Examination System SQL Documentation

ExaminationSystem

Server	sql-server-iti.database.windows.net
Author	Team 1: Ammar Elghandour, Ahmed Abdelnaser, Aya Kamel, Walid Essam, Sohaila Elhakim
Created	Monday, February 27, 2023 11:28:15 PM
File Path	C:\Users\laptop\Documents\My Database Documentation\ExaminationSystem_documentation-2023-02-27T23-28-15.pdf

SQL Database for automated system that can perform online exams

Table of Contents

sql-server-iti.database.windows.net

Databases (1)

- ExaminationSystem

Server Properties

Property	Value
Edition	SQL Azure

User databases

Databases (1)

- ExaminationSystem

ExaminationSystem Database

MS_Description

Data Base for Examination System

Database Properties

Property	Value
SQL Server Version	Azure Sql Database

Tables

Objects

Name
dbo.courses
dbo.courses_students
dbo.Department
dbo.exam_answers
dbo.exams
dbo.exams_questions
dbo.Instructors
dbo.questions
dbo.students
dbo.topic

[dbo].[courses]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	6
Created	3:01:53 PM Friday, February 17, 2023
Last Modified	1:32:03 PM Monday, February 20, 2023

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK C	id	int	4	NOT NULL	1 - 1
	course_name	nvarchar(200)	400	NOT NULL	
FK	instructor_id	int	4	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
PK C	course_id_pk	id	True

Foreign Keys

Name	Columns
fk_courses_instructor_id	instructor_id->[dbo].[Instructors].[Ins_Id]

SQL Script

```
CREATE TABLE [dbo].[courses]
(
[id] [int] NOT NULL IDENTITY(1, 1),
[course_name] [nvarchar] (200) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
[instructor_id] [int] NULL
)
GO
ALTER TABLE [dbo].[courses] ADD CONSTRAINT [course_id_pk] PRIMARY KEY CLUSTERED
([id])
GO
ALTER TABLE [dbo].[courses] ADD CONSTRAINT [fk_courses_instructor_id] FOREIGN KEY
([instructor_id]) REFERENCES [dbo].[Instructors] ([Ins_Id])
GO
```

Uses

[dbo].[Instructors]

Used By

[dbo].[courses_students]
[dbo].[exams]
[dbo].[questions]
[dbo].[topic]
[dbo].[deleteCourseById]
[dbo].[gradesInAllCoursesById]
[dbo].[insertCourses]
[dbo].[PROC_addNewQuestion]
[dbo].[PROC_examCorrect]
[dbo].[PROC_generate_exam]
[dbo].[PROC_getInstructorCourses]
[dbo].[PROC_getQuestionById]
[dbo].[PROC_getStudentAnswerWithModel]
[dbo].[PROC_updateAQuestion]
[dbo].[ReportForIns]
[dbo].[selectCourseById]
[dbo].[selectCourses]
[dbo].[selectTopicById]
[dbo].[updateCourses]
[dbo].[updateTopic]

[dbo].[courses_students]

Properties

Property	Value
Row Count (~)	10
Created	1:32:03 PM Monday, February 20, 2023
Last Modified	8:55:18 PM Monday, February 20, 2023

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK C	id	int	4	NOT NULL	1 - 1
FK	course_id	int	4	NULL allowed	
FK	student_id	int	4	NULL allowed	
	grade	int	4	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
PK C	id_course_student_pk	id	True

Foreign Keys

Name	Update	Delete	Columns
ci_student_id_fk	Cascade	SetNull	student_id->[dbo].[students].[Std_id]
cs_course_id_fk	Cascade	SetNull	course_id->[dbo].[courses].[id]

SQL Script

```
CREATE TABLE [dbo].[courses_students]
(
[id] [int] NOT NULL IDENTITY(1, 1),
[course_id] [int] NULL,
[student_id] [int] NULL,
[grade] [int] NULL
)
GO
ALTER TABLE [dbo].[courses_students] ADD CONSTRAINT [id_course_student_pk] PRIMARY
KEY CLUSTERED ([id])
GO
ALTER TABLE [dbo].[courses_students] ADD CONSTRAINT [ci_student_id_fk] FOREIGN KEY
([student_id]) REFERENCES [dbo].[students] ([Std_id]) ON DELETE SET NULL ON UPDATE
```

```
CASCADE
GO
ALTER TABLE [dbo].[courses_students] ADD CONSTRAINT [cs_course_id_fk] FOREIGN KEY
([course_id]) REFERENCES [dbo].[courses] ([id]) ON DELETE SET NULL ON UPDATE CASCADE
GO
```

Uses

[dbo].[courses]
[dbo].[students]

Used By

[dbo].[gradesInAllCoursesById]
[dbo].[PROC_examCorrect]
[dbo].[PROC_getInstructorCourses]
[dbo].[ReportForIns]

[dbo].[Department]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	1
Created	9:10:14 AM Sunday, February 19, 2023
Last Modified	9:10:14 AM Sunday, February 19, 2023

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Default
PK C	Dept_Id	int	4	NOT NULL	
	Dept_Name	varchar(20)	20	NOT NULL	
	Dept_Phone	numeric(18,0)	9	NULL allowed	
	Dept_Location	varchar(50)	50	NULL allowed	
	Dept_Manager	int	4	NOT NULL	
	Manager_HireDate	date	3	NULL allowed	(getdate())

Indexes

Key	Name	Key Columns	Unique
PK C	PK__Departme__72ABC2CC0E40E427	Dept_Id	True

SQL Script

```
CREATE TABLE [dbo].[Department]
(
[Dept_Id] [int] NOT NULL,
[Dept_Name] [varchar] (20) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
[Dept_Phone] [numeric] (18, 0) NULL,
[Dept_Location] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[Dept_Manager] [int] NOT NULL,
[Manager_HireDate] [date] NULL CONSTRAINT [DF__Departmen__Manag__3BFFE745] DEFAULT
(getdate())
)
GO
ALTER TABLE [dbo].[Department] ADD CONSTRAINT [PK__Departme__72ABC2CC0E40E427]
PRIMARY KEY CLUSTERED ([Dept_Id])
GO
```

[dbo].[exam_answers]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	1
Created	6:02:26 AM Sunday, February 19, 2023
Last Modified	6:02:26 AM Sunday, February 19, 2023

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK C	id	int	4	NOT NULL	1 - 1
FK	exam_id	int	4	NOT NULL	
FK	question_id	int	4	NOT NULL	
FK	student_id	int	4	NOT NULL	
	answer	nchar(1)	2	NOT NULL	
	std_score	int	4	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
PK C	exam_answers_id_pk	id	True

Foreign Keys

Name	Update	Delete	Columns
exam_id_exam_answers_fk			exam_id->[dbo].[exams].[id]
ques_id_exam_answers_fk	Cascade	Cascade	question_id->[dbo].[questions].[id]
student_id_exam_answers_fk	Cascade	Cascade	student_id->[dbo].[students].[Std_id]

SQL Script

```
CREATE TABLE [dbo].[exam_answers]
(
[id] [int] NOT NULL IDENTITY(1, 1),
[exam_id] [int] NOT NULL,
[question_id] [int] NOT NULL,
[student_id] [int] NOT NULL,
[answer] [nchar] (1) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
```

```
[std_score] [int] NULL
)
GO
ALTER TABLE [dbo].[exam_answers] ADD CONSTRAINT [exam_answers_id_pk] PRIMARY KEY
CLUSTERED ([id])
GO
ALTER TABLE [dbo].[exam_answers] ADD CONSTRAINT [exam_id_exam_answers_fk] FOREIGN
KEY ([exam_id]) REFERENCES [dbo].[exams] ([id])
GO
ALTER TABLE [dbo].[exam_answers] ADD CONSTRAINT [ques_id_exam_answers_fk] FOREIGN
KEY ([question_id]) REFERENCES [dbo].[questions] ([id]) ON DELETE CASCADE ON UPDATE
CASCADE
GO
ALTER TABLE [dbo].[exam_answers] ADD CONSTRAINT [student_id_exam_answers_fk] FOREIGN
KEY ([student_id]) REFERENCES [dbo].[students] ([Std_id]) ON DELETE CASCADE ON
UPDATE CASCADE
GO
```

Uses

[dbo].[exams]
[dbo].[questions]
[dbo].[students]

Used By

[dbo].[PRO_examAnswers]
[dbo].[PROC_examCorrect]
[dbo].[PROC_getStudentAnswerWithModel]
[dbo].[reportForStudentAns]
[dbo].[reportForStudentAnss]
[dbo].[TestlastReport]

[dbo].[exams]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	8
Created	5:53:49 AM Sunday, February 19, 2023
Last Modified	6:02:26 AM Sunday, February 19, 2023

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK C	id	int	4	NOT NULL	1 - 1
	exam_name	nvarchar(100)	200	NOT NULL	
	duration	int	4	NOT NULL	
	exam_date	date	3	NOT NULL	
	full_mark	int	4	NOT NULL	
FK	course_id	int	4	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
PK C	exam_id_pk	id	True

Foreign Keys

Name	Update	Delete	Columns
course_id_exam_fk	Cascade	Cascade	course_id->[dbo].[courses].[id]

SQL Script

```
CREATE TABLE [dbo].[exams]
(
[id] [int] NOT NULL IDENTITY(1, 1),
[exam_name] [nvarchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
[duration] [int] NOT NULL,
[exam_date] [date] NOT NULL,
[full_mark] [int] NOT NULL,
[course_id] [int] NULL
)
GO
```

```
ALTER TABLE [dbo].[exams] ADD CONSTRAINT [exam_id_pk] PRIMARY KEY CLUSTERED ([id])
GO
ALTER TABLE [dbo].[exams] ADD CONSTRAINT [course_id_exam_fk] FOREIGN KEY
([course_id]) REFERENCES [dbo].[courses] ([id]) ON DELETE CASCADE ON UPDATE CASCADE
GO
```

Uses

[dbo].[courses]

Used By

[dbo].[exam_answers]
[dbo].[exams_questions]
[dbo].[PRO_examAnswers]
[dbo].[PROC_examCorrect]
[dbo].[PROC_generate_exam]
[dbo].[PROC_getStudentAnswerWithModel]
[dbo].[reportForExamQues]
[dbo].[reportForStudentAns]
[dbo].[reportForStudentAnss]

[dbo].[exams_questions]

Properties

Property	Value
Row Count (~)	80
Created	5:53:49 AM Sunday, February 19, 2023
Last Modified	5:53:49 AM Sunday, February 19, 2023

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK C	id	int	4	NOT NULL	1 - 1
FK	exam_id	int	4	NOT NULL	
FK	question_id	int	4	NOT NULL	
	score	int	4	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
PK C	exam_quest_pk	id	True

Foreign Keys

Name	Update	Delete	Columns
exam_id_ques_fk			exam_id->[dbo].[exams].[id]
exam_id_ques_id_fk	Cascade	Cascade	question_id->[dbo].[questions].[id]

SQL Script

```
CREATE TABLE [dbo].[exams_questions]
(
[id] [int] NOT NULL IDENTITY(1, 1),
[exam_id] [int] NOT NULL,
[question_id] [int] NOT NULL,
[score] [int] NULL
)
GO
ALTER TABLE [dbo].[exams_questions] ADD CONSTRAINT [exam_quest_pk] PRIMARY KEY
CLUSTERED ([id])
GO
ALTER TABLE [dbo].[exams_questions] ADD CONSTRAINT [exam_id_ques_fk] FOREIGN KEY
([exam_id]) REFERENCES [dbo].[exams] ([id])
```

```
GO
ALTER TABLE [dbo].[exams_questions] ADD CONSTRAINT [exam_id_ques_id_fk] FOREIGN KEY
([question_id]) REFERENCES [dbo].[questions] ([id]) ON DELETE CASCADE ON UPDATE
CASCADE
GO
```

Uses

[dbo].[exams]
[dbo].[questions]

Used By

[dbo].[PROC_generate_exam]
[dbo].[reportForExamQues]

[dbo].[Instructors]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	6
Created	3:49:49 PM Friday, February 17, 2023
Last Modified	1:28:55 PM Monday, February 20, 2023

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PK C	Ins_Id	int	4	NOT NULL
	Fname	nchar(10)	20	NULL allowed
	Lname	nchar(10)	20	NULL allowed
	age	int	4	NULL allowed
	email	varchar(25)	25	NULL allowed
	street	nchar(10)	20	NULL allowed
	City	nchar(20)	40	NULL allowed
	Zip_Code	char(10)	10	NULL allowed
	password	char(20)	20	NULL allowed
	salary	numeric(18,0)	9	NULL allowed
	Dept_Id	int	4	NOT NULL

Indexes

Key	Name	Key Columns	Unique
PK C	PK_Instructors	Ins_Id	True

SQL Script

```
CREATE TABLE [dbo].[Instructors]
(
[Ins_Id] [int] NOT NULL,
[Fname] [nchar] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[Lname] [nchar] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[age] [int] NULL,
[email] [varchar] (25) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[street] [nchar] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[City] [nchar] (20) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[Zip_Code] [char] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
```

```
[password] [char] (20) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[salary] [numeric] (18, 0) NULL,
[Dept_Id] [int] NOT NULL
)
GO
ALTER TABLE [dbo].[Instructors] ADD CONSTRAINT [PK_Instructors] PRIMARY KEY
CLUSTERED ([Ins_Id])
GO
```

Used By

[dbo].[courses]
[dbo].[ADDINS]
[dbo].[DeleteIns]
[dbo].[GETInsByID]
[dbo].[PROC_getInstructorCourses]
[dbo].[PROC_IUD]
[dbo].[PROC_IUDD]
[dbo].[ReportForIns]
[dbo].[SelectAllIns]
[dbo].[selectAllInstructors]
[dbo].[UpdateInstructor]

[dbo].[questions]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	20
Created	11:32:04 PM Saturday, February 18, 2023
Last Modified	6:02:26 AM Sunday, February 19, 2023

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK C	id	int	4	NOT NULL	1 - 1
	content	nvarchar(200)	400	NOT NULL	
	correct_ans	char(1)	1	NOT NULL	
	ques_type	char(3)	3	NOT NULL	
FK	course_id	int	4	NULL allowed	
	A	nvarchar(100)	200	NULL allowed	
	B	nvarchar(100)	200	NULL allowed	
	C	nvarchar(100)	200	NULL allowed	
	D	nvarchar(100)	200	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
PK C	ques_id_pk	id	True

Check Constraints

Name	On Column	Constraint
CK__questions__corre__65370702	correct_ans	([correct_ans]='d' OR [correct_ans]='c' OR [correct_ans]='b' OR [correct_ans]='a')
CK__questions__ques___662B2B3B	ques_type	([ques_type]='MCQ' OR [ques_type]='T/F')

Foreign Keys

Name	Update	Delete	Columns
course_id_fk	Cascade	Cascade	course_id->[dbo].[courses].[id]

SQL Script

```
CREATE TABLE [dbo].[questions]
(
[id] [int] NOT NULL IDENTITY(1, 1),
[content] [nvarchar] (200) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
[correct_ans] [char] (1) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
[ques_type] [char] (3) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
[course_id] [int] NULL,
[A] [nvarchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[B] [nvarchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[C] [nvarchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[D] [nvarchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
)
GO
ALTER TABLE [dbo].[questions] ADD CONSTRAINT [CK__questions__corre__65370702] CHECK
(([correct_ans]='d' OR [correct_ans]='c' OR [correct_ans]='b' OR [correct_ans]='a'))
GO
ALTER TABLE [dbo].[questions] ADD CONSTRAINT [CK__questions__ques___662B2B3B] CHECK
(([ques_type]='MCQ' OR [ques_type]='T/F'))
GO
ALTER TABLE [dbo].[questions] ADD CONSTRAINT [ques_id_pk] PRIMARY KEY CLUSTERED
([id])
GO
ALTER TABLE [dbo].[questions] ADD CONSTRAINT [course_id_fk] FOREIGN KEY
([course_id]) REFERENCES [dbo].[courses] ([id]) ON DELETE CASCADE ON UPDATE CASCADE
GO
```

Uses

[dbo].[courses]

Used By

[dbo].[exam_answers]
[dbo].[exams_questions]
[dbo].[PRO_examAnswers]
[dbo].[PROC_addNewQuestion]
[dbo].[PROC_deleteAQuestion]
[dbo].[PROC_examCorrect]
[dbo].[PROC_generate_exam]
[dbo].[PROC_getQuestionById]
[dbo].[PROC_getStudentAnswerWithModel]
[dbo].[PROC_updateAQuestion]
[dbo].[reportForExamQues]
[dbo].[reportForStudentAns]
[dbo].[reportForStudentAnss]

[dbo].[students]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	4
Created	9:47:12 PM Saturday, February 18, 2023
Last Modified	1:32:03 PM Monday, February 20, 2023

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK C	Std_id	int	4	NOT NULL	1 - 1
	Fname	varchar(50)	50	NOT NULL	
	Lname	varchar(50)	50	NULL allowed	
	age	int	4	NULL allowed	
	email	varchar(20)	20	NULL allowed	
	Street	varchar(10)	10	NULL allowed	
	city	varchar(10)	10	NULL allowed	
	Zip_code	varchar(10)	10	NULL allowed	
	phone	varchar(20)	20	NULL allowed	
	password	varchar(20)	20	NULL allowed	
	Dept_id	int	4	NULL allowed	
	supervise_id	int	4	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
PK C	PK_students	Std_id	True

SQL Script

```
CREATE TABLE [dbo].[students]
(
[Std_id] [int] NOT NULL IDENTITY(1, 1),
[Fname] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
[Lname] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[age] [int] NULL,
[email] [varchar] (20) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[Street] [varchar] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[city] [varchar] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
```

```
[Zip_code] [varchar] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[phone] [varchar] (20) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[password] [varchar] (20) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[Dept_id] [int] NULL,
[supervise_id] [int] NULL
)
GO
ALTER TABLE [dbo].[students] ADD CONSTRAINT [PK_students] PRIMARY KEY CLUSTERED
([Std_id])
GO
```

Used By

[dbo].[courses_students]
[dbo].[exam_answers]
[dbo].[deleteStudent]
[dbo].[getAllStudents]
[dbo].[gradesInAllCoursesById]
[dbo].[insertStudent]
[dbo].[PRO_examAnswers]
[dbo].[PROC_examCorrect]
[dbo].[PROC_getInstructorCourses]
[dbo].[PROC_getStudentAnswerWithModel]
[dbo].[ReportForIns]
[dbo].[reportForStudentAns]
[dbo].[reportForStudentAnss]
[dbo].[selectAllStudents]
[dbo].[selectStudentById]
[dbo].[TestlastReport]
[dbo].[updateStudent]

[dbo].[topic]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	17
Created	11:12:17 AM Sunday, February 19, 2023
Last Modified	11:12:17 AM Sunday, February 19, 2023

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK C	id	int	4	NOT NULL	1 - 1
FK	course_id	int	4	NULL allowed	
	topic_name	nvarchar(200)	400	NOT NULL	

Indexes

Key	Name	Key Columns	Unique
PK C	topic_id_pk	id	True

Foreign Keys

Name	Update	Delete	Columns
topic_course_id	Cascade	Cascade	course_id->[dbo].[courses].[id]

SQL Script

```
CREATE TABLE [dbo].[topic]
(
[id] [int] NOT NULL IDENTITY(1, 1),
[course_id] [int] NULL,
[topic_name] [nvarchar] (200) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL
)
GO
ALTER TABLE [dbo].[topic] ADD CONSTRAINT [topic_id_pk] PRIMARY KEY CLUSTERED ([id])
GO
ALTER TABLE [dbo].[topic] ADD CONSTRAINT [topic_course_id] FOREIGN KEY ([course_id])
REFERENCES [dbo].[courses] ([id]) ON DELETE CASCADE ON UPDATE CASCADE
GO
```

Uses

[dbo].[courses]

Used By

[dbo].[deleteTopic]
[dbo].[insertTopic]
[dbo].[reportForTopic]
[dbo].[selectTopic]
[dbo].[selectTopicById]
[dbo].[updateTopic]

Stored Procedures

Objects

Name
dbo.ADDINS
dbo.deleteColumn
dbo.deleteCourseById
dbo.DeleteIns
dbo.deleteStudent
dbo.deleteTopic
dbo.getAllStudents
dbo.GETInsByID
dbo.gradesInAllCoursesById
dbo.insertCourses
dbo.insertData
dbo.insertStudent
dbo.insertTopic
dbo.PRO_examAnswers
dbo.PROC_addNewQuestion
dbo.PROC_deleteAQuestion
dbo.PROC_examCorrect
dbo.PROC_generate_exam
dbo.PROC_getInstructorCourses
dbo.PROC_getQuestionById
dbo.PROC_getStudentAnswerWithModel
dbo.PROC_IUD
dbo.PROC_IUDD
dbo.PROC_updateAQuestion
dbo.reportForExamQues
dbo.ReportForIns
dbo.reportForStudentAns
dbo.reportForStudentAnss
dbo.reportForTopic
dbo.SelectAllIns
dbo.selectAllInstructors
dbo.selectAllStudents
dbo.selectCourseById
dbo.selectCourses
dbo.selectOne
dbo.selectStudentById

dbo.selectTopic
dbo.selectTopicById
dbo.TestlastReport
dbo.updateCourses
dbo.UpdateInstructor
dbo.updateStudent
dbo.UpdateTable
dbo.updateTopic

[dbo].[ADDINS]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@id	int	4
@first_name	nchar(10)	20
@last_name	nchar(10)	20
@age	int	4
@email	varchar(25)	25
@street	nchar(10)	20
@City	nchar(20)	40
@zip_Code	char(10)	10
@password	char(20)	20
@salary	decimal(10,2)	9
@Dept_Id	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[ADDINS] (@id  INTEGER,
                                        @first_name   NCHAR(10),
                                        @last_name     NCHAR(10),
                                        @age           INT,
                                        @email       VARCHAR(25),
                                        @street      NCHAR(10),
                                        @City        NCHAR(20),
                                        @zip_Code     CHAR(10),
                                        @password     CHAR(20),

                                        @salary         DECIMAL(10, 2),
                                            @Dept_Id           INT )
AS

        BEGIN TRY

            INSERT INTO Instructors
            VALUES      ( @id,
                          @first_name,
```

```
                         @last_name,
                        @age,
                        @email,
                        @street,
                        @City,
                        @zip_Code,
                        @password,
                        @salary,
                        @Dept_Id
                        )
       EnD TRY
       BEGIN CATCH
          PRINT 'the error is '+ERROR_MESSAGE();

        END CATCH
GO
```

Uses

[dbo].[Instructors]

[dbo].[deleteColumn]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@TableName	nvarchar(50)	100
@ColumnName	nvarchar(50)	100
@ColumnValue	nvarchar(50)	100

SQL Script

```
CREATE PROCEDURE [dbo].[deleteColumn]
    @TableName nvarchar(50),
    @ColumnName nvarchar(50),
    @ColumnValue nvarchar(50)
AS
begin try
    DECLARE @sql nvarchar(max)
    SET @sql = 'DELETE FROM ' + @TableName + ' WHERE ' + @ColumnName + ' = @Column-
Value'
    EXEC sp_executesql @sql, N'@ColumnValue nvarchar(50)', @ColumnValue
end try
begin CATCH
    select 'not found'
end catch
GO
```

[dbo].[deleteCourseById]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
create PROCEDURE [dbo].[deleteCourseById]
     @id INT
AS
BEGIN TRY
IF NOT EXISTS (SELECT*FROM courses WHERE id=@id)
       BEGIN
       PRINT 'Course is not exist'
       END
ELSE
       BEGIN
       DELETE courses WHERE id=@id
       END
END TRY
BEGIN CATCH
PRINT 'Faild delete courses, the error'+ERROR_MESSAGE()
END CATCH
GO
```

Uses

[dbo].[courses]

[dbo].[DeleteIns]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE PROCEDURE [dbo].[DeleteIns] (@Id INT)
AS
BEGIN
  DECLARE @Rowcount INT=0
       BEGIN  TRY

       SET @Rowcount = (SELECT COUNT(1) FROM Instructors i  WHERE i.Ins_Id=@Id)
                IF (@Rowcount>0)
                         BEGIN
                                BEGIN  TRAN
                                  DELETE FROM Instructors WHERE Ins_Id=@Id
                                 COMMIT TRAN
                           END
        END TRY


      BEGIN CATCH
       PRINT 'the Error Is'+ERROR_MESSAGE()
      END CATCH
END
GO
```

Uses

[dbo].[Instructors]

[dbo].[deleteStudent]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE PROCEDURE [dbo].[deleteStudent]
  @id INT
AS
BEGIN
BEGIN TRY
IF NOT EXISTS(SELECT * FROM students s WHERE s.Std_id=@id)
       BEGIN
       PRINT 'Student is not exist'
       END
ELSE
    BEGIN
    DELETE students WHERE Std_id=@id
    END
END TRY
BEGIN CATCH
PRINT 'Faild delete student'
END CATCH
END
GO
```

Uses

[dbo].[students]

[dbo].[deleteTopic]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE PROCEDURE [dbo].[deleteTopic]
  @id INT
AS
BEGIN
BEGIN TRY
IF NOT EXISTS(SELECT * FROM topic WHERE id=@id)
       BEGIN
       PRINT 'Topic is not exist'
       END
ELSE
    BEGIN
    DELETE topic WHERE id=@id
    END
END TRY
BEGIN CATCH
PRINT 'Faild delete topic'+ ERROR_MESSAGE()
END CATCH
END
GO
```

Uses

[dbo].[topic]

[dbo].[getAllStudents]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE PROCEDURE [dbo].[getAllStudents]
AS
SELECT * FROM students
GO
```

Uses

[dbo].[students]

[dbo].[GETInsByID]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
 CREATE PROCEDURE [dbo].[GETInsByID] ( @Id INT)
  AS
  BEGIN  TRY
  PRINT 'this is Ins'
  SELECT  * FROM Instructors i WHERE i.Ins_Id=@Id
 END TRY
 BEGIN CATCH
 PRINT 'this Error is ' +ERROR_MESSAGE()
 END CATCH
GO
```

Uses

[dbo].[Instructors]

[dbo].[gradesInAllCoursesById]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE PROCEDURE [dbo].[gradesInAllCoursesById]
  @Std_id INT
AS
BEGIN TRY
SELECT s.Fname+' '+s.Lname AS [FULL Name],c.course_name,cs.grade
FROM  courses_students cs ,courses c ,students s
WHERE cs.student_id=@Std_id AND c.id=cs.course_id AND s.Std_id=@Std_id
END TRY
BEGIN CATCH
print 'select student faild'
END CATCH
GO
```

Uses

[dbo].[courses]
[dbo].[courses_students]
[dbo].[students]

[dbo].[insertCourses]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@course_name	nvarchar(200)	400

SQL Script

```
CREATE PROCEDURE [dbo].[insertCourses] (@course_name NVARCHAR(200))
AS
BEGIN
BEGIN TRY
INSERT INTO courses
VALUES(@course_name)
END TRY
BEGIN CATCH
PRINT 'Insert into courses faild, the error'+ERROR_MESSAGE()
END CATCH
END
GO
```

Uses

[dbo].[courses]

[dbo].[insertData]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@TableName	nvarchar(50)	100
@Fields	nvarchar(max)	max
@Values	nvarchar(max)	max

SQL Script

```
CREATE PROCEDURE [dbo].[insertData]
    @TableName NVARCHAR(50),
    @Fields nvarchar(max),
    @Values nvarchar(max)
AS
BEGIN TRY
    DECLARE @sql NVARCHAR(max)
    SET @sql = 'INSERT INTO ' + @TableName + '(' + @Fields + ') VALUES (' + @Values
+ ')'
    EXEC sp_executesql @sql
END TRY
BEGIN CATCH
    SELECT 'not found'
END CATCH
----
GO
```

[dbo].[insertStudent]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Fname	nvarchar(20)	40
@Lname	nvarchar(20)	40
@age	int	4
@email	nvarchar(20)	40
@Street	nvarchar(20)	40
@city	nvarchar(20)	40
@Zip_code	int	4
@phone	nvarchar(20)	40
@password	nvarchar(20)	40
@Dept_id	int	4
@supervise_id	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[insertStudent]

   @Fname NVARCHAR(20)=NULL,
   @Lname NVARCHAR(20)=NULL,
   @age INT=NULL,
   @email NVARCHAR(20)=NULL,
   @Street NVARCHAR(20)=NULL,
   @city NVARCHAR(20)=NULL,
   @Zip_code INT=NULL,
   @phone NVARCHAR(20)=NULL,
   @password NVARCHAR(20)=NULL,
   @Dept_id INT=NULL,
   @supervise_id INT=NULL
AS
BEGIN
BEGIN TRY
INSERT INTO students
VALUES(@Fname,@Lname,@age,@email,@Street,@city,@Zip_code,@phone,@password,@Dept_id,@
supervise_id)
END TRY
```

```
BEGIN CATCH
PRINT 'Insert into student faild, the error'
END CATCH
END
GO
```

Uses

[dbo].[students]

[dbo].[insertTopic]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@course_id	int	4
@topic_name	nvarchar(200)	400

SQL Script

```
CREATE PROCEDURE [dbo].[insertTopic]
   @course_id INT, @topic_name NVARCHAR(200)
AS
BEGIN TRY
IF NOT EXISTS(SELECT * FROM topic t WHERE t.course_id=@course_id)
      PRINT'This Course id is not exist'
INSERT INTO topic
VALUES(@course_id,@topic_name)
END TRY
BEGIN CATCH
PRINT 'Insert into topic faild'
END CATCH
GO
```

Uses

[dbo].[topic]

[dbo].[PRO_examAnswers]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@examId	int	4
@student_id	int	4
@questionId	int	4
@ans	nchar	1

SQL Script

```
CREATE PROCEDURE  [dbo].[PRO_examAnswers]
    @examId INT,
    @student_id INT,
    @questionId INT,
    @ans NCHAR(1)
AS
    BEGIN TRY
        IF NOT EXISTS (SELECT * FROM exams WHERE id=@examId)
            BEGIN
                SELECT 'Exam do not exist' AS 'errMessage'
            END
        ELSE IF NOT EXISTS (SELECT * FROM questions q WHERE id=@questionId)
        BEGIN
            SELECT 'Question does not exist' AS 'errMessage'
        END
        ELSE IF NOT EXISTS (SELECT * FROM students s WHERE s.Std_id=@student_id)
            BEGIN
                SELECT 'Student does not exist' AS 'errMessage'
            END
        ELSE
            BEGIN
                INSERT INTO exam_answers (exam_id, question_id, student_id, answer)
                VALUES (@examId, @questionId, @student_id, @ans);
            END
    END TRY
    BEGIN CATCH
     SELECT ERROR_MESSAGE() AS errorMessage
    END CATCH
```

```
GO
```

Uses

[dbo].[exam_answers]
[dbo].[exams]
[dbo].[questions]
[dbo].[students]

[dbo].[PROC_addNewQuestion]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@courseId	int	4
@content	nvarchar(200)	400
@type	char(3)	3
@correctAns	char	1
@ch1	nvarchar(100)	200
@ch2	nvarchar(100)	200
@ch3	nvarchar(100)	200
@ch4	nvarchar(100)	200

SQL Script

```
CREATE PROCEDURE  [dbo].[PROC_addNewQuestion]
    @courseId INT,
    @content NVARCHAR(200),
    @type CHAR(3),
    @correctAns CHAR(1),
    @ch1 NVARCHAR(100)=NULL,
    @ch2 NVARCHAR(100)=NULL,
    @ch3 NVARCHAR(100)=NULL,
    @ch4 NVARCHAR(100)=NULL
AS
    BEGIN TRY
        IF NOT EXISTS (SELECT * FROM courses WHERE id=@courseId)
            BEGIN
                SELECT 'Course do not exist' AS 'message'
            END
        ELSE
            BEGIN
                IF (@type = 'mcq')
                    BEGIN
                        INSERT INTO questions(course_id, content,
ques_type,correct_ans, A, B, C, D)
                        VALUES(@courseId, @content, @type, @correctAns, @ch1, @ch2,
@ch3, @ch4)
                    END
                ELSE
```

```
                    BEGIN
                        INSERT INTO questions(course_id, content,
ques_type,correct_ans, A, B)
                        VALUES(@courseId, @content, @type, @correctAns, 'T', 'F')
                    END
                --EXECUTE PROC_addChoicesToQuestions @@identity, @ch1, @ch2, @ch3,
@ch4
            END
    END TRY
    BEGIN CATCH
     SELECT ERROR_MESSAGE() AS errorMessage
    END CATCH
GO
```

Uses

[dbo].[courses]
[dbo].[questions]

[dbo].[PROC_deleteAQuestion]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE PROC [dbo].[PROC_deleteAQuestion]
    @questionId INT
AS
BEGIN TRY
        IF NOT EXISTS (SELECT * FROM questions q WHERE id=@questionId)
            BEGIN
                SELECT 'Question does not exist' AS 'errorMessage'
            END
        ELSE
            BEGIN
                DELETE questions
                WHERE id = @questionId
            END
    END TRY
    BEGIN CATCH
     SELECT ERROR_MESSAGE() AS errorMessage
    END CATCH
GO
```

Uses

[dbo].[questions]

[dbo].[PROC_examCorrect]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE PROCEDURE  [dbo].[PROC_examCorrect]
    @exam_id INT,
    @student_id INT,
    @course_id INT
AS
    BEGIN TRY
        IF NOT EXISTS (SELECT * FROM exams WHERE id=@exam_id)
            BEGIN
                SELECT 'Exam do not exist' AS 'errMessage'
            END
        ELSE IF NOT EXISTS (SELECT * FROM students s WHERE s.Std_id=@student_id)
            BEGIN
                SELECT 'Student does not exist' AS 'errMessage'
            END
        ELSE IF NOT EXISTS (SELECT * FROM courses c WHERE c.id=@course_id)
        BEGIN
            SELECT 'Course does not exist' AS 'errMessage'
        END
        ELSE
            BEGIN
                DECLARE @score INT

                SELECT @score = COUNT(*) FROM exam_answers ea
                INNER JOIN exams e ON e.id = ea.exam_id
                INNER JOIN questions q ON ea.question_id = q.id
                WHERE ea.student_id = @student_id AND ea.exam_id = @exam_id AND
q.correct_ans = ea.answer

                UPDATE courses_students SET grade = @score WHERE student_id =
@student_id AND course_id = @course_id
            END
```

```
    END TRY
    BEGIN CATCH
     SELECT ERROR_MESSAGE() AS errorMessage
    END CATCH
GO
```

Uses

[dbo].[courses]
[dbo].[courses_students]
[dbo].[exam_answers]
[dbo].[exams]
[dbo].[questions]
[dbo].[students]

[dbo].[PROC_generate_exam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@examName	nvarchar(100)	200
@courseId	int	4
@duration	int	4
@date	date	3
@fullScore	int	4
@tfNum	int	4
@mcqNum	int	4

SQL Script

```
CREATE PROCEDURE  [dbo].[PROC_generate_exam]
    @examName NVARCHAR(100),
    @courseId INT,
    @duration INT,
    @date DATE,
    @fullScore INT,
    @tfNum INT,
    @mcqNum INT
AS
    BEGIN TRY
        IF NOT EXISTS (SELECT * FROM courses WHERE id=@courseId)
            BEGIN
                SELECT 'Course do not exist' AS 'message'
            END
        ELSE
            BEGIN
            INSERT INTO exams (exam_name, duration, exam_date, full_mark, course_id)
            VALUES (@examName, @duration, @date, @fullScore, @courseId);

            DECLARE @examId INT
            SET @examId = SCOPE_IDENTITY()

            INSERT INTO dbo.exams_questions(exam_id, question_id)
            SELECT TOP(@tfNum) @examId, q.id
            FROM questions q
```

```
            WHERE q.ques_type = 't/f'
            ORDER BY NEWID()

            INSERT INTO dbo.exams_questions(exam_id, question_id)
            SELECT TOP(@mcqNum) @examId, q.id
            FROM questions q
            WHERE q.ques_type = 'mcq'
            ORDER BY NEWID()

            SELECT q.* FROM exams_questions eq, questions q, exams e
            WHERE eq.exam_id = e.id AND eq.question_id = q.id AND eq.exam_id = @exam-
Id
            END
    END TRY
    BEGIN CATCH
     SELECT ERROR_MESSAGE() AS errorMessage
    END CATCH
GO
```

Uses

[dbo].[courses]
[dbo].[exams]
[dbo].[exams_questions]
[dbo].[questions]

[dbo].[PROC_getInstructorCourses]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE PROCEDURE  [dbo].[PROC_getInstructorCourses]
    @instructorId INT
AS
    BEGIN TRY
        IF NOT EXISTS (SELECT * FROM Instructors i WHERE i.Ins_Id=@instructorId)
            BEGIN
                SELECT 'Instructor does not exist' AS 'errorMessage'
            END
        ELSE
            BEGIN
                SELECT COUNT(s.Std_id) AS '#OfStud', c.course_name, i.Fname +
i.Lname AS 'Name' FROM students s, courses c, Instructors i, courses_students cs
                WHERE cs.course_id = c.id AND cs.student_id = s.Std_id AND i.Ins_Id
= c.instructor_id AND c.instructor_id = @instructorId
                GROUP BY c.course_name, i.Fname, i.Lname
            END
    END TRY
    BEGIN CATCH
     SELECT ERROR_MESSAGE() AS errorMessage
    END CATCH
GO
```

Uses

[dbo].[courses]
[dbo].[courses_students]
[dbo].[Instructors]
[dbo].[students]

[dbo].[PROC_getQuestionById]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE PROC [dbo].[PROC_getQuestionById]
    @questionId INT
AS
    BEGIN TRY
        SELECT q.*, c.course_name
        FROM questions q, courses c
        WHERE q.course_id = c.id AND q.id=@questionId
    END TRY
    BEGIN CATCH
     SELECT ERROR_MESSAGE() AS errorMessage
    END CATCH
GO
```

Uses

[dbo].[courses]
[dbo].[questions]

[dbo].[PROC_getStudentAnswerWithModel]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE PROCEDURE  [dbo].[PROC_getStudentAnswerWithModel]
    @exam_id INT,
    @student_id INT,
    @course_id INT
AS
    BEGIN TRY
        IF NOT EXISTS (SELECT * FROM exams WHERE id=@exam_id)
            BEGIN
                SELECT 'Exam do not exist' AS 'errMessage'
            END
        ELSE IF NOT EXISTS (SELECT * FROM students s WHERE s.Std_id=@student_id)
            BEGIN
                SELECT 'Student does not exist' AS 'errMessage'
            END
        ELSE IF NOT EXISTS (SELECT * FROM courses c WHERE c.id=@course_id)
        BEGIN
            SELECT 'Course does not exist' AS 'errMessage'
        END
        ELSE
            BEGIN
                SELECT q.content, q.correct_ans, ea.answer AS 'student_ans' FROM
exam_answers ea
                INNER JOIN exams e ON e.id = ea.exam_id
                INNER JOIN questions q ON ea.question_id = q.id
                WHERE ea.student_id = @student_id AND ea.exam_id = @exam_id AND
q.correct_ans = ea.answer
            END
    END TRY
    BEGIN CATCH
     SELECT ERROR_MESSAGE() AS errorMessage
    END CATCH
```

```
GO
```

Uses

[dbo].[courses]
[dbo].[exam_answers]
[dbo].[exams]
[dbo].[questions]
[dbo].[students]

[dbo].[PROC_IUD]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@id	int	4
@first_name	nchar(10)	20
@last_name	nchar(10)	20
@age	int	4
@email	varchar(25)	25
@street	nchar(10)	20
@City	nchar(20)	40
@zip_Code	char(10)	10
@password	char(20)	20
@salary	decimal(10,2)	9
@Dept_Id	int	4
@ActionType	nvarchar(20)	40

SQL Script

```
create PROCEDURE [dbo].[PROC_IUD] (@id            INTEGER,
                                   @first_name    NCHAR(10),
                                   @last_name     NCHAR(10),
                                   @age           INT,
                                   @email       VARCHAR(25),
                                   @street      NCHAR(10),
                                   @City        NCHAR(20),
                                   @zip_Code     CHAR(10),
                                   @password     CHAR(20),

                                   @salary         DECIMAL(10, 2),
                                       @Dept_Id           INT,

                                   @ActionType NVARCHAR(20) = '')
AS
  BEGIN
      IF @ActionType = 'Insert'
        BEGIN
            INSERT INTO Instructors
```

```
            VALUES      ( @id,
                          @first_name,
                          @last_name,
                         @age,
                         @email,
                         @street,
                         @City,
                         @zip_Code,
                         @password,
                         @salary,
                         @Dept_Id
                         )
        END

      IF @ActionType = 'Select'
        BEGIN
            SELECT *
            FROM   Instructors
        END

      IF @ActionType = 'Update'
        BEGIN
            UPDATE Instructors
            SET    Fname = @first_name,
                   Lname= @last_name,
                   age = @age,
                   email = @email,
                   street=@street,
                   city=@City,
                   Zip_Code=@zip_Code,
                   password=@password,
                   salary=@salary,
                   Dept_Id=@Dept_Id
            WHERE  Ins_Id = @id
        END
      ELSE IF @ActionType = 'Delete'
        BEGIN
            DELETE FROM Instructors
            WHERE  Ins_Id = @id
        END
  END
GO
```

Uses

[dbo].[Instructors]

[dbo].[PROC_IUDD]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@id	int	4
@first_name	nchar(10)	20
@last_name	nchar(10)	20
@age	int	4
@email	varchar(25)	25
@street	nchar(10)	20
@City	nchar(20)	40
@zip_Code	char(10)	10
@password	char(20)	20
@salary	decimal(10,2)	9
@Dept_Id	int	4
@ActionType	nvarchar(20)	40

SQL Script

```
CREATE PROCEDURE [dbo].[PROC_IUDD] (@id  INTEGER,
                                    @first_name   NCHAR(10),
                                    @last_name     NCHAR(10),
                                    @age          INT,
                                    @email       VARCHAR(25),
                                    @street      NCHAR(10),
                                    @City        NCHAR(20),
                                    @zip_Code     CHAR(10),
                                    @password     CHAR(20),

                                    @salary         DECIMAL(10, 2),
                                        @Dept_Id          INT,

                                    @ActionType NVARCHAR(20) = '')
AS
  BEGIN
      IF @ActionType = 'Insert'
        BEGIN TRY
        PRINT ' this is insert statement'
```

```
            INSERT INTO Instructors
            VALUES      ( @id,
                          @first_name,
                          @last_name,
                         @age,
                         @email,
                         @street,
                         @City,
                         @zip_Code,
                         @password,
                         @salary,
                         @Dept_Id
                         )
       EnD TRY
       BEGIN CATCH
          PRINT 'the error is '
        --  +ERROR_MESSAGE();

        END CATCH


      IF @ActionType = 'Select'
        BEGIN TRY

            SELECT *
            FROM   Instructors
         END TRY
         BEGIN CATCH
         PRINT 'the error is '+ERROR_MESSAGE()
         END CATCH
      IF @ActionType = 'Update'
        BEGIN TRY
            UPDATE Instructors
            SET    Fname = @first_name,
                   Lname= @last_name,
                   age = @age,
                   email = @email,
                   street=@street,
                   city=@City,
                   Zip_Code=@zip_Code,
                   password=@password,
                   salary=@salary,
                   Dept_Id=@Dept_Id
            WHERE  Ins_Id = @id
        END TRY
        BEGIN CATCH
         PRINT 'the error is '+ERROR_MESSAGE();
        END CATCH
      ELSE IF @ActionType = 'Delete'
        BEGIN TRY
            DELETE FROM Instructors
            WHERE  Ins_Id = @id
        END TRY
        BEGIN CATCH
          PRINT 'the error is '+ERROR_MESSAGE();
```

```
        END CATCH
  END
GO
```

Uses

[dbo].[Instructors]

[dbo].[PROC_updateAQuestion]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@questionId	int	4
@content	nvarchar(200)	400
@courseId	int	4
@type	char(3)	3
@correctAns	char	1
@ch1	nvarchar(100)	200
@ch2	nvarchar(100)	200
@ch3	nvarchar(100)	200
@ch4	nvarchar(100)	200

SQL Script

```
CREATE PROC [dbo].[PROC_updateAQuestion]
    @questionId INT,
    @content NVARCHAR(200),
    @courseId INT=NULL,
    @type CHAR(3)=NULL,
    @correctAns CHAR(1)=NULL,
    @ch1 NVARCHAR(100)=NULL,
    @ch2 NVARCHAR(100)=NULL,
    @ch3 NVARCHAR(100)=NULL,
    @ch4 NVARCHAR(100)=NULL
AS
BEGIN TRY
        IF NOT EXISTS (SELECT * FROM questions q WHERE id=@questionId)
            BEGIN
                SELECT 'Question does not exist' AS 'errorMessage'
            END
        ELSE IF (@courseId IS NOT NULL AND NOT EXISTS(SELECT * FROM courses WHERE
id=@courseId))
            BEGIN
                SELECT 'Course does not exist' AS 'errorMessage'
            END
        ELSE
            BEGIN
```

```
                IF (@type = 'mcq') BEGIN
                    UPDATE questions SET content=@content,
                    ques_type=COALESCE(@type, ques_type),
                    course_id=COALESCE(@courseId, course_id),
                    correct_ans=COALESCE(@correctAns, correct_ans),
                    A=COALESCE(@ch1, A),
                    B=COALESCE(@ch2, B),
                    C=COALESCE(@ch3, C),
                    D=COALESCE(@ch4, D)
                    WHERE id = @questionId
                END
                ELSE
                    BEGIN
                        UPDATE questions SET content=@content,
                        ques_type=COALESCE(@type, ques_type),
                        course_id=COALESCE(@courseId, course_id),
                        correct_ans=COALESCE(@correctAns, correct_ans),
                        A=COALESCE('T', A),
                        B=COALESCE('F', B),
                        C=NULL,
                        D=NULL
                        WHERE id = @questionId
                    END
            END
    END TRY
    BEGIN CATCH
     SELECT ERROR_MESSAGE() AS errorMessage
    END CATCH
GO
```

Uses

[dbo].[courses]
[dbo].[questions]

[dbo].[reportForExamQues]

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].[reportForExamQues]
  @exam_id INT
  AS
  BEGIN
  BEGIN TRY
  IF NOT EXISTS(SELECT * FROM exams e WHERE e.id=@exam_id)
    PRINT 'This exam is not exist'
  ELSE
  SELECT q.id,q.content , q.A , q.B , q.C, q.D
  FROM questions q , exams e ,exams_questions eq
  WHERE e.id=eq.exam_id AND  eq.question_id =q.id
  END TRY
  BEGIN CATCH
  print 'Select topic faild'
  END CATCH
  END
GO
```

Uses

[dbo].[exams]
[dbo].[exams_questions]
[dbo].[questions]

[dbo].[ReportForIns]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE PROCEDURE [dbo].[ReportForIns]
  @Ins_id INT
  AS
  BEGIN
  BEGIN TRY
  IF NOT EXISTS(SELECT * FROM Instructors i WHERE i.Ins_Id=@Ins_id)
    PRINT 'This Ins is not exist'
  ELSE
  SELECT c.course_name , COUNT(cs.student_id) AS Numberofstudent
  FROM courses c ,courses_students cs,students s, Instructors i
  WHERE cs.course_id=c.id AND c.instructor_id=i.Ins_Id AND cs.student_id=s.Std_id
AND i.Ins_Id=@Ins_id
  GROUP BY c.course_name
  END TRY
  BEGIN CATCH
  print 'Select  faild'
  END CATCH
  END
GO
```

Uses

[dbo].[courses]
[dbo].[courses_students]
[dbo].[Instructors]
[dbo].[students]

[dbo].[reportForStudentAns]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@exam_Id	int	4
@student_Id	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[reportForStudentAns]
    @exam_Id INT, @student_Id INT
AS
BEGIN
BEGIN TRY
    SELECT q.content , ea.answer
    FROM questions q , exam_answers ea,exams e,students s
    WHERE ea.question_id=q.id  AND ea.student_id =s.Std_id AND e.id=ea.exam_id AND
ea.id=@exam_Id AND ea.student_id=@student_Id
END TRY
BEGIN CATCH
PRINT 'Select Report faild'
END CATCH
END
GO
```

Uses

[dbo].[exam_answers]
[dbo].[exams]
[dbo].[questions]
[dbo].[students]

[dbo].[reportForStudentAnss]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@exam_Id	int	4
@student_Id	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[reportForStudentAnss]
    @exam_Id INT, @student_Id INT
    AS

    BEGIN
    BEGIN TRY
    SELECT q.content , ea.answer
    FROM questions q , exam_answers ea,exams e,students s

    WHERE ea.question_id=q.id  AND ea.student_id =s.Std_id AND e.id=ea.exam_id AND
ea.id=@exam_Id AND ea.student_id=@student_Id
END TRY
     BEGIN CATCH
  print 'Select topic faild'
  END CATCH
  END
GO
```

Uses

[dbo].[exam_answers]
[dbo].[exams]
[dbo].[questions]
[dbo].[students]

[dbo].[reportForTopic]

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 PROCEDURE [dbo].[reportForTopic]
  @course_id INT
AS
BEGIN
BEGIN TRY
IF NOT EXISTS(SELECT * FROM topic t WHERE t.course_id=@course_id)
  PRINT 'This course is not exist'
ELSE
SELECT T.topic_name
FROM topic t
WHERE t.course_id=@course_id
END TRY
BEGIN CATCH
print 'Select topic faild'
END CATCH
END
--Calling
EXECUTE reportForTopic 25
-----------------------------------------------
--Create stored procedure that takes exam number and returns the Questions in it and
chocies [freeform report]
DROP PROCEDURE IF EXISTS reportForExamQues
GO
```

Uses

[dbo].[topic]

[dbo].[SelectAllIns]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE PROCEDURE [dbo].[SelectAllIns]

    AS
BEGIN

  begin try
   SELECT * FROM  Instructors
   PRINT ' this is all the instructors '
  end try

  begin catch
      PRINT 'the error is'+ERROR_MESSAGE()
  end catch
END
GO
```

Uses

[dbo].[Instructors]

[dbo].[selectAllInstructors]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE PROCEDURE [dbo].[selectAllInstructors]
AS
BEGIN TRy
SELECT * FROM Instructors
END TRY

BEGIN CATCH
 PRINT 'the error is'+ ERROR_MESSAGE()
END CATCH
GO
```

Uses

[dbo].[Instructors]

[dbo].[selectAllStudents]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE PROCEDURE [dbo].[selectAllStudents]
AS
BEGIN TRY
SELECT * FROM students
END TRY
BEGIN CATCH
print 'select topic faild'
END CATCH
--Calling
GO
```

Uses

[dbo].[students]

[dbo].[selectCourseById]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE PROCEDURE [dbo].[selectCourseById]
  @id INT
AS
BEGIN
BEGIN TRY
SELECT * FROM courses c
WHERE c.id=@id
END TRY
BEGIN CATCH
PRINT  'Faild to select courses'+ERROR_MESSAGE()
END CATCH
END
GO
```

Uses

[dbo].[courses]

[dbo].[selectCourses]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE PROCEDURE [dbo].[selectCourses]
AS
BEGIN
BEGIN TRY
SELECT * FROM courses
END TRY
BEGIN CATCH
PRINT 'Faild to select courses'+ERROR_MESSAGE()
END CATCH
END
GO
```

Uses

[dbo].[courses]

[dbo].[selectOne]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@TableName	nvarchar(50)	100
@Fields	nvarchar(max)	max
@WhereClause	nvarchar(max)	max

SQL Script

```
CREATE PROCEDURE [dbo].[selectOne]
    @TableName nvarchar(50),
    @Fields nvarchar(max) = '*',
    @WhereClause nvarchar(max) = ''
AS
BEGIN TRY
    DECLARE @sql nvarchar(max)
    SET @sql = 'SELECT ' + @Fields + ' FROM ' + @TableName
    IF @WhereClause <> ''
    SET @sql = @sql + ' WHERE ' + @WhereClause
    EXEC sp_executesql @sql
END TRY
BEGIN CATCH
    SELECT 'record not found'
END CATCH
GO
```

[dbo].[selectStudentById]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE PROCEDURE [dbo].[selectStudentById]
  @Std_id INT
AS
BEGIN TRY
SELECT s.*
FROM students s
WHERE s.Std_id=@Std_id
END TRY
BEGIN CATCH
print 'select student faild'
END CATCH
--Calling
EXECUTE selectStudentById 3


------------------------------------------------------------
-- insert new student PROCEDURE --
------------------------------------------------------------
DROP PROCEDURE IF EXISTS insertStudent
GO
```

Uses

[dbo].[students]

[dbo].[selectTopic]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE PROCEDURE [dbo].[selectTopic]
AS
BEGIN TRY
SELECT * FROM topic
END TRY
BEGIN CATCH
print 'Select topic faild'
END CATCH
GO
```

Uses

[dbo].[topic]

[dbo].[selectTopicById]

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 PROCEDURE [dbo].[selectTopicById]
  @course_id INT
AS
BEGIN TRY
SELECT t.* , c.course_name
FROM topic t , courses c
WHERE t.course_id=@course_id AND c.id=@course_id
END TRY
BEGIN CATCH
print 'Select topic faild'
END CATCH
GO
```

Uses

[dbo].[courses]
[dbo].[topic]

[dbo].[TestlastReport]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@exam_Id	int	4
@student_Id	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[TestlastReport]
    @exam_Id INT, @student_Id INT
    AS

    BEGIN
    BEGIN TRY
    SELECT *
    FROM  exam_answers ,students s

    WHERE exam_answers.student_id=s.Std_id AND exam_id=@exam_Id AND
s.Std_id=@student_Id
END TRY
     BEGIN CATCH
  print 'Select  faild'
  END CATCH
  END
GO
```

Uses

[dbo].[exam_answers]
[dbo].[students]

[dbo].[updateCourses]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@id	int	4
@course_name	nvarchar(200)	400

SQL Script

```
CREATE PROCEDURE [dbo].[updateCourses]
  @id INT,
  @course_name NVARCHAR(200)
AS
BEGIN
BEGIN TRY
IF NOT EXISTS(SELECT * FROM courses WHERE id=@id)
        PRINT'Course is not exist'
ELSE
UPDATE courses
SET  course_name = @course_name
WHERE id = @id;
END TRY
BEGIN CATCH
PRINT 'Error updating courses'+ERROR_MESSAGE()
--THROW;
END CATCH;
END
GO
```

Uses

[dbo].[courses]

[dbo].[UpdateInstructor]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Id	int	4
@first_name	nchar(10)	20
@last_name	nchar(10)	20
@age	int	4
@email	varchar(25)	25
@street	nchar(10)	20
@City	nchar(20)	40
@zip_Code	char(10)	10
@password	char(20)	20
@salary	decimal(10,2)	9
@Dept_Id	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[UpdateInstructor]( @Id  INT,
                                           @first_name   NCHAR(10),
                                          @last_name     NCHAR(10),
                                         @age          INT,
                                         @email      VARCHAR(25),
                                         @street     NCHAR(10),
                                         @City       NCHAR(20),
                                         @zip_Code    CHAR(10),
                                         @password    CHAR(20),

                                         @salary        DECIMAL(10, 2),
                                           @Dept_Id          INT
)
AS
BEGIN
DECLARE @Rowcount INT =0
BEGIN TRY
SET @Rowcount = (SELECT COUNT(1) FROM Instructors i  WHERE i.Ins_Id=@Id)
IF (@Rowcount>0)
```

```
 BEGIN
   BEGIN TRAN

UPDATE  Instructors
SET
                        Ins_Id=@Id,
                       Fname= @first_name,
                      Lname=  @last_name,
                      age= @age,
                       email=@email,
                       street=@street,
                        City=@City,
                       Zip_Code=@zip_Code,
                          password=@password,
                       salary=@salary,
                   Dept_Id=@Dept_Id
                   WHERE Ins_Id = @Id
COMMIT TRAN
END
END TRY
BEGIN CATCH

  PRINT 'the Error is'+ ERROR_MESSAGE()
END CATCH
END
GO
```

Uses

[dbo].[Instructors]

[dbo].[updateStudent]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Std_id	int	4
@Fname	nvarchar(20)	40
@Lname	nvarchar(20)	40
@age	int	4
@email	nvarchar(20)	40
@Street	nvarchar(20)	40
@city	nvarchar(20)	40
@Zip_code	int	4
@phone	nvarchar(20)	40
@password	nvarchar(20)	40
@Dept_id	int	4
@supervise_id	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[updateStudent]
   @Std_id INT,
   @Fname NVARCHAR(20)=NULL,
   @Lname NVARCHAR(20)=NULL,
   @age INT=NULL,
   @email NVARCHAR(20)=NULL,
   @Street NVARCHAR(20)=NULL,
   @city NVARCHAR(20)=NULL,
   @Zip_code INT=NULL,
   @phone NVARCHAR(20)=NULL,
   @password NVARCHAR(20)=NULL,
   @Dept_id INT=NULL,
   @supervise_id INT=NULL
AS
BEGIN
BEGIN TRY
IF NOT EXISTS(SELECT * FROM students WHERE Std_id=@Std_id)
        PRINT 'The student is not exist'
ELSE
```

```
BEGIN
UPDATE students
SET Fname=COALESCE(@Fname,Fname),
    Lname=COALESCE(@Lname,Lname),
    age=COALESCE(@age,age),
    email=COALESCE(@email,email),
    Street=COALESCE(@Street,Street),
    city=COALESCE(@city,city),
    Zip_code=COALESCE(@Zip_code,Zip_code),
    phone=COALESCE(@phone,phone),
    password=COALESCE(@password,password),
    Dept_id=COALESCE(@Dept_id,Dept_id),
    supervise_id=COALESCE(@supervise_id,supervise_id)

WHERE Std_id=@Std_id
END
END TRY
BEGIN CATCH
PRINT 'update student failed'
END CATCH
END
GO
```

Uses

[dbo].[students]

[dbo].[UpdateTable]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@TableName	nvarchar(50)	100
@ColumnName	nvarchar(50)	100
@ColumnValue	nvarchar(50)	100
@Condition	nvarchar(max)	max

SQL Script

```
CREATE PROCEDURE [dbo].[UpdateTable]
    @TableName nvarchar(50),
    @ColumnName nvarchar(50),
    @ColumnValue nvarchar(50),
    @Condition nvarchar(max)
AS
begin try
    DECLARE @sql nvarchar(max)
    SET @sql = 'UPDATE ' + @TableName + ' SET ' + @ColumnName + ' = @ColumnValue
WHERE ' + @Condition
    EXEC sp_executesql @sql, N'@ColumnValue nvarchar(50)', @ColumnValue
end try
begin CATCH
    select 'record not found'
end catch
GO
```

[dbo].[updateTopic]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@id	int	4
@course_id	int	4
@topic_name	nvarchar(200)	400

SQL Script

```
CREATE PROCEDURE [dbo].[updateTopic]
   @id INT,
   @course_id INT=NULL,
   @topic_name NVARCHAR(200)=NULL
AS
BEGIN
BEGIN TRY
IF NOT EXISTS(SELECT * FROM topic WHERE id=@id)
        PRINT 'The topic is not exist'
ELSE IF @course_id IS NOT NULL AND NOT EXISTS(SELECT * FROM courses c WHERE
c.id=@course_id)
        PRINT 'The Course is not exist'
ELSE
BEGIN
UPDATE topic
SET course_id=COALESCE(@course_id,course_id),
    topic_name=COALESCE(@topic_name,topic_name)
WHERE id=@id
END
END TRY
BEGIN CATCH
PRINT 'Update topic failed'
END CATCH
END
GO
```

Uses

[dbo].[courses]

[dbo].[topic]

Users

Objects

Name
dbo

dbo

Properties

Property	Value
Type	SqlUser
Login Name	sql_iti
Default Schema	dbo

Database Level Permissions

Type	Action
CONNECT	Grant

SQL Script

```
GO
```