System Documentation

examination_system

Server (local)\

Author Abdelrahman, Aya, Maha, Mayar, Nahla

Created 2023/02/28

File Path C:\Users\moham\Desktop\Documentation-2023-02-28T20-09-49.pdf

Examination System

Table of Contents

able of Contents	2
(local)\	5
User databases	
examination_system Database	8
Tables	11
[dbo].[Course]	12
[dbo].[Department]	14
Ⅲ [dbo].[Exam]	16
[dbo].[Instructor]	18
[dbo].[Instructor_courses]	20
[dbo].[Question]	22
[dbo].[Question_choices]	24
[dbo].[Student]	26
[dbo].[Student_course]	28
[dbo].[Student_exams_questions]	30
[dbo].[Topic]	32
Stored Procedures	34
[dbo].[choice_delete_by_id]	36
[dbo].[choice_update]	37
[dbo].[correct_exam]	38
[dbo].[coures_select_by_id]	40
[dbo].[coures_select_sp]	41
[dbo].[couress_delete_by_courese]	42
[dbo].[course_update_sp]	43
[dbo].[create_Exam]	45
[dbo].[delete_exam_by_id]	47
[dbo].[delete_student_exams_Questions_by_id]	48
[dbo].[delete_topic_by_id]	49
[dbo].[department_delete_sp]	50
[dbo].[department_insert_sp]	51
[dbo].[department_select_by_id_sp]	52
[dbo].[department_select_sp]	53
[dbo].[department_update_sp]	54
[dbo].[display_st_report_deprt]	55
[dbo].[exam_insert]	56
[dbo].[exam_qustions_choices]	57
[dbo].[exam_select_all]	58

[dbo].[exam_select_by_id]	59
[dbo].[exam_update]	60
[dbo].[ExamCorrection_calc_std_garde_sp]	61
[dbo].[get_Question_with_Student_answer]	63
[dbo].[get_student_answers]	64
[dbo].[get_topic_name_sp]	65
[dbo].[insert_choices]	66
[dbo].[insert_courese_sp]	67
[dbo].[insert_question]	68
[dbo].[instr_id]	70
[dbo].[instructor_course_insert_sp]	71
[dbo].[instructor_course_select_sp]	72
[dbo].[instructor_course_update_sp]	73
[dbo].[instructor_courses_delete_by_crsId_sp]	74
[dbo].[instructor_courses_delete_by_insld_sp]	75
[dbo].[instructor_delete_sp]	76
[dbo].[instructor_insert_sp]	77
[dbo].[instructor_select_by_course_id_sp]	78
[dbo].[instructor_select_by_id_sp]	79
[dbo].[instructor_select_by_instructor_id_sp]	80
[dbo].[instructor_select_by_instructor_id_sp] [dbo].[instructor_select_sp]	
	81
[dbo].[instructor_select_sp]	81 82
[dbo].[instructor_select_sp] [dbo].[instructor_update_sp]	
[dbo].[instructor_select_sp] [dbo].[instructor_update_sp] [dbo].[qChoice_select_by_id]	
[dbo].[instructor_select_sp]	
[dbo].[instructor_select_sp]	
[dbo].[instructor_select_sp] [dbo].[instructor_update_sp] [dbo].[qChoice_select_by_id] [dbo].[question_delete_by_id] [dbo].[question_select] [dbo].[question_select_by_id].	
[dbo].[instructor_select_sp] [dbo].[instructor_update_sp] [dbo].[qChoice_select_by_id] [dbo].[question_delete_by_id] [dbo].[question_select] [dbo].[question_select_by_id] [dbo].[question_select_by_id]	
[dbo].[instructor_select_sp] [dbo].[instructor_update_sp] [dbo].[qChoice_select_by_id] [dbo].[question_delete_by_id] [dbo].[question_select] [dbo].[question_select_by_id]. [dbo].[question_update] [dbo].[question_update]	
[dbo].[instructor_select_sp] [dbo].[instructor_update_sp] [dbo].[qChoice_select_by_id] [dbo].[question_delete_by_id] [dbo].[question_select] [dbo].[question_select_by_id] [dbo].[question_update] [dbo].[quetion_update] [dbo].[report_st_crs] [dbo].[st_crs_delete_by_cr_id]	
[dbo].[instructor_select_sp] [dbo].[instructor_update_sp] [dbo].[qChoice_select_by_id] [dbo].[question_delete_by_id] [dbo].[question_select] [dbo].[question_select_by_id]. [dbo].[question_update] [dbo].[question_update] [dbo].[st_crs_delete_by_cr_id] [dbo].[st_crs_delete_by_id].	
[dbo].[instructor_select_sp] [dbo].[instructor_update_sp] [dbo].[qChoice_select_by_id] [dbo].[question_delete_by_id] [dbo].[question_select] [dbo].[question_select_by_id]. [dbo].[question_update] [dbo].[question_update] [dbo].[report_st_crs] [dbo].[st_crs_delete_by_cr_id] [dbo].[st_crs_delete_by_id] [dbo].[st_ex_qs_insert]	
[dbo].[instructor_select_sp] [dbo].[instructor_update_sp] [dbo].[qChoice_select_by_id] [dbo].[question_delete_by_id] [dbo].[question_select] [dbo].[question_select_by_id]. [dbo].[question_update] [dbo].[report_st_crs] [dbo].[st_crs_delete_by_cr_id] [dbo].[st_crs_delete_by_id] [dbo].[st_ex_qs_insert] [dbo].[st_ex_qs_select_all]	
[dbo].[instructor_select_sp] [dbo].[instructor_update_sp] [dbo].[qChoice_select_by_id] [dbo].[question_delete_by_id] [dbo].[question_select] [dbo].[question_select_by_id]. [dbo].[question_update] [dbo].[report_st_crs] [dbo].[st_crs_delete_by_cr_id] [dbo].[st_crs_delete_by_id] [dbo].[st_ex_qs_insert] [dbo].[st_ex_qs_select_all] [dbo].[st_ex_qs_select_by_ids]	
[dbo].[instructor_select_sp] [dbo].[instructor_update_sp] [dbo].[qChoice_select_by_id] [dbo].[question_delete_by_id] [dbo].[question_select] [dbo].[question_select_by_id]. [dbo].[question_update] [dbo].[question_update] [dbo].[st_crs_delete_by_cr_id] [dbo].[st_crs_delete_by_id] [dbo].[st_ex_qs_insert] [dbo].[st_ex_qs_select_all] [dbo].[st_ex_qs_select_by_ids] [dbo].[st_ex_qs_update]	
[dbo].[instructor_select_sp] [dbo].[qchoice_select_by_id] [dbo].[question_delete_by_id] [dbo].[question_select] [dbo].[question_select_by_id] [dbo].[question_update] [dbo].[quetion_update] [dbo].[st_crs_delete_by_cr_id] [dbo].[st_crs_delete_by_id] [dbo].[st_ex_qs_insert] [dbo].[st_ex_qs_select_all] [dbo].[st_ex_qs_select_by_ids] [dbo].[st_ex_qs_update] [dbo].[st_ex_qs_update] [dbo].[std_cs_select_by_id]	

[dbo].[student_course_update_sp]	103
[dbo].[student_delete_sp]	105
[dbo].[student_insert_sp]	
[dbo].[student_select_by_id_sp]	
[dbo].[student_select_sp]	
[dbo].[student_update_sp]	110
[dbo].[topic_insert]	112
[dbo].[topic_select_all]	113
[dbo].[topic_select_by_id]	
[dbo].[topic update]	

≡ (local)\

Databases (1)

• **I** examination_system

Server Properties

Property	Value
Product	Microsoft SQL Server
Version	14.0.1000.169
Language	English (United States)
Platform	NT x64
Edition	Developer Edition (64-bit)
Engine Edition	3 (Enterprise)
Processors	4
OS Version	6.3 (19044)
Physical Memory	8051
Is Clustered	False
Root Directory	C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL
Collation	SQL_Latin1_General_CP1_CI_AS

Server Settings

Property	Value
Default data file path	C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\
Default backup file path	C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\Backup
Default log file path	C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\
Recovery Interval (minutes)	0
Default index fill factor	0
Default backup media retention	0
Compress Backup	False

Advanced Server Settings

Property	Value
Locks	0

Nested triggers enabled	True
Allow triggers to fire others	True
Default language	English
Network packet size	4096
Default fulltext language LCID	1033
Two-digit year cutoff	2049
Remote login timeout	10
Cursor threshold	-1
Max text replication size	65536
Parallelism cost threshold	5
Max degree of parallelism	0
Min server memory	16
Max server memory	2147483647
Scan for startup procs	False
Transform noise words	False
CLR enabled	False
Blocked process threshold	0
Filestream access level	False
Optimize for ad hoc workloads	False
CLR strict security	True

\sim					
$I \cap I$	l la	Y C	late	sha	ses
	USE	: L U	1010	11)0	26.2

Databases (1)

• examination_system

\boxminus examination_system Database

Database Properties

Property	Value
SQL Server Version	SQL Server 2017
Compatibility Level	SQL Server 2017
Last backup time	-
Last log backup time	-
Creation date	Feb 19 2023
Users	5
Database Encryption Enabled	False
Database Encryption Algorithm	None
Database size	16.00 MB
Unallocated space	3.81 MB

Database Options

Property	Value
Compatibility Level	140
Database collation	SQL_Latin1_General_CP1_CI_AS
Restrict access	MULTI_USER
Is read-only	False
Auto close	False
Auto shrink	False
Database status	ONLINE
In standby	False
Cleanly shutdown	False
Supplemental logging enabled	False
Snapshot isolation state	OFF
Read committed snapshot on	False
Recovery model	FULL
Page verify option	CHECKSUM
Auto create statistics	True
Auto update statistics	True
Auto update statistics asynchronously	False
ANSI NULL default	False
ANSI NULL enabled	False

ANSI padding enabled	False
ANSI warnings enabled	False
Arithmetic abort enabled	False
Concatenating NULL yields NULL	False
Numeric roundabort enabled	False
Quoted Identifier On	False
Recursive triggers enabled	False
Close cursors on commit	False
Local cursors by default	False
Fulltext enabled	True
Trustworthy	False
Database chaining	False
Forced parameterization	False
Master key encrypted by server	False
Published	False
Subscribed	False
Merge published	False
Is distribution database	False
Sync with backup	False
Service broker GUID	3de5414e-bf95-4f63-b40f-60b1ee302e4d
Service broker enabled	False
Log reuse wait	NOTHING
Date correlation	False
CDC enabled	False
Encrypted	False
Honor broker priority	False
Default language	English
Default fulltext language LCID	1033
Nested triggers enabled	True
Transform noise words	False
Two-digit year cutoff	2049
Containment	NONE
Target recovery time	60
Database owner	DESKTOP-UUS66HM\Maha

Files

Name	Туре	Size	Maxsize	Autogrowth	File Name
examination_system	Data	8.00 MB	unlimited	64.00 MB	C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERV ER\MSSQL\DATA\examination_

					system.mdf
examination_system_log	Log	8.00 MB	2048.00 GB	64.00 MB	C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERV ER\MSSQL\DATA\examination_ system_log.ldf

■ Tables

Objects

Name
dbo.Course
dbo.Department
dbo.Exam
dbo.Instructor
dbo.Instructor_courses
dbo.Question
dbo.Question_choices
dbo.Student
dbo.Student_course
dbo.Student_exams_questions
dbo.Topic

[dbo].[Course]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	7
Created	11:51:08 PM Tuesday, February 21, 2023
Last Modified	8:42:09 PM Thursday, February 23, 2023

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK	courses_id	int	4	NOT NULL	1 - 1
	course_name	varchar(50)	50	NULL allowed	
	duration	int	4	NULL allowed	
FK	topic_id	int	4	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
PK	PK_Course	courses_id	True

Foreign Keys

Name	Columns
FK_Course_Topic	topic_id->[dbo].[Topic].[topic_id]

```
CREATE TABLE [dbo].[Course]

(
[courses_id] [int] NOT NULL IDENTITY(1, 1),
[course_name] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[duration] [int] NULL,
[topic_id] [int] NULL
) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Course] ADD CONSTRAINT [PK_Course] PRIMARY KEY CLUSTERED
([courses_id]) ON [PRIMARY]
```

```
GO
ALTER TABLE [dbo].[Course] ADD CONSTRAINT [FK_Course_Topic] FOREIGN KEY ([topic_id])
REFERENCES [dbo].[Topic] ([topic_id])
GO
```

[dbo].[Topic]

Used By

[dbo].[Instructor_courses]

[dbo].[Question]

[dbo].[Student_course]

[dbo].[coures_select_by_id]

[dbo].[coures select sp]

[dbo].[couress_delete_by_courese]

[dbo].[course_update_sp]

[dbo].[create_Exam]

[dbo].[get_topic_name_sp]

[dbo].[insert_courese_sp]

[dbo].[insert_question]

[dbo].[instr_id]

[dbo].[instructor_course_update_sp]

[dbo].[quetion_update]

[dbo].[report_st_crs]

[dbo].[student_course_update_sp]

[dbo].[Department]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	5
Created	11:51:08 PM Tuesday, February 21, 2023
Last Modified	11:51:08 PM Tuesday, February 21, 2023

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK	dept_id	int	4	NOT NULL	1 - 1
FK	dept_manager	int	4	NULL allowed	
	dept_name	varchar(50)	50	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
PK G	PK_Department	dept_id	True

Foreign Keys

Name	Columns	
FK_Department_Instructor	dept_manager->[dbo].[Instructor].[insrt_id]	

```
CREATE TABLE [dbo].[Department]

(
[dept_id] [int] NOT NULL IDENTITY(1, 1),
[dept_manager] [int] NULL,
[dept_name] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Department] ADD CONSTRAINT [PK_Department] PRIMARY KEY CLUSTERED
([dept_id]) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Department] ADD CONSTRAINT [FK Department Instructor] FOREIGN KEY
```

```
([dept_manager]) REFERENCES [dbo].[Instructor] ([insrt_id])

GO
```

[dbo].[Instructor]

Used By

[dbo].[Instructor]

[dbo].[Student]

[dbo].[department_delete_sp]

[dbo].[department_insert_sp]

[dbo].[department_select_by_id_sp]

[dbo].[department_select_sp]

[dbo].[department_update_sp]

[dbo].[display_st_report_deprt]

[dbo].[instructor_update_sp]

[dbo].[student_insert_sp]

[dbo].[student_update_sp]

[dbo].[Exam]

Properties

Property	Value
Row Count (~)	2
Created	11:51:08 PM Tuesday, February 21, 2023
Last Modified	4:18:27 PM Saturday, February 25, 2023

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK G	exam_id	int	4	NOT NULL	1 - 1
	ex_date	date	3	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
PKP C	PK_Exam	exam_id	True

SQL Script

```
CREATE TABLE [dbo].[Exam]

(
    [exam_id] [int] NOT NULL IDENTITY(1, 1),
    [ex_date] [date] NULL

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Exam] ADD CONSTRAINT [PK_Exam] PRIMARY KEY CLUSTERED ([exam_id]) ON [PRIMARY]

GO

GO
```

```
[dbo].[Student_exams_questions]
[dbo].[delete_exam_by_id]
[dbo].[exam_insert]
[dbo].[exam_qustions_choices]
[dbo].[exam_select_all]
[dbo].[exam_select_by_id]
[dbo].[exam_update]

Author:
```

[dbo].[get_Question_with_Student_answer] [dbo].[st_ex_qs_insert] [dbo].[st_ex_qs_update] [dbo].[student_answers]

[dbo].[Instructor]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	7
Created	11:51:08 PM Tuesday, February 21, 2023
Last Modified	11:51:08 PM Tuesday, February 21, 2023

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK P	insrt_id	int	4	NOT NULL	1 - 1
	fname	varchar(50)	50	NULL allowed	
	Iname	varchar(50)	50	NULL allowed	
	ins_phone	varchar(11)	11	NULL allowed	
FK)	dept_id	int	4	NULL allowed	
	city	varchar(50)	50	NULL allowed	
	street	varchar(50)	50	NULL allowed	
	hire_date	date	3	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
PK G	PK_Instructor	insrt_id	True

Foreign Keys

Name Columns	
FK_Instructor_Department	dept_id->[dbo].[Department].[dept_id]
FK_Instructor_Department1	dept_id->[dbo].[Department].[dept_id]

```
CREATE TABLE [dbo].[Instructor]
```

```
[insrt_id] [int] NOT NULL IDENTITY(1, 1),
[fname] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[lname] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[ins_phone] [varchar] (11) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[dept_id] [int] NULL,
[city] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[street] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[hire_date] [date] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Instructor] ADD CONSTRAINT [PK_Instructor] PRIMARY KEY CLUSTERED
([insrt_id]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Instructor] ADD CONSTRAINT [FK_Instructor_Department] FOREIGN KEY
([dept_id]) REFERENCES [dbo].[Department] ([dept_id])
GO
ALTER TABLE [dbo].[Instructor] ADD CONSTRAINT [FK_Instructor_Department1] FOREIGN KEY
([dept_id]) REFERENCES [dbo].[Department] ([dept_id])
GO
```

[dbo].[Department]

```
[dbo].[Department]
[dbo].[Instructor_courses]
[dbo].[department_insert_sp]
[dbo].[instr_id]
[dbo].[instructor_course_update_sp]
[dbo].[instructor_delete_sp]
[dbo].[instructor_insert_sp]
[dbo].[instructor_select_by_id_sp]
[dbo].[instructor_select_sp]
[dbo].[instructor_update_sp]
```

[dbo].[Instructor_courses]

Properties

Property	Value
Row Count (~)	6
Created	11:51:08 PM Tuesday, February 21, 2023
Last Modified	11:51:08 PM Tuesday, February 21, 2023

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PKO FKO	instr_id	int	4	NOT NULL
PKO FKO	course_id	int	4	NOT NULL

Indexes

Key	Name	Key Columns	Unique
PKP C	PK_instructor_courses	instr_id, course_id	True

Foreign Keys

Name	Columns
FK_Instructor_courses_Course	course_id->[dbo].[Course].[courses_id]
FK_Instructor_courses_Instructor	instr_id->[dbo].[Instructor].[insrt_id]

```
CREATE TABLE [dbo].[Instructor_courses]

(
[instr_id] [int] NOT NULL,
[course_id] [int] NOT NULL

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Instructor_courses] ADD CONSTRAINT [PK_instructor_courses] PRIMARY

KEY CLUSTERED ([instr_id], [course_id]) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Instructor_courses] ADD CONSTRAINT [FK_Instructor_courses_Course]
```

```
FOREIGN KEY ([course_id]) REFERENCES [dbo].[Course] ([courses_id])

GO

ALTER TABLE [dbo].[Instructor_courses] ADD CONSTRAINT [FK_Instructor_courses_-
Instructor] FOREIGN KEY ([instr_id]) REFERENCES [dbo].[Instructor] ([insrt_id])

GO
```

[dbo].[Course] [dbo].[Instructor]

```
[dbo].[instr_id]
[dbo].[instructor_course_insert_sp]
[dbo].[instructor_course_select_sp]
[dbo].[instructor_course_update_sp]
[dbo].[instructor_courses_delete_by_crsId_sp]
[dbo].[instructor_courses_delete_by_insId_sp]
[dbo].[instructor_select_by_course_id_sp]
[dbo].[instructor_select_by_instructor_id_sp]
```

[dbo].[Question]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	9
Created	8:42:09 PM Thursday, February 23, 2023
Last Modified	7:23:04 PM Tuesday, February 28, 2023

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK	q_id	int	4	NOT NULL	1 - 1
	question	varchar(200)	200	NULL allowed	
	q_type	varchar(50)	50	NULL allowed	
	q_grade	int	4	NULL allowed	
	q_answer	varchar(10)	10	NULL allowed	
FK	courses_id	int	4	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
PK2 C	PK_Question	q_id	True

Foreign Keys

Name	Columns
FK_Question_Course	courses_id->[dbo].[Course].[courses_id]

```
CREATE TABLE [dbo].[Question]
(
[q_id] [int] NOT NULL IDENTITY(1, 1),
[question] [varchar] (200) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[q_type] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[q_grade] [int] NULL,
[q_answer] [varchar] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
```

```
[courses_id] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Question] ADD CONSTRAINT [PK_Question] PRIMARY KEY CLUSTERED
([q_id]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Question] ADD CONSTRAINT [FK_Question_Course] FOREIGN KEY
([courses_id]) REFERENCES [dbo].[Course] ([courses_id])
GO
```

[dbo].[Course]

```
[dbo].[Question_choices]
[dbo].[Student_exams_questions]
[dbo].[correct_exam]
[dbo].[create_Exam]
[dbo].[exam_qustions_choices]
[dbo].[ExamCorrection_calc_std_garde_sp]
[dbo].[get_Question_with_Student_answer]
[dbo].[get_student_answers]
[dbo].[insert_choices]
[dbo].[insert_question]
[dbo].[question_delete_by_id]
[dbo].[question_select]
[dbo].[question_select_by_id]
[dbo].[quetion_update]
[dbo].[st_ex_qs_insert]
[dbo].[st_ex_qs_update]
[dbo].[student_answers]
```

[dbo].[Question_choices]

Properties

Property	Value
Collation SQL_Latin1_General_CP1_CI_AS	
Row Count (~)	21
Created	7:23:03 PM Tuesday, February 28, 2023
Last Modified	7:23:04 PM Tuesday, February 28, 2023

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PKC FKC	q_id	int	4	NOT NULL
PK C	choices	varchar(50)	50	NOT NULL
PK C	choice_num	nchar(10)	20	NOT NULL

Indexes

Key	Name	Key Columns	Unique
PKP C	PK_Question_choices_1	q_id, choices, choice_num	True

Foreign Keys

Name	Columns
FK_Question_choices_Question	q_id->[dbo].[Question].[q_id]

```
CREATE TABLE [dbo].[Question_choices]

(
    [q_id] [int] NOT NULL,
    [choices] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
    [choice_num] [nchar] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Question_choices] ADD CONSTRAINT [PK_Question_choices_1] PRIMARY KEY
CLUSTERED ([q_id], [choices], [choice_num]) ON [PRIMARY]

GO
```

```
ALTER TABLE [dbo].[Question_choices] ADD CONSTRAINT [FK_Question_choices_Question]
FOREIGN KEY ([q_id]) REFERENCES [dbo].[Question] ([q_id])
GO
```

[dbo].[Question]

Used By

[dbo].[choice_delete_by_id]
[dbo].[choice_update]
[dbo].[create_Exam]
[dbo].[exam_qustions_choices]
[dbo].[insert_choices]
[dbo].[qChoice_select_by_id]

[dbo].[Student]

Properties

Property	Value		
Collation	tion SQL_Latin1_General_CP1_CI_AS		
Row Count (~)	7		
Created	11:51:08 PM Tuesday, February 21, 2023		
Last Modified	4:18:27 PM Saturday, February 25, 2023		

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK	st_id	int	4	NOT NULL	1 - 1
	fname	varchar(50)	50	NULL allowed	
	Iname	varchar(50)	50	NULL allowed	
	st_gender	varchar(3)	3	NULL allowed	
	st_phone	varchar(11)	11	NULL allowed	
	st_birthdate	date	3	NULL allowed	
	st_joindate	date	3	NULL allowed	
FK	dept_id	int	4	NULL allowed	
	city	varchar(50)	50	NULL allowed	
	street	varchar(50)	50	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
PKP C	PK_Student	st_id	True

Foreign Keys

Name	Columns
FK_Student_Department	dept_id->[dbo].[Department].[dept_id]

SQL Script

CREATE TABLE [dbo].[Student]

```
[st id] [int] NOT NULL IDENTITY(1, 1),
[fname] [varchar] (50) COLLATE SQL Latin1 General CP1 CI AS NULL,
[lname] [varchar] (50) COLLATE SQL Latin1 General CP1 CI AS NULL,
[st gender] [varchar] (3) COLLATE SQL Latin1 General CP1 CI AS NULL,
[st_phone] [varchar] (11) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[st birthdate] [date] NULL,
[st_joindate] [date] NULL,
[dept id] [int] NULL,
[city] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[street] \ [varchar] \ (50) \ COLLATE \ SQL\_Latin1\_General\_CP1\_CI\_AS \ {\tt NULL}
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Student] ADD CONSTRAINT [PK Student] PRIMARY KEY CLUSTERED ([st id])
ON [PRIMARY]
ALTER TABLE [dbo].[Student] ADD CONSTRAINT [FK Student Department] FOREIGN KEY
([dept id]) REFERENCES [dbo].[Department] ([dept id])
GO
```

[dbo].[Department]

```
[dbo].[Student_course]
[dbo].[Student_exams_questions]
[dbo].[display_st_report_deprt]
[dbo].[get_Question_with_Student_answer]
[dbo].[report_st_crs]
[dbo].[st_ex_qs_insert]
[dbo].[st_ex_qs_update]
[dbo].[student_answers]
[dbo].[student_course_update_sp]
[dbo].[student_delete_sp]
[dbo].[student_insert_sp]
[dbo].[student_select_by_id_sp]
[dbo].[student_select_sp]
[dbo].[student_update_sp]
```

[dbo].[Student_course]

Properties

Property	Value
Row Count (~)	6
Created	11:51:08 PM Tuesday, February 21, 2023
Last Modified	11:51:08 PM Tuesday, February 21, 2023

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PKO FKO	course_id	int	4	NOT NULL
PKO FKO	st_id	int	4	NOT NULL
	grade	int	4	NULL allowed

Indexes

Key	Name	Key Columns	Unique
PK G	PK_Student_course	course_id, st_id	True

Foreign Keys

Name	Columns
FK_Student_course_Course	course_id->[dbo].[Course].[courses_id]
FK_Student_course_Student	st_id->[dbo].[Student].[st_id]

```
CREATE TABLE [dbo].[Student_course]

(
[course_id] [int] NOT NULL,

[st_id] [int] NOT NULL,

[grade] [int] NULL

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Student_course] ADD CONSTRAINT [PK_Student_course] PRIMARY KEY

CLUSTERED ([course_id], [st_id]) ON [PRIMARY]
```

```
ALTER TABLE [dbo].[Student_course] ADD CONSTRAINT [FK_Student_course_Course] FOREIGN
KEY ([course_id]) REFERENCES [dbo].[Course] ([courses_id])

GO
ALTER TABLE [dbo].[Student_course] ADD CONSTRAINT [FK_Student_course_Student] FOREIGN
KEY ([st_id]) REFERENCES [dbo].[Student] ([st_id])

GO
```

[dbo].[Course] [dbo].[Student]

Used By

[dbo].[ExamCorrection_calc_std_garde_sp]
[dbo].[instr_id]
[dbo].[report_st_crs]
[dbo].[st_crs_delete_by_cr_id]
[dbo].[st_crs_delete_by_id]
[dbo].[std_cs_select_by_id]
[dbo].[std_cs_select_sp]
[dbo].[student_Course_insert_sp]
[dbo].[student_course_update_sp]

[dbo].[Student_exams_questions]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	3
Created	4:18:27 PM Saturday, February 25, 2023
Last Modified	4:18:47 PM Saturday, February 25, 2023

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Default
PKO FKO	st_id	int	4	NOT NULL	
PKO FKO	exam_id	int	4	NOT NULL	
PKO FKO	qs_id	int	4	NOT NULL	
	grade	int	4	NULL allowed	
	answer	varchar(10)	10	NULL allowed	
	is_corrected	int	4	NULL allowed	((0))

Indexes

Key	Name	Key Columns	Unique
PK G	PK_Student_exams_questions	st_id, exam_id, qs_id	True

Foreign Keys

Name	Columns
FK_Student_exams_questions_Exam	exam_id->[dbo].[Exam].[exam_id]
FK_Student_exams_questions_Question	qs_id->[dbo].[Question].[q_id]
FK_Student_exams_questions_Student	st_id->[dbo].[Student].[st_id]

SQL Script

CREATE TABLE [dbo].[Student exams questions]

```
[st id] [int] NOT NULL,
[exam id] [int] NOT NULL,
[qs id] [int] NOT NULL,
[grade] [int] NULL,
[answer] [varchar] (10) COLLATE SQL Latin1 General CP1 CI AS NULL,
[is_corrected] [int] NULL CONSTRAINT [DF_Student_exams_questions_is_corrected] DEFAULT
((0))
ON [PRIMARY]
ALTER TABLE [dbo].[Student_exams_questions] ADD CONSTRAINT [PK Student exams questions]
PRIMARY KEY CLUSTERED ([st_id], [exam_id], [qs_id]) ON [PRIMARY]
ALTER TABLE [dbo].[Student_exams_questions] ADD CONSTRAINT [FK_Student_exams_questions_-
Exam] FOREIGN KEY ([exam_id]) REFERENCES [dbo].[Exam] ([exam_id])
ALTER TABLE [dbo]. [Student exams questions] ADD CONSTRAINT [FK Student exams questions -
Question] FOREIGN KEY ([qs_id]) REFERENCES [dbo].[Question] ([q_id])
GO
ALTER TABLE [dbo].[Student exams questions] ADD CONSTRAINT [FK Student exams questions -
Student] FOREIGN KEY ([st_id]) REFERENCES [dbo].[Student] ([st_id])
```

[dbo].[Exam]

[dbo].[Question]

[dbo].[Student]

```
[dbo].[correct_exam]
[dbo].[delete_student_exams_Questions_by_id]
[dbo].[exam_qustions_choices]
[dbo].[ExamCorrection_calc_std_garde_sp]
[dbo].[get_Question_with_Student_answer]
[dbo].[get_student_answers]
[dbo].[st_ex_qs_insert]
[dbo].[st_ex_qs_select_all]
[dbo].[st_ex_qs_select_by_ids]
[dbo].[st_ex_qs_update]
[dbo].[student_answers]
```

■ [dbo].[Topic]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	6
Created	11:51:08 PM Tuesday, February 21, 2023
Last Modified	11:51:08 PM Tuesday, February 21, 2023

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK	topic_id	int	4	NOT NULL	1 - 1
	topic_name	varchar(50)	50	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
PK G	PK_Topic	topic_id	True

SQL Script

```
CREATE TABLE [dbo].[Topic]

(
[topic_id] [int] NOT NULL IDENTITY(1, 1),

[topic_name] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Topic] ADD CONSTRAINT [PK_Topic] PRIMARY KEY CLUSTERED ([topic_id])

ON [PRIMARY]

GO
```

```
[dbo].[Course]
[dbo].[delete_topic_by_id]
[dbo].[get_topic_name_sp]
[dbo].[topic_insert]
[dbo].[topic_select_all]
```

Project > (local)\ > User databases > examination_system > Tables > dbo.Topic

[dbo].[topic_select_by_id] [dbo].[topic_update]

Stored Procedures

Objects

Name
dbo.choice_delete_by_id
dbo.choice_update
dbo.correct_exam
dbo.coures_select_by_id
dbo.coures_select_sp
dbo.couress_delete_by_courese
dbo.course_update_sp
dbo.create_Exam
dbo.delete_exam_by_id
dbo.delete_student_exams_Questions_by_id
dbo.delete_topic_by_id
dbo.department_delete_sp
dbo.department_insert_sp
dbo.department_select_by_id_sp
dbo.department_select_sp
dbo.department_update_sp
dbo.display_st_report_deprt
dbo.exam_insert
dbo.exam_qustions_choices
dbo.exam_select_all
dbo.exam_select_by_id
dbo.exam_update
dbo.ExamCorrection_calc_std_garde_sp
dbo.get_Question_with_Student_answer
dbo.get_student_answers
dbo.get_topic_name_sp
dbo.insert_choices
dbo.insert_courese_sp
dbo.insert_question
dbo.instr_id
dbo.instructor_course_insert_sp
dbo.instructor_course_select_sp
dbo.instructor_course_update_sp

dbo.instructor_courses_delete_by_insld_sp dbo.instructor_insert_sp dbo.instructor_select_by_iourse_id_sp dbo.instructor_select_by_id_sp dbo.instructor_select_by_instructor_id_sp dbo.instructor_select_by_instructor_id_sp dbo.instructor_select_by_instructor_id_sp dbo.instructor_update_sp dbo.instructor_update_sp dbo.question_delete_by_id dbo.question_delete_by_id dbo.question_update dbo.question_update dbo.uper_id_delete_by_id dbo.st_crs_delete_by_ir_did dbo.st_crs_delete_by_ir_did dbo.st_crs_delete_by_id dbo.st_crs_delete_by_id dbo.st_crs_delete_by_id dbo.st_crs_delete_by_id dbo.st_crs_qs_select_all dbo.st_crs_qs_update dbo.st_crs_select_by_id dbo.st_crs_select_by_id_sp dbo.student_course_update_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.topic_insert dbo.topic_select_all dbo.topic_select_all dbo.topic_select_by_id dbo.topic_select_by_id dbo.topic_select_by_id dbo.topic_select_by_id dbo.topic_update	dbo.instructor_courses_delete_by_crsld_sp
dbo.instructor_insert_sp dbo.instructor_select_by_course_id_sp dbo.instructor_select_by_id_sp dbo.instructor_select_by_instructor_id_sp dbo.instructor_select_by_instructor_id_sp dbo.instructor_update_sp dbo.question_delete_by_id dbo.question_select dbo.question_select dbo.question_select_by_id dbo.question_select_by_id dbo.question_select_by_id dbo.question_select_by_id dbo.question_select_by_id dbo.question_select_by_id dbo.st_crs_delete_by_cr_id dbo.st_crs_delete_by_cr_id dbo.st_crs_delete_by_id dbo.st_ex_qs_insert dbo.st_ex_qs_insert dbo.st_ex_qs_select_all dbo.st_ex_qs_select_by_ids dbo.st_ex_qs_select_by_id dbo.st_crs_select_by_id dbo.st_crs_select_by_id dbo.st_crs_select_sp dbo.student_course_update_sp dbo.student_course_update_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.student_update_sp dbo.topic_insert dbo.topic_select_all dbo.topic_select_all	dbo.instructor_courses_delete_by_insld_sp
dbo.instructor_select_by_id_sp dbo.instructor_select_by_instructor_id_sp dbo.instructor_select_sp dbo.instructor_update_sp dbo.qChoice_select_by_id dbo.question_delete_by_id dbo.question_select_dby_id dbo.question_select_by_id dbo.question_select_by_id dbo.question_select_by_id dbo.question_select_by_id dbo.question_select_by_id dbo.question_select_by_id dbo.st_crs_delete_by_cr_id dbo.st_crs_delete_by_rd dbo.st_crs_delete_by_id dbo.st_ex_qs_insert dbo.st_ex_qs_select_all dbo.st_ex_qs_select_by_ids dbo.st_ex_qs_select_by_id dbo.st_ex_qs_select_by_id dbo.st_crs_delete_by_id dbo.st_crs_delete_by_id dbo.st_ex_qs_select_by_id dbo.st_ex_qs_select_by_id dbo.st_ex_qs_select_by_id dbo.st_ex_qs_select_by_id dbo.st_ex_qs_select_by_id dbo.st_ex_qs_select_sp dbo.student_course_insert_sp dbo.student_course_insert_sp dbo.student_delete_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.student_select_sp dbo.student_select_sp dbo.topic_insert dbo.topic_select_all dbo.topic_select_all	dbo.instructor_delete_sp
dbo.instructor_select_by_instructor_id_sp dbo.instructor_select_sp dbo.instructor_update_sp dbo.qChoice_select_by_id dbo.question_delete_by_id dbo.question_select dbo.question_select_by_id dbo.question_select_by_id dbo.question_select_by_id dbo.question_select_by_id dbo.question_select_by_id dbo.question_select_by_id dbo.st_crs_delete_by_cr_id dbo.st_crs_delete_by_rd dbo.st_crs_delete_by_id dbo.st_ex_qs_select_all dbo.st_ex_qs_select_by_ids dbo.st_ex_qs_select_by_ids dbo.st_ex_qs_select_by_id dbo.st_ex_qs_select_by_id dbo.st_cs_select_by_id dbo.st_cs_select_by_id dbo.std_cs_select_sp dbo.student_course_insert_sp dbo.student_course_update_sp dbo.student_felete_sp dbo.student_jelect_by_id_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.student_update_sp dbo.topic_insert dbo.topic_select_bly_id dbo.topic_select_bly_id	dbo.instructor_insert_sp
dbo.instructor_select_by_instructor_id_sp dbo.instructor_update_sp dbo.qChoice_select_by_id dbo.question_delete_by_id dbo.question_select dbo.question_select_by_id dbo.question_update dbo.question_update dbo.question_update dbo.question_update dbo.st_ors_delete_by_or_id dbo.question_update dbo.st_ors_delete_by_or_id dbo.st_ors_delete_by_id dbo.st_ex_qs_insert dbo.st_ex_qs_select_all dbo.st_ex_qs_select_by_ids dbo.st_ex_qs_select_by_ids dbo.st_ex_qs_update dbo.st_ors_select_by_id dbo.st_ors_select_sp dbo.student_course_update_sp dbo.student_course_update_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.student_select_sp dbo.student_update_sp dbo.student_update_sp dbo.topic_insert dbo.topic_select_all dbo.topic_select_all	dbo.instructor_select_by_course_id_sp
dbo.instructor_select_sp dbo.qChoice_select_by_id dbo.question_select dbo.question_select_by_id dbo.question_select_by_id dbo.question_select_by_id dbo.question_update dbo.report_st_crs dbo.st_crs_delete_by_cr_id dbo.st_crs_delete_by_cr_id dbo.st_crs_delete_by_id dbo.st_ex_qs_insert dbo.st_ex_qs_select_all dbo.st_ex_qs_select_by_ids dbo.st_ex_qs_select_by_ids dbo.st_crs_delete_by_id dbo.st_crs_delete_by_id dbo.st_crs_delete_by_id dbo.st_crs_select_by_ids dbo.st_crs_select_by_ids dbo.st_crs_select_by_ids dbo.st_crs_select_by_id dbo.st_crs_select_by_id dbo.st_crs_select_sp dbo.student_course_insert_sp dbo.student_course_update_sp dbo.student_course_update_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.student_update_sp dbo.student_update_sp dbo.topic_insert dbo.topic_select_all dbo.topic_select_all	dbo.instructor_select_by_id_sp
dbo.nstructor_update_sp dbo.qChoice_select_by_id dbo.question_select dbo.question_select dbo.question_select_by_id dbo.question_select_by_id dbo.question_update dbo.report_st_crs dbo.st_crs_delete_by_or_id dbo.st_crs_delete_by_id dbo.st_ex_qs_insert dbo.st_ex_qs_insert dbo.st_ex_qs_select_all dbo.st_ex_qs_select_by_ids dbo.st_ex_qs_select_by_id dbo.st_ex_qs_select_by_id dbo.st_ex_qs_select_by_id dbo.st_ex_qs_select_sp dbo.std_cs_select_sp dbo.stdoent_course_insert_sp dbo.student_course_update_sp dbo.student_clelete_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.student_update_sp dbo.student_update_sp dbo.student_update_sp dbo.topic_insert dbo.topic_select_all dbo.topic_select_by_id	dbo.instructor_select_by_instructor_id_sp
dbo.qChoice_select_by_id dbo.question_select dbo.question_select_by_id dbo.question_select_by_id dbo.question_update dbo.report_st_crs dbo.st_crs_delete_by_or_id dbo.st_crs_delete_by_id dbo.st_ex_qs_insert dbo.st_ex_qs_select_all dbo.st_ex_qs_select_by_ids dbo.st_ex_qs_update dbo.st_cs_select_by_id dbo.st_ex_qs_select_by_id dbo.st_cs_select_by_id dbo.st_cs_select_sp dbo.stdent_answers dbo.student_course_insert_sp dbo.student_course_update_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.student_update_sp dbo.student_update_sp dbo.topic_insert dbo.topic_select_all dbo.topic_select_by_id	dbo.instructor_select_sp
dbo.question_select dbo.question_select_by_id dbo.question_update dbo.report_st_crs dbo.st_crs_delete_by_cr_id dbo.st_crs_delete_by_cr_id dbo.st_ex_qs_insert dbo.st_ex_qs_select_all dbo.st_ex_qs_select_by_ids dbo.st_ex_qs_select_by_id dbo.st_ex_qs_select_by_ids dbo.st_ex_qs_select_by_id dbo.st_crs_select_by_id dbo.st_crs_select_by_id dbo.st_crs_select_by_id dbo.st_crs_select_by_id dbo.st_dc_select_sp dbo.student_answers dbo.student_course_insert_sp dbo.student_course_update_sp dbo.student_telete_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.student_update_sp dbo.student_update_sp dbo.student_update_sp dbo.student_update_sp dbo.student_update_sp dbo.student_update_sp dbo.student_update_sp dbo.student_update_sp dbo.topic_insert dbo.topic_select_all dbo.topic_select_by_id	dbo.instructor_update_sp
dbo.question_select_by_id dbo.question_update dbo.report_st_crs dbo.st_crs_delete_by_cr_id dbo.st_crs_delete_by_id dbo.st_ex_qs_insert dbo.st_ex_qs_select_all dbo.st_ex_qs_select_by_ids dbo.st_ex_qs_update dbo.std_cs_select_by_id dbo.std_cs_select_by_id dbo.std_cs_select_by_id dbo.std_cs_select_sp dbo.stduent_answers dbo.student_course_insert_sp dbo.student_course_update_sp dbo.student_telete_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.student_update_sp dbo.topic_insert dbo.topic_select_all dbo.topic_select_by_id	dbo.qChoice_select_by_id
dbo.question_select_by_id dbo.quetion_update dbo.st_crs_delete_by_cr_id dbo.st_crs_delete_by_id dbo.st_ex_qs_insert dbo.st_ex_qs_select_all dbo.st_ex_qs_select_by_ids dbo.st_ex_qs_update dbo.st_cs_select_by_id dbo.std_cs_select_by_id dbo.std_cs_select_sp dbo.student_answers dbo.student_course_insert_sp dbo.student_delete_sp dbo.student_linsert_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.student_select_sp dbo.topic_insert dbo.topic_select_all dbo.topic_select_by_id	dbo.question_delete_by_id
dbo.quetion_update dbo.report_st_crs dbo.st_crs_delete_by_cr_id dbo.st_crs_delete_by_id dbo.st_ex_qs_insert dbo.st_ex_qs_select_all dbo.st_ex_qs_select_by_ids dbo.st_ex_qs_select_by_id dbo.st_ex_qs_update dbo.st_cs_select_by_id dbo.st_cs_select_sp dbo.student_answers dbo.student_Course_insert_sp dbo.student_course_update_sp dbo.student_delete_sp dbo.student_insert_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.student_select_sp dbo.student_update_sp dbo.topic_insert dbo.topic_select_all dbo.topic_select_by_id	dbo.question_select
dbo.report_st_crs dbo.st_crs_delete_by_cr_id dbo.st_crs_delete_by_id dbo.st_ex_qs_insert dbo.st_ex_qs_select_all dbo.st_ex_qs_select_by_ids dbo.st_ex_qs_update dbo.st_ex_qs_update dbo.std_cs_select_by_id dbo.std_cs_select_sp dbo.student_answers dbo.student_Course_insert_sp dbo.student_course_update_sp dbo.student_delet_sp dbo.student_insert_sp dbo.student_insert_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.topic_insert dbo.topic_insert	dbo.question_select_by_id
dbo.st_crs_delete_by_cr_id dbo.st_crs_delete_by_id dbo.st_ex_qs_insert dbo.st_ex_qs_select_all dbo.st_ex_qs_select_by_ids dbo.st_ex_qs_update dbo.st_cs_select_by_id dbo.std_cs_select_by_id dbo.std_cs_select_sp dbo.student_answers dbo.student_Course_insert_sp dbo.student_course_update_sp dbo.student_delete_sp dbo.student_insert_sp dbo.student_insert_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.student_select_sp dbo.topic_insert dbo.topic_insert dbo.topic_select_all dbo.topic_select_by_id	dbo.quetion_update
dbo.st_crs_delete_by_id dbo.st_ex_qs_insert dbo.st_ex_qs_select_all dbo.st_ex_qs_select_by_ids dbo.st_ex_qs_update dbo.std_cs_select_by_id dbo.std_cs_select_by_id dbo.std_cs_select_sp dbo.student_answers dbo.student_Course_insert_sp dbo.student_course_update_sp dbo.student_delete_sp dbo.student_insert_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.student_update_sp dbo.student_update_sp dbo.topic_insert dbo.topic_insert dbo.topic_select_all dbo.topic_select_by_id	dbo.report_st_crs
dbo.st_ex_qs_insert dbo.st_ex_qs_select_all dbo.st_ex_qs_select_by_ids dbo.st_ex_qs_update dbo.std_cs_select_by_id dbo.std_cs_select_sp dbo.student_answers dbo.student_Course_insert_sp dbo.student_course_update_sp dbo.student_insert_sp dbo.student_insert_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.topic_insert dbo.topic_insert	dbo.st_crs_delete_by_cr_id
dbo.st_ex_qs_select_by_ids dbo.st_ex_qs_update dbo.st_ex_qs_update dbo.std_cs_select_by_id dbo.std_cs_select_sp dbo.student_answers dbo.student_Course_insert_sp dbo.student_course_update_sp dbo.student_delete_sp dbo.student_insert_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.student_select_sp	dbo.st_crs_delete_by_id
dbo.st_ex_qs_select_by_ids dbo.st_ex_qs_update dbo.std_cs_select_by_id dbo.std_cs_select_sp dbo.student_answers dbo.student_Course_insert_sp dbo.student_course_update_sp dbo.student_delete_sp dbo.student_insert_sp dbo.student_insert_sp dbo.student_select_by_id_sp dbo.student_select_by_id_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.topic_insert dbo.topic_insert dbo.topic_select_all dbo.topic_select_by_id	dbo.st_ex_qs_insert
dbo.std_cs_select_by_id dbo.std_cs_select_sp dbo.student_answers dbo.student_Course_insert_sp dbo.student_course_update_sp dbo.student_delete_sp dbo.student_insert_sp dbo.student_select_by_id_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.topic_insert dbo.topic_select_all dbo.topic_select_by_id	dbo.st_ex_qs_select_all
dbo.std_cs_select_sp dbo.student_answers dbo.student_Course_insert_sp dbo.student_course_update_sp dbo.student_delete_sp dbo.student_insert_sp dbo.student_select_by_id_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.student_select_sp dbo.topic_select_all dbo.topic_select_by_id dbo.topic_select_by_id	dbo.st_ex_qs_select_by_ids
dbo.std_cs_select_sp dbo.student_answers dbo.student_Course_insert_sp dbo.student_course_update_sp dbo.student_delete_sp dbo.student_insert_sp dbo.student_select_by_id_sp dbo.student_select_sp dbo.student_select_sp dbo.student_update_sp dbo.topic_insert dbo.topic_select_all dbo.topic_select_by_id	dbo.st_ex_qs_update
dbo.student_answers dbo.student_Course_insert_sp dbo.student_course_update_sp dbo.student_delete_sp dbo.student_insert_sp dbo.student_select_by_id_sp dbo.student_select_sp dbo.student_select_sp dbo.student_update_sp dbo.topic_insert dbo.topic_insert dbo.topic_select_all dbo.topic_select_by_id	dbo.std_cs_select_by_id
dbo.student_Course_insert_sp dbo.student_course_update_sp dbo.student_delete_sp dbo.student_insert_sp dbo.student_select_by_id_sp dbo.student_select_sp dbo.student_update_sp dbo.topic_insert dbo.topic_select_all dbo.topic_select_by_id	dbo.std_cs_select_sp
dbo.student_course_update_sp dbo.student_delete_sp dbo.student_insert_sp dbo.student_select_by_id_sp dbo.student_select_sp dbo.student_update_sp dbo.topic_insert dbo.topic_select_all dbo.topic_select_by_id	dbo.student_answers
dbo.student_delete_sp dbo.student_insert_sp dbo.student_select_by_id_sp dbo.student_select_sp dbo.student_update_sp dbo.topic_insert dbo.topic_select_all dbo.topic_select_by_id	dbo.student_Course_insert_sp
dbo.student_insert_sp dbo.student_select_by_id_sp dbo.student_select_sp dbo.student_update_sp dbo.topic_insert dbo.topic_select_all dbo.topic_select_by_id	dbo.student_course_update_sp
dbo.student_select_by_id_sp dbo.student_select_sp dbo.student_update_sp dbo.topic_insert dbo.topic_select_all dbo.topic_select_by_id	dbo.student_delete_sp
dbo.student_select_sp dbo.student_update_sp dbo.topic_insert dbo.topic_select_all dbo.topic_select_by_id	dbo.student_insert_sp
dbo.student_update_sp dbo.topic_insert dbo.topic_select_all dbo.topic_select_by_id	dbo.student_select_by_id_sp
dbo.topic_insert dbo.topic_select_all dbo.topic_select_by_id	dbo.student_select_sp
dbo.topic_select_all dbo.topic_select_by_id	dbo.student_update_sp
dbo.topic_select_by_id	dbo.topic_insert
	dbo.topic_select_all
dbo.topic_update	dbo.topic_select_by_id
	dbo.topic_update

```
[dbo].[choice_delete_by_id]
```

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@q_id	int	4
@c_id	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[choice_delete_by_id] @q_id INT, @c_id INT

AS

BEGIN

BEGIN TRY

DELETE FROM Question_choices WHERE q_id=@q_id AND choices=@c_id

END TRY

BEGIN CATCH

PRINT 'You cannot delete this Question Choice'

END CATCH

END

GO
```

Uses

[dbo].[Question_choices]

[dbo].[choice_update]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@q_id	int	4
@old_choice	varchar(30)	30
@new_choice	varchar(30)	30

SQL Script

```
CREATE PROCEDURE [dbo].[choice update] @q id INT, @old choice VARCHAR(30), @new choice
VARCHAR (30)
AS
BEGIN
  IF @q_id IN (SELECT q_id FROM Question_choices) AND @old_choice IN (SELECT choices
FROM Question_choices)
       BEGIN
           UPDATE Question choices
           SET choices = @new_choice
           WHERE choices = @old_choice AND q_id = @q_id
       END
   ELSE
       BEGIN
           PRINT('Selected id not exist, check question id and old choice value')
        END
END
GO
```

Uses

[dbo].[Question_choices]

[dbo].[correct_exam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@st_id	int	4
@exam_id	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[correct exam] @st id INT, @exam id INT
AS
BEGIN
   IF (select top(1) is_corrected from Student_exams_questions where exam_id=@exam_id
and st_id=@st_id)=0
   BEGIN
         EXEC get student answers @st id,@exam id;
         UPDATE dbo.Student exams questions
         SET Student exams questions.grade = CASE WHEN seq.answer = q.q answer THEN
q.q_grade ELSE 0 END
         FROM dbo.Student exams questions seq
         JOIN question q ON seq.qs_id = q.q_id
         WHERE seq.st_id = @st_id AND seq.exam_id =@exam_id;
         update dbo.Student exams questions
         SET is corrected=1
         WHERE exam_id=@exam_id and st_id=@st_id
         exec ExamCorrection calc std garde sp @st id,@exam id
         PRINT 'exam for this student is corrected'
   END
   ELSE
       PRINT 'exam for this student is already corrected'
END
GO
```

Project > (local)\ > User databases > examination_system > Programmability > Stored Procedures > dbo.correct_exam

Uses

[dbo].[Question]
[dbo].[Student_exams_questions]
[dbo].[ExamCorrection_calc_std_garde_sp]
[dbo].[get_student_answers]

[dbo].[coures_select_by_id]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@id	int	4

SQL Script

```
CREATE PROC [dbo].[coures_select_by_id] @id int

AS

BEGIN

SELECT * FROM Course c

WHERE c.courses_id = @id

END

GO
```

Uses

[dbo].[coures_select_sp]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE PROC [dbo].[coures_select_sp]

AS

BEGIN

SELECT * FROM Course c

END

GO
```

Uses

[dbo].[couress_delete_by_courese]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@coures_id	varchar(50)	50

SQL Script

```
CREATE PROCEDURE [dbo].[couress_delete_by_courese] @coures_id VARCHAR(50)

AS

BEGIN

BEGIN TRY

DELETE FROM Course WHERE courses_id = @coures_id

END TRY

BEGIN CATCH

PRINT 'You cannot delete this course'

END CATCH

END

GO
```

Uses

[dbo].[course_update_sp]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@crs_id	int	4
@crs_name	nvarchar(50)	100
@crs_duration	int	4

SQL Script

```
create proc [dbo].[course update sp] @crs id int, @crs name nvarchar(50),
@crs_duration int
as
begin
   if @crs_id not in (select courses_id from Course)
           if @crs_name not in (select course_name from Course)
           insert into course
           values (@crs id,@crs name,@crs duration)
           print 'Course name already exists'
       end
   else
   begin
       if @crs name not in (select course name from Course)
           update Course
           set course_name = @crs_name , duration=@crs_duration
           where courses_id=@crs_id
       else
           update Course
           \verb"set" duration=@crs_duration"
           where courses_id=@crs_id and course_name = @crs_name
       END
END
GO
```

Project > (local)\ > User databases > examination_system > Programmability > Stored Procedures >	
dbo.course_update_sp	

Uses

[dbo].[create_Exam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@crs_name	varchar(50)	50
@t_f	int	4
@mcq	int	4

SQL Script

```
CREATE PROC [dbo].[create Exam] @crs name VARCHAR(50) ,@t f INT , @mcq INT
BEGIN
   IF EXISTS (SELECT * FROM Course WHERE course name = @crs name )
       BEGIN
           IF (@t f > 0) AND (@t f <= (
               SELECT COUNT (q id) q id
                {\tt FROM} Question q , Course c
                WHERE q.q type = 'tf' AND q.courses id = c.courses id AND c.course name
= @crs_name ))
               BEGIN
                   IF (@mcq > 0) AND (@mcq <= (</pre>
                        SELECT COUNT (q id) q id
                        FROM Question q , Course c
                        WHERE q.q type = 'choice' AND q.courses id = c.courses id AND
c.course_name = @crs_name))
                        BEGIN
                            DECLARE @exam table TABLE (ex id INT)
                            INSERT INTO @exam_table
                            SELECT * FROM
                            (SELECT TOP(@t f) q.q id
                            FROM Question q JOIN Course c
                            ON c.courses_id = q.courses_id AND c.course_name =@crs_name
AND q.q type = 't f'
                            order by NEWID()
                            UNION
```

```
SELECT TOP(@mcq) q id
                            FROM Question q JOIN Course c
                            ON c.courses_id = q.courses_id AND c.course_name =@crs_name
AND q.q type = 'choice'
                            ORDER BY NEWID()) exam_table
                            SELECT *
                            FROM (
                                SELECT q.question, qc.choices, ROW NUMBER() OVER
(PARTITION BY qc.q_id ORDER BY qc.q_id ) AS choice_num
                                FROM Question choices qc
                                JOIN Question q ON q.q_id = qc.q_id AND q.q_id IN
(SELECT * FROM @exam table)
                                ) AS exam_table2
                                PIVOT (
                                    MAX(exam_table2.choices) FOR exam_table2.choice_num
IN ([1], [2], [3], [4])
                                ) AS pivot_table;
                        END
                    ELSE
                        PRINT 'please enter a valid mcq number'
                END
            ELSE
                PRINT 'please enter a valid t_f number'
        END
    ELSE
        PRINT 'This course is not Exist'
END
GO
```

Uses

[dbo].[Course] [dbo].[Question] [dbo].[Question_choices]

```
[dbo].[delete_exam_by_id]
```

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@exam_id	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[delete_exam_by_id] @exam_id INT

AS

BEGIN

BEGIN TRY

DELETE FROM Exam WHERE exam_id = @exam_id

END TRY

BEGIN CATCH

PRINT 'You cannot delete this exam'

END CATCH

END

GO
```

Uses

[dbo].[delete_student_exams_Questions_by_id]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@st_id	int	4
@exam_id	int	4
@qs_id	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[delete_student_exams_Questions_by_id] @st_id INT , @exam_id INT , @qs_id INT , @qs_id INT AS

BEGIN

BEGIN TRY

DELETE FROM student_exams_Questions WHERE st_id =@st_id AND exam_id = @exam_id AND qs_id =@qs_id

END TRY

BEGIN CATCH

PRINT 'You cannot delete this student exam'

END CATCH

END

GO
```

Uses

 $[dbo]. [Student_exams_questions] \\$

```
[dbo].[delete_topic_by_id]
```

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@topic_id	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[delete_topic_by_id] @topic_id INT

AS

BEGIN

BEGIN TRY

DELETE FROM Topic WHERE topic_id = @topic_id

END TRY

BEGIN CATCH

PRINT 'You cannot delete this topic'

END CATCH

END

GO
```

Uses

[dbo].[Topic]

[dbo].[department_delete_sp]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@dept_id	int	4

SQL Script

```
CREATE PROC [dbo].[department_delete_sp] @dept_id INT

AS

BEGIN

BEGIN TRY

DELETE FROM Department WHERE dept_id = @dept_id

END TRY

BEGIN CATCH

PRINT 'You cannot delete this Deprtment'

END CATCH

END

GO
```

Uses

[dbo].[department_insert_sp]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@dept_name	varchar(50)	50
@dept_manager	int	4

SQL Script

```
CREATE PROC [dbo].[department_insert_sp] @dept_name VARCHAR(50),@dept_manager INT
AS
BEGIN
   IF @dept manager IN (SELECT insrt id FROM Instructor i)
       BEGIN
          BEGIN TRY
              INSERT INTO Department
              VALUES (@dept_manager , @dept_name)
           END TRY
           BEGIN CATCH
                PRINT 'Please check your data'
           END CATCH
       END
   ELSE
       BEGIN
           PRINT 'Department Manager id that you try to enter is not exist '
       END
END
GO
```

Uses

[dbo].[Department] [dbo].[Instructor]

[dbo].[department_select_by_id_sp]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@dept_id	int	4

SQL Script

```
CREATE PROC [dbo].[department_select_by_id_sp] @dept_id INT

AS

BEGIN

SELECT * FROM Department d WHERE d.dept_id = @dept_id

END

GO
```

Uses

[dbo].[department_select_sp]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE PROC [dbo].[department_select_sp]
AS
BEGIN
SELECT * FROM Department d
END
GO
```

Uses

[dbo].[department_update_sp]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@dept_id	int	4
@dept_name	varchar(50)	50
@dept_manager	int	4

SQL Script

```
CREATE PROC [dbo].[department_update_sp] @dept_id INT , @dept_name VARCHAR(50),
@dept_manager INT
AS
BEGIN
   IF @dept_id IN (SELECT dept_id FROM Department d)
          UPDATE Department
           SET dept_manager = @dept_manager,
              dept name = @dept name
           WHERE dept id=@dept id
       END
   ELSE
       BEGIN
        PRINT 'This Department id is not exist'
       END
END
GO
```

Uses

[dbo].[display_st_report_deprt]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@id	int	4

SQL Script

```
CREATE PROC [dbo].[display_st_report_deprt] @id int
 AS
BEGIN
 IF @id IN (SELECT d.dept id FROM Department d)
               BEGIN
                    SELECT
 \verb|s.st| id, \verb|s.fname|, \verb|s.st| pender|, \verb|s.st| phone|, \verb|s.st| birthdate|, \verb|s.st| joindate|, \verb|s.st| id, \verb|s.fname|, \verb|s.st| joindate|, \verb|s.st| id, \verb|s.fname|, \verb|s.st| joindate|, s.st| j
 d.dept_name
                                                                                   FROM Student s INNER JOIN Department d
                                                                                 ON s.dept id = d.dept id
                                                                                  WHERE d.dept id = @id
             END
             ELSE
                       BEGIN
                          PRINT 'Deprtment Id is not Exist'
                           END
 END
 GO
```

Uses

[dbo].[Department] [dbo].[Student]

[dbo].[exam_insert]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@ex_date	date	3

SQL Script

```
CREATE PROCEDURE [dbo].[exam_insert] @ex_date DATE

AS

BEGIN

BEGIN TRY

INSERT INTO Exam(ex_date)

VALUES (@ex_date)

END TRY

BEGIN CATCH

PRINT 'Please check your data'

END CATCH

END

GO
```

Uses

[dbo].[exam_qustions_choices]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@exam_id	int	4

SQL Script

```
CREATE PROC [dbo].[exam_qustions_choices] @exam_id INT

AS

BEGIN

IF @exam_id IN (SELECT e.exam_id FROM Exam e)

BEGIN

SELECT q.question, qc.choices

FROM (Student_exams_questions seq JOIN Question q

ON seq.qs_id = q.q_id) JOIN Question_choices qc

ON q.q_id = qc.q_id

WHERE seq.exam_id = @exam_id

END

ELSE

PRINT 'Exam id is not exist'

END

GO
```

Uses

```
[dbo].[Exam]
[dbo].[Question]
[dbo].[Question_choices]
[dbo].[Student_exams_questions]
```

[dbo].[exam_select_all]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE PROCEDURE [dbo].[exam_select_all]
AS
BEGIN
SELECT * FROM Exam ex
END
GO
```

Uses

[dbo].[exam_select_by_id]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@exam_id	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[exam_select_by_id] @exam_id INT

AS

BEGIN

SELECT *

FROM Exam

WHERE exam_id =@exam_id

END

GO
```

Uses

[dbo].[exam_update]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@exam_id	int	4
@exam_date	date	3

SQL Script

```
CREATE PROCEDURE [dbo].[exam_update] @exam_id INT , @exam_date DATE

AS

BEGIN

IF @exam_id IN (SELECT @exam_id FROM Exam e )

BEGIN

UPDATE Exam

SET @exam_date = @exam_date

WHERE exam_id = @exam_id

END

ELSE

BEGIN

PRINT 'this exam is not exist'

END

END

END

END

GO
```

Uses

[dbo].[ExamCorrection_calc_std_garde_sp]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@std_id	int	4
@exam_id	int	4

SQL Script

```
create proc [dbo].[ExamCorrection_calc_std_garde_sp] @std_id int,@exam_id int
as
begin
    declare @crs_id int
    declare @grade int

set @crs_id=(select top(1) courses_id
    from Question q,Student_exams_questions e
    WHERE q.q_id=e.qs_id and e.st_id=@std_id and e.exam_id=@exam_id)

set @grade = ( select sum([grade]) from Student_exams_questions where
st_id=@std_id and exam_id=@exam_id)
    update student_course
    set grade=grade+@grade
    where [st_id]=@std_id and [course_id]=@crs_id

END
GO
```

Uses

[dbo].[Question] [dbo].[Student_course] [dbo].[Student_exams_questions] Project > (local)\ > User databases > examination_system > Programmability > Stored Procedures > dbo.Exam-Correction_calc_std_garde_sp

Used By

[dbo].[correct_exam]

[dbo].[get_Question_with_Student_answer]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@exam_id	int	4
@st_id	int	4

SQL Script

```
CREATE PROC [dbo].[get Question with Student answer] @exam id INT, @st id INT
AS
BEGIN
   IF @exam id IN (SELECT exam id FROM Exam)
       BEGIN
           IF @st id IN (SELECT s.st id FROM Student s)
                   SELECT q.question, seq.answer AS [Student answer] ,q.q_answer
AS[Question answer]
                   FROM Student_exams_questions seq JOIN Question q
                   ON seq.qs id = q.q id
                   WHERE seq.exam_id = @exam_id AND seq.st_id = @st_id
           ELSE
              PRINT 'Student id is not exist'
       END
   ELSE
       PRINT 'Exam id is not exist'
END
GO
```

Uses

```
[dbo].[Exam]
[dbo].[Question]
[dbo].[Student]
[dbo].[Student_exams_questions]
Author:
```

Abdelrahman, Aya, Maha, Mayar, Nahl

[dbo].[get_student_answers]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@st_id	int	4
@exam_id	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[get_student_answers] @st_id INT, @exam_id INT

AS

BEGIN

SELECT q.q_id, seq.answer

FROM dbo.Student_exams_questions seq

JOIN dbo.Question q ON seq.qs_id = q.q_id

WHERE seq.st_id = @st_id AND seq.exam_id = @exam_id;

END

GO
```

Uses

[dbo].[Question]

[dbo].[Student_exams_questions]

Used By

[dbo].[correct_exam]

[dbo].[get_topic_name_sp]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@course_id	int	4

SQL Script

```
CREATE PROC [dbo].[get_topic_name_sp] @course_id INT

AS

BEGIN

IF @course_id IN (SELECT c.courses_id FROM Course c)

BEGIN

SELECT c.course_name,t.topic_name

FROM Course c, Topic t

WHERE c.topic_id = t.topic_id AND c.courses_id = @course_id

END

ELSE

PRINT 'Course id is not exist'

END

GO
```

Uses

[dbo].[Course] [dbo].[Topic]

[dbo].[insert_choices]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@q_id	int	4
@choice	varchar(30)	30
@choiceNum	nvarchar(10)	20

SQL Script

```
CREATE PROCEDURE [dbo].[insert choices] @q id INT, @choice VARCHAR(30) ,@choiceNum
NVARCHAR (10)
AS
BEGIN
   IF @q_id IN (SELECT q_id FROM Question)
       BEGIN
           BEGIN TRY
              INSERT INTO Question_choices (q_id,choices,choice_num)
              VALUES (@q id, @choice, @choiceNum)
           END TRY
           BEGIN CATCH
             PRINT 'Please check your data'
           END CATCH
       END
   ELSE
       BEGIN
           PRINT 'Question id you tried to enter is not exist'
       END
END
```

Uses

[dbo].[Question] [dbo].[Question_choices]

```
[dbo].[insert_courese_sp]
```

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@courses_name	varchar(50)	50
@courser_durtion	int	4
@topic_id	int	4

SQL Script

```
CREATE PROC [dbo].[insert_courese_sp] @courses_name VARCHAR(50), @courser_durtion INT
,@topic_id INT

AS

BEGIN

BEGIN TRY

INSERT INTO Course

VALUES (@courses_name,@courser_durtion,@topic_id)

END TRY

BEGIN CATCH

PRINT ' topic you try to enter is not exist'

END CATCH

END

GO
```

Uses

[dbo].[insert_question]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@question	varchar(50)	50
@q_type	varchar(10)	10
@grade	int	4
@answer	varchar(3)	3
@c_id	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[insert_question] @question VARCHAR(50), @q_type VARCHAR(10),
@grade INT, @answer VARCHAR(3), @c_id INT
BEGIN
   IF @c_id IN (SELECT courses_id FROM Course)
       BEGIN
           BEGIN TRY
              INSERT INTO Question (question, q_type, q_grade, q_answer, courses_id)
              VALUES (@question,@q_type,@grade,@answer,@c_id)
           END TRY
           BEGIN CATCH
              PRINT 'Please check your data'
           END CATCH
       END
   ELSE
          PRINT 'course you tried to enter is not exist'
       END
END
GO
```

 $\label{local} \mbox{Project > (local)$\backslash $>$ User databases > examination_system > Programmability > Stored Procedures > dbo.insert_question$

Uses

[dbo].[Course] [dbo].[Question]

[dbo].[instr_id]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@id	int	4

SQL Script

```
CREATE PROC [dbo].[instr id] @id INT
BEGIN
IF @id IN (SELECT i.insrt id FROM Instructor i)
     SELECT COUNT(sc.st id) AS Student Number, i.fname+' '+i.lname AS
      FullName , c.course_name
      FROM Instructor i INNER JOIN Instructor courses ic
       ON i.insrt id = ic.instr id INNER JOIN Course c
       ON ic.course id = c.courses id INNER JOIN Student course sc
       ON c.courses_id = sc.course_id
       WHERE i.insrt id = @id
      GROUP BY i.fname, i.lname , c.course_name
  END
ELSE
  BEGIN
  PRINT 'instructure id is exist'
  END
END
GO
```

Uses

[dbo].[Course]
[dbo].[Instructor]
[dbo].[Instructor_courses]
[dbo].[Student_course]

[dbo].[instructor_course_insert_sp]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@instructor_id	int	4
@course_id	int	4

SQL Script

```
create procedure [dbo].[instructor_course_insert_sp] @instructor_id INT, @course_id int
as
begin

BEGIN TRY
    insert into Instructor_courses (instr_id,course_id)
    values(@instructor_id , @course_id)

END TRY

BEGIN CATCH
    print 'Instructor_Id or course_Id you try to enter is not exist'
END CATCH
end
GO
```

Uses

[dbo].[Instructor_courses]

[dbo].[instructor_course_select_sp]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE PROC [dbo].[instructor_course_select_sp]

AS

BEGIN

SELECT *FROM Instructor_courses

END

GO
```

Uses

[dbo].[Instructor_courses]

[dbo].[instructor_course_update_sp]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@instructor_id	int	4
@course_id	int	4

SQL Script

Uses

[dbo].[Course]
[dbo].[Instructor]
[dbo].[Instructor_courses]

[dbo].[instructor_courses_delete_by_crsld_sp]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Course_ld	varchar(50)	50

SQL Script

```
CREATE PROCEDURE [dbo].[instructor_courses_delete_by_crsId_sp] @Course_Id varchar(50)

AS

BEGIN

BEGIN TRY

DELETE FROM Instructor_courses WHERE @Course_Id=course_Id

END TRY

BEGIN CATCH

PRINT 'You cannot delete this instructor course'

END CATCH

END

GO
```

Uses

[dbo].[Instructor_courses]

[dbo].[instructor_courses_delete_by_insld_sp]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Instructor_ld	varchar(50)	50

SQL Script

```
CREATE PROCEDURE [dbo].[instructor_courses_delete_by_insId_sp] @Instructor_Id varchar(50)

AS

BEGIN

BEGIN TRY

DELETE FROM Instructor_courses WHERE instr_id = @Instructor_Id

END TRY

BEGIN CATCH

PRINT 'You cannot delete this instructor course'

END CATCH

END

GO
```

Uses

[dbo].[Instructor_courses]

[dbo].[instructor_delete_sp]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Ins_ID	varchar(50)	50

SQL Script

```
CREATE PROCEDURE [dbo].[instructor_delete_sp] @Ins_ID varchar(50)

AS

BEGIN

BEGIN TRY

DELETE FROM Instructor WHERE insrt_id = @Ins_ID

END TRY

BEGIN CATCH

PRINT 'You cannot delete this instructor'

END CATCH

END

GO
```

Uses

[dbo].[Instructor]

```
[dbo].[instructor_insert_sp]
```

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@ins_fname	nvarchar(50)	100
@ins_Iname	nvarchar(50)	100
@dept_id	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[instructor_insert_sp] @ins_fname NVARCHAR(50),@ins_lname NVARCHAR(50),@dept_id INT

AS

BEGIN

BEGIN TRY

INSERT INTO Instructor (fname,lname,dept_id)

VALUES(@ins_fname,@ins_lname,@dept_id)

END TRY

BEGIN CATCH

PRINT 'department_id you try to enter is not exist'

END CATCH

END

GO
```

Uses

[dbo].[Instructor]

[dbo].[instructor_select_by_course_id_sp]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@course_id	int	4

SQL Script

```
CREATE PROC [dbo].[instructor_select_by_course_id_sp] @course_id int

AS

BEGIN

SELECT *FROM Instructor_courses

WHERE course_id = @course_id

END

GO
```

Uses

[dbo].[Instructor_courses]

[dbo].[instructor_select_by_id_sp]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@id	int	4

SQL Script

```
CREATE PROC [dbo].[instructor_select_by_id_sp] @id int

AS

BEGIN

SELECT *FROM Instructor

WHERE insrt_id = @id

END

GO
```

Uses

[dbo].[Instructor]

[dbo].[instructor_select_by_instructor_id_sp]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@instructor_id	int	4

SQL Script

```
CREATE PROC [dbo].[instructor_select_by_instructor_id_sp] @instructor_id int

AS

BEGIN

SELECT *FROM Instructor_courses

WHERE instr_id = @instructor_id

END

GO
```

Uses

[dbo].[Instructor_courses]

[dbo].[instructor_select_sp]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE PROC [dbo].[instructor_select_sp]

AS

BEGIN

SELECT *FROM Instructor

END

GO
```

Uses

[dbo].[Instructor]

[dbo].[instructor_update_sp]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@ins_id	int	4
@ins_fname	nvarchar(50)	100
@ins_Iname	nvarchar(50)	100
@dept_id	int	4

SQL Script

```
create proc [dbo].[instructor_update_sp] @ins_id int, @ins_fname nvarchar(50),
@ins_lname nvarchar(50), @dept_id int
   begin
        if @dept_id in (select dept_id from Department)
           begin
                   if @ins_id not in (select insrt_id from Instructor)
                   insert into Instructor(fname,lname,dept id)
                   values (@ins_fname, @ins_lname, @dept_id)
                   update Instructor
                    set fname=@ins_fname, lname=@ins_lname, dept_id=@dept_id
                    where insrt id=@ins id
        end
       else
           print 'Department does not exist'
   END
GO
```

Uses

[dbo].[Department] [dbo].[Instructor]

[dbo].[qChoice_select_by_id]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@q_id	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[qChoice_select_by_id] @q_id INT

AS

BEGIN

SELECT * FROM Question_choices WHERE q_id = @q_id

END

GO
```

Uses

[dbo].[Question_choices]

[dbo].[question_delete_by_id]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@q_id	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[question_delete_by_id] @q_id INT

AS

BEGIN

BEGIN TRY

DELETE FROM Question WHERE q_id=@q_id

END TRY

BEGIN CATCH

PRINT 'You cannot delete this Question'

END CATCH

END GO
```

Uses

[dbo].[question_select]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE PROCEDURE [dbo].[question_select]
AS
BEGIN
SELECT * FROM Question
END
GO
```

Uses

[dbo].[question_select_by_id]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@q_id	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[question_select_by_id] @q_id INT

AS

BEGIN

SELECT * FROM Question WHERE q_id=@q_id

END

GO
```

Uses

[dbo].[quetion_update]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@q_id	int	4
@question	varchar(50)	50
@q_type	varchar(10)	10
@grade	int	4
@answer	varchar(3)	3
@c_id	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[quetion_update] @q_id INT, @question VARCHAR(50), @q_type VARCHAR(10), @grade INT, @answer VARCHAR(3), @c_id INT

AS

BEGIN

IF @q_id IN (SELECT q_id FROM Question) AND @c_id IN (SELECT courses_id FROM Course)

BEGIN

UPDATE Question

SET question = @question, q_type = @q_type, q_grade = @grade, q_answer = @answer

WHERE q_id = @q_id AND courses_id = @c_id

END

ELSE

BEGIN

PRINT('Selected id not exist, check question id and course id')

END

END

GO
```

Uses

[dbo].[Course]

Project > (local)\ > User databases > examination_system > Programmability > Stored Procedures > dbo.quetion_update

[dbo].[report_st_crs]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@id	int	4

SQL Script

```
CREATE PROC [dbo].[report_st_crs] @id INT

AS

BEGIN

IF @id IN (SELECT s.st_id FROM Student s)

BEGIN

SELECT s.fname+' '+s.lname AS FullName ,sc.grade,c.course_name

FROM Student s INNER JOIN Student_course sc

ON s.st_id = sc.st_id INNER JOIN Course c ON sc.course_id = c.courses_id

WHERE s.st_id = @id

END

ELSE

BEGIN

PRINT 'Student Id is Exist'

END

END

END

GO
```

Uses

[dbo].[Course]
[dbo].[Student]
[dbo].[Student_course]

```
[dbo].[st_crs_delete_by_cr_id]
```

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@crs_id	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[st_crs_delete_by_cr_id] @crs_id INT

AS

BEGIN

BEGIN TRY

DELETE FROM Student_course WHERE course_id = @crs_id

END TRY

BEGIN CATCH

PRINT 'You cannot delete this student course'

END CATCH

END

GO
```

Uses

[dbo].[Student_course]

```
[dbo].[st_crs_delete_by_id]
```

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@st_id	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[st_crs_delete_by_id] @st_id INT

AS

BEGIN

BEGIN TRY

DELETE FROM Student_course WHERE st_id =@st_id

END TRY

BEGIN CATCH

PRINT 'You cannot delete this student course'

END CATCH

END

GO
```

Uses

[dbo].[Student_course]

[dbo].[st_ex_qs_insert]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@st_id	int	4
@ex_id	int	4
@qs_id	int	4
@grade	int	4
@answer	varchar(10)	10
@is_corrected	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[st ex qs insert] @st id INT, @ex id INT, @qs id INT, @grade
INT, @answer VARCHAR(10), @is corrected INT
AS
BEGIN
   IF
        (@st id IN (SELECT st id FROM Student))
        (@ex id IN (SELECT exam id FROM Exam ))
        (@qs_id IN (SELECT q_id FROM Question ))
           IF @st id IN (SELECT st id FROM Student exams questions
           WHERE st id =@st id AND exam id = @ex id AND qs id = @qs id )
           BEGIN
              PRINT 'dublicatec student id , exam id and qs id'
           END
        ELSE
        BEGIN
           BEGIN TRY
               INSERT INTO student_exams_-
Questions(st_id,exam_id,qs_id,grade,answer,is_corrected)
               VALUES (@st_id,@ex_id,@qs_id,@grade,@answer,@is_corrected)
           END TRY
```

```
BEGIN CATCH

PRINT 'Please check your data'

END CATCH

END

END

ELSE

BEGIN

PRINT 'this student_Exam_question not exist'

END

END

GO
```

Uses

[dbo].[Exam]
[dbo].[Question]
[dbo].[Student]
[dbo].[Student_exams_questions]

[dbo].[st_ex_qs_select_all]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE PROCEDURE [dbo].[st_ex_qs_select_all]

AS

BEGIN

SELECT *

FROM student_exams_Questions

END

GO
```

Uses

[dbo].[Student_exams_questions]

[dbo].[st_ex_qs_select_by_ids]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@st_id	int	4
@exam_id	int	4
@qs_id	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[st_ex_qs_select_by_ids] @st_id INT, @exam_id INT, @qs_id INT

AS

BEGIN

SELECT *

FROM student_exams_Questions

WHERE st_id = @st_id AND exam_id = @exam_id AND qs_id = @qs_id

END

GO
```

Uses

[dbo].[Student_exams_questions]

[dbo].[st_ex_qs_update]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@st_id	int	4
@ex_id	int	4
@qs_id	int	4
@grade	int	4
@answer	varchar(10)	10
@is_corrected	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[st ex qs update] @st id INT, @ex id INT, @qs id INT, @grade
INT, @answer VARCHAR(10), @is corrected INT
AS
BEGIN
   IF
        (@st id IN (SELECT st id FROM Student))
        (@ex id IN (SELECT exam id FROM Exam ))
        (@qs_id IN (SELECT q_id FROM Question ))
           IF @st id IN (SELECT st id FROM Student exams questions
           WHERE st id =@st id AND exam id = @ex id AND qs id = @qs id )
               BEGIN
                   UPDATE student exams Questions
                   SET grade =@grade , answer = @answer , is_corrected = @is_corrected
                   WHERE st id =@st id AND exam id = @ex id AND qs id = @qs id
               END
           ELSE
               PRINT 'dublicated student_id , exam_id and qs_id'
           END
        END
```

Project > (local)\ > User databases > examination_system > Programmability > Stored Procedures > dbo.st_ex_qs_update

```
ELSE

BEGIN

PRINT 'this student_Exam_question not exist'

END

END

GO
```

Uses

[dbo].[Exam] [dbo].[Question]

[dbo].[Student]

[dbo].[Student_exams_questions]

```
[dbo].[std_cs_select_by_id]
```

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@id	int	4

SQL Script

```
CREATE PROC [dbo].[std_cs_select_by_id] @id int

AS

BEGIN

SELECT* FROM

Student_course sc

WHERE sc.st_id = @id

END

GO
```

Uses

[dbo].[Student_course]

[dbo].[std_cs_select_sp]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE PROC [dbo].[std_cs_select_sp]

AS

BEGIN

SELECT* FROM

Student_course sc

END

GO
```

Uses

[dbo].[Student_course]

[dbo].[student_answers]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@ex_id	int	4
@q_id	int	4
@s_id	int	4
@answer	varchar(10)	10

SQL Script

```
CREATE PROCEDURE [dbo].[student answers] @ex id INT, @q id INT, @s id INT, @answer
VARCHAR (10)
AS
BEGIN
   IF @s id IN (SELECT st id FROM Student)
       AND @ex_id IN (SELECT exam_id FROM Exam)
       AND @q id IN (SELECT q id FROM Question q)
       BEGIN
           INSERT INTO Student_exams_questions (st_id,exam_id,qs_id,answer)
           VALUES (@s id, @ex id, @q id, @answer)
       END
   ELSE
       BEGIN
           PRINT('ERROR, please check the inserted ids')
       END
END
GO
```

Uses

[dbo].[Exam] [dbo].[Question]

[dbo].[Student]

 $\label{local} \mbox{Project > (local)$\ > User databases > examination_system > Programmability > Stored Procedures > dbo.student_answers$

[dbo].[Student_exams_questions]

[dbo].[student_Course_insert_sp]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@course_id	int	4
@st_id	int	4
@grade	int	4

SQL Script

```
CREATE PROC [dbo].[student_Course_insert_sp] @course_id INT , @st_id INT , @grade INT

AS

BEGIN

BEGIN TRY

INSERT INTO Student_course

VALUES(@course_id,@st_id,@grade)

END TRY

BEGIN CATCH

PRINT 'course or student you try to enter is not exist'

END CATCH

END

GO
```

Uses

[dbo].[Student_course]

[dbo].[student_course_update_sp]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@std_id	int	4
@crs_id	int	4
@grade	int	4

SQL Script

```
CREATE PROC [dbo].[student course update sp] @std id INT, @crs id INT, @grade INT
BEGIN
         if @std id in (select st id from Student)
             begin
                    if @crs id in (select courses id from Course)
                       begin
if @std_id not in (select st_id from student_course where
course_id = @crs_id) and @crs_id not in (select course_id from student_course where
st id = @std id)
                                 begin
                                 insert into student course
                                 values (@std id,@crs id,@grade)
                                 end
                            else
                           update student_course
                           set grade = @grade
                           where st_id=@std_id and course_id= @crs_id
                        end
                 else
                 print 'Course does not exist'
        else
        print 'Student does not exist'
END
GO
```

Project > (local)\ > User databases > examination_system > Programmability > Stored Procedures > dbo.student_course_update_sp

Uses

[dbo].[Course] [dbo].[Student] [dbo].[Student_course] [dbo].[student_delete_sp]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@st_id	int	4

SQL Script

```
CREATE PROC [dbo].[student_delete_sp] @st_id INT

AS

BEGIN

BEGIN TRY

DELETE FROM Student WHERE st_id = @st_id

END TRY

BEGIN CATCH

PRINT 'You cannot delete this student'

END CATCH

END CATCH

GO
```

Uses

[dbo].[Student]

[dbo].[student_insert_sp]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@fname	varchar(50)	50
@Iname	varchar(50)	50
@st_gender	varchar(3)	3
@st_phone	varchar(11)	11
@st_birthdate	date	3
@st_joindate	date	3
@dept_id	int	4
@city	varchar(50)	50
@street	varchar(50)	50

SQL Script

```
CREATE PROC [dbo].[student_insert_sp] @fname VARCHAR(50) , @lname VARCHAR(50)
,@st_gender VARCHAR(3), @st_phone VARCHAR(11),
@st birthdate DATE , @st joindate DATE ,@dept id INT, @city VARCHAR(50),@street
VARCHAR (50)
AS
BEGIN
   IF @dept id IN (SELECT dept id FROM Department d)
       BEGIN
           BEGIN TRY
              INSERT INTO Student
(@fname,@lname,@st_gender,@st_phone,@st_birthdate,@st_joindate,@dept_id,@city,@street)
           END TRY
           BEGIN CATCH
               PRINT 'Please check your data'
           END CATCH
       END
   ELSE
       BEGIN
```

Project > (local)\ > User databases > examination_system > Programmability > Stored Procedures > dbo.student_insert_sp

```
PRINT 'department you try to enter is not exist'
END
END
GO
```

Uses

[dbo].[Department] [dbo].[Student]

[dbo].[student_select_by_id_sp]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@st_id	int	4

SQL Script

```
CREATE PROC [dbo].[student_select_by_id_sp] @st_id INT

AS

BEGIN

SELECT * FROM Student s WHERE s.st_id = @st_id

END

GO
```

Uses

[dbo].[Student]

[dbo].[student_select_sp]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE PROC [dbo].[student_select_sp]
AS
BEGIN
SELECT * FROM Student s
END
GO
```

Uses

[dbo].[Student]

[dbo].[student_update_sp]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@st_id	int	4
@fname	varchar(50)	50
@Iname	varchar(50)	50
@st_gender	varchar(3)	3
@st_phone	varchar(11)	11
@st_birthdate	date	3
@st_joindate	date	3
@dept_id	int	4
@city	varchar(50)	50
@street	varchar(50)	50

SQL Script

```
CREATE PROC [dbo].[student_update_sp] @st_id INT, @fname VARCHAR(50) , @lname
VARCHAR(50) , @st_gender VARCHAR(3), @st_phone VARCHAR(11),
@st birthdate DATE , @st joindate DATE ,@dept id INT, @city VARCHAR(50),@street
VARCHAR (50)
AS
BEGIN
   IF @dept id IN (SELECT d.dept id FROM Department d)
       BEGIN
            IF @st id NOT IN (SELECT s.st id FROM Student s)
               BEGIN
                   INSERT INTO Student
                   VALUES
(@fname,@lname,@st_gender,@st_phone,@st_birthdate,@st_joindate,@dept_id,@city,@street)
            ELSE
               BEGIN
                   UPDATE Student
                   SET fname = @fname, lname=@lname, st gender=@st gender,
st_phone=@st_phone,st_birthdate=@st_birthdate,
```

Project > (local)\ > User databases > examination_system > Programmability > Stored Procedures > dbo.student_update_sp

```
st_joindate= @st_joindate, dept_id = @dept_id, city=@city,

street=@street

WHERE st_id = @st_id

END

END

ELSE

BEGIN

PRINT 'Department id that you enter is not exist'

END

END

END

END

GO
```

Uses

[dbo].[Department] [dbo].[Student]

[dbo].[topic_insert]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@topic_name	varchar(50)	50

SQL Script

```
CREATE PROCEDURE [dbo].[topic_insert] @topic_name VARCHAR(50)

AS

BEGIN

INSERT INTO topic (topic_name)

VALUES(@topic_name)

END

GO
```

Uses

[dbo].[topic_select_all]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE PROCEDURE [dbo].[topic_select_all]

AS

BEGIN

SELECT *

FROM Topic

END

GO
```

Uses

[dbo].[topic_select_by_id]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@topic_id	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[topic_select_by_id] @topic_id INT

AS

BEGIN

SELECT *

FROM Topic

WHERE topic_id = @topic_id

END

GO
```

Uses

[dbo].[topic_update]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@topic_id	int	4
@topic_name	varchar(50)	50

SQL Script

```
CREATE PROCEDURE [dbo].[topic_update] @topic_id INT , @topic_name VARCHAR(50)

AS

BEGIN

IF @topic_id IN (SELECT topic_id FROM Topic )

BEGIN

UPDATE Topic

SET topic_name = @topic_name

WHERE topic_id = @topic_id

END

ELSE

BEGIN

PRINT 'this topic is not exist'

END

GO
```

Uses