Database Dictionary

Table of Contents

User databases

Databases (1)

- SQL_Project

SQL_Project Database

Files

Name	Type	Size	Maxsize	Autogrowth	File Name
SQL_Project	Data	72.00 MB	unlimited	64.00 MB	C:\Program Files\Microsoft SQL Server\MSSQL16.SQLEXPRESS\MSSQL\DATA\SQL_Project.mdf
SQL_Project_log	Log	8.00 MB	2048.00 GB	64.00 MB	C:\Program Files\Microsoft SQL Server\MSSQL16.SQLEXPRESS\MSSQL\DATA\SQL_Project_log.ldf

Tables

Objects

Name
dbo.Course
dbo.DBAudit
dbo.Department
dbo.DeptCourse
dbo.Exam
dbo.Exam_Question
dbo.Instructor
dbo.QuesChoices
dbo.Question
dbo.StdCourse
dbo.StdExam
dbo.Student
dbo.Topic

[dbo].[Course]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	5
Created	3:48:56 PM Friday, February 9, 2024
Last Modified	12:39:48 AM Saturday, February 24, 2024

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity	Default
PK C	Crs_ID	int	4	NOT NULL	1 - 1	
🖧	Crs_Name	varchar(25)	25	NOT NULL		
	Crs_Duration	int	4	NULL allowed		((25))
FK	Ins_ID	int	4	NULL allowed		

Indexes

Key	Name	Key Columns	Unique
PK C	PK__Course__56CAA5F5342B3217	Crs_ID	True
	UQ__Course__E2D24123BC708597	Crs_Name	True

Foreign Keys

Name	Delete	Columns
FK_Course_Instructor	SetNull	Ins_ID->[dbo].[Instructor].[Ins_SSN]

SQL Script

```sql
CREATE TABLE [dbo].[Course]
(
[Crs_ID] [int] NOT NULL IDENTITY(1, 1),
[Crs_Name] [varchar] (25) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
[Crs_Duration] [int] NULL CONSTRAINT [DF__Course__Crs_Dura__4D94879B] DEFAULT ((25)),
[Ins_ID] [int] NULL
) ON [PRIMARY]
GO
```

```
ALTER TABLE [dbo].[Course] ADD CONSTRAINT [PK__Course__56CAA5F5342B3217] PRIMARY KEY CLUSTERED
([Crs_ID]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Course] ADD CONSTRAINT [UQ__Course__E2D24123BC708597] UNIQUE NONCLUSTERED
([Crs_Name]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Course] ADD CONSTRAINT [FK_Course_Instructor] FOREIGN KEY ([Ins_ID])
REFERENCES [dbo].[Instructor] ([Ins_SSN]) ON DELETE SET NULL
GO
```

Uses

[dbo].[Instructor]

Used By

[dbo].[DeptCourse]
[dbo].[Question]
[dbo].[StdCourse]
[dbo].[Topic]
[dbo].[SP_DeleteCourse]
[dbo].[SP_GetStdCourses]
[dbo].[SP_InsertCourse]
[dbo].[SP_InsertDeptCourse]
[dbo].[SP_InsertQuestion]
[dbo].[SP_InsertStdCourse]
[dbo].[SP_ReopSTDCourseGrade]
[dbo].[SP_SelectCourse]
[dbo].[SP_SelectDeptCourse]
[dbo].[SP_UpdateCourse]
[dbo].[SP_UpdateQuestion]
[dbo].[SP_UpdateStdCourse]
[dbo].[SP_UpdateTopic]
[repo].[SP_ExamQuestions]
[repo].[SP_InstructorInfo]
[repo].[SP_STDCourseGrade]

[dbo].[DBAudit]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Heap	True
Row Count (~)	1
Created	12:36:25 AM Sunday, February 18, 2024
Last Modified	12:36:26 AM Sunday, February 18, 2024

Columns

Name	Data Type	Max Length (Bytes)	Nullability	Default
Entity	varchar(50)	50	NOT NULL	
UserID	varchar(50)	50	NOT NULL	(suser_sname())
Action	varchar(50)	50	NOT NULL	('Delete')
Date	datetime	8	NOT NULL	(getdate())

SQL Script

```
CREATE TABLE [dbo].[DBAudit]
(
[Entity] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
[UserID] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL CONSTRAINT [DF_DBAudit_-
UserID] DEFAULT (suser_sname()),
[Action] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL CONSTRAINT [DF_Table_1_-
QueryAction] DEFAULT ('Delete'),
[Date] [datetime] NOT NULL CONSTRAINT [DF_DBAudit_Date] DEFAULT (getdate())
) ON [PRIMARY]
GO
```

Used By

[dbo].[SP_DeleteStdExam]
[dbo].[SP_UpdateDeptCourse]
[dbo].[SP_UpdateExamQues]
[dbo].[SP_UpdateStdExam]

[dbo].[Department]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	5
Created	11:02:12 PM Saturday, February 17, 2024
Last Modified	12:44:32 AM Wednesday, February 21, 2024

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK C	Dept_ID	int	4	NOT NULL	1 - 1
⋔	Dept_Name	varchar(15)	15	NOT NULL	
FK	Manager_ID	int	4	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
PK C	PK__Departme__72ABC12C691CA668	Dept_ID	True
	UQ__Departme__B744FF0FD6262392	Dept_Name	True

Foreign Keys

Name	Delete	Columns
FK_Department_Instructor	SetNull	Manager_ID->[dbo].[Instructor].[Ins_SSN]

SQL Script

```
CREATE TABLE [dbo].[Department]
(
[Dept_ID] [int] NOT NULL IDENTITY(1, 1),
[Dept_Name] [varchar] (15) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
[Manager_ID] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Department] ADD CONSTRAINT [PK__Departme__72ABC12C691CA668] PRIMARY KEY
CLUSTERED ([Dept_ID]) ON [PRIMARY]
GO
```

```sql
ALTER TABLE [dbo].[Department] ADD CONSTRAINT [UQ__Departme__B744FF0FD6262392] UNIQUE
NONCLUSTERED ([Dept_Name]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Department] ADD CONSTRAINT [FK_Department_Instructor] FOREIGN KEY ([Manager_-
ID]) REFERENCES [dbo].[Instructor] ([Ins_SSN]) ON DELETE SET NULL
GO
```

Uses

[dbo].[Instructor]

Used By

[dbo].[DeptCourse]
[dbo].[Instructor]
[dbo].[Student]
[dbo].[SP_DeleteDepartment]
[dbo].[SP_InsertDepartment]
[dbo].[SP_InsertDeptCourse]
[dbo].[SP_InsertInstructor]
[dbo].[SP_InsertStudent]
[dbo].[SP_SelectDepartment]
[dbo].[SP_SelectDeptCourse]
[dbo].[SP_UpdateDepartment]
[dbo].[SP_UpdateStudent]
[repo].[SP_DeptStudentsInfo]
[repo].[SP_STDCourseGrade]
[repo].[SP_StudentExamAns]

[dbo].[DeptCourse]

Properties

Property	Value
Row Count (~)	16
Created	3:48:56 PM Friday, February 9, 2024
Last Modified	12:26:59 AM Wednesday, February 21, 2024

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PK C FK	Dept_ID	int	4	NOT NULL
PK C FK	Crs_ID	int	4	NOT NULL

Indexes

Key	Name	Key Columns	Unique
PK C	PK__DeptCour__37C76B735316AE12	Dept_ID, Crs_ID	True

Foreign Keys

Name	Update	Delete	Columns
FK_DeptCourse_Course	Cascade	Cascade	Crs_ID->[dbo].[Course].[Crs_ID]
FK_DeptCourse_Department		Cascade	Dept_ID->[dbo].[Department].[Dept_ID]

SQL Script

```sql
CREATE TABLE [dbo].[DeptCourse]
(
[Dept_ID] [int] NOT NULL,
[Crs_ID] [int] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[DeptCourse] ADD CONSTRAINT [PK__DeptCour__37C76B735316AE12] PRIMARY KEY
CLUSTERED ([Dept_ID], [Crs_ID]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[DeptCourse] ADD CONSTRAINT [FK_DeptCourse_Course] FOREIGN KEY ([Crs_ID])
REFERENCES [dbo].[Course] ([Crs_ID]) ON DELETE CASCADE ON UPDATE CASCADE
GO
ALTER TABLE [dbo].[DeptCourse] ADD CONSTRAINT [FK_DeptCourse_Department] FOREIGN KEY ([Dept_ID])
```

```
REFERENCES [dbo].[Department] ([Dept_ID]) ON DELETE CASCADE
GO
```

Uses

[dbo].[Course]

[dbo].[Department]

Used By

[dbo].[SP_DeleteDeptCourse]

[dbo].[SP_InsertDeptCourse]

[dbo].[SP_SelectDeptCourse]

Project > (local)\SQLEXPRESS > User databases > SQL_Project >
Tables > dbo.Exam

[dbo].[Exam]

Properties

Property	Value
Row Count (~)	20
Created	3:48:56 PM Friday, February 9, 2024
Last Modified	10:32:49 PM Saturday, February 17, 2024

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity	Default
PK C	Exam_ID	int	4	NOT NULL	1 - 1	
	Exam_Duration	int	4	NULL allowed		((60))

Indexes

Key	Name	Key Columns	Unique
PK C	PK__Exam__C782CA7994CE8A16	Exam_ID	True

SQL Script

```
CREATE TABLE [dbo].[Exam]
(
[Exam_ID] [int] NOT NULL IDENTITY(1, 1),
[Exam_Duration] [int] NULL CONSTRAINT [DF__Exam__Exam_Durat__4316F928] DEFAULT ((60))
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Exam] ADD CONSTRAINT [PK__Exam__C782CA7994CE8A16] PRIMARY KEY CLUSTERED
([Exam_ID]) ON [PRIMARY]
GO
```

Used By

[dbo].[Exam_Question]
[dbo].[StdExam]
[dbo].[SP_DeleteExam]
[dbo].[SP_ExamGeneration]
[dbo].[SP_InsertExam]
[dbo].[SP_SelectExam]
[dbo].[SP_SelectExamQues]

[dbo].[SP_UpdateExam]

Project > (local)\SQLEXPRESS > User databases > SQL_Project > Tables > dbo.Exam_Question

[dbo].[Exam_Question]

Properties

Property	Value
Row Count (~)	200
Created	3:48:56 PM Friday, February 9, 2024
Last Modified	1:28:53 AM Sunday, February 18, 2024

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PK C FK	Exam_ID	int	4	NOT NULL
PK C FK	Ques_ID	int	4	NOT NULL

Indexes

Key	Name	Key Columns	Unique
PK C	PK__Exam_Que__3D00D852AE941201	Exam_ID, Ques_ID	True

Foreign Keys

Name	Columns
FK_Exam_Question_Exam	Exam_ID->[dbo].[Exam].[Exam_ID]
FK_Exam_Question_Question	Ques_ID->[dbo].[Question].[Ques_ID]

SQL Script

```
CREATE TABLE [dbo].[Exam_Question]
(
[Exam_ID] [int] NOT NULL,
[Ques_ID] [int] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Exam_Question] ADD CONSTRAINT [PK__Exam_Que__3D00D852AE941201] PRIMARY KEY
CLUSTERED ([Exam_ID], [Ques_ID]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Exam_Question] ADD CONSTRAINT [FK_Exam_Question_Exam] FOREIGN KEY ([Exam_ID])
REFERENCES [dbo].[Exam] ([Exam_ID])
GO
ALTER TABLE [dbo].[Exam_Question] ADD CONSTRAINT [FK_Exam_Question_Question] FOREIGN KEY ([Ques_-
ID]) REFERENCES [dbo].[Question] ([Ques_ID])
GO
```

Uses

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

Used By

[dbo].[SP_DeleteExam]
[dbo].[SP_DeleteExamQues]
[dbo].[SP_DeleteQuestion]
[dbo].[SP_ExamAnswers]
[dbo].[SP_ExamGeneration]
[dbo].[SP_InsertExamQues]
[dbo].[SP_RepoExamQuestions]
[dbo].[SP_SelectExamQues]
[repo].[SP_ExamQuestions]

Project > (local)\SQLEXPRESS > User databases > SQL_Project >
Tables > dbo.Instructor

[dbo].[Instructor]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	5
Created	3:48:56 PM Friday, February 9, 2024
Last Modified	12:44:32 AM Wednesday, February 21, 2024

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK C	Ins_SSN	int	4	NOT NULL	1 - 1
	Ins_Fname	varchar(25)	25	NOT NULL	
	Ins_Lname	varchar(25)	25	NOT NULL	
	Ins_Phone	varchar(11)	11	NULL allowed	
	Ins_Salary	int	4	NULL allowed	
	Ins_Address	varchar(50)	50	NULL allowed	
FK	Dept_ID	int	4	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
PK C	PK__Instruct__5F8D95E9370769AD	Ins_SSN	True

Foreign Keys

Name	Delete	Columns
FK_Instructor_Department	SetNull	Dept_ID->[dbo].[Department].[Dept_ID]

SQL Script

```
CREATE TABLE [dbo].[Instructor]
(
[Ins_SSN] [int] NOT NULL IDENTITY(1, 1),
[Ins_Fname] [varchar] (25) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
[Ins_Lname] [varchar] (25) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
[Ins_Phone] [varchar] (11) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[Ins_Salary] [int] NULL,
[Ins_Address] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[Dept_ID] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Instructor] ADD CONSTRAINT [PK__Instruct__5F8D95E9370769AD] PRIMARY KEY
CLUSTERED ([Ins_SSN]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Instructor] ADD CONSTRAINT [FK_Instructor_Department] FOREIGN KEY ([Dept_ID])
REFERENCES [dbo].[Department] ([Dept_ID]) ON DELETE SET NULL
GO
```

Uses

[dbo].[Department]

Used By

[dbo].[Course]
[dbo].[Department]
[dbo].[SP_InsertCourse]
[dbo].[SP_InsertDepartment]
[dbo].[SP_InsertInstructor]
[dbo].[SP_SelectInstructor]
[repo].[SP_InstructorInfo]
[repo].[SP_STDCourseGrade]

[dbo].[QuesChoices]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	300
Created	12:55:09 AM Sunday, February 25, 2024
Last Modified	10:28:18 PM Sunday, February 25, 2024

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PK C FK	Ques_ID	int	4	NOT NULL
PK C	Ques_Choices	varchar(250)	250	NOT NULL

Indexes

Key	Name	Key Columns	Unique
PK C	PK_QuesChoices	Ques_ID, Ques_Choices	True

Foreign Keys

Name	Update	Delete	Columns
FK_QuesChoices_Question	Cascade	Cascade	Ques_ID->[dbo].[Question].[Ques_ID]

SQL Script

```sql
CREATE TABLE [dbo].[QuesChoices]
(
[Ques_ID] [int] NOT NULL,
[Ques_Choices] [varchar] (250) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[QuesChoices] ADD CONSTRAINT [PK_QuesChoices] PRIMARY KEY CLUSTERED ([Ques_ID],
[Ques_Choices]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[QuesChoices] ADD CONSTRAINT [FK_QuesChoices_Question] FOREIGN KEY ([Ques_ID])
REFERENCES [dbo].[Question] ([Ques_ID]) ON DELETE CASCADE ON UPDATE CASCADE
GO
```

Uses

[dbo].[Question]

Used By

[dbo].[SP_DeleteQuesChoice]
[dbo].[SP_ExamGeneration]
[dbo].[SP_InsertQuesChoice]
[dbo].[SP_RepoExamQuestions]
[dbo].[SP_SelectQuesChoices]
[dbo].[SP_UpdateQuesChoice]
[repo].[SP_ExamQuestions]
[repo].[SP_StudentExamAns]

Project > (local)\SQLEXPRESS > User databases > SQL_Project >
Tables > dbo.Question

[dbo].[Question]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	100
Created	1:28:53 AM Sunday, February 18, 2024
Last Modified	12:55:10 AM Sunday, February 25, 2024

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK C	Ques_ID	int	4	NOT NULL	1 - 1
	Ques_Type	varchar(100)	100	NOT NULL	
	Ques_Body	text	max	NOT NULL	
	Ques_Answer	varchar(1)	1	NOT NULL	
	Ques_Grade	int	4	NOT NULL	
FK	Crs_ID	int	4	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
PK C	PK__Question__A82122BE4D6210C6	Ques_ID	True

Foreign Keys

Name	Update	Delete	Columns
FK_Question_Course	Cascade	SetNull	Crs_ID->[dbo].[Course].[Crs_ID]

SQL Script

```
CREATE TABLE [dbo].[Question]
(
[Ques_ID] [int] NOT NULL IDENTITY(1, 1),
[Ques_Type] [varchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
[Ques_Body] [text] COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
[Ques_Answer] [varchar] (1) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
[Ques_Grade] [int] NOT NULL,
[Crs_ID] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Question] ADD CONSTRAINT [PK__Question__A82122BE4D6210C6] PRIMARY KEY
CLUSTERED ([Ques_ID]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Question] ADD CONSTRAINT [FK_Question_Course] FOREIGN KEY ([Crs_ID])
REFERENCES [dbo].[Course] ([Crs_ID]) ON DELETE SET NULL ON UPDATE CASCADE
GO
```

Uses

[dbo].[Course]

Used By

[dbo].[Exam_Question]
[dbo].[QuesChoices]
[dbo].[StdExam]
[dbo].[SP_DeleteQuestion]
[dbo].[SP_ExamCorrection]
[dbo].[SP_ExamGeneration]
[dbo].[SP_InsertExamQues]
[dbo].[SP_InsertQuestion]
[dbo].[SP_RepoExamQuestions]
[dbo].[SP_RepoStudentAns]
[dbo].[SP_SelectQuestion]
[dbo].[SP_UpdateQuestion]
[repo].[SP_ExamQuestions]
[repo].[SP_StudentExamAns]

Project > (local)\SQLEXPRESS > User databases > SQL_Project > Tables > dbo.StdCourse

[dbo].[StdCourse]

Properties

Property	Value
Row Count (~)	64
Created	3:48:56 PM Friday, February 9, 2024
Last Modified	2:00:59 AM Sunday, February 18, 2024

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PK FK FK	St_ID	int	4	NOT NULL
PK FK FK	Crs_ID	int	4	NOT NULL
	Grade	int	4	NULL allowed

Indexes

Key	Name	Key Columns	Unique
PK	PK__StdCours__FC6FE5CCA16E6E7F	St_ID, Crs_ID	True

Foreign Keys

Name	Update	Delete	Columns
FK_StdCourse_Course	Cascade	Cascade	Crs_ID->[dbo].[Course].[Crs_ID]
FK_StdCourse_Student	Cascade	Cascade	St_ID->[dbo].[Student].[St_ID]

SQL Script

```
CREATE TABLE [dbo].[StdCourse]
(
[St_ID] [int] NOT NULL,
[Crs_ID] [int] NOT NULL,
[Grade] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[StdCourse] ADD CONSTRAINT [PK__StdCours__FC6FE5CCA16E6E7F] PRIMARY KEY
CLUSTERED ([St_ID], [Crs_ID]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[StdCourse] ADD CONSTRAINT [FK_StdCourse_Course] FOREIGN KEY ([Crs_ID])
REFERENCES [dbo].[Course] ([Crs_ID]) ON DELETE CASCADE ON UPDATE CASCADE
```

```
GO
ALTER TABLE [dbo].[StdCourse] ADD CONSTRAINT [FK_StdCourse_Student] FOREIGN KEY ([St_ID])
REFERENCES [dbo].[Student] ([St_ID]) ON DELETE CASCADE ON UPDATE CASCADE
GO
```

Uses

[dbo].[Course]

[dbo].[Student]

Used By

[dbo].[SP_DeleteStdCourse]

[dbo].[SP_ExamCorrection]

[dbo].[SP_GetStdCourses]

[dbo].[SP_InsertStdCourse]

[dbo].[SP_ReopSTDCourseGrade]

[dbo].[SP_SelectStdCourse]

[dbo].[SP_UpdateStdCourse]

[repo].[SP_InstructorInfo]

[repo].[SP_STDCourseGrade]

[dbo].[StdExam]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	70
Created	3:48:56 PM Friday, February 9, 2024
Last Modified	7:00:24 PM Wednesday, February 28, 2024

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
FK	St_SSN	int	4	NOT NULL
FK C FK	Exam_ID	int	4	NOT NULL
FK C FK	Ques_ID	int	4	NOT NULL
	St_Answer	varchar(50)	50	NOT NULL
	Crs_ID	int	4	NULL allowed

Indexes

Key	Name	Key Columns	Unique
FK C	PK_StdExam	Exam_ID, Ques_ID	True

Foreign Keys

Name	Update	Delete	Columns
FK_StdExam_Exam			Exam_ID->[dbo].[Exam].[Exam_ID]
FK_StdExam_Question			Ques_ID->[dbo].[Question].[Ques_ID]
FK_StdExam_Student	Cascade	Cascade	St_SSN->[dbo].[Student].[St_ID]

SQL Script

```
CREATE TABLE [dbo].[StdExam]
(
[St_SSN] [int] NOT NULL,
[Exam_ID] [int] NOT NULL,
[Ques_ID] [int] NOT NULL,
[St_Answer] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
[Crs_ID] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[StdExam] ADD CONSTRAINT [PK_StdExam] PRIMARY KEY CLUSTERED ([Exam_ID], [Ques_ID]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[StdExam] ADD CONSTRAINT [FK_StdExam_Exam] FOREIGN KEY ([Exam_ID]) REFERENCES [dbo].[Exam] ([Exam_ID])
GO
ALTER TABLE [dbo].[StdExam] ADD CONSTRAINT [FK_StdExam_Question] FOREIGN KEY ([Ques_ID]) REFERENCES [dbo].[Question] ([Ques_ID])
GO
ALTER TABLE [dbo].[StdExam] ADD CONSTRAINT [FK_StdExam_Student] FOREIGN KEY ([St_SSN]) REFERENCES [dbo].[Student] ([St_ID]) ON DELETE CASCADE ON UPDATE CASCADE
GO
```

Uses

[dbo].[Exam]

[dbo].[Question]

[dbo].[Student]

Used By

[dbo].[SP_ExamAnswers]

[dbo].[SP_ExamCorrection]

[dbo].[SP_InsertStdExam]

[dbo].[SP_RepoStudentAns]

[dbo].[SP_SelectStdExam]

[repo].[SP_StudentExamAns]

Project > (local)\SQLEXPRESS > User databases > SQL_Project >
Tables > dbo.Student

[dbo].[Student]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	20
Created	10:39:02 PM Saturday, February 17, 2024
Last Modified	9:07:28 AM Tuesday, February 27, 2024

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK C	St_ID	int	4	NOT NULL	1 - 1
	St_Fname	varchar(25)	25	NOT NULL	
	St_Lname	varchar(25)	25	NOT NULL	
	St_Phone	varchar(11)	11	NULL allowed	
	St_Address	varchar(50)	50	NULL allowed	
	St_BD	date	3	NULL allowed	
FK	Dept_ID	int	4	NULL allowed	
	St_SSN	varchar(14)	14	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
PK C	PK__Student__B9034F935ADD2437	St_ID	True

Foreign Keys

Name	Delete	Columns
FK_Student_Department	SetNull	Dept_ID->[dbo].[Department].[Dept_ID]

SQL Script

```
CREATE TABLE [dbo].[Student]
(
[St_ID] [int] NOT NULL IDENTITY(1, 1),
[St_Fname] [varchar] (25) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
[St_Lname] [varchar] (25) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
[St_Phone] [varchar] (11) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[St_Address] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[St_BD] [date] NULL,
[Dept_ID] [int] NULL,
[St_SSN] [varchar] (14) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Student] ADD CONSTRAINT [PK__Student__B9034F935ADD2437] PRIMARY KEY CLUSTERED
([St_ID]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Student] ADD CONSTRAINT [FK_Student_Department] FOREIGN KEY ([Dept_ID])
REFERENCES [dbo].[Department] ([Dept_ID]) ON DELETE SET NULL
GO
```

Uses

[dbo].[Department]

Used By

[dbo].[StdCourse]
[dbo].[StdExam]
[dbo].[SP_DeleteStudent]
[dbo].[SP_InsertStdCourse]
[dbo].[SP_InsertStudent]
[dbo].[SP_logIN]
[dbo].[SP_ReopSTDCourseGrade]
[dbo].[SP_SelectStudent]
[dbo].[SP_UpdateStudent]
[repo].[SP_DeptStudentsInfo]
[repo].[SP_STDCourseGrade]
[repo].[SP_StudentExamAns]

[dbo].[Topic]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	28
Created	12:39:48 AM Saturday, February 24, 2024
Last Modified	12:39:48 AM Saturday, February 24, 2024

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK C FK	Crs_ID	int	4	NOT NULL	
PK C	Topic_ID	int	4	NOT NULL	1 - 1
	Topic_Name	varchar(50)	50	NOT NULL	

Indexes

Key	Name	Key Columns	Unique
PK C	PK__Topic__1E140FB79323CECB	Crs_ID, Topic_ID	True

Foreign Keys

Name	Update	Delete	Columns
FK_Topic_Course	Cascade	Cascade	Crs_ID->[dbo].[Course].[Crs_ID]

SQL Script

```
CREATE TABLE [dbo].[Topic]
(
[Crs_ID] [int] NOT NULL,
[Topic_ID] [int] NOT NULL IDENTITY(1, 1),
[Topic_Name] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Topic] ADD CONSTRAINT [PK__Topic__1E140FB79323CECB] PRIMARY KEY CLUSTERED
([Crs_ID], [Topic_ID]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Topic] ADD CONSTRAINT [FK_Topic_Course] FOREIGN KEY ([Crs_ID]) REFERENCES
[dbo].[Course] ([Crs_ID]) ON DELETE CASCADE ON UPDATE CASCADE
```

```
GO
```

Uses

[dbo].[Course]

Used By

[dbo].[SP_SelectTopics]

[dbo].[SP_UpdateTopic]

Stored Procedures

Objects

Name
dbo.SP_DeleteCourse
dbo.SP_DeleteDepartment
dbo.SP_DeleteDeptCourse
dbo.SP_DeleteExam
dbo.SP_DeleteExamQues
dbo.SP_DeleteQuesAnswer
dbo.SP_DeleteQuesChoice
dbo.SP_DeleteQuestion
dbo.SP_DeleteStdCourse
dbo.SP_DeleteStdExam
dbo.SP_DeleteStudent
dbo.SP_ExamAnswers
dbo.SP_ExamCorrection
dbo.SP_ExamGeneration
dbo.SP_GetStdCourses
dbo.SP_InsertCourse
dbo.SP_InsertDepartment
dbo.SP_InsertDeptCourse
dbo.SP_InsertExam
dbo.SP_InsertExamQues
dbo.SP_InsertInstructor
dbo.SP_InsertQuesChoice
dbo.SP_InsertQuestion
dbo.SP_InsertStdCourse
dbo.SP_InsertStdExam
dbo.SP_InsertStudent
dbo.SP_logIN
dbo.SP_ReopSTDCourseGrade
dbo.SP_RepoExamQuestions
dbo.SP_RepoStudentAns
dbo.SP_SelectCourse
dbo.SP_SelectDepartment
dbo.SP_SelectDeptCourse
dbo.SP_SelectExam

dbo.SP_SelectExamQues
dbo.SP_SelectInstructor
dbo.SP_SelectQuesChoices
dbo.SP_SelectQuestion
dbo.SP_SelectStdCourse
dbo.SP_SelectStdExam
dbo.SP_SelectStudent
dbo.SP_SelectTopics
dbo.SP_UpdateCourse
dbo.SP_UpdateDepartment
dbo.SP_UpdateDeptCourse
dbo.SP_UpdateExam
dbo.SP_UpdateExamQues
dbo.SP_UpdateQuesChoice
dbo.SP_UpdateQuestion
dbo.SP_UpdateStdCourse
dbo.SP_UpdateStdExam
dbo.SP_UpdateStudent
dbo.SP_UpdateTopic
repo.SP_DeptStudentsInfo
repo.SP_ExamQuestions
repo.SP_InstructorInfo
repo.SP_SelectTopicsByCourse
repo.SP_STDCourseGrade
repo.SP_StudentExamAns

Project > (local)\SQLEXPRESS > User databases > SQL_Project >
Programmability > Stored Procedures > dbo.SP_DeleteCourse

[dbo].[SP_DeleteCourse]

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].[SP_DeleteCourse]
    @id int
as begin
     if @id is null or not exists (select 1 from Course where Crs_ID = @id)
        begin /*Check if Course ID is Null Value or Not Exist in Table*/
            select 'Course ID is Not Valid' as Result
            return;
        end
    else
        begin
            delete Course
            where Crs_ID = @id
            return;
        end
end
GO
```

Uses

[dbo].[Course]

[dbo].[SP_DeleteDepartment]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```sql
CREATE   proc [dbo].[SP_DeleteDepartment]
    @id int
as begin
     if @id is null or not exists (select 1 from Department where Dept_ID = @id)
        begin /*Check if Department ID is Null Value or Not Exist in Table*/
            select 'Department ID is Not Valid' as Result
            return;
        end
    else
        begin
            delete Department
            where Dept_ID = @id;
            return;
        end
end
GO
```

Uses

[dbo].[Department]

[dbo].[SP_DeleteDeptCourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@dept	int	4
@crs	int	4

SQL Script

```
CREATE   proc [dbo].[SP_DeleteDeptCourse]
@dept int = NULL, @crs int = NULL
as begin
     if @dept is null or @crs is null
        begin /*Check Department ID and Course ID have Null value*/
            select 'Insufficient Params.. Department ID and Course ID are Required' as Result
            return;
        end
    else
        begin /*Handle Any Code Now*/
            delete DeptCourse
            where Dept_ID = @dept and Crs_ID = @crs
            return;
        end
end
GO
```

Uses

[dbo].[DeptCourse]

Project > (local)\SQLEXPRESS > User databases > SQL_Project >
Programmability > Stored Procedures > dbo.SP_DeleteExam

[dbo].[SP_DeleteExam]

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].[SP_DeleteExam]
    @id int = NULL
as
    BEGIN TRY
        if @id is null
            select 'You Must Enter Exam ID'
        else if not exists(select Exam_ID from Exam where Exam_ID = @id)
            select 'Invalid Exam ID'
        else if exists(select Exam_ID from Exam_Question where Exam_ID = @id)
            select 'Can not delete the exam'
        else
            begin
                delete from Exam
                where Exam_ID = @id
            end
    END TRY
    BEGIN CATCH
        select 'Failed Exam Deletion, Try Again'
    END CATCH
GO
```

Uses

[dbo].[Exam]
[dbo].[Exam_Question]

[dbo].[SP_DeleteExamQues]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```sql
CREATE   PROCEDURE [dbo].[SP_DeleteExamQues] @exam_id int, @ques_id int
as
    BEGIN TRY
        if exists (select Exam_ID from Exam_Question where Exam_ID = @exam_id and Ques_ID =
@ques_id)
            begin
                if exists (select Ques_ID from Exam_Question where Ques_ID = @ques_id and Exam_ID
= @exam_id)
                    begin
                        delete from Exam_Question
                        where Exam_ID = @exam_id and Ques_ID = @ques_id
                    end
                else
                    select 'Invalid Question ID'
            end
        else
            select 'Invalid Exam ID'
    END TRY
    BEGIN CATCH
        select 'Failed Exam_Question Deletion, Try Again'
    END CATCH
GO
```

Uses

[dbo].[Exam_Question]

[dbo].[SP_DeleteQuesAnswer]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@ques_id	int	4
@answer	varchar	1

SQL Script

```
--Delete From Table Question
CREATE   PROCEDURE [dbo].[SP_DeleteQuesAnswer] @ques_id int, @answer varchar(1)
as
    BEGIN TRY
        if exists (select Ques_ID from QuesAnswers where Ques_ID = @ques_id and Ques_Answer =
@answer)
            begin
                if exists (select Ques_Answer from QuesAnswers where Ques_ID = @ques_id and
Ques_Answer = @answer)
                    begin
                        delete from QuesAnswers
                        where Ques_ID = @ques_id and Ques_Answer = @answer
                    end
                else
                    select 'Answer Not Exist'
            end
        else
            select 'Invalid Question ID'
    END TRY
    BEGIN CATCH
        select 'Failed Question Answer Deletion, Try Again'
    END CATCH
GO
```

[dbo].[SP_DeleteQuesChoice]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Ques	int	4
@Choice	varchar(250)	250

SQL Script

```
CREATE   PROC [dbo].[SP_DeleteQuesChoice]
    @Ques INT = NULL, @Choice VARCHAR(250) = NULL
AS BEGIN
    BEGIN TRY
        IF @Ques IS NULL OR @Choice IS NULL
            BEGIN
                SELECT 'Please Insert Question ID and it`s Choice'
                RETURN;
            END
        ELSE
            BEGIN
                DELETE FROM dbo.QuesChoices
                WHERE Ques_ID = @Ques AND Ques_Choices = @Choice;
            END
    END TRY
    BEGIN CATCH
        PRINT 'An error occurred: ';
    END CATCH
END
GO
```

Uses

[dbo].[QuesChoices]

Project > (local)\SQLEXPRESS > User databases > SQL_Project >
Programmability > Stored Procedures > dbo.SP_DeleteQuestion

[dbo].[SP_DeleteQuestion]

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].[SP_DeleteQuestion]
    @id int = NULL
as
    BEGIN TRY
        if @id is null
            select 'You  Must Enter Question ID'
        else if not exists(select Ques_ID from Question where Ques_ID = @id)
            select 'Invalid Question ID'
        else if exists(select Ques_ID from Exam_Question where Ques_ID = @id)
            select 'Can not delete the question'
        else
            begin
                delete from Question
                where Ques_ID = @id
            end
    END TRY
    BEGIN CATCH
        select 'Failed Question Deletion, Try Again'
    END CATCH
GO
```

Uses

[dbo].[Exam_Question]
[dbo].[Question]

[dbo].[SP_DeleteStdCourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@StdID	int	4
@CrsID	int	4

SQL Script

```
create   proc [dbo].[SP_DeleteStdCourse]
@StdID int=NULL , @CrsID int=NULL
as begin
    BEGIN TRY
     if @StdID is null or @CrsID is null
        begin
            select 'Student ID and Course ID are Required' as IDsError
            return;
        end
    else
        begin
            delete StdCourse
                where St_ID = @StdID and Crs_ID = @CrsID
            return;
        end
    END TRY
    BEGIN CATCH
    select 'Failed Student Course Deletion, Try Again'
    END CATCH
end
GO
```

Uses

[dbo].[StdCourse]

[dbo].[SP_DeleteStdExam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@St_SSN	int	4
@Exam_ID	int	4
@Ques_ID	int	4

SQL Script

```sql
CREATE   PROCEDURE [dbo].[SP_DeleteStdExam]
       @St_SSN INT = NULL, @Exam_ID INT = NULL, @Ques_ID INT = NULL
AS
BEGIN
    BEGIN TRY
      IF @St_SSN IS NULL AND @Exam_ID IS NULL AND @Ques_ID IS NULL
            BEGIN
                SELECT 'Insufficient Param... Student ID, Exam ID and Ques ID are Required'
                RETURN;
            END
      ELSE
        BEGIN
            INSERT INTO DBAudit (Entity) VALUES('StdExam')
            SELECT 'Delete Action Not Avilable for Student Exam'
        END
    END TRY
    BEGIN CATCH
        PRINT 'An error occurred: ';
    END CATCH;
END;
GO
```

Uses

[dbo].[DBAudit]

Project > (local)\SQLEXPRESS > User databases > SQL_Project >
Programmability > Stored Procedures > dbo.SP_DeleteStudent

[dbo].[SP_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   proc [dbo].[SP_DeleteStudent]
    @ID int
as begin
    BEGIN TRY
     if @ID is null or not exists (select St_ID from Student where St_ID = @ID)
        begin
            select 'Student ID is Not Exist' as IDError
            return;
        end
    else
        begin
            delete Student
                where St_ID = @ID;
            return;
        end
    END TRY
    BEGIN CATCH
    select 'Failed Student Deletion, Try Again'
    END CATCH
end
GO
```

Uses

[dbo].[Student]

[dbo].[SP_ExamAnswers]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@examID	int	4
@stID	int	4
@crsID	int	4
@a1	varchar	1
@a2	varchar	1
@a3	varchar	1
@a4	varchar	1
@a5	varchar	1
@a6	varchar	1
@a7	varchar	1
@a8	varchar	1
@a9	varchar	1
@a10	varchar	1

SQL Script

```
CREATE   proc [dbo].[SP_ExamAnswers]
    @examID int, @stID int, @crsID int,
    @a1 varchar(1), @a2 varchar(1), @a3 varchar(1), @a4 varchar(1), @a5 varchar(1),
    @a6 varchar(1), @a7 varchar(1), @a8 varchar(1), @a9 varchar(1), @a10 varchar(1)
as begin
      DECLARE @AnswerTable TABLE (rowNum int identity, St_ans VARCHAR(1));
      DECLARE @Ques TABLE (rowNum int identity, Ques_ID INT);

    INSERT INTO @AnswerTable (St_ans)
    VALUES (@a1), (@a2), (@a3), (@a4), (@a5), (@a6), (@a7), (@a8), (@a9), (@a10);

    INSERT INTO @Ques
    select Ques_ID
    from Exam_Question
    where Exam_ID = @examID
    INSERT INTO StdExam (St_SSN, Exam_ID, Ques_ID, St_Answer, Crs_ID)
    SELECT @stID, @examID, q.Ques_ID ,a.St_ans, @crsID
    FROM  @AnswerTable a, @Ques q
    where a.rowNum = q.rowNum
end;
GO
```

Uses

[dbo].[Exam_Question]
[dbo].[StdExam]

[dbo].[SP_ExamCorrection]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@examID	int	4
@stID	int	4

SQL Script

```sql
CREATE   proc [dbo].[SP_ExamCorrection]
    @examID int, @stID int
as begin
    declare @AnswerTable table (Ques_ID int, St_ans varchar(1));
    declare @res float = 0;
    declare @total float = 0;
    declare @crs varchar(25);
    select
        @crs = ques.Crs_ID,
        @total += ques.Ques_Grade,
        @res += case when lower(ques.Ques_Answer) = lower(exam.St_Answer) then ques.Ques_Grade
else 0 end
    from Question ques, StdExam exam
    where  @examID = exam.Exam_ID and @stID = exam.St_SSN and ques.Ques_ID = exam.Ques_ID

    update StdCourse
        set Grade = (@res / @total) * 100
        where Crs_ID = @crs and St_ID = @stID

    select  (@res/@total)*100
end
GO
```

Uses

[dbo].[Question]
[dbo].[StdCourse]
[dbo].[StdExam]

[dbo].[SP_ExamGeneration]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@crs	int	4
@mcq	int	4
@tf	int	4
@dur	int	4

SQL Script

```
CREATE   proc [dbo].[SP_ExamGeneration]
    @crs INT =null, @mcq INT =null, @tf INT =null, @dur int = 60
as begin

            exec SP_InsertExam @duration = @dur
            declare @id int = (select MAX(Exam_ID)from Exam)
            exec sp_InsertExamQues @examID = @id, @crsID = @crs, @num = @tf, @type = 'tf'
            exec sp_InsertExamQues @examID = @id, @crsID = @crs, @num = @mcq, @type = 'mcq'
            SELECT Exam_ID, exam.Ques_ID, Ques_Type,Ques_Body,Ques_Choices FROM dbo.Exam_Question
AS exam, dbo.Question AS ques,dbo.QuesChoices AS choice
            WHERE exam.Exam_ID = @id AND exam.Ques_ID = ques.Ques_ID AND ques.Ques_ID  =
choice.Ques_ID
end
GO
```

Uses

[dbo].[Exam]
[dbo].[Exam_Question]
[dbo].[QuesChoices]
[dbo].[Question]
[dbo].[SP_InsertExam]
[dbo].[SP_InsertExamQues]

Project > (local)\SQLEXPRESS > User databases > SQL_Project >
Programmability > Stored Procedures > dbo.SP_GetStdCourses

[dbo].[SP_GetStdCourses]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```sql
CREATE procedure [dbo].[SP_GetStdCourses]
    @stID int
as
begin
    select crs.Crs_ID, crs.Crs_Name
    from Course crs, StdCourse scrs
    where scrs.St_ID = @stID and crs.Crs_ID = scrs.Crs_ID
    and scrs.Grade is null
end
GO
```

Uses

[dbo].[Course]
[dbo].[StdCourse]

[dbo].[SP_InsertCourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@name	varchar(15)	15
@ins	int	4
@duration	int	4

SQL Script

```sql
CREATE   proc [dbo].[SP_InsertCourse]
    @name varchar(15), @ins int, @duration int=25
as begin
    if @name is null or @ins is null /*Check if Course Name or Instructor are Null Value or Not*/
        begin
            select 'Insufficient Params.. Course Name and Instructor ID are Required' as Result
            return;
        end
    else if exists (select 1 from Course where LOWER(Crs_Name) = LOWER(@name) )
        begin /*Check if Course Name Existance*/
            select 'Course is Added Before' as Result
            return;
        end
    else if(LEN(@name) < 2) /*Check if Length of Department Name at least 2 char*/
        begin
            select 'Length of Course Name Should be more than 1 char'
            return;
        end
    else if not exists (select 1 from Instructor where Ins_SSN = @ins)
        begin /*Check if Instructor ID Not Exist in Instructor Table*/
            select 'The Instrucor ID is Not Valid' as Result
            return;
        end
    else
        insert into Course
        (Crs_Name, Crs_Duration, Ins_ID)
        values(@name, @duration, @ins);
end
```

```
GO
```

Uses

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

[dbo].[SP_InsertDepartment]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@name	varchar(15)	15
@ins	int	4

SQL Script

```
CREATE   proc [dbo].[SP_InsertDepartment]
    @name varchar(15), @ins int = null
as begin
     if @name is null /*Check if Department Name is Null Value or Not*/
        begin
            select 'Department Name is Required' as Result
            return;
        end
    else if exists (select 1 from Department where lower(Dept_Name) = lower(@name) )
        begin /*Check if Department Name Existance*/
            select 'Department is Added Before' as Result
            return;
        end
    else if @ins is not null and not exists (select 1 from Instructor where Ins_SSN = @ins)
        begin /*Check if Instructor ID (Manager) Not Exist in Instructor Table*/
            select 'The Instrucor ID is Not Valid' as Result
            return;
        end
    else if(LEN(@name) < 2)/*Check if Length of Department Name at least 2 char*/
        begin
            select 'Length of Department Name Should be more than 1 char' as Result
            return;
        end
    else
        insert into Department
        (Dept_Name, Manager_ID)
        values(@name, @ins);
end
GO
```

Uses

[dbo].[Department]

[dbo].[Instructor]

[dbo].[SP_InsertDeptCourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@dept	varchar(15)	15
@crs	varchar(25)	25

SQL Script

```sql
create   proc [dbo].[SP_InsertDeptCourse]
@dept varchar(15), @crs varchar(25)
as begin
     if @dept is null or @crs is null
        begin /*Check Department ID and Course ID have Null value*/
            select 'Insufficient Params.. Department ID and Course ID are Required' as Result
            return;
        end
    else if    not exists (select 1 from Department where Dept_ID = @dept)
        begin /*Check Course Existance in Course Table*/
            select 'Department ID is Not Valid' as Result
            return;
        end
    else if not exists (select 1 from Course where Crs_ID = @crs)
        begin /*Check Course Existance in Course Table*/
            select 'Course ID is Not Valid' as Result
            return;
        end
        else if exists (select 1 from DeptCourse where Crs_ID = @crs and Dept_ID = @dept)
        begin /*Check if Course was assigned to Department*/
            select 'Course is Assigned Before to This Department' as Result
            return;
        end
        else
            insert into DeptCourse
            (Dept_ID, Crs_ID)
            values(@dept, @crs);
end
GO
```

Uses

[dbo].[Course]
[dbo].[Department]
[dbo].[DeptCourse]

[dbo].[SP_InsertExam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE   PROCEDURE [dbo].[SP_InsertExam]
    @duration int = 60
as
    BEGIN TRY
        INSERT INTO Exam
        VALUES(@duration)
    END TRY
    BEGIN CATCH
        select 'Failed Exam Insertion, Try Again'
    END CATCH
GO
```

Uses

[dbo].[Exam]

Used By

[dbo].[SP_ExamGeneration]

[dbo].[SP_InsertExamQues]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@examID	int	4
@crsID	int	4
@num	int	4
@type	varchar(3)	3

SQL Script

```
CREATE   PROCEDURE [dbo].[SP_InsertExamQues]
    @examID int, @crsID int, @num int, @type varchar(3)
as BEGIN
    BEGIN TRY
        if(@num < 1)
            return
        insert into Exam_Question(Exam_ID, Ques_ID)
        select top (@num) @examID as [ExamID], Ques_ID
        from Question
        where Ques_Type = @type and Crs_ID = @crsID
        order by newid();
    END TRY
    BEGIN CATCH
        select 'Failed Exam_Question Insertion, Try Again'
    END CATCH
END
GO
```

Uses

[dbo].[Exam_Question]
[dbo].[Question]

Used By

[dbo].[SP_ExamGeneration]

Project > (local)\SQLEXPRESS > User databases > SQL_Project >
Programmability > Stored Procedures > dbo.SP_InsertInstructor

[dbo].[SP_InsertInstructor]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Fname	varchar(50)	50
@Lname	varchar(50)	50
@Phone	varchar(11)	11
@Salary	decimal(18,0)	9
@Ins_Address	varchar(50)	50
@Ins_Dept_ID	int	4

SQL Script

```
create   proc [dbo].[SP_InsertInstructor]  @Fname varchar(50) = null, @Lname varchar(50) = null,
@Phone varchar(11) = null,
    @Salary decimal = null,
    @Ins_Address varchar(50) = null,
    @Ins_Dept_ID int = null
as
    if @Fname is null
        begin
            select 'First Name shouln''t be null'
        end
    else if @Lname is null
        begin
            select 'Last Name shouln''t be null'
        end
    else if (len(@Phone)!=11)
        begin
            select 'Enter a valid Phone Number'
        end
    else if exists(select Dept_ID from Department where Dept_ID=@Ins_Dept_ID)
        begin
            insert into Instructor(Ins_Fname,Ins_Lname,Ins_Phone,Ins_Salary,Ins_Address,Dept_ID)
            values(@Fname,@Lname,@Phone,@Salary,@Ins_Address,@Ins_Dept_ID)
        end
    else
```

```
        begin
            select 'Department ID doesn''t exist'
    end
GO
```

Uses

[dbo].[Department]

[dbo].[Instructor]

[dbo].[SP_InsertQuesChoice]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Ques	int	4
@Choice	varchar(250)	250

SQL Script

```sql
CREATE   PROCEDURE [dbo].[SP_InsertQuesChoice]
    @Ques INT = null,
    @Choice VARCHAR(250) = null
AS
BEGIN
    BEGIN TRY
        IF @Ques IS NULL OR @Choice IS NULL
            BEGIN
                SELECT 'Please Insert Question ID and it`s Choice'
                RETURN;
            END
        ELSE
            BEGIN
                INSERT INTO [dbo].[QuesChoices] ([Ques_ID], [Ques_Choices])
                VALUES (@Ques, @Choice);
            END
    END TRY
    BEGIN CATCH
        PRINT 'An error occurred: ';
    END CATCH;
END;
GO
```

Uses

[dbo].[QuesChoices]

Project > (local)\SQLEXPRESS > User databases > SQL_Project > Programmability > Stored Procedures > dbo.SP_InsertQuestion

[dbo].[SP_InsertQuestion]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@body	text	16
@type	varchar(100)	100
@answer	varchar	1
@grade	int	4
@crs_id	int	4

SQL Script

```
CREATE   PROCEDURE [dbo].[SP_InsertQuestion]
    @body text = NULL, @type varchar(100) = NULL, @answer varchar(1) = NULL,
    @grade int = NULL, @crs_id int = NULL
as
    BEGIN TRY
        if @body is null
            select 'You  Must Enter Question Body'
        else if @type is null
            select 'You  Must Enter Question Type'
        else if @grade is null
            select 'You  Must Enter Question Grade'
        else if @answer is null
            select 'You  Must Enter Question Answer'
        else if @crs_id is null
            select 'You  Must Enter Course ID'
        else if exists (select Crs_ID from Course where Crs_id = @crs_id)
            begin
                insert into Question
                values(@type, @body, @answer, @grade, @crs_id)
            end
        else
            select 'Invalid Course ID'
    END TRY
    BEGIN CATCH
        select 'Failed Question Insertion, Try Again'
    END CATCH
GO
```

Uses

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

Project > (local)\SQLEXPRESS > User databases > SQL_Project > Programmability > Stored Procedures > dbo.SP_InsertStdCourse

[dbo].[SP_InsertStdCourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@StdID	int	4
@CrsID	int	4

SQL Script

```
CREATE   proc [dbo].[SP_InsertStdCourse]
@StdID int, @CrsID int
as begin
    BEGIN TRY
     if @StdID is null or @CrsID is null
        begin
            select 'Insufficient Params.. Student ID and Course ID are Required' as Result
            return;
        end
    else if not exists (select ST_ID from Student where St_ID = @StdID)
        begin
            select 'Student ID is not valid' as StudentIDError
            return;
        end
        else if not exists (select Crs_ID from Course where Crs_ID = @CrsID)
        begin
            select 'Course ID is not valid' as CourseIDError
            return;
        end
        else if exists (select * from StdCourse where St_ID = @StdID and Crs_ID = @CrsID)
        begin
            select 'Course is Assigned Before to This Student' as ComplexKeyError
            return;
        end
        else
            insert into StdCourse
            (ST_ID ,Crs_ID)
            values(@StdID, @CrsID);
    END TRY
    BEGIN CATCH
    select 'Failed Student Course Insertion, Try Again'
    END CATCH
end
GO
```

Uses

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

[dbo].[SP_InsertStdExam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@St_SSN	int	4
@Exam_ID	int	4
@Ques_ID	int	4
@St_Answer	varchar(50)	50

SQL Script

```
CREATE   PROCEDURE [dbo].[SP_InsertStdExam]
    @St_SSN INT = NULL, @Exam_ID INT = NULL, @Ques_ID INT = NULL, @St_Answer VARCHAR(50) = NULL
AS
BEGIN
    BEGIN TRY
        IF @St_SSN IS NULL AND @Exam_ID IS NULL AND @Ques_ID IS NULL AND @St_Answer IS NULL
            BEGIN
                SELECT 'Insufficient Param... Student ID, Exam ID, Ques ID and Student Answer are
Required'
                RETURN;
            END
        ELSE
            BEGIN
                INSERT INTO [dbo].[StdExam] ([St_SSN], [Exam_ID], [Ques_ID], [St_Answer])
                VALUES (@St_SSN, @Exam_ID, @Ques_ID, @St_Answer);
            END
    END TRY
    BEGIN CATCH
        PRINT 'An error occurred: ';
    END CATCH;
END;

GO
```

Uses

[dbo].[StdExam]

[dbo].[SP_InsertStudent]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@FName	varchar(15)	15
@LName	varchar(15)	15
@Phone	varchar(11)	11
@Address	varchar(20)	20
@BirthDate	date	3
@DeptID	int	4

SQL Script

```
create   proc [dbo].[SP_InsertStudent]
     @FName varchar(15)=NULL, @LName varchar(15)=NULL,@Phone varchar(11)=NULL, @Address
varchar(20)=NULL,@BirthDate Date=NULL, @DeptID int=NULL
as begin
    BEGIN TRY
     if @FName is null
        begin
            select 'Student first name is required' as FNameError
            return;
        end
    else if @LName is null
        begin
            select 'Student last name is required' as LNameError
            return;
        end
    else if(LEN(@Phone) !=11) or NOT ISNUMERIC(@Phone) = 1
        begin
            select 'Length of phone number must be 11 number' as PhoneNumberError
            return;
        end
    else if @DeptID is not NULL and not exists (select Dept_ID from Department where Dept_ID =
@DeptID)
        begin
            select 'Department ID is not valid' as DeptIDError
            return;
        end
    else
        insert into Student
        (St_Fname, St_Lname,St_Phone,St_Address,St_BD,Dept_ID)
        values(@FName, @LName,@Phone,@Address,@BirthDate,@DeptID);
    END TRY
    BEGIN CATCH
    select 'Failed Student Insertion, Try Again'
    END CATCH
end
GO
```

Uses

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

Project > (local)\SQLEXPRESS > User databases > SQL_Project >
Programmability > Stored Procedures > dbo.SP_logIN

[dbo].[SP_logIN]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@SSN	varchar(14)	14

SQL Script

```sql
create   proc [dbo].[SP_logIN]
@SSN varchar(14)
as
begin
    select St_id, St_Fname from Student where St_SSN=@SSN
end
GO
```

Uses

[dbo].[Student]

Project > (local)\SQLEXPRESS > User databases > SQL_Project > Programmability > Stored Procedures > dbo.SP_ReopSTDCourseGrade

[dbo].[SP_ReopSTDCourseGrade]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE   procedure [dbo].[SP_ReopSTDCourseGrade]
    @stdID int = NULL
as
        begin
            select concat (st.St_Fname,' ',st.St_Lname) StudentName, crs.Crs_Name, StCrs.Grade,crs.Crs_ID
            from Student st, Course crs, StdCourse StCrs
            where StCrs.St_ID=st.St_ID
            and StCrs.Crs_ID=crs.Crs_ID
            and st.St_ID=@stdID
            and StCrs.Grade is not null
        end
GO
```

Uses

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

[dbo].[SP_RepoExamQuestions]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE   PROCEDURE [dbo].[SP_RepoExamQuestions]
    @exam_no int = NULL
as
    BEGIN TRY
        if @exam_no is null
            select 'You Must Enter Exam Number'
        else if not exists (select Exam_ID from Exam_Question where Exam_ID = @exam_no)
            select 'Invalid Exam Number'
        else
            begin
                select Ques_Body, Ques_Choices
                from Question q, QuesChoices qc, Exam_Question e
                where q.Ques_ID = qc.Ques_ID and e.Ques_ID = q.Ques_ID
            end
    END TRY
    BEGIN CATCH
        select 'Try Again'
    END CATCH
GO
```

Uses

[dbo].[Exam_Question]
[dbo].[QuesChoices]
[dbo].[Question]

[dbo].[SP_RepoStudentAns]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@CrsID	int	4
@stID	int	4

SQL Script

```
CREATE   proc [dbo].[SP_RepoStudentAns]
    @CrsID int, @stID int
as begin
    select
        ques.Ques_Body as [Question],
        ques.Ques_Answer as [Answer],
        exam.St_Answer as [Std Answer],
        ques.Ques_ID as [QuesID]
    from Question ques, StdExam exam
    where  @CrsID=exam.crs_ID and @stID = exam.St_SSN and ques.Ques_ID = exam.Ques_ID

end
GO
```

Uses

[dbo].[Question]
[dbo].[StdExam]

Project > (local)\SQLEXPRESS > User databases > SQL_Project >
Programmability > Stored Procedures > dbo.SP_SelectCourse

[dbo].[SP_SelectCourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```sql
create   proc [dbo].[SP_SelectCourse]
    @id int = null
as begin
    if @id is null
        begin
            select * from Course
        end
    else if @id is not null and exists(select 1 from Course where Crs_ID = @id)
        begin
            select * from Course where Crs_ID = @id
        end
    else
        begin
            select 'Course ID Not Valid'
        end
end
GO
```

Uses

[dbo].[Course]

[dbo].[SP_SelectDepartment]

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].[SP_SelectDepartment]
    @id int = null
as begin
    if @id is null
        begin
            select * from Department
        end
    else if @id is not null and exists(select 1 from Department where Dept_ID = @id)
        begin
            select * from Department where Dept_ID = @id
        end
    else
        begin
            select 'Depratment ID Not Valid'
        end

end
GO
```

Uses

[dbo].[Department]

[dbo].[SP_SelectDeptCourse]

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].[SP_SelectDeptCourse]
    @id int = null
as begin
    if @id is null
        begin
            select
                dept.Dept_Name as[Department],
                crs.Crs_Name as[Course]
            from DeptCourse dept_crs, Department dept, Course crs
        end
    else if @id is not null and exists(select 1 from Course where Crs_ID = @id)
        begin
            select * from Course where Crs_ID = @id
        end
    else
        begin
            select 'Course ID Not Valid'
        end
select * from DeptCourse
end
GO
```

Uses

[dbo].[Course]
[dbo].[Department]
[dbo].[DeptCourse]

[dbo].[SP_SelectExam]

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].[SP_SelectExam]
@id int = null
as
    BEGIN TRY
        if @id is null
            begin
                select *
                from Exam
            end
        else if not exists(select Exam_ID from Exam where Exam_ID = @id)
            select 'Invalid Exam ID'
        else
            begin
                select *
                from Exam
                where Exam_ID = @id
            end
    END TRY
    BEGIN CATCH
        select 'Failed Exam Selection, Try Again'
    END CATCH
GO
```

Uses

[dbo].[Exam]

Project > (local)\SQLEXPRESS > User databases > SQL_Project >
Programmability > Stored Procedures > dbo.SP_SelectExamQues

[dbo].[SP_SelectExamQues]

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].[SP_SelectExamQues]
@exam_id int = NULL
as
    BEGIN TRY
           SELECT 'Start ::> SP_SelectExamQues'
        if @exam_id is null
            begin
                select *
                from Exam_Question
            end
        else if not exists(select Exam_ID from Exam where Exam_ID = @exam_id)
            select 'Invalid Exam ID'
        else
            begin
                select *
                from Exam_Question
                where Exam_ID = @exam_id
            end
    END TRY
    BEGIN CATCH
        select 'Failed Exam Questions Selection, Try Again'
    END CATCH

GO
```

Uses

[dbo].[Exam]
[dbo].[Exam_Question]

[dbo].[SP_SelectInstructor]

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].[SP_SelectInstructor] @id int = null
as
    if @id is null
        begin
            select * from Instructor
        end
    else if not exists (select Ins_SSN from Instructor where Ins_SSN = @id)
        begin
            select 'Topic id doesn''t exist'
        end
    else
        begin
            select * from Instructor where Ins_SSN = @id
        end
GO
```

Uses

[dbo].[Instructor]

Project > (local)\SQLEXPRESS > User databases > SQL_Project >
Programmability > Stored Procedures > dbo.SP_SelectQuesChoices

[dbo].[SP_SelectQuesChoices]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE   PROCEDURE [dbo].[SP_SelectQuesChoices]
    @Ques INT= NULL
AS
BEGIN
    SELECT [Ques_ID], [Ques_Choices]
    FROM [dbo].[QuesChoices]
    WHERE [Ques_ID] = @Ques;
END;
GO
```

Uses

[dbo].[QuesChoices]

Project > (local)\SQLEXPRESS > User databases > SQL_Project >
Programmability > Stored Procedures > dbo.SP_SelectQuestion

[dbo].[SP_SelectQuestion]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```sql
CREATE   PROCEDURE [dbo].[SP_SelectQuestion]
    @id int = NULL
as
    BEGIN TRY
        if @id is null
            begin
                select *
                from Question
            end
        else if not exists(select Ques_ID from Question where Ques_ID = @id)
            select 'Invalid Question ID'
        else
            begin
                select *
                from Question
                where Ques_ID = @id
            end
    END TRY
    BEGIN CATCH
        select 'Failed Question Selection, Try Again'
    END CATCH

GO
```

Uses

[dbo].[Question]

[dbo].[SP_SelectStdCourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@stdID	int	4
@CrsID	int	4

SQL Script

```
create   proc [dbo].[SP_SelectStdCourse]
    @stdID int=null,@CrsID int =null
as
begin
    BEGIN TRY
    if @StdID is null or @CrsID is null
        begin
            select * from StdCourse
            return;
        end
    else if exists(select * from StdCourse where St_ID = @stdID and Crs_ID=@CrsID)
        begin
            select * from StdCourse where  St_ID = @stdID and Crs_ID=@CrsID
        end
    else
        begin
            select 'Student ID and Course ID are Invalid'
        end
    END TRY
    BEGIN CATCH
    select 'Failed Student Course Selection, Try Again'
    END CATCH
end
GO
```

Uses

[dbo].[StdCourse]

[dbo].[SP_SelectStdExam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@St_SSN	int	4
@Exam_ID	int	4

SQL Script

```
CREATE   PROCEDURE [dbo].[SP_SelectStdExam]
    @St_SSN INT = NULL, @Exam_ID INT = NULL
AS BEGIN
    IF @St_SSN IS NULL AND @Exam_ID IS NULL
            BEGIN
                SELECT 'Insufficient Param... Student ID and Exam ID are Required'
                RETURN;
            END
    ELSE
        BEGIN
            SELECT [St_SSN], [Exam_ID], [Ques_ID], [St_Answer]
            FROM [dbo].[StdExam]
            WHERE [St_SSN] = @St_SSN AND [Exam_ID] = @Exam_ID;
        END
END;
GO
```

Uses

[dbo].[StdExam]

[dbo].[SP_SelectStudent]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```sql
create   proc [dbo].[SP_SelectStudent]
    @id INT =NULL
as
begin
    BEGIN TRY
        if @id is null
        begin
            select * from Student
        end
    else if exists(select St_ID from Student where St_ID = @id)
        begin
            select * from Student where St_ID = @id
        end
    else
        begin
            select 'Student ID is Not Valid'
        end
    END TRY
    BEGIN CATCH
    select 'Failed Student Selection, Try Again'
    END CATCH
end
GO
```

Uses

[dbo].[Student]

Project > (local)\SQLEXPRESS > User databases > SQL_Project >
Programmability > Stored Procedures > dbo.SP_SelectTopics

[dbo].[SP_SelectTopics]

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].[SP_SelectTopics]
@id int = null
as
    if @id is null
        begin
            select * from Topic
        end
    else if not exists (select Topic_ID from Topic where Crs_ID = @id)
        BEGIN
            SELECT 'Course ID not valid'
        END
    ELSE
        BEGIN
            SELECT * FROM Topic WHERE Crs_ID = @id
        END
GO
```

Uses

[dbo].[Topic]

Used By

[repo].[SP_SelectTopicsByCourse]

Project > (local)\SQLEXPRESS > User databases > SQL_Project > Programmability > Stored Procedures > dbo.SP_UpdateCourse

[dbo].[SP_UpdateCourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@id	int	4
@name	varchar(15)	15
@ins	int	4
@duration	int	4

SQL Script

```
CREATE   proc [dbo].[SP_UpdateCourse]
    @id int, @name varchar(15) = NULL, @ins int = NULL, @duration int = NULL
as begin
     if @id is null or not exists (select 1 from Course where Crs_ID = @id)
        begin /*Check if Course ID is Null Value or Not Exist in Table*/
            select 'Course ID is Not Valid' as Result
            return;
        end
    if @name is null and @ins is null and @duration is null
        begin /*Check if Course Name && Instructor ID && Duration are Null Value or Not*/
            select 'No Updates Occured' as Result
            return;
        end
    else
        begin
            update Course
            set
                Ins_ID = isnull(@ins, Ins_ID),
                Crs_Name = isnull(@name, Crs_Name),
                Crs_Duration = isnull(@duration, Crs_Duration)
            where Crs_ID = @id
            return;
        end
end
GO
```

Uses

[dbo].[Course]

[dbo].[SP_UpdateDepartment]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@id	int	4
@name	varchar(15)	15
@ins	int	4

SQL Script

```
CREATE   proc [dbo].[SP_UpdateDepartment]
    @id int = null, @name varchar(15) = NULL, @ins int = NULL
as begin
     if @id is null or not exists (select 1 from Department where Dept_ID = @id)
        begin /*Check if Department ID is Null Value or Not Exist in Table*/
            select 'Department ID is Not Valid' as Result
            return;
        end
    else if @name is null and @ins is null
        begin /*Check if Department Name and Instrucor ID both are Null Value*/
            select 'No Updates Occured' as Result
            return;
        end
    else
        begin
            update Department
                set
                    Dept_Name = isnull(@name, Dept_Name),
                    Manager_ID = isnull(@ins, Manager_ID)
                where Dept_ID = @id;
            return;
        end
end
GO
```

Uses

[dbo].[Department]

[dbo].[SP_UpdateDeptCourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@dept	int	4
@crs	int	4

SQL Script

```
create   proc [dbo].[SP_UpdateDeptCourse]
@dept int = null, @crs int = null
as begin
     insert into DBAudit(Entity, Action) values ('DeptCourse','Update')
    select 'Update Not Avalibale for this Table'
end
GO
```

Uses

[dbo].[DBAudit]

[dbo].[SP_UpdateExam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE   PROCEDURE [dbo].[SP_UpdateExam]
    @id int = NULL, @duration int = NULL
as
    BEGIN TRY
        if @id is null
            select 'You Must Enter Exam ID'
        else if not exists(select Exam_ID from Exam where Exam_ID = @id)
            select 'Invalid Exam ID'
        else if @duration is null
            select 'You Must Enter Exam Duration'
        else
            begin
                Update Exam
                set Exam_Duration = @duration
                where Exam_ID = @id
            end
    END TRY
    BEGIN CATCH
        select 'Failed Exam Update, Try Again'
    END CATCH
GO
```

Uses

[dbo].[Exam]

Project > (local)\SQLEXPRESS > User databases > SQL_Project >
Programmability > Stored Procedures > dbo.SP_UpdateExamQues

[dbo].[SP_UpdateExamQues]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE   PROCEDURE [dbo].[SP_UpdateExamQues]
    @exam_id int = null, @ques_id int = null
as begin
    insert into DBAudit(Entity, Action) values ('Exam_Question','Update')
    select 'Update Not Avalibale for this Table'
end

GO
```

Uses

[dbo].[DBAudit]

[dbo].[SP_UpdateQuesChoice]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Ques	int	4
@Choice	varchar(250)	250
@NewChoice	varchar(250)	250

SQL Script

```
CREATE   PROCEDURE [dbo].[SP_UpdateQuesChoice]
    @Ques INT = null,
    @Choice VARCHAR(250) = null,
    @NewChoice VARCHAR(250) = null
AS
BEGIN
    BEGIN TRY
        IF @Ques IS NULL AND @Choice IS NULL
            BEGIN
                SELECT 'Please Insert Question ID and it`s Choice that you want to change'
                RETURN;
            END
        IF @NewChoice IS NULL
            BEGIN
                SELECT 'No Update Occured... Please, Insert New Choice'
                RETURN;
            END
        ELSE
            BEGIN
                UPDATE [dbo].[QuesChoices]
                SET [Ques_Choices] = @NewChoice
                WHERE [Ques_ID] = @Ques AND Ques_Choices = @Choice;
            END

    END TRY
    BEGIN CATCH
        PRINT 'An error occurred: ';
    END CATCH;
```

```
END;
GO
```

Uses

[dbo].[QuesChoices]

[dbo].[SP_UpdateQuestion]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@id	int	4
@body	text	16
@type	varchar(100)	100
@answer	varchar	1
@grade	int	4
@crs_id	int	4

SQL Script

```
CREATE   PROCEDURE [dbo].[SP_UpdateQuestion]
    @id int = NULL, @body text = NULL, @type varchar(100)  = NULL,
    @answer varchar(1)  = NULL, @grade int = NULL, @crs_id int = NULL
as
    BEGIN TRY
        if @id is null
            select 'You  Must Enter Question ID'
         else if not exists(select Ques_ID from Question where Ques_ID = @id)
            select 'Invalid Question ID'
        else if @body is null and @type is null and @grade is null and @answer is null and
@crs_id is null
            select 'No value will be updated'

        else if @crs_id is not null and not exists(select Crs_ID from Course where Crs_ID =
@crs_id)
            select 'Invalid Course ID'
        else
            begin
                Update Question
                set
                Ques_Body = isnull(@body, Ques_Body),
                Ques_Type = isnull(@type, Ques_Type),
                Ques_Answer = isnull(@answer, Ques_Answer),
                Ques_Grade = isnull(@grade, Ques_Grade),
                Crs_ID = isnull(@crs_id, Crs_ID)
                where Ques_ID = @id
            end
    END TRY
    BEGIN CATCH
        select 'Failed Question Update, Try Again'
    END CATCH
GO
```

Uses

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

[dbo].[SP_UpdateStdCourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@StdID	int	4
@CrsID	int	4
@newCrs	int	4

SQL Script

```
create   proc [dbo].[SP_UpdateStdCourse]
@StdID int = NULL, @CrsID int = NULL, @newCrs int = NULL
as
    begin
    BEGIN TRY
        if @StdID is null or @CrsID is null and @newCrs is null
        begin
            select 'Student ID, Course ID and New Course ID are Required'
        end
        else if exists (select St_ID, Crs_ID from StdCourse where St_ID = @StdID and Crs_ID=
@CrsID )
            begin
            if exists (select @newCrs from Course where Crs_ID = @newCrs)
            begin
                update StdCourse
                set St_ID = @StdID,
                    Crs_ID = @newCrs
                where St_ID =@StdID
                and    Crs_ID = @CrsID
            end
            else
                select 'Invalid Course ID'
            end
        else
            select 'Invalid Student ID'
    END TRY
    BEGIN CATCH
        select 'Failed Student Course Update, Try Again'
    END CATCH
end
GO
```

Uses

[dbo].[Course]

[dbo].[StdCourse]

[dbo].[SP_UpdateStdExam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@St_SSN	int	4
@Exam_ID	int	4
@Ques_ID	int	4
@St_Answer	varchar(50)	50

SQL Script

```
CREATE   PROCEDURE [dbo].[SP_UpdateStdExam]
    @St_SSN INT = NULL, @Exam_ID INT = NULL, @Ques_ID INT = NULL, @St_Answer VARCHAR(50) = NULL
AS
BEGIN
    BEGIN TRY
        IF @St_SSN IS NULL AND @Exam_ID IS NULL AND @Ques_ID IS NULL AND @St_Answer IS NULL
            BEGIN
                SELECT 'Insufficient Param... Student ID, Exam ID, Ques ID and Student Answer are Required'
                RETURN;
            END
        ELSE
            BEGIN
               INSERT INTO DBAudit (Entity, Action) VALUES('StdExam','Update')
               SELECT 'Student can`t Change his Answers'
            END
    END TRY
    BEGIN CATCH

        PRINT 'An error occurred: ';
    END CATCH;
END;
GO
```

Uses

[dbo].[DBAudit]

Project > (local)\SQLEXPRESS > User databases > SQL_Project > Programmability > Stored Procedures > dbo.SP_UpdateStudent

[dbo].[SP_UpdateStudent]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@ID	int	4
@FName	varchar(15)	15
@LName	varchar(15)	15
@Phone	varchar(11)	11
@Address	varchar(20)	20
@BirthDate	date	3
@DeptID	int	4

SQL Script

```
create   proc [dbo].[SP_UpdateStudent]
     @ID int=NULL, @FName varchar(15)=NULL, @LName varchar(15)=NULL,
     @Phone varchar(11)=NULL, @Address varchar(20)=NULL,@BirthDate Date=NULL, @DeptID int=NULL
as begin
    BEGIN TRY
    if @ID is null or not exists (select St_ID from Student where St_ID = @ID)
        begin
            select 'Student ID is not Exist' as IDError
            return;
        end
    else if @Phone is not null and ((LEN(@Phone) !=11) or NOT ISNUMERIC(@Phone) = 1)
        begin
            select 'Length of phone number must be 11 number' as PhoneNumberError
            return;
        end
    else if @DeptID is not NULL and not exists (select Dept_ID from Department where Dept_ID =
@DeptID)
        begin
            select 'Department ID is not valid' as DeptIDError
            return;
        end
    else
        begin
```

```
            update Student
                set
                    St_Fname=isnull(@FName,St_Fname),
                    St_Lname=isnull(@LName,St_Lname),
                    St_Phone=isnull(@Phone,St_Phone),
                    St_Address=isnull(@Address,St_Address),
                    St_BD=isnull(@BirthDate,St_BD),
                    Dept_ID=isnull(@DeptID,Dept_ID)
                where St_ID = @ID;
            return;
        end
    END TRY
    BEGIN CATCH
    select 'Failed Student Update, Try Again'
    END CATCH
end
GO
```

Uses

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

Project > (local)\SQLEXPRESS > User databases > SQL_Project >
Programmability > Stored Procedures > dbo.SP_UpdateTopic

[dbo].[SP_UpdateTopic]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Topic_name	varchar(50)	50
@Crs_ID	int	4
@newTopic	varchar(50)	50

SQL Script

```sql
create   proc [dbo].[SP_UpdateTopic]
@Topic_name varchar(50) = null, @Crs_ID int = null, @newTopic varchar(50) = null
as
    if @Topic_name is null
        begin
            select 'Topic name shouldn''t be null'
        end
    else if exists(select Crs_ID from Course where Crs_ID = @Crs_ID)
        begin
            update Topic
            set
            Topic_Name = @newTopic,
            Crs_ID = @Crs_ID
            where Crs_ID = @Crs_ID and Topic_Name = @Topic_Name
        end
    else
        begin
            select 'Course ID doesn''t exist'
        end
GO
```

Uses

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

[repo].[SP_DeptStudentsInfo]

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 [repo].[SP_DeptStudentsInfo]
    @dept_id int = NULL
AS BEGIN
    BEGIN TRY
        if @dept_id is null
            SELECT
                dept.Dept_Name AS [Department],
                CONCAT(std.St_Fname, ' ', std.St_Lname) AS [Student Name],
                std.St_Phone AS [Phone],
                std.St_Address AS [Address],
                DATEDIFF(yy, std.St_BD, GETDATE()) AS age
            FROM dbo.Student AS std, dbo.Department AS dept
            WHERE std.Dept_ID = dept.Dept_ID
            ORDER BY dept.Dept_ID

        else
            begin
                SELECT
                dept.Dept_Name AS [Department],
                CONCAT(std.St_Fname, ' ', std.St_Lname) AS [Student Name],
                std.St_Phone AS [Phone],
                std.St_Address AS [Address],
                DATEDIFF(yy, std.St_BD, GETDATE()) AS age
            FROM dbo.Student AS std, dbo.Department AS dept
            WHERE std.Dept_ID = dept.Dept_ID AND dept.Dept_ID = @dept_id
            ORDER BY dept.Dept_ID
            end
    END TRY
    BEGIN CATCH
        select 'Try Again'
    END CATCH
```

```
END
GO
```

Uses

[dbo].[Department]

[dbo].[Student]

repo

[repo].[SP_ExamQuestions]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```sql
CREATE   PROC [repo].[SP_ExamQuestions]
    @exam_no int
AS BEGIN
    BEGIN TRY
        select
            Ques_Body [Body],
            Ques_Choices [Choice],
            crs.Crs_Name [Course]
        from Question q, QuesChoices qc, Exam_Question e, Course crs
        where q.Ques_ID = qc.Ques_ID and e.Ques_ID = q.Ques_ID AND e.Exam_ID = @exam_no AND
crs.Crs_ID = q.Crs_ID
    END TRY
    BEGIN CATCH
        select 'Try Again'
    END CATCH
END
GO
```

Uses

[dbo].[Course]
[dbo].[Exam_Question]
[dbo].[QuesChoices]
[dbo].[Question]
repo

[repo].[SP_InstructorInfo]

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 [repo].[SP_InstructorInfo]
    @id int
as begin
    if @id is null
        begin
            select
                crs.Crs_Name [Course Name],
                count(*) [Students]
            from Course crs, Instructor ins, dbo.StdCourse std
            where crs.Ins_ID = ins.Ins_SSN and std.Crs_ID = crs.Crs_ID
            group by crs.Crs_Name
        end
    else
        begin
            select
                crs.Crs_Name [Course Name],
                count(*) [Students]
            from Course crs, Instructor ins, dbo.StdCourse std
            where crs.Ins_ID = ins.Ins_SSN and ins.Ins_SSN =@id AND std.Crs_ID = crs.Crs_ID
            group by crs.Crs_Name
        end
END
GO
```

Uses

[dbo].[Course]
[dbo].[Instructor]
[dbo].[StdCourse]
repo

[repo].[SP_SelectTopicsByCourse]

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     PROC [repo].[SP_SelectTopicsByCourse]
    @Crs_ID INT = NULL
AS BEGIN
    BEGIN TRY
        /*(SELECT
            CONCAT (ins.Ins_Fname,' ',ins.Ins_Lname) [Instructor]
        FROM Instructor ins, Course crs
        WHERE crs.Ins_ID = ins.Ins_SSN AND crs.Crs_ID = @Crs_ID)*/
        EXEC SP_SelectTopics @Crs_ID
    END TRY
    BEGIN CATCH
       PRINT 'An error occurred: ';
    END CATCH;
END;
GO
```

Uses

[dbo].[SP_SelectTopics]
repo

Project > (local)\SQLEXPRESS > User databases > SQL_Project >
Programmability > Stored Procedures > repo.SP_STDCourseGrade

[repo].[SP_STDCourseGrade]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE    PROC [repo].[SP_STDCourseGrade]
    @stdID int
AS begin
    select
        dept.Dept_Name [Department],
        CONCAT (st.St_Fname,' ',st.St_Lname) [Student],
        CONCAT (ins.Ins_Fname,' ',ins.Ins_Lname) [Instructor],
        crs.Crs_Name AS [Course],
        ISNULL(StCrs.Grade,-1) AS [Grade]
    from Student st, Course crs, StdCourse StCrs, dbo.Instructor ins, dbo.Department dept
    where StCrs.St_ID=st.St_ID AND ins.Ins_SSN = crs.Ins_ID AND dept.Dept_ID = st.Dept_ID AND
st.St_ID = @stdID
    and StCrs.Crs_ID=crs.Crs_ID

END
GO
```

Uses

[dbo].[Course]
[dbo].[Department]
[dbo].[Instructor]
[dbo].[StdCourse]
[dbo].[Student]
repo

[repo].[SP_StudentExamAns]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@examID	int	4
@stID	int	4

SQL Script

```
CREATE     proc [repo].[SP_StudentExamAns]
    @examID int, @stID int
as begin
    SELECT
        dept.Dept_Name [Department],
        CONCAT (st.St_Fname,' ',st.St_Lname) [Student],
        ques.Ques_Body AS [Question],
        ISNULL(SUBSTRING((SELECT c.Ques_Choices
                          FROM dbo.QuesChoices c
                          WHERE c.Ques_ID = ques.Ques_ID
                          AND LOWER(SUBSTRING(c.Ques_Choices, 1, 1)) = LOWER(ques.Ques_Answer)),
3,500), '') AS [Model Answer],

        ISNULL(SUBSTRING((SELECT c.Ques_Choices
                      FROM dbo.QuesChoices c
                      WHERE c.Ques_ID = ques.Ques_ID
                      AND LOWER(SUBSTRING(c.Ques_Choices, 1, 1)) = LOWER(exam.St_Answer)),
3,500), '') AS [Student Answer]
FROM
    Question ques, StdExam exam , Department dept, Student st
WHERE ques.Ques_ID = exam.Ques_ID AND   st.St_ID = exam.St_SSN AND st.Dept_ID = dept.Dept_ID AND
exam.Exam_ID = @examID AND exam.St_SSN = @stID
ORDER BY
    ques.Ques_ID;
END
GO
```

Uses

[dbo].[Department]
[dbo].[QuesChoices]
[dbo].[Question]
[dbo].[StdExam]
[dbo].[Student]
repo

Schemas

Objects

Name
repo

repo

Properties

Property	Value
Owner	dbo

SQL Script

```
CREATE SCHEMA [repo]
AUTHORIZATION [dbo]
GO
```

[repo].[SP_¬Dept¬Students¬Info]
[repo].[SP_¬Exam¬Questions]