ExamSystem Data Dictionary

2022-03-21





Table of contents

ExamSy	ystem		7
1. Tab	oles		7
1.1.	Table: Choice		7
1.2.	Table: Course		8
1.3.	Table: Crs_Top		9
1.4.	Table: Department		10
1.5.	Table: Exam		11
1.6.	Table: Exm_Ques		12
1.7.	Table: Ins_Crs		13
1.8.	Table: Instructor		14
1.9.	Table: Question		16
1.10.	Table: Std_Crs		18
1.11.	Table: Std_Ques_Exm		19
1.12.	Table: Student		20
1.13.	Table: User		22
2.2.	Procedure: DeleteInstructor		24
2.3.	Procedure: DeleteStdByld		25
2.6.	Procedure: InsertInstructor		28
2.7.	Procedure: InsertNewStd		29
2.11.	Procedure: SelectInstructor		33
2.12	. Procedure: SelectStdById		34
2.13	. Procedure: SelectUserByld	*	35
2.14	. Procedure: sp_AssignAnsToStd		36
2.15	. Procedure: sp_CorrectStdExam		37
2.16	. Procedure: sp_CourseNameWithStudentCount		39
2.17	Procedure: sp_deleteChoice		40
2.18	. Procedure: sp_deleteCourse		41
2.19	. Procedure: sp_deleteDepartment		42
2.20	Procedure: sp_deleteExam		43
2.21	. Procedure: sp_DeleteQues		44
2.22	Procedure: sp_DeleteStdFromCrs		45
2.23	Procedure: sp_deleteTopic		46
2.24	Procedure: sp_ExamQuestionsAndAnswers		47
2.25	. Procedure: sp_GenerateExam		48
2.26	Procedure: sp_InsertChoice		49
2.27	. Procedure: sp_insertCourse		50
2.28	Procedure: sp_insertDepartment		51
2.29	Procedure: sp insertExam		52

2.30.	Procedure: sp_InsertStdAnsForExm	53
2.31.	Procedure: sp_InsertStdForCrs	54
2.32.	Procedure: sp_insertTopic	55
2.33.	Procedure: sp_selectchoice	56
2.34.	Procedure: sp_selectCourse	57
2.35.	Procedure: sp_selectDepartment	58
2.36.	Procedure: sp_selectExam	59
2.37.	Procedure: sp_SelectQues	60
2.38.	Procedure: sp_SelectStdAnsForExm	61
2.39.	Procedure: sp_SelectStdOfCrs	62
2.40.	Procedure: sp_selectTopic	63
2.41.	Procedure: sp_StdExamAnswers	64
2.42.	Procedure: sp_StudentGradePerCourse	65
2.43.	Procedure: sp_StudentperDepartment	66
2.44.	Procedure: sp_TopicsPerCourse	67
	Procedure: sp_updateChoice	
	Procedure: sp_updateCourse	
2.47.	Procedure: sp_UpdateCrsForIns	70
	Procedure: sp_UpdateCrsOfStd	
	Procedure: sp_updateDepartment	
	Procedure: sp_updateExam	
	Procedure: sp_updateTopic	
	Procedure: UpdateInstructor	
2.53.	Procedure: UpdateQues	76
	Procedure: UpdateStdInfo	
2.55.	Procedure: UpdateUser	78

Legend

- **?** Primary key
- Primary key disabled
- **1** 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
- → 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



ExamSystem

1. Tables

1.1. Table: Choice

Columns

		Name	Data type	Description / Attributes	
目	1	Cho_ld	int	Identity / Auto increment	
■		Ques_ld	int	Nullable References: Question	
■		Cho_Content	varchar(100)	Nullable	
■		Cho_Char	varchar(1)	Nullable	

Links to

Table	Join	Title / Name / Description
→ Question	ChoiceQues_Id = QuestionQues_Id	FK_Choice_Question

Linked from

	Table	Join	Title / Name / Description
\rightarrow	Question	ChoiceCho_ld = QuestionModel_Ans	FK_Question_Choice

Unique keys

	Columns			Name / Description
P	Cho_ld	PK_Choice		

Uses

	Name	
E Choice		
→ Question		

	Name
E Choice	
sp_CorrectStdExam	
sp_deleteChoice	
sp_ExamQuestionsAndAnswers	
sp_InsertChoice	
sp_selectchoice	
sp_StdExamAnswers	
sp_updateChoice	
→ Question	

1.2. Table: Course

Columns

		Name	Data type	Description / Attributes
▤	1	Crs_ld	int	Identity / Auto increment
■		Crs_Name	varchar(50)	Nullable
■		Crs_Desc	varchar(50)	Nullable
■		Crs_Dur	int	Nullable

Linked from

	Table	Join	Title / Name / Description
\leftarrow	Crs_Top	CourseCrs_Id = Crs_TopCrs_Id	FK_Crs_Top_Course
\rightarrow	Exam	CourseCrs_Id = ExamCrs_Id	FK_Exam_Course
\rightarrow	Ins_Crs	CourseCrs_Id = Ins_CrsCrs_Id	FK_Ins_Crs_Course
\rightarrow	Std_Crs	CourseCrs_Id = Std_CrsCrs_Id	FK_Std_Crs_Course

Unique keys

Columns			Name / Description	
?	Crs_ld	PK_Course		

Osed by	
	Name
E Course	
sp_CourseNameWithStudentCount	
☆ sp_deleteCourse	
sp_insertCourse	
sp_selectCourse	
sp_StudentGradePerCourse	
sp_updateCourse	
→ Crs_Top	
→ Exam	
→ Ins_Crs	
→ Std_Crs	

1.3. Table: Crs_Top

Columns

		Name	Data type	Description / Attributes
▤	1	Crs_ld	int	References: Course
■	1	Crs_Top	varchar(50)	

Links to

	Table	Join	Title / Name / Description
\rightarrow	Course	Crs_TopCrs_Id = CourseCrs_Id	FK_Crs_Top_Course

Unique keys

Columns	Name / Description
Crs_ld, Crs_Top	PK_Crs_Top

Uses

0303	
	Name
Ⅲ Crs_Top	
→ Course	

	Name
sp_deleteTopic	
sp_insertTopic	
sp_selectTopic	
sp_TopicsPerCourse	
sp_updateTopic	

1.4. Table: Department

Columns

		Name	Data type	Description / Attributes
■	1	Dept_ld	int	Identity / Auto increment
■		Dept_Name	varchar(50)	Nullable
■		Dept_Loc	varchar(50)	Nullable
■		Dept_ManagerHireDate	date	Nullable
B		Dept_ManagerId	int	Nullable References: Instructor

Links to

	Table	Join	Title / Name / Description
>	- I Instructor	Department Dept_ManagerId = InstructorIns_Id	FK_Department_Instructor

Linked from

	Table	Join	Title / Name / Description
\rightarrow	Instructor	DepartmentDept_Id = InstructorDept_Id	FK_Instructor_Department
\rightarrow	Student	DepartmentDept_Id = StudentDept_Id	FK_Student_Department

Unique keys

	Columns	Name / Description
9	Dept_ld	PK_Department

Uses

	Name
■ Department	
→ Instructor	

	Name
III Department	
sp_deleteDepartment	
sp_insertDepartment	
sp_selectDepartment	
sp_updateDepartment	
→ Instructor	
→ Student	

1.5. Table: Exam

Columns

		Name	Data type	Description / Attributes
■	1	Exm_ld	int	Identity / Auto increment
		Crs_ld	int	Nullable References: Course
		Generator_Id	int	Nullable References: Instructor
■		Exm_Grade	int	Nullable

Links to

Table		Join	Title / Name / Description
>	Course	Exam Crs_ld = CourseCrs_ld	FK_Exam_Course
—	Instructor	Exam Generator_ld = InstructorIns_ld	FK_Exam_Instructor

Linked from

	Table	Join	Title / Name / Description
\rightarrow	Exm_Ques	ExamExm_Id = Exm_QuesExm_Id	FK_Exm_Ques_Exam
\rightarrow	Std_Ques_Exm	Exam Exm_Id = Std_Ques_ExmExm_Id	FK_Std_Ques_Exm_Exam

Unique keys

	Columns		Name / Description
P Exm_ld		PK_Exam	

Uses

	Name
Ⅲ Exam	
→ Course	
→ Instructor	

Name	
Ⅲ Exam	
sp_CorrectStdExam	
☆ sp_deleteExam	
☆ sp_GenerateExam	
☆ sp_insertExam	
sp_selectExam	
☆ sp_updateExam	
→ Exm_Ques	
→ Std_Ques_Exm	

1.6. Table: Exm_Ques

Columns

		Name	Data type	Description / Attributes
■	1	Exm_ld	int	References: Exam
■	1	Ques_ld	int	References: Question
■		Grade	int	Nullable Default: 1

Links to

	Table	Join	Title / Name / Description
\rightarrow	Exam	Exm_QuesExm_Id = ExamExm_Id	FK_Exm_Ques_Exam
—	Question	Exm_QuesQues_ld = QuestionQues_ld	FK_Exm_Ques_Question

Unique keys

	Columns	Name / Description
P	Exm_ld, Ques_ld	PK_Exm_Ques

Uses

	Name	
Exm_Ques		
→ Exam		
→ Question		

	Name
Exm_Ques	
sp_ExamQuestionsAndAnswers	
sp_GenerateExam	

1.7. Table: Ins_Crs

Columns

		Name	Data type	Description / Attributes
■	1	lns_ld	int	References: Instructor
■	1	Crs_ld	int	References: Course

Links to

	Table	Join	Title / Name / Description
\rightarrow	Course	Ins_CrsCrs_Id = CourseCrs_Id	FK_Ins_Crs_Course
—	Instructor	Ins_CrsIns_Id = InstructorIns_Id	FK_Ins_Crs_Instructor

Unique keys

Columns	Name / Description
ns_ld, Crs_ld	PK_Ins_Crs

Uses

	Name
III Ins_Crs	
→ Course	
→ Instructor	

Name		
Ins_Crs		
DelCrsIns		
InsertInsForCrs		
SelectCrsIns		
sp_CourseNameWithStudentCount		
sp_UpdateCrsForIns		

1.8. Table: Instructor

Columns

		Name	Data type	Description / Attributes
▤	1	lns_ld	int	Identity / Auto increment
■		U_ld	int	Nullable References: User
■		Dept_ld	int	Nullable References: Department
■		Ins_Fname	varchar(50)	Nullable
▤		Ins_Lname	varchar(50)	Nullable
■		Ins_Degree	varchar(50)	Nullable
■		Ins_Salary	int	Nullable

Links to

	Table	Join	Title / Name / Description
-	Department	<pre>InstructorDept_Id = DepartmentDept_Id</pre>	FK_Instructor_Department
\rightarrow	User	InstructorU_Id = UserU_Id	FK_Instructor_User

Linked from

	Table	Join	Title / Name / Description
\rightarrow	Department	InstructorIns_Id = DepartmentDept_ManagerId	FK_Department_Instructor
\rightarrow	Exam	InstructorIns_Id = ExamGenerator_Id	FK_Exam_Instructor
\rightarrow	Ins_Crs	InstructorIns_Id = Ins_CrsIns_Id	FK_Ins_Crs_Instructor

Unique keys

Columns	Name / Description
Ŷ Ins_Id	PK_Instructor

Uses

	Name
Instructor	
→ Department	
→ User	

	Name
Instructor	
DeleteInstructor	
InsertInstructor	
Selectinstructor	
\$\tag{\partial} UpdateInstructor	

	Name
→ Department	
→ Exam	
→ Ins_Crs	



1.9. Table: Question

Columns

		Name	Data type	Description / Attributes
■	1	Ques_ld	int	Identity / Auto increment
■		Crs_ld	int	Nullable
■		Ques_Content	varchar(300)	Nullable
■		Ques_Grade	int	Nullable
■		Ques_Type	varchar(3)	Nullable
B		Model_Ans	int	Nullable References: Choice

Links to

	Table	Join	Title / Name / Description
\rightarrow	Choice	QuestionModel_Ans = ChoiceCho_Id	FK_Question_Choice

Linked from

		Table	Join	Title / Name / Description
-	→	Choice	QuestionQues_Id = ChoiceQues_Id	FK_Choice_Question
-	\rightarrow	Exm_Ques	Question Ques_Id = Exm_QuesQues_Id	FK_Exm_Ques_Question
	→	Std_Ques_Exm	QuestionQues_ld = Std_Ques_ExmQues_ld	FK_Std_Ques_Exm_Question

Unique keys

Columns			Name / Description
P	Ques_ld	PK_Question	

Uses

	Name	
■ Question		
→ Choice		

Name
Ⅲ Question
insertQues InsertQues
№ sp_CorrectStdExam
№ sp_DeleteQues
sp_ExamQuestionsAndAnswers
🔯 sp_GenerateExam
sp_SelectQues
🗽 sp_StdExamAnswers
UpdateQues

Name	
→ Choice	
→ Exm_Ques	
→ Std_Ques_Exm	



1.10. Table: Std_Crs

Columns

		Name	Data type	Description / Attributes
▤	1	Std_ld	int	References: Student
■	1	Crs_ld	int	References: Course
■		Grade	int	Nullable
■		Date	date	Nullable

Links to

	Table	Join	Title / Name / Description
\rightarrow	Course	Std_CrsCrs_Id = CourseCrs_Id	FK_Std_Crs_Course
\rightarrow	Student	Std_CrsStd_Id = StudentStd_Id	FK_Std_Crs_Student

Unique keys

Columns		Name / Description
Ŷ	Std_ld, Crs_ld	PK_Std_Crs

Uses

	Name
→ Course	
→ Student	

osed by					
	Name				
■ Std_Crs					
sp_CorrectStdExam					
sp_CourseNameWithStudentCou	nt				
sp_DeleteStdFromCrs					
sp_InsertStdForCrs					
sp_SelectStdOfCrs					
sp_StudentGradePerCourse					
sp_UpdateCrsOfStd					

1.11. Table: Std_Ques_Exm

Columns

	Name		Data type	Description / Attributes
■	1	Std_ld	int	References: Student
■	1	Exm_ld	int	References: Exam
■	1	Ques_ld	int	References: Question
■		Std_Answer	varchar(1)	

Links to

	Table	Join	Title / Name / Description
\rightarrow	Exam	Std_Ques_ExmExm_Id = ExamExm_Id	FK_Std_Ques_Exm_Exam
—	Question	Std_Ques_ExmQues_Id = QuestionQues_Id	FK_Std_Ques_Exm_Question
—	Student	Std_Ques_ExmStd_Id = StudentStd_Id	FK_Std_Ques_Exm_Student

Unique keys

	Columns		Name / Description
?	Std_ld, Exm_ld, Ques_ld	PK_Std_Ques_Exm	

Uses

	Name	
→ Exam		
→ Question		
→ Student		

		Name
sp_AssignAnsToStd		
sp_CorrectStdExam	Ť	
sp_InsertStdAnsForExm		
sp_SelectStdAnsForExm		
sp_StdExamAnswers		

1.12. Table: Student

Columns

		Name	Data type	Description / Attributes
	1	Std_Id	int	Identity / Auto increment
■		U_ld	int	Nullable References: User
		Dept_ld	int	Nullable References: Department
■		Std_Fname	varchar(50)	Nullable
		Std_Lname	varchar(50)	Nullable
		Std_BOD	date	Nullable
■		Std_Address	varchar(100)	Nullable

Links to

	Table		Join	Title / Name / Description
3	→ Department		StudentDept_Id = DepartmentDept_Id	FK_Student_Department
3	→ User		StudentU_Id = UserU_Id	FK_Student_User

Linked from

	Table	Join	Title / Name / Description
\rightarrow	Std_Crs	StudentStd_Id = Std_CrsStd_Id	FK_Std_Crs_Student
\rightarrow	Std_Ques_Exm	StudentStd_Id = Std_Ques_ExmStd_Id	FK_Std_Ques_Exm_Student

Unique keys

	Columns	Name / Description
Ŷ	Std_ld	PK_Student

Uses

	Name	
→ Department		
→ User		

Name
■ Student
DeleteStdByld
InsertNewStd
SelectStdByld
☆ sp_StudentGradePerCourse
sp_StudentperDepartment
₩ UpdateStdInfo

Nan	ne



 \leftarrow Std_Crs

→ Std_Ques_Exm

1.13. Table: User

Columns

		Name	Data type	Description / Attributes
■	1	U_ld	int	Identity / Auto increment
■		U_UserName	varchar(50)	
■		U_Email	varchar(50)	Nullable
■		U_Password	varchar(50)	
■		U_Sex	varchar(1)	Nullable
■		U_lsStd	bit	

Linked from

	Table	Join	Title / Name / Description
-	→ Instructor	User U_Id = InstructorU_Id	FK_Instructor_User
-	→ Student	User U_Id = StudentU_Id	FK_Student_User

Unique keys

	Columns		Name / Description
?	U_ld	PK_User	
P	U_ld	IX_User	

,	Name	
Ⅲ User		
DeleteUserByUserName		
InsertNewUser		
SelectUserByld		
UpdateUser		
→ Instructor		
→ Student		

2. Procedures

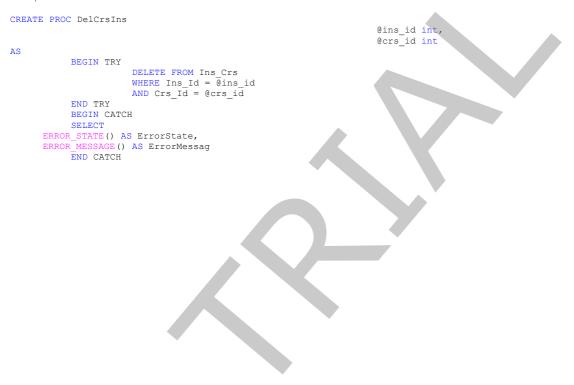
2.1. Procedure: DelCrsIns

Input/Output

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

Uses

	Name
DelCrsIns	
Ins_Crs	

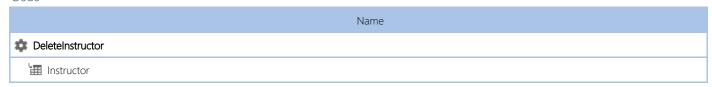


2.2. Procedure: DeleteInstructor

Input/Output

	Name	Data type	Description
→ @ ins_id		int	

Uses



```
CREATE PROC DeleteInstructor @ins_id int

AS

BEGIN TRY

DELETE FROM Instructor
WHERE Ins_Id = @ins_id

END TRY
BEGIN CATCH
SELECT
ERROR STATE() AS ErrorState,
ERROR_MESSAGE() AS ErrorMessag
END CATCH
```

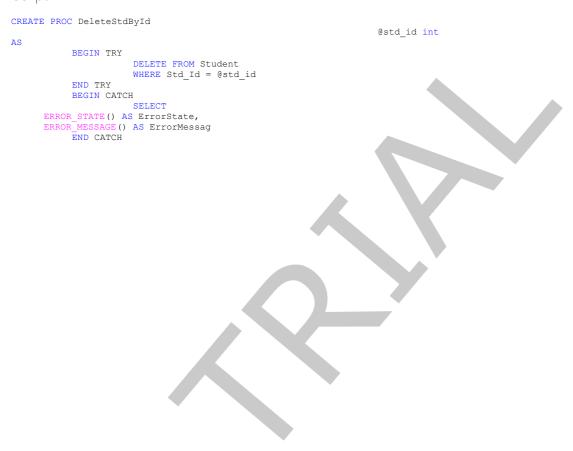
2.3. Procedure: DeleteStdByld

Input/Output

	Name	Data type	Description
→ @ std_id		int	

Uses

	Name	
DeleteStdByld		
Student		

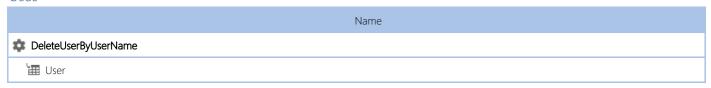


2.4. Procedure: DeleteUserByUserName

Input/Output

	Name	Data type	Description
→ @	userName	varchar(50)	

Uses



```
CREATE PROC DeleteUserByUserName @userName varchar(50)

AS

BEGIN TRY

DELETE FROM [User]
WHERE U_UserName = @userName

END TRY
BEGIN CATCH
SELECT

ERROR STATE() AS ErrorState,
ERROR MESSAGE () AS ErrorMessag

END CATCH
```

2.5. Procedure: InsertInsForCrs

Input/Output

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

Uses

	Name
Ins_Crs	

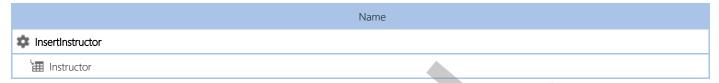


2.6. Procedure: InsertInstructor

Input/Output

	Name	Data type	Description
→@	u_id	int	
→ @	dept_id	int	
→ @	fname	varchar(50)	
→ @	Iname	varchar(50)	
→@	deg	varchar(50)	
→@	sal	int	

Uses



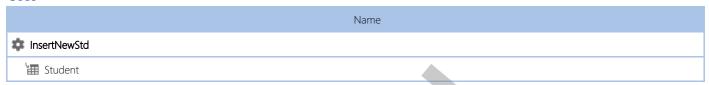


2.7. Procedure: InsertNewStd

Input/Output

	Name	Data type	Description
→ @	us_id	int	
→ @	dept_id	int	
→@	fname	varchar(50)	
→@	Iname	varchar(50)	
→ @	bod	date	
→ @	address	varchar(50)	

Uses



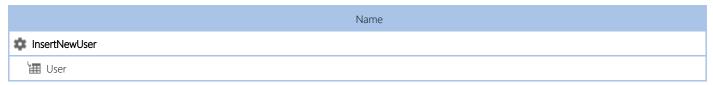
```
Script
CREATE PROC InsertNewStd
                                                                                                                                   @us_id int,
@dept_id int,
@fname varchar(50),
@lname varchar(50),
@bod date,
@address varchar(50)
AS
                     BEGIN TRY
                                           INSERT INTO Student VALUES (
                                                                  @us_id,
@dept_id,
@fname,
                                                                  @lname,
                                                                  @bod,
@address
                     END TRY
          SELECT
ERROR_STATE() AS ErrorState,
ERROR_MESSAGE() AS ErrorMessag
END CATCH
```

2.8. Procedure: InsertNewUser

Input/Output

	Name	Data type	Description
→ @	us_name	varchar(50)	
→ @	us_mail	varchar(50)	
→ @	us_pass	varchar(50)	
→ @	us_sex	varchar(1)	
→ @	us_isStd	bit	

Uses



```
Script
CREATE PROC [dbo].[InsertNewUser]
                                                                                                                  @us_name varchar(50),
@us_mail varchar(50),
@us_pass varchar(50),
@us_sex varchar(1),
@us_isStd_bit
AS
                BEGIN TRY
                                INSERT INTO [User]
                                VALUES (@us_name, @us_mail, @us_pass, @us_sex, @us_isStd)
                END TRY
                BEGIN CATCH
                                SELECT
                                                ERROR_STATE() AS ErrorState,
ERROR_MESSAGE() AS ErrorMessage
                END CATCH
```

2.9. Procedure: InsertQues

Input/Output

	Name	Data type	Description
→ @	crs_id	int	
→ @	cont	varchar(200)	
→ @	grade	int	
→ @	typ	varchar(3)	
→ @	ans	int	

Uses

	Name
InsertQues	
Question	



2.10. Procedure: SelectCrsIns

Input/Output

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

Uses

	Name
SelectCrsIns	
Ins_Crs	

Script

CREATE PROC SelectCrsIns

2.11. Procedure: SelectInstructor

Input/Output

	Name	Data type	Description
•@	ins_id	int	

Uses

	Name	
Selectinstructor		
Instructor		

```
CREATE PROC SelectInstructor @ins_id int

AS

BEGIN TRY

SELECT *
FROM Instructor
WHERE Ins_Id = @ins_id

END TRY
BEGIN CATCH
SELECT
ERROR STATE() AS ErrorState,
ERROR MESSAGE() AS ErrorMessag
END CATCH
```

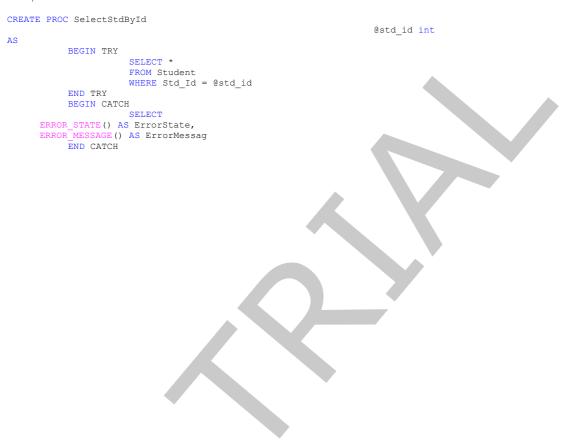
2.12. Procedure: SelectStdByld

Input/Output

	Name	Data type	Description
→@ st	td_id	int	

Uses

	Name	
SelectStdByld		
₩ Student		



2.13. Procedure: SelectUserByld

Input/Output

	Name	Data type	Description
→ @	us_id	int	

Uses

	Name
SelectUserByld	
⊞ User	

```
CREATE PROC [dbo].[SelectUserById] @us_id int

AS

BEGIN TRY

SELECT *
FROM [User]
WHERE U_Id = @us_id

END TRY

BEGIN CATCH
SELECT

FEROR STATE() AS ErrorState,
ERROR MESSAGE() AS ErrorMessag
END CATCH
```

2.14. Procedure: sp_AssignAnsToStd

Input/Output

	Name	Data type	Description
→@	std_id	int	
→ @	exm_id	int	
→ @	ans_dict	table type	

Uses

	Name
sp_AssignAnsToStd	
₩ Std_Ques_Exm	

2.15. Procedure: sp_CorrectStdExam

Input/Output

	Name	Data type	Description
→@	std_id	int	
•@	exm_id	int	

Uses

	Name
sp_CorrectStdExam	
Choice	
Exam	
Question	
₩ Std_Crs	
₩ Std_Ques_Exm	



```
CREATE PROC [dbo].[sp CorrectStdExam]
                                                                                                                               @std_id int,
                                                                                                                               @exm_id int
AS
               BEGIN TRY
                                DECLARE @LOCAL TABLEVARIABLE TABLE
                                                mod_ans varchar(1),
                                               std_ans varchar(1)
                                INSERT INTO @LOCAL_TABLEVARIABLE SELECT Cho_Char, Std_Answer
                                FROM Choice c
                               JOIN Question q ON (q.Model_Ans = c.Cho_Id)
JOIN Std_Ques_Exm sqe ON (sqe.Ques_Id = q.Ques_Id)
WHERE Exm_Id = @exm_id
                               AND Std_Id = @std_id
declare c cursor
FOR SELECT * FROM @LOCAL_TABLEVARIABLE
                               DECLARE @mod varchar(1), @ans varchar(1)
DECLARE @scr int, @grade int, @crs_id int
                                SET @scr = 0
                                SET @grade = 0
                               OPEN c
FETCH c INTO @mod, @ans
while @@FETCH_STATUS = 0
                                BEGIN
                                                IF @mod = @ans
                                                BEGIN
                                                              SET @scr = @scr + 1
                                                END
                                               FETCH c INTO @mod, @ans
                                SELECT @grade = Exm_Grade,
                                                               @crs_id = Crs_Id
                                FROM Exam
                               WHERE Exm_Id = @exm_id
                                DECLARE @ques grade int
                               SET @ques_grade = @grade / 10
SET @scr = @scr * @ques_grade
UPDATE Std_Crs
                               SET
                                                Grade = @scr,
                               Date = GETDA
WHERE Std_Id = @std_id
AND Crs_Id = @crs_id
               END TRY
               BEGIN CATCH
               SELECT
        ERROR_STATE() AS ErrorState,
ERROR_MESSAGE() AS ErrorMessag
               END CATCH
```

2.16. Procedure: sp_CourseNameWithStudentCount

Input/Output

	Name	Data type	Description
→ @	ins_id	int	

Uses

Name
sp_CourseNameWithStudentCount
E Course
Ins_Crs
描 Std_Crs

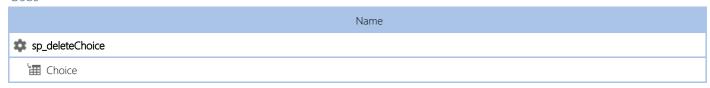


2.17. Procedure: sp_deleteChoice

Input/Output

	Name	Data type	Description
→ @	cho_id	int	

Uses



```
CREATE PROC sp_deletchoice @cho_id INT
AS

BEGIN TRY

DELETE PROM dbo.Choice
WHERE Cho_Id = @cho_id;
END TRY

BEGIN CATCH

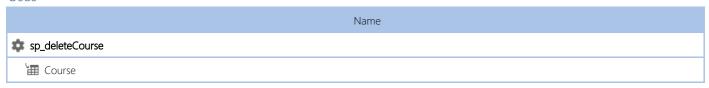
SELECT ERROR_STATE() AS ErrorState,
ERROR_MESSAGE() AS ErrorMessag;
END CATCH;
```

2.18. Procedure: sp_deleteCourse

Input/Output

	Name	Data type	Description
→ @ crs_id		int	

Uses



```
CREATE PROC sp_deleteCourse @crs_id INT
AS

BEGIN TRY

DELETE PROM dbo.Course
WHERE Crs_Id = @crs_id;
END TRY

BEGIN CATCH

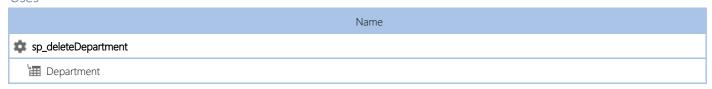
SELECT ERROR_STATE() AS ErrorState,
ERROR_MESSAGE() AS ErrorMessag;
END CATCH;
```

2.19. Procedure: sp_deleteDepartment

Input/Output

	Name	Data type	Description
→@ deptId		int	

Uses



```
CREATE PROC sp_deleteDepartment @deptId INT
AS
BEGIN TRY
DELETE FROM dbo.Department WHERE Dept_Id=@deptId;
END TRY
BEGIN CATCH
SELECT ERROR_STATE() AS ErrorState,
ERROR_MESSAGE() AS ErrorMessag;
END CATCH
```

2.20. Procedure: sp_deleteExam

Input/Output

	Name	Data type	Description
→ @	exam_id	int	

Uses

	Name
sp_deleteExam	
Ⅲ Exam	

```
CREATE PROC sp_deleteExam @exam_id INT
AS
BEGIN TRY
delete FROM dbo.Exam WHERE Exm_Id =@exam_id
END TRY

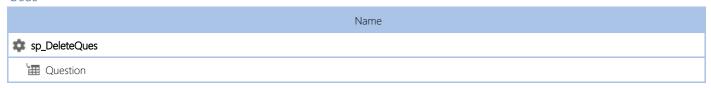
SELECT ERROR STATE() AS ErrorState,
ERROR_MESSAGE() AS ErrorMessag;
END CATCH
```

2.21. Procedure: sp_DeleteQues

Input/Output

	Name	Data type	Description
→ ques_id		int	

Uses





2.22. Procedure: sp_DeleteStdFromCrs

Input/Output

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

Uses

	Name
sp_DeleteStdFromCrs	
⊞ Std_Crs	



2.23. Procedure: sp_deleteTopic

Input/Output

	Name	Data type	Description
→ @	crs_id	int	
→ @	crs_top	varchar(50)	

Uses

	Name	
sp_deleteTopic		
⊞ Crs_Top		

```
CREATE PROC sp_deleteTopic
@crs_id INT,
@crs_top VARCHAR(50)

AS

BEGIN TRY

DELETE FROM dbo.Crs_Top
WHERE Crs_Id = @crs_id
AND Crs_Top = @crs_top;

END TRY

BEGIN CATCH
SELECT ERROR_STATE() AS ErrorState,
ERROR_MESSAGE() AS ErrorMessag;

END CATCH;
```

2.24. Procedure: sp_ExamQuestionsAndAnswers

Input/Output

	Name	Data type	Description
→@ exm_id		int	

Uses

1	Name
sp_ExamQuestionsAndAnswers	
E Choice	
Exm_Ques	
a Question	

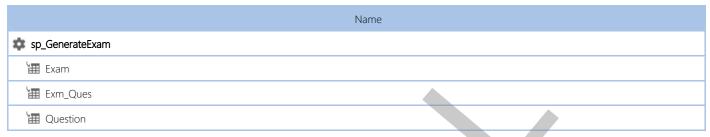


2.25. Procedure: sp_GenerateExam

Input/Output

	Name	Data type	Description
→@	mcq	int	
→@	tfq	int	
→@	crs_id	int	
→@	ins_id	int	
→@	grade	int	

Uses



```
CREATE PROC [dbo].[sp_GenerateExam]
                                                                                                                     @tfq int,
@crs_id int,
@ins_id int,
                                                                                                                     @grade int
              BEGIN TRY
                            INSERT INTO Exam
VALUES (@crs_id, @ins_id, @grade);
                             DECLARE @exm_id int;
SET @exm_id = (SELECT MAX(Exm_Id) FROM Exam);
                             DECLARE @LOCAL TABLEVARIABLE TABLE
                                           ex_id int NULL,
ques_id int NULL,
grade int NULL
                             INSERT INTO @LOCAL TABLEVARIABLE (ques_id)
SELECT TOP(@mcq) Ques_Id
                             FROM Question
                            WHERE Ques_Type = 'MCQ' AND Crs_Id = @crs_id ORDER BY NEWID()
                             INSERT INTO @LOCAL_TABLEVARIABLE (ques_id)
SELECT TOP(@tfq) Ques_Id
                            FROM Question
WHERE Ques_Type = 'TFQ' AND Crs_Id = @crs_id
ORDER BY NEWID()
                             UPDATE @LOCAL TABLEVARIABLE
                             SET ex_id = @exm_id,
                                          grade = 1
                             INSERT INTO Exm Ques
                             SELECT * FROM @LOCAL_TABLEVARIABLE
              END TRY
              BEGIN CATCH
              SELECT
       ERROR_STATE() AS ErrorState,
ERROR_MESSAGE() AS ErrorMessag
              END CATCH
/*********
```

2.26. Procedure: sp_InsertChoice

Input/Output

	Name	Data type	Description
→ @	ques_id	int	
→ @	cho_content	varchar(200)	
→ @	cho_char	varchar(1)	

Uses

	Name	
sp_InsertChoice		
E Choice		

2.27. Procedure: sp_insertCourse

Input/Output

	Name	Data type	Description
→ @	crs_name	varchar(50)	
→ @	crs_desc	varchar(50)	
→ @	crs_dur	int	

Uses

	Name
sp_insertCourse	
Course	

2.28. Procedure: sp_insertDepartment

Input/Output

	Name	Data type	Description
→@	dept_name	varchar(50)	
→ @	dept_loc	varchar(50)	
→@	dept_managerHireDate	date	
→ @	dept_managerld	int	

Uses

	Name
sp_insertDepartment	
Emportment	

2.29. Procedure: sp_insertExam

Input/Output

	Name	Data type	Description
→ @	crs_id	int	
→ @	generator_id	int	
→ @	exam_grade	int	

Uses

	Name
sp_insertExam	
Exam	

2.30. Procedure: sp_InsertStdAnsForExm

Input/Output

	Name	Data type	Description
→ @	std_id	int	
→ @	Exm_id	int	
→ @	ques_id	int	
→ @	ans	varchar(1)	

Uses

	Name	
sp_InsertStdAnsForExm		
└⊞ Std_Ques_Exm		



2.31. Procedure: sp_InsertStdForCrs

Input/Output

	Name	Data type	Description
→ @	std_id	int	
→ @	crs_id	int	
→ @	grade	int	
→ @	date	date	

Uses

	Name	
sp_InsertStdForCrs		
⊞ Std_Crs		



2.32. Procedure: sp_insertTopic

Input/Output

	Name	Data type	Description
→@	crs_id	int	
→ @	crs_top	varchar(50)	

Uses

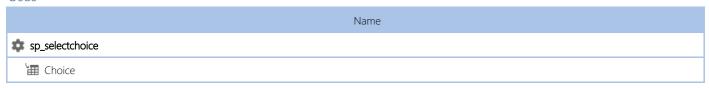
	Name
sp_insertTopic	
₩ Crs_Top	

2.33. Procedure: sp_selectchoice

Input/Output

	Name	Data type	Description
→ @	id	int	

Uses



```
CREATE PROC sp_selectchoice @id INT

AS
BEGIN TRY
SELECT *
FROM dbo.Choice
WHERE Cho_Id = @id;
END TRY

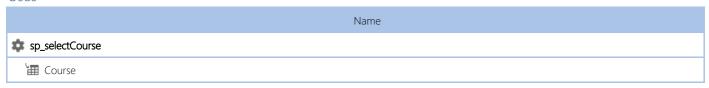
BEGIN CATCH
SELECT ERROR_STATE() AS ErrorState,
ERROR_MESSAGE() AS ErrorMessag;
END CATCH:
---course
```

2.34. Procedure: sp_selectCourse

Input/Output

	Name	Data type	Description
→ @ crs_id		int	

Uses



```
CREATE PROC sp_selectCourse @crs_id INT

AS

BEGIN TRY

SELECT *
FROM dbo.Course
WHERE Crs_id = @crs_id;

END TRY

BEGIN CATCH
SELECT ERROR_STATE() AS ErrorState,
ERROR_MESSAGE() AS ErrorMessag;

END CATCH;

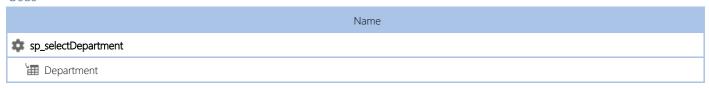
--crs_top
```

2.35. Procedure: sp_selectDepartment

Input/Output

	Name	Data type	Description
→@	dept_id	int	

Uses



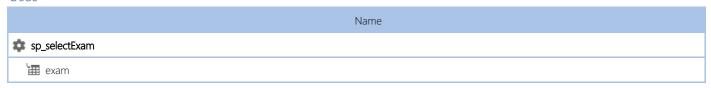
```
CREATE PROC sp_selectDepartment @dept_id INT
AS
BEGIN TRY
SELECT * FROM dbo.Department WHERE Dept_id =@dept_id;
END TRY
BEGIN CATCH
SELECT ERROR_STATE() AS ErrorState,
ERROR_MESSAGE() AS ErrorMessag;
END CATCH
```

2.36. Procedure: sp_selectExam

Input/Output

	Name	Data type	Description
→ @	exam_id	int	

Uses



```
CREATE PROC sp_selectExam @exam_id INT
AS
BEGIN TRY
SSLECT * FROM exam WHERE Exm_Id =@exam_id;
END TRY
BEGIN CATCH
SELECT ERROR_STATE() AS ErrorState,
ERROR_MESSAGE() AS ErrorMessag;
END CATCH
```

2.37. Procedure: sp_SelectQues

Input/Output

	Name	Data type	Description
→ ques_id		int	

Uses

	Name
sp_SelectQues	
(III Question	

```
CREATE PROC sp_SelectQues

AS

BEGIN TRY
FROM Question
WHERE Ques_Id = @ques_id

END TRY
BEGIN CATCH
SELECT
ERROR_STATE() AS ErrorState,
ERROR_MESSAGE() AS ErrorMessag
END CATCH
```

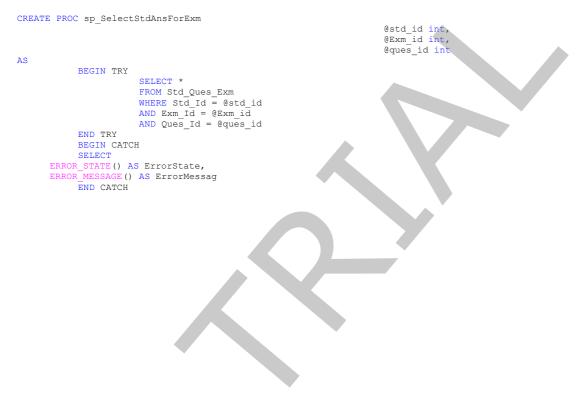
2.38. Procedure: sp_SelectStdAnsForExm

Input/Output

	Name	Data type	Description
→ @	std_id	int	
→ @	Exm_id	int	
→ @	ques_id	int	

Uses

	Name
sp_SelectStdAnsForExm	
₩ Std_Ques_Exm	



2.39. Procedure: sp_SelectStdOfCrs

Input/Output

	Name	Data type	Description
→ @	std_id	int	
•@	crs_id	int	

Uses

	Name
sp_SelectStdOfCrs	
Std_Crs	

```
AS

BEGIN TRY

SELECT *

FROM Std_Crs

WHERE Std_Id = @std_id

AND Crs_Id = @crs_id

END TRY

BEGIN CATCH

SELECT

ERROR_STATE() AS ErrorState,

ERROR_MESSAGE() AS ErrorMessag

END CATCH
```

2.40. Procedure: sp_selectTopic

Input/Output

	Name	Data type	Description
→ @	crs_id	int	
→ @	crs_topic	varchar(50)	

Uses

	Name	
sp_selectTopic		
Crs_Top		

```
CREATE PROC sp_selectTopic
@crs_id INT,
@crs_topic VARCHAR(50)

AS
BEGIN TRY
SELECT *
FROM dbo.Crs_Top
WHERE Crs_Id = @crs_id
AND Crs_Top = @crs_topic;
END TRY
BEGIN CATCH
SELECT ERROR_STATE() AS ErrorState,
ERROR_MESSAGE() AS ErrorMessag;
END CATCH;
--dept
```

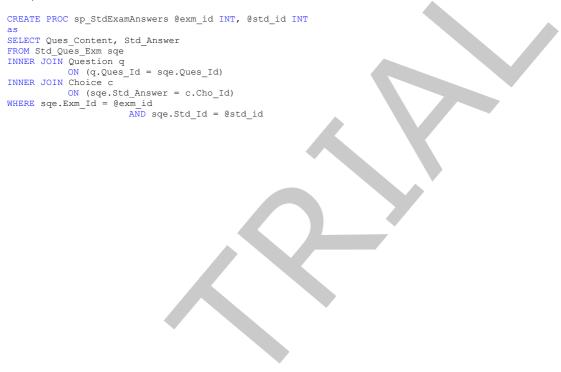
2.41. Procedure: sp_StdExamAnswers

Input/Output

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

Uses

	Name	
sp_StdExamAnswers		
Em Choice		
Question		
₩ Std_Ques_Exm		



2.42. Procedure: sp_StudentGradePerCourse

Input/Output

	Name	Data type	Description
→ @ std_id		int	

Uses

	Name
sp_StudentGradePerCourse	
E Course	
₩ Std_Crs	
₩ Student	

```
CREATE PROC sp_StudentGradePerCourse @std_id INT
AS
SELECT CONCAT(Std_Fname, ' ', Std_Lname) AS Student, c.Crs_Name AS Course, grade AS Grade
FROM Std_Crs sc
JOIN Student s
ON (s.Std_Id = sc.Std_Id)
JOIN Course c
ON (c.Crs_Id = sc.Crs_Id)
WHERE sc.Std_Id = @std_id
AND sc.Grade IS NOT NULL
```

2.43. Procedure: sp_StudentperDepartment

Input/Output

	Name	Data type	Description
→@	dept_id	int	

Uses

	Name	
sp_StudentperDepartment		
⊞ Student		

```
CREATE PROC sp_StudentperDepartment @dept_id INT
as
SELECT * FROM dbo.Student s WHERE s.Dept_Id = @dept_id;
```



2.44. Procedure: sp_TopicsPerCourse

Input/Output

١	Name	Data type	Description
→ @ crs_id	int	:	

Uses

	Name	
sp_TopicsPerCourse		
⊞ Crs_Top		

```
CREATE PROC sp_TopicsPerCourse @crs_id INT as
SELECT dbo.Crs_Top.Crs_Top AS Topics FROM dbo.Crs_Top WHERE crs_id =@crs_id
```



2.45. Procedure: sp_updateChoice

Input/Output

	Name	Data type	Description
→ @	cho_id	int	
→ @	ques_id	int	
→ @	cho_content	varchar(200)	
→ @	cho_char	varchar(1)	

Uses

	Name	
sp_updateChoice		
E Choice		



2.46. Procedure: sp_updateCourse

Input/Output

	Name	Data type	Description
→ @	crs_id	int	
→ @	crs_name	varchar(50)	
→ @	crs_desc	varchar(50)	
→ @	crs_dur	int	

Uses

	Name	
sp_updateCourse		
E Course		



2.47. Procedure: sp_UpdateCrsForIns

Input/Output

	Name	Data type	Description
→ @	ins_id	int	
→ @	old_crs_id	int	
→@	new_crs_id	int	

Uses

	Name
sp_UpdateCrsForIns	
Ins_Crs	

```
CREATE PROC sp_UpdateCrsForIns

@ins_id int,
@old_crs id int,
@new_crs_id int,
@new_crs_id int

AS

BEGIN TRY

UPDATE Ins_Crs
SET
Crs_Id = @new_crs_id
WHERE Crs_Id = @ins_id
AND Crs_Id = @old_crs_id
END TRY
BEGIN CATCH
SELECT
ERROR_STATE() AS ErrorState,
ERROR_MESSAGE() AS ErrorMessag
END CATCH
```

2.48. Procedure: sp_UpdateCrsOfStd

Input/Output

	Name	Data type	Description
→ @	std_id	int	
→ @	old_crs_id	int	
→ @	new_crs_id	int	
→ @	grade	int	
→ @	date	date	

Uses

	Name
sp_UpdateCrsOfStd	
⊞ Std_Crs	

```
Script

CREATE PROC sp_UpdateCrsofStd

@std_id int,
@old_crs_id int,
@new ors_id,
int,
@grade int = NULL,
@date date = NULL

BEGIN TRY

UPDATE Std_Crs
SET

Crs_Id = @new_crs_id,
Grade = @grade,
[Date] = @date
WHERE Std_Id = @std_id
AND Crs_Id = @old_crs_id

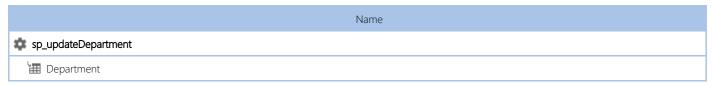
END TRY
BEGIN CATCH
SSLECT
ERROR_STATE() AS ErrorState,
ERROR_MESSAGE() AS ErrorMessag
END CATCH
```

2.49. Procedure: sp_updateDepartment

Input/Output

	Name	Data type	Description
→ @	dept_id	int	
→ @	dept_name	varchar(50)	
→ @	dept_loc	varchar(50)	
→ @	dept_managerHireDate	date	
→ @	dept_managerld	int	

Uses



```
Script
CREATE PROC sp_updateDepartment
@dept_id INT,
@dept_name VARCHAR(50),
@dept_loc VARCHAR(50),
@dept_managerHireDate DATE,
        @dept_managerId INT
BEGIN TRY
        UPDATE dbo.Department
       SET Dept_Name = @dept_name,
Dept_Loc = @dept_loc,
Dept_ManagerHireDate = @dept_managerHireDate,
Dept_ManagerId = @dept_managerId
WHERE Dept_Id = @dept_id;
END TRY
BEGIN CATCH
       SELECT ERROR_STATE() AS ErrorState,
ERROR_MESSAGE() AS ErrorMessag;
END CATCH;
```

2.50. Procedure: sp_updateExam

Input/Output

	Name	Data type	Description
→ @	exam_id	int	
→ @	crs_id	int	
→@	generator_id	int	
→ @	exam_grade	int	

Uses

	Name
sp_updateExam	
Exam	

```
CREATE PROC sp_updateExam @exam_id INT ,@crs_id INT ,@generator_id INT ,@exam_grade INT

AS
BEGIN TRY

UPDATE dbo.Exam SET Crs_Id =@crs_id,Generator_Id =@generator_id,Exm_Grade=@exam_grade

WHERE Exm_Id =@exam_id;

END TRY

BEGIN CATCH

SELECT ERROR_STATE() AS ErrorState,

ERROR_MESSAGE() AS ErrorMessag;

END CATCH
```

2.51. Procedure: sp_updateTopic

Input/Output

	Name	Data type	Description
→ @	crs_id	int	
→ @	crs_top	varchar(50)	

Uses

	Name	
sp_updateTopic		
☐ Crs_Top		

```
CREATE PROC sp_updateTopic
@crs_id INT,
@crs_top VARCHAR(50)

AS
BEGIN TRY

UPDATE dbo.Crs_Top
SET Crs_Top = @crs_top
WHERE Crs_Id = @crs_id
AND Crs_Top = @crs_top;

EEND TRY

BEGIN CATCH
SELECT ERROR_STATE() AS ErrorState,
ERROR_MESSAGE() AS ErrorMessag;

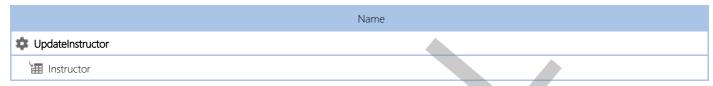
END CATCH;
```

2.52. Procedure: UpdateInstructor

Input/Output

	Name	Data type	Description
→ @	ins_id	int	
→ @	u_id	int	
→ @	dept_id	int	
→ @	fname	varchar(50)	
→ @	Iname	varchar(50)	
→ @	deg	varchar(50)	
→ @	sal	int	

Uses



Script

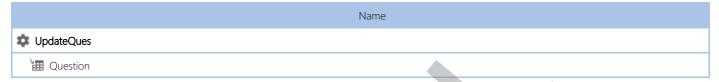
CREATE PROC UpdateInstructor

2.53. Procedure: UpdateQues

Input/Output

	Name	Data type	Description
→ @	ques_id	int	
→ @	crs_id	int	
→ @	cont	varchar(200)	
→ @	grade	int	
→ @	typ	varchar(3)	
→ @	ans	int	

Uses



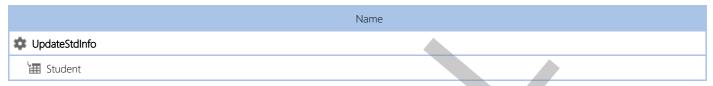
```
CREATE PROC UpdateQues
                                                                                                                                                         @ques_id int,
@crs_id int,
@cont varchar(200),
@grade int,
@typ varchar(3),
@ans_int
                         BEGIN TRY
                                                   UPDATE Question
                                                  Crs_Id = @crs_id,
    Ques_Content = @cont,
    Ques_Grade = @grade,
    Ques_Type = @typ,
    Model_Ans = @ans
WHERE Ques_Id = @ques_id
                                                   SET
                         END TRY
                         BEGIN CATCH
             SELECT
ERROR STATE () AS ErrorState,
ERROR MESSAGE () AS ErrorMessag
END CATCH
```

2.54. Procedure: UpdateStdInfo

Input/Output

	Name	Data type	Description
→ @	std_id	int	
→ @	us_id	int	
→ @	dept_id	int	
→ @	fname	varchar(50)	
→ @	Iname	varchar(50)	
→ @	bod	date	
→ @	address	varchar(50)	

Uses



```
CREATE PROC UpdateStdInfo

@std_id int,
@us_id int,
@dept_id int,
@fname varchar(50),
@lname varchar(50),
@bod date,
@address varchar(50)

Dept_Id = @us_id,
Dept_Id = @dept_id,
Std_Fname = @fname,
Std_Lname = @lname,
Std_Lname = @lname,
Std_BOD = @bod,
Std_BOD = @bod,
Std_Address = @address

WHERE Std_Id = @std_id

END TRY
BEGIN CATCH

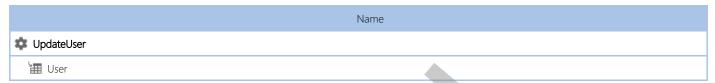
SELECT
ERROR_STATE() AS ErrorState,
ERROR_MESSAGE() AS ErrorMessag
END CATCH
```

2.55. Procedure: UpdateUser

Input/Output

	Name	Data type	Description
→ @	us_id	int	
→ @	us_name	varchar(50)	
→ @	us_mail	varchar(50)	
→ @	us_pass	varchar(50)	
→ @	us_sex	varchar(1)	
→@	us_isStd	bit	

Uses



```
CREATE PROC [dbo].[UpdateUser]

@us_id int,
@us_name varchar(50),
@us_mail varchar(50),
@us_pass varchar(50),
@us_pass varchar(50),
@us_sex varchar(50),
@us_mail var
```

