------------------------------Questions Procedure ------------------------------------------------------
-----Select * --------
```

2.33. Procedure: select_Questions

Script

```
create Procedure select_Questions
as
select * from Questions

--execute select_Questions

------- Select Questions id ------
```

2.34. Procedure: select_Questions_id

Script

```
create Procedure select_Questions_id
as
select Quest_id from Questions

--execute select_Questions_id

-----------------------------------------------QuestMultiChoices Procedure ------------------------------------------------------
--
-----Select * --------
```

2.35. Procedure: select_QuestMultiChoices

Script

```
create Procedure select_QuestMultiChoices
as
select * from QuestMultiChoices

--execute select_QuestMultiChoices


-----------------------------------------------Std_Ans Procedure ---------------------------------------------------
-----Select * --------
```

2.36. Procedure: select_Std_Ans

Script

```
create Procedure select_Std_Ans
as
select * from Std_Ans

--execute select_Std_Ans

-----------------------------------------------std_crs Procedure ------------------------------------------------------
-----Select * --------
```

2.37. Procedure: select_std_crs

Script

```
create Procedure select_std_crs
as
select * from std_crs

--execute select_std_crs

---------------------------------------------- Topic Procedure ------------------------------------------------------

-----Select * --------
```

2.38. Procedure: select_stdAns_corrrect

Input/Output

	Name	Data type	Description
→@	stdid	int	

Script

```
create   proc select_stdAns_corrrect @stdid int
as
select a.Std_Ans ,q.correct_ans
                    from Questions q, Std_Ans a
                    where q.Quest_id=a.Quest_id and a.Std_id=@stdid
```

2.39. Procedure: select_student

Script

```
create Procedure select_student
as
select * from Student

--execute select_student

------- Select Student id ------
```

2.40. Procedure: select_student_id

Script

```
create Procedure select_student_id
as
select stdID from Student

--execute select_student_id

---------------------------------------------- Course Procedure ------------------------------------------------------

-----Select * --------
```

2.41. Procedure: select_StudentDept

Input/Output

	Name	Data type	Description
→@	deptId	int	

Script

```
create proc select_StudentDept @deptId int
as
          select s.*,d.Dept_name
          from Student s, Department d
          where s.Dept_Id=d.Dept_Id and @deptId=d.Dept_Id
```

2.42. Procedure: select_StudentGrades

Input/Output

	Name	Data type	Description
→@	std_id	int	

Script

```
create   proc select_StudentGrades @std_id int
as
          select  c.Crs_Name,sc.Grades,CONCAT( s.std_f_name,' ',s.std_l_name)
          from Student s, std_crs sc, Course c
          where s.stdID = sc.Std_id and sc.Crs_id = c.Crs_ID and sc.Std_id=@std_id
```

2.43. Procedure: select_Topic

Script

```
create Procedure select_Topic
as
select * from Topic

--execute select_Topic

------- Select Topic id ------
```

2.44. Procedure: select_Topic_id

Script

```
create Procedure select_Topic_id
as
select Topic_id from Topic
```

2.45. Procedure: updateCrs

Input/Output

	Name	Data type	Description
→@	crsId	int	
→@	CrsName	varchar(20)	

Script

```
create   proc updateCrs  @crsId int , @CrsName varchar(20)
as
          update Course
          set  Crs_Name = @CrsName
          where  Crs_ID = @crsId
```

2.46. Procedure: updateGrade

Input/Output

	Name	Data type	Description
→@	stdid	int	
→@	crs_id	int	
→@	newGrade	float	

Script

```
create   proc updateGrade @stdid int , @crs_id int , @newGrade float
as
          update std_crs
          set  std_crs.Grades = @newGrade
          where  Std_id = @stdid  and Crs_id = @crs_id
```

2.47. Procedure: updateStdAns

Input/Output

	Name	Data type	Description
→@	stdid	int	
→@	ExamId	int	
→@	QuestId	int	
→@	stdAns	varchar(20)	

Script

```
create   proc updateStdAns @stdid int , @ExamId int , @QuestId int , @stdAns varchar(20)
as
          update Std_Ans
          set  Std_Ans = @stdAns
          where  Std_id = @stdid  and Quest_id = @QuestId
```

2.48. Procedure: updateTopic

Input/Output

	Name	Data type	Description
→@	crsId	int	
→@	topicId	int	
→@	topicName	varchar(20)	

Script

```
create   proc updateTopic  @crsId int , @topicId int , @topicName varchar(20)
as
          update Topic
          set  Topic_name = @topicName
          where  course_id = @crsId  and Topic_id = @topicId
```

3. Functions

3.1. Function: Create_TrueAndFalse

Input/Output

	Name	Data type	Description
@	Returns	table type	
@	examid	int	

Script

```
create  function Create_TrueAndFalse  (@examid int)
returns @table table
                      (
                       body varchar(200),
                       choice nvarchar(200)
                      )
           as
           begin

           declare c1 cursor
           for  (select q.question_body
                      from Qest_Exam e, Questions q
                      where  e.Qest_id=q.Quest_id and q.Quest_type='2' and e.Exam_id=@examid)

           for read only     --no update  --no modification
           declare @questBodey varchar(200)
           open c1
           fetch c1 into @questBodey

           while @@FETCH_STATUS=0
                      begin
                                 insert into @table
                                 values (@questBodey,'True'),(@questBodey,'False')
                                 fetch c1 into @questBodey

                      end
           close c1
           deallocate c1
           return
           end
```