

none

ExaminationSystem@.

Data Dictionary

2/7/2021

TRIAL

TRIAL

## Table of contents

ExaminationSystem@ .....	9
1. Tables .....	9
1.1. Table: adm.Users .....	9
1.2. Table: ex.Exam .....	10
1.3. Table: ex.exam_answers .....	11
1.4. Table: ex.Exam_Questions .....	12
1.5. Table: ex.MCQ_Choices .....	13
1.6. Table: ex.Questions .....	14
1.7. Table: ex.St_Takes_Exam .....	15
1.8. Table: uni.Course .....	16
1.9. Table: uni.department .....	17
1.10. Table: uni.Ins_Crs .....	18
1.11. Table: uni.ins_works_dept .....	19
1.12. Table: uni.instructor .....	20
1.13. Table: uni.student .....	21
1.14. Table: uni.StudentCourse .....	22
1.15. Table: uni.Topic .....	23
2. Procedures .....	24
2.1. Procedure: adm.spUsersCheckUser .....	24
2.2. Procedure: dbo.spCourseDeleteRecordByld .....	25
2.3. Procedure: dbo.spDeleteStudentByld .....	26
2.4. Procedure: dbo.spDeleteStudentByName .....	27
2.5. Procedure: dbo.spGetAllStudents .....	28
2.6. Procedure: dbo.spGetStudentByld .....	29
2.7. Procedure: dbo.spGetStudentByName .....	30
2.8. Procedure: dbo.spGetStudentsInDepartment .....	31
2.9. Procedure: dbo.spInsertStudent .....	32
2.10. Procedure: dbo.spUpdateStudentDepartment .....	33
2.11. Procedure: dbo.spUpdateStudentsAge .....	34
2.12. Procedure: ex.spDeleteExamByld .....	35
2.13. Procedure: ex.spDeleteMCQQuestionByld .....	36
2.14. Procedure: ex.spDeleteQuestionByld .....	37
2.15. Procedure: ex.spExam_QuestionsDeleteAllData .....	38
2.16. Procedure: ex.spExam_QuestionsDeleteExamByQuestionId .....	39
2.17. Procedure: ex.spExam_QuestionsDeleteQuestionbyExamId .....	40
2.18. Procedure: ex.spExam_QuestionsDeleteQuestionIdByQuestionNumber .....	41
2.19. Procedure: ex.spExam_QuestionsDeleteQuestionNumberByQuestionId .....	42
2.20. Procedure: ex.spExam_QuestionsGetAll .....	43
2.21. Procedure: ex.spExam_QuestionsGetInfoAboutSpecficExam .....	44
2.22. Procedure: ex.spExam_QuestionsGetinfoaboutspeficQuestionNumber .....	45
2.23. Procedure: ex.spExam_QuestionsGetinfoaboutspeficQuestions .....	46
2.24. Procedure: ex.spExam_QuestionsInserNewQuestion .....	47
2.25. Procedure: ex.spExam_QuestionsUpdateExamIdReplaceltByAnthorOne .....	48
2.26. Procedure: ex.spExam_QuestionsUpdateQuestionIdInExam .....	49
2.27. Procedure: ex.spExam_QuestionsUpdateQuestionIdReplaceltByAnthorOne .....	50
























2.28.	Procedure: ex.spExam_QuestionsUpdateQuestionNumberInExam .....	51
2.29.	Procedure: ex.spExam_QuestionsUpdateQuestionNumberReplaceltByAnthorOne .....	52
2.30.	Procedure: ex.spExamAnswerDeleteExamByld .....	53
2.31.	Procedure: ex.spExamAnswerDeleteQuestion .....	54
2.32.	Procedure: ex.spExamAnswerInserNewExam .....	55
2.33.	Procedure: ex.spExamAnswerInserNewExamAnswerByTable .....	56
2.34.	Procedure: ex.spExamAnswersGetall .....	57
2.35.	Procedure: ex.spExamAnswersGetByld .....	58
2.36.	Procedure: ex.spExamAnswersGetSpecificAnswer .....	59
2.37.	Procedure: ex.spExamAnswerUpdateSpecificAnswer .....	60
2.38.	Procedure: ex.spExamAnswerUpdateSpecificExam .....	61
2.39.	Procedure: ex.spExamAnswerUpdateSpecificNumQuestion .....	62
2.40.	Procedure: ex.spExamCorrection .....	63
2.41.	Procedure: ex.spGenerateExam .....	64
2.42.	Procedure: ex.spGetAllExamsData .....	65
2.43.	Procedure: ex.spGetAllMCQQuestionsData .....	66
2.44.	Procedure: ex.spGetExambyld .....	67
2.45.	Procedure: ex.spGetMCQQuestionByld .....	68
2.46.	Procedure: ex.spGetQuestionsByCourseId .....	69
2.47.	Procedure: ex.spInsertExam .....	70
2.48.	Procedure: ex.spInsertMCQuestion .....	71
2.49.	Procedure: ex.spInsertQuestion .....	72
2.50.	Procedure: ex.spRetriveTheExamQuestionsStudentAnswers .....	73
2.51.	Procedure: ex.spStudentTakeExamDeleteCourseRecordsByCourseId .....	74
2.52.	Procedure: ex.spStudentTakeExamDeleteRecordByExamId .....	75
2.53.	Procedure: ex.spStudentTakeExamGetAll .....	76
2.54.	Procedure: ex.spStudentTakeExamGetAllByCourseId .....	77
2.55.	Procedure: ex.spStudentTakeExamGetAllByExamId .....	78
2.56.	Procedure: ex.spStudentTakeExamGetAllByGrade .....	79
2.57.	Procedure: ex.spStudentTakeExamGetAllByStudentId .....	80
2.58.	Procedure: ex.spStudentTakeExamInsertRecord .....	81
2.59.	Procedure: ex.spStudentTakeExamUpdateRecordByExamId .....	82
2.60.	Procedure: ex.spUpdateExamTypeByld .....	83
2.61.	Procedure: ex.spUpdateMCQuestionDataByld .....	84
2.62.	Procedure: ex.spUpdateQuestionAnswerByld .....	85
2.63.	Procedure: ex.spUpdateQuestionMarksByld .....	86
2.64.	Procedure: ex.spUpdateQuestionStateByld .....	87
2.65.	Procedure: uni.spCourseDeleteRecordByName .....	88
2.66.	Procedure: uni.spCourseGetAll .....	89
2.67.	Procedure: uni.spCourseGetAllByCrslld .....	90
2.68.	Procedure: uni.spCourseGetAllByCrslName .....	91
2.69.	Procedure: uni.spCourseGetAllByTopicId .....	92
2.70.	Procedure: uni.spCourseGetAllByTopicName .....	93
2.71.	Procedure: uni.spCourseInsertRecord .....	94
2.72.	Procedure: uni.spCourseInsertRecordWithTopicId .....	95
2.73.	Procedure: uni.spCourseInsertRecordWithTopicName .....	96

2.74.	Procedure: uni.spCourseUpdateRecordWithCourseId	97
2.75.	Procedure: uni.spCourseUpdateRecordWithCourseName	98
2.76.	Procedure: uni.spDeleteInstructorByFullName	99
2.77.	Procedure: uni.spDeleteInstructorById	100
2.78.	Procedure: uni.spDeleteInstructorCourse	101
2.79.	Procedure: uni.spDepartmentDeleteById	102
2.80.	Procedure: uni.spDepartmentDeleteByName	103
2.81.	Procedure: uni.spDepartmentGetAll	104
2.82.	Procedure: uni.spDepartmentGetById	105
2.83.	Procedure: uni.spDepartmentInsert	106
2.84.	Procedure: uni.spDepartmentUpdateManagerHireDate	107
2.85.	Procedure: uni.spDepartmentUpdateManagerID	108
2.86.	Procedure: uni.spDepartmentUpdateName	109
2.87.	Procedure: uni.spGetAllInstructorsData	110
2.88.	Procedure: uni.spGetDepartmentById	111
2.89.	Procedure: uni.spGetDeptNameById	112
2.90.	Procedure: uni.spGetInsNameByDeptId	113
2.91.	Procedure: uni.spGetInstructorByDepartmentId	114
2.92.	Procedure: uni.spGetInstructorById	115
2.93.	Procedure: uni.spIns_works_deptDeleteAllData	116
2.94.	Procedure: uni.spIns_works_deptDeleteDeptIdById	117
2.95.	Procedure: uni.spIns_works_deptDeleteInsIdByDeptId	118
2.96.	Procedure: uni.spIns_works_deptGetAll	119
2.97.	Procedure: uni.spIns_works_deptSetValue	120
2.98.	Procedure: uni.spInsertInstructor	121
2.99.	Procedure: uni.spInsertInstructorCourse	122
2.100.	Procedure: uni.spReplaceInstructorToAnotherOne	123
2.101.	Procedure: uni.spSelectCoursesById	124
2.102.	Procedure: uni.spSelectInstructorsByCrsId	125
2.103.	Procedure: uni.spStudentCourseDeleteRecordByStudentIdAndCourseId	126
2.104.	Procedure: uni.spStudentCourseGetCourseAndGradeByStudentId	127
2.105.	Procedure: uni.spStudentCourseGetCourseAndGradeByStudentName	128
2.106.	Procedure: uni.spStudentCourseGetStudentAndGradeByCourseId	129
2.107.	Procedure: uni.spStudentCourseGetStudentAndGradeByCourseName	130
2.108.	Procedure: uni.spStudentCourseInsertRecord	131
2.109.	Procedure: uni.spStudentCourseUpdateRecordByStudentIdAndCourseId	132
2.110.	Procedure: uni.spTopicDeleteRecordById	133
2.111.	Procedure: uni.spTopicDeleteRecordByName	134
2.112.	Procedure: uni.spTopicGetAll	135
2.113.	Procedure: uni.spTopicGetAllByCourseId	136
2.114.	Procedure: uni.spTopicGetAllByCourseName	137
2.115.	Procedure: uni.spTopicGetTopicIdByTopicName	138
2.116.	Procedure: uni.spTopicGetTopicNameByCourseId	139
2.117.	Procedure: uni.spTopicGetTopicNameByCourseName	140
2.118.	Procedure: uni.spTopicGetTopicNames	141
2.119.	Procedure: uni.spTopicInsertRecord	142

2.120.	Procedure: uni.spTopicUpdateRecordWithTopicId .....	143
2.121.	Procedure: uni.spTopicUpdateRecordWithTopicName .....	144
2.122.	Procedure: uni.spUpdateCourseInstructor .....	145
2.123.	Procedure: uni.spUpdateDepartMentToMangeByNewOne .....	146
2.124.	Procedure: uni.spUpdateInstructorCourse .....	147
2.125.	Procedure: uni.spUpdateInstructorDataById .....	148
2.126.	Procedure: uni.spUpdateInstructorofDepartMent .....	149

TRIAL

## Legend




-  Primary key
-  Primary key disabled
-  User-defined primary key
-  Unique key
-  Unique key disabled
-  User-defined unique key
-  Active trigger
-  Disabled trigger
-  Many to one relation
-  User-defined many to one relation
-  One to many relation
-  User-defined one to many relation
-  Many to many relation
-  User-defined many to many relation
-  One to one relation
-  User-defined one to one relation
-  Input
-  Output
-  Input/Output
-  Uses dependency
-  User-defined uses dependency
-  Used by dependency
-  User-defined used by dependency



## 1. Tables

## 1.1. Table: adm.Users


## Columns

	Name	Data type	Description / Attributes
	UserId	int	
	userPass	int	Nullable
	userType	varchar(20)	Nullable

## Linked from



	Table	Join	Title / Name / Description
←	uni.instructor	adm.UsersUserId = uni.instructorins_id	fk_ins_userId
←	uni.student	adm.UsersUserId = uni.studentst_id	fk_st_userId

## Unique keys



	Columns	Name / Description
	UserId	PK_Users_1788CC4C1E289926

## 1.2. Table: ex.Exam

### Columns

Name		Data type	Description / Attributes
	Exam_id	int	Identity / Auto increment
	Exam_Type	varchar(20)	Nullable

### Linked from






Table	Join	Title / Name / Description
 ex.Exam_Questions	<b>ex.Exam</b> Exam_id = ex.Exam_QuestionsExam_id	fk_exam_id
 ex.St_Takes_Exam	<b>ex.Exam</b> Exam_id = ex.St_Takes_ExamExam_id	fk_exam_id01

### Unique keys


Columns	Name / Description
 Exam_id	PK__Exam__C79BD6716D448E7A

### 1.3. Table: ex.exam\_answers

#### Columns

Name		Data type	Description / Attributes
	 Exam_Id	int	<b>References:</b> ex.St_Takes_Exam
	 QuestionNum	int	
	QuestionAns	varchar(2)	<b>Nullable</b>

#### Links to






Table	Join	Title / Name / Description
 ex.St_Takes_Exam	<b>ex.exam_answers</b> Exam_Id = ex.St_Takes_ExamExam_id	fk_exam_id02

#### Unique keys



Columns	Name / Description
 Exam_Id, QuestionNum	PK_exam_ans_BB4DD90A5D52B1CC

## 1.4. Table: ex.Exam\_Questions

### Columns

Name		Data type	Description / Attributes
	 Ques_id	int	<b>References:</b> ex.Questions
	 Exam_id	int	<b>References:</b> ex.Exam
	QuestionNumberInExam	bigint	<b>Default:</b> NEXT VALUE FOR [CountTo10]

### Links to






Table		Join	Title / Name / Description
	ex.Exam	<b>ex.Exam_Questions</b> Exam_id = ex.ExamExam_id	fk_exam_id
	ex.Questions	<b>ex.Exam_Questions</b> Ques_id = ex.QuestionsQues_ID	fk_Ques_id

### Unique keys


Columns		Name / Description
	Ques_id, Exam_id	PK_Exam_Que_A4599A413AC1CC86

## 1.5. Table: ex.MCQ\_Choices


### Columns

Name		Data type	Description / Attributes
	 ques_id	int	<b>References:</b> ex.Questions
	 lable	varchar(1)	
	text	varchar(50)	<b>Nullable</b>

### Links to








Table	Join	Title / Name / Description
 ex.Questions	<b>ex.MCQ_Choices</b> ques_id = ex.QuestionsQues_ID	fk_MCQ_Choices_id

### Unique keys


Columns	Name / Description
 ques_id, lable	PK_MCQ_Choi__BF7654DE1C4C6587

## 1.6. Table: ex.Questions



### Columns

	Name	Data type	Description / Attributes
	Ques_ID	int	Identity / Auto increment
	Model_Answer	varchar(2)	
	Marks	int	
	Type	varchar(25)	
	body	varchar(200)	
	CrsId	int	References: uni.Course
	qState	bit	Nullable Default: 1

### Links to

Table	Join	Title / Name / Description
 uni.Course	<b>ex.Questions</b> CrsId = uni.CourseCrs_ID	fk_crs_ques

### Linked from







Table	Join	Title / Name / Description
 ex.Exam_Questions	<b>ex.Questions</b> Ques_ID = ex.Exam_QuestionsQues_id	fk_Ques_id
 ex.MCQ_Choices	<b>ex.Questions</b> Ques_ID = ex.MCQ_Choicesques_id	fk_MCQ_Choices_id

### Unique keys




Columns	Name / Description
 Ques_ID	PK__Question__A82122BEF9C314FE

## 1.7. Table: ex.St\_Takes\_Exam

### Columns

	Name	Data type	Description / Attributes
	st_id	int	<b>References:</b> uni.student
 	Exam_id	int	<b>References:</b> ex.Exam
	Crs_Id	int	<b>References:</b> uni.Course
	Exam_Date	date	<b>Nullable</b>
	Grade	int	<b>Nullable</b>


### Links to

	Table	Join	Title / Name / Description
	uni.Course	<b>ex.St_Takes_Exam</b> Crs_Id = uni.CourseCrs_ID	fk_crs_id
	ex.Exam	<b>ex.St_Takes_Exam</b> Exam_id = ex.ExamExam_id	fk_exam_id01
	uni.student	<b>ex.St_Takes_Exam</b> st_id = uni.studentst_id	fk_st_id

### Linked from






	Table	Join	Title / Name / Description
	ex.exam_answers	<b>ex.St_Takes_Exam</b> Exam_id = ex.exam_answersExam_Id	fk_exam_id02

### Unique keys


	Columns	Name / Description
	Exam_id	PK__St_Takes__C79BD67120636138

## 1.8. Table: uni.Course





### Columns

Name		Data type	Description / Attributes
	 Crs_ID	int	Identity / Auto increment
	Crs_Name	nvarchar(25)	
	Crs_Duration	int	
	Topic_ID	int	References: uni.Topic


### Links to

Table	Join	Title / Name / Description
 uni.Topic	<b>uni.Course</b> Topic_ID = uni.TopicTopic_ID	ins_topic

### Linked from

Table	Join	Title / Name / Description
 uni.Ins_Crs	<b>uni.Course</b> Crs_ID = uni.Ins_CrsCrS_ID	fk2_Ins_Crs_id
 ex.Questions	<b>uni.Course</b> Crs_ID = ex.QuestionsCrSId	fk_crs_ques
 ex.St_Takes_Exam	<b>uni.Course</b> Crs_ID = ex.St_Takes_ExamCrS_Id	fk_crs_id
 uni.StudentCourse	<b>uni.Course</b> Crs_ID = uni.StudentCoursecrs_id	fk_crs







### Unique keys

Columns	Name / Description
 Crs_ID	PK__Course__56CAA5F559C50A22




## 1.9. Table: uni.department



### Columns

Name		Data type	Description / Attributes
	Dept_ID	int	Identity / Auto increment
	DeptName	varchar(20)	
	Dept_Desc	varchar(300)	
	Dept_location	varchar(50)	Nullable
	mgr_id	int	References: uni.instructor
	mgr_HireDate	date	Nullable

### Links to

Table	Join	Title / Name / Description
 uni.instructor	<b>uni.department</b> mgr_id = uni.instructor ins_id	fk_ins_dept02

### Linked from





Table	Join	Title / Name / Description
 uni.ins_works_dept	<b>uni.department</b> Dept_ID = uni.ins_works_dept dept_id	fk_dept_id04
 uni.student	<b>uni.department</b> Dept_ID = uni.student dept_id	dept_fk

### Unique keys

Columns	Name / Description
 Dept_ID	PK_department__72ABC12CD75B25C2

## 1.10. Table: uni.Ins\_Crs

### Columns

		Name	Data type	Description / Attributes
		ins_id	int	<b>References:</b> uni.instructor
		Crs_ID	int	<b>References:</b> uni.Course

### Links to





	Table	Join	Title / Name / Description
➤	uni.Course	<b>uni.Ins_Crs</b> Crs_ID = uni.CourseCrs_ID	fk2_Ins_Crs_id
➤	uni.instructor	<b>uni.Ins_Crs</b> ins_id = uni.instructorins_id	fk_Ins_Crs_id

### Unique keys



Columns		Name / Description
	ins_id, Crs_ID	PK_Ins_Crs_D9DB877F68782F79

## 1.11. Table: uni.ins\_works\_dept

### Columns

Name		Data type	Description / Attributes
	 dept_id	int	<b>References:</b> uni.department
	 ins_id	int	<b>References:</b> uni.instructor

### Links to








Table	Join	Title / Name / Description
 uni.department	<b>uni.ins_works_dept</b> dept_id = uni.departmentDept_ID	fk_dept_id04
 uni.instructor	<b>uni.ins_works_dept</b> ins_id = uni.instructorins_id	fk_ins_id03

### Unique keys


Columns	Name / Description
 dept_id, ins_id	PK__ins_work__C56D2BA680289C9A

## 1.12. Table: uni.instructor




### Columns

	Name	Data type	Description / Attributes
	ins_id	int	<b>Default:</b> NEXT VALUE FOR [UserIdSeq] <b>References:</b> adm.Users
	ins_FName	varchar(25)	
	ins_LName	varchar(25)	
	ins_age	int	
	ins_salary	decimal(18, 2)	<b>Nullable</b>
	ins_hireDate	date	<b>Nullable</b>
	ins_gender	varchar(1)	

### Links to

Table	Join	Title / Name / Description
 adm.Users	<b>uni.instructor</b> ins_id = adm.UsersUserId	fk_ins_userid


### Linked from

Table	Join	Title / Name / Description
 uni.department	<b>uni.instructor</b> ins_id = uni.departmentmgr_id	fk_ins_dept02
 uni.Ins_Crs	<b>uni.instructor</b> ins_id = uni.Ins_Crsins_id	fk_Inc_Crs_id
 uni.ins_works_dept	<b>uni.instructor</b> ins_id = uni.ins_works_depts_id	fk_ins_id03

### Unique keys







Columns	Name / Description
 ins_id	PK__instruct__9CB72D20C239E3F9

### Triggers



Name	When	Description
 AddInsToUsers	Instead Of Insert	

### 1.13. Table: uni.student



#### Columns

	Name	Data type	Description / Attributes
	st_id	int	<b>Default:</b> NEXT VALUE FOR [UserIdSeq] <b>References:</b> adm.Users
	st_fname	varchar(20)	
	st_lname	varchar(20)	<b>Nullable</b>
	st_age	int	
	st_gender	varchar(1)	
	dept_id	int	<b>Nullable</b> <b>References:</b> uni.department


#### Links to

Table	Join	Title / Name / Description
 uni.department	<b>uni.student</b> dept_id = uni.departmentDept_ID	dept_fk
 adm.Users	<b>uni.student</b> st_id = adm.UsersUserId	fk_st_userid

#### Linked from

Table	Join	Title / Name / Description
 ex.St_Takes_Exam	<b>uni.student</b> st_id = ex.St_Takes_Examst_id	fk_st_id
 uni.StudentCourse	<b>uni.student</b> st_id = uni.StudentCoursest_id	fk_st

#### Unique keys






Columns	Name / Description
 st_id	PK_student_A85E81CF2976193C

#### Triggers



Name	When	Description
 AddStudentToUsers	Instead Of Insert	

## 1.14. Table: uni.StudentCourse


### Columns

Name		Data type	Description / Attributes
	 st_id	int	<b>References:</b> uni.student
	 crs_id	int	<b>References:</b> uni.Course
	FinalGrade	int	<b>Nullable</b>

### Links to




Table	Join	Title / Name / Description
 uni.Course	<b>uni.StudentCourse</b> crs_id = uni.CourseCrs_ID	fk_crs
 uni.student	<b>uni.StudentCourse</b> st_id = uni.studentst_id	fk_st

### Unique keys

Columns	Name / Description
 st_id, crs_id	PK_StudentC__E69474F8E8837FDB

## 1.15. Table: uni.Topic


### Columns

		Name	Data type	Description / Attributes
		Topic_ID	int	Identity / Auto increment
		Topic_Name	nvarchar(25)	

### Linked from

		Table	Join	Title / Name / Description
		uni.Course	<b>uni.Topic</b> Topic_ID = uni.CourseTopic_ID	ins_topic

### Unique keys

		Columns	Name / Description
		Topic_ID	PK__Topic__8DEAA42524D1E929

## 2. Procedures

### 2.1. Procedure: adm.spUsersCheckUser

-- Author: Shawky  
-- Description: Check if user exists in users files  
-- Input : Exam ID

#### Input/Output

	Name	Data type	Description
→@	UserId	int	
→@	UserPass	int	

TRIAL



## 2.2. Procedure: dbo.spCourseDeleteRecordById

```
-- Author: Abdallh
-- Description: Delete Record By CourseID From Course Table
-- Input : Course ID
-- Returns: 0 -> Success
-- 1 -> Course ID NULL
-- 2 -> Course ID Doesn't Exist
-- 3 -> failure
```

### Input/Output

Name		Data type	Description
→@	crs_id	int	

### 2.3. Procedure: dbo.spDeleteStudentById

```
-- Author:      Mohamed Shawky
-- Create date: 20/1/2020 - My BirthDate :)
-- Description: Delete Student by ID
-- Parameters:
-- @st_id
```

#### Input/Output

Name		Data type	Description
→@	StudentId	int	

TRIAL

## 2.4. Procedure: dbo.spDeleteStudentByName

```
-- Author:      Mohamed Shawky
-- Create date: 20/1/2020 - My BirthDate :)
-- Description: Delete Student by Name
-- Parameters:
-- @st_FullName
```

### Input/Output

Name		Data type	Description
→@	FullName	varchar(50)	

TRIAL

## 2.5. Procedure: dbo.spGetAllStudents

```
-- Author:      Mohamed Shawky
-- Create date: 20/1/2020 - My BirthDate :)
--version update 22/1/2020 fdfdf
-- Description: Get all students in All departments
-- Parameters: NON
```

TRIAL

## 2.6. Procedure: dbo.spGetStudentById

```
-- Author:      Mohamed Shawky
-- Create date: 20/1/2020 - My BirthDate :)
-- Description: Get a student Information knowing his ID
-- Parameters:
--   @Student_id
```

### Input/Output

Name		Data type	Description
→@	StudentId	int	

## 2.7. Procedure: dbo.spGetStudentByName

```
-- Author:      Mohamed Shawky
-- Create date: 20/1/2020 - My BirthDate :)
-- Description: Get a student Information knowing his Name
-- Parameters:
--   @Student_id
```

### Input/Output

	Name	Data type	Description
→@	StudentFullName	varchar(60)	

## 2.8. Procedure: dbo.spGetStudentsInDepartment

```
-- Author:      Mohamed Shawky
-- Create date: 20/1/2020 - My BirthDate :)
-- Description: Get all students in specific department
-- Parameters:
--   @Department_id
```

### Input/Output

Name		Data type	Description
→@	DepartmentId	int	

## 2.9. Procedure: dbo.spInsertStudent

```
-- Author:      Mohamed Shawky
-- Create date: 20/1/2020 - My BirthDate :)
-- Description: Insert New Student
-- Parameters:
-- @st_fname
-- @st_lname Accepts Null
-- @st_age
-- @stGender
-- @Dept_id Accepts Null
```

### Input/Output

	Name	Data type	Description
→@	stfname	varchar(20)	
→@	stlname	varchar(20)	
→@	stAge	int	
→@	stGender	varchar(1)	
→@	StDept	int	



## 2.10. Procedure: dbo.spUpdateStudentDepartment

```
-- Author:      Mohamed Shawky
-- Create date: 20/1/2020 - My BirthDate :)
-- Description: Updates Department Id for a student
-- Parameters:
--   @Student_id
--   @Department_id
```

### Input/Output

Name		Data type	Description
→@	StudentId	int	
→@	DeptID	int	

TRIAL

## 2.11. Procedure: dbo.spUpdateStudentsAge

```
-- Author:      Mohamed Shawky  
-- Create date: 20/1/2020 - My BirthDate :)  
-- Description: update All Students Ages, this run every year  
-- Parameters:  NON
```

TRIAL

## 2.12. Procedure: ex.spDeleteExamById

```
-- Author: Ahmed Nasr  
-- CREATE date: 25/1/2020  
-- Description: Delete Record By Exam_Id From Exam Table  
-- Parameters: Exam ID
```

### Input/Output

Name		Data type	Description
→@	Exam_id	int	

TRIAL

### 2.13. Procedure: ex.spDeleteMCQQuestionById

```
-- Author: Ahmed Nasr
-- Create date: 25/1/2020
-- Description: Delete Record By Question_Id From MCQ_Choices Table
-- Parameters: Question ID
```

#### Input/Output

Name		Data type	Description
→@	Question_id	int	

## 2.14. Procedure: ex.spDeleteQuestionById

```
-- Author:      Mohamed Shawky
-- Create date: 23/1/2020
-- Description: Delete Question ..change it's state to be not active
-- Parameters:
-- @QuestionID
```

### Input/Output

Name		Data type	Description
→@	QuestionId	int	

TRIAL

## 2.15. Procedure: ex.spExam\_QuestionsDeleteAllData

```
-- Author:      lojiun Elsayed  
-- Create date: 22/1/2020  
-- Description: Delete All Data in Exam_Questions  
-- Parameters: NON
```

TRIAL

## 2.16. Procedure: ex.spExam\_QuestionsDeleteExamByQuestionId

```
-- Author:      lojiun Elsayed
-- Create date: 22/1/2020
-- Description: Delete specific Exam by QuestionId
-- Parameters:
--QuestionId
```

### Input/Output

Name		Data type	Description
→@	QuestionId	int	

TRIAL

## 2.17. Procedure: ex.spExam\_QuestionsDeleteQuestionbyExamId

```
-- Author:      lojiun Elsayed
-- Create date: 22/1/2020
-- Description: Delete specific Question  in Specfic Exam
-- Parameters:
--@ExamId
```

### Input/Output

Name		Data type	Description
→@	ExamId	int	

TRIAL



## 2.18. Procedure: ex.spExam\_QuestionsDeleteQuestionIdByQuestionNumber

```
-- Author:      lojiun Elsayed
-- Create date: 22/1/2020
-- Description: Delete QuestionId  by questionNum
-- Parameters:
--@questionNumber
```

### Input/Output

	Name	Data type	Description
→@	questionNumber	int	

TRIAL

## 2.19. Procedure: ex.spExam\_QuestionsDeleteQuestionNumberByQuestionId

```
-- Author:      lojiun Elsayed
-- Create date: 22/1/2020
-- Description: Delete QuestionNumber by questionId
-- Parameters:
--@questionId
```

### Input/Output

Name		Data type	Description
→@	questionId	int	

TRIAL

## 2.20. Procedure: ex.spExam\_QuestionsGetAll

```
-- Author:      lojiun Elsayed  
-- Create date: 24/1/2020  
-- Description: get All Data in Exam_Questions  
-- Parameters: NON
```

TRIAL

## 2.21. Procedure: ex.spExam\_QuestionsGetInfoAboutSpecficExam

```
-- Author:      lojiun Elsayed
-- Create date: 24/1/2020
-- Description: Get info about specific Exam
-- Parameters:
-- @Exam_Id
```

### Input/Output

	Name	Data type	Description
→@	Exam_Id	int	

TRIAL

## 2.22. Procedure: ex.spExam\_QuestionsGetinfoaboutspeficQuestionNumber

```
-- Author:      lojiun Elsayed
-- Create date: 24/1/2020
-- Description: Get info about specific Question
-- Parameters:
-- @question_Number
```

### Input/Output

Name		Data type	Description
→@	question_Number	int	

TRIAL

## 2.23. Procedure: ex.spExam\_QuestionsGetinfoaboutspeficQuestions

```
-- Author:      lojiun Elsayed
-- Create date: 24/1/2020
-- Description: Get info about specific Question
-- Parameters:
-- @question_Id
```

### Input/Output

Name		Data type	Description
→@	question_Id	int	

TRIAL

## 2.24. Procedure: ex.spExam\_QuestionsInserNewQuestion

```
-- Author:      lojiun
-- Create date: 22/1/2020
-- Description: Insert New question
-- Parameters:
--@questionm_Id INT
--@examId int Null
--@QuestionNumber int Null
```

### Input/Output

	Name	Data type	Description
→@	questionm_Id	int	
→@	examId	int	
→@	QuestionNumber	int	

TRIAL

## 2.25. Procedure: ex.spExam\_QuestionsUpdateExamIdReplaceltByAnthorOne

```
-- Author:      lojiun elsayed
-- Create date: 24/1/2020
-- Description: Replace Exam id  by anthor one
-- Parameters:
--@OldExamId
--@NewExamId
```

### Input/Output

Name		Data type	Description
→@	OldExamId	int	
→@	NewExamId	int	

TRIAL



## 2.26. Procedure: ex.spExam\_QuestionsUpdateQuestionIDInExam

```
-- Author:      lojiun elsayed
-- Create date: 24/1/2020
-- Description: Updates QuestionID in Specific Exam
-- Parameters:
-- @Exam_Id
-- @OldQuestionNumber
-- @NewQuestionNumber
```

### Input/Output

Name		Data type	Description
→@	Exam_Id	int	
→@	OldQuestionID	int	
→@	NewQuestionID	int	

TRIAL

## 2.27. Procedure: ex.spExam\_QuestionsUpdateQuestionIdReplaceltByAnthorOne

```
--Author:      lojiun elsayed
-- Create date: 24/1/2020
-- Description: Replace QuestionID  by anthor one
-- Parameters:
--@OldQuestionNumber
--@NewQuestionNumber
```

### Input/Output

Name		Data type	Description
→@	OldQuestionID	int	
→@	NewQuestionID	int	

## 2.28. Procedure: ex.spExam\_QuestionsUpdateQuestionNumberInExam

### Input/Output

Name		Data type	Description
→@	Exam_Id	int	
→@	OldQuestionNumber	int	
→@	NewQuestionNumber	int	

## 2.29. Procedure: ex.spExam\_QuestionsUpdateQuestionNumberReplaceltByAnthorOne

```
-- Author:      lojiun elsayed
-- Create date: 24/1/2020
-- Description: Replace QuestionNumber by anthor one
-- Parameters:
--@OldQuestionNumber
--@NewQuestionNumber
```

### Input/Output

Name		Data type	Description
→@	OldQuestionNumber	int	
→@	NewQuestionNumber	int	

TRIAL

### 2.30. Procedure: ex.spExamAnswerDeleteExamById

```
-- Author:      Moaz ayman
-- Create date: 22/1/2020
-- Description: Delete Exam by ID
-- Parameters:
-- @Exam_Id
```

#### Input/Output

Name		Data type	Description
→@	Exam_Id	int	

### 2.31. Procedure: ex.spExamAnswerDeleteQuestion

```
-- Author:      Moaz ayman
-- Create date: 22/1/2020
-- Description: Delete specific Question in Eaxm and Itis Answer
-- Parameters:
--  @Exam_Id INT,@QuestionNum int
```

#### Input/Output

Name		Data type	Description
→@	Exam_Id	int	
→@	QuestionNum	int	

TRIAL

## 2.32. Procedure: ex.spExamAnswerInserNewExam

```
-- Author:      Moaz ayman
-- Create date: 22/1/2020
-- Description: Insert New Exam
-- Parameters:
--@Exam_Id INT
--@QuestionNum int
--@QuestionAns int
```

### Input/Output

Name		Data type	Description
→@	Exam_Id	int	
→@	QuestionNum	int	
→@	QuestionAns	varchar(2)	

### 2.33. Procedure: ex.spExamAnswerInserNewExamAnswerByTable

Author: Moaz

Purpose: Takes Table with answers values and inserts them into the database in one go.

Procedure which accepts one table valued parameter

Input Table Table

```
(
--This type has structure similar to the DB table
    [Exam_Id] int Not Null
    ,[QuestionNum] int Not Null
    ,[QuestionAns] varchar(10) Not Null -- Having one String
)
```

Input/Output

Name		Data type	Description
→@	Sample	table type	



## 2.34. Procedure: ex.spExamAnswersGetall

```
-- Author:      Moaz ayman  
-- Create date: 22/1/2020  
-- Description: Get all information about All Exams'Answers  
-- Parameters:  NON
```

TRIAL

### 2.35. Procedure: ex.spExamAnswersGetById

```
-- Author:      Moaz ayman
-- Create date: 22/1/2020
-- Description:  Get all information about specific Exam
-- Parameters:
--      @Exam_Id
```

#### Input/Output

Name		Data type	Description
→@	Exam_Id	int	

TRIAL

## 2.36. Procedure: ex.spExamAnswersGetSpecificAnswer

```
-- Author:      Moaz ayman
-- Create date: 22/1/2020
-- Description:  Get Answer about specific Question in Eaxm
-- Parameters:
--      @Exam_Id,@QuestionNum
```

### Input/Output

Name		Data type	Description
→@	Exam_Id	int	
→@	QuestionNum	int	

TRIAL

## 2.37. Procedure: ex.spExamAnswerUpdateSpecificAnswer

```
-- Author:      moaz ayman
-- Create date: 22/1/2020
-- Description: Updates Answer of Specific Question  in Specific Exam
-- @Exam_Id
--@QuestionNum
--@QuestionAns
```

### Input/Output

Name		Data type	Description
→@	Exam_Id	int	
→@	QuestionNum	int	
→@	QuestionAns	varchar(2)	

## 2.38. Procedure: ex.spExamAnswerUpdateSpecificExam

```
-- Author:      moaz ayman
-- Create date: 20/1/2020
-- Description: Updates Specific Exam By Anthon
-- Parameters:  @oldExam_Id INT,@NewExam_Id int
```

### Input/Output

Name		Data type	Description
→@	oldExam_Id	int	
→@	NewExam_Id	int	

TRIAL

## 2.39. Procedure: ex.spExamAnswerUpdateSpecificNumQuestion

```
-- Author:      moaz ayman
-- Create date: 22/1/2020
-- Description: Updates Number of Specific Question in Specific Exam
-- Parameters:
--   @Exam_Id
-- @OldQuestionNum
-- @NewQuestionNum
```

### Input/Output

Name		Data type	Description
→@	Exam_Id	int	
→@	OldQuestionNum	int	
→@	NewQuestionNum	int	

## 2.40. Procedure: ex.spExamCorrection

```
-- =====  
-- Author: Abdallh  
-- Description: Exam Correction, used to correct an exam and update the corresponding grades.  
-- Inputs: Exam Id  
-- Returns:  
-- 0-> Success  
-- 1-> Exam id Invalid : Null  
-- 2-> Exam Not Found  
-- 3-> Failure  
-- More Than 10 -> Error in Updating StudentTakeExamTable, check for error - 10 code  
-- More Than 20 -> Error in Updating Student Course, check for error - 20 code  
-- =====
```

### Input/Output

Name		Data type	Description
→@	ExamId	int	

## 2.41. Procedure: ex.spGenerateExam

```
-- Author:      Mohamed Shawky
-- Create date: 23/1/2020
--version update 26/1/2020
-- Description: Generate new Exam with random questions
-- Parameters:
-- @CourseName
-- @@StudentID
-- @ExamType
-- @NumberOfTrueFalseQuestions
-- @NumberOfMCQQuestions
```

### Input/Output

	Name	Data type	Description
→@	CourseName	varchar(25)	
→@	StudentID	int	
→@	ExamType	varchar(20)	
→@	NumberOfTrueFalseQuestions	int	
→@	NumberOfMCQQuestions	int	



## 2.42. Procedure: ex.spGetAllExamsData

```
-- Author: Ahmed Nasr  
-- CREATE date: 25/1/2020  
-- Description: Retrive All Data From Exam Table  
-- Parameters: NON
```

TRIAL

#### 2.43. Procedure: ex.spGetAllMCQQuestionsData

```
-- Author: Ahmed Nasr  
-- Create date: 25/1/2020  
-- Description: Retrive All Data From MCQ_Choices Table  
-- Parameters: NON
```

TRIAL

## 2.44. Procedure: ex.spGetExambyID

```
-- Author:      Mohamed Shawky
-- Create date: 23/1/2020
--version update 22/1/2020
-- Description: Get Questions with multi choices in specific Exam
-- Parameters:
-- @ExamID
```

### Input/Output

	Name	Data type	Description
→@	ExamID	int	

TRIAL

## 2.45. Procedure: ex.spGetMCQQuestionById

```
-- Author: Ahmed Nasr  
-- Create date: 25/1/2020  
-- Description: Retrive Record By Question_Id From MCQ_Choices Table  
-- Parameters: Question ID
```

### Input/Output

Name		Data type	Description
→@	Question_id	int	

## 2.46. Procedure: ex.spGetQuestionsByCourseId

```
-- Author:      Mohamed Shawky
-- Create date: 23/1/2020
-- Description:  Get all Questions within a course
-- Parameters:
-- @CourseID
```

### Input/Output

	Name	Data type	Description
→@	CourseId	int	

TRIAL

## 2.47. Procedure: ex.splInsertExam

```
-- Author: Ahmed Nasr
-- CREATE date: 25/1/2020
-- Description: Insert Record In Exam Table
-- Parameters:
-- Exam_Id
-- Exam_Type
```

### Input/Output

Name		Data type	Description
→@	Exam_Type	varchar(20)	

TRIAL

## 2.48. Procedure: ex.splInsertMCQuestion

```
-- Author: Ahmed Nasr
-- Create date: 25/1/2020
-- Description: Insert Record In MCQ_Choices Table
-- Parameters:
-- Question_id
-- Label
-- Text
```

### Input/Output

Name		Data type	Description
→@	Question_id	int	
→@	Label	varchar(1)	
→@	Text	varchar(50)	

TRIAL

## 2.49. Procedure: ex.splInsertQuestion

```
-- Author:      Mohamed Shawky
-- Create date: 23/1/2020
-- Description: Delete Question if it's not choosen in exam
-- Parameters:
-- @ModelAnswer
-- @Marks
-- @Type
-- @Body
-- @CourseID
```

### Input/Output

	Name	Data type	Description
→@	ModelAnswer	varchar(25)	
→@	Marks	int	
→@	Type	varchar(25)	
→@	Body	varchar(200)	
→@	CourseID	int	



## 2.50. Procedure: ex.spRetriveTheExamQuestionsStudentAnswers

```
-- Author: Ahmed Nasr
-- Create date: 25/1/2020
-- Description: Takes Exam number and the student ID then returns the Questions in
-- this exam with the student answers.
-- Parameters:
-- Exam ID
-- Student ID
```

### Input/Output

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

TRIAL

## 2.51. Procedure: ex.spStudentTakeExamDeleteCourseRecordsByCourseId

```
-- Author: Abdallh
-- Description: Delete Records By Course ID From StudentTakesExam Table
-- NOTE (DELETES ALL COURSE EXAMS) ( CAN DELETE MULTIPLE ROWS )
-- Input : Course ID
-- Returns: 0 -> Success
-- 1 -> Course ID NULL
-- 2 -> Course ID Doesn't Exist
-- 3 -> failure
```

### Input/Output

Name		Data type	Description
→@	CourseId	int	

## 2.52. Procedure: ex.spStudentTakeExamDeleteRecordByExamId

```
-- Author: Abdallh
-- Description: Delete Record From StudentTakesExam Table By Exam ID
-- Input : Exam ID
-- Returns: 0 -> Success
-- 1 -> Exam ID NULL
-- 2 -> Exam ID Doesn't Exist
-- 3 -> failure
```

### Input/Output

Name		Data type	Description
→@	ExamId	int	

### 2.53. Procedure: ex.spStudentTakeExamGetAll

```
-- Author: Abdalh  
-- Description: Get All Data From StudentTakesExam Table
```

TRIAL

## 2.54. Procedure: ex.spStudentTakeExamGetAllByCourseId

-- Author: Abdallh

-- Description: Get All Data From StudentTakesExam Table By Course ID

-- Inputs : Course ID

### Input/Output

Name		Data type	Description
→@	CourseId	int	

TRIAL

## 2.55. Procedure: ex.spStudentTakeExamGetAllByExamId

-- Author: Abdallh

-- Description: Get All Data From StudentTakesExam Table By Exam Id

-- Inputs : Exam ID

### Input/Output

	Name	Data type	Description
→@	ExamId	int	

TRIAL

## 2.56. Procedure: ex.spStudentTakeExamGetAllByGrade

-- Author: Abdallh

-- Description: Get All Data From StudentTakesExam Table By Grade -- Gets Who is higher than this grade

-- Inputs: Grade

### Input/Output

Name		Data type	Description
→@	Grade	int	

## 2.57. Procedure: ex.spStudentTakeExamGetAllByStudentId

-- Author: Abdallh

-- Description: Get All Data From Student StudentTakesExam By Student ID

### Input/Output

	Name	Data type	Description
→@	StudentId	int	



## 2.58. Procedure: ex.spStudentTakeExamInsertRecord

```
-- Author: Abdallh
-- Description: Insert New Record  Data into StudentTakesExam  Table
-- Inputs : Student ID , Exam ID, Course ID , Exam Date , Grade
-- Returns: 0 -> Success
-- 1 -> Student id cant be NULL
-- 2 -> Course id Cant Be NULL
-- 3 -> Exam id Cant Be Null
-- 4 -> Exam Date Cant Be Null
-- 5 -> Exam Grade Cant Be Null
-- 6 -> Student Doesn't exist
-- 7 -> Course Doesn't exist
-- 8 -> Exam Doesn't exist
-- 9 -> Exam ID Already Exists
-- 10-> Failure
```

### Input/Output

Name		Data type	Description
→@	StudentId	int	
→@	CourseId	int	
→@	ExamId	int	
→@	ExamDate	date	
→@	ExamGrade	int	

## 2.59. Procedure: ex.spStudentTakeExamUpdateRecordByExamId

```
-- Author: Abdallh
-- Description: Update a Record In StudentTakesExam Table With Exam Id
-- Input : Exam Id ,New Grade [optional], New Course Id [optional],
-- New Student Id [optional], New Exam Date [optional]
-- Returns: 0 -> Success
-- 1 -> invalid Exam Id : NULL
-- 2 -> Exam doesn't exist
-- 3 -> Course not found
-- 4 -> Student not found
-- 5 -> Failure
```

### Input/Output

Name		Data type	Description
→@	ExamId	int	
→@	NewGrade	int	
→@	NewCourseId	int	
→@	NewExamDate	date	
→@	NewStudentId	int	

## 2.60. Procedure: ex.spUpdateExamTypeById

```
-- Author: Ahmed Nasr
-- CREATE date: 25/1/2020
-- Description: Update Exam Type By Exam_Id From Exam Table
-- Parameters:
-- Exam_id
-- Exam_Type
```

### Input/Output

Name		Data type	Description
→@	Exam_id	int	
→@	Exam_Type	varchar(20)	

## 2.61. Procedure: ex.spUpdateMCQuestionDataById

```
-- Author: Ahmed Nasr
-- Create date: 25/1/2020
-- Description: Update Question Data By Question_Id From MCQ_Choices Table
-- Parameters:
-- Question_id
-- Label
-- Text
```

### Input/Output

	Name	Data type	Description
→@	Question_id	int	
→@	Label	varchar(1)	
→@	Text	varchar(50)	

## 2.62. Procedure: ex.spUpdateQuestionAnswerByld

```
-- Author:      Mohamed Shawky
-- Create date: 23/1/2020
-- Description: Update Question ModelAnswer if it's not choosen in exam
-- Parameters:
-- @QuestionID
```

### Input/Output

Name		Data type	Description
→@	QuestionId	int	
→@	NewModelAnswer	int	

TRIAL

## 2.63. Procedure: ex.spUpdateQuestionMarksByld

```
-- Author:      Mohamed Shawky
-- Create date: 23/1/2020
-- Description: Update Question Marks if it's not choosen in exam
-- Parameters:
-- @QuestionID
```

### Input/Output

Name		Data type	Description
→@	QuestionId	int	
→@	NewMarks	int	

## 2.64. Procedure: ex.spUpdateQuestionStateByld

```
-- Author:      Mohamed Shawky
-- Create date: 23/1/2020
-- Description: Update Question state to be not active
-- Parameters:
-- @QuestionID
```

### Input/Output

Name		Data type	Description
→@	QuestionId	int	
→@	NewStateOfQuestion	bit	

## 2.65. Procedure: uni.spCourseDeleteRecordByName

```
-- Author: Abdallh
-- Description: Delete Record By Course Name From Course Table
-- Input : Course Name
-- Returns: 0 -> Success
-- 1 -> Course Name NULL | Empty
-- 2 -> Course Name Doesn't Exist
-- 3 -> Failure
```

### Input/Output

Name		Data type	Description
→@	CourseName	nvarchar(25)	



## 2.66. Procedure: uni.spCourseGetAll

```
-- Author: Abdalh  
-- Description: Get All Course Data From Course Table  
-- Inputs: None
```

TRIAL

## 2.67. Procedure: uni.spCourseGetAllByCrsId

-- Author: Abdallh

-- Description: Get All Course Data By Course ID From Course Table

### Input/Output

Name		Data type	Description
→@	id	int	

## 2.68. Procedure: uni.spCourseGetAllByCrsName

-- Author: Abdallh

-- Description: Get All Course Data By Course Name From Course Table

### Input/Output

Name		Data type	Description
→@	crsName	nvarchar(25)	

## 2.69. Procedure: uni.spCourseGetAllByTopicId

-- Author: Abdallh

-- Description: Get All Course Data By Topic ID From Course Table

### Input/Output

Name		Data type	Description
→@	topicId	int	

TRIAL

## 2.70. Procedure: uni.spCourseGetAllByTopicName

-- Author: Abdallh

-- Description: Get All Course Data By Topic Name From Course Table

### Input/Output

Name		Data type	Description
→@	topicName	nvarchar(25)	

## 2.71. Procedure: uni.spCourseInsertRecord

```
-- Author: Abdallh
-- Description: Insert a record into Course Table
-- Input : Course Name , Course Duration , Topic ID
-- Returns: 0 -> Success
-- 1 -> invalid topic ID / NULL topic ID
-- 2 -> Course Name Cant Be Null
-- 3 -> Course Duration Cant Be Null
```

### Input/Output

Name		Data type	Description
→@	CourseName	nvarchar(25)	
→@	CourseDuration	int	
→@	TopicID	int	

## 2.72. Procedure: uni.spCourseInsertRecordWithTopicId

```
-- Author: Abdallh
-- Description: Insert a Record into Course Table With Topic ID
-- Input : Topic ID , CourseName , Course Duration
-- Returns: 0 -> Success
-- 1 -> Topic Id Doesn't Exist
-- 2 -> invalid Course Name
-- 3 -> invalid Course Duration
-- 4 -> General Failure
```

### Input/Output

Name		Data type	Description
→@	CourseName	nvarchar(25)	
→@	CourseDuration	int	
→@	TopicID	int	

## 2.73. Procedure: uni.spCourseInsertRecordWithTopicName

```
-- Author: Abdallh
-- Description: Insert a Record into Course Table With Topic Name
-- Input : TopicName , Course Name , Course Duration
-- Returns:      0 -> Success
-- 1 -> invalid Topic Name : null | ''
-- 2 -> invalid Course Name : null | ''
-- 3 -> invalid Course Duration
-- 4 -> Topic Name Doesn't Exist
-- 5 -> General Failure ??
```

### Input/Output

Name		Data type	Description
→@	CourseName	nvarchar(25)	
→@	CourseDuration	int	
→@	TopicName	nvarchar(25)	



## 2.74. Procedure: uni.spCourseUpdateRecordWithCourseId

```
-- Author: Abdallh
-- Description: Update a Record In Course Table With Course Id
-- Input : Course Id, New Course Name [optional], new Course Duration [optional], new Topic ID [optional],
-- Returns: 0 -> Success
-- 1 -> invalid Course Id : Null
-- 2 -> Course doesn't exist
-- 3 -> New Topic key doesn't exist
-- 4 -> General Failure ??
```

### Input/Output

	Name	Data type	Description
→@	CourseId	int	
→@	NewCourseName	nvarchar(25)	
→@	NewCourseDuration	int	
→@	NewTopicId	int	

## 2.75. Procedure: uni.spCourseUpdateRecordWithCourseName

```
-- Author: Abdallh
-- Description: Update a Record In Course Table With Course Name
-- Input : Old Course Name, New Course Name [optional], new Course Duration [optional] , new Topic
ID [optional],
-- Returns: 0 -> Success
-- 1 -> invalid Course Name : null | ''
-- 2 -> Course doesn't exist
-- 3 -> New Topic key doesn't exist
-- 4 -> General Failure ??
```

### Input/Output

	Name	Data type	Description
→@	CourseName	nvarchar(25)	
→@	NewCourseName	nvarchar(25)	
→@	NewCourseDuration	int	
→@	NewTopicId	int	

## 2.76. Procedure: uni.spDeleteInstructorByFullName

```
-- Author: Ahmed Nasr
-- Create date: 25/1/2020
-- Description: Delete Record By Instructor Full Name From Instructor Table
-- Parameters:
-- ins_FName
-- ins_LName
```

### Input/Output

Name		Data type	Description
→@	ins_FName	varchar(25)	
→@	ins_LName	varchar(25)	

## 2.77. Procedure: uni.spDeleteInstructorById

```
-- Author: Ahmed Nasr  
-- Create date: 25/1/2020  
-- Description: Delete Record By Ins_Id From Instructor Table  
-- Parameters: Instrcutor ID
```

### Input/Output

Name		Data type	Description
→@	ins_id	int	

TRIAL

## 2.78. Procedure: uni.spDeleteInstructorCourse

```
-- Author:      Mohamed Shawky
-- Create date: 20/1/2020 - My BirthDate :)
-- Description: Delete relation between an instructor and course
-- Parameters:
-- @CourseId
```

### Input/Output

Name		Data type	Description
→@	CourseId	int	

TRIAL

## 2.79. Procedure: uni.spDepartmentDeleteById

```
-- Author:      Moaz ayman
-- Create date: 20/1/2020
-- Description: Delete Department by ID
-- Parameters:
-- @Dept_ID
```

### Input/Output

Name		Data type	Description
→@	Dept_ID	int	

## 2.80. Procedure: uni.spDepartmentDeleteByName

```
-- Author:      Moaz ayman
-- Create date: 20/1/2020
-- Description: Delete Department by Name
-- Parameters:
-- @DeptName
```

### Input/Output

Name		Data type	Description
→@	DeptName	varchar(20)	

TRIAL

## 2.81. Procedure: uni.spDepartmentGetall

```
-- Author:      Moaz ayman  
-- Create date: 20/1/2020  
-- Description: Get all information about All departments  
-- Parameters: NON
```

TRIAL



## 2.82. Procedure: uni.spDepartmentGetById

```
-- Author:      Moaz ayman
-- Create date: 20/1/2020
-- Description:  Get all information about specific department
-- Parameters:
--   @Department_id
```

### Input/Output

Name		Data type	Description
→@	PID	int	

## 2.83. Procedure: uni.spDepartmentInser

```
-- Author:      Moaz ayman
-- Create date: 20/1/2020
-- Description: Insert New Department
-- Parameters:
-- @DeptName
-- @Dept_Desc NULL
-- @Dept_location NULL
-- @mgr_id
-- @mgr_HireDate
```

### Input/Output

	Name	Data type	Description
→@	DeptName	varchar(20)	
→@	Dept_Desc	varchar(300)	
→@	Dept_location	varchar(50)	
→@	mgr_id	int	
→@	mgr_HireDate	date	

## 2.84. Procedure: uni.spDepartmentUpdateMangerHireDate

```
-- Author:      moaz ayman
-- Create date: 20/1/2020
-- Description: update Department's Manger__HireDate for specific department
-- Parameters:  @Dept_ID,@mgr_HireDate
```

### Input/Output

Name		Data type	Description
→@	Dept_ID	int	
→@	mgr_HireDate	date	

## 2.85. Procedure: uni.spDepartmentUpdateMangerID

```
-- Author:      moaz ayman
-- Create date: 20/1/2020
-- Description: update Department's Manger_ID for specific department
-- Parameters:  @Dept_ID, @mgr_id ,@mgr_HireDate
```

### Input/Output

Name		Data type	Description
→@	Dept_ID	int	
→@	mgr_id	int	
→@	mgr_HireDate	date	

TRIAL

## 2.86. Procedure: uni.spDepartmentUpdateName

```
-- Author:      moaz ayman
-- Create date: 20/1/2020
-- Description: Updates Department Name for a Department
-- Parameters:
--   @Dept_ID
--   @DeptName
```

### Input/Output

Name		Data type	Description
→@	Dept_ID	int	
→@	DeptName	varchar(20)	

## 2.87. Procedure: uni.spGetAllInstructorsData

```
-- Author: Ahmed Nasr  
-- Create date: 25/1/2020  
-- Description: Retrive All Data From Instructor Table  
-- Parameters: NON
```

TRIAL

## 2.88. Procedure: uni.spGetDepartmentByInsId

```
-- Author:      Lojiun Elsayed
-- Create date: 22/1/2020
-- Description: get department by instructorID
-- Parameters:
-- @Ins_Id
```

### Input/Output

	Name	Data type	Description
→@	InstructorID	int	

## 2.89. Procedure: uni.spGetDeptNameByInsId

```
-- Author:      lojiun Elsayed
-- Create date: 22/1/2020
-- Description: get Get Department Name By Instructor ID
-- Parameters: @ID
```

### Input/Output

Name		Data type	Description
→@	ID	int	

TRIAL



## 2.90. Procedure: uni.spGetInsNameByDeptId

```
-- Author:      lojiun Elsayed
-- Create date: 22/1/2020
-- Description: get Get Instructor Name By Department Id
-- Parameters: @ID
```

### Input/Output

Name		Data type	Description
→@	ID	int	

## 2.91. Procedure: uni.spGetInstructorByDepartmentid

```
-- Author:      lojiun Elsayed
-- Create date: 22/1/2020
-- Description: get Instructor working on department
-- Parameters:
-- @Dept_Id
```

### Input/Output

Name		Data type	Description
→@	Dept_Id	int	

## 2.92. Procedure: uni.spGetInstructorById

```
-- Author: Ahmed Nasr  
-- Create date: 25/1/2020  
-- Description: Retrive Record By Ins_Id From Instructor Table  
-- Parameters: Instrcutor ID
```

### Input/Output

Name		Data type	Description
→@	ins_id	int	

TRIAL

### 2.93. Procedure: uni.splns\_works\_deptDeleteAllData

```
-- Author:      lojiun Elsayed  
-- Create date: 22/1/2020  
-- Description: Delete All Data in Ins_works_dept  
-- Parameters: NON
```

TRIAL

## 2.94. Procedure: uni.spIns\_works\_deptDeleteDeptIdByInsId

```
-- Author:      lojiun Elsayed
-- Create date: 22/1/2020
-- Description: Delete department in Ins_works_dept by instructor
-- Parameters:  NON
```

### Input/Output

Name		Data type	Description
→@	id	int	

## 2.95. Procedure: uni.spIns\_works\_deptDeleteInsIdByDeptId

-- Author: lojiun Elsayed  
-- Create date: 22/1/2020  
-- Description: Delete department in Ins\_works\_dept by instructor  
-- Parameters: NON

### Input/Output

Name		Data type	Description
→@	id	int	

TRIAL

## 2.96. Procedure: uni.splns\_works\_deptGetAll1

```
-- Author:      lojiun Elsayed  
-- Create date: 22/1/2020  
-- Description: get All Data in Ins_works_dept  
-- Parameters: NON
```

TRIAL

## 2.97. Procedure: uni.splns\_works\_deptSetValue

```
-- Author:      Lojiun elsayed
-- Create date: 22/1/2020
-- Description: Set  Instructor to Department
-- Parameters:
-- @InsID
-- @deptId
```

### Input/Output

Name		Data type	Description
→@	Dept_id	int	
→@	Ins_id	int	

TRIAL



## 2.98. Procedure: uni.spInsertInstructor

```
-- Author: Ahmed Nasr
-- Create date: 25/1/2020
-- Description: Insert Record From Instructor Table
-- Parameters:
-- ins_id
-- ins_FName
-- ins_LName
-- ins_age
-- ins_salary
-- ins_hireDate
-- ins_gender
```

### Input/Output

	Name	Data type	Description
→@	ins_FName	varchar(25)	
→@	ins_LName	varchar(25)	
→@	ins_age	int	
→@	ins_salary	decimal(18, 2)	
→@	ins_hireDate	date	
→@	ins_gender	varchar(1)	

## 2.99. Procedure: uni.spInsertInstructorCourse

```
-- Author:      Mohamed Shawky
-- Create date: 20/1/2020 - My BirthDate :)
-- Description: Set new course taught by specific Instructor
-- Parameters:
-- @InstructorID
-- @CourseId
```

### Input/Output

Name		Data type	Description
→@	InstructorID	int	
→@	CourseID	int	

TRIAL

## 2.100. Procedure: uni.spReplaceInstructorToOntherOne

```
-- Author:      Mohamed Shawky
-- Create date: 20/1/2020 - My BirthDate :)
-- Description: replace instructor with new one to teach the same courses
-- Parameters:
-- @OldInstructorId
-- @NewOldInstructorId
```

### Input/Output

Name		Data type	Description
→@	OldInstructor	int	
→@	NewInstructor	int	

## 2.101. Procedure: uni.spSelectCoursesByInslId

```
-- Author:      Mohamed Shawky
-- Create date: 20/1/2020 - My BirthDate :)
-- Description: get All courses taught by specific instructor
-- Parameters:
-- @Instructor_Id
```

### Input/Output

Name		Data type	Description
→@	InstructorID	int	

TRIAL

## 2.102. Procedure: uni.spSelectInstructorsByCrslId

```
-- Author:      Mohamed Shawky
-- Create date: 20/1/2020 - My BirthDate :)
-- Description: get All Instructors teaching specific course
-- Parameters:
-- @Course_Id
```

### Input/Output

	Name	Data type	Description
→@	CourseID	int	

TRIAL

### 2.103. Procedure: uni.spStudentCourseDeleteRecordByStudentIdAndCourseId

```
-- Author: Abdallh
-- Description: Delete Record From StudentCourse Table By StudentId + Course Id
-- Input : Student id , Course ID
-- Returns: 0 -> Success
-- 1 -> Course ID NULL
-- 2 -> Student ID NULL
-- 5 -> failure
```

#### Input/Output

Name		Data type	Description
→@	StudentId	int	
→@	CourseId	int	

## 2.104. Procedure: uni.spStudentCourseGetCourseAndGradeByStudentId

```
-- Author: Abdallh
-- Description: Get All Courses and thier Grade By Student Id From StudentCourse Table
-- Inputs: Student Id
-- Returns:
-- 0-> Success
-- 1-> Student Id Not Valid : NULL
-- 2-> Student not found
-- 3-> Failure
```

### Input/Output

Name		Data type	Description
→@	StudentId	int	

## 2.105. Procedure: uni.spStudentCourseGetCourseAndGradeByStudentName

```
-- Author: Abdallh
-- Description: Get All Courses and thier Grade By Student Full Name From StudentCourse Table
-- Inputs: Student Full Name
-- Returns:
-- 0-> Success
-- 1-> Student Name Not Valid : NULL | ''
-- 2-> Student not found
-- 3-> Failure
```

### Input/Output

Name		Data type	Description
→@	StudentFullName	varchar(60)	



## 2.106. Procedure: uni.spStudentCourseGetStudentAndGradeByCourseId

```
-- Author: Abdallh
-- Description: Get All Students and thier Grade By Course Id From StudentCourse Table
-- Inputs: Course Id
-- Returns:
-- 0-> Success
-- 1-> Course Id Not Valid : NULL
-- 2-> Course not found
-- 3-> Failure
```

### Input/Output

Name		Data type	Description
→@	CourseId	int	

## 2.107. Procedure: uni.spStudentCourseGetStudetAndGradeByCourseName

```
-- Author: Abdallh
-- Description: Get All Student and thier Grade By Course Name From StudentCourse Table
-- Inputs: Student Full Name
-- Returns:
-- 0-> Success
-- 1-> Course Name Not Valid : NULL | ''
-- 2-> Course not found
-- 3-> Failure
```

### Input/Output

Name		Data type	Description
→@	CourseName	varchar(60)	

## 2.108. Procedure: uni.spStudentCourseInsertRecord

```
-- Author: Abdallh
-- Description: Insert New Record Data into StudentCourse Table
-- Inputs : Student ID ,Course ID ,FinalGrade
-- Returns: 0 -> Success
-- 1 -> Student id cant be NULL
-- 2 -> Course id Cant Be NULL
-- 3 -> Final Grade Cant Be Null
-- 4 -> Student Doesn't exist
-- 5 -> Course Doesn't exist
-- 6-> Record Already Exists
```

### Input/Output

Name		Data type	Description
→@	StudentId	int	
→@	CourseId	int	
→@	FinalGrade	int	

## 2.109. Procedure: uni.spStudentCourseUpdateRecordByStudentIdAndCourseId

```
-- Author: Abdallh
-- Description: Update Student Grade in Student Course Table By Student ID + Course ID
-- Input : Student ID , Course ID , New Grade
-- Returns: 0 -> Success
-- 1 -> invalid Student Id : NULL
-- 2 -> invalid Course Id : NULL
-- 3 -> Student doesn't exist
-- 4 -> Course doesn't exist
-- 5 -> Failure
```

### Input/Output

Name		Data type	Description
→@	StudentId	int	
→@	CourseId	int	
→@	NewGrade	int	

## 2.110. Procedure: uni.spTopicDeleteRecordByld

```
-- Author: Abdallh
-- Description: Delete Record By Topic ID From Topic Table
-- Input : Topic ID
-- Returns: 0 -> Success
-- 1 -> Topic Name NULL | Empty
-- 2 -> Topic Name Doesn't Exist
-- 3 -> FK Error , Topic has subjects not deleted.
```

### Input/Output

Name		Data type	Description
→@	TopicID	int	

## 2.111. Procedure: uni.spTopicDeleteRecordByName

```
-- Author: Abdallh
-- Description: Delete Record By Topic Name From Topic Table
-- Input : Topic Name
-- Returns: 0 -> Success
-- 1 -> Topic Name NULL | Empty
-- 2 -> Topic Name Doesn't Exist
-- 3 -> FK Error , Topic has subjects not deleted.
```

### Input/Output

Name		Data type	Description
→@	TopicName	nvarchar(25)	

## 2.112. Procedure: uni.spTopicGetAll

-- Author: Abdalh

-- Description: Get All Topic Data From Topic Table

TRIAL

### 2.113. Procedure: uni.spTopicGetAllByCourseId

```
-- Author: Abdallh
-- Description: Get All Topic Data By Course ID From Topic Table
-- Input: Course ID
-- Return: 0 -> Success
-- 1 -> Course Id Invalid : NULL
-- 2 -> Course Not Found
```

#### Input/Output

Name		Data type	Description
→@	crsId	int	



## 2.114. Procedure: uni.spTopicGetAllByCourseName

-- Author: Abdallh

-- Description: Get All Topic Data By Course Name From Topic Table

### Input/Output

Name		Data type	Description
→@	crsName	nvarchar(25)	

## 2.115. Procedure: uni.spTopicGetTopicIdByTopicName

```
-- Author: Abdallh
-- Description: Get Topic Id By Topic Name from Topic Table
-- Input : Topic Name , Topic ID OUTPUT
-- output: @TopicID
```

### Input/Output

	Name	Data type	Description
→@	TopicName	nvarchar(25)	
→@	TopicID	int	

## 2.116. Procedure: uni.spTopicGetTopicNameByCourseId

-- Author: Abdallh

-- Description: Get Topic Name By Course ID From Topic Table

### Input/Output

Name		Data type	Description
→@	crsld	int	

TRIAL

## 2.117. Procedure: uni.spTopicGetTopicNameByCourseName

-- Author: Abdallh

-- Description: Get TopicName By Course Name From Topic Table

### Input/Output

Name		Data type	Description
→@	crsName	nvarchar(25)	

## 2.118. Procedure: uni.spTopicGetTopicNames

-- Author: Abdalh  
-- Description: Get All Topic Names From Topic Table

TRIAL

## 2.119. Procedure: uni.spTopicInsertRecord

```
-- Author: Abdallh
-- Description: Insert a record into Topic Table
-- Input : Topic Name
-- Returns: 0 -> Success
-- 1 -> invalid topic Name
```

### Input/Output

Name		Data type	Description
→@	TopicName	nvarchar(25)	

2.120. Procedure: uni.spTopicUpdateRecordWithTopicId

```
-- Author: Abdallh
-- Description: Update a Record In Topic Table With Topic Id
-- Input : Topic Id, New Topic Name
-- Returns: 0 -> Success
-- 1 -> invalid Topic Id : NULL
-- 2 -> Invalid New Topic Name: Null | ''
-- 3 -> Topic not found
-- 4 -> General Failure ??
```

Input/Output

Name		Data type	Description
➤@	TopicID	int	
➤@	NewTopicName	nvarchar(25)	

## 2.121. Procedure: uni.spTopicUpdateRecordWithTopicName

```
-- Author: Abdallh
-- Description: Update a Record In Topic Table With Topic Name
-- Input : Old Topic Name, New Topic Name
-- Returns: 0 -> Success
-- 1 -> invalid Topic Name : NULL | ''
-- 2 -> invalid New Topic Name : NULL | ' '
-- 3 -> Topic not found
-- 4 -> General Failure ??
```

### Input/Output

Name		Data type	Description
➤@	TopicName	nvarchar(25)	
➤@	NewTopicName	nvarchar(25)	



## 2.122. Procedure: uni.spUpdateCourseInstructor

```
-- Author:      Mohamed Shawky
-- Create date: 20/1/2020 - My BirthDate :)
-- Description: set new instructor to teach specific course
-- Parameters:
-- @courseId
-- @InstructorId
```

### Input/Output

Name		Data type	Description
→@	CourseId	int	
→@	NewInstructorId	int	

### 2.123. Procedure: uni.spUpdateDepartMentToMangeByNewOne

-- Author: Lojiun elsayed  
-- Create date: 22/1/2020  
-- Description: set new Department to the same Instructor  
-- Parameters: @OldInsId @NewInsId

#### Input/Output

Name		Data type	Description
→@	OldDeptId	int	
→@	NewDeptId	int	

TRIAL

## 2.124. Procedure: uni.spUpdateInstructorCourse

```
-- Author:      Mohamed Shawky
-- Create date: 22/1/2020 - My BirthDate :)
-- Description: set instructor to teach another course
-- Parameters:
-- @courseId
-- @InstructorId
```

### Input/Output

Name		Data type	Description
→@	InstructorId	int	
→@	NewCourseId	int	

## 2.125. Procedure: uni.spUpdateInstructorDataById

```
-- Author: Ahmed Nasr
-- Create date: 25/1/2020
-- Description: Update Record By Ins_Id From Instructor Table
-- Parameters:
-- ins_id
-- ins_FName
-- ins_LName
-- ins_age
-- ins_salary
-- ins_hireDate
-- ins_gender
```

### Input/Output

Name		Data type	Description
→@	ins_id	int	
→@	ins_FName	varchar(25)	
→@	ins_LName	varchar(25)	
→@	ins_age	int	
→@	ins_salary	decimal(18, 2)	
→@	ins_hireDate	date	
→@	ins_gender	varchar(1)	

## 2.126. Procedure: uni.spUpdateInstructorofDepartMent

```
-- Author:      Lojiun elsayed
-- Create date: 22/1/2020
-- Description: set new Instructor to work in the same department
-- Parameters:  @OldInsId @NewInsId
```

### Input/Output

Name		Data type	Description
→@	OldInsId	int	
→@	NewInsId	int	