DESKTOP-3CE5M2E

ITIExaminationSystem

Server DESKTOP-3CE5M2E

Author Softlaptop

Created 12 December 2024 21:39:59

File Path C:\Users\Softlaptop\OneDrive\Desktop\DESKTOP-3CE5M2E-2024-12-12T21-39-59.pdf

Table of Contents

able of Contents	2
■ DESKTOP-3CE5M2E	6
User databases	8
ITIExaminationSystem Database	9
Tables	12
[dbo].[branch]	13
[dbo].[Branch_Phone]	15
[dbo].[Course]	17
[dbo].[CourseHasTrack]	19
Ⅲ [dbo].[Exam]	21
[dbo].[Exam_has_Quesstion]	23
[dbo].[Instructor]	25
[dbo].[Instructor_Teach_Cource]	27
[dbo].[InstWorksInTrack]	29
[dbo].[Question]	31
[dbo].[Question_choice]	33
[dbo].[Student]	35
[dbo].[Student_Exam_Question]	37
[dbo].[student_phone]	39
[dbo].[Topic]	41
[dbo].[track]	43
[dbo].[track_branch]	45
Stored Procedures	47
[dbo].[AddPhone]	50
[dbo].[Correct_Exam]	51
[dbo].[DeleteCourseData]	53
[dbo].[DeleteCourseHasTrack]	54
[dbo].[deleteGradeTernary]	55
[dbo].[DeleteInst]	56
[dbo].[DeleteInstWorksInTrack]	57
[dbo].[delExam]	58
[dbo].[delphone]	59
[dbo].[Exam_Answers]	
[dbo].[ExamGeneration]	63
[dbo].[ExQ_Delete]	65
[dbo].[ExQ_get_sp]	66
[dbo].[ExQ_Insert]	67

[dbo].[ExQ_Update]	68
[dbo].[GetCourseAndNumOfStudByInstructorId]	70
[dbo].[GetCourseData]	71
[dbo].[GetCourseTrack]	72
[dbo].[GetExam_Questions_Model_stud_Answer]	73
[dbo].[GetExamDatabyld]	74
[dbo].[GetGradesInAllCouresBystdId]	75
[dbo].[GetInstData]	76
[dbo].[GetInstTrack]	77
[dbo].[GetQuestionsByExamNum]	78
[dbo].[GetstdgradeTernary]	79
[dbo].[GetstdgradeTernaryby3_IDs]	80
[dbo].[GetstdgradeTernarybySt_ID]	81
[dbo].[getstdphone]	82
[dbo].[GetStudentdPhonebyId]	83
[dbo].[GetStudentsByDeptNum]	84
[dbo].[GetTopicByCourseId]	85
[dbo].[InsertExam]	86
[dbo].[InsertInto_InsWorksInTrack]	88
[dbo].[InsertIntoCourse]	90
[dbo].[InsertIntoCourseHasTrack]	91
[dbo].[InsGradeTernary]	93
[dbo].[InsStdPhone]	94
[dbo].[Q_Delete]	95
[dbo].[Q_Get_sp]	96
[dbo].[Q_Insert]	97
[dbo].[Q_Update]	98
[dbo].[QCh_Delete]	99
[dbo].[QCh_Get]	100
[dbo].[QCh_Insert]	101
[dbo].[QCh_Update]	102
[dbo].[selexam]	103
[dbo].[sp_InsertIntoInst]	104
[dbo].[sp_InstructorTeachCource_Delete]	105
[dbo].[sp_InstructorTeachCource_Insert]	106
[dbo].[sp_InstructorTeachCource_SelectByPK]	107
[dbo].[sp_Topic_Delete]	108
[dbo].[sp_Topic_Insert]	109

[dbo].[sp_Topic_SelectAll]	110
[dbo].[sp_Topic_SelectByPK]	111
[dbo].[sp_Topic_Update]	112
[dbo].[sp_TrackBranch_Delete]	114
[dbo].[updateanswerTernary]	115
[dbo].[updateCourse]	116
[dbo].[UpdateCourseHasTrack]	118
[dbo].[updateExam]	120
[dbo].[updateGradeTernary]	122
[dbo].[UpdateInst]	123
[dbo].[updateInstWorksInTrack]	125
[dbo].[usp_Branch_Delete]	127
[dbo].[usp_Branch_Insert]	128
[dbo].[usp_Branch_SelectAll]	129
[dbo].[usp_Branch_SelectById]	130
[dbo].[usp_Branch_Update]	131
[dbo].[usp_BranchPhone_Delete]	132
[dbo].[usp_BranchPhone_Insert]	133
[dbo].[usp_BranchPhone_SelectAll]	134
[dbo].[usp_BranchPhone_SelectByBranch]	135
[dbo].[usp_InstructorTeachCource_SelectAll]	136
[dbo].[usp_InstructorTeachCource_Update]	137
[dbo].[usp_Student_Delete]	139
[dbo].[usp_Student_Insert]	140
[dbo].[usp_Student_SelectAll]	142
[dbo].[usp_Student_SelectByPK]	143
[dbo].[usp_Student_Update]	144
[dbo].[usp_Track_Delete]	146
[dbo].[usp_Track_Insert]	147
[dbo].[usp_Track_SelectAll]	148
[dbo].[usp_Track_SelectById]	149
[dbo].[usp_Track_Update]	150
[dbo].[usp_TrackBranch_Insert]	151
[dbo].[usp_TrackBranch_SelectAll]	153
[dbo].[usp_TrackBranch_SelectByPK]	
[dbo].[usp_TrackBranch_Update]	155
♣ Users	157
1 dbo	158

	1 guest	159
-	Database Roles	
	db_accessadmin	160
	db_backupoperator	160
	db_datareader	161
	db_datawriter	161
	db_ddladmin	161
	db_denydatareader	162
	db_denydatawriter	162
	db_owner	162
	db_securityadmin	
	public	

■ DESKTOP-3CE5M2E

Databases (1)

• 🛘 ITIExaminationSystem

Server Properties

Property	Value
Product	Microsoft SQL Server
Version	16.0.1000.6
Language	English (United States)
Platform	NT x64
Edition	Developer Edition (64-bit)
Engine Edition	3 (Enterprise)
Processors	4
OS Version	6.3 (19045)
Physical Memory	16267
Is Clustered	False
Root Directory	C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL
Collation	SQL_Latin1_General_CP1_CI_AS

Server Settings

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

Advanced Server Settings

Property	Value
Locks	0
Nested triggers enabled	True
Allow triggers to fire others	True
Default language	English

Network packet size	4096
Default fulltext language LCID	1033
Two-digit year cutoff	2049
Remote login timeout	10
Cursor threshold	-1
Max text replication size	65536
Parallelism cost threshold	5
Max degree of parallelism	4
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)

• | ITIExaminationSystem

Database Properties

Property	Value
SQL Server Version	Max
Compatibility Level	Max
Last backup time	12/12/2024
Last log backup time	-
Creation date	Dec 12 2024
Users	4
Database Encryption Enabled	False
Database Encryption Algorithm	None
Database size	16.00 MB
Unallocated space	0.90 MB

Database Options

Property	Value
Compatibility Level	160
Database collation	SQL_Latin1_General_CP1_CI_AS
Restrict access	MULTI_USER
Is read-only	False
Auto close	True
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	fbb8ff23-8b02-4e62-95d6-3e56671a4f28
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-3CE5M2E\Softlaptop

Files

Name	Туре	Size	Maxsize	Autogrowth	File Name
ITIExamination- System	Data	8.00 MB	unlimited	64.00 MB	C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\DATA\ITIExamination- System.mdf
ITIExamination- System_log	Log	8.00 MB	2048.00 GB	64.00 MB	C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\DATA\ITIExamination- System_log.ldf

Project > DESKTOP-3CE5M2E > User databases > ITIExamination-System

■ Tables

Objects

Name
dbo.branch
dbo.Branch_Phone
dbo.Course
dbo.CourseHasTrack
dbo.Exam
dbo.Exam_has_Quesstion
dbo.Instructor
dbo.Instructor_Teach_Cource
dbo.InstWorksInTrack
dbo.Question
dbo.Question_choice
dbo.Student
dbo.Student_Exam_Question
dbo.student_phone
dbo.Topic
dbo.track
dbo.track_branch

[dbo].[branch]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	10
Created	20:00:12 04 December 2024
Last Modified	18:00:43 12 December 2024

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK2 C	B_id	int	4	NOT NULL	1000 - 1
	B_name	varchar(20)	20	NOT NULL	
	Location	varchar(50)	50	NULL allowed	

Indexes

Key	Name	Key Columns	Unique	
PK G	PKbranch4B25E3FE97A4A545	B_id	True	

SQL Script

```
CREATE TABLE [dbo].[branch]

(
[B_id] [int] NOT NULL IDENTITY(1000, 1),

[B_name] [varchar] (20) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,

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

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[branch] ADD CONSTRAINT [PK_branch_4B25E3FE97A4A545] PRIMARY KEY CLUSTERED ([B_id]) ON [PRIMARY]

GO
```

Used By

[dbo].[Branch_Phone]
[dbo].[track_branch]
[dbo].[usp_Branch_Delete]
[dbo].[usp_Branch_Insert]

Project > DESKTOP-3CE5M2E > User databases > ITIExamination-System > Tables > dbo.branch

[dbo].[usp_Branch_SelectAll]

[dbo].[usp_Branch_SelectById]

[dbo].[usp_Branch_Update]

[dbo].[usp_BranchPhone_Insert]

[dbo].[usp_TrackBranch_Insert]

[dbo].[usp_TrackBranch_Update]

[dbo].[Branch_Phone]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Неар	True
Row Count (~)	10
Created	20:00:12 04 December 2024
Last Modified	20:00:12 04 December 2024

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
FK	B_id_fk	int	4	NULL allowed
	Phone	varchar(11)	11	NULL allowed

Foreign Keys

Name	Update	Delete	Columns
FK_Branch_Ph_B_id38996AB5	Cascade	Cascade	B_id_fk->[dbo].[branch].[B_id]

SQL Script

```
CREATE TABLE [dbo].[Branch_Phone]

(
[B_id_fk] [int] NULL,
[Phone] [varchar] (11) COLLATE SQL_Latin1_General_CP1_CI_AS NULL

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Branch_Phone] ADD CONSTRAINT [FK_Branch_Ph_B_id__38996AB5] FOREIGN KEY

([B_id_fk]) REFERENCES [dbo].[branch] ([B_id]) ON DELETE CASCADE ON UPDATE CASCADE

GO
```

Uses

[dbo].[branch]

Used By

[dbo].[usp_BranchPhone_Delete]

Project > DESKTOP-3CE5M2E > User databases > ITIExamination-System > Tables > dbo.Branch_Phone

[dbo].[usp_BranchPhone_Insert]
[dbo].[usp_BranchPhone_SelectAll]
[dbo].[usp_BranchPhone_SelectByBranch]



Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	7
Created	17:57:28 08 December 2024
Last Modified	22:03:14 09 December 2024

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK G	C_ld	int	4	NOT NULL	100 - 20
	C_Name	varchar(max)	max	NOT NULL	
	C_Duration	int	4	NOT NULL	
	C_Description	varchar(max)	max	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
PKP C	PKCourseA9FDEC32BD5AE624	C_ld	True

SQL Script

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

(
[C_Id] [int] NOT NULL IDENTITY(100, 20),

[C_Name] [varchar] (max) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,

[C_Duration] [int] NOT NULL,

[C_Description] [varchar] (max) COLLATE SQL_Latin1_General_CP1_CI_AS NULL

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Course] ADD CONSTRAINT [PK__Course__A9FDEC32BD5AE624] PRIMARY KEY CLUSTERED

([C_Id]) ON [PRIMARY]

GO
```

Used By

[dbo].[CourseHasTrack] [dbo].[Exam]

Project > DESKTOP-3CE5M2E > User databases > ITIExamination-System > Tables > dbo.Course

[dbo].[Instructor_Teach_Cource]

[dbo].[Question]

[dbo].[Topic]

[dbo].[DeleteCourseData]

[dbo].[DeleteInst]

[dbo].[ExamGeneration]

[dbo].[GetCourseAndNumOfStudByInstructorId]

[dbo].[GetCourseData]

[dbo].[GetCourseTrack]

[dbo].[GetGradesInAllCouresBystdId]

[dbo].[InsertExam]

[dbo].[InsertIntoCourse]

[dbo].[InsertIntoCourseHasTrack]

[dbo].[Q_Update]

[dbo].[sp_InstructorTeachCource_Insert]

[dbo].[sp_Topic_Insert]

[dbo].[updateCourse]

[dbo].[UpdateCourseHasTrack]

[dbo].[updateExam]

[dbo].[usp_InstructorTeachCource_Update]

[dbo].[CourseHasTrack]

Properties

Property	Value
Row Count (~)	12
Created	20:00:12 04 December 2024
Last Modified	17:57:30 08 December 2024

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PKOFKO C	C_ld	int	4	NOT NULL
PKOFKO	Track_Id	int	4	NOT NULL

Indexes

Key	Name	Key Columns	Unique	
PK C	PKCourseHa5C272E58BF736055	C_ld, Track_ld	True	

Foreign Keys

Name	Update	Delete	Columns
cons3	Cascade	Cascade	Track_ld->[dbo].[track].[T_id]
cons4	Cascade	Cascade	C_ld->[dbo].[Course].[C_ld]

```
CREATE TABLE [dbo].[CourseHasTrack]

(
[C_Id] [int] NOT NULL,

[Track_Id] [int] NOT NULL

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[CourseHasTrack] ADD CONSTRAINT [PK_CourseHa_5C272E58BF736055] PRIMARY KEY

CLUSTERED ([C_Id], [Track_Id]) ON [PRIMARY]

GO

ALTER TABLE [dbo].[CourseHasTrack] ADD CONSTRAINT [cons3] FOREIGN KEY ([Track_Id]) REFERENCES

[dbo].[track] ([T_id]) ON DELETE CASCADE ON UPDATE CASCADE

GO

ALTER TABLE [dbo].[CourseHasTrack] ADD CONSTRAINT [cons4] FOREIGN KEY ([C_Id]) REFERENCES

[dbo].[Course] ([C_Id]) ON DELETE CASCADE ON UPDATE CASCADE
```

Uses
[dbo].[Course]
[dbo].[track]

Used By

[dbo].[DeleteCourseHasTrack]
[dbo].[GetCourseAndNumOfStudByInstructorId]
[dbo].[GetCourseTrack]
[dbo].[InsertIntoCourseHasTrack]
[dbo].[UpdateCourseHasTrack]

[dbo].[Exam]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	5
Created	20:00:12 04 December 2024
Last Modified	17:57:29 08 December 2024

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PK	ex_id	int	4	NOT NULL
	ex_name	varchar(50)	50	NULL allowed
	ex_date	date	3	NULL allowed
FK	c_id	int	4	NULL allowed
	total_score	int	4	NULL allowed

Indexes

Key	Name	Key Columns	Unique
PK	PKExamF6D3E489C8103E01	ex_id	True

Foreign Keys

Name	Update	Delete	Columns
FK_Exam_c_id_5EBF139D	Cascade	SetNull	c_id->[dbo].[Course].[C_ld]

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

(
[ex_id] [int] NOT NULL,
[ex_name] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[ex_date] [date] NULL,
[c_id] [int] NULL,
[total_score] [int] NULL
) ON [PRIMARY]

GO
```

```
ALTER TABLE [dbo].[Exam] ADD CONSTRAINT [PK_Exam_F6D3E489C8103E01] PRIMARY KEY CLUSTERED ([ex_id]) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Exam] ADD CONSTRAINT [FK_Exam_c_id_5EBF139D] FOREIGN KEY ([c_id])

REFERENCES [dbo].[Course] ([C_id]) ON DELETE SET NULL ON UPDATE CASCADE

GO
```

Uses

[dbo].[Course]

Used By

[dbo].[Exam_has_Quesstion]

[dbo].[Student_Exam_Question]

[dbo].[Correct_Exam]

[dbo].[delExam]

[dbo].[Exam_Answers]

[dbo].[ExamGeneration]

[dbo].[ExQ_Insert]

[dbo].[ExQ Update]

[dbo].[GetExamDatabyId]

[dbo].[GetGradesInAllCouresBystdId]

[dbo].[InsertExam]

[dbo].[InsGradeTernary]

[dbo].[selexam]

[dbo].[updateExam]

[dbo].[Exam_has_Quesstion]

Properties

Property	Value
Row Count (~)	50
Created	20:00:12 04 December 2024
Last Modified	05:09:22 09 December 2024

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PKPFKP C	ex_id	int	4	NOT NULL
PKPFKP C	Q_id	int	4	NOT NULL

Indexes

Key	Name	Key Columns	Unique	
PK C	PK_Exam_has899C363F8FC41B02	ex_id, Q_id	True	

Foreign Keys

Name	Update	Delete	Columns
FKExam_has_QQ_id628FA481			Q_id->[dbo].[Question].[Q_id]
FK_Exam_hasex_id619B8048	Cascade	Cascade	ex_id->[dbo].[Exam].[ex_id]

```
CREATE TABLE [dbo].[Exam_has_Quesstion]

(
[ex_id] [int] NOT NULL,
[Q_id] [int] NOT NULL
) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Exam_has_Quesstion] ADD CONSTRAINT [PK_Exam_has_899C363F8FC41B02] PRIMARY

KEY CLUSTERED ([ex_id], [Q_id]) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Exam_has_Quesstion] ADD CONSTRAINT [FK_Exam_has_Q_Q_id_628FA481] FOREIGN

KEY ([Q_id]) REFERENCES [dbo].[Question] ([Q_id])

GO

ALTER TABLE [dbo].[Exam_has_Quesstion] ADD CONSTRAINT [FK_Exam_has_ex_id_619B8048] FOREIGN

KEY ([ex_id]) REFERENCES [dbo].[Exam] ([ex_id]) ON DELETE CASCADE ON UPDATE CASCADE
```

GO

Uses

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

Used By

[dbo].[Exam_Answers]

[dbo].[ExamGeneration]

[dbo].[ExQ_Delete]

[dbo].[ExQ_get_sp]

[dbo].[ExQ_Insert]

[dbo].[ExQ_Update]

[dbo].[GetExam_Questions_Model_stud_Answer]

[dbo].[GetQuestionsByExamNum]

 $[dbo]. [GetstdgradeTernaryby3_IDs] \\$



Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	7
Created	18:24:10 08 December 2024
Last Modified	18:24:11 08 December 2024

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK	Ins_Id	int	4	NOT NULL	1 - 1
	Ins_Name	varchar(20)	20	NOT NULL	
	Ins_Email	varchar(max)	max	NULL allowed	
	Ins_salary	int	4	NOT NULL	
	Ins_HourRate	int	4	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
PKP C	PKInstruct151409ED5EA09863	Ins_Id	True

```
CREATE TABLE [dbo].[Instructor]
(
[Ins_Id] [int] NOT NULL IDENTITY(1, 1),
[Ins_Name] [varchar] (20) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
[Ins_Email] [varchar] (max) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[Ins_salary] [int] NOT NULL,
[Ins_HourRate] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Instructor] ADD CONSTRAINT [PK_Instruct_151409ED5EA09863] PRIMARY KEY CLUSTERED ([Ins_Id]) ON [PRIMARY]
GO
```

Used By

[dbo].[Instructor_Teach_Cource]

[dbo].[InstWorksInTrack]

[dbo].[DeleteInst]

[dbo].[GetInstData]

[dbo].[InsertInto_InsWorksInTrack]

[dbo].[sp_InsertIntoInst]

 $[dbo].[sp_InstructorTeachCource_Insert]$

[dbo].[sp_InstructorTeachCource_SelectByPK]

[dbo].[UpdateInst]

[dbo].[updateInstWorksInTrack]

 $[dbo].[usp_InstructorTeachCource_Update] \\$

[dbo].[Instructor_Teach_Cource]

Properties

Property	Value
Row Count (~)	13
Created	20:00:12 04 December 2024
Last Modified	18:24:11 08 December 2024

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PKOFKO	ins_id_fk	int	4	NOT NULL
PKPFKP C	C_id_fk	int	4	NOT NULL

Indexes

Key	Name	Key Columns	Unique
PKP C	PKInstructFE3B86A39F11FA5A	ins_id_fk, C_id_fk	True

Foreign Keys

Name	Update	Delete	Columns
FK_Instructo_C_id5070F446	Cascade	Cascade	C_id_fk->[dbo].[Course].[C_ld]
FK_Instructo_ins_i_4F7CD00D	Cascade	Cascade	ins_id_fk->[dbo].[Instructor].[Ins_ld]

```
CREATE TABLE [dbo].[Instructor_Teach_Cource]

(
[ins_id_fk] [int] NOT NULL,
[C_id_fk] [int] NOT NULL
) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Instructor_Teach_Cource] ADD CONSTRAINT [PK__Instruct__FE3B86A39F11FA5A]
PRIMARY KEY CLUSTERED ([ins_id_fk], [C_id_fk]) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Instructor_Teach_Cource] ADD CONSTRAINT [FK__Instructo__C_id__5070F446]
FOREIGN KEY ([C_id_fk]) REFERENCES [dbo].[Course] ([C_Id]) ON DELETE CASCADE ON UPDATE CASCADE

GO

ALTER TABLE [dbo].[Instructor_Teach_Cource] ADD CONSTRAINT [FK__Instructo__ins_i__4F7CD00D]
FOREIGN KEY ([ins_id_fk]) REFERENCES [dbo].[Instructor] ([Ins_Id]) ON DELETE CASCADE ON UPDATE
```

CASCADE GO

Uses

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

Used By

[dbo].[DeleteInst]

[dbo].[GetCourseAndNumOfStudByInstructorId]

[dbo].[sp_InstructorTeachCource_Delete]

[dbo].[sp_InstructorTeachCource_Insert]

[dbo].[sp_InstructorTeachCource_SelectByPK]

[dbo].[usp_InstructorTeachCource_SelectAll]

[dbo].[usp_InstructorTeachCource_Update]

[dbo].[InstWorksInTrack]

Properties

Property	Value
Row Count (~)	12
Created	20:00:12 04 December 2024
Last Modified	18:24:11 08 December 2024

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PKOFKO C	lns_ld	int	4	NOT NULL
PKOFKO C	Track_ld	int	4	NOT NULL
	HiringDate	date	3	NULL allowed

Indexes

Key	Name	Key Columns	Unique
PKP C	PKInstWorkE0CECB87E189B861	Ins_Id, Track_Id	True

Foreign Keys

Name	Update	Delete	Columns
cons1	Cascade	Cascade	Ins_Id->[dbo].[Instructor].[Ins_Id]
cons2	Cascade	Cascade	Track_ld->[dbo].[track].[T_id]

```
CREATE TABLE [dbo].[InstWorksInTrack]

(
[Ins_Id] [int] NOT NULL,

[Track_Id] [int] NOT NULL,

[HiringDate] [date] NULL

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[InstWorksInTrack] ADD CONSTRAINT [PK_InstWork_E0CECB87E189B861] PRIMARY KEY

CLUSTERED ([Ins_Id], [Track_Id]) ON [PRIMARY]

GO

ALTER TABLE [dbo].[InstWorksInTrack] ADD CONSTRAINT [cons1] FOREIGN KEY ([Ins_Id]) REFERENCES

[dbo].[Instructor] ([Ins_Id]) ON DELETE CASCADE ON UPDATE CASCADE
```

```
GO
ALTER TABLE [dbo].[InstWorksInTrack] ADD CONSTRAINT [cons2] FOREIGN KEY ([Track_id]) REFERENCES [dbo].[track] ([T_id]) ON DELETE CASCADE ON UPDATE CASCADE
GO
```

Uses

[dbo].[Instructor] [dbo].[track]

Used By

[dbo].[DeleteInst]
[dbo].[DeleteInstWorksInTrack]
[dbo].[GetInstTrack]
[dbo].[InsertInto_InsWorksInTrack]
[dbo].[updateInstWorksInTrack]

[dbo].[Question]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	196
Created	05:09:21 09 December 2024
Last Modified	20:04:15 09 December 2024

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PK	Q_id	int	4	NOT NULL
	type	varchar(50)	50	NULL allowed
	title	nvarchar(max)	max	NULL allowed
	model_answer	nvarchar(max)	max	NULL allowed
FK	c_id	int	4	NULL allowed

Indexes

Key	Name	Key Columns	Unique
PK	c10	Q_id	True

Foreign Keys

Name	Columns		
FK_Question_Course1	c_id->[dbo].[Course].[C_ld]		

```
CREATE TABLE [dbo].[Question]

(
[Q_id] [int] NOT NULL,

[type] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,

[title] [nvarchar] (max) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,

[model_answer] [nvarchar] (max) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,

[c_id] [int] NULL

) ON [PRIMARY]

GO
```

```
ALTER TABLE [dbo].[Question] ADD CONSTRAINT [c10] PRIMARY KEY CLUSTERED ([Q_id]) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Question] ADD CONSTRAINT [FK_Question_Course1] FOREIGN KEY ([c_id]) REFERENCES [dbo].[Course] ([C_Id])

GO
```

Uses

[dbo].[Course]

[dbo].[QCh_Insert] [dbo].[QCh_Update]

Used By

[dbo].[Exam_has_Quesstion]
[dbo].[Question_choice]
[dbo].[Student_Exam_Question]
[dbo].[Correct_Exam]
[dbo].[ExamGeneration]
[dbo].[ExQ_Insert]
[dbo].[ExQ_Update]
[dbo].[GetExam_Questions_Model_stud_Answer]
[dbo].[GetQuestionsByExamNum]
[dbo].[InsGradeTernary]
[dbo].[Q_Delete]
[dbo].[Q_Get_sp]
[dbo].[Q_Insert]
[dbo].[Q_Update]

[dbo].[Question_choice]

Properties

Property	Value		
Collation	SQL_Latin1_General_CP1_CI_AS		
Row Count (~)	604		
Created	20:00:12 04 December 2024		
Last Modified	05:09:22 09 December 2024		

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PKDFKD C	Q_id	int	4	NOT NULL
PKP C	choice	varchar(20)	20	NOT NULL
	textOfChoice	nvarchar(max)	max	NULL allowed

Indexes

Key	Name	Key Columns	Unique
PK	PKQuestionE38E2236A61F0679	Q_id, choice	True

Foreign Keys

Name	Update	Delete	Columns
FKQuestion_cQ_id5BE2A6F2	Cascade	Cascade	Q_id->[dbo].[Question].[Q_id]

```
CREATE TABLE [dbo].[Question_choice]

(
[Q_id] [int] NOT NULL,
[choice] [varchar] (20) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
[textOfChoice] [nvarchar] (max) COLLATE SQL_Latin1_General_CP1_CI_AS NULL

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Question_choice] ADD CONSTRAINT [PK_Question_E38E2236A61F0679] PRIMARY KEY CLUSTERED ([Q_id], [choice]) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Question_choice] ADD CONSTRAINT [FK_Question_c_Q_id_5BE2A6F2] FOREIGN KEY ([Q_id]) REFERENCES [dbo].[Question] ([Q_id]) ON DELETE CASCADE ON UPDATE CASCADE
```

GO

Uses

[dbo].[Question]

Used By

[dbo].[GetExam_Questions_Model_stud_Answer]
[dbo].[GetQuestionsByExamNum]
[dbo].[QCh_Delete]
[dbo].[QCh_Get]
[dbo].[QCh_Insert]
[dbo].[QCh_Update]

[dbo].[Student]

Properties

Property	Value		
Collation	SQL_Latin1_General_CP1_CI_AS		
Row Count (~)	22		
Created	17:54:56 08 December 2024		
Last Modified	17:54:56 08 December 2024		

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK	st_id	int	4	NOT NULL	1 - 1
	fname	varchar(20)	20	NOT NULL	
	Iname	varchar(20)	20	NOT NULL	
	gender	varchar(1)	1	NOT NULL	
	Email	varchar(max)	max	NULL allowed	
	BOD	date	3	NULL allowed	
FK	T_id_fk	int	4	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
PK C	PKStudentA85E81CFE31AF120	st_id	True

Foreign Keys

Name	Update	Delete	Columns
FK_Student_T_id_fk534D60F1	Cascade	Cascade	T_id_fk->[dbo].[track].[T_id]

```
CREATE TABLE [dbo].[Student]
(
[st_id] [int] NOT NULL IDENTITY(1, 1),
[fname] [varchar] (20) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
[lname] [varchar] (20) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
[gender] [varchar] (1) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
```

```
[Email] [varchar] (max) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,

[BOD] [date] NULL,

[T_id_fk] [int] NULL

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Student] ADD CONSTRAINT [PK__Student__A85E81CFE31AF120] PRIMARY KEY CLUSTERED ([st_id]) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Student] ADD CONSTRAINT [FK__Student__T_id_fk__534D60F1] FOREIGN KEY ([T_id_fk]) REFERENCES [dbo].[track] ([T_id]) ON DELETE CASCADE ON UPDATE CASCADE

GO
```

Uses

[dbo].[track]

Used By

```
[dbo].[Student_Exam_Question]
[dbo].[student_phone]
[dbo].[Correct_Exam]
[dbo].[Exam_Answers]
[dbo].[GetCourseAndNumOfStudByInstructorId]
[dbo].[GetStudentsByDeptNum]
[dbo].[InsGradeTernary]
[dbo].[InsStdPhone]
[dbo].[usp_Student_Delete]
[dbo].[usp_Student_Insert]
[dbo].[usp_Student_SelectAll]
[dbo].[usp_Student_SelectByPK]
[dbo].[usp_Student_Update]
```

[dbo].[Student_Exam_Question]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	30
Created	20:00:12 04 December 2024
Last Modified	05:09:21 09 December 2024

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PKPFKP C	st_id	int	4	NOT NULL
PKPFKP C	ex_id	int	4	NOT NULL
PKPFKP C	Q_id	int	4	NOT NULL
	grade	int	4	NULL allowed
	Answer	varchar(20)	20	NULL allowed

Indexes

Key	Name	Key Columns	Unique
PKP C	PKStudent40C742AC6BD83E88	st_id, ex_id, Q_id	True

Foreign Keys

Name	Update	Delete	Columns
FK_Student_E_st_id656C112C			st_id->[dbo].[Student].[st_id]
FK_Student_ExQ_id6754599E	Cascade	Cascade	Q_id->[dbo].[Question].[Q_id]
FK_Student_E_ex_id66603565			ex_id->[dbo].[Exam].[ex_id]

```
CREATE TABLE [dbo].[Student_Exam_Question]

(
[st_id] [int] NOT NULL,

[ex_id] [int] NOT NULL,

[Q_id] [int] NOT NULL,

[grade] [int] NULL,
```

```
[Answer] [varchar] (20) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]

GO

ALTER TABLE [dbo]. [Student_Exam_Question] ADD CONSTRAINT [PK_Student___40C742AC6BD83E88] PRIMARY

KEY CLUSTERED ([st_id], [ex_id], [Q_id]) ON [PRIMARY]

GO

ALTER TABLE [dbo]. [Student_Exam_Question] ADD CONSTRAINT [FK_Student_E_st_id_656C112C] FOREIGN

KEY ([st_id]) REFERENCES [dbo]. [Student] ([st_id])

GO

ALTER TABLE [dbo]. [Student_Exam_Question] ADD CONSTRAINT [FK_Student_Ex_Q_id_6754599E] FOREIGN

KEY ([Q_id]) REFERENCES [dbo]. [Question] ([Q_id]) ON DELETE CASCADE ON UPDATE CASCADE

GO

ALTER TABLE [dbo]. [Student_Exam_Question] ADD CONSTRAINT [FK_Student_E_ex_id_66603565] FOREIGN

KEY ([ex_id]) REFERENCES [dbo]. [Exam] ([ex_id])

GO
```

Uses

[dbo].[Exam]

[dbo].[Question]

[dbo].[Student]

Used By

[dbo].[Correct_Exam]

[dbo].[deleteGradeTernary]

[dbo].[Exam Answers]

[dbo].[GetExam_Questions_Model_stud_Answer]

[dbo].[GetGradesInAllCouresBystdId]

[dbo].[GetstdgradeTernary]

[dbo].[GetstdgradeTernaryby3 IDs]

[dbo].[GetstdgradeTernarybySt ID]

[dbo].[InsGradeTernary]

[dbo].[updateanswerTernary]

[dbo].[updateGradeTernary]

[dbo].[student_phone]

Properties

Property	Value
Row Count (~)	22
Created	20:00:12 04 December 2024
Last Modified	17:54:56 08 December 2024

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	
PK <mark>P</mark> FKP	st_id	int	4	NOT NULL	
PK <mark>P</mark> C	phone	int	4	NOT NULL	

Indexes

Key	Name	Key Columns	Unique
PK C	PK_student531D308A6CBC6DAD	st_id, phone	True

Foreign Keys

Name	Update	Delete	Columns
FK_student_p_st_id5629CD9C	Cascade	Cascade	st_id->[dbo].[Student].[st_id]

```
CREATE TABLE [dbo].[student_phone]

(
[st_id] [int] NOT NULL,
[phone] [int] NOT NULL

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[student_phone] ADD CONSTRAINT [PK_student__531D308A6CBC6DAD] PRIMARY KEY
CLUSTERED ([st_id], [phone]) ON [PRIMARY]

GO

ALTER TABLE [dbo].[student_phone] ADD CONSTRAINT [FK_student_p_st_id__5629CD9C] FOREIGN KEY
([st_id]) REFERENCES [dbo].[Student] ([st_id]) ON DELETE CASCADE ON UPDATE CASCADE

GO
```

Project > DESKTOP-3CE5M2E > User databases > ITIExamination-System > Tables > dbo.student_phone

Uses

[dbo].[Student]

Used By

[dbo].[AddPhone]
[dbo].[delphone]
[dbo].[getstdphone]
[dbo].[GetStudentdPhonebyld]
[dbo].[InsStdPhone]

[dbo].[Topic]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Неар	True
Row Count (~)	21
Created	22:03:13 09 December 2024
Last Modified	22:03:14 09 December 2024

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
	name	varchar(max)	max	NOT NULL
FK	C_id_fk	int	4	NOT NULL

Foreign Keys

Name	Update	Delete	Columns
FKTopicC_id_fk4CA06362	Cascade	Cascade	C_id_fk->[dbo].[Course].[C_ld]

SQL Script

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

(
[name] [varchar] (max) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,

[C_id_fk] [int] NOT NULL

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Topic] ADD CONSTRAINT [FK_ Topic_ C_id_fk_ 4CA06362] FOREIGN KEY ([C_id_fk])

REFERENCES [dbo].[Course] ([C_Id]) ON DELETE CASCADE ON UPDATE CASCADE

GO
```

Uses

[dbo].[Course]

Used By

[dbo].[GetTopicByCourseId]

Project > DESKTOP-3CE5M2E > User databases > ITIExamination-System > Tables > dbo.Topic

[dbo].[sp_Topic_Delete] [dbo].[sp_Topic_Insert] [dbo].[sp_Topic_SelectAll] [dbo].[sp_Topic_SelectByPK] [dbo].[sp_Topic_Update]



Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	3
Created	17:41:35 08 December 2024
Last Modified	17:54:56 08 December 2024

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK2 C	T_id	int	4	NOT NULL	1000 - 1000
	T_name	varchar(max)	max	NULL allowed	
	description	varchar(max)	max	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
PKO	PKtrack83B8138AEE8EB957	T_id	True

SQL Script

```
CREATE TABLE [dbo].[track]

(
[T_id] [int] NOT NULL IDENTITY(1000, 1000),

[T_name] [varchar] (max) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,

[description] [varchar] (max) COLLATE SQL_Latin1_General_CP1_CI_AS NULL

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[track] ADD CONSTRAINT [PK_track_83B8138AEE8EB957] PRIMARY KEY CLUSTERED

([T_id]) ON [PRIMARY]

GO
```

Used By

[dbo].[CourseHasTrack] [dbo].[InstWorksInTrack] [dbo].[Student] [dbo].[track_branch] [dbo].[DeleteInst]

[dbo].[GetCourseAndNumOfStudByInstructorId]

[dbo].[InsertInto_InsWorksInTrack]

[dbo].[InsertIntoCourseHasTrack]

[dbo].[UpdateCourseHasTrack]

[dbo].[updateInstWorksInTrack]

[dbo].[usp_Student_Insert]

[dbo].[usp_Student_Update]

[dbo].[usp_Track_Delete]

[dbo].[usp_Track_Insert]

[dbo].[usp_Track_SelectAll]

[dbo].[usp_Track_SelectById]

[dbo].[usp_Track_Update]

[dbo].[usp_TrackBranch_Insert]

[dbo].[usp_TrackBranch_Update]

[dbo].[track_branch]

Properties

Property	Value
Row Count (~)	8
Created	20:00:12 04 December 2024
Last Modified	17:41:36 08 December 2024

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PKOFKO C	B_id_fk	int	4	NOT NULL
PKPFKP C	T_id_fk	int	4	NOT NULL

Indexes

Key	Name	Key Columns	Unique	
PK C	PKtrack_br2F26A121581D7F3D	B_id_fk, T_id_fk	True	

Foreign Keys

Name		Delete	Columns	
FKtrack_braT_id3E52440B	Cascade	Cascade	T_id_fk->[dbo].[track].[T_id]	
FKtrack_braB_id3D5E1FD2	Cascade	Cascade	B_id_fk->[dbo].[branch].[B_id]	

```
CREATE TABLE [dbo].[track_branch]

(
[B_id_fk] [int] NOT NULL,

[T_id_fk] [int] NOT NULL

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[track_branch] ADD CONSTRAINT [PK_track_br_2F26A121581D7F3D] PRIMARY KEY CLUSTERED ([B_id_fk], [T_id_fk]) ON [PRIMARY]

GO

ALTER TABLE [dbo].[track_branch] ADD CONSTRAINT [FK_track_bra_T_id_3E52440B] FOREIGN KEY ([T_id_fk]) REFERENCES [dbo].[track] ([T_id]) ON DELETE CASCADE ON UPDATE CASCADE

GO

ALTER TABLE [dbo].[track_branch] ADD CONSTRAINT [FK_track_bra_B_id_3D5E1FD2] FOREIGN KEY ([B_id_fk]) REFERENCES [dbo].[branch] ([B_id]) ON DELETE CASCADE ON UPDATE CASCADE
```

GO

Uses

[dbo].[branch] [dbo].[track]

Used By

[dbo].[sp_TrackBranch_Delete] [dbo].[usp_TrackBranch_Insert] [dbo].[usp_TrackBranch_SelectAll] [dbo].[usp_TrackBranch_SelectByPK] [dbo].[usp_TrackBranch_Update]

■ Stored Procedures

Objects

Name
dbo.AddPhone
dbo.Correct_Exam
dbo.DeleteCourseData
dbo.DeleteCourseHasTrack
dbo.deleteGradeTernary
dbo.DeleteInst
dbo.DeleteInstWorksInTrack
dbo.delExam
dbo.delphone
dbo.Exam_Answers
dbo.ExamGeneration
dbo.ExQ_Delete
dbo.ExQ_get_sp
dbo.ExQ_Insert
dbo.ExQ_Update
dbo.GetCourseAndNumOfStudByInstructorId
dbo.GetCourseData
dbo.GetCourseTrack
dbo.GetExam_Questions_Model_stud_Answer
dbo.GetExamDatabyId
dbo.GetGradesInAllCouresBystdId
dbo.GetInstData
dbo.GetInstTrack
dbo.GetQuestionsByExamNum
dbo.GetstdgradeTernary
dbo.GetstdgradeTernaryby3_IDs
dbo.GetstdgradeTernarybySt_ID
dbo.getstdphone
dbo.GetStudentdPhonebyId
dbo.GetStudentsByDeptNum
dbo.GetTopicByCourseId
dbo.lnsertExam
dbo.lnsertInto_InsWorksInTrack
dbo.InsertIntoCourse

dbo.InsertIntoCourseHasTrack
dbo.InsGradeTernary
dbo.InsStdPhone
dbo.Q_Delete
dbo.Q_Get_sp
dbo.Q_Insert
dbo.Q_Update
dbo.QCh_Delete
dbo.QCh_Get
dbo.QCh_Insert
dbo.QCh_Update
dbo.selexam
dbo.sp_InsertIntoInst
dbo.sp_InstructorTeachCource_Delete
dbo.sp_InstructorTeachCource_Insert
dbo.sp_InstructorTeachCource_SelectByPK
dbo.sp_Topic_Delete
dbo.sp_Topic_Insert
dbo.sp_Topic_SelectAll
dbo.sp_Topic_SelectByPK
dbo.sp_Topic_Update
dbo.sp_TrackBranch_Delete
dbo.updateanswerTernary
dbo.updateCourse
dbo.UpdateCourseHasTrack
dbo.updateExam
dbo.updateGradeTernary
dbo.UpdateInst
dbo.updateInstWorksInTrack
dbo.usp_Branch_Delete
dbo.usp_Branch_Insert
dbo.usp_Branch_SelectAll
dbo.usp_Branch_SelectById
dbo.usp_Branch_Update
dbo.usp_BranchPhone_Delete
dbo.usp_BranchPhone_Insert
dbo.usp_BranchPhone_SelectAll
dbo.usp_BranchPhone_SelectByBranch
dbo.usp_InstructorTeachCource_SelectAll
dbo.usp_InstructorTeachCource_Update
dbo.usp_Student_Delete

dbo.usp_Student_Insert
dbo.usp_Student_SelectAll
dbo.usp_Student_SelectByPK
dbo.usp_Student_Update
dbo.usp_Track_Delete
dbo.usp_Track_Insert
dbo.usp_Track_SelectAll
dbo.usp_Track_SelectById
dbo.usp_Track_Update
dbo.usp_TrackBranch_Insert
dbo.usp_TrackBranch_SelectAll
dbo.usp_TrackBranch_SelectByPK
dbo.usp_TrackBranch_Update

[dbo].[AddPhone]

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 proc [dbo].[AddPhone] @st_id int , @newphone int
as
begin
if exists (select st_id from student_phone)
begin
update student_phone
set
phone = @newphone
where st_id = @st_id
end
else
select ' This is Invalid ID'
end
GO
```

Uses

[dbo].[student_phone]

[dbo].[Correct_Exam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@ex_id	int	4
@student_name	nvarchar(50)	100

```
CREATE PROC [dbo].[Correct_Exam]
   @ex id INT,
   @student name NVARCHAR(50)
AS
BEGIN
   BEGIN TRY
       -- Parse the student's first and last name
       DECLARE @fname NVARCHAR(20) = PARSENAME(REPLACE(@student name, ' ', '.'), 2);
       DECLARE @lname NVARCHAR(20) = PARSENAME(REPLACE(@student name, ' ', '.'), 1);
        -- Retrieve the student's ID
       DECLARE @st_id INT;
        SELECT @st_id = st id
        FROM Student
       WHERE fname = @fname AND lname = @lname;
        -- Validate if the student exists
       IF @st_id IS NULL
        BEGIN
           SELECT 'Student name not found' AS ErrorMessage;
           RETURN;
        END
        -- Validate if the exam exists
        IF NOT EXISTS (SELECT 1 FROM Exam WHERE ex id = @ex id)
           SELECT 'Exam ID is not valid' AS ErrorMessage;
            RETURN;
        END
```

```
-- Validate if the student has answers for the exam
        IF NOT EXISTS (SELECT 1 FROM Student Exam Question WHERE st id = @st id AND ex id =
@ex id)
        BEGIN
           SELECT 'No answers found for the student in the given exam' AS ErrorMessage;
           RETURN;
        END
        -- Update grades for the student's answers
        UPDATE SEO
        SET SEQ.grade = CASE
                           WHEN SEQ.Answer = Q.model_answer THEN (SELECT total_score FROM Exam
WHERE ex id=@ex id) *.1
                           ELSE 0
                        END
        FROM Student_Exam_Question SEQ
        INNER JOIN Question Q ON SEQ.Q id = Q.Q id
        WHERE SEQ.st id = @st id AND SEQ.ex id = @ex id;
        declare @percentage int
       SET @percentage = ((SELECT SUM(grade) FROM Student_Exam_Question WHERE ex_id=@ex_id AND
st_id=@st_id GROUP BY ex_id,st_id)*(SELECT total_score FROM Exam WHERE ex_id=@ex_id))/100
       SELECT 'Exam correction completed successfully' AS SuccessMessage,
CONCAT(@percentage,'%') As Result ;
   END TRY
   BEGIN CATCH
      SELECT ERROR MESSAGE() AS ErrorMessage;
   END CATCH
END;
GO
```

Uses

```
[dbo].[Exam]
[dbo].[Question]
[dbo].[Student]
[dbo].[Student_Exam_Question]
```

[dbo].[DeleteCourseData]

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].[DeleteCourseData] @id int
as
   if not exists (select C_Id from Course where C_Id = @id)
   select 'This Id Dose Not Exists.'
   else
   delete from Course where C_Id = @id
GO
```

Uses

[dbo].[Course]

[dbo].[DeleteCourseHasTrack]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@crsid	int	4
@trackid	int	4

SQL Script

```
create proc [dbo].[DeleteCourseHasTrack]
@crsid int,
@trackid int

as
   if not exists (select * from CourseHasTrack where C_Id = @crsid and Track_Id = @trackid)
   select 'This Id Dose Not Exists.'
   else
   delete from CourseHasTrack where C_Id = @crsid and Track_Id = @trackid
GO
```

Uses

[dbo].[CourseHasTrack]

Project > DESKTOP-3CE5M2E > User databases > ITIExamination-System > Programmability > Stored Procedures > dbo.deleteGrade-Ternary

[dbo].[deleteGradeTernary]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

Uses

[dbo].[Student_Exam_Question]

[dbo].[DeleteInst]

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].[DeleteInst] @id int
as
begin
    if not exists (select Ins_Id from Instructor where Ins_Id = @id)
    begin
        select 'Instructor not found'
    end
    if exists (select T_Id from Track where T_Id in
        (select T_Id from InstWorksInTrack where Ins_Id = @id))
    begin
        select 'Cannot delete. This instructor is associated with a track.'
    end
    if exists (select C_Id from Course where C_Id in
        (select C_Id from Instructor_Teach_Cource where ins_id_fk = @id))
    begin
        Select 'Cannot delete. This instructor is teaching a course.'
    end
    delete from Instructor where Ins_Id = @id
    select 'Instructor deleted successfully.'
end
GO
```

Uses

[dbo].[Course]
[dbo].[Instructor]
[dbo].[Instructor_Teach_Cource]
[dbo].[InstWorksInTrack]
[dbo].[track]

[dbo].[DeleteInstWorksInTrack]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@instid	int	4
@trackid	int	4

SQL Script

```
create proc [dbo].[DeleteInstWorksInTrack]
@instid int,
@trackid int
  if not exists (select ins_id from InstWorksInTrack where ins_id = @instid and track_id
=@trackid)
  select 'Sorry this record dose not exist.'
 delete from InstWorksInTrack where ins_id = @instid and track_id =@trackid
```

Uses

[dbo].[InstWorksInTrack]

[dbo].[delExam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
create proc [dbo].[delExam] @ex_id int
as
begin
if @ex_id is not null and @ex_id in (select ex_id from Exam )
begin
delete from Exam
where ex_id = @ex_id
select 'Delete Exam successfully'
end
else
begin
select 'This Exam Not Found '
end
end
GO
```

Uses

[dbo].[Exam]

[dbo].[delphone]

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].[delphone] @id int
as
begin
    if @id is not null and @id in (select st_id from student_phone)
    begin
        delete from student_phone
        where st_id = @id
    end
    else
    select ' Can not Delete Invalid ID'
end
GO
```

Uses

[dbo].[student_phone]

[dbo].[Exam_Answers]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@ex_id	int	4
@name	varchar(50)	50
@A1	varchar(20)	20
@A2	varchar(20)	20
@A3	varchar(20)	20
@A4	varchar(20)	20
@A5	varchar(20)	20
@A6	varchar(20)	20
@A7	varchar(20)	20
@A8	varchar(20)	20
@A9	varchar(20)	20
@A10	varchar(20)	20

```
CREATE PROC [dbo].[Exam_Answers]
    @ex_id INT,
    @name VARCHAR(50),
    @A1 VARCHAR(20), @A2 VARCHAR(20), @A3 VARCHAR(20), @A4 VARCHAR(20),
    @A5 VARCHAR(20), @A6 VARCHAR(20), @A7 VARCHAR(20), @A8 VARCHAR(20),
    @A9 VARCHAR(20), @A10 VARCHAR(20)

AS

BEGIN

DECLARE @StuAns TABLE (ex_id INT, Answer VARCHAR(20));

INSERT INTO @StuAns(ex_id, Answer)

VALUES

    (@ex_id, @A1), (@ex_id, @A2), (@ex_id, @A3), (@ex_id, @A4), (@ex_id, @A5),
    (@ex_id, @A6), (@ex_id, @A7), (@ex_id, @A8), (@ex_id, @A9), (@ex_id, @A10);

BEGIN TRY

DECLARE @st_id INT;
```

```
SET @st id = (
    SELECT st id
   FROM Student
    WHERE
        fname = (SELECT PARSENAME(REPLACE(@name, ' ', '.'), 2)) AND
       lname = (SELECT PARSENAME(REPLACE(@name, ' ', '.'), 1))
);
-- Validate Exam ID
IF NOT EXISTS (SELECT ex id FROM Exam WHERE ex id = @ex id)
   SELECT 'Exam ID is not valid';
   RETURN;
END;
-- Validate Student ID
IF @st id IS NULL
   SELECT 'Your name does not exist';
   RETURN;
END;
-- Validate Questions for the Exam
IF NOT EXISTS (SELECT Q id FROM Exam has Quesstion WHERE ex id = @ex id)
   SELECT 'No questions found for the given Exam ID';
   RETURN;
END;
-- Cursor declarations
DECLARE c1 CURSOR LOCAL FAST FORWARD FOR
   SELECT Q id
    FROM Exam has Quesstion
    WHERE ex id = @ex id;
DECLARE c2 CURSOR LOCAL FAST FORWARD FOR
   SELECT Answer
   FROM @StuAns;
DECLARE @Q id INT;
DECLARE @Answer VARCHAR(20);
-- Open cursors
OPEN c1;
OPEN c2;
FETCH NEXT FROM c1 INTO @Q id;
FETCH NEXT FROM c2 INTO @Answer;
WHILE @@FETCH STATUS = 0
    INSERT INTO Student_Exam_Question (st_id, ex_id, Q_id, Answer)
```

```
VALUES (@st_id, @ex_id, @Q_id, @Answer);

FETCH NEXT FROM c1 INTO @Q_id
FETCH NEXT FROM c2 INTO @Answer
END

-- Close and deallocate cursors
CLOSE c1;
DEALLOCATE c1;

CLOSE c2;
DEALLOCATE c2;

END TRY
BEGIN CATCH
SELECT ERROR_MESSAGE() AS ErrorMessage;
END CATCH
END;
GO
```

Uses

[dbo].[Exam]
[dbo].[Exam_has_Quesstion]
[dbo].[Student]
[dbo].[Student_Exam_Question]

[dbo].[ExamGeneration]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@CrsName	varchar(50)	50
@NumOf_MCQ	int	4
@Numof_TF	int	4
@ExamName	varchar(50)	50

```
create proc [dbo].[ExamGeneration]
    @CrsName VARCHAR(50),
    @NumOf_MCQ INT,
    @Numof TF INT,
    @ExamName VARCHAR(50)
AS
BEGIN TRY
   BEGIN
        DECLARE @Crs id INT;
        SET @Crs_id = (SELECT C_id FROM Course WHERE C_Name = @CrsName);
        IF @Crs_id IS NULL
        BEGIN
            SELECT 'This Course Does Not Exist.' AS ErrorMessage;
             RETURN;
        END
         --validate total no of questions
        \  \  \text{if} \ \ (@NumOf\_MCQ \ + \ @Numof\_TF) \ > \ 10 \ \ \text{OR} \ \ (@NumOf\_MCQ \ + \ @Numof\_TF) \ < \ 10 \\
            select 'The total number of questions (MCQ + T/F) must equql 10.' AS ErrorMessage;
             return;
         end
```

```
if (SELECT COUNT(*) FROM Question WHERE C Id = @Crs id AND type = 'MCQ') < @NumOf MCQ
           select 'Not enough MCQ questions available for this course.' as ErrorMessage;
           return;
        end
       if (SELECT COUNT(*) FROM Question WHERE C Id = @Crs id AND type = 'T/F') < @Numof TF
           select 'Not enough T/F questions available for this course.' AS ErrorMessage;
       END
      declare @ExamId INT;
      select @ExamId = ISNULL(MAX(Ex_Id), 0) + 100 FROM Exam;
       insert into Exam (Ex id, Ex name, Ex date, C id, total score)
      values (@ExamId, @ExamName, GETDATE(), @Crs id, 100);
       insert into Exam_Has_Quesstion (Ex_id, Q_id)
       SELECT TOP (@NumOf MCQ) @ExamId, Q id
        FROM Ouestion
       WHERE C Id = @Crs id AND type = 'MCQ'
       ORDER BY NEWID();
       INSERT INTO Exam Has Quesstion (Ex id, Q id)
       SELECT TOP (@Numof TF) @ExamId, Q id
       FROM Question
       WHERE C Id = @Crs id AND type = 'T/F';
       SELECT 'Exam created successfully!' AS SuccessMessage;
   END
END TRY
BEGIN CATCH
   SELECT 'ERROR! Invalid Data.' AS ErrorMessage;
END CATCH;
GO
```

Uses

[dbo].[Course]
[dbo].[Exam]
[dbo].[Exam_has_Quesstion]
[dbo].[Question]

[dbo].[ExQ_Delete]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@ex_id	int	4
@Q_id	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[ExQ_Delete] @ex_id int, @Q_id int

AS

BEGIN

if exists ( select ex_id from Exam_has_Quesstion where ex_id =@ex_id and Q_id = @Q_id)

begin

DELETE FROM Exam_has_Quesstion

WHERE ex_id = @ex_id AND Q_id = @Q_id

end
else
select'This Record Does Not Exist.'

END

GO
```

Uses

[dbo].[Exam_has_Quesstion]

[dbo].[ExQ_get_sp]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@ex_id	int	4
@Q_id	int	4

SQL Script

```
CREATE Procedure [dbo].[ExQ_get_sp] @ex_id int=null, @Q_id int=null

AS

BEGIN

if exists ( select 1 from Exam_has_Quesstion where ex_id = @ex_id and Q_id = @Q_id)

begin

SELECT * FROM Exam_has_Quesstion

WHERE ex_id = isnull(@ex_id,ex_id) AND Q_id = isnull(@Q_id, Q_id)

end

else

select'This Record Does not exist.'

END

GO
```

Uses

[dbo].[Exam_has_Quesstion]

[dbo].[ExQ_Insert]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@ex_id	int	4
@Q_id	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[ExQ Insert] @ex id int, @Q id int
AS
BEGIN
   IF EXISTS (SELECT 1 FROM Exam_has_Quesstion WHERE ex_id = @ex_id AND Q_id = @Q_id)
    BEGIN
    SELECT 'this key already Exists'
    END
    else if not exists (select ex id from Exam where ex id =@ex id)
    select 'This Exam is not exist.'
    else if not exists (select q_id from Question where Q_id =@Q_id)
    select 'This Question is not exist.'
    else
   INSERT INTO Exam has Quesstion(ex id, Q id)
   VALUES (@ex_id, @Q_id)
END
GO
```

Uses

```
[dbo].[Exam]
[dbo].[Exam_has_Quesstion]
[dbo].[Question]
```

[dbo].[ExQ_Update]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@old_ex_id	int	4
@old_Q_id	int	4
@new_ex_id	int	4
@new_Q_id	int	4

```
CREATE PROCEDURE [dbo].[ExQ_Update] @old_ex_id int ,@old_Q_id int, @new_ex_id int=null, @new_Q_id
int=null
BEGIN try
   IF @new ex id IS NOT NULL AND @new Q id IS NOT NULL
   BEGIN
       IF EXISTS (
           SELECT 1
           FROM Exam_has_Quesstion
            WHERE ex id = @new ex id AND Q id = @new Q id
       BEGIN
           Select 'New composite key already exists.'
       END
       else if not exists (select ex_id from Exam where ex_id =@old_ex_id)
       select 'This Exam is not exist.'
       else if not exists (select q id from Question where Q id =@old Q id)
       select 'This Question is not exist.'
       else
       begin
       UPDATE Exam has Quesstion
       SET ex id = @new ex id,
          Q_id = @new_Q_id
       WHERE ex id = @old ex id AND Q id = @old Q id
        end
     END
End try
```

Project > DESKTOP-3CE5M2E > User databases > ITIExamination-System > Programmability > Stored Procedures > dbo.ExQ_Update

```
begin catch
select'ERROR! Invalid Data.'
end catch
GO
```

Uses

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

[dbo].[GetCourseAndNumOfStudByInstructorId]

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 proc [dbo].[GetCourseAndNumOfStudByInstructorId] @ins_id int
   select c.C Name , count(s.st id) as 'Number of Students'
   from Instructor_Teach_Cource itc
   inner join Course c
   on c.C_Id =itc.C_id_fk and itc.ins_id_fk =@ins_id
   inner join CourseHasTrack ct
   on c.C_Id=ct.C_Id
   inner join track t
   on t.T_id=ct.Track_Id
   inner join Student s
   on t.T_id=s.T_id_fk
   group by c.C_Name
GO
```

Uses

[dbo].[Course] [dbo].[CourseHasTrack] [dbo].[Instructor_Teach_Cource] [dbo].[Student] [dbo].[track]

[dbo].[GetCourseData]

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].[GetCourseData] @id int = Null
begin
  if @id is Null
  select * from Course
  else
  if @id is Not Null and @id in (select C_Id from Course)
       select * from Course where C_Id = @id
      end
   else
       begin
        select'This Id Dose Not Exist'
       end
   end
end
GO
```

Uses

[dbo].[Course]

[dbo].[GetCourseTrack]

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].[GetCourseTrack] @id int = Null
as
begin
  if @id is Null
  select * from CourseHasTrack
  else
  begin
  if @id is Not Null and @id in (select C_Id from CourseHasTrack)
  begin
    select * from Course where C_Id = @id
  end
  end
  else
    begin
    select'This Id Dose Not Exist'
  end
end
end
end
GO
```

Uses

[dbo].[Course] [dbo].[CourseHasTrack]

[dbo].[GetExam_Questions_Model_stud_Answer]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@ExamNum	int	4
@StudId	int	4

SQL Script

```
CREATE proc [dbo].[GetExam_Questions_Model_stud_Answer] @ExamNum int ,@StudId int as

select distinct(q.title),q.model_answer,sq.Answer
from Exam_has_Quesstion exq inner join Question q
on q.Q_id=exq.Q_id and exq.ex_id=@ExamNum
inner join Question_choice c
on c.Q_id=q.Q_id
inner join Student_Exam_Question sq
on sq.Q_id=q.Q_id and sq.st_id=@StudId and sq.ex_id=@ExamNum

GO
```

Uses

[dbo].[Exam_has_Quesstion]
[dbo].[Question]
[dbo].[Question_choice]
[dbo].[Student_Exam_Question]

[dbo].[GetExamDatabyld]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
create proc [dbo].[GetExamDatabyId] @ex_id int
select * from Exam
else if not exists (select ex_id from Exam where ex_id=@ex_id)
select ' This Exam Not found'
select 'Invalid Exaam ID '
end
GO
```

Uses

[dbo].[Exam]



[dbo].[GetGradesInAllCouresBystdId]

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].[GetGradesInAllCouresBystdId] @st id int
   select c.C Name , SUM(sq.grade)'grade' --/e.total score *100 as
   from Student_Exam_Question sq,Exam e,Course c
   where sq.ex_id=e.ex_id and e.c_id=c.C_Id and sq.st_id=@st_id
   group by c.C_Name
GO
```

Uses

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

[dbo].[GetInstData]

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].[GetInstData] @id int = Null
as
begin
   if @id is Null
   select * from Instructor
   else
   begin
   if @id is Not Null and @id in (select Ins_Id from Instructor)
        begin
        select * from Instructor where Ins_Id = @id
        end
        else
        begin
        select * from Instructor where Ins_Id = @id
        end
        end
        else
        begin
            select'This Id Dose Not Exist.'
        end
        end
end
end
end
```

Uses

[dbo].[Instructor]

[dbo].[GetInstTrack]

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].[GetInstTrack] @id int = NULL
as
begin
  if @id is Null
  select * from InstWorksInTrack
  else
  begin
  if @id is Not Null and @id in (select Ins_Id from InstWorksInTrack)
  begin
    select * from InstWorksInTrack where Ins_Id = @id
  end
  end
  else
    begin
    select'This Id Dose Not Exist'
  end
  end
end
end
GO
```

Uses

[dbo].[InstWorksInTrack]

[dbo].[GetQuestionsByExamNum]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
create proc [dbo].[GetQuestionsByExamNum] @ExamNum int
as
    select q.title,c.choice,c.textOfChoice
    from Exam_has_Quesstion ex inner join Question q
    on q.Q_id=ex.Q_id and ex.ex_id=@ExamNum
    inner join Question_choice c
    on c.Q_id=q.Q_id
GO
```

Uses

[dbo].[Exam_has_Quesstion] [dbo].[Question] [dbo].[Question_choice] Project > DESKTOP-3CE5M2E > User databases > ITIExamination-System > Programmability > Stored Procedures > dbo.Getstdgrade-



[dbo].[GetstdgradeTernary]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
create proc [dbo].[GetstdgradeTernary]
select * from Student_Exam_Question
```

Uses

[dbo].[Student_Exam_Question]

[dbo].[GetstdgradeTernaryby3_IDs]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
create proc [dbo].[GetstdgradeTernaryby3_IDs] @id int , @ex_id int , @q_id int
as
begin
if( @id is not null and @id in (select st_id from Student_Exam_Question)
and @ex_id is not null and @ex_id in (select ex_id from Exam_has_Question)
and @q_id is not null and @q_id in (select Q_id from Student_Exam_Question)
)
begin
select * from Student_Exam_Question
end
else
select 'Not Found This ID'
end
GO
```

Uses

[dbo].[Exam_has_Quesstion]
[dbo].[Student_Exam_Question]

[dbo].[GetstdgradeTernarybySt_ID]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
create proc [dbo].[GetstdgradeTernarybySt_ID]
@st id int
as
begin
select * from Student_Exam_Question
where st_id =@st_id
GO
```

Uses

[dbo].[Student_Exam_Question]

[dbo].[getstdphone]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
create proc [dbo].[getstdphone]
as
select * from student_phone
GO
```

Uses

[dbo].[student_phone]

[dbo].[GetStudentdPhonebyld]

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].[GetStudentdPhonebyId] @id int
if @id is not null and @id in (select st_id from student_phone)
select * from student_phone
where st id = @id
end
select NULL as st_id , NULL as phone , 'This Id Not Found' as message
GO
```

Uses

[dbo].[student_phone]

Project > DESKTOP-3CE5M2E > User databases > ITIExamination-System > Programmability > Stored Procedures > dbo.GetStudents-ByDeptNum



[dbo].[GetStudentsByDeptNum]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
create proc [dbo].[GetStudentsByDeptNum] @dnum int
   select *
   from Student
   where T_id_fk=@dnum
GO
```

Uses

[dbo].[Student]

[dbo].[GetTopicByCourseld]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
create proc [dbo].[GetTopicByCourseId] @course_id int
as
    select t.name
    from Topic t
    where t.C_id_fk=@course_id
GO
```

Uses

[dbo].[Topic]

[dbo].[InsertExam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@ex_id	int	4
@ex_name	varchar(50)	50
@ex_date	date	3
@crs_id	int	4
@totalscore	int	4

SQL Script

```
create proc [dbo].[InsertExam] @ex_id int , @ex_name varchar(50) , @ex_date date , @crs_id int ,
@totalscore int
begin
begin try
if exists(select ex_id from exam where ex_id=@ex_id)
select'Dublicate Exam id'
else if not exists(select c id from course where c id=@crs id)
select'corse id does not exist'
begin
insert into Exam (ex id , ex name , ex date , c id , total score)
values(@ex_id , @ex_name , @ex_date , @crs_id , @totalscore)
end
end try
begin catch
select 'ERROR!!'
end catch
end
GO
```

Uses

[dbo].[Course]

Project > DESKTOP-3CE5M2E > User databases > ITIExamination-System > Programmability > Stored Procedures > dbo.InsertExam

[dbo].[Exam]

[dbo].[InsertInto_InsWorksInTrack]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@insld	int	4
@trackId	int	4
@hiringdate	date	3

SQL Script

```
create proc [dbo].[InsertInto_InsWorksInTrack]
@insId int,
@trackId int,
@hiringdate date
begin
  begin try
  begin
  if not exists (select T Id from Track where T Id = @trackid)
  select'Sorry This Track Dose Not Exist.'
  end
  if not exists (select Ins_Id from Instructor where Ins_Id = @insId)
  select'Sorry This Instructor Dose Not Exist.'
  if exists (select Ins_Id from InstWorksInTrack where Ins_Id = @insId and Track_Id = @trackid)
  select'This Record Alraedy Exists.'
  insert into InstWorksInTrack (Ins_ID , Track_id , hiringdate)
  values (@insId,@trackId,@hiringdate)
  end try
  begin catch
  select 'Invalid data'
  end catch
```

Project > DESKTOP-3CE5M2E > User databases > ITIExamination-System > Programmability > Stored Procedures > dbo.InsertInto_-InsWorksInTrack

GO

Uses

[dbo].[Instructor] [dbo].[InstWorksInTrack] [dbo].[track]

[dbo].[InsertIntoCourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@name	varchar(20)	20
@duration	int	4
@desc	varchar(50)	50

SQL Script

Uses

[dbo].[Course]

[dbo].[InsertIntoCourseHasTrack]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Courseld	int	4
@TrackId	int	4

SQL Script

```
create proc [dbo].[InsertIntoCourseHasTrack]
@CourseId int,
@TrackId int
  if exists (select C id from CourseHasTrack where c id=@CourseId and Track Id=@TrackId)
  select'This Record Already Exists.'
  end
  else
  if exists (select C_id from Course where C_id = @CourseId)
     if exists (select T_id from track where T_id = @TrackId)
             insert into CourseHasTrack (C_id,Track_Id)
             values (@courseId,@TrackId)
           end
       else
       select'This Id Dose Not Match Either Course Or Track.'
     else
     select'This Id Dose Not Match Either Course Or Track.'
GO
```

Project > DESKTOP-3CE5M2E > User databases > ITIExamination-System > Programmability > Stored Procedures > dbo.InsertInto-CourseHasTrack

Uses

[dbo].[Course] [dbo].[CourseHasTrack] [dbo].[track]

[dbo].[InsGradeTernary]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@St_id	int	4
@ex_id	int	4
@Q_id	int	4
@grade	int	4
@answer	varchar(20)	20

SQL Script

Uses

```
[dbo].[Exam]
[dbo].[Question]
[dbo].[Student]
[dbo].[Student_Exam_Question]
```

[dbo].[InsStdPhone]

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 proc [dbo].[InsStdPhone] @phone int , @id int
as
begin
if @id is not null and @id in(select st_id from student)
begin
insert into student_phone (st_id, phone)
values (@id, @phone)
end
else
begin
select'This Student Does Not Exist.'
end
end
GO
```

Uses

[dbo].[Student]
[dbo].[student_phone]

[dbo].[Q_Delete]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE PROCEDURE [dbo].[Q_Delete] @Q_id int

AS

BEGIN

if exists (select 1 from Question where Q_id=@Q_id)

begin

DELETE FROM Question

WHERE Q_id = @Q_id

end

else

select'This Record Does not exist.'

END

GO
```

Uses

[dbo].[Question]

[dbo].[Q_Get_sp]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
--select

CREATE PROCEDURE [dbo].[Q_Get_sp] @Q_id int =null

AS

BEGIN

if exists (select 1 from Question where Q_id = @Q_id)

begin

SELECT * FROM Question

WHERE Q_id = isnull(@Q_id,Q_id)

end

else

select'This Record Does Not Exist.'

END

GO
```

Uses

[dbo].[Question]

[dbo].[Q_Insert]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Q_id	int	4
@type	varchar(20)	20
@title	nvarchar(max)	max
@model_ans	nvarchar(max)	max
@c_id	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[Q_Insert] @Q_id int, @type varchar(20),@title nvarchar(max),@model_ans nvarchar(max),@c_id int

AS

begin

BEGIN try

IF EXISTS (SELECT 1 FROM Question WHERE Q_id = @Q_id)

BEGIN

SELECT 'this key already Exists'

END

INSERT INTO Question(Q_id, type,title,model_answer,c_id)

VALUES (@Q_id, @type, @title, @model_ans, @c_id)

END try

begin catch

select'ERROR! Invalid Data.'
end catch
end
GO
```

Uses

[dbo].[Question]

[dbo].[Q_Update]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Q_id	int	4
@type	varchar(20)	20
@title	nvarchar(max)	max
@model_ans	nvarchar(max)	max
@c_id	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[Q_Update] @Q_id int, @type varchar(20)=null, @title nvarchar(max)=null,
@model_ans nvarchar(max) = null, @c_id int = null
BEGIN
  if not exists (select 1 from Question where Q_id = @Q_id)
   select'This Record Does Not Exist.'
   else if exists (select 1 from course where c_id= @C_id)
   begin
   UPDATE Question
   SET type = isnull(@type, type),
       title = isnull(@title, title),
       model answer = isnull(@model ans, model answer),
       c_id = isnull(@c_id, c_id)
   WHERE Q_id = @Q_id
    end
END
GO
```

Uses

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

[dbo].[QCh_Delete]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Q_id	int	4
@choice	varchar(20)	20

SQL Script

```
CREATE PROCEDURE [dbo].[QCh_Delete] @Q_id int, @choice varchar(20)

AS

BEGIN

IF EXISTS(SELECT 1 FROM Question_choice WHERE Q_id= @Q_id AND choice = @choice)

begin

DELETE FROM Question_choice

WHERE Q_id = @Q_id AND choice = @choice

end

else

select'Record you try to delete does not exist.'

END

GO
```

Uses

[dbo].[Question_choice]

[dbo].[QCh_Get]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Q_id	int	4
@choice	varchar(20)	20

SQL Script

```
CREATE PROCEDURE [dbo].[QCh_Get] @Q_id int=null, @choice varchar(20)=null

AS

BEGIN

if exists (select 1 from Question_choice where Q_id=@Q_id and choice=@choice)

begin

SELECT * FROM Question_choice

WHERE Q_id = ISNULL(@Q_id, Q_id) AND choice = ISNULL(@choice, choice)

end

else

select'This Record Does Not Exist'

END

GO
```

Uses

[dbo].[Question_choice]

[dbo].[QCh_Insert]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Q_id	int	4
@choice	varchar(20)	20
@text	nvarchar(max)	max

SQL Script

```
CREATE PROCEDURE [dbo].[QCh_Insert] @Q_id int, @choice varchar(20), @text nvarchar(max)

AS

BEGIN

IF EXISTS(SELECT 1 FROM Question_choice WHERE Q_id= @Q_id AND choice = @choice)

BEGIN

SELECT 'this key already exists'

END

else if exists (select 1 from Question where Q_id = @Q_id)

begin

INSERT INTO Question_choice( Q_id, choice, textOfChoice)

VALUES (@Q_id, @choice, @text)

end

else

select'This Question Does Not Exist.'

END

GO
```

Uses

[dbo].[Question]
[dbo].[Question_choice]

[dbo].[QCh_Update]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Q_id	int	4
@choice	varchar(20)	20
@text	nvarchar(max)	max

SQL Script

```
CREATE PROCEDURE [dbo].[QCh_Update] @Q_id int, @choice varchar(20), @text nvarchar(max)

AS

BEGIN

IF Not EXISTS(SELECT 1 FROM Question_choice WHERE Q_id= @Q_id AND choice = @choice)

select'This Record you trying to update does not exist.'

else if not exists (select Q_id from Question where Q_id = @Q_Id)

select'This Question Does Not Exist.'

else

begin

UPDATE Question_choice

SET textOfChoice = isnull(@text, textOfChoice)

WHERE Q_id = @Q_id AND choice = @choice

end

END

GO
```

Uses

[dbo].[Question]
[dbo].[Question_choice]



Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
--1- sp select all without any condition
create proc [dbo].[selexam]
as
select * from Exam
GO
```

Uses

[dbo].[Exam]

Project > DESKTOP-3CE5M2E > User databases > ITIExamination-System > Programmability > Stored Procedures > dbo.sp_InsertInto-Inst

[dbo].[sp_InsertIntoInst]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

Uses

[dbo].[Instructor]

[dbo].[sp_InstructorTeachCource_Delete]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@ins_id_fk	int	4
@C_id_fk	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[sp_InstructorTeachCource_Delete]
         @ins id fk INT ,
         @C id fk INT
      AS
      BEGIN
         if (@ins_id_fk in (select i.ins_id_fk from Instructor_Teach_Cource i)
          and @C_id_fk in (select i.C_id_fk from Instructor_Teach_Cource i))
               begin
               DELETE FROM Instructor_Teach_Cource
               WHERE ins_id_fk = @ins_id_fk AND C_id_fk = @C_id_fk;
               select 'Delete completed successfully.'
          end
          else
               select('This Table is Not Exist')
      END
GO
```

Uses

[dbo].[Instructor_Teach_Cource]

[dbo].[sp_InstructorTeachCource_Insert]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@ins_id_fk	int	4
@C_id_fk	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[sp_InstructorTeachCource_Insert] @ins_id_fk INT=null, @C_id_fk INT=null

AS

BEGIN

if exists (select 1 from Instructor_Teach_Cource where @ins_id_fk=@ins_id_fk and

C_id_fk=@C_id_fk)

select'This Record already exist.'

else if (@ins_id_fk in (select i.Ins_Id from Instructor i)

and @C_id_fk in (select c.C_Id from course c))

begin

INSERT INTO Instructor_Teach_Cource (ins_id_fk, C_id_fk)

VALUES (@ins_id_fk, @C_id_fk);

select ('Inserted completed successfully.')

end

else

select ('Data does not match')

END

GO
```

Uses

```
[dbo].[Course]
[dbo].[Instructor]
[dbo].[Instructor_Teach_Cource]
```

[dbo].[sp_InstructorTeachCource_SelectByPK]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE PROCEDURE [dbo].[sp_InstructorTeachCource_SelectByPK] @ins_id_fk INT

AS

BEGIN

if( @ins_id_fk in (select Ins_Id from Instructor) )

begin

SELECT ins_id_fk, C_id_fk

FROM Instructor_Teach_Cource

WHERE ins_id_fk = @ins_id_fk

end

else

select('This instructor does not teach course.')

END

GO
```

Uses

```
[dbo].[Instructor]
[dbo].[Instructor_Teach_Cource]
```

[dbo].[sp_Topic_Delete]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE PROCEDURE [dbo].[sp_Topic_Delete] @name VARCHAR(20) = null, @C_id_fk INT = null

AS
BEGIN

if (@name in (select t.name from Topic t) and @C_id_fk in (select t.C_id_fk from Topic t))

begin

DELETE FROM Topic

WHERE name = @name AND C_id_fk = @C_id_fk;

select 'Delete completed successfully.'

end
else

select('This Topic is Not Exist')

END

GO
```

Uses

[dbo].[Topic]

[dbo].[sp_Topic_Insert]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE PROCEDURE [dbo].[sp_Topic_Insert] @C_id_fk INT = null ,@name VARCHAR(20) = null

AS

IF @name IS NULL OR LEN(@name) = 0

BEGIN

select('Topic name cannot be NULL or empty.')

END

else if @C_id_fk in (select c.C_Id from Course c )

BEGIN

INSERT INTO Topic (name, C_id_fk)

VALUES (@name, @C_id_fk);

select ('Inserted completed successfully.')

END

else

BEGIN

SELECT 'This id of Course is not exist or this topic is already exist'

END

GO
```

Uses

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



Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE PROCEDURE [dbo].[sp_Topic_SelectAll]

AS

BEGIN

SELECT *

FROM Topic;

END

GO
```

Uses

[dbo].[Topic]

[dbo].[sp_Topic_SelectByPK]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Nam	е	Data Type	Max Length (Bytes)
@C_	id_fk	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[sp_Topic_SelectByPK] @C_id_fk INT

AS

if @C_id_fk in (select C_id_fk from Topic)

begin

select *

from Topic

where C_id_fk = @C_id_fk

end

else

select('This Topic is Not Exist')

GO
```

Uses

[dbo].[Topic]

[dbo].[sp_Topic_Update]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@name	varchar(20)	20
@C_id_fk	int	4
@new_name	varchar(20)	20
@new_C_id_fk	int	4

```
CREATE PROCEDURE [dbo].[sp_Topic_Update]
         @name varchar(20) = null,
         @C_id_fk INT = null,
         @new name VARCHAR(20) = null,
          @new C id fk INT = null
      AS
    begin
         begin try
          if (@name in (select t.name from Topic t) and @C id fk in (select t.C id fk from Topic
t))
               begin
              update Topic
               name=COALESCE(@new_name, name),
               C_id_fk=COALESCE(@new_C_id_fk,C_id_fk)
               where C id fk = @C id fk and name=@name
               select 'Update completed successfully.'
         else
            select 'this id is not exist'
         end try
         Begin catch
         select 'Data does not match';
         end catch
     end
GO
```

Uses		
[dba] [Tania]		
[dbo].[Topic]		

Project > DESKTOP-3CE5M2E > User databases > ITIExamination-System > Programmability > Stored Procedures > dbo.sp_Topic_-Update

[dbo].[sp_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].[sp_TrackBranch_Delete]

@B_id_fk INT = null,

@T_id_fk INT = null

AS

BEGIN

if (@B_id_fk in (select t.B_id_fk from track_branch t)

and @T_id_fk in (select t.T_id_fk from track_branch t))

begin

DELETE FROM Track_Branch

WHERE B_id_fk = @B_id_fk AND T_id_fk = @T_id_fk;

END

else

select 'This Track is Not Exist in This Branch'

END

GO
```

Uses

[dbo].[track_branch]

[dbo].[updateanswerTernary]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@st_id	int	4
@ex_id	int	4
@Q_id	int	4
@answer	varchar(20)	20

SQL Script

Uses

[dbo].[Student_Exam_Question]

[dbo].[updateCourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@id	int	4
@name	varchar(20)	20
@duration	int	4

```
create procedure [dbo].[updateCourse]
@name varchar(20) = null,
@duration int = null
begin
       if exists(select c_id from course where c_id = @id)
       begin
           update course
               c_name = coalesce(@name, c_name),
               C_Duration = coalesce(@duration, C_Duration)
           where C id = @id
           select 'Course updated successfully.'
       end
       else
           select 'Error: Course does not exist.'
       end
   end try
   begin catch
    select'Error occurs'
   end catch
end
GO
```

Uses			
[dbo].[Course]			

Project > DESKTOP-3CE5M2E > User databases > ITIExamination-System > Programmability > Stored Procedures > dbo.update-Course

[dbo].[UpdateCourseHasTrack]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@oldCrsid	int	4
@oldTrackid	int	4
@newCrsid	int	4
@newTrackid	int	4

```
CREATE PROC [dbo].[UpdateCourseHasTrack]
   @oldCrsid INT,
   @oldTrackid INT,
   @newCrsid INT = NULL,
   @newTrackid INT = NULL
AS
BEGIN
   IF NOT EXISTS (SELECT C_id FROM CourseHasTrack WHERE C_Id = @oldCrsid AND Track_Id = @old-
   BEGIN
      SELECT 'ERROR! This Record Does Not Exist.';
      RETURN;
   IF @newCrsid IS NOT NULL
      IF NOT EXISTS (SELECT C Id FROM Course WHERE C Id = @newCrsid)
          SELECT 'ERROR! This Course Does Not Exist.';
          RETURN;
      END
   END
   IF @newTrackid IS NOT NULL
       IF NOT EXISTS (SELECT T_Id FROM Track WHERE T_Id = @newTrackid)
```

```
BEGIN
           SELECT 'ERROR! This Track Does Not Exist.';
           RETURN;
       END
   END
   IF @newCrsid IS NOT NULL AND @newTrackid IS NOT NULL
       IF EXISTS (SELECT C_Id FROM CourseHasTrack WHERE C_Id = @newCrsid AND Track_Id = @new-
Trackid)
       BEGIN
           SELECT 'ERROR! This Record Already Exists.';
           RETURN;
       END
   END
   UPDATE CourseHasTrack
       C Id = COALESCE (@newCrsid, C Id),
       Track_Id = COALESCE(@newTrackid, Track_Id)
       C_Id = @oldCrsid AND Track_Id = @oldTrackid;
   SELECT 'Record Updated Successfully.';
END
GO
```

Uses

[dbo].[Course]
[dbo].[CourseHasTrack]
[dbo].[track]

[dbo].[updateExam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@ex_id	int	4
@ex_name	varchar(50)	50
@ex_date	date	3
@crs_id	int	4
@totalscore	int	4

```
create proc [dbo].[updateExam] @ex_id int , @ex_name varchar(50) , @ex_date date , @crs_id int ,
@totalscore int
begin
if not exists (select 1 from Exam where ex id = @ex id)
select'Exam does not exist.'
else if not exists (select C id from course where C id=@crs id)
select'This course dose not exist'
else if exists (select 1 from Exam where ex_id=@ex_id and ex_name=@ex_name
                                    and ex date=@ex date and c id=@crs id and
total score=@totalscore)
select ' This Record already exists'
begin
update Exam
   set
   ex_name = Isnull(@ex_name,ex_name) ,
   ex date = Isnull(@ex date, ex date) ,
   c_id = Isnull(@crs_id, c_id),
   total score = Isnull(@totalscore, total score)
   where ex_id = @ex_id
select ' Update Successfully'
end
end try
begin catch
```

Project > DESKTOP-3CE5M2E > User databases > ITIExamination-System > Programmability > Stored Procedures > dbo.updateExam

```
select 'ERROR!!'
end catch
end
GO
```

Uses

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

[dbo].[updateGradeTernary]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

Uses

[dbo].[Student_Exam_Question]

[dbo].[UpdateInst]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@id	int	4
@name	varchar(20)	20
@salary	int	4
@email	varchar(50)	50

```
create proc [dbo].[UpdateInst]
@id int,
@name varchar(20) = NULL ,
@salary int = NULL ,
@email varchar(50) = NULL
  begin try
       if not exists (select Ins Id from Instructor where Ins id = @id)
         select 'Instructor Dose Not Exist.'
        end
      else
        begin
           update Instructor
             Ins_Name = COALESCE(@name, Ins_Name),
             Ins_Salary = COALESCE(@salary, Ins_Salary),
             Ins_Email = COALESCE(@email, Ins_Email)
            where Ins Id = @id
  end try
  begin catch
       select 'Invalid Data.'
  end catch
GO
```

Project > DESKTOP-3CE5M2E > User databases > ITIExamination-
System > Programmability > Stored Procedures > dbo.UpdateInst

Uses

[dbo].[Instructor]

[dbo].[updateInstWorksInTrack]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Date	ita Type	Max Length (Bytes)
@oldtrackid	int		4
@oldinstid	int		4
@newtrackid	int		4
@newinstid	int		4
@hiringdate	dat	te	3

```
create procedure [dbo].[updateInstWorksInTrack]
   @oldtrackid int,
   @oldinstid int,
   @newtrackid int = null,
   @newinstid int = null,
   @hiringdate date = null
as
begin
   begin try
        if not exists (select Ins_Id
                      from InstWorksInTrack
                       where Ins_Id = @oldinstid and Track_Id = @oldtrackid)
        begin
           select 'error: the record does not exist in instworksintrack.'
        end
        else
            if @newtrackid is not null and not exists (select T id from track where T id =
@newtrackid)
                select 'error: this track does not exist.'
            end
            else
            begin
                if @newinstid is not null and not exists (select Ins Id from Instructor where
```

```
Ins Id = @newinstid)
                begin
                    select 'error: this instructor does not exist.'
                end
                else
                begin
                    if @newtrackid is not null and @newinstid is not null and exists
                    (select Ins Id
                     from InstWorksInTrack
                       where Track Id = @newtrackid and Ins Id = @newinstid)
                    begin
                        select 'error: a record with the new track id and instructor id already
exists.'
                    end
                    else
                    begin
                        update InstWorksInTrack
                            Track Id = coalesce(@newtrackid, Track Id),
                            Ins Id = coalesce(@newinstid, Ins Id),
                            HiringDate = coalesce(@hiringdate, HiringDate)
                        where
                            Track Id = @oldtrackid and Ins Id = @oldinstid;
                        select 'record updated successfully.'
                    end
                end
            end
        end
   end try
    begin catch
        select 'error: an unexpected error occurred.'
    end catch
end
GO
```

Uses

[dbo].[Instructor] [dbo].[InstWorksInTrack] [dbo].[track]

[dbo].[usp_Branch_Delete]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

Uses

[dbo].[usp_Branch_Insert]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@B_name	varchar(20)	20
@Location	varchar(50)	50

SQL Script

Uses



[dbo].[usp_Branch_SelectAll]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE PROCEDURE [dbo].[usp_Branch_SelectAll]
     BEGIN
       SELECT *
       FROM Branch;
     END
GO
```

Uses

[dbo].[usp_Branch_SelectByld]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

Uses

[dbo].[usp_Branch_Update]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@B_id	int	4
@B_name	varchar(20)	20
@Location	varchar(50)	50

SQL Script

```
CREATE PROCEDURE [dbo].[usp_Branch_Update]
         @B_id INT = null,
         @B name VARCHAR(20) = null,
         @Location VARCHAR(50) = null
      AS
      BEGIN
         if exists (select b.B_id from branch b WHERE B_id = @B_id)
         begin
         UPDATE Branch
         B_name = coalesce (@B_name, B_name),
         Location = coalesce (@Location, Location)
         WHERE B_id = @B_id;
         select 'Update completed successfully.'
          select 'This id is not exist'
      END
GO
```

Uses

[dbo].[usp_BranchPhone_Delete]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@B_id_fk	int	4
@Phone	varchar(11)	11

SQL Script

```
CREATE PROCEDURE [dbo].[usp_BranchPhone_Delete]

    @B_id_fk INT = null,
    @Phone VARCHAR(11) = null

AS

BEGIN

if (@B_id_fk in (select b.B_id_fk from Branch_Phone b )
    and @Phone in (select b.Phone from Branch_Phone b))

begin

DELETE FROM Branch_Phone

WHERE B_id_fk = @B_id_fk AND Phone = @Phone

END

else

select 'This Branch is Not Exist.'

END

GO
```

Uses

[dbo].[Branch_Phone]

[dbo].[usp_BranchPhone_Insert]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@B_id_fk	int	4
@Phone	varchar(11)	11

SQL Script

```
CREATE PROCEDURE [dbo].[usp_BranchPhone_Insert]
        @B id fk INT = null,
        @Phone VARCHAR(11) = null
    AS
    BEGIN
         begin try
         if exists (select B_id from branch where B_id = @B_id_fk)
          INSERT INTO Branch Phone (B id fk, Phone)
         VALUES (@B_id_fk, @Phone);
         select ('Inserted completed successfully.')
          end
          else
          select'This branch Does Not Exist.'
          end try
         begin catch
         select ('Data does not match');
         end catch
     END
GO
```

Uses

[dbo].[branch] [dbo].[Branch_Phone] Project > DESKTOP-3CE5M2E > User databases > ITIExamination-System > Programmability > Stored Procedures > dbo.usp_Branch-Phone_SelectAll



[dbo].[usp_BranchPhone_SelectAll]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

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

Uses

[dbo].[Branch_Phone]

[dbo].[usp_BranchPhone_SelectByBranch]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

Uses

[dbo].[Branch_Phone]

[dbo].[usp_InstructorTeachCource_SelectAll]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE PROCEDURE [dbo].[usp_InstructorTeachCource_SelectAll]

AS

BEGIN

SELECT *

FROM Instructor_Teach_Cource;

END

GO
```

Uses

[dbo].[Instructor_Teach_Cource]

[dbo].[usp_InstructorTeachCource_Update]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@ins_id_fk	int	4
@C_id_fk	int	4
@new_ins_id_fk	int	4
@new_C_id_fk	int	4

```
CREATE PROCEDURE [dbo].[usp_InstructorTeachCource_Update]
         @ins id fk INT = null,
         @C id fk INT = null,
         @new ins id fk INT = null,
         @new C id fk INT = null
     AS
     BEGIN
         if not exists(select ins_id from instructor where ins_id= @new_ins_id_fk)
         select'This Instructor Does Not exist.'
         else if not exists(select C id from Course where C id= @new c id fk)
         select'This Instructor Does Not exist.'
         else if (@ins_id_fk in (select i.ins_id_fk from Instructor_Teach_Cource i)
         and @C id fk in (select i.C id fk from Instructor Teach Cource i))
         begin
         UPDATE Instructor_Teach_Cource
         ins_id_fk = COALESCE (@new_ins_id_fk,ins_id_fk),
         C id fk = COALESCE (@new C id fk,C id fk)
         WHERE ins_id_fk = @ins_id_fk AND C_id_fk = @C_id_fk;
         select ('Updated completed successfully.')
         else
         select ('Data does not match');
     END
GO
```

Project > DESKTOP-3CE5M2E > User databases > ITIExamination-System > Programmability > Stored Procedures > dbo.usp_-InstructorTeachCource_Update

Uses

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

[dbo].[usp_Student_Delete]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

Uses

[dbo].[Student]

[dbo].[usp_Student_Insert]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@fname	varchar(20)	20
@Iname	varchar(20)	20
@gender	varchar	1
@Email	varchar(20)	20
@BOD	date	3
@T_id_fk	int	4

```
CREATE PROCEDURE [dbo].[usp_Student_Insert]
         @fname VARCHAR(20) = null,
         @lname VARCHAR(20) = null,
         @gender VARCHAR(1) = null,
         @Email VARCHAR(20) = null ,
         @BOD DATE = null,
         @T id fk INT = null
     AS
     BEGIN
     begin try
     if not exists(select T id from Track where t id =@T id fk)
     select'This track does not exist.'
     begin
         INSERT INTO Student (fname, lname, gender, Email, BOD, T_id_fk)
         VALUES (@fname, @lname, @gender, @Email, @BOD, @T id fk);
         select ('Inserted completed successfully.')
     end
     \operatorname{\underline{End}}\ \operatorname{try}
     Begin catch
       select 'Data does not match';
     end catch
```

Project > DESKTOP-3CE5M2E > User databases > ITIExamination-System > Programmability > Stored Procedures > dbo.usp_-Student_Insert

GO		

Uses

[dbo].[Student] [dbo].[track]

[dbo].[usp_Student_SelectAll]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE PROCEDURE [dbo].[usp_Student_SelectAll]

AS

BEGIN

SELECT *

FROM Student;

END

GO
```

Uses

[dbo].[Student]

[dbo].[usp_Student_SelectByPK]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

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

AS

BEGIN

IF EXISTS (SELECT 1 FROM Student WHERE st_id = @st_id)

BEGIN

SELECT *

FROM Student

WHERE st_id = @st_id;

END

select 'This Student Is Not Exist'

END

GO
```

Uses

[dbo].[Student]

[dbo].[usp_Student_Update]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@st_id	int	4
@fname	varchar(20) 20
@Iname	varchar(20) 20
@gender	varchar	1
@Email	varchar(20) 20
@BOD	date	3
@T_id_fk	int	4

```
CREATE PROCEDURE [dbo].[usp_Student_Update]
          @st id INT = null ,
          @fname VARCHAR(20) = null,
         @lname VARCHAR(20) = null,
          @gender VARCHAR(1) = null,
          @Email VARCHAR(20) = null,
         @BOD DATE = null,
          @T id fk INT = null
      AS
      BEGIN
          if not exists(select T id from Track where t id =@T id fk)
          select'This track does not exist.'
          else if exists (select s.st id from Student s where s.st id = @st id)
          begin
          UPDATE Student
          SET fname = COALESCE(@fname, fname),
             lname = COALESCE(@lname, lname),
             gender =COALESCE (@gender, gender),
             Email = COALESCE (@Email, Email),
              BOD = COALESCE (@BOD, BOD),
              T_id_fk =Coalesce(@T_id_fk,T_id_fk)
          WHERE st id = @st id;
```

Project > DESKTOP-3CE5M2E > User databases > ITIExamination-System > Programmability > Stored Procedures > dbo.usp_-Student_Update

```
select 'Update completed successfully.'
end
else
select'This record does not exist to be updated.'
END
GO
```

Uses

[dbo].[Student] [dbo].[track]

[dbo].[usp_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

Uses

[dbo].[usp_Track_Insert]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@T_name	varchar(20)	20
@description	varchar(50)	50

SQL Script

```
CREATE PROCEDURE [dbo].[usp_Track_Insert]
    @T_name VARCHAR(20) = null,
    @description VARCHAR(50) = null
AS
BEGIN
begin try
    INSERT INTO Track (T_name, description)
    VALUES (@T_name, @description);
    select ('Inserted completed successfully.')
end try
begin catch
    select ('Data does not match');
end catch
END
GO
```

Uses



Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE PROCEDURE [dbo].[usp_Track_SelectAll]

AS

BEGIN

SELECT *

FROM Track

END

GO
```

Uses

[dbo].[usp_Track_SelectByld]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

Uses

[dbo].[usp_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(20) 20
@description	varchar(50) 50

SQL Script

```
CREATE PROCEDURE [dbo].[usp_Track_Update]
       @T id INT =null,
       @T name VARCHAR(20) = null,
       @description VARCHAR(50) = null
     AS
     BEGIN
         if exists (select t.T_id from track t where T_id = @T_id )
         begin
         UPDATE Track
          SET
           T_name = coalesce (@T_name,t_name),
          description =coalesce (@description, description)
          WHERE T_id = @T_id;
          end
          else
          select 'This id is not exist'
     END
GO
```

Uses

[dbo].[usp_TrackBranch_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

```
CREATE PROCEDURE [dbo].[usp_TrackBranch_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)
         select'This Track Does Not exist.'
         else if not exists (select B id from branch where B id =@B id fk)
         select'This Branch Does Not exist.'
         else if exists (select 1 from track_branch where T_id_fk =@T_id_fk and B_id_fk
=@B id fk)
         select'This Record Already Exist.'
          else
         begin
          INSERT INTO Track Branch (B id fk, T id fk)
          VALUES (@B_id_fk, @T_id_fk);
          select ('Inserted completed successfully.')
          end
     end try
     begin catch
     select ('Data does not match');
     end catch
     END
GO
```

Project > DESKTOP-3CE5M2E > User databases > ITIExamination-System > Programmability > Stored Procedures > dbo.usp_Track-Branch_Insert

Uses

[dbo].[branch] [dbo].[track] [dbo].[track_branch]

[dbo].[usp_TrackBranch_SelectAll]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE PROCEDURE [dbo].[usp_TrackBranch_SelectAll]

AS

BEGIN

SELECT *

FROM Track_Branch;
END

GO
```

Uses

[dbo].[track_branch]

[dbo].[usp_TrackBranch_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].[usp_TrackBranch_SelectByPK]

@B_id_fk INT = null,

@T_id_fk INT = null

AS

BEGIN

if (@B_id_fk in (select t.B_id_fk from track_branch t)

and @T_id_fk in (select t.T_id_fk from track_branch t))

begin

SELECT B_id_fk, T_id_fk

FROM Track_Branch

WHERE B_id_fk = @B_id_fk AND T_id_fk = @T_id_fk;

end

else

select 'This Track is Not Exist in This Branch'

END

GO
```

Uses

[dbo].[track_branch]

[dbo].[usp_TrackBranch_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].[usp_TrackBranch_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
         if not exists (select T_id from track where t_id =@new_T_id_fk)
          select'This Track Does Not exist.'
         else if not exists (select B_id from branch where B_id =@new_B_id_fk)
          select'This Branch Does Not exist.'
          else if (@B_id_fk in (select t.B_id_fk from track_branch t)
                  and @T_id_fk in (select t.T_id_fk from track_branch t))
         begin
          UPDATE track_branch
         B_id_fk = coalesce (@new_B_id_fk,B_id_fk),
          T id fk =coalesce (@new T id fk,T id fk)
          WHERE B_id_fk = @B_id_fk AND T_id_fk = @T_id_fk;
          end
          else
          select 'This id is not exist'
      END
GO
```

Project > DESKTOP-3CE5M2E > User databases > ITIExamination-System > Programmability > Stored Procedures > dbo.usp_Track-Branch_Update

Uses

[dbo].[branch] [dbo].[track] [dbo].[track_branch]



Objects

Name	
dbo	
guest	



Property	Value
Туре	WindowsUser
Default Schema	dbo

Database Level Permissions

Туре	Action
CONNECT	Grant

SQL Script

GO



Property	Value
Туре	SqlUser
Default Schema	guest

SQL Script

GO			

La Database Roles

Objects

Name
db_accessadmin
db_backupoperator
db_datareader
db_datawriter
db_ddladmin
db_denydatareader
db_denydatawriter
db_owner
db_securityadmin
public

db_accessadmin

Properties

Property	Value
Owner	dbo

db_backupoperator

Properties

Property	Value
Owner	dbo

♣ db_datareader

Properties

Property	Value
Owner	dbo

db_datawriter

Properties

Property	Value
Owner	dbo

db_ddladmin

Properties

Property	Value
Owner	dbo

db_denydatareader

Properties

Property	Value
Owner	dbo

db_denydatawriter

Properties

Property	Value
Owner	dbo

db_owner

Properties

Property	Value
Owner	dbo

♣ db_securityadmin

Property	Value
Owner	dbo

♣ public

Properties

Property	Value
Owner	dbo