

















































































(local)\MSSQLSERVER01 Documentation










































OnlineExamination








Server	(local)\MSSQLSERVER01
Author	Omar
Created	Monday, February 10, 2025 12:50:00 AM
File Path	C:\Users\Omar\Music\local_MSSQLSERVER01_documentation-2025-02-10T00-50-00.docx

Table of Contents

Table of Contents	2
 (local)\MSSQLSERVER01	6
 User databases	8
 OnlineExamination Database.....	9
 Tables	11
 [dbo].[Admin].....	12
 [dbo].[Branch].....	14
 [dbo].[Branch_Instructor].....	16
 [dbo].[Branch_Track].....	18
 [dbo].[Course]	20
 [dbo].[Course_Exam]	22
 [dbo].[Course_Student]	24
 [dbo].[Exam].....	26
 [dbo].[Exam_Question].....	28
 [dbo].[Exam_Track]	30
 [dbo].[Instructor]	32
 [dbo].[Instructor_Course].....	34
 [dbo].[Instructor_EExam].....	36
 [dbo].[Instructor_Phone].....	38
 [dbo].[Instructor_Track]	40
 [dbo].[Question].....	42
 [dbo].[Question_Choices].....	44
 [dbo].[StudenAnswer].....	46
 [dbo].[Student].....	48
 [dbo].[Student_Exam].....	50
 [dbo].[Student_Phone]	52
 [dbo].[Track].....	54
 [dbo].[Track_Course].....	56
 Stored Procedures	58
 [dbo].[add_ Question_ToExam].....	61
 [dbo].[AddAndInsertQuestion]	62
 [dbo].[AddChoice]	63
 [dbo].[AddQuestionWithChoices].....	64
 [dbo].[CalTotalGrade].....	66
 [dbo].[CalTotalGradeForExam].....	67
 [dbo].[DeleteBranch]	68
 [dbo].[DeleteBranchPhone]	69
 [dbo].[DeleteChoice].....	70
 [dbo].[deletecourse].....	71
 [dbo].[deletecoursetrack]	73


 [dbo].[DeleteExam].....	74
 [dbo].[DeleteInstructor].....	75
 [dbo].[DeleteInstructorTeachCourse].....	76
 [dbo].[DeleteQuestion]	77
 [dbo].[DeleteStudent]	78
 [dbo].[deleteStudentPhone].....	79
 [dbo].[deleteworkon].....	80
 [dbo].[GetAllCourses]	81
 [dbo].[GetAllCoursesWithTracks]	82
 [dbo].[GetAllExams]	83
 [dbo].[GetAllExamsAndCoursesAndStudents].....	84
 [dbo].[GetAllExamsAndCoursesAndStudentsForSpecificInstructor].....	85
 [dbo].[GetAllExamsForSpecificStudentBasedOnTrackID].....	87
 [dbo].[GetAllExamsWithDetails].....	88
 [dbo].[GetAllInstructors].....	89
 [dbo].[GetAllInstructorsWithBranches].....	90
 [dbo].[GetAllInstructorsWithCourses].....	91
 [dbo].[GetAllStudentsReportWithCourseId]	92
 [dbo].[GetAllStudentsReportWithTrackId]	93
 [dbo].[GetAllStudentsWithCourses]	94
 [dbo].[GetAllTracksByBranchId]	95
 [dbo].[GetAllTracksByInsId].....	96
 [dbo].[GetAllTracksWithCourses]	97
 [dbo].[GetCoursesByTrackIdAndBranchId].....	98
 [dbo].[GetCoursesByTrackIdAndInsId]	99
 [dbo].[GetExam]	100
 [dbo].[getExamGrade]	101
 [dbo].[GetExamsForSpecificBranchTrackCourse].....	102
 [dbo].[GetExamsForSpecificBranchTrackCourseInsId].....	103
 [dbo].[GetExamsForSpecificInstructor]	104
 [dbo].[GetExamsForSpecificInstructorTrackCourse]	105
 [dbo].[GetExamWithQuestionsForSpecificInstructor]	106
 [dbo].[GetInstructorCoursesByInsId].....	107
 [dbo].[GetQuestion]	108
 [dbo].[GetStudentsByBranchTrackCourse].....	109
 [dbo].[GetStudentsByBranchTrackCourseExamToGetGrades].....	110
 [dbo].[GetStudentsByInstructorTrackCourseExamToGetGrades]	112
 [dbo].[Insert_Question].....	114
 [dbo].[InsertAnswer]	115
 [dbo].[InsertBranch].....	116
 [dbo].[insertBranchInstructor]	117

 [dbo].[InsertBranchPhone].....	118
 [dbo].[insertcourse].....	119
 [dbo].[InsertExam].....	120
 [dbo].[InsertInstructor].....	122
 [dbo].[insertInstructorCourse].....	124
 [dbo].[InsertInstructorTeachCourse].....	125
 [dbo].[InsertIntoStudentPhone].....	126
 [dbo].[InsertStudent].....	127
 [dbo].[InsertStudentExam].....	129
 [dbo].[InsertStudentsWithCourse].....	130
 [dbo].[insertTrackCourse].....	131
 [dbo].[insertTrackInstructor].....	133
 [dbo].[ModelAnswer].....	134
 [dbo].[ModelAnswerEx].....	136
 [dbo].[online_BranchTrack_Insert].....	138
 [dbo].[online_BranchTrack_SelectAll].....	140
 [dbo].[online_BranchTrack_SelectByPK].....	141
 [dbo].[online_BranchTrack_Update].....	142
 [dbo].[online_GetAllStudentsphones].....	144
 [dbo].[online_GetStudentPhoneById].....	145
 [dbo].[online_Track_Delete].....	146
 [dbo].[online_Track_Insert].....	147
 [dbo].[online_Track_SelectAll].....	148
 [dbo].[online_Track_SelectByTrackId].....	149
 [dbo].[online_Track_Update].....	150
 [dbo].[online_TrackBranch_Delete].....	151
 [dbo].[SelectAllBranches].....	152
 [dbo].[SelectAllBranchPhones].....	153
 [dbo].[SelectAllInstructorTeachCourse].....	154
 [dbo].[SelectAllStudents].....	155
 [dbo].[SelectBranchById].....	156
 [dbo].[SelectBranchPhoneById].....	157
 [dbo].[selectcourse].....	158
 [dbo].[selectinstructor].....	159
 [dbo].[SelectInstructorTeachCourseById].....	160
 [dbo].[SelectStudentById].....	161
 [dbo].[selectworkon].....	162
 [dbo].[sp_AuthenticateUser].....	163
 [dbo].[Track_course_SelctById].....	165
 [dbo].[UpdateBranch].....	166
 [dbo].[updatecourse].....	167

 [dbo].[updatecoursetrack]	168
 [dbo].[UpdateExam]	169
 [dbo].[UpdateInstructor]	171
 [dbo].[UpdateInstructorTeachCourse]	173
 [dbo].[UpdateStudent]	174
 [dbo].[updateStudentPhone]	175
 [dbo].[updateworkon]	176

(local)\MSSQLSERVER01

Databases (1)

-  OnlineExamination

Server Properties

Property	Value
Product	Microsoft SQL Server
Version	16.0.1135.2
Language	English (United States)
Platform	NT x64
Edition	Developer Edition (64-bit)
Engine Edition	3 (Enterprise)
Processors	12
OS Version	6.3 (19045)
Physical Memory	16146
Is Clustered	False
Root Directory	C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER01\MSSQL
Collation	SQL_Latin1_General_CP1_CI_AS

Server Settings

Property	Value
Default data file path	C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER01\MSSQL\DATA\
Default backup file path	C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER01\MSSQL\Backup
Default log file path	C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER01\MSSQL\DATA\
Recovery Interval (minutes)	0
Default index fill factor	0
Default backup media retention	0
Compress Backup	False


Advanced Server Settings

Property	Value
Locks	0
Nested triggers enabled	True
Allow triggers to fire others	True
Default language	English
Network packet size	4096

Default fulltext language LCID	1033
Two-digit year cutoff	2049
Remote login timeout	10
Cursor threshold	-1
Max text replication size	65536
Parallelism cost threshold	5
Max degree of parallelism	12
Min server memory	16
Max server memory	2147483647
Scan for startup procs	False
Transform noise words	False
CLR enabled	False
Blocked process threshold	0
Filestream access level	False
Optimize for ad hoc workloads	False
CLR strict security	True

User databases

Databases (1)

-  OnlineExamination

OnlineExamaination Database

Database Properties

Property	Value
SQL Server Version	Max
Compatibility Level	Max
Last backup time	02/10/2025
Last log backup time	-
Creation date	Jan 25 2025
Users	4
Database Encryption Enabled	False
Database Encryption Algorithm	None
Database size	80.00 MB
Unallocated space	62.15 MB

Database Options

Property	Value
Compatibility Level	160
Database collation	Arabic_CI_AS
Restrict access	MULTI_USER
Is read-only	False
Auto close	False
Auto shrink	False
Database status	ONLINE
In standby	False
Cleanly shutdown	False
Supplemental logging enabled	False
Snapshot isolation state	OFF
Read committed snapshot on	False
Recovery model	SIMPLE
Page verify option	CHECKSUM
Auto create statistics	True
Auto update statistics	True
Auto update statistics asynchronously	False
ANSI NULL default	False
ANSI NULL enabled	False
ANSI padding enabled	False
ANSI warnings enabled	False
Arithmetic abort enabled	False
Concatenating NULL yields NULL	False

Numeric roundabort enabled	False
Quoted Identifier On	False
Recursive triggers enabled	False
Close cursors on commit	False
Local cursors by default	False
Fulltext enabled	True
Trustworthy	False
Database chaining	False
Forced parameterization	False
Master key encrypted by server	False
Published	False
Subscribed	False
Merge published	False
Is distribution database	False
Sync with backup	False
Service broker GUID	9e78ed9e-3d1c-4c17-8609-5f2ce200c095
Service broker enabled	False
Date correlation	False
CDC enabled	False
Encrypted	False
Honor broker priority	False
Default language	English
Default fulltext language LCID	1033
Nested triggers enabled	True
Transform noise words	False
Two-digit year cutoff	2049
Containment	NONE
Target recovery time	60
Database owner	DESKTOP-EPS264S\Omar

Files

Name	Type	Size	Maxsize	Autogrowth	File Name
Online-Examination	Data	72.00 MB	unlimited	64.00 MB	C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER01\MSSQL\DATA\OnlineExamination.mdf
Online-Examination_log	Log	8.00 MB	2048.00 GB	64.00 MB	C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER01\MSSQL\DATA\OnlineExamination_log.ldf

Tables

Objects


Name
dbo.Admin
dbo.Branch
dbo.Branch_Instructor
dbo.Branch_Track
dbo.Course
dbo.Course_Exam
dbo.Course_Student
dbo.Exam
dbo.Exam_Question
dbo.Exam_Track
dbo.Instructor
dbo.Instructor_Course
dbo.Instructor_EExam
dbo.Instructor_Phone
dbo.Instructor_Track
dbo.Question
dbo.Question_Choices
dbo.StudenAnswer
dbo.Student
dbo.Student_Exam
dbo.Student_Phone
dbo.Track
dbo.Track_Course

[dbo].[Admin]

Properties

Property	Value
Collation	Arabic_CI_AS
Row Count (~)	1
Created	4:26:36 PM Friday, February 7, 2025
Last Modified	4:26:37 PM Friday, February 7, 2025

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity	Default
	A_ID	int	4	NOT NULL	1 - 1	
	F_Name	varchar(50)	50	NULL allowed		
	L_Name	varchar(50)	50	NULL allowed		
	A_Email	varchar(50)	50	NULL allowed		
	Role	varchar(50)	50	NOT NULL		('Admin')
	Password	varchar(255)	255	NULL allowed		

Indexes

Key	Name	Key Columns	Unique
	PK_Admin	A_ID	True

SQL Script

```
CREATE TABLE [dbo].[Admin]
(
    [A_ID] [int] NOT NULL IDENTITY(1, 1),
    [F_Name] [varchar] (50) COLLATE Arabic_CI_AS NULL,
    [L_Name] [varchar] (50) COLLATE Arabic_CI_AS NULL,
    [A_Email] [varchar] (50) COLLATE Arabic_CI_AS NULL,
    [Role] [varchar] (50) COLLATE Arabic_CI_AS NOT NULL CONSTRAINT [DF_Admin_Role] DEFAULT ('Admin'),
    [Password] [varchar] (255) COLLATE Arabic_CI_AS NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Admin] ADD CONSTRAINT [PK_Admin] PRIMARY KEY CLUSTERED ([A_ID]) ON [PRIMARY]
GO
```

Project > (local)\MSSQLSERVER01 > User databases > Online-
Examination > Tables > dbo.Admin

Used By


[dbo].[sp_AuthenticateUser]

[dbo].[Branch]

Properties

Property	Value
Collation	Arabic_CI_AS
Row Count (~)	11
Created	8:50:24 PM Monday, January 27, 2025
Last Modified	8:08:29 PM Sunday, February 9, 2025

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
	B_Id	int	4	NOT NULL	1 - 1
	B_Name	varchar(20)	20	NOT NULL	
	B_Location	varchar(20)	20	NULL allowed	
	B_Phone	int	4	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
	PK__Branch__4B26EFC6FD36E633	B_Id	True

SQL Script

```
CREATE TABLE [dbo].[Branch]
(
    [B_Id] [int] NOT NULL IDENTITY(1, 1),
    [B_Name] [varchar] (20) COLLATE Arabic_CI_AS NOT NULL,
    [B_Location] [varchar] (20) COLLATE Arabic_CI_AS NULL,
    [B_Phone] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Branch] ADD CONSTRAINT [PK__Branch__4B26EFC6FD36E633] PRIMARY KEY
CLUSTERED ([B_Id]) ON [PRIMARY]
GO
```

Used By

[dbo].[Branch_Instructor]
[dbo].[Branch_Track]
[dbo].[DeleteBranch]
[dbo].[GetAllInstructorsWithBranches]
[dbo].[GetAllTracksByBranchId]
[dbo].[InsertBranch]

Project > (local)\MSSQLSERVER01 > User databases > Online-
Examination > Tables > dbo.Branch



[dbo].[online_BranchTrack_Insert]
[dbo].[online_BranchTrack_Update]
[dbo].[SelectAllBranches]
[dbo].[SelectBranchById]
[dbo].[UpdateBranch]

[dbo].[Branch_Instructor]

Properties

Property	Value
Row Count (~)	3
Created	9:40:18 AM Saturday, January 18, 2025
Last Modified	8:08:29 PM Sunday, February 9, 2025

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
	B_Id	int	4	NOT NULL
	Ins_Id	int	4	NOT NULL

Indexes

Key	Name	Key Columns	Unique
	PK__Branch_I__8A77AF589C15CA54	B_Id, Ins_Id	True

Foreign Keys

Name	No Check	Columns
FK__Branch_In__Ins_I__5165187F	True	Ins_Id->[dbo].[Instructor].[Ins_Id]
FK__Branch_Ins__B_Id__5070F446	True	B_Id->[dbo].[Branch].[B_Id]

SQL Script

```
CREATE TABLE [dbo].[Branch_Instructor]
(
    [B_Id] [int] NOT NULL,
    [Ins_Id] [int] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Branch_Instructor] ADD CONSTRAINT [PK__Branch_I__8A77AF589C15CA54]
PRIMARY KEY CLUSTERED ([B_Id], [Ins_Id]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Branch_Instructor] WITH NOCHECK ADD CONSTRAINT [FK__Branch_In__Ins_I__5165187F] FOREIGN KEY ([Ins_Id]) REFERENCES [dbo].[Instructor] ([Ins_Id])
GO
ALTER TABLE [dbo].[Branch_Instructor] WITH NOCHECK ADD CONSTRAINT [FK__Branch_Ins__B_Id__5070F446] FOREIGN KEY ([B_Id]) REFERENCES [dbo].[Branch] ([B_Id])
GO
```


Uses

[dbo].[Branch]
[dbo].[Instructor]

Used By



[dbo].[GetAllInstructorsWithBranches]
[dbo].[insertBranchInstructor]
[dbo].[InsertInstructor]

[dbo].[Branch_Track]

Properties

Property	Value
Row Count (~)	7
Created	9:40:18 AM Saturday, January 18, 2025
Last Modified	8:08:29 PM Sunday, February 9, 2025

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
	B_Id	int	4	NOT NULL
	T_Id	int	4	NOT NULL

Indexes

Key	Name	Key Columns	Unique
	PK__Branch_T__731D5E3F95590C19	B_Id, T_Id	True

Foreign Keys

Name	No Check	Columns
FK__Branch_Tra__B_Id__48CFD27E	True	B_Id->[dbo].[Branch].[B_Id]
FK__Branch_Tra__T_Id__49C3F6B7	True	T_Id->[dbo].[Track].[T_Id]

SQL Script

```
CREATE TABLE [dbo].[Branch_Track]
(
    [B_Id] [int] NOT NULL,
    [T_Id] [int] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Branch_Track] ADD CONSTRAINT [PK__Branch_T__731D5E3F95590C19] PRIMARY
KEY CLUSTERED ([B_Id], [T_Id]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Branch_Track] WITH NOCHECK ADD CONSTRAINT [FK__Branch_Tra__B_-
Id__48CFD27E] FOREIGN KEY ([B_Id]) REFERENCES [dbo].[Branch] ([B_Id])
GO
ALTER TABLE [dbo].[Branch_Track] WITH NOCHECK ADD CONSTRAINT [FK__Branch_Tra__T_-
Id__49C3F6B7] FOREIGN KEY ([T_Id]) REFERENCES [dbo].[Track] ([T_Id])
GO
```

Uses

[dbo].[Branch]
[dbo].[Track]

Used By


[dbo].[GetAllTracksByBranchId]
[dbo].[GetCoursesByTrackIdAndBranchId]
[dbo].[GetExamsForSpecificBranchTrackCourse]
[dbo].[GetStudentsByBranchTrackCourse]
[dbo].[GetStudentsByBranchTrackCourseExamToGetGrades]
[dbo].[online_BranchTrack_Insert]
[dbo].[online_BranchTrack_SelectAll]
[dbo].[online_BranchTrack_SelectByPK]
[dbo].[online_BranchTrack_Update]
[dbo].[online_TrackBranch_Delete]

[dbo].[Course]

Properties

Property	Value
Collation	Arabic_CI_AS
Row Count (~)	7
Created	8:51:31 PM Monday, January 27, 2025
Last Modified	8:08:29 PM Sunday, February 9, 2025

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
	C_Id	int	4	NOT NULL	1 - 1
	C_Name	varchar(30)	30	NOT NULL	
	C_Duration	varchar(50)	50	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
	PK__Course__A9FDEC324ABC83FD	C_Id	True

SQL Script

```
CREATE TABLE [dbo].[Course]
(
    [C_Id] [int] NOT NULL IDENTITY(1, 1),
    [C_Name] [varchar] (30) COLLATE Arabic_CI_AS NOT NULL,
    [C_Duration] [varchar] (50) COLLATE Arabic_CI_AS NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Course] ADD CONSTRAINT [PK Course A9FDEC324ABC83FD] PRIMARY KEY
CLUSTERED ([C_Id]) ON [PRIMARY]
GO
```

Used By

[dbo].[Course_Exam]
[dbo].[Course_Student]
[dbo].[Instructor_Course]
[dbo].[Track_Course]
[dbo].[deletecourse]
[dbo].[GetAllCourses]
[dbo].[GetAllCoursesWithTracks]
[dbo].[GetAllExamsAndCoursesAndStudents]

Project > (local)\MSSQLSERVER01 > User databases > Online-
Examaination > Tables > dbo.Course



[dbo].[GetAllExamsAndCoursesAndStudentsForSpecificInstructor]
[dbo].[GetAllInstructorsWithCourses]
[dbo].[GetAllStudentsReportWithCourseId]
[dbo].[GetAllStudentsWithCourses]
[dbo].[GetCoursesByTrackIdAndBranchId]
[dbo].[GetCoursesByTrackIdAndInsId]
[dbo].[GetInstructorCoursesByInsId]
[dbo].[insertcourse]
[dbo].[insertTrackCourse]
[dbo].[selectcourse]
[dbo].[updatecourse]

[dbo].[Course_Exam]

Properties

Property	Value
Row Count (~)	7
Created	9:55:35 PM Saturday, February 8, 2025
Last Modified	8:08:29 PM Sunday, February 9, 2025

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
	C_Id	int	4	NOT NULL
	E_Id	int	4	NOT NULL

Indexes

Key	Name	Key Columns	Unique
	PK_Course_Exam	C_Id, E_Id	True

Foreign Keys

Name	No Check	Columns
FK_Course_Exam_Course	True	C_Id->[dbo].[Course].[C_Id]
FK_Course_Exam_Exam	True	E_Id->[dbo].[Exam].[E_Id]

SQL Script

```
CREATE TABLE [dbo].[Course_Exam]
(
    [C_Id] [int] NOT NULL,
    [E_Id] [int] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Course_Exam] ADD CONSTRAINT [PK_Course_Exam] PRIMARY KEY CLUSTERED ([C_Id], [E_Id]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Course_Exam] WITH NOCHECK ADD CONSTRAINT [FK_Course_Exam_Course] FOREIGN KEY ([C_Id]) REFERENCES [dbo].[Course] ([C_Id])
GO
ALTER TABLE [dbo].[Course_Exam] WITH NOCHECK ADD CONSTRAINT [FK_Course_Exam_Exam] FOREIGN KEY ([E_Id]) REFERENCES [dbo].[Exam] ([E_Id])
GO
```

Uses

[dbo].[Course]
[dbo].[Exam]

Used By



[dbo].[GetExamsForSpecificBranchTrackCourse]
[dbo].[GetExamsForSpecificBranchTrackCourseInsId]
[dbo].[GetExamsForSpecificInstructorTrackCourse]
[dbo].[InsertExam]

[dbo].[Course_Student]

Properties

Property	Value
Row Count (~)	3
Created	9:40:18 AM Saturday, January 18, 2025
Last Modified	8:08:29 PM Sunday, February 9, 2025

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
	St_Id	int	4	NOT NULL
	C_Id	int	4	NOT NULL

Indexes

Key	Name	Key Columns	Unique
	PK__Course_S__839C9170ECC7CECA	St_Id, C_Id	True

Foreign Keys

Name	No Check	Columns
FK__Course_St__St_Id__5AEE82B9	True	St_Id->[dbo].[Student].[St_Id]
FK__Course_Stu__C_Id__5BE2A6F2	True	C_Id->[dbo].[Course].[C_Id]

SQL Script

```
CREATE TABLE [dbo].[Course_Student]
(
    [St_Id] [int] NOT NULL,
    [C_Id] [int] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Course_Student] ADD CONSTRAINT [PK__Course_S__839C9170ECC7CECA] PRIMARY
KEY CLUSTERED ([St_Id], [C_Id]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Course_Student] WITH NOCHECK ADD CONSTRAINT [FK__Course_St__St_-
Id__5AEE82B9] FOREIGN KEY ([St_Id]) REFERENCES [dbo].[Student] ([St_Id])
GO
ALTER TABLE [dbo].[Course_Student] WITH NOCHECK ADD CONSTRAINT [FK__Course_Stu__C_-
Id__5BE2A6F2] FOREIGN KEY ([C_Id]) REFERENCES [dbo].[Course] ([C_Id])
GO
```


Uses

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

Used By


[dbo].[deletecourse]
[dbo].[GetAllExamsAndCoursesAndStudentsForSpecificInstructor]
[dbo].[GetAllStudentsReportWithCourseId]
[dbo].[GetAllStudentsWithCourses]
[dbo].[GetStudentsByBranchTrackCourse]
[dbo].[GetStudentsByBranchTrackCourseExamToGetGrades]
[dbo].[GetStudentsByInstructorTrackCourseExamToGetGrades]
[dbo].[InsertStudentsWithCourse]

[dbo].[Exam]

Properties

Property	Value
Collation	Arabic_CI_AS
Row Count (~)	7
Created	7:23:34 PM Thursday, January 23, 2025
Last Modified	8:08:29 PM Sunday, February 9, 2025

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
	E_Id	int	4	NOT NULL	1 - 1
	E_Total_Score	int	4	NOT NULL	
	E_Name	varchar(20)	20	NULL allowed	
	E_Duration	float	8	NULL allowed	
	E_Date	date	3	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
	PK__Exam__D730AF34FFBA1034	E_Id	True

SQL Script

```
CREATE TABLE [dbo].[Exam]
(
    [E_Id] [int] NOT NULL IDENTITY(1, 1),
    [E_Total_Score] [int] NOT NULL,
    [E_Name] [varchar] (20) COLLATE Arabic_CI_AS NULL,
    [E_Duration] [float] NULL,
    [E_Date] [date] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Exam] ADD CONSTRAINT [PK__Exam__D730AF34FFBA1034] PRIMARY KEY CLUSTERED
([E_Id]) ON [PRIMARY]
GO
```

Used By

[dbo].[Course_Exam]
[dbo].[Exam_Question]
[dbo].[Exam_Track]

Project > (local)\MSSQLSERVER01 > User databases > Online-
Examination > Tables > dbo.Exam





[dbo].[Instructor_EExam]
[dbo].[StudenAnswer]
[dbo].[Student_Exam]
[dbo].[CalTotalGrade]
[dbo].[CalTotalGradeForExam]
[dbo].[DeleteExam]
[dbo].[GetAllExams]
[dbo].[GetAllExamsAndCoursesAndStudents]
[dbo].[GetAllExamsAndCoursesAndStudentsForSpecificInstructor]
[dbo].[GetAllExamsForSpecificStudentBasedOnTrackID]
[dbo].[GetAllExamsWithDetails]
[dbo].[GetExam]
[dbo].[getExamGrade]
[dbo].[GetExamsForSpecificBranchTrackCourse]
[dbo].[GetExamsForSpecificBranchTrackCourseInsId]
[dbo].[GetExamsForSpecificInstructor]
[dbo].[GetExamsForSpecificInstructorTrackCourse]
[dbo].[GetExamWithQuestionsForSpecificInstructor]
[dbo].[GetStudentsByBranchTrackCourseExamToGetGrades]
[dbo].[GetStudentsByInstructorTrackCourseExamToGetGrades]
[dbo].[InsertExam]
[dbo].[UpdateExam]

[dbo].[Exam_Question]

Properties

Property	Value
Row Count (~)	7
Created	9:40:18 AM Saturday, January 18, 2025
Last Modified	8:08:29 PM Sunday, February 9, 2025

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
 	E_Id	int	4	NOT NULL
 	Q_Id	int	4	NOT NULL

Indexes

Key	Name	Key Columns	Unique
	PK__Exam_Que__287F6C447074BDB1	E_Id, Q_Id	True

Triggers

Name	ANSI Nulls On	Quoted Identifier On	On
T_Exam_Question	True	True	After Insert

Foreign Keys

Name	No Check	Columns
FK__Exam_Quest__E_Id__656C112C	True	E_Id->[dbo].[Exam].[E_Id]
FK__Exam_Quest__Q_Id__66603565	True	Q_Id->[dbo].[Question].[Q_Id]

SQL Script

```
CREATE TABLE [dbo].[Exam_Question]
(
    [E_Id] [int] NOT NULL,
    [Q_Id] [int] NOT NULL
) ON [PRIMARY]
GO

Create trigger [dbo].[T_Exam_Question]
```

```
on [dbo].[Exam_Question]
after insert
as
    if exists(select * from inserted)
        select * from inserted
GO
ALTER TABLE [dbo].[Exam_Question] ADD CONSTRAINT [PK__Exam_Que__287F6C447074BDB1] PRIMARY
KEY CLUSTERED ([E_Id], [Q_Id]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Exam_Question] WITH NOCHECK ADD CONSTRAINT [FK__Exam_Quest__E_-
Id__656C112C] FOREIGN KEY ([E_Id]) REFERENCES [dbo].[Exam] ([E_Id])
GO
ALTER TABLE [dbo].[Exam_Question] WITH NOCHECK ADD CONSTRAINT [FK__Exam_Quest__Q_-
Id__66603565] FOREIGN KEY ([Q_Id]) REFERENCES [dbo].[Question] ([Q_Id])
GO
```

Uses

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

Used By





[dbo].[add_Question_ToExam]
[dbo].[CalTotalGrade]
[dbo].[CalTotalGradeForExam]
[dbo].[DeleteQuestion]
[dbo].[GetAllExamsAndCoursesAndStudentsForSpecificInstructor]
[dbo].[GetExam]
[dbo].[GetExamWithQuestionsForSpecificInstructor]

[dbo].[Exam_Track]

Properties

Property	Value
Row Count (~)	7
Created	2:07:03 AM Sunday, February 9, 2025
Last Modified	8:08:29 PM Sunday, February 9, 2025

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
 	E_Id	int	4	NOT NULL
 	T_Id	int	4	NOT NULL

Indexes

Key	Name	Key Columns	Unique
	PK_Exam_Track	E_Id, T_Id	True

Foreign Keys

Name	No Check	Columns
FK_Exam_Track_Exam	True	E_Id->[dbo].[Exam].[E_Id]
FK_Exam_Track_Track	True	T_Id->[dbo].[Track].[T_Id]

SQL Script

```
CREATE TABLE [dbo].[Exam_Track]
(
    [E_Id] [int] NOT NULL,
    [T_Id] [int] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Exam_Track] ADD CONSTRAINT [PK_Exam_Track] PRIMARY KEY CLUSTERED ([E_
Id], [T_Id]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Exam_Track] WITH NOCHECK ADD CONSTRAINT [FK_Exam_Track_Exam] FOREIGN KEY
([E_Id]) REFERENCES [dbo].[Exam] ([E_Id])
GO
ALTER TABLE [dbo].[Exam_Track] WITH NOCHECK ADD CONSTRAINT [FK_Exam_Track_Track] FOREIGN
KEY ([T_Id]) REFERENCES [dbo].[Track] ([T_Id])
GO
```

Uses

[dbo].[Exam]
[dbo].[Track]

Used By


[dbo].[GetAllExamsForSpecificStudentBasedOnTrackID]
[dbo].[InsertExam]

[dbo].[Instructor]

Properties

Property	Value
Collation	Arabic_CI_AS
Row Count (~)	3
Created	8:52:42 PM Monday, January 27, 2025
Last Modified	8:08:29 PM Sunday, February 9, 2025

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity	Default
	Ins_Id	int	4	NOT NULL	1 - 1	
	F_Name	varchar(20)	20	NOT NULL		
	L_Name	varchar(20)	20	NOT NULL		
	Ins_Age	int	4	NULL allowed		
	Ins_Email	varchar(30)	30	NULL allowed		
	Ins_Address	varchar(50)	50	NULL allowed		
	Ins_Gender	varchar(6)	6	NULL allowed		
	Ins_DOB	date	3	NULL allowed		
	Ins_Salary	decimal(10,2)	9	NULL allowed		
	Role	varchar(50)	50	NULL allowed		('Instructor')
	Password	varchar(255)	255	NULL allowed		((123456))

Indexes

Key	Name	Key Columns	Unique
	PK__Instruct__151409EDE99388E5	Ins_Id	True

SQL Script

```
CREATE TABLE [dbo].[Instructor]
(
    [Ins_Id] [int] NOT NULL IDENTITY(1, 1),
    [F_Name] [varchar] (20) COLLATE Arabic_CI_AS NOT NULL,
    [L_Name] [varchar] (20) COLLATE Arabic_CI_AS NOT NULL,
    [Ins_Age] [int] NULL,
    [Ins_Email] [varchar] (30) COLLATE Arabic_CI_AS NULL,
    [Ins_Address] [varchar] (50) COLLATE Arabic_CI_AS NULL,
    [Ins_Gender] [varchar] (6) COLLATE Arabic_CI_AS NULL,
    [Ins_DOB] [date] NULL,
    [Ins_Salary] [decimal] (10, 2) NULL,
```


Project > (local)\MSSQLSERVER01 > User databases > Online-
Examination > Tables > dbo.Instructor

```
[Role] [varchar] (50) COLLATE Arabic_CI_AS NULL CONSTRAINT [DF_Instructor_Role] DEFAULT ('Instructor'),  
[Password] [varchar] (255) COLLATE Arabic_CI_AS NULL CONSTRAINT [DF_Instructor_Password] DEFAULT ((123456))  
) ON [PRIMARY]  
GO  
ALTER TABLE [dbo].[Instructor] ADD CONSTRAINT [PK__Instruct__151409EDE99388E5] PRIMARY KEY CLUSTERED ([Ins_Id]) ON [PRIMARY]  
GO
```

Used By



[dbo].[Branch_Instructor]
[dbo].[Instructor_Course]
[dbo].[Instructor_EExam]
[dbo].[Instructor_Phone]
[dbo].[Instructor_Track]
[dbo].[DeleteInstructor]
[dbo].[GetAllInstructors]
[dbo].[GetAllInstructorsWithBranches]
[dbo].[GetAllInstructorsWithCourses]
[dbo].[GetAllTracksByInsId]
[dbo].[InsertInstructor]
[dbo].[selectinstructor]
[dbo].[sp_AuthenticateUser]
[dbo].[UpdateInstructor]

[dbo].[Instructor_Course]

Properties

Property	Value
Row Count (~)	6
Created	9:40:18 AM Saturday, January 18, 2025
Last Modified	8:08:29 PM Sunday, February 9, 2025

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
	Ins_Id	int	4	NOT NULL
	C_Id	int	4	NOT NULL

Indexes

Key	Name	Key Columns	Unique
	PK__Instruct__2F8BD72E24769282	Ins_Id, C_Id	True

Foreign Keys

Name	No Check	Columns
FK__Instructo__Ins_I_5441852A	True	Ins_Id->[dbo].[Instructor].[Ins_Id]
FK__Instructor__C_Id_5535A963	True	C_Id->[dbo].[Course].[C_Id]

SQL Script

```
CREATE TABLE [dbo].[Instructor_Course]
(
    [Ins_Id] [int] NOT NULL,
    [C_Id] [int] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Instructor_Course] ADD CONSTRAINT [PK__Instruct__2F8BD72E24769282]
PRIMARY KEY CLUSTERED ([Ins_Id], [C_Id]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Instructor_Course] WITH NOCHECK ADD CONSTRAINT [FK__Instructo__Ins_I_5441852A] FOREIGN KEY ([Ins_Id]) REFERENCES [dbo].[Instructor] ([Ins_Id])
GO
ALTER TABLE [dbo].[Instructor_Course] WITH NOCHECK ADD CONSTRAINT [FK__Instructor__C_Id_5535A963] FOREIGN KEY ([C_Id]) REFERENCES [dbo].[Course] ([C_Id])
GO
```

Uses

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

Used By



[dbo].[deletecourse]
[dbo].[DeleteInstructorTeachCourse]
[dbo].[GetAllExamsAndCoursesAndStudentsForSpecificInstructor]
[dbo].[GetAllInstructorsWithCourses]
[dbo].[GetAllStudentsReportWithCourseId]
[dbo].[GetInstructorCoursesByInslId]
[dbo].[insertInstructorCourse]
[dbo].[InsertInstructorTeachCourse]
[dbo].[insertTrackCourse]
[dbo].[SelectAllInstructorTeachCourse]
[dbo].[SelectInstructorTeachCourseById]
[dbo].[UpdateInstructorTeachCourse]

[dbo].[Instructor_EExam]


Properties

Property	Value
Row Count (~)	7
Created	6:35:08 PM Saturday, February 8, 2025
Last Modified	8:08:29 PM Sunday, February 9, 2025

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
	Ins_id	int	4	NOT NULL
	E_id	int	4	NOT NULL

Indexes

Key	Name	Key Columns	Unique
	PK_Instructor_EExam	Ins_id, E_id	True

Foreign Keys

Name	No Check	Columns
FK_Instructor_EExam_Exam	True	E_id->[dbo].[Exam].[E_Id]
FK_Instructor_EExam_Instructor	True	Ins_id->[dbo].[Instructor].[Ins_Id]

SQL Script

```
CREATE TABLE [dbo].[Instructor_EExam]
(
    [Ins_id] [int] NOT NULL,
    [E_id] [int] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Instructor_EExam] ADD CONSTRAINT [PK_Instructor_EExam] PRIMARY KEY
CLUSTERED ([Ins_id], [E_id]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Instructor_EExam] WITH NOCHECK ADD CONSTRAINT [FK_Instructor_EExam_Exam]
FOREIGN KEY ([E_id]) REFERENCES [dbo].[Exam] ([E_Id])
GO
ALTER TABLE [dbo].[Instructor_EExam] WITH NOCHECK ADD CONSTRAINT [FK_Instructor_EExam_-
Instructor] FOREIGN KEY ([Ins_id]) REFERENCES [dbo].[Instructor] ([Ins_Id])
GO
```

Uses

[dbo].[Exam]
[dbo].[Instructor]

Used By


[dbo].[GetAllExamsAndCoursesAndStudentsForSpecificInstructor]
[dbo].[GetExamsForSpecificInstructor]
[dbo].[GetExamsForSpecificInstructorTrackCourse]
[dbo].[InsertExam]

[dbo].[Instructor_Phone]

Properties

Property	Value
Row Count (~)	3
Created	9:40:18 AM Saturday, January 18, 2025
Last Modified	8:08:29 PM Sunday, February 9, 2025

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
	Ins_Id	int	4	NOT NULL
	Ins_Phone	int	4	NOT NULL

Indexes

Key	Name	Key Columns	Unique
	PK__Instruct__47293D19001767F7	Ins_Id, Ins_Phone	True

Foreign Keys

Name	No Check	Columns
FK__Instructo__Ins_I__5812160E	True	Ins_Id->[dbo].[Instructor].[Ins_Id]

SQL Script

```
CREATE TABLE [dbo].[Instructor_Phone]
(
    [Ins_Id] [int] NOT NULL,
    [Ins_Phone] [int] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Instructor_Phone] ADD CONSTRAINT [PK__Instruct__47293D19001767F7]
PRIMARY KEY CLUSTERED ([Ins_Id], [Ins_Phone]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Instructor_Phone] WITH NOCHECK ADD CONSTRAINT [FK__Instructo__Ins_I__5812160E] FOREIGN KEY ([Ins_Id]) REFERENCES [dbo].[Instructor] ([Ins_Id])
GO
```

Uses

[dbo].[Instructor]

Project > (local)\MSSQLSERVER01 > User databases > Online-
Examination > Tables > dbo.Instructor_Phone

Used By

[dbo].[GetAllInstructors]



[dbo].[InsertInstructor]

[dbo].[Instructor_Track]

Properties

Property	Value
Row Count (~)	5
Created	2:15:27 AM Sunday, February 9, 2025
Last Modified	8:08:29 PM Sunday, February 9, 2025

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
	Ins_Id	int	4	NOT NULL
	T_Id	int	4	NOT NULL

Indexes

Key	Name	Key Columns	Unique
	PK_Instructor_Track	Ins_Id, T_Id	True

Foreign Keys

Name	No Check	Columns
FK_Instructor_Track_Instructor	True	Ins_Id->[dbo].[Instructor].[Ins_Id]
FK_Instructor_Track_Track	True	T_Id->[dbo].[Track].[T_Id]

SQL Script

```
CREATE TABLE [dbo].[Instructor_Track]
(
    [Ins_Id] [int] NOT NULL,
    [T_Id] [int] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Instructor_Track] ADD CONSTRAINT [PK_Instructor_Track] PRIMARY KEY CLUSTERED ([Ins_Id], [T_Id]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Instructor_Track] WITH NOCHECK ADD CONSTRAINT [FK_Instructor_Track_Instructor] FOREIGN KEY ([Ins_Id]) REFERENCES [dbo].[Instructor] ([Ins_Id])
GO
ALTER TABLE [dbo].[Instructor_Track] WITH NOCHECK ADD CONSTRAINT [FK_Instructor_Track_Track] FOREIGN KEY ([T_Id]) REFERENCES [dbo].[Track] ([T_Id])
GO
```


Uses

[dbo].[Instructor]
[dbo].[Track]

Used By


[dbo].[GetAllTracksByInsId]
[dbo].[GetCoursesByTrackIdAndInsId]
[dbo].[GetExamsForSpecificBranchTrackCourseInsId]
[dbo].[GetExamsForSpecificInstructorTrackCourse]
[dbo].[GetInstructorCoursesByInsId]
[dbo].[InsertInstructor]
[dbo].[insertTrackCourse]

[dbo].[Question]

Properties

Property	Value
Collation	Arabic_CI_AS
Row Count (~)	7
Created	4:09:43 PM Sunday, January 19, 2025
Last Modified	8:08:29 PM Sunday, February 9, 2025

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
	Q_Id	int	4	NOT NULL	1 - 1
	Q_Grade	int	4	NULL allowed	
	Q_Type	varchar(10)	10	NULL allowed	
	Q_Title	varchar(255)	255	NULL allowed	
	Q_Model_Answer	int	4	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
	PK__Question__F4FC370EA00580F0	Q_Id	True

SQL Script

```
CREATE TABLE [dbo].[Question]
(
    [Q_Id] [int] NOT NULL IDENTITY(1, 1),
    [Q_Grade] [int] NULL,
    [Q_Type] [varchar] (10) COLLATE Arabic_CI_AS NULL,
    [Q_Title] [varchar] (255) COLLATE Arabic_CI_AS NULL,
    [Q_Model_Answer] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Question] ADD CONSTRAINT [PK__Question__F4FC370EA00580F0] PRIMARY KEY
CLUSTERED ([Q_Id]) ON [PRIMARY]
GO
```

Used By

[dbo].[Exam_Question]
[dbo].[Question_Choices]
[dbo].[StudenAnswer]

Project > (local)\MSSQLSERVER01 > User databases > Online-
Examination > Tables > dbo.Question



[dbo].[add_Question_ToExam]
[dbo].[AddQuestionWithChoices]
[dbo].[CalTotalGrade]
[dbo].[CalTotalGradeForExam]
[dbo].[DeleteQuestion]
[dbo].[GetAllExamsAndCoursesAndStudentsForSpecificInstructor]
[dbo].[GetExam]
[dbo].[GetExamWithQuestionsForSpecificInstructor]
[dbo].[GetQuestion]
[dbo].[GetStudentsByBranchTrackCourseExamToGetGrades]
[dbo].[GetStudentsByInstructorTrackCourseExamToGetGrades]
[dbo].[Insert_Question]
[dbo].[ModelAnswer]
[dbo].[ModelAnswerEx]

[dbo].[Question_Choices]

Properties

Property	Value
Collation	Arabic_CI_AS
Row Count (~)	22
Created	4:58:05 PM Sunday, January 19, 2025
Last Modified	8:08:29 PM Sunday, February 9, 2025

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
	Q_Id	int	4	NOT NULL
	Choice_Num	int	4	NOT NULL
	answer	varchar(50)	50	NOT NULL

Indexes

Key	Name	Key Columns	Unique
	PK__Question__7D7C5D169BF0F483	Q_Id, Choice_Num	True

Foreign Keys

Name	No Check	Columns
FK__Question_C__Q_Id__45F365D3	True	Q_Id->[dbo].[Question].[Q_Id]

SQL Script

```
CREATE TABLE [dbo].[Question_Choices]
(
    [Q_Id] [int] NOT NULL,
    [Choice_Num] [int] NOT NULL,
    [answer] [varchar] (50) COLLATE Arabic_CI_AS NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Question_Choices] ADD CONSTRAINT [PK__Question__7D7C5D169BF0F483]
PRIMARY KEY CLUSTERED ([Q_Id], [Choice_Num]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Question_Choices] WITH NOCHECK ADD CONSTRAINT [FK__Question_C__Q_-
Id__45F365D3] FOREIGN KEY ([Q_Id]) REFERENCES [dbo].[Question] ([Q_Id])
GO
```

Uses

[dbo].[Question]

Used By

[dbo].[AddChoice]

[dbo].[AddQuestionWithChoices]

[dbo].[DeleteChoice]

[dbo].[DeleteQuestion]

[dbo].[GetQuestion]

[dbo].[ModelAnswer]



[dbo].[ModelAnswerEx]

[dbo].[StudenAnswer]

Properties

Property	Value
Collation	Arabic_CI_AS
Heap	True
Row Count (~)	12
Created	5:36:57 PM Friday, February 7, 2025
Last Modified	8:08:29 PM Sunday, February 9, 2025

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
	S_Id	int	4	NULL allowed
	Exam_Name	varchar(50)	50	NULL allowed
	QID	int	4	NULL allowed
	S_Anwer	int	4	NULL allowed
	C_Answer	int	4	NULL allowed
	Exam_ID	int	4	NULL allowed

Foreign Keys

Name	No Check	Columns
FK_StudenAnswer_Exam	True	Exam_ID->[dbo].[Exam].[E_Id]
FK_StudenAnswer_Question	True	QID->[dbo].[Question].[Q_Id]
FK_StudenAnswer_Question1	True	QID->[dbo].[Question].[Q_Id]

SQL Script

```
CREATE TABLE [dbo].[StudenAnswer]
(
    [S_Id] [int] NULL,
    [Exam_Name] [varchar] (50) COLLATE Arabic_CI_AS NULL,
    [QID] [int] NULL,
    [S_Anwer] [int] NULL,
    [C_Answer] [int] NULL,
    [Exam_ID] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[StudenAnswer] WITH NOCHECK ADD CONSTRAINT [FK_StudenAnswer_Exam] FOREIGN
KEY ([Exam_ID]) REFERENCES [dbo].[Exam] ([E_Id])
GO
```

Project > (local)\MSSQLSERVER01 > User databases > Online-
Examination > Tables > dbo.StudenAnswer

```
ALTER TABLE [dbo].[StudenAnswer] WITH NOCHECK ADD CONSTRAINT [FK_StudenAnswer_Question]
FOREIGN KEY ([QID]) REFERENCES [dbo].[Question] ([Q_Id])
GO
ALTER TABLE [dbo].[StudenAnswer] WITH NOCHECK ADD CONSTRAINT [FK_StudenAnswer_Question1]
FOREIGN KEY ([QID]) REFERENCES [dbo].[Question] ([Q_Id])
GO
```

Uses

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

Used By



[dbo].[GetStudentsByBranchTrackCourseExamToGetGrades]
[dbo].[GetStudentsByInstructorTrackCourseExamToGetGrades]
[dbo].[InsertAnswer]
[dbo].[ModelAnswer]
[dbo].[ModelAnswerEx]

[dbo].[Student]

Properties

Property	Value
Collation	Arabic_CI_AS
Row Count (~)	5
Created	8:53:18 PM Monday, January 27, 2025
Last Modified	8:08:29 PM Sunday, February 9, 2025

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity	Default
	St_Id	int	4	NOT NULL	1 - 1	
	F_Name	varchar(20)	20	NOT NULL		
	L_Name	varchar(20)	20	NOT NULL		
	St_Email	varchar(40)	40	NULL allowed		
	Track_Id	int	4	NULL allowed		
	St_DOB	date	3	NULL allowed		
	St_Address	varchar(40)	40	NULL allowed		
	St_Gender	varchar(6)	6	NULL allowed		
	Role	varchar(50)	50	NULL allowed		('Student')
	Password	varchar(255)	255	NULL allowed		((123456))

Indexes

Key	Name	Key Columns	Unique
	PK__Student__B9034FB32AC96806	St_Id	True

Foreign Keys

Name	No Check	Columns
FK__Student__Track_I__3F466844	True	Track_Id->[dbo].[Track].[T_Id]

SQL Script

```
CREATE TABLE [dbo].[Student]
(
    [St_Id] [int] NOT NULL IDENTITY(1, 1),
    [F_Name] [varchar] (20) COLLATE Arabic_CI_AS NOT NULL,
    [L_Name] [varchar] (20) COLLATE Arabic_CI_AS NOT NULL,
    [St_Email] [varchar] (40) COLLATE Arabic_CI_AS NULL,
```



```
[Track_Id] [int] NULL,  
[St_DOB] [date] NULL,  
[St_Address] [varchar] (40) COLLATE Arabic_CI_AS NULL,  
[St_Gender] [varchar] (6) COLLATE Arabic_CI_AS NULL,  
[Role] [varchar] (50) COLLATE Arabic_CI_AS NULL CONSTRAINT [DF_Student_Role] DEFAULT  
('Student'),  
[Password] [varchar] (255) COLLATE Arabic_CI_AS NULL CONSTRAINT [DF_Student_Password]  
DEFAULT ((123456))  
) ON [PRIMARY]  
GO  
ALTER TABLE [dbo].[Student] ADD CONSTRAINT [PK__Student__B9034FB32AC96806] PRIMARY KEY  
CLUSTERED ([St_Id]) ON [PRIMARY]  
GO  
ALTER TABLE [dbo].[Student] WITH NOCHECK ADD CONSTRAINT [FK__Student__Track_I__3F466844]  
FOREIGN KEY ([Track_Id]) REFERENCES [dbo].[Track] ([T_Id])  
GO
```

Uses

[dbo].[Track]

Used By



[dbo].[Course_Student]
[dbo].[Student_Exam]
[dbo].[Student_Phone]
[dbo].[DeleteStudent]
[dbo].[GetAllExamsAndCoursesAndStudents]
[dbo].[GetAllExamsAndCoursesAndStudentsForSpecificInstructor]
[dbo].[GetAllExamsForSpecificStudentBasedOnTrackID]
[dbo].[GetAllStudentsReportWithCourseId]
[dbo].[GetAllStudentsReportWithTrackId]
[dbo].[GetAllStudentsWithCourses]
[dbo].[GetStudentsByBranchTrackCourse]
[dbo].[GetStudentsByBranchTrackCourseExamToGetGrades]
[dbo].[GetStudentsByInstructorTrackCourseExamToGetGrades]
[dbo].[InsertIntoStudentPhone]
[dbo].[InsertStudent]
[dbo].[SelectAllStudents]
[dbo].[SelectStudentById]
[dbo].[sp_AuthenticateUser]
[dbo].[UpdateStudent]

[dbo].[Student_Exam]

Properties

Property	Value
Row Count (~)	2
Created	9:40:18 AM Saturday, January 18, 2025
Last Modified	8:08:29 PM Sunday, February 9, 2025

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
	St_Id	int	4	NOT NULL
	E_Id	int	4	NOT NULL
	Grade	float	8	NULL allowed

Indexes

Key	Name	Key Columns	Unique
	PK__Student___E47045401DC06A73	St_Id, E_Id	True

Foreign Keys

Name	No Check	Columns
FK__Student_E__St_Id__619B8048	True	St_Id->[dbo].[Student].[St_Id]
FK__Student_Ex__E_Id__628FA481	True	E_Id->[dbo].[Exam].[E_Id]

SQL Script

```
CREATE TABLE [dbo].[Student_Exam]
(
    [St_Id] [int] NOT NULL,
    [E_Id] [int] NOT NULL,
    [Grade] [float] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Student_Exam] ADD CONSTRAINT [PK__Student___E47045401DC06A73] PRIMARY
KEY CLUSTERED ([St_Id], [E_Id]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Student_Exam] WITH NOCHECK ADD CONSTRAINT [FK__Student_E__St_-
Id__619B8048] FOREIGN KEY ([St_Id]) REFERENCES [dbo].[Student] ([St_Id])
GO
```

Project > (local)\MSSQLSERVER01 > User databases > Online-
Examination > Tables > dbo.Student_Exam

```
ALTER TABLE [dbo].[Student_Exam] WITH NOCHECK ADD CONSTRAINT [FK_Student_Ex_E_  
Id__628FA481] FOREIGN KEY ([E_Id]) REFERENCES [dbo].[Exam] ([E_Id])  
GO
```

Uses

[dbo].[Exam]
[dbo].[Student]

Used By


[dbo].[GetStudentsByBranchTrackCourseExamToGetGrades]
[dbo].[GetStudentsByInstructorTrackCourseExamToGetGrades]
[dbo].[InsertStudentExam]

[dbo].[Student_Phone]

Properties

Property	Value
Row Count (~)	5
Created	9:40:18 AM Saturday, January 18, 2025
Last Modified	8:08:29 PM Sunday, February 9, 2025

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
	St_Id	int	4	NOT NULL
	St_Phone	int	4	NOT NULL

Indexes

Key	Name	Key Columns	Unique
	PK__Student__55D7222A382AD58F	St_Id, St_Phone	True

Foreign Keys

Name	No Check	Columns
FK__Student_P__St_Id__5EBF139D	True	St_Id->[dbo].[Student].[St_Id]

SQL Script

```
CREATE TABLE [dbo].[Student_Phone]
(
    [St_Id] [int] NOT NULL,
    [St_Phone] [int] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Student_Phone] ADD CONSTRAINT [PK__Student__55D7222A382AD58F] PRIMARY
KEY CLUSTERED ([St_Id], [St_Phone]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Student_Phone] WITH NOCHECK ADD CONSTRAINT [FK__Student_P__St_-
Id__5EBF139D] FOREIGN KEY ([St_Id]) REFERENCES [dbo].[Student] ([St_Id])
GO
```

Uses

[dbo].[Student]

Used By


[dbo].[deleteStudentPhone]
[dbo].[GetAllStudentsReportWithCourseId]
[dbo].[InsertIntoStudentPhone]
[dbo].[InsertStudent]
[dbo].[online_GetAllStudentsphones]
[dbo].[online_GetStudentPhoneById]
[dbo].[updateStudentPhone]

[dbo].[Track]

Properties

Property	Value
Collation	Arabic_CI_AS
Row Count (~)	7
Created	8:53:58 PM Monday, January 27, 2025
Last Modified	8:08:29 PM Sunday, February 9, 2025

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
	T_Id	int	4	NOT NULL	1 - 1
	T_Name	varchar(100)	100	NOT NULL	

Indexes

Key	Name	Key Columns	Unique
	PK__Track__83BB1F920617E3A7	T_Id	True

SQL Script

```
CREATE TABLE [dbo].[Track]
(
    [T_Id] [int] NOT NULL IDENTITY(1, 1),
    [T_Name] [varchar] (100) COLLATE Arabic_CI_AS NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Track] ADD CONSTRAINT [PK__Track__83BB1F920617E3A7] PRIMARY KEY
CLUSTERED ([T_Id]) ON [PRIMARY]
GO
```

Used By

[dbo].[Branch_Track]
[dbo].[Exam_Track]
[dbo].[Instructor_Track]
[dbo].[Student]
[dbo].[Track_Course]
[dbo].[GetAllTracksByBranchId]
[dbo].[GetAllTracksByInsId]
[dbo].[GetAllTracksWithCourses]
[dbo].[insertTrackCourse]
[dbo].[online_BranchTrack_Insert]

Project > (local)\MSSQLSERVER01 > User databases > Online-
Examaination > Tables > dbo.Track



[dbo].[online_BranchTrack_Update]
[dbo].[online_Track_Delete]
[dbo].[online_Track_Insert]
[dbo].[online_Track_SelectAll]
[dbo].[online_Track_SelectByTrackId]
[dbo].[online_Track_Update]

[dbo].[Track_Course]

Properties

Property	Value
Row Count (~)	4
Created	9:40:18 AM Saturday, January 18, 2025
Last Modified	8:08:29 PM Sunday, February 9, 2025

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
	T_Id	int	4	NOT NULL
	C_Id	int	4	NOT NULL

Indexes

Key	Name	Key Columns	Unique
	PK__Track_Co__B924C1516616183B	T_Id, C_Id	True

Foreign Keys

Name	No Check	Columns
FK__Track_Cour__C_Id__4D94879B	True	C_Id->[dbo].[Course].[C_Id]
FK__Track_Cour__T_Id__4CA06362	True	T_Id->[dbo].[Track].[T_Id]

SQL Script

```
CREATE TABLE [dbo].[Track_Course]
(
    [T_Id] [int] NOT NULL,
    [C_Id] [int] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Track_Course] ADD CONSTRAINT [PK__Track_Co__B924C1516616183B] PRIMARY
KEY CLUSTERED ([T_Id], [C_Id]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Track_Course] WITH NOCHECK ADD CONSTRAINT [FK__Track_Cour__C_-
Id__4D94879B] FOREIGN KEY ([C_Id]) REFERENCES [dbo].[Course] ([C_Id])
GO
ALTER TABLE [dbo].[Track_Course] WITH NOCHECK ADD CONSTRAINT [FK__Track_Cour__T_-
Id__4CA06362] FOREIGN KEY ([T_Id]) REFERENCES [dbo].[Track] ([T_Id])
GO
```


Uses

[dbo].[Course]

[dbo].[Track]

Used By

[dbo].[deletecourse]

[dbo].[deletecoursetrack]

[dbo].[GetAllCoursesWithTracks]

[dbo].[GetAllTracksWithCourses]

[dbo].[GetCoursesByTrackIdAndBranchId]

[dbo].[GetCoursesByTrackIdAndInsId]

[dbo].[GetExamsForSpecificBranchTrackCourse]

[dbo].[GetExamsForSpecificBranchTrackCourseInsId]

[dbo].[GetExamsForSpecificInstructorTrackCourse]

[dbo].[GetStudentsByBranchTrackCourse]

[dbo].[GetStudentsByBranchTrackCourseExamToGetGrades]

[dbo].[GetStudentsByInstructorTrackCourseExamToGetGrades]

[dbo].[insertTrackCourse]

[dbo].[Track_course_SelectByID]

[dbo].[updatecoursetrack]

Stored Procedures

Objects

Name
dbo.add_Question_ToExam
dbo.AddAndInsertQuestion
dbo.AddChoice
dbo.AddQuestionWithChoices
dbo.CalTotalGrade
dbo.CalTotalGradeForExam
dbo.DeleteBranch
dbo.DeleteBranchPhone
dbo.DeleteChoice
dbo.deletecourse
dbo.deletecoursetrack
dbo.DeleteExam
dbo.DeleteInstructor
dbo.DeleteInstructorTeachCourse
dbo.DeleteQuestion
dbo.DeleteStudent
dbo.deleteStudentPhone
dbo.deleteworkon
dbo.GetAllCourses
dbo.GetAllCoursesWithTracks
dbo.GetAllExams
dbo.GetAllExamsAndCoursesAndStudents
dbo.GetAllExamsAndCoursesAndStudentsForSpecificInstructor
dbo.GetAllExamsForSpecificStudentBasedOnTrackId
dbo.GetAllExamsWithDetails
dbo.GetAllInstructors
dbo.GetAllInstructorsWithBranches
dbo.GetAllInstructorsWithCourses
dbo.GetAllStudentsReportWithCourseId
dbo.GetAllStudentsReportWithTrackId
dbo.GetAllStudentsWithCourses
dbo.GetAllTracksByBranchId
dbo.GetAllTracksByInsId
dbo.GetAllTracksWithCourses
dbo.GetCoursesByTrackIdAndBranchId
dbo.GetCoursesByTrackIdAndInsId
dbo.GetExam

dbo.getExamGrade
dbo.GetExamsForSpecificBranchTrackCourse
dbo.GetExamsForSpecificBranchTrackCourseInsId
dbo.GetExamsForSpecificInstructor
dbo.GetExamsForSpecificInstructorTrackCourse
dbo.GetExamWithQuestionsForSpecificInstructor
dbo.GetInstructorCoursesByInsId
dbo.GetQuestion
dbo.GetStudentsByBranchTrackCourse
dbo.GetStudentsByBranchTrackCourseExamToGetGrades
dbo.GetStudentsByInstructorTrackCourseExamToGetGrades
dbo.Insert_ Question
dbo.InsertAnswer
dbo.InsertBranch
dbo.insertBranchInstructor
dbo.InsertBranchPhone
dbo.insertcourse
dbo.InsertExam
dbo.InsertInstructor
dbo.insertInstructorCourse
dbo.InsertInstructorTeachCourse
dbo.InsertIntoStudentPhone
dbo.InsertStudent
dbo.InsertStudentExam
dbo.InsertStudentsWithCourse
dbo.insertTrackCourse
dbo.insertTrackInstructor
dbo.ModelAnswer
dbo.ModelAnswerEx
dbo.online_BranchTrack_Insert
dbo.online_BranchTrack_SelectAll
dbo.online_BranchTrack_SelectByPK
dbo.online_BranchTrack_Update
dbo.online_GetAllStudentsphones
dbo.online_GetStudentPhoneById
dbo.online_Track_Delete
dbo.online_Track_Insert
dbo.online_Track_SelectAll
dbo.online_Track_SelectByTrackId
dbo.online_Track_Update
dbo.online_TrackBranch_Delete
dbo.SelectAllBranches
dbo.SelectAllBranchPhones
dbo.SelectAllInstructorTeachCourse

dbo.SelectAllStudents
dbo.SelectBranchById
dbo.SelectBranchPhoneById
dbo.selectcourse
dbo.selectinstructor
dbo.SelectInstructorTeachCourseById
dbo.SelectStudentById
dbo.selectworkon
dbo.sp_AuthenticateUser
dbo.Track_course_SelectById
dbo.UpdateBranch
dbo.updatecourse
dbo.updatecoursetrack
dbo.UpdateExam
dbo.UpdateInstructor
dbo.UpdateInstructorTeachCourse
dbo.UpdateStudent
dbo.updateStudentPhone
dbo.updateworkon

[dbo].[add_Question_ToExam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
Create Proc [dbo].[add_Question_ToExam] @ExamID int
as
begin
    declare @qID int;
    select @qID=Max(Q_Id) from Question
    insert into Exam_Question values (@ExamID,@qID);
end
GO
```

Uses

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

Used By

[dbo].[AddAndInsertQuestion]
[dbo].[AddQuestionWithChoices]

[dbo].[AddAndInsertQuestion]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Grade	int	4
@Qtype	int	4
@Question	varchar(255)	255
@ModelAnswer	int	4
@ExamID	int	4

SQL Script

```
--5) Add Qestion and Insert into Exam
create Proc [dbo].[AddAndInsertQuestion] @Grade int,@Qtype int, @Question varchar(255)
,@ModelAnswer int,@ExamID int
as
begin
    exec Insert_Question @Grade, @Qtype,@Question,@ModelAnswer
    exec add_Question_ToExam @ExamID
    exec CalTotalGrade @ExamID
end
GO
```

Uses

[dbo].[add_Question_ToExam]
[dbo].[CalTotalGrade]
[dbo].[Insert_Question]

[dbo].[AddChoice]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Qid	int	4
@Q_choice	varchar(50)	50

SQL Script

```
Create Proc [dbo].[AddChoice] @Qid int , @Q_choice varchar(50)
as
begin
    declare @cNumber int;
    select @cNumber=Max(Choice_Num) from Question_Choices where Q_Id=@Qid;
    if (@cNumber is NULL)
        set @cNumber=0;
    set @cNumber+=1;
    insert into Question_Choices values (@Qid,@cNumber,@Q_choice);
end
GO
```

Uses

[dbo].[Question_Choices]

[dbo].[AddQuestionWithChoices]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Grade	int	4
@Qtype	int	4
@Question	varchar(255)	255
@ModelAnswer	int	4
@ExamID	int	4
@Choices	nvarchar(max)	max

SQL Script

```
CREATE PROC [dbo].[AddQuestionWithChoices]
    @Grade INT,
    @Qtype INT,
    @Question VARCHAR(255),
    @ModelAnswer INT,
    @ExamID INT,
    @Choices NVARCHAR(MAX)
AS
BEGIN
    DECLARE @QID INT;

    -- Insert Question
    EXEC Insert_Question @Grade, @Qtype, @Question, @ModelAnswer;

    -- Retrieve the last inserted Question ID
    SELECT @QID = MAX(Q_Id) FROM Question;

    -- Insert the Question into the Exam
    EXEC add_Question_ToExam @ExamID;

    -- Insert Choices if it's a multiple-choice question
    IF @Qtype = 2
    BEGIN
        DECLARE @Choice NVARCHAR(50);
        DECLARE @ChoiceNum INT = 1;
        DECLARE @Pos INT = 0;
        DECLARE @NextPos INT;
```



```
WHILE LEN(@Choices) > 0
BEGIN
    SET @NextPos = CHARINDEX(',', @Choices);
    IF @NextPos = 0
    BEGIN
        SET @Choice = @Choices;
        SET @Choices = '';
    END
    ELSE
    BEGIN
        SET @Choice = LEFT(@Choices, @NextPos - 1);
        SET @Choices = RIGHT(@Choices, LEN(@Choices) - @NextPos);
    END

    -- Insert choice
    INSERT INTO Question_Choices (Q_Id, Choice_Num, answer)
    VALUES (@QID, @ChoiceNum, @Choice);

    SET @ChoiceNum = @ChoiceNum + 1;
END
END
ELSE IF @Qtype = 1
BEGIN
    INSERT INTO Question_Choices (Q_Id, Choice_Num, answer)
    VALUES
        (@QID, 1, 'True'),
        (@QID, 2, 'False');
END

-- Recalculate total exam grade
EXEC CalTotalGrade @ExamID;
END
GO
```

Uses

[dbo].[Question]
[dbo].[Question_Choices]
[dbo].[add_Question_ToExam]
[dbo].[CalTotalGrade]
[dbo].[Insert_Question]

[dbo].[CalTotalGrade]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@exID	varchar(20)	20

SQL Script

```
Create Proc [dbo].[CalTotalGrade] @exID varchar(20)
as
begin
    update Exam
    set E_Total_Score=(select SUM(Q_Grade) from Exam E
    inner join Exam_Question EQ
    on E.E_Id=EQ.E_Id
    inner join Question Q
    on EQ.Q_Id=Q.Q_Id
    where E.E_Id=@exID)
    where E_Id=@exID
end
GO
```

Uses

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

Used By

[dbo].[AddAndInsertQuestion]
[dbo].[AddQuestionWithChoices]
[dbo].[DeleteQuestion]

[dbo].[CalTotalGradeForExam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE    PROCEDURE [dbo].[CalTotalGradeForExam]
    @exID INT
AS
BEGIN
    UPDATE Exam
    SET E_Total_Score = ISNULL((
        SELECT SUM(Q.Q_Grade)
        FROM Exam_Question EQ
        INNER JOIN Question Q ON EQ.Q_Id = Q.Q_Id
        WHERE EQ.E_Id = @exID
    ), 0)
    WHERE E_Id = @exID;
END;
GO
```

Uses

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

Used By

[dbo].[GetStudentsByBranchTrackCourseExamToGetGrades]
[dbo].[GetStudentsByInstructorTrackCourseExamToGetGrades]

[dbo].[DeleteBranch]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE PROCEDURE [dbo].[DeleteBranch] @branchId INT
as
BEGIN
    IF EXISTS (SELECT 1 from Branch WHERE B_Id = @branchId)
    BEGIN
        DELETE FROM Branch WHERE B_Id = @branchId;
    END
    ELSE
    BEGIN
        SELECT 'This branch is Not Found' AS Message;
    END
END;

GO
```

Uses

[dbo].[Branch]

[dbo].[DeleteBranchPhone]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@branchId	int	4
@phone	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[DeleteBranchPhone] @branchId INT, @phone INT
as
BEGIN
    IF EXISTS (SELECT 1 FROM DeleteBranchPhone WHERE B_Id = @branchId AND B_Phone = @phone)
    BEGIN
        DELETE FROM DeleteBranchPhone WHERE B_Id = @branchId AND B_Phone = @phone;
    END
    ELSE
    BEGIN
        SELECT 'This phone is Not Found in this branch' AS Message;
    END
END;
GO
```

[dbo].[DeleteChoice]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Qid	int	4
@cNumber	int	4

SQL Script

```
Create Proc [dbo].[DeleteChoice] @Qid int,@cNumber int
as
begin
    delete from Question_Choices
    where Q_Id=@Qid and Choice_Num=@cNumber
end
GO
```

Uses

[dbo].[Question_Choices]

[dbo].[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].[deletecourse]
@id int
as
begin
if exists(
select s.C_Id from Course_Student s
where s.C_Id=@id
)
begin
print 'Course you want to delete is learned by students...!'
return;
end
if exists(
select s.C_Id from Instructor_Course s
where s.C_Id=@id
)
begin
print 'Course you want to delete is teached by instructors...!'
return;
end
if exists(
select s.C_Id from Track_Course s
where s.C_Id=@id
)
begin
print 'Course you want to delete exist in track...!'
return;
end
delete from Course
where C_Id=@id
end
GO
```

Project > (local)\MSSQLSERVER01 > User databases > Online-
Examination > Programmability > Stored Procedures >
dbo.deletecourse

Uses

[dbo].[Course]
[dbo].[Course_Student]
[dbo].[Instructor_Course]
[dbo].[Track_Course]

[dbo].[deletecoursetrack]

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].[deletecoursetrack]
@id int
as
begin
if not exists(
select t.C_Id from Track_Course t
where t.C_Id=@id
)
begin
print 'course not exist ...!'
return;
end
delete from Track_Course
where C_Id=@id
end
GO
```

Uses

[dbo].[Track_Course]

[dbo].[DeleteExam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
Create PROC [dbo].[DeleteExam]
(
    @E_Id INT
)
AS
BEGIN
    -- If you have dependent records in Exam_Question or other tables,
    -- either delete them first or rely on cascading constraints.

    DELETE FROM Exam
    WHERE E_Id = @E_Id;
END;
GO
```

Uses

[dbo].[Exam]

[dbo].[DeleteInstructor]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
-----  
---- Delete  
  
CREATE    PROCEDURE [dbo].[DeleteInstructor]  
(  
    @Ins_Id INT  
)  
AS  
BEGIN  
    DELETE FROM Instructor  
    WHERE Ins_Id = @Ins_Id;  
END;  
GO
```

Uses

[dbo].[Instructor]

[dbo].[DeleteInstructorTeachCourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE PROCEDURE [dbo].[DeleteInstructorTeachCourse] @instructorId INT, @courseId INT
as
BEGIN
    IF EXISTS (SELECT 1 FROM Instructor_Course WHERE Ins_Id = @instructorId AND C_Id =
@courseId)
    BEGIN
        DELETE FROM Instructor_Course WHERE Ins_Id = @instructorId AND C_Id = @courseId;
    END
    ELSE
    BEGIN
        SELECT 'This instructor who give that course is Not Found' AS Message;
    END

END;
GO
```

Uses

[dbo].[Instructor_Course]

[dbo].[DeleteQuestion]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Qid	int	4
@ExamID	int	4

SQL Script

```
Create Proc [dbo].[DeleteQuestion] @Qid int,@ExamID int
as
begin
    delete from Question_Choices where Q_Id=@Qid

    delete from Exam_Question where Q_Id=@Qid

    delete from Question where Q_Id=@Qid

    exec CalTotalGrade @ExamID
end
GO
```

Uses

[dbo].[Exam_Question]
[dbo].[Question]
[dbo].[Question_Choices]
[dbo].[CalTotalGrade]

[dbo].[DeleteStudent]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE PROCEDURE [dbo].[DeleteStudent] @studentId INT
as
BEGIN
    IF EXISTS (SELECT 1 FROM Student WHERE St_Id = @studentId)
    BEGIN
        DELETE FROM Student WHERE St_Id = @studentId;
    END
    ELSE
    BEGIN
        SELECT 'This student is Not Found' AS Message;
    END
END;
GO
```

Uses

[dbo].[Student]

[dbo].[deleteStudentPhone]

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].[deleteStudentPhone]
    @id INT
AS
BEGIN
    IF @id IS NOT NULL AND EXISTS (SELECT 1 FROM student_phone WHERE st_id = @id)
    BEGIN
        DELETE FROM student_phone
        WHERE st_id = @id;

        SELECT 'Phone number deleted successfully.' AS ResultMessage;
    END
    ELSE
    BEGIN
        SELECT 'Cannot delete invalid ID.' AS ResultMessage;
    END
END;
GO
```

Uses

[dbo].[Student_Phone]

Project > (local)\MSSQLSERVER01 > User databases > Online-
Examination > Programmability > Stored Procedures >
dbo.deleteworkon

 [dbo].[deleteworkon]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
/*delete*/  
create proc [dbo].[deleteworkon]  
@id int  
as  
begin  
delete from Track_Instructor  
where Ins_Id=@id;  
end  
GO
```


[dbo].[GetAllCourses]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
/*#####*/  
/*Course*/  
-- Get All Courses  
create proc [dbo].[GetAllCourses]  
as  
begin  
    select C_Name,C_Duration from Course  
End  
GO
```

Uses

[dbo].[Course]

[dbo].[GetAllCoursesWithTracks]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE    PROCEDURE [dbo].[GetAllCoursesWithTracks]
AS
BEGIN
    SELECT
        C.C_Name,
        C.C_Duration,
        C.C_Id,
        TC.T_Id
    FROM Course C
    LEFT JOIN Track_Course TC ON C.C_Id = TC.C_Id;
END;
GO
```

Uses

[dbo].[Course]
[dbo].[Track_Course]

[dbo].[GetAllExams]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
Create Proc [dbo].[GetAllExams]
as
begin
    select E_Id,E_Name from Exam
end
GO
```

Uses

[dbo].[Exam]

[dbo].[GetAllExamsAndCoursesAndStudents]

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].[GetAllExamsAndCoursesAndStudents]
    @ins_id INT
AS
BEGIN
    SELECT
        (SELECT COUNT(E_Id) FROM Exam) AS [Total Exams],
        (SELECT COUNT(St_Id) FROM Student) AS [Total Students],
        (SELECT COUNT(C_Id) FROM Course) AS [Total Courses]
END;
GO
```

Uses

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

[dbo].[GetAllExamsAndCoursesAndStudentsForSpecificInstructor]

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].[GetAllExamsAndCoursesAndStudentsForSpecificInstructor]
    @ins_id INT
AS
BEGIN
    SELECT
        -- ☒ Count Exams related to the Instructor
        (
            SELECT COUNT(E.E_Id)
            FROM Exam E
            JOIN Instructor_EExam IE ON E.E_Id = IE.E_Id
            WHERE IE.Ins_Id = @ins_id
        ) AS [Total Exams],

        -- ☒ Count Students related to the Instructor's Courses
        (
            SELECT COUNT(DISTINCT S.St_Id)
            FROM Student S
            JOIN Course_Student CS ON S.St_Id = CS.St_Id
            JOIN Instructor_Course IC ON CS.C_Id = IC.C_Id
            WHERE IC.Ins_Id = @ins_id
        ) AS [Total Students],

        -- ☒ Count Courses assigned to the Instructor
        (
            SELECT COUNT(DISTINCT C.C_Id)
            FROM Course C
            JOIN Instructor_Course IC ON C.C_Id = IC.C_Id
            WHERE IC.Ins_Id = @ins_id
        ) AS [Total Courses],

        -- ☒ Count Questions in Exams assigned to the Instructor
        (
            SELECT COUNT(DISTINCT Q.Q_Id)
```

Project > (local)\MSSQLSERVER01 > User databases > Online-
Examination > Programmability > Stored Procedures > dbo.GetAll-
ExamsAndCoursesAndStudentsForSpecificInstructor

```
FROM Question Q
JOIN Exam_Question EQ ON Q.Q_Id = EQ.Q_Id
JOIN Instructor_EExam IE ON EQ.E_Id = IE.E_Id
WHERE IE.Ins_Id = @ins_id
) AS [Total Questions];
END;
GO
```

Uses

[dbo].[Course]
[dbo].[Course_Student]
[dbo].[Exam]
[dbo].[Exam_Question]
[dbo].[Instructor_Course]
[dbo].[Instructor_EExam]
[dbo].[Question]
[dbo].[Student]

[dbo].[GetAllExamsForSpecificStudentBasedOnTrackID]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE PROC [dbo].[GetAllExamsForSpecificStudentBasedOnTrackID]
    @St_Id INT
AS
BEGIN

    DECLARE @TrackId INT;
    SELECT @TrackId = Track_Id
    FROM Student
    WHERE St_Id = @St_Id;

    SELECT E.*
    FROM Exam E
    JOIN Exam_Track ET ON ET.E_Id = E.E_Id
    JOIN Student S ON S.St_Id = @St_Id
    WHERE S.Track_Id = @TrackId;

END;
GO
```

Uses

[dbo].[Exam]
[dbo].[Exam_Track]
[dbo].[Student]

[dbo].[GetAllExamsWithDetails]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
Create    PROC [dbo].[GetAllExamsWithDetails]
AS
BEGIN
    Select E_Name,E_Total_Score,E_Duration,E_Date from Exam

END;
GO
```

Uses

[dbo].[Exam]

[dbo].[GetAllInstructors]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE    PROCEDURE [dbo].[GetAllInstructors]
AS
BEGIN
    SELECT
        I.Ins_Id,
        I.F_Name,
        I.L_Name,
        I.Ins_Age,
        I.Ins_Email,
        IPh.Ins_Phone,
        I.Ins_Address,
        I.Ins_Gender,
        I.Ins_DOB,
        I.Ins_Salary
    FROM Instructor I
    Join Instructor_Phone IPh ON IPh.Ins_Id = I.Ins_Id
END;
GO
```

Uses

[dbo].[Instructor]
[dbo].[Instructor_Phone]

[dbo].[GetAllInstructorsWithBranches]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE    PROCEDURE [dbo].[GetAllInstructorsWithBranches]
AS
BEGIN
    SELECT
        I.Ins_Id,
        I.F_Name,
        I.L_Name,
        B.B_Name AS [Branch Name]
    FROM Instructor I
    -- Joining with Instructor_Course table
    INNER JOIN Branch_Instructor IB ON I.Ins_Id = IB.Ins_Id
    -- Joining with Course table
    INNER JOIN Branch B ON IB.B_Id = B.B_Id;
END;
GO
```

Uses

[dbo].[Branch]
[dbo].[Branch_Instructor]
[dbo].[Instructor]

[dbo].[GetAllInstructorsWithCourses]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE PROCEDURE [dbo].[GetAllInstructorsWithCourses]
AS
BEGIN
    SELECT
        I.Ins_Id,
        I.F_Name,
        I.L_Name,
        C.C_Name AS Course_Name
    FROM Instructor I
    -- Using LEFT JOIN to include all instructors, even those without courses
    LEFT JOIN Instructor_Course IC ON I.Ins_Id = IC.Ins_Id
    LEFT JOIN Course C ON IC.C_Id = C.C_Id;
END;
GO
```

Uses

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

[dbo].[GetAllStudentsReportWithCourseId]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE    PROCEDURE [dbo].[GetAllStudentsReportWithCourseId]
    @C_id INT,
    @Ins_id INT
AS
BEGIN
    SELECT
        S.St_Id,
        S.F_Name,
        S.L_Name,
        S.St_Email,
        SP.St_Phone,
        C.C_Name,
        C.C_Duration
    FROM Student S
    JOIN Course_Student CS ON CS.St_Id = S.St_Id
    JOIN Course C ON CS.C_Id = C.C_Id
    JOIN Instructor_Course IC ON IC.C_Id = C.C_Id
    Join Student_Phone SP ON SP.St_Id = S.St_Id
    WHERE C.C_Id = @C_id AND IC.Ins_Id = @Ins_id;
END;
GO
```

Uses

[dbo].[Course]
[dbo].[Course_Student]
[dbo].[Instructor_Course]
[dbo].[Student]
[dbo].[Student_Phone]

[dbo].[GetAllStudentsReportWithTrackId]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
create proc [dbo].[GetAllStudentsReportWithTrackId] @T_id int
as
    select *
    from Student
    where Track_Id =@T_id
GO
```

Uses

[dbo].[Student]

[dbo].[GetAllStudentsWithCourses]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE    PROCEDURE [dbo].[GetAllStudentsWithCourses]
AS
BEGIN
    SELECT
        S.St_Id,
        S.F_Name,
        S.L_Name,
        C.C_Name AS [Course Name]
    FROM Student S
    -- LEFT JOIN ensures students with no courses are also included
    LEFT JOIN Course_Student CS ON S.St_Id = CS.St_Id
    LEFT JOIN Course C ON CS.C_Id = C.C_Id;
END;
GO
```

Uses

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

[dbo].[GetAllTracksByBranchId]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
-----  
  
--GetAllTracksByBranchId --> Testing : done  
CREATE    PROCEDURE [dbo].[GetAllTracksByBranchId]  
    @B_id INT  
AS  
BEGIN  
    SELECT  
        *  
    FROM Track T  
    INNER JOIN Branch_Track BT ON T.T_Id = BT.T_Id  
    INNER JOIN Branch B ON B.B_Id = BT.B_Id  
    WHERE BT.B_Id = @B_id;  
END;  
GO
```

Uses

[dbo].[Branch]
[dbo].[Branch_Track]
[dbo].[Track]

[dbo].[GetAllTracksByInsId]

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].[GetAllTracksByInsId]
    @Ins_id INT
AS
BEGIN
    SELECT
        *
    FROM Track T
    INNER JOIN Instructor_Track IT ON T.T_Id = IT.T_Id
    INNER JOIN Instructor I ON I.Ins_id = IT.Ins_Id
    WHERE IT.Ins_Id = @Ins_id;
END;
GO
```

Uses

[dbo].[Instructor]
[dbo].[Instructor_Track]
[dbo].[Track]

[dbo].[GetAllTracksWithCourses]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
-----  
  
CREATE    PROCEDURE [dbo].[GetAllTracksWithCourses]  
AS  
BEGIN  
    SELECT  
        T.T_Name,  
        T.T_Id,  
        TC.C_Id  
    FROM Track T  
    LEFT JOIN Track_Course TC ON T.T_Id = TC.T_Id;  
END;  
GO
```

Uses

[dbo].[Track]
[dbo].[Track_Course]

[dbo].[GetCoursesByTrackIdAndBranchId]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@B_Id	int	4
@T_Id	int	4

SQL Script

```
CREATE    PROCEDURE [dbo].[GetCoursesByTrackIdAndBranchId]
    @B_Id INT,
    @T_Id INT
AS
BEGIN
    SELECT
        C.C_Id,
        C.C_Name,
        C.C_Duration
    FROM Course C
    INNER JOIN Track_Course TC ON TC.C_Id = C.C_Id
    INNER JOIN Branch_Track BT ON BT.T_Id = TC.T_Id -- Corrected JOIN condition
    WHERE BT.B_Id = @B_Id AND TC.T_Id = @T_Id; -- Ensure both conditions are applied
END;
GO
```

Uses

[dbo].[Branch_Track]
[dbo].[Course]
[dbo].[Track_Course]

[dbo].[GetCoursesByTrackIdAndInsId]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Ins_Id	int	4
@T_Id	int	4

SQL Script

```
CREATE    PROCEDURE [dbo].[GetCoursesByTrackIdAndInsId]
    @Ins_Id INT,
    @T_Id INT
AS
BEGIN
    SELECT
        C.C_Id,
        C.C_Name,
        C.C_Duration
    FROM Course C
    INNER JOIN Track_Course TC ON TC.C_Id = C.C_Id
    INNER JOIN Instructor_Track IT ON IT.T_Id = TC.T_Id
    WHERE IT.Ins_Id = @Ins_Id AND TC.T_Id = @T_Id;
END;
GO
```

Uses

[dbo].[Course]
[dbo].[Instructor_Track]
[dbo].[Track_Course]

[dbo].[GetExam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@eName	varchar(20)	20

SQL Script

```
Create Proc [dbo].[GetExam] @eName varchar(20)
as
begin
    select * from Exam E
    inner join Exam_Question EQ
    on E.E_Id= EQ.E_Id
    inner join Question Q
    on EQ.Q_Id=Q.Q_Id
    where E.E_Name=@eName
    order by newid()
end
GO
```

Uses

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

[dbo].[getExamGrade]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@eName	varchar(20)	20

SQL Script

```
Create Proc [dbo].[getExamGrade] @eName varchar(20)
as
begin
    select E_Total_Score from Exam
    where E_Name=@eName;
end
GO
```

Uses

[dbo].[Exam]

[dbo].[GetExamsForSpecificBranchTrackCourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@B_id	int	4
@T_id	int	4
@C_id	int	4

SQL Script

```
-----  
CREATE    PROCEDURE [dbo].[GetExamsForSpecificBranchTrackCourse]  
    @B_id INT,  
    @T_id INT,  
    @C_id INT  
AS  
BEGIN  
    SELECT  
        E.E_Id,  
        E.E_Name,  
        E.E_Date,  
        E.E_Total_Score  
    FROM Exam E  
    INNER JOIN Course_Exam CE ON E.E_Id = CE.E_Id  
    INNER JOIN Track_Course TC ON TC.C_Id = CE.C_Id  
    INNER JOIN Branch_Track BT ON BT.T_Id = TC.T_Id  
    WHERE BT.B_Id = @B_id  
        AND TC.T_Id = @T_id  
        AND CE.C_Id = @C_id;  
  
END;  
GO
```

Uses

[dbo].[Branch_Track]
[dbo].[Course_Exam]
[dbo].[Exam]
[dbo].[Track_Course]

[dbo].[GetExamsForSpecificBranchTrackCourseInsId]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE    PROCEDURE [dbo].[GetExamsForSpecificBranchTrackCourseInsId]
    @Ins_id INT,
    @T_id INT,
    @C_id INT
AS
BEGIN
    SELECT
        E.E_Id,
        E.E_Name,
        E.E_Date,
        E.E_Total_Score
    FROM Exam E
    INNER JOIN Course_Exam CE ON E.E_Id = CE.E_Id
    INNER JOIN Track_Course TC ON TC.C_Id = CE.C_Id
    INNER JOIN Instructor_Track IT ON IT.T_Id = TC.T_Id
    WHERE IT.Ins_Id = @Ins_id
        AND TC.T_Id = @T_id
        AND CE.C_Id = @C_id;

END;
GO
```

Uses

[dbo].[Course_Exam]
[dbo].[Exam]
[dbo].[Instructor_Track]
[dbo].[Track_Course]

[dbo].[GetExamsForSpecificInstructor]

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].[GetExamsForSpecificInstructor]
    @ins_id INT
AS
BEGIN
    SELECT E.E_Id, E.E_Name,E_Duration,E_Total_Score,E_Date
    FROM Exam E
    INNER JOIN Instructor_EExam IE ON E.E_Id = IE.E_Id
    WHERE IE.Ins_Id = @ins_id;
END;
GO
```

Uses

[dbo].[Exam]
[dbo].[Instructor_EExam]

[dbo].[GetExamsForSpecificInstructorTrackCourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@T_id	int	4
@C_id	int	4

SQL Script

```
CREATE    PROCEDURE [dbo].[GetExamsForSpecificInstructorTrackCourse]
    @T_id INT,
    @C_id INT
AS
BEGIN

SELECT
    E.*,
    TC.*
FROM Exam E
INNER JOIN Course_Exam CE ON E.E_Id = CE.E_Id
INNER JOIN Track_Course TC ON TC.C_Id = CE.C_Id
INNER JOIN Instructor_EExam IE ON E.E_Id = IE.E_Id
INNER JOIN Instructor_Track IT ON IT.Ins_Id = IE.Ins_Id AND IT.T_Id = @T_id
where TC.C_Id = @C_id

END;
GO
```

Uses

[dbo].[Course_Exam]
[dbo].[Exam]
[dbo].[Instructor_EExam]
[dbo].[Instructor_Track]
[dbo].[Track_Course]

[dbo].[GetExamWithQuestionsForSpecificInstructor]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE    PROCEDURE [dbo].[GetExamWithQuestionsForSpecificInstructor]
    @E_id INT
AS
BEGIN
    SELECT *
    FROM Exam E
    INNER JOIN Exam_Question EQ ON E.E_Id = EQ.E_Id
    INNER JOIN Question Q ON EQ.Q_Id = Q.Q_Id
    WHERE E.E_Id = @E_id
    ORDER BY NEWID();
END;
GO
```

Uses

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

[dbo].[GetInstructorCoursesByInsId]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Ins_Id	int	4
@T_Id	int	4

SQL Script

```
CREATE    PROCEDURE [dbo].[GetInstructorCoursesByInsId]
    @Ins_Id INT,
    @T_Id INT
AS
BEGIN
    SELECT C.C_Id, C.C_Name
    FROM Course C
    INNER JOIN Instructor_Course IC ON C.C_Id = IC.C_Id
    INNER JOIN Instructor_Track IT ON IC.Ins_Id = IT.Ins_Id
    WHERE IC.Ins_Id = @Ins_Id
    AND IT.T_Id = @T_Id; -- Filtering by track
END;
GO
```

Uses

[dbo].[Course]
[dbo].[Instructor_Course]
[dbo].[Instructor_Track]

[dbo].[GetQuestion]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
Create Proc [dbo].[GetQuestion] @q_ID int
as
begin
    select * from Question Q
    inner join Question_Choices QC
    on Q.Q_Id=QC.Q_Id
    where Q.Q_Id=@q_ID
end
GO
```

Uses

[dbo].[Question]
[dbo].[Question_Choices]

[dbo].[GetStudentsByBranchTrackCourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@B_id	int	4
@T_id	int	4
@C_id	int	4

SQL Script

```
CREATE    PROCEDURE [dbo].[GetStudentsByBranchTrackCourse]
    @B_id INT,
    @T_id INT,
    @C_id INT
AS
BEGIN
    SELECT
        S.ST_Id,
        S.F_Name,
        S.L_Name,
        S.ST_Email
    FROM Student S
    INNER JOIN Course_Student CS ON CS.St_Id = S.St_Id
    INNER JOIN Track_Course TC ON TC.C_Id = CS.C_Id
    INNER JOIN Branch_Track BT ON BT.T_Id = TC.T_Id
    WHERE BT.B_Id = @B_id
        AND TC.T_Id = @T_id
        AND TC.C_Id = @C_id;
END;
GO
```

Uses

[dbo].[Branch_Track]
[dbo].[Course_Student]
[dbo].[Student]
[dbo].[Track_Course]

[dbo].[GetStudentsByBranchTrackCourseExamToGetGrades]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@B_id	int	4
@T_id	int	4
@C_id	int	4
@E_id	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[GetStudentsByBranchTrackCourseExamToGetGrades]
    @B_id INT,
    @T_id INT,
    @C_id INT,
    @E_id INT
AS
BEGIN
    -- First, update the total exam score using CalTotalGrade procedure
    EXEC CalTotalGradeForExam @E_id;

    -- Retrieve student grades for the given exam
    SELECT
        S.ST_Id,
        S.F_Name,
        S.L_Name,
        E.E_Name,
        -- Use COALESCE to handle NULL Total Score
        COALESCE(E.E_Total_Score, 0) AS E_Total_Score,
        -- Use COALESCE to handle NULL Final Grade
        COALESCE(SUM(
            CASE
                WHEN SA.S_Answer = SA.C_Answer THEN Q.Q_Grade
                ELSE 0
            END
        ), 0) AS Final_Grade
    FROM Student S
    LEFT JOIN Course_Student CS ON CS.St_Id = S.St_Id
    LEFT JOIN Track_Course TC ON TC.C_Id = CS.C_Id
    LEFT JOIN Branch_Track BT ON BT.T_Id = TC.T_Id
    LEFT JOIN Student_Exam SE ON SE.St_Id = S.St_Id AND SE.E_Id = @E_id
```

Project > (local)\MSSQLSERVER01 > User databases > Online-
Examination > Programmability > Stored Procedures > dbo.Get-
StudentsByBranchTrackCourseExamToGetGrades

```
LEFT JOIN Exam E ON E.E_Id = @E_id
LEFT JOIN StudenAnswer SA ON SA.Exam_ID = @E_id AND SA.S_Id = S.St_Id
LEFT JOIN Question Q ON Q.Q_Id = SA.QID
WHERE BT.B_Id = @B_id
      AND TC.T_Id = @T_id
      AND TC.C_Id = @C_id
GROUP BY S.ST_Id, S.F_Name, S.L_Name, E.E_Name, E.E_Total_Score;
END;
GO
```

Uses

[dbo].[Branch_Track]
[dbo].[Course_Student]
[dbo].[Exam]
[dbo].[Question]
[dbo].[StudenAnswer]
[dbo].[Student]
[dbo].[Student_Exam]
[dbo].[Track_Course]
[dbo].[CalTotalGradeForExam]

[dbo].[GetStudentsByInstructorTrackCourseExamToGetGrades]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
-----  
CREATE PROCEDURE [dbo].[GetStudentsByInstructorTrackCourseExamToGetGrades]  
    @E_id INT  
AS  
BEGIN  
  
    EXEC CalTotalGradeForExam @E_id;  
  
    Select Top 1  
    S.ST_Id,  
    S.F_Name,  
    S.L_Name,  
    E.E_Name,  
    COALESCE(E.E_Total_Score, 0) AS E_Total_Score,  
    COALESCE(SE.Grade, 0) AS Final_Grade  
    from Student s  
    INNER JOIN Course_Student CS ON CS.St_Id = S.St_Id  
    INNER JOIN Track_Course TC ON TC.C_Id = CS.C_Id  
    INNER JOIN Student_Exam SE ON SE.St_Id = S.St_Id AND SE.E_Id = @E_id  
    INNER JOIN StudenAnswer SA ON SA.Exam_ID = @E_id AND SA.S_Id = S.St_Id  
    INNER JOIN Exam E ON E.E_Id = @E_id  
    INNER JOIN Question Q ON Q.Q_Id = SA.QID  
  
END;  
GO
```

Uses

[dbo].[Course_Student]
[dbo].[Exam]

Project > (local)\MSSQLSERVER01 > User databases > Online-
Examination > Programmability > Stored Procedures > dbo.Get-
StudentsByInstructorTrackCourseExamToGetGrades

[dbo].[Question]
[dbo].[StudenAnswer]
[dbo].[Student]
[dbo].[Student_Exam]
[dbo].[Track_Course]
[dbo].[CalTotalGradeForExam]

[dbo].[Insert_Question]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Grade	int	4
@Qtype	int	4
@Question	varchar(255)	255
@ModelAnswer	int	4

SQL Script

```
Create Proc [dbo].[Insert_Question] @Grade int,@Qtype int, @Question varchar(255)
,@ModelAnswer int
as
begin
declare @tye varchar(20);
if (@Qtype=1)
set @tye='T/F'
else
set @tye='Choices'

insert into Question
values (@Grade,@tye,@Question,@ModelAnswer);
end
GO
```

Uses

[dbo].[Question]

Used By

[dbo].[AddAndInsertQuestion]
[dbo].[AddQuestionWithChoices]

[dbo].[InsertAnswer]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@s_ID	int	4
@E_Name	varchar(50)	50
@Q_ID	int	4
@S_Ans	int	4
@C_ID	int	4
@E_ID	int	4

SQL Script

```
Create Proc [dbo].[InsertAnswer] @s_ID int,@E_Name varchar(50),@Q_ID int,@S_Ans int,@C_ID
int,@E_ID int
as
begin
insert into StudenAnswer
values (@s_ID,@E_Name,@Q_ID,@S_Ans,@C_ID,@E_ID);
end
GO
```

Uses

[dbo].[StudenAnswer]

[dbo].[InsertBranch]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@name	varchar(20)	20
@location	varchar(20)	20
@phone	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[InsertBranch] @name VARCHAR(20), @location VARCHAR(20), @phone INT
as
BEGIN
    INSERT INTO Branch (B_Name, B_Location, B_Phone) VALUES (@name, @location, @phone);
END;
GO
```

Uses

[dbo].[Branch]

[dbo].[insertBranchInstructor]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@B_Id	int	4
@Ins_Id	int	4

SQL Script

```
CREATE    PROCEDURE [dbo].[insertBranchInstructor]
    @B_Id INT,
    @Ins_Id INT
AS
BEGIN
    BEGIN TRY

        INSERT INTO Branch_Instructor (B_Id, Ins_Id)
        VALUES (@B_Id, @Ins_Id)

        PRINT 'Instructor successfully assigned to branch.'
    END TRY
    BEGIN CATCH

        PRINT 'Error: ' + ERROR_MESSAGE()
    END CATCH
END
GO
```

Uses

[dbo].[Branch_Instructor]

[dbo].[InsertBranchPhone]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@branchId	int	4
@phone	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[InsertBranchPhone] @branchId INT, @phone INT
as
BEGIN
    INSERT INTO Branch_Phone (B_Id, B_Phone) VALUES (@branchId, @phone);
END;

GO
```

[dbo].[insertcourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@C_Name	varchar(50)	50
@C_Duration	varchar(50)	50

SQL Script

```
CREATE    PROCEDURE [dbo].[insertcourse]
    @C_Name VARCHAR(50),
    @C_Duration VARCHAR(50)
AS
BEGIN
    -- Check if the course name already exists
    IF EXISTS (SELECT 1 FROM Course c WHERE c.C_Name = @C_Name)
    BEGIN
        PRINT 'This course name already exists...!'
        RETURN;
    END

    -- Insert the new course
    INSERT INTO Course (C_Name, C_Duration)
    VALUES (@C_Name, @C_Duration)
END
GO
```

Uses

[dbo].[Course]

[dbo].[InsertExam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@E_Name	varchar(20)	20
@E_Total_Score	int	4
@E_Duration	float	8
@E_Date	date	3
@Ins_Id	int	4
@C_Id	int	4
@T_Id	int	4

SQL Script

```
CREATE    PROCEDURE [dbo].[InsertExam]
(
    @E_Name      VARCHAR(20),
    @E_Total_Score INT,
    @E_Duration   FLOAT,
    @E_Date       DATE,
    @Ins_Id       INT ,
    @C_Id         INT,
    @T_Id         INT
)
AS
BEGIN
    DECLARE @NewExamID INT;

    INSERT INTO Exam (E_Name, E_Total_Score, E_Duration, E_Date)
    VALUES (@E_Name, @E_Total_Score, @E_Duration, @E_Date);

    SET @NewExamID = SCOPE_IDENTITY();

    INSERT INTO Instructor_EExam (Ins_Id, E_Id)
    VALUES (@Ins_Id, @NewExamID);

    INSERT INTO Course_Exam
    VALUES (@C_Id, @NewExamID);
```


Project > (local)\MSSQLSERVER01 > User databases > Online-
Examination > Programmability > Stored Procedures > dbo.Insert-
Exam

```
INSERT INTO Exam_Track
VALUES (@NewExamID,@T_Id);

SELECT @NewExamID AS NewExamID;
END;
GO
```

Uses

[dbo].[Course_Exam]
[dbo].[Exam]
[dbo].[Exam_Track]
[dbo].[Instructor_EExam]

[dbo].[InsertInstructor]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@F_Name	varchar(20)	20
@L_Name	varchar(20)	20
@Ins_Age	int	4
@Ins_Email	varchar(30)	30
@Ins_Address	varchar(50)	50
@Ins_Gender	varchar(6)	6
@Ins_DOB	date	3
@Ins_Salary	decimal(10,2)	9
@B_id	int	4
@Phone	int	4
@T_Id	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[InsertInstructor]
(
    @F_Name    VARCHAR(20),
    @L_Name    VARCHAR(20),
    @Ins_Age   INT,
    @Ins_Email VARCHAR(30),
    @Ins_Address VARCHAR(50),
    @Ins_Gender VARCHAR(6),
    @Ins_DOB   DATE,
    @Ins_Salary DECIMAL(10,2),
    @B_id      INT,
    @Phone     INT,
    @T_Id      INT
)
AS
BEGIN
    DECLARE @NewInstructorID INT;

    INSERT INTO Instructor
    (
        F_Name,
```

```
        L_Name,  
        Ins_Age,  
        Ins_Email,  
        Ins_Address,  
        Ins_Gender,  
        Ins_DOB,  
        Ins_Salary  
    )  
VALUES  
(  
    @F_Name,  
    @L_Name,  
    @Ins_Age,  
    @Ins_Email,  
    @Ins_Address,  
    @Ins_Gender,  
    @Ins_DOB,  
    @Ins_Salary  
);  
  
SET @NewInstructorID = SCOPE_IDENTITY(); -- Get the new Instructor ID  
  
-- Insert into Branch_Instructor (Assuming columns are Branch_Id and Instructor_Id)  
INSERT INTO Branch_Instructor (B_Id, Ins_Id)  
VALUES (@B_id, @NewInstructorID);  
  
-- Insert into Instructor_Phone (Assuming columns are Ins_Id and Phone)  
INSERT INTO Instructor_Phone  
VALUES (@NewInstructorID, @Phone);  
  
INSERT INTO Instructor_Track  
VALUES (@NewInstructorID, @T_Id);  
  
-- Return the new Instructor ID  
SELECT @NewInstructorID AS NewInstructorID;  
END;  
GO
```

Uses

[dbo].[Branch_Instructor]
[dbo].[Instructor]
[dbo].[Instructor_Phone]
[dbo].[Instructor_Track]

[dbo].[insertInstructorCourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Ins_Id	int	4
@C_Id	int	4

SQL Script

```
-- Instructor Course

CREATE PROCEDURE [dbo].[insertInstructorCourse]
@Ins_Id INT,
  @C_Id INT
AS
BEGIN
    BEGIN TRY
        -- Insert instructor into the Track
        INSERT INTO Instructor_Course (Ins_Id, C_Id)
        VALUES (@Ins_Id, @C_Id)

        PRINT 'Instructor successfully assigned to Course.'
    END TRY
    BEGIN CATCH

        PRINT 'Error: ' + ERROR_MESSAGE()
    END CATCH
END
GO
```

Uses

[dbo].[Instructor_Course]

[dbo].[InsertInstructorTeachCourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE PROCEDURE [dbo].[InsertInstructorTeachCourse] @instructorId INT, @courseId INT
as
BEGIN
    INSERT INTO Instructor_Course (Ins_Id, C_Id) VALUES (@instructorId, @courseId);
END;
GO
```

Uses

[dbo].[Instructor_Course]

[dbo].[InsertIntoStudentPhone]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE PROCEDURE [dbo].[InsertIntoStudentPhone]
    @phone INT,
    @id INT
AS
BEGIN
    IF @id IS NOT NULL AND EXISTS (SELECT 1 FROM student WHERE st_id = @id)
    BEGIN
        INSERT INTO student_phone (st_id, St_Phone)
        VALUES (@id, @phone);

        SELECT 'Phone number added successfully.' AS Message;
    END
    ELSE
    BEGIN
        SELECT 'This Student Does Not Exist.' AS Message;
    END
END;
GO
```

Uses

[dbo].[Student]
[dbo].[Student_Phone]

[dbo].[InsertStudent]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@fName	varchar(20)	20
@lName	varchar(20)	20
@email	varchar(40)	40
@trackId	int	4
@dob	date	3
@address	varchar(40)	40
@gender	varchar(6)	6
@Phone	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[InsertStudent]
    @fName VARCHAR(20),
    @lName VARCHAR(20),
    @email VARCHAR(40),
    @trackId INT,
    @dob DATE,
    @address VARCHAR(40),
    @gender VARCHAR(6),
    @Phone INT
AS
BEGIN
    DECLARE @NewStudentID INT; -- Declare variable for new student ID

    -- Insert student record
    INSERT INTO Student (F_Name, L_Name, St_Email, Track_Id, St_DOB, St_Address, St_Gender)
    VALUES (@fName, @lName, @email, @trackId, @dob, @address, @gender);

    -- Get the newly inserted student ID
    SET @NewStudentID = SCOPE_IDENTITY();

    -- Insert student phone
    INSERT INTO Student_Phone
    VALUES (@NewStudentID, @Phone);

    -- Return the new Student ID
```

Project > (local)\MSSQLSERVER01 > User databases > Online-
Examination > Programmability > Stored Procedures > dbo.Insert-
Student

```
SELECT @NewStudentID AS NewStudentID;  
END;  
GO
```

Uses

[dbo].[Student]

[dbo].[Student_Phone]

[dbo].[InsertStudentExam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@s_ID	int	4
@E_ID	int	4
@Grade	float	8

SQL Script

```
CREATE    PROCEDURE [dbo].[InsertStudentExam]
    @s_ID INT,
    @E_ID INT,
    @Grade float
AS
BEGIN

    INSERT INTO Student_Exam (St_Id, E_Id, Grade)
    VALUES (@s_ID, @E_ID, @Grade);
END;
GO
```

Uses

[dbo].[Student_Exam]

[dbo].[InsertStudentsWithCourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@St_Id	int	4
@C_Id	int	4

SQL Script

```
CREATE    PROCEDURE [dbo].[InsertStudentsWithCourse]
    @St_Id INT,
    @C_Id INT
AS
BEGIN
    IF NOT EXISTS (SELECT 1 FROM Course_Student WHERE St_Id = @St_Id AND C_Id = @C_Id)
    BEGIN
        INSERT INTO Course_Student (St_Id, C_Id)
        VALUES (@St_Id, @C_Id);
    END
    ELSE
    BEGIN
        PRINT 'Student is already enrolled in this course.';
    END
END;
GO
```

Uses

[dbo].[Course_Student]

[dbo].[insertTrackCourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@T_Id	int	4
@C_Id	int	4
@Ins_Id	int	4

SQL Script

```
CREATE PROC [dbo].[insertTrackCourse]
@T_Id INT,
@C_Id INT,
@Ins_Id INT
AS
BEGIN
    -- Check if Course exists
    IF NOT EXISTS (SELECT 1 FROM Course WHERE C_Id = @C_Id)
    BEGIN
        RAISERROR ('Error: Course ID does not exist.', 16, 1);
        RETURN;
    END

    -- Check if Track exists
    IF NOT EXISTS (SELECT 1 FROM Track WHERE T_Id = @T_Id)
    BEGIN
        RAISERROR ('Error: Track ID does not exist.', 16, 1);
        RETURN;
    END

    INSERT INTO Instructor_Track
VALUES (@Ins_Id, @T_Id);

    INSERT INTO Instructor_Course
VALUES (@Ins_Id, @C_Id);

    INSERT INTO Track_Course (T_Id, C_Id)
VALUES (@T_Id, @C_Id);
END;
```

Project > (local)\MSSQLSERVER01 > User databases > Online-
Examination > Programmability > Stored Procedures > dbo.insert-
TrackCourse

GO

Uses

[dbo].[Course]
[dbo].[Instructor_Course]
[dbo].[Instructor_Track]
[dbo].[Track]
[dbo].[Track_Course]

[dbo].[insertTrackInstructor]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Ins_Id	int	4
@T_Id	int	4

SQL Script

```
create proc [dbo].[insertTrackInstructor]
@Ins_Id int,
@T_Id int
as
begin
insert into Track_Instructor (T_Id, Ins_Id) values (@T_Id, @Ins_Id)
end
GO
```

[dbo].[ModelAnswer]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@ExamID	int	4
@StudentID	int	4

SQL Script

```
CREATE    PROCEDURE [dbo].[ModelAnswer]
    @ExamID INT,
    @StudentID INT
AS
BEGIN
    BEGIN TRY
        -- Ensure the procedure runs within a valid transaction scope
        BEGIN TRANSACTION;

        -- Select student answers and correct answers
        SELECT
            sa.Exam_ID,
            sa.S_Id,
            q.Q_Title,
            MAX(qa.answer) AS StudentAnswer,
            MAX(qc.answer) AS ModelAnswer
        FROM Question q
        INNER JOIN Question_Choices qa
            ON q.Q_Id = qa.Q_Id
        INNER JOIN StudenAnswer sa
            ON sa.QID = qa.Q_id AND sa.S_Anwer = qa.Choice_Num
        INNER JOIN Question_Choices qc
            ON sa.QID = qc.Q_Id AND sa.C_Answer = qc.Choice_Num
        WHERE sa.Exam_ID = @ExamID
            AND sa.S_Id = @StudentID
        GROUP BY sa.Exam_ID, q.Q_Title, sa.S_Id;

        -- Commit transaction if successful
        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        -- Rollback transaction in case of error
        ROLLBACK TRANSACTION;
```

```
-- Return error details
DECLARE @ErrorMessage NVARCHAR(4000), @ErrorSeverity INT, @ErrorState INT;
SELECT @ErrorMessage = ERROR_MESSAGE(),
       @ErrorSeverity = ERROR_SEVERITY(),
       @ErrorState = ERROR_STATE();

RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);
END CATCH;
END;
GO
```

Uses

[dbo].[Question]
[dbo].[Question_Choices]
[dbo].[StudentAnswer]

[dbo].[ModelAnswerEx]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@ExamID	int	4
@StudentID	int	4

SQL Script

```

----- OMAR MOHAMED -----

/*Model Answer
Report Name : ModelExam
*/
CREATE PROCEDURE [dbo].[ModelAnswerEx]
    @ExamID INT,
    @StudentID INT
AS
BEGIN
    BEGIN TRY
        -- Ensure the procedure runs within a valid transaction scope
        BEGIN TRANSACTION;

        -- Select student answers and correct answers
        SELECT
            sa.Exam_ID,
            sa.S_Id,
            q.Q_Title,
            MAX(qa.answer) AS StudentAnswer,
            MAX(qc.answer) AS ModelAnswer
        FROM Question q
        INNER JOIN Question_Choices qa
            ON q.Q_Id = qa.Q_Id
        INNER JOIN StudenAnswer sa
            ON sa.QID = qa.Q_id AND sa.S_Anwer = qa.Choice_Num
        INNER JOIN Question_Choices qc
            ON sa.QID = qc.Q_Id AND sa.C_Answer = qc.Choice_Num
        WHERE sa.Exam_ID = @ExamID
            AND sa.S_Id = @StudentID
        GROUP BY sa.Exam_ID, q.Q_Title, sa.S_Id;
    
```



```
-- Commit transaction if successful
COMMIT TRANSACTION;
END TRY
BEGIN CATCH
    -- Rollback transaction in case of error
    ROLLBACK TRANSACTION;

    -- Return error details
    DECLARE @ErrorMessage NVARCHAR(4000), @ErrorSeverity INT, @ErrorState INT;
    SELECT @ErrorMessage = ERROR_MESSAGE(),
           @ErrorSeverity = ERROR_SEVERITY(),
           @ErrorState = ERROR_STATE();

    RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);
END CATCH;
END;
GO
```

Uses

[dbo].[Question]
[dbo].[Question_Choices]
[dbo].[StudenAnswer]

[dbo].[online_BranchTrack_Insert]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@B_id_fk	int	4
@T_id_fk	int	4

SQL Script

```
--Insert into Branch_Track --> Testing : done
CREATE PROCEDURE [dbo].[online_BranchTrack_Insert]
    @B_id_fk INT = NULL,
    @T_id_fk INT = NULL
AS
BEGIN
    BEGIN TRY
        IF NOT EXISTS (SELECT T_id FROM track WHERE T_id = @T_id_fk)
        BEGIN
            SELECT 'This Track Does Not Exist.' AS Message;
        END
        ELSE IF NOT EXISTS (SELECT B_id FROM branch WHERE B_id = @B_id_fk)
        BEGIN
            SELECT 'This Branch Does Not Exist.' AS Message;
        END
        ELSE IF EXISTS (SELECT 1 FROM branch_track WHERE T_id = @T_id_fk AND B_id =
@B_id_fk)
        BEGIN
            SELECT 'This Record Already Exists.' AS Message;
        END
        ELSE
        BEGIN
            -- Insert the new record
            INSERT INTO branch_track (B_id, T_id)
            VALUES (@B_id_fk, @T_id_fk);

            SELECT 'Inserted successfully.' AS Message;
        END
    END TRY
    BEGIN CATCH
        -- Handle errors
        SELECT ERROR_MESSAGE() AS ErrorMessage;
    END CATCH
END
```

Project > (local)\MSSQLSERVER01 > User databases > Online-
Examination > Programmability > Stored Procedures >
dbo.online_BranchTrack_Insert

END

GO

Uses

[dbo].[Branch]

[dbo].[Branch_Track]

[dbo].[Track]

Project > (local)\MSSQLSERVER01 > User databases > Online-
Examination > Programmability > Stored Procedures >
dbo.online_BranchTrack_SelectAll

[dbo].[online_BranchTrack_SelectAll]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE PROCEDURE [dbo].[online_BranchTrack_SelectAll]
AS
BEGIN
    SELECT * FROM Branch_Track;
END
GO
```

Uses

[dbo].[Branch_Track]

[dbo].[online_BranchTrack_SelectByPK]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@B_id_fk	int	4
@T_id_fk	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[online_BranchTrack_SelectByPK]
    @B_id_fk INT = NULL,
    @T_id_fk INT = NULL
AS
BEGIN
    -- Check for NULL and existence of the provided values
    IF @B_id_fk IS NOT NULL AND @T_id_fk IS NOT NULL
    BEGIN
        IF EXISTS (SELECT 1 FROM branch_track t WHERE t.B_Id = @B_id_fk AND t.T_Id =
@T_id_fk)
        BEGIN
            SELECT B_Id, T_Id
            FROM Branch_Track
            WHERE B_Id = @B_id_fk AND T_Id = @T_id_fk;
        END
        ELSE
        BEGIN
            SELECT 'This Track is Not Found in This Branch' AS Message;
        END
        ELSE
        BEGIN
            SELECT 'Invalid Input: Both B_id_fk and T_id_fk must be provided' AS Message;
        END
    END
GO
```

Uses

[dbo].[Branch_Track]

[dbo].[online_BranchTrack_Update]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@B_id_fk	int	4
@T_id_fk	int	4
@new_B_id_fk	int	4
@new_T_id_fk	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[online_BranchTrack_Update]
    @B_id_fk INT = NULL,
    @T_id_fk INT = NULL,
    @new_B_id_fk INT = NULL,
    @new_T_id_fk INT = NULL
AS
BEGIN
    -- Check if the new Track exists
    IF NOT EXISTS (SELECT T_id FROM track WHERE T_id = @new_T_id_fk)
    BEGIN
        SELECT 'This Track Does Not Exist.' AS Message;
    END
    -- Check if the new Branch exists
    ELSE IF NOT EXISTS (SELECT B_id FROM branch WHERE B_id = @new_B_id_fk)
    BEGIN
        SELECT 'This Branch Does Not Exist.' AS Message;
    END
    -- Check if the provided B_id_fk and T_id_fk exist in the branch_track table
    ELSE IF EXISTS (SELECT 1 FROM branch_track WHERE B_id = @B_id_fk AND T_id = @T_id_fk)
    BEGIN
        -- Update the branch_track record
        UPDATE branch_track
        SET
            B_id = COALESCE(@new_B_id_fk, B_id),
            T_id = COALESCE(@new_T_id_fk, T_id)
        WHERE B_id = @B_id_fk AND T_id = @T_id_fk;

        SELECT 'Update completed successfully.' AS Message;
    END
END
```

Project > (local)\MSSQLSERVER01 > User databases > Online-
Examination > Programmability > Stored Procedures >
dbo.online_BranchTrack_Update

```
ELSE
BEGIN
    -- If the original B_id_fk and T_id_fk do not exist in the branch_track table
    SELECT 'This id is not exist.' AS Message;
END
END
GO
```

Uses

[dbo].[Branch]
[dbo].[Branch_Track]
[dbo].[Track]

Project > (local)\MSSQLSERVER01 > User databases > Online-
Examination > Programmability > Stored Procedures >
dbo.online_GetAllStudentsphones

[dbo].[online_GetAllStudentsphones]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE PROCEDURE [dbo].[online_GetAllStudentsphones]
AS
BEGIN
    SELECT * FROM student_phone;
END;
GO
```

Uses

[dbo].[Student_Phone]

[dbo].[online_GetStudentPhoneById]

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].[online_GetStudentPhoneById]
    @id INT
AS
BEGIN
    IF @id IS NOT NULL AND EXISTS (SELECT 1 FROM student_phone WHERE st_id = @id)
    BEGIN
        SELECT * FROM student_phone WHERE st_id = @id;
    END
    ELSE
    BEGIN
        SELECT NULL AS st_id, NULL AS phone, 'This Id Not Found' AS message;
    END
END;
GO
```

Uses

[dbo].[Student_Phone]

Project > (local)\MSSQLSERVER01 > User databases > Online-
Examination > Programmability > Stored Procedures >
dbo.online_Track_Delete

[dbo].[online_Track_Delete]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE PROCEDURE [dbo].[online_Track_Delete]
    @T_id INT = NULL
AS
BEGIN
    IF EXISTS (SELECT 1 FROM track WHERE T_id = @T_id)
    BEGIN
        DELETE FROM Track WHERE T_id = @T_id;
    END
    ELSE
    BEGIN
        SELECT 'This Track is Not Found' AS Message;
    END
END
GO
```

Uses

[dbo].[Track]

Project > (local)\MSSQLSERVER01 > User databases > Online-
Examination > Programmability > Stored Procedures >
dbo.online_Track_Insert

[dbo].[online_Track_Insert]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@T_name	varchar(100)	100

SQL Script

```
CREATE    PROCEDURE [dbo].[online_Track_Insert]
    @T_name VARCHAR(100)
AS
BEGIN
    BEGIN TRY
        INSERT INTO Track (T_Name)
            VALUES (@T_name);

        SELECT 'Inserted completed successfully.' AS Message;
    END TRY
    BEGIN CATCH
        SELECT ERROR_MESSAGE() AS Message;
    END CATCH
END;
GO
```

Uses

[dbo].[Track]

Project > (local)\MSSQLSERVER01 > User databases > Online-
Examination > Programmability > Stored Procedures >
dbo.online_Track_SelectAll

[dbo].[online_Track_SelectAll]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE    PROCEDURE [dbo].[online_Track_SelectAll]
AS
BEGIN
    SELECT * FROM Track
END
GO
```

Uses

[dbo].[Track]

Project > (local)\MSSQLSERVER01 > User databases > Online-
Examination > Programmability > Stored Procedures >
dbo.online_Track_SelectByTrackId

[dbo].[online_Track_SelectByTrackId]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE PROCEDURE [dbo].[online_Track_SelectByTrackId]
    @Track_Id INT = NULL
AS
BEGIN
    IF EXISTS (SELECT 1 FROM Track WHERE T_Id = @Track_Id)
    BEGIN
        SELECT *
        FROM Track
        WHERE T_Id = @Track_Id
    END
    ELSE
    BEGIN
        SELECT 'This Track is Not Found' AS Message
    END
END
GO
```

Uses

[dbo].[Track]

[dbo].[online_Track_Update]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@T_id	int	4
@T_name	varchar(100)	100

SQL Script

```
CREATE    PROCEDURE [dbo].[online_Track_Update]
    @T_id INT = NULL,
    @T_name VARCHAR(100) = NULL
AS
BEGIN
    IF EXISTS (SELECT 1 FROM Track WHERE T_id = @T_id)
    BEGIN
        UPDATE Track
        SET
            T_name = COALESCE(@T_name, T_name)
        WHERE T_id = @T_id;

        SELECT 'Update completed successfully.' AS Message;
    END
    ELSE
    BEGIN
        SELECT 'This ID is not found.' AS Message;
    END
END;
GO
```

Uses

[dbo].[Track]

[dbo].[online_TrackBranch_Delete]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@B_id_fk	int	4
@T_id_fk	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[online_TrackBranch_Delete]
    @B_id_fk INT = NULL,
    @T_id_fk INT = NULL
AS
BEGIN
    -- Check if the combination of B_id_fk and T_id_fk exists in the branch_track table
    IF EXISTS (SELECT 1 FROM branch_track WHERE B_id = @B_id_fk AND T_id = @T_id_fk)
    BEGIN
        -- Delete the record from branch_track
        DELETE FROM Branch_Track
        WHERE B_id = @B_id_fk AND T_id = @T_id_fk;

        SELECT 'Record deleted successfully.' AS Message;
    END
    ELSE
    BEGIN
        -- Return a message if the record does not exist
        SELECT 'This Track Does Not Exist in This Branch.' AS Message;
    END
END
GO
```

Uses

[dbo].[Branch_Track]

[dbo].[SelectAllBranches]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE    PROCEDURE [dbo] . [SelectAllBranches]
as
BEGIN
    SELECT B_Id,B_Name,B_Phone,B_Location FROM Branch;
END;

GO
```

Uses

[dbo].[Branch]

[dbo].[SelectAllBranchPhones]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE PROCEDURE [dbo].[SelectAllBranchPhones]
as
BEGIN
    SELECT * FROM Branch_Phone;
END;
GO
```

[dbo].[SelectAllInstructorTeachCourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE PROCEDURE [dbo].[SelectAllInstructorTeachCourse]
as
BEGIN
    SELECT * FROM Instructor_Course;
END;
GO
```

Uses

[dbo].[Instructor_Course]

[dbo].[SelectAllStudents]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE PROCEDURE [dbo].[SelectAllStudents]
as
BEGIN
    SELECT St_Id,F_Name,L_Name,St_Email,St_Gender,Track_Id,St_DOB,St_Address FROM Student;
END;

GO
```

Uses

[dbo].[Student]

[dbo].[SelectBranchById]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE PROCEDURE [dbo].[SelectBranchById] @branchId INT
as
BEGIN
    SELECT * FROM Branch WHERE B_Id = @branchId;
END;
GO
```

Uses

[dbo].[Branch]

[dbo].[SelectBranchPhoneById]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE PROCEDURE [dbo].[SelectBranchPhoneById] @branchId INT
as
BEGIN
    SELECT * FROM Branch_Phone WHERE B_Id = @branchId;
END;
GO
```

Project > (local)\MSSQLSERVER01 > User databases > Online-
Examination > Programmability > Stored Procedures >
dbo.selectcourse

[dbo].[selectcourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
create proc [dbo].[selectcourse]
@C_ID int
as
begin
if not exists(
select * from Course c
where c.C_Id=@C_ID
)
begin
print 'Course not found ...!'
return;
end
select * from Course c
where c.C_Id=@C_ID;
end
GO
```

Uses

[dbo].[Course]

[dbo].[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].[selectinstructor]
@id int
as
begin
if not exists(select i.Ins_Id from Instructor i where i.Ins_Id=@id)
begin
print 'Instructor not found'
return;
end
select *from Instructor i where i.Ins_Id=@id;
end;
GO
```

Uses

[dbo].[Instructor]

[dbo].[SelectInstructorTeachCourseById]

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].[SelectInstructorTeachCourseById] @instructorId INT
as
BEGIN
    SELECT * FROM Instructor_Course WHERE Ins_Id = @instructorId;
END;
GO
```

Uses

[dbo].[Instructor_Course]

[dbo].[SelectStudentById]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE PROCEDURE [dbo].[SelectStudentById] @studentId INT
as
BEGIN

    SELECT * FROM Student WHERE St_Id = @studentId;
END;
GO
```

Uses

[dbo].[Student]

[dbo].[selectworkon]

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].[selectworkon]
@id int
as
begin
if exists(
select *from Track_Instructor t
where t.Ins_Id=@id
)
begin
select *from Track_Instructor t
where t.Ins_Id=@id
end
else
begin
print 'Instructor not exist ...!'
end
end;
GO
```

[dbo].[sp_AuthenticateUser]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Email	nvarchar(150)	300
@Password	nvarchar(255)	510

SQL Script

```
CREATE PROCEDURE [dbo].[sp_AuthenticateUser]
    @Email NVARCHAR(150),
    @Password NVARCHAR(255)
AS
BEGIN
    SET NOCOUNT ON;

    -- Check Students
    IF EXISTS (SELECT 1 FROM Student WHERE St_Email = @Email AND Password = @Password)
    BEGIN
        SELECT 'Student' AS Role, St_Id AS UserId, F_Name, L_Name, St_Email
        FROM Student WHERE St_Email = @Email;
        RETURN;
    END

    -- Check Instructors
    IF EXISTS (SELECT 1 FROM Instructor WHERE Ins_Email = @Email AND Password = @Password)
    BEGIN
        SELECT 'Instructor' AS Role, Ins_Id AS UserId, F_Name, L_Name, Ins_Email
        FROM Instructor WHERE Ins_Email = @Email;
        RETURN;
    END

    -- Check Admins
    IF EXISTS (SELECT 1 FROM Admin WHERE A_Email = @Email AND Password = @Password)
    BEGIN
        SELECT 'Admin' AS Role, A_Id AS UserId, F_Name, L_Name, A_Email
        FROM Admin WHERE A_Email = @Email;
        RETURN;
    END

    -- Invalid login
    SELECT NULL AS Role, NULL AS UserId, NULL AS F_Name, NULL AS L_Name, NULL AS Email;
```

Project > (local)\MSSQLSERVER01 > User databases > Online-
Examination > Programmability > Stored Procedures > dbo.sp_-
AuthenticateUser

```
END;  
GO
```

Uses

[dbo].[Admin]
[dbo].[Instructor]
[dbo].[Student]

[dbo].[Track_course_SelctByID]

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].[Track_course_SelctByID]
@id int
as
begin
if not exists(
select t.C_Id from Track_Course t
where C_Id=@id
)
begin
print 'Course not exist in track..!'
return;
end
select * from Track_Course t
where t.C_Id=@id
end
GO
```

Uses

[dbo].[Track_Course]

[dbo].[UpdateBranch]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@branchId	int	4
@name	varchar(20)	20
@location	varchar(20)	20
@phone	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[UpdateBranch] @branchId INT, @name VARCHAR(20), @location  
VARCHAR(20), @phone INT  
as  
BEGIN  
    UPDATE Branch  
    SET B_Name = @name, B_Location = @location, B_Phone = @phone  
    WHERE B_Id = @branchId;  
END;  
GO
```

Uses

[dbo].[Branch]

[dbo].[updatecourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@id	int	4
@C_Name	varchar(50)	50
@C_Duration	varchar(50)	50

SQL Script

```
create proc [dbo].[updatecourse]
@id int,
@C_Name varchar(50),
@C_Duration varchar(50)
as
begin
    if not exists
    (select c.C_Id from Course c
    where c.C_Id=@id)
    begin
        print 'Thid id not exist...!'
        return;
    end
    update Course set
    C_Name=@C_Name,
    C_Duration=@C_Duration
    where C_Id=@id
end
GO
```

Uses

[dbo].[Course]

[dbo].[updatecoursetrack]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
create proc [dbo].[updatecoursetrack]
@id int,
@T_ID int
as
begin
if not exists(
select t.T_Id from Track_Course t
where T_Id=@T_ID
)
begin
print 'track not exist ...!'
return;
end
begin
print 'The combination of Track ID and Course ID already exists.';
return;
end
update Track_Course
set
C_Id=@id
where T_Id=@T_ID
end
GO
```

Uses

[dbo].[Track_Course]

[dbo].[UpdateExam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@E_Id	int	4
@E_Name	varchar(20)	20
@E_Total_Score	int	4
@E_Duration	float	8
@E_Date	date	3

SQL Script

```
--UpdateExam
Create  PROC [dbo].[UpdateExam]
(
    @E_Id          INT,
    @E_Name        VARCHAR(20),
    @E_Total_Score INT,
    @E_Duration    FLOAT,
    @E_Date        DATE
)
AS
BEGIN
    UPDATE Exam
    SET E_Name          = @E_Name,
        E_Total_Score = @E_Total_Score,
        E_Duration     = @E_Duration,
        E_Date         = @E_Date
    WHERE E_Id = @E_Id;
END;
GO
```

Project > (local)\MSSQLSERVER01 > User databases > Online-
Examination > Programmability > Stored Procedures >
dbo.UpdateExam

Uses

[dbo].[Exam]

[dbo].[UpdateInstructor]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Ins_Id	int	4
@F_Name	varchar(20)	20
@L_Name	varchar(20)	20
@Ins_Age	int	4
@Ins_Email	varchar(30)	30
@Ins_Address	varchar(50)	50
@Ins_Gender	varchar(6)	6
@Ins_DOB	date	3
@Ins_Salary	decimal(10,2)	9

SQL Script

```
-----  
  
-- Update  
CREATE    PROCEDURE [dbo].[UpdateInstructor]  
(  
    @Ins_Id      INT,  
    @F_Name      VARCHAR(20),  
    @L_Name      VARCHAR(20),  
    @Ins_Age     INT,  
    @Ins_Email   VARCHAR(30),  
    @Ins_Address VARCHAR(50),  
    @Ins_Gender  VARCHAR(6),  
    @Ins_DOB     DATE,  
    @Ins_Salary  DECIMAL(10,2)  
)  
AS  
BEGIN  
    UPDATE Instructor  
    SET  
        F_Name      = @F_Name,  
        L_Name      = @L_Name,
```

Project > (local)\MSSQLSERVER01 > User databases > Online-
Examination > Programmability > Stored Procedures >
dbo.UpdateInstructor

```
Ins_Age      = @Ins_Age,  
Ins_Email    = @Ins_Email,  
Ins_Address=  @Ins_Address,  
Ins_Gender   = @Ins_Gender,  
Ins_DOB      = @Ins_DOB,  
Ins_Salary   = @Ins_Salary  
WHERE Ins_Id = @Ins_Id;  
END;  
GO
```

Uses

[dbo].[Instructor]

[dbo].[UpdateInstructorTeachCourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@oldInstructorId	int	4
@oldCourseId	int	4
@newInstructorId	int	4
@newCourseId	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[UpdateInstructorTeachCourse] @oldInstructorId INT, @oldCourseId  
INT, @newInstructorId INT, @newCourseId INT  
as  
BEGIN  
    UPDATE Instructor_Course  
    SET Ins_Id = @newInstructorId, C_Id = @newCourseId  
    WHERE Ins_Id = @oldInstructorId AND C_Id = @oldCourseId;  
END;  
GO
```

Uses

[dbo].[Instructor_Course]

[dbo].[UpdateStudent]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@studentId	int	4
@fName	varchar(20)	20
@lName	varchar(20)	20
@email	varchar(40)	40
@trackId	int	4
@dob	date	3
@address	varchar(40)	40
@gender	varchar(6)	6

SQL Script

```
CREATE PROCEDURE [dbo].[UpdateStudent] @studentId INT, @fName VARCHAR(20), @lName  
VARCHAR(20), @email VARCHAR(40), @trackId INT, @dob DATE, @address VARCHAR(40), @gender  
VARCHAR(6)  
as  
BEGIN  
    UPDATE Student  
    SET F_Name = @fName, L_Name = @lName, St_Email = @email, Track_Id = @trackId, St_DOB =  
@dob, St_Address = @address, St_Gender = @gender  
    WHERE St_Id = @studentId;  
END;  
GO
```

Uses

[dbo].[Student]

[dbo].[updateStudentPhone]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE PROCEDURE [dbo].[updateStudentPhone]
    @st_id INT,
    @newphone INT
AS
BEGIN
    IF EXISTS (SELECT st_id FROM student_phone WHERE st_id = @st_id)
    BEGIN
        UPDATE student_phone
        SET St_Phone = @newphone
        WHERE st_id = @st_id;

        SELECT 'Phone number updated successfully.' AS ResultMessage;
    END
    ELSE
    BEGIN
        SELECT 'Invalid ID' AS ResultMessage;
    END
END;
GO
```

Uses

[dbo].[Student_Phone]

[dbo].[updateworkon]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
create proc [dbo].[updateworkon]
@id int,
@I_ID int
as
begin
if not exists(
select *from Track_Instructor t
where t.Ins_Id=@id
)
begin
print 'Instructor not exist ...!'
end

update Track_Instructor
set
T_Id=@I_ID
where Ins_Id=@id

end
GO
```