DESKTOP7QIRQTB\SQLEXPRESS Documentation

SQLProject

Server DESKTOP-7QIRQTB\SQLEXPRESS

Author Goharista

Created Friday, April 1, 2022 9:40:32 PM

File Path C:\Users\Goharista\Desktop\T-Sql Project - Documentation-2022-04-01T21-40-32.pdf

Table of Contents

able of Contents	2
DESKTOP-7QIRQTB\SQLEXPRESS	5
User databases	
SQLProject Database	8
Ⅲ Tables	11
[dbo].[choices]	12
[dbo].[courses]	14
[dbo].[department]	16
[dbo].[exams]	18
[dbo].[instructor]	20
[dbo].[Instructor_Courses]	22
[dbo].[questions]	24
[dbo].[student_courses]	26
[dbo].[student_exam]	28
[dbo].[students]	30
[dbo].[Students_exams_questions]	32
[dbo].[topics]	34
Stored Procedures	36
[dbo].[AssignExamTOStudent]	38
[dbo].[correctExam]	40
[dbo].[delete_choices]	42
[dbo].[delete_Course]	43
[dbo].[delete_Department]	44
[dbo].[delete_Exams]	45
[dbo].[delete_instructor]	46
[dbo].[delete_instructorCourse]	47
[dbo].[delete_Questions]	48
[dbo].[delete_Student]	49
[dbo].[delete_studentCourse]	50
[dbo].[delete_studentExam]	51
[dbo].[delete_studentExamQuestionByExamAndQuestion]	52
[dbo].[delete_studentExamQuestionByExamAndStudent]	53
[dbo].[delete_Topics]	54
[dbo].[exam_answers]	55
[dbo].[generateExam]	57
[dbo].[insert_choices]	59
[dbo].[insert_Course]	60

[dbo].[insert_Department]	61
[dbo].[insert_Exams]	
[dbo].[insert_instructor]	63
[dbo].[insert_instructorCourse]	64
[dbo].[insert_question]	65
[dbo].[insert_Student]	66
[dbo].[insert_studentCourse]	
[dbo].[insert_studentExam]	68
[dbo].[insert_studentExamQuestion]	
[dbo].[insert_Topics]	70
[dbo].[r1_student_info_for_each_department]	71
[dbo].[r2_student_grades]	
[dbo].[r3_instructor_courses]	
[dbo].[r4_course_topics]	74
[dbo].[r5_exam_questions]	
[dbo].[r6_student_exam]	76
[dbo].[select_choices]	
[dbo].[select_Course]	
[dbo].[select_department]	80
[dbo].[select_Exams]	
[dbo].[select_Instructor]	83
[dbo].[select_instructorCourseByCourse]	
[dbo].[select_instructorCourseByInstructor]	
[dbo].[select_Questions]	86
[dbo].[select_Student]	
[dbo].[select_studentCourseByCourse]	
[dbo].[select_studentCourseByStudent]	
[dbo].[select_studentExamByExam]	
[dbo].[select_studentExamByExamAndStudebt]	
[dbo].[select_studentExamByStudent]	92
[dbo].[select_studentExamQuestionByExam]	
[dbo].[select_studentExamQuestionByExamAndStudent]	95
[dbo].[select_studentExamQuestionByStudent]	96
[dbo].[select_Topics]	
[dbo].[update_choices]	
[dbo].[update_Course]	
[dbo].[update_Department]	
[dbo].[update_Exams]	

	[dbo].[update_Instructor]	104
	[dbo].[update_questions]	
	[dbo].[update_Student]	107
	[dbo].[update_studentExam]	109
	[dbo].[update_studentExamQuestion]	110
	[dbo].[update_Topics]	111
1	Users	112
	1 dbo	113
	1 guest	114
	1 mansour	115
	1 torky	118
	♣ wahby	119
•	Database Roles	120
	db_accessadmin	120
	db_backupoperator	121
	db_datareader	121
	db_datawriter	122
	♣ db_ddladmin	123
	db_denydatareader	123
	db_denydatawriter	124
	db_owner	125
	db_securityadmin	125
	♣ public	126

■ DESKTOP-7QIRQTB\SQLEXPRESS

Databases (1)

• 🛮 SQLProject

Server Properties

Property	Value
Product	Microsoft SQL Server
Version	15.0.2000.5
Language	English (United States)
Platform	NT x64
Edition	Express Edition (64-bit)
Engine Edition	4 (Express)
Processors	4
OS Version	6.3 (18362)
Physical Memory	8152
Is Clustered	False
Root Directory	C:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL
Collation	Arabic_CI_AS

Server Settings

Property	Value
Default data file path	C:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\
Default backup file path	C:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\Backup
Default log file path	C:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\
Recovery Interval (minutes)	0
Default index fill factor	0
Default backup media retention	0

Advanced Server Settings

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

Project > DESKTOP-7QIRQTB\SQLEXPRESS

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

□ User databases				
------------------	--	--	--	--

Databases (1)

• 🗏 SQLProject

■ SQLProject Database

Database Properties

Property	Value
SQL Server Version	Max
Compatibility Level	Max
Last backup time	04/01/2022
Last log backup time	04/01/2022
Creation date	Mar 31 2022
Users	7
Database Encryption Enabled	False
Database Encryption Algorithm	None
Database size	80.00 MB
Unallocated space	3.56 MB

Database Options

Property	Value
Compatibility Level	150
Database collation	Arabic_CI_AS
Restrict access	MULTI_USER
Is read-only	False
Auto close	False
Auto shrink	False
Database status	ONLINE
In standby	False
Cleanly shutdown	False
Supplemental logging enabled	False
Snapshot isolation state	OFF
Read committed snapshot on	False
Recovery model	FULL
Page verify option	CHECKSUM
Auto create statistics	True
Auto update statistics	True
Auto update statistics asynchronously	False
ANSI NULL default	False
ANSI NULL enabled	False
ANSI padding enabled	False

ANSI warnings enabled	False
Arithmetic abort enabled	False
Concatenating NULL yields NULL	False
Numeric roundabort enabled	False
Quoted Identifier On	False
Recursive triggers enabled	False
Close cursors on commit	False
Local cursors by default	False
Fulltext enabled	True
Trustworthy	False
Database chaining	False
Forced parameterization	False
Master key encrypted by server	False
Published	False
Subscribed	False
Merge published	False
Is distribution database	False
Sync with backup	False
Service broker GUID	23240466-6204-4d1e-b8dc-0a89b1da0c0b
Service broker enabled	False
Log reuse wait	NOTHING
Date correlation	False
CDC enabled	False
Encrypted	False
Honor broker priority	False
Default language	English
Default fulltext language LCID	1033
Nested triggers enabled	True
Transform noise words	False
Two-digit year cutoff	2049
Containment	NONE
Target recovery time	60
Database owner	DESKTOP-7QIRQTB\Goharista

Files

Name	Туре	Size	Maxsize	Autogrowth	File Name
SQLProject	Data	8.00 MB	unlimited	64.00 MB	C:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQ L\DATA\SQLProject.mdf
SQLProject_log	Log	72.00 MB	2048.00 GB	64.00 MB	C:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQ

	L\DATA\SQLProject_log.ldf
	· - ·

■ Tables

Objects

Name
dbo.choices
dbo.courses
dbo.department
dbo.exams
dbo.instructor
dbo.Instructor_Courses
dbo.questions
dbo.student_courses
dbo.student_exam
dbo.students
dbo.Students_exams_questions
dbo.topics

[dbo].[choices]

Properties

Property	Value
Collation	Arabic_CI_AS
Row Count (~)	150
Created	12:55:25 PM Wednesday, March 9, 2022
Last Modified	3:56:18 PM Wednesday, March 9, 2022

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK G	choice_ID	int	4	NOT NULL	1 - 1
	choice_description	varchar(max)	max	NOT NULL	
	isCorrect	char(1)	1	NOT NULL	
FK	question_ID	int	4	NOT NULL	

Indexes

Key	Name	Key Columns	Unique
PKP C	PK_choices	choice_ID	True

Foreign Keys

Name	No Check	Update	Delete	Columns
FK_choices_questions	True	Cascade	Cascade	question_ID->[dbo].[questions].[question_ID]

```
CREATE TABLE [dbo].[choices]

(
[choice_ID] [int] NOT NULL IDENTITY(1, 1),
[choice_description] [varchar] (max) COLLATE Arabic_CI_AS NOT NULL,
[isCorrect] [char] (1) COLLATE Arabic_CI_AS NOT NULL,
[question_ID] [int] NOT NULL

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[choices] ADD CONSTRAINT [PK_choices] PRIMARY KEY CLUSTERED ([choice_-ID]) ON [PRIMARY]

GO
```

ALTER TABLE [dbo].[choices] WITH NOCHECK ADD CONSTRAINT [FK_choices_questions] FOREIGN
KEY ([question_ID]) REFERENCES [dbo].[questions] ([question_ID]) ON DELETE CASCADE ON
UPDATE CASCADE
GO

Uses

[dbo].[questions]

Used By

[dbo].[correctExam]

[dbo].[delete_choices]

[dbo].[insert_choices]

[dbo].[r5_exam_questions]

[dbo].[r6_student_exam]

[dbo].[select_choices]

[dbo].[update_choices]

[dbo].[courses]

Properties

Property	Value
Collation	Arabic_CI_AS
Row Count (~)	3
Created	12:55:24 PM Wednesday, March 9, 2022
Last Modified	4:00:27 PM Wednesday, March 9, 2022

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK C	course_ID	int	4	NOT NULL	1 - 1
	course_name	varchar(50)	50	NOT NULL	

Indexes

Key	Name	Key Columns	Unique
PK G	PK_courses	course_ID	True

SQL Script

```
CREATE TABLE [dbo].[courses]

(
[course_ID] [int] NOT NULL IDENTITY(1, 1),
[course_name] [varchar] (50) COLLATE Arabic_CI_AS NOT NULL

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[courses] ADD CONSTRAINT [PK_courses] PRIMARY KEY CLUSTERED ([course_-ID]) ON [PRIMARY]

GO
```

Used By

[dbo].[exams]

[dbo].[Instructor_Courses]

[dbo].[questions]

[dbo].[student_courses]

[dbo].[topics]

[dbo].[delete_Course]

[dbo].[generateExam]
[dbo].[insert_Course]
[dbo].[r2_student_grades]
[dbo].[r3_instructor_courses]
[dbo].[select_Course]
[dbo].[update_Course]

[dbo].[department]

Properties

Property	Value
Collation	Arabic_CI_AS
Row Count (~)	3
Created	1:10:45 PM Wednesday, March 9, 2022
Last Modified	10:10:05 PM Saturday, March 12, 2022

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK	department_ID	int	4	NOT NULL	1 - 1
	department_name	varchar(50)	50	NOT NULL	
FK	supervisior_ID	int	4	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
PK	PK_department	department_ID	True

Foreign Keys

Name	Update	Delete	Columns
FK_department_instructor	Cascade	SetNull	supervisior_ID->[dbo].[instructor].[instructor_ID]

```
CREATE TABLE [dbo].[department]

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

[department_name] [varchar] (50) COLLATE Arabic_CI_AS NOT NULL,

[supervisior_ID] [int] NULL

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[department] ADD CONSTRAINT [PK_department] PRIMARY KEY CLUSTERED

([department_ID]) ON [PRIMARY]

GO

ALTER TABLE [dbo].[department] ADD CONSTRAINT [FK_department_instructor] FOREIGN KEY

([supervisior_ID]) REFERENCES [dbo].[instructor] ([instructor_ID]) ON DELETE SET NULL
```

ON UPDATE CASCADE
GO

Uses

[dbo].[instructor]

Used By

[dbo].[students]
[dbo].[delete_Department]
[dbo].[insert_Department]
[dbo].[select_department]
[dbo].[update_Department]

[dbo].[exams]

Properties

Property	Value
Row Count (~)	3
Created	12:55:25 PM Wednesday, March 9, 2022
Last Modified	10:24:27 PM Thursday, March 10, 2022

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PKP C	exam_ID	int	4	NOT NULL	1 - 1
FX	course_ID	int	4	NOT NULL	
	duration	int	4	NULL allowed	
	exam_total_degree	int	4	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
PK C	PK_exams	exam_ID	True

Foreign Keys

Name	Update	Delete	Columns
FK_exams_courses	Cascade	Cascade	course_ID->[dbo].[courses].[course_ID]

```
CREATE TABLE [dbo].[exams]

(
[exam_ID] [int] NOT NULL IDENTITY(1, 1),
[course_ID] [int] NOT NULL,
[duration] [int] NULL,
[exam_total_degree] [int] NULL

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[exams] ADD CONSTRAINT [PK_exams] PRIMARY KEY CLUSTERED ([exam_ID])

ON [PRIMARY]

GO

ALTER TABLE [dbo].[exams] ADD CONSTRAINT [FK_exams_courses] FOREIGN KEY ([course_ID])
```

REFERENCES [dbo].[courses] ([course_ID]) ON DELETE CASCADE ON UPDATE CASCADE

GO

Uses

[dbo].[courses]

Used By

[dbo].[student_exam]

[dbo].[Students_exams_questions]

[dbo].[AssignExamTOStudent]

[dbo].[delete_Exams]

[dbo].[generateExam]

[dbo].[insert_Exams]

[dbo].[select_Exams]

[dbo].[update_Exams]

[dbo].[instructor]

Properties

Property	Value
Collation	Arabic_CI_AS
Row Count (~)	7
Created	2:50:41 PM Wednesday, March 9, 2022
Last Modified	6:38:19 PM Monday, March 21, 2022

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK C	instructor_ID	int	4	NOT NULL	1 - 1
	instructor_name	varchar(50)	50	NOT NULL	
	instructor_UserName	varchar(50)	50	NULL allowed	
	instructor_Password	varchar(50)	50	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
PK G	PK_instructor	instructor_ID	True

SQL Script

```
CREATE TABLE [dbo].[instructor]

(
[instructor_ID] [int] NOT NULL IDENTITY(1, 1),
[instructor_name] [varchar] (50) COLLATE Arabic_CI_AS NOT NULL,
[instructor_UserName] [varchar] (50) COLLATE Arabic_CI_AS NULL,
[instructor_Password] [varchar] (50) COLLATE Arabic_CI_AS NULL
) ON [PRIMARY]

GO

ALTER TABLE [dbo].[instructor] ADD CONSTRAINT [PK_instructor] PRIMARY KEY CLUSTERED
([instructor_ID]) ON [PRIMARY]

GO
```

Used By

[dbo].[department]

Author: Goharista

[dbo].[Instructor_Courses]
[dbo].[delete_instructor]
[dbo].[insert_instructor]
[dbo].[r3_instructor_courses]
[dbo].[update_Instructor]

[dbo].[Instructor_Courses]

Properties

Property	Value
Row Count (~)	7
Created	10:03:29 PM Sunday, February 27, 2022
Last Modified	6:38:20 PM Monday, March 21, 2022

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Default
PKO FKO	instructor_ID	int	4	NOT NULL	((1000))
PKO FKO	course_ID	int	4	NOT NULL	

Indexes

Key	Name	Key Columns	Unique
PK	PK_Inst_Courses	instructor_ID, course_ID	True

Foreign Keys

Name	Update	Delete	Columns
FK_Inst_Courses_courses	Cascade	Cascade	course_ID->[dbo].[courses].[course_ID]
FK_Inst_Courses_instructor	Cascade	Cascade	instructor_ID->[dbo].[instructor].[instructor_ID]

```
CREATE TABLE [dbo].[Instructor_Courses]

(
[instructor_ID] [int] NOT NULL CONSTRAINT [DF_Inst_Courses_instructor_ID] DEFAULT
((1000)),
[course_ID] [int] NOT NULL
) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Instructor_Courses] ADD CONSTRAINT [PK_Inst_Courses] PRIMARY KEY
CLUSTERED ([instructor_ID], [course_ID]) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Instructor_Courses] ADD CONSTRAINT [FK_Inst_Courses_courses] FOREIGN
KEY ([course_ID]) REFERENCES [dbo].[courses] ([course_ID]) ON DELETE CASCADE ON UPDATE
```

```
CASCADE

GO

ALTER TABLE [dbo].[Instructor_Courses] ADD CONSTRAINT [FK_Inst_Courses_instructor]

FOREIGN KEY ([instructor_ID]) REFERENCES [dbo].[instructor] ([instructor_ID]) ON DELETE

CASCADE ON UPDATE CASCADE

GO
```

Uses

[dbo].[courses] [dbo].[instructor]

Used By

[dbo].[delete_instructorCourse]
[dbo].[insert_instructorCourse]
[dbo].[r3_instructor_courses]
[dbo].[select_instructorCourseByCourse]
[dbo].[select_instructorCourseByInstructor]

■ [dbo].[questions]

Properties

Property	Value
Collation	Arabic_CI_AS
Row Count (~)	60
Created	12:55:25 PM Wednesday, March 9, 2022
Last Modified	7:57:45 PM Friday, March 11, 2022

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK	question_ID	int	4	NOT NULL	1 - 1
	qusetion_desc	varchar(max)	max	NOT NULL	
	question_type	varchar(10)	10	NOT NULL	
FK	course_ID	int	4	NOT NULL	

Indexes

Key	Name	Key Columns	Unique
PKC	PK_questions	question_ID	True

Foreign Keys

Name	Update	Columns
FK_questions_courses	Cascade	course_ID->[dbo].[courses].[course_ID]

```
CREATE TABLE [dbo].[questions]

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

[qusetion_desc] [varchar] (max) COLLATE Arabic_CI_AS NOT NULL,

[question_type] [varchar] (10) COLLATE Arabic_CI_AS NOT NULL,

[course_ID] [int] NOT NULL

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[questions] ADD CONSTRAINT [PK_questions] PRIMARY KEY CLUSTERED

([question_ID]) ON [PRIMARY]

GO
```

```
ALTER TABLE [dbo].[questions] ADD CONSTRAINT [FK_questions_courses] FOREIGN KEY
([course_ID]) REFERENCES [dbo].[courses] ([course_ID]) ON UPDATE CASCADE

GO
```

Uses

[dbo].[courses]

Used By

[dbo].[choices]

[dbo].[Students_exams_questions]

[dbo].[correctExam]

[dbo].[delete_Questions]

[dbo].[generateExam]

[dbo].[insert question]

[dbo].[r5_exam_questions]

[dbo].[r6_student_exam]

[dbo].[select_Questions]

[dbo].[update_questions]

[dbo].[student_courses]

Properties

Property	Value
Row Count (~)	34
Created	10:03:14 PM Sunday, February 27, 2022
Last Modified	6:38:20 PM Monday, March 21, 2022

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Default
PKO C FKO	student_ID	int	4	NOT NULL	((0))
PKO C FKO	course_ID	int	4	NOT NULL	

Indexes

Key	Name	Key Columns	Unique
PK C	PK_stud_courses	student_ID, course_ID	True

Foreign Keys

Name	Update	Delete	Columns
FK_stud_courses_courses	Cascade	Cascade	course_ID->[dbo].[courses].[course_ID]
FK_stud_courses_students	Cascade	Cascade	student_ID->[dbo].[students].[student_ID]

```
CREATE TABLE [dbo].[student_courses]

(
[student_ID] [int] NOT NULL CONSTRAINT [DF_stud_courses_student_ID] DEFAULT ((0)),
[course_ID] [int] NOT NULL

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[student_courses] ADD CONSTRAINT [PK_stud_courses] PRIMARY KEY
CLUSTERED ([student_ID], [course_ID]) ON [PRIMARY]

GO

ALTER TABLE [dbo].[student_courses] ADD CONSTRAINT [FK_stud_courses_courses] FOREIGN
KEY ([course_ID]) REFERENCES [dbo].[courses] ([course_ID]) ON DELETE CASCADE ON UPDATE
CASCADE
```

```
ALTER TABLE [dbo].[student_courses] ADD CONSTRAINT [FK_stud_courses_students] FOREIGN
KEY ([student_ID]) REFERENCES [dbo].[students] ([student_ID]) ON DELETE CASCADE ON
UPDATE CASCADE
GO
```

Uses

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

Used By

[dbo].[delete_studentCourse]

[dbo].[insert_studentCourse]

[dbo].[r2_student_grades]

[dbo].[r3_instructor_courses]

[dbo].[select_studentCourseByCourse]

[dbo].[select_studentCourseByStudent]

[dbo].[student_exam]

Properties

Property	Value
Row Count (~)	20
Created	10:24:26 PM Thursday, March 10, 2022
Last Modified	10:10:05 PM Saturday, March 12, 2022

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PKO C FKO	student_ID	int	4	NOT NULL
PKC FKG	exam_ID	int	4	NOT NULL
	exam_grade	int	4	NULL allowed

Indexes

Key	Name	Key Columns	Unique
PK C	PK_student_exam	student_ID, exam_ID	True

Foreign Keys

Name	Update	Delete	Columns
FK_student_exam_exams	Cascade	Cascade	exam_ID->[dbo].[exams].[exam_ID]
FK_student_exam_students	Cascade	Cascade	student_ID->[dbo].[students].[student_ID]

```
CREATE TABLE [dbo].[student_exam]

(
[student_ID] [int] NOT NULL,

[exam_ID] [int] NOT NULL,

[exam_grade] [int] NULL

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[student_exam] ADD CONSTRAINT [PK_student_exam] PRIMARY KEY CLUSTERED

([student_ID], [exam_ID]) ON [PRIMARY]

GO
```

```
ALTER TABLE [dbo].[student_exam] ADD CONSTRAINT [FK_student_exam_exams] FOREIGN KEY
([exam_ID]) REFERENCES [dbo].[exams] ([exam_ID]) ON DELETE CASCADE ON UPDATE CASCADE

GO
ALTER TABLE [dbo].[student_exam] ADD CONSTRAINT [FK_student_exam_students] FOREIGN KEY
([student_ID]) REFERENCES [dbo].[students] ([student_ID]) ON DELETE CASCADE ON UPDATE
CASCADE

GO
```

Uses

[dbo].[exams]

[dbo].[students]

Used By

[dbo].[AssignExamTOStudent]

[dbo].[correctExam]

[dbo].[delete_studentExam]

[dbo].[insert_studentExam]

[dbo].[r2_student_grades]

[dbo].[select studentExamByExam]

[dbo].[select_studentExamByExamAndStudebt]

[dbo].[select_studentExamByStudent]

[dbo].[update_studentExam]

[dbo].[update_studentExamQuestion]

[dbo].[students]

Properties

Property	Value
Collation	Arabic_CI_AS
Row Count (~)	17
Created	10:10:05 PM Saturday, March 12, 2022
Last Modified	5:25:11 PM Friday, March 25, 2022

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity	Default
PK	student_ID	int	4	NOT NULL	1 - 1	
	student_Fname	varchar(50)	50	NOT NULL		('no name')
	student_Lname	varchar(50)	50	NOT NULL		
FK	department_ID	int	4	NOT NULL		
	student_UserName	varchar(50)	50	NOT NULL		
	student_Password	varchar(50)	50	NOT NULL		

Indexes

Key	Name	Key Columns	Unique
PKP C	PK_students	student_ID	True

Foreign Keys

Name	Update	Columns
FK_students_department	Cascade	department_ID->[dbo].[department].[department_ID]

```
CREATE TABLE [dbo].[students]

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

[student_Fname] [varchar] (50) COLLATE Arabic_CI_AS NOT NULL CONSTRAINT

[DF_students_stud_Fname] DEFAULT ('no name'),

[student_Lname] [varchar] (50) COLLATE Arabic_CI_AS NOT NULL,

[department_ID] [int] NOT NULL,
```

```
[student_UserName] [varchar] (50) COLLATE Arabic_CI_AS NOT NULL,
[student_Password] [varchar] (50) COLLATE Arabic_CI_AS NOT NULL
) ON [PRIMARY]

GO
ALTER TABLE [dbo].[students] ADD CONSTRAINT [PK_students] PRIMARY KEY CLUSTERED
([student_ID]) ON [PRIMARY]

GO
ALTER TABLE [dbo].[students] ADD CONSTRAINT [FK_students_department] FOREIGN KEY
([department_ID]) REFERENCES [dbo].[department] ([department_ID]) ON UPDATE CASCADE
GO
```

Uses

[dbo].[department]

Used By

[dbo].[student_courses]
[dbo].[student_exam]
[dbo].[Students_exams_questions]
[dbo].[AssignExamTOStudent]
[dbo].[delete_Student]
[dbo].[insert_Student]
[dbo].[r1_student_info_for_each_department]
[dbo].[r2_student_grades]
[dbo].[r6_student_exam]
[dbo].[update_Student]

[dbo].[Students_exams_questions]

Properties

Property	Value
Collation	Arabic_CI_AS
Row Count (~)	230
Created	2:27:10 PM Wednesday, March 9, 2022
Last Modified	5:25:11 PM Friday, March 25, 2022

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PKO C FKO	exam_ID	int	4	NOT NULL
PKC FKD	qustion_ID	int	4	NOT NULL
	student_answer	varchar(50)	50	NULL allowed
PKC FKG	student_ID	int	4	NOT NULL

Indexes

Key	Name	Key Columns	Unique
PK C	PK_Students_exams_questions	exam_ID, qustion_ID, student_ID	True

Foreign Keys

Name	Columns
FK_exams_questions_exams	exam_ID->[dbo].[exams].[exam_ID]
FK_exams_questions_questions	qustion_ID->[dbo].[questions].[question_ID]
FK_Students_exams_questions_students1	student_ID->[dbo].[students].[student_ID]

```
CREATE TABLE [dbo].[Students_exams_questions]

(
[exam_ID] [int] NOT NULL,
[qustion_ID] [int] NOT NULL,
```

```
[student_answer] [varchar] (50) COLLATE Arabic_CI_AS NULL,
[student_ID] [int] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Students_exams_questions] ADD CONSTRAINT [PK_-
Students_exams_questions] PRIMARY KEY CLUSTERED ([exam_ID], [qustion_ID], [student_ID])
ON [PRIMARY]
GO
ALTER TABLE [dbo].[Students_exams_questions] ADD CONSTRAINT [FK_exams_questions_exams]
FOREIGN KEY ([exam_ID]) REFERENCES [dbo].[exams] ([exam_ID])
GO
ALTER TABLE [dbo].[Students_exams_questions] ADD CONSTRAINT
[FK_exams_questions_questions] FOREIGN KEY ([qustion_ID]) REFERENCES [dbo].[questions]
([question_ID])
GO
ALTER TABLE [dbo].[Students_exams_questions] ADD CONSTRAINT [FK_-
Students_exams_questions_students1] FOREIGN KEY ([student_ID]) REFERENCES
[dbo].[students] ([student_ID])
GO
```

Uses

[dbo].[exams]

[dbo].[questions]

[dbo].[students]

Used By

[dbo].[AssignExamTOStudent]

[dbo].[correctExam]

[dbo].[delete studentExamQuestionByExamAndQuestion]

[dbo].[delete_studentExamQuestionByExamAndStudent]

[dbo].[exam_answers]

[dbo].[generateExam]

[dbo].[insert_studentExamQuestion]

[dbo].[r5_exam_questions]

[dbo].[r6_student_exam]

[dbo].[select_studentExamQuestionByExam]

[dbo].[select_studentExamQuestionByExamAndStudent]

 $[dbo]. [select_studentExamQuestionByStudent] \\$

[dbo].[update_studentExamQuestion]

[dbo].[topics]

Properties

Property	Value
Collation	Arabic_CI_AS
Row Count (~)	6
Created	12:55:26 PM Wednesday, March 9, 2022
Last Modified	3:40:54 PM Wednesday, March 9, 2022

Columns

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

Indexes

Key	Name	Key Columns	Unique
PKP C	PK_topics	topic_ID	True

Foreign Keys

Name	Update	Delete	Columns
FK_topics_courses	Cascade	Cascade	course_ID->[dbo].[courses].[course_ID]

```
CREATE TABLE [dbo].[topics]

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

[topic_name] [varchar] (50) COLLATE Arabic_CI_AS NOT NULL,

[course_ID] [int] NOT NULL

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[topics] ADD CONSTRAINT [PK_topics] PRIMARY KEY CLUSTERED ([topic_-
ID]) ON [PRIMARY]

GO

ALTER TABLE [dbo].[topics] ADD CONSTRAINT [FK_topics_courses] FOREIGN KEY ([course_ID])

REFERENCES [dbo].[courses] ([course_ID]) ON DELETE CASCADE ON UPDATE CASCADE
```

GO
Uses
[dbo].[courses]
Used By
[dbo].[delete_Topics]
[dbo].[insert_Topics]
[dbo].[r4_course_topics]
[dbo].[select_Topics]
[dbo].[update_Topics]

Stored Procedures

Objects

Name
dbo.AssignExamTOStudent
dbo.correctExam
dbo.delete_choices
dbo.delete_Course
dbo.delete_Department
dbo.delete_Exams
dbo.delete_instructor
dbo.delete_instructorCourse
dbo.delete_Questions
dbo.delete_Student
dbo.delete_studentCourse
dbo.delete_studentExam
dbo.delete_studentExamQuestionByExamAndQuestion
dbo.delete_studentExamQuestionByExamAndStudent
dbo.delete_Topics
dbo.exam_answers
dbo.generateExam
dbo.insert_choices
dbo.insert_Course
dbo.insert_Department
dbo.insert_Exams
dbo.insert_instructor
dbo.insert_instructorCourse
dbo.insert_question
dbo.insert_Student
dbo.insert_studentCourse
dbo.insert_studentExam
dbo.insert_studentExamQuestion
dbo.insert_Topics
dbo.r1_student_info_for_each_department
dbo.r2_student_grades
dbo.r3_instructor_courses
dbo.r4_course_topics
dbo.r5_exam_questions

dbo.r6_student_exam
dbo.select_choices
dbo.select_Course
dbo.select_department
dbo.select_Exams
dbo.select_Instructor
dbo.select_instructorCourseByCourse
dbo.select_instructorCourseByInstructor
dbo.select_Questions
dbo.select_Student
dbo.select_studentCourseByCourse
dbo.select_studentCourseByStudent
dbo.select_studentExamByExam
dbo.select_studentExamByExamAndStudebt
dbo.select_studentExamByStudent
dbo.select_studentExamQuestionByExam
dbo.select_studentExamQuestionByExamAndStudent
dbo.select_studentExamQuestionByStudent
dbo.select_Topics
dbo.update_choices
dbo.update_Course
dbo.update_Department
dbo.update_Exams
dbo.update_Instructor
dbo.update_questions
dbo.update_Student
dbo.update_studentExam
dbo.update_studentExamQuestion
dbo.update_Topics

[dbo].[AssignExamTOStudent]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE proc [dbo].[AssignExamTOStudent] @examID int,@stdID int
begin try
if exists (select exam ID from exams where exam ID=@examID )
    if exists (select student ID from students where student ID=@stdID)
           --insert in to student_exam table
           insert into student exam (student ID, exam ID)
           values(@stdID, @examID)
            --insert student into Students exams questions table
           insert into Students exams questions (exam ID, qustion ID, student ID)
           select @examID, qustion ID, @stdID
           from Students_exams_questions where exam_ID = @examID and student_ID = 0;
           select'inserted successfully'
        end
   else
       begin
       Select 'Student ID is wrong'
       end
    end
else
   select 'Exam ID is wrong'
end try
begin catch
   select' error Exam or Student ID is incorrect'
end catch
```

GO

Uses

[dbo].[exams]
[dbo].[student_exam]
[dbo].[students]
[dbo].[Students_exams_questions]

[dbo].[correctExam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE proc [dbo].[correctExam] @examID int, @stdID int
as
begin try
    -- variable to check if exam is graded befor or not
   declare @gradedExam int
       select @gradedExam = se.exam_grade
        from student exam as se
        where se.exam_ID = @examID and se.student_ID = @stdID
    -- declare varaible
   \label{eq:declare @stu_Answer varchar} \textbf{(50)} \text{ , } \textbf{@q\_id int , } \textbf{@CorrectAns varchar} \textbf{(50)}
    --declare the grade
   declare @grade int = 0
    -- declare cursor
    DECLARE answerCursor CURSOR FOR
        -- select student answer
        select seq.student answer , seq.qustion ID
        from Students_exams_questions seq
        where seq.exam ID = @examID and seq.student ID = @stdID
   FOR UPDATE
   OPEN answerCursor
   FETCH answerCursor INTO @stu Answer, @q id
    IF @gradedExam is not null
        select 'Exam is graded'
    ELSE
        WHILE @@FETCH STATUS = 0
        BEGIN
           -- model answer
                select @CorrectAns = c.choice_description
                from questions q
```

Project > DESKTOP-7QIRQTB\SQLEXPRESS > User databases > SQLProject > Programmability > Stored Procedures > dbo.correctExam

```
inner join choices c on q.question_ID = c.question_ID
               where c.question ID = @q id and c.isCorrect = 'T'
           -- student answer is the correct answer
           if @stu Answer = @CorrectAns
               set @grade = @grade + 2
           -- next answer
           FETCH answerCursor INTO @stu Answer, @q id
   --insert the grade to the students_exam table
   update student_exam
   set exam_grade = @grade
   where student ID= @stdID and exam ID = @examID
CLOSE answerCursor
DEALLOCATE answerCursor
select 'exam corrected successfully'
end try
begin catch
   select 'error'
end catch
GO
```

Uses

[dbo].[choices]
[dbo].[questions]
[dbo].[student_exam]
[dbo].[Students_exams_questions]

[dbo].[delete_choices]

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].[delete_choices] @id int
as
begin try
if exists ( select choice_ID from choices where choice_ID=@id)
begin
delete from choices where choice_ID=@id
end
else
begin
select 'Error in choice ID'
end
end try
begin catch
select 'id not found'
end catch
GO
```

Uses

[dbo].[choices]

[dbo].[delete_Course]

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].[delete_Course] @id int
as
begin try
if exists ( select Course_ID from Courses where Course_ID=@id)
begin
delete from Courses where Course_ID=@id
end
else
begin
select 'Error in Course ID'
end
end try
begin catch
select 'Error in Course ID'
end catch
```

Uses

[dbo].[courses]

[dbo].[delete_Department]

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].[delete_Department] @id int
as
begin try
if exists ( select Department_ID from Department where Department_ID=@id)
begin
delete from Department where Department_ID=@id
end
else
begin
select 'Error in Department ID'
end
end try
begin catch
select 'Error in Department ID'
end catch
```

Uses

[dbo].[department]

[dbo].[delete_Exams]

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].[delete_Exams] @id int
as
begin try
if exists (select exam_ID from exams where exam_ID=@id)
begin
delete from exams where exam_ID=@id
end
else
begin
select 'Error in Exam ID'
end
end try
begin catch
select 'Error in Exam ID'
end catch

GO
```

Uses

[dbo].[exams]

[dbo].[delete_instructor]

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].[delete_instructor] @id int
as
begin try
delete from instructor where instructor_ID=@id
end try
begin catch

select'must enter valid ID'
end catch
GO
```

Uses

[dbo].[instructor]

[dbo].[delete_instructorCourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Course_ld	int	4
@Instructor_id	int	4

SQL Script

Uses

[dbo].[Instructor_Courses]

[dbo].[delete_Questions]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

Uses

[dbo].[questions]

[dbo].[delete_Student]

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].[delete_Student] @id int
as
begin try
delete from students where student_ID=@id
end try
begin catch
select 'Error in Student ID'
end catch
```

Uses

[dbo].[students]

[dbo].[delete_studentCourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Course_ld	int	4
@Student_id	int	4

SQL Script

Uses

[dbo].[student_courses]

[dbo].[delete_studentExam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Exam_ld	int	4
@Student_id	int	4

SQL Script

Uses

[dbo].[student_exam]

[dbo].[delete_studentExamQuestionByExamAndQuestion]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Exam_ld	int	4
@question_ID	int	4

SQL Script

```
--delete
create proc [dbo].[delete studentExamQuestionByExamAndQuestion] @Exam Id int,
@question_ID int
begin try
if exists ( select * from Students exams questions where exam ID=@Exam Id and qustion -
ID=@question ID)
delete from Students_exams_questions where exam_ID=@Exam_Id and qustion_ID =
@question ID
end
else
begin
select 'Error in one of two IDs'
end
end try
begin catch
select 'invalid IDs'
end catch
GO
```

Uses

[dbo].[Students_exams_questions]

[dbo].[delete_studentExamQuestionByExamAndStudent]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Exam_ld	int	4
@Student_id	int	4

SQL Script

```
--delete
create proc [dbo].[delete_studentExamQuestionByExamAndStudent] @Exam_Id int,
@Student id int
if exists ( select * from Students_exams_questions where exam_ID=@Exam_Id and student_-
ID=@Student id)
delete from Students_exams_questions where exam_ID=@Exam_Id and student_ID =
@Student id
end
else
begin
select 'Error in one of two IDs'
end try
begin catch
select 'invalid IDs'
end catch
GO
```

Uses

[dbo].[Students_exams_questions]

[dbo].[delete_Topics]

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].[delete_Topics] @id int
as
begin try
if exists ( select Topic_ID from Topics where Topic_ID=@id)
begin
delete from Topics where Topic_ID=@id
end
else
begin
select 'Error in Topic ID'
end
end try
begin catch
select 'Error in Topic ID'
end catch
GO
```

Uses

[dbo].[topics]

[dbo].[exam_answers]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@examID	int	4
@stdID	int	4
@QuestionRN	int	4
@stdAnswer	varchar(50)	50

SQL Script

```
CREATE proc [dbo].[exam_answers] @examID int, @stdID int,@QuestionRN int, @stdAnswer
varchar(50)
begin try
if exists (select * from Students exams questions where exam ID = @examID and student -
ID = @stdID)
   begin
   with cte as
       select * from
           (select seq.student_answer,ROW_NUMBER() over(ORDER BY seq.exam_ID) as RN
           from Students_exams_questions as seq
           where seq.exam ID = @examID and seq.student ID = @stdID ) as x
       where x.RN = @QuestionRN
   update cte set student_answer = @stdAnswer
select 'exam answers inserted in DB successfully'
end try
begin catch
  select 'error'
end catch
GO
```

Project > DESKTOP-7QIRQTB\SQLEXPRESS > User databases > SQLProject > Programmability >	Stored
Procedures > dbo.exam answers	

Uses

[dbo].[Students_exams_questions]

[dbo].[generateExam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@courseName	varchar(50)	50
@TFnum	int	4
@mcqNum	int	4
@examDuration	int	4

SQL Script

```
CREATE procedure [dbo].[generateExam] @courseName varchar(50), @TFnum int, @mcqNum int,
@examDuration int
begin try
--get the course id
declare @courseID int
select @courseID = c.course_ID
from courses c
where c.course_name = @courseName
--insert the new exam into table exams
insert into exams (course ID, duration, exam total degree)
values(@courseID, @examDuration, 20)
--selecting the exam id
declare @examID int
select @examID = max(exam ID)
from exams
----selecting the questions for the exam
----inserting the exam into the student_exams table and student_exam_questions table
--true or false questions
insert into Students_exams_questions(exam_ID, qustion_ID, student_ID)
select top(@TFnum) @examID, q.question ID, 0
from questions q
where q.question_type = 'torf' and q.course_ID = @courseID
order by NEWID()
--mcq questions
insert into Students_exams_questions(exam_ID, qustion_ID, student_ID)
select top(@mcqNum) @examID, q.question_ID, 0
```

Project > DESKTOP-7QIRQTB\SQLEXPRESS > User databases > SQLProject > Programmability > Stored Procedures > dbo.generateExam

```
from questions q
where q.question_type = 'mcq' and q.course_ID = @courseID
order by NEWID()
select'inserted successfully'
end try
begin catch
    select 'error'
end catch
GO
```

Uses

[dbo].[courses]
[dbo].[exams]
[dbo].[questions]
[dbo].[Students_exams_questions]

[dbo].[insert_choices]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@qs_desc	varchar(max)	max
@qs_isCorrect	char	1
@qs_ID	int	4

SQL Script

Uses

[dbo].[choices]

[dbo].[insert_Course]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
Create proc [dbo].[insert_Course] @C_Name varchar(50)
as
begin try

insert into Courses (course_name) values (@C_Name)
end try
begin catch

select'must enter Course desc or type '
end catch
```

Uses

[dbo].[courses]

[dbo].[insert_Department]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Dept_Name	varchar(50)	50
@supervisior_ID	int	4

SQL Script

Uses

[dbo].[department]

[dbo].[insert_Exams]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Course_id	int	4
@time	int	4

SQL Script

```
Create proc [dbo].[insert_Exams] @Course_id int, @time int as begin try

insert into exams (course_ID,duration) values (@Course_id,@time) end try begin catch

select'must enter exam ' end catch

GO
```

Uses

[dbo].[exams]

[dbo].[insert_instructor]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@userName	varchar(50)	50
@pass	varchar(50)	50
@name	varchar(20)	20

SQL Script

```
create proc [dbo].[insert_instructor] @userName varchar(50),@pass varchar(50),@name
varchar(20)='NoName'
as
begin try
insert into instructor (instructor_name,instructor_UserName,instructor_Password) values
(@name,@userName,@pass)
end try
begin catch
select'must enter username and password'
end catch
GO
```

Uses

[dbo].[instructor]

[dbo].[insert_instructorCourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Course_ld	int	4
@Instructor_id	int	4

SQL Script

```
--insert

create PROC [dbo].[insert_instructorCourse] @Course_Id int, @Instructor_id int

AS

BEGIN TRY

INSERT INTO Instructor_Courses

VALUES(@Instructor_id,@Course_Id);

RETURN 1;

END TRY

BEGIN CATCH

RETURN 'error';

END CATCH

GO
```

Uses

[dbo].[Instructor_Courses]

[dbo].[insert_question]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@qs_desc	varchar(max)	max
@qs_type	varchar(10)	10
@course_ID	int	4

SQL Script

```
create PROC [dbo].[insert_question] @qs_desc VARCHAR(max), @qs_type varchar(10)
, @course_ID INT
    AS
    BEGIN TRY
        INSERT INTO questions
        VALUES(@qs_desc,@qs_type,@course_ID);
        RETURN 1;
    END TRY
    BEGIN CATCH
        RETURN 'error';
    END CATCH
GO
```

Uses

[dbo].[questions]

[dbo].[insert_Student]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Fname	varchar(30)	30
@Lname	varchar(30)	30
@Dep	int	4
@Username	varchar(30)	30
@pass	varchar(30)	30

SQL Script

```
create proc [dbo].[insert_Student] @Fname varchar(30),@Lname varchar(30),@Dep
int,@Username varchar(30),@pass varchar(30)
as
begin try
insert into students (student_Fname,student_Lname,student_UserName,student_-
Password,department_ID)
values (@Fname ,@Lname,@Username ,@pass,@Dep )
end try
begin catch
select'Error hapend in inserting new student pleas confirm the data'
end catch
```

Uses

[dbo].[students]

[dbo].[insert_studentCourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Course_ld	int	4
@Student_id	int	4

SQL Script

```
--insert

create PROC [dbo].[insert_studentCourse] @Course_Id int, @Student_id int

AS

BEGIN TRY

INSERT INTO student_courses

VALUES(@Student_id,@Course_Id);

RETURN 1;

END TRY

BEGIN CATCH

RETURN 'error';

END CATCH
```

Uses

[dbo].[student_courses]

[dbo].[insert_studentExam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Exam_ld	int	4
@Student_id	int	4
@grade	int	4

SQL Script

```
--insert

create PROC [dbo].[insert_studentExam] @Exam_Id int, @Student_id int, @grade int

AS

BEGIN TRY

INSERT INTO student_exam

VALUES(@Student_id,@Exam_Id,@grade);

RETURN 1;

END TRY

BEGIN CATCH

RETURN 'error';

END CATCH
```

Uses

[dbo].[student_exam]

[dbo].[insert_studentExamQuestion]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Exam_ld	int	4
@Student_id	int	4
@Question_ld	int	4
@student_answer	varchar(50)	50

SQL Script

Uses

 $[dbo]. [Students_exams_questions] \\$

[dbo].[insert_Topics]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Course_id	int	4
@T_name	varchar(50)	50

SQL Script

```
Create proc [dbo].[insert_Topics] @Course_id int, @T_name varchar(50)
as
begin try

insert into Topics (topic_name,course_ID) values (@T_name,@Course_id)
end try
begin catch

select'must enter Topic '
end catch
GO
```

Uses

[dbo].[topics]

[dbo].[r1_student_info_for_each_department]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
--report 1

create proc [dbo].[r1_student_info_for_each_department] @depID int

as

begin try

select * from students where department_ID = @depID

end try

begin catch

select 'invalid id'

end catch

GO
```

Uses

[dbo].[students]

[dbo].[r2_student_grades]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
--report 2
create proc [dbo].[r2_student_grades] @student_ID int
as
begin try
    select concat(s.student_Fname,' ',s.student_Lname) as 'Full name', c.course_name,
se.exam_grade
    from courses as c, student_courses as sc, student_exam as se, students as s
    where sc.student_ID = @student_ID and se.student_ID = @student_ID
end try
begin catch
    select 'invalid student ID'
end catch
GO
```

Uses

[dbo].[courses]

[dbo].[student_courses]

[dbo].[student_exam]

[dbo].[students]

[dbo].[r3_instructor_courses]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE proc [dbo].[r3_instructor_courses] @instructor_ID int

as

begin try

select i.instructor_name, c.course_name, count(sc.student_ID) as 'number of
students'

from courses as c, instructor as i, Instructor_Courses as ic, student_courses as
sc

where i.instructor_ID = @instructor_ID and ic.instructor_ID = i.instructor_ID and
ic.course_ID = c.course_ID and sc.course_ID = c.course_ID

group by i.instructor_name, c.course_name
end try
begin catch
select 'invalid student ID'
end catch
GO
```

Uses

[dbo].[courses]
[dbo].[instructor]
[dbo].[Instructor_Courses]
[dbo].[student_courses]

[dbo].[r4_course_topics]

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].[r4_course_topics] @course_ID int
as
begin try
    select topic_name from topics where course_ID = @course_ID
end try
begin catch
    select 'invalid course ID'
end catch
GO
```

Uses

[dbo].[topics]

[dbo].[r5_exam_questions]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE proc [dbo].[r5_exam_questions] @exam_ID int

as

begin try

select distinct q.qusetion_desc, c.choice_description

from questions as q, choices as c, Students_exams_questions as seq

where q.question_ID = seq.qustion_ID and q.question_ID = c.question_ID and

seq.exam_ID = @exam_ID

end try

begin catch

select 'invalid exam ID'

end catch

GO
```

Uses

[dbo].[choices]
[dbo].[questions]
[dbo].[Students_exams_questions]

[dbo].[r6_student_exam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@exam_ID	int	4
@student_ID	int	4

SQL Script

```
CREATE proc [dbo].[r6_student_exam] @exam_ID int,@student_ID int

as

begin try

    select concat(s.student_Fname,' ',s.student_Lname) as 'Full name', q.qusetion_desc,
seq.answer as 'student answers', c.choice_description as 'model answer'

    from questions as q, Students_exams_questions as seq, choices as c, students as s

    where q.question_ID = seq.qustion_ID and seq.exam_ID = @exam_ID and seq.student_ID

= @student_ID and q.question_ID = c.question_ID and c.isCorrect = 't'

end try

begin catch
    select 'invalid exam ID'

end catch

GO
```

Uses

[dbo].[choices]

[dbo].[questions]

[dbo].[students]

 $[dbo]. [Students_exams_questions] \\$

[dbo].[select_choices]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@id	varchar(5)	5

SQL Script

```
create PROC [dbo].[select_choices] @id varchar(5)='*'
begin try
      if @id='*'
   begin
      exec('select * from choices')
   end
   else
      if exists ( select choice_ID from choices where choice_ID= CAST(@id AS INT))
       exec('select * from choices where choice_ID='+@id)
       end
       else
       begin
       select 'Error in choice ID'
       end
   end
end try
  begin catch
   select' id not found'
end catch
GO
```

Uses

[dbo].[choices]

[dbo].[select_Course]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@selectedData	varchar(10)	10

```
CREATE proc [dbo].[select Course] @selectedData varchar(10) = '*'
begin try
if @selectedData = '*'
   exec('select' + @selectedData +'from courses')
end
else
   if exists ( select Course_ID from courses where Course_ID= cast(@selectedData as
int))
  exec('select * from courses where Course ID = ' + @selectedData )
   end
   else
  begin
   select 'Error in course ID'
end
end try
begin catch
select 'enter a valid id '
end catch
GO
```

Project > DESKTOP-7QIRQTB\SQLEXPRESS > User databases > SQLProject > Programmability > Store	эd
Procedures > dbo.select_Course	

Uses

[dbo].[courses]

[dbo].[select_department]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@id	varchar(5)	5

SQL Script

```
create PROC [dbo].[select_department] @id varchar(5)='*'
begin try
       if @id='*'
   begin
       exec('select * from department')
   end
   else
      if exists ( select department_ID from department where department_ID=@id)
       exec('select * from students where student_ID='+@id)
       end
       else
       begin
       select 'Error in Student ID'
       end
   end
end try
  begin catch
   select' id not found'
end catch
GO
```

Uses

[dbo].[department]

[dbo].[select_Exams]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@selectedData	varchar(10)	10

```
Create proc [dbo].[select_Exams] @selectedData varchar(10) = '*'
begin try
if @selectedData = '*'
   exec('select' + @selectedData +'from exams')
end
else
begin
  if exists ( select exam ID from exams where exam ID= cast(@selectedData as int))
  exec('select * from exams where exam ID = ' + @selectedData )
   end
   else
  begin
  select 'Error in exam ID'
   end
end
end try
begin catch
select 'enter a valid id '
end catch
GO
```

Procedures > dbo.select_Exams		
Uses		

Project > DESKTOP-7QIRQTB\SQLEXPRESS > User databases > SQLProject > Programmability > Stored

[dbo].[exams]

[dbo].[select_Instructor]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@selectedData	varchar(10)	10

```
CREATE proc [dbo].[select_Instructor] @selectedData varchar(10) = '*'
as
  begin try

if @selectedData = '*'
begin
    exec('select' + @selectedData +'from instructor')
end
else
begin
    exec('select * from instructor where inst_id = ' + @selectedData)
end
end try
begin catch

select 'enter a valid id '
end catch
GO
```

[dbo].[select_instructorCourseByCourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Course_ld	varchar(5)	5

SQL Script

```
create PROC [dbo].[select instructorCourseByCourse] @Course Id varchar(5) = "*"
AS
begin try
      if @Course_Id='*'
   begin
       exec('select * from Instructor Courses')
   end
   else
   begin
     if exists ( select * from Instructor Courses where course ID = CAST(@Course Id
AS INT))
       begin
       exec('select * from Instructor Courses where course ID='+@Course Id)
       else
       begin
       select 'Error in course ID'
       end
   end
end try
  begin catch
   select' id not found'
end catch
GO
```

Uses

[dbo].[Instructor_Courses]

[dbo].[select_instructorCourseByInstructor]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Instructor_id	varchar(5)	5

SQL Script

```
-- 2 select
create PROC [dbo].[select instructorCourseByInstructor] @Instructor id varchar(5) = "*"
begin try
       if @Instructor_id='*'
   begin
       exec('select * from Instructor Courses')
   end
   else
   begin
       if exists ( select * from Instructor Courses where instructor ID =
CAST(@Instructor_id AS INT))
       begin
       exec('select * from Instructor Courses where instructor ID='+@Instructor id)
       else
       begin
       select 'Error in instructor ID'
       end
   end
end try
  begin catch
   select' id not found'
end catch
GO
```

Uses

[dbo].[Instructor_Courses]

[dbo].[select_Questions]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@id	varchar(5)	5

SQL Script

```
create PROC [dbo].[select_Questions] @id varchar(5)='*'
begin try
      if @id='*'
   begin
      exec('select * from questions')
   end
   else
      if exists ( select question_ID from questions where question_ID = CAST(@id AS
INT))
       exec('select * from questions where question ID='+@id)
       end
       else
       begin
       select 'Error in question ID'
       end
   end
end try
  begin catch
   select' id not found'
end catch
GO
```

Uses

[dbo].[questions]

[dbo].[select_Student]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@id	varchar(5)	5

```
create proc [dbo].[select_Student] @id varchar(5)='*'
as
begin try
if @id='*'
begin
exec('select * from students')
end
else
begin
exec('select * from students where student_ID='+@id)
end
end try
begin catch
select' id not found'
end catch

GO
```

[dbo].[select_studentCourseByCourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Course_ld	varchar(5)	5

SQL Script

```
create PROC [dbo].[select studentCourseByCourse] @Course Id varchar(5) = "*"
AS
begin try
      if @Course_Id='*'
   begin
       exec('select * from student courses')
   end
   else
   begin
       if exists ( select * from student courses where course ID = CAST(@Course Id AS
INT))
       begin
       exec('select * from student_courses where course_ID='+@Course_Id)
       else
       begin
       select 'Error in course ID'
       end
   end
end try
  begin catch
   select' id not found'
end catch
GO
```

Uses

[dbo].[student_courses]

[dbo].[select_studentCourseByStudent]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Student_id	varchar(5)	5

SQL Script

```
-- 2 select
create PROC [dbo].[select studentCourseByStudent] @Student id varchar(5) = '*'
begin try
       if @Student_id='*'
   begin
       exec('select * from student courses')
   end
   else
   begin
      if exists ( select * from student courses where @Student id = CAST(@Student id
AS INT))
       begin
       exec('select * from student_courses where student_ID='+@Student_id)
       else
       begin
       select 'Error in student ID'
       end
   end
end try
  begin catch
   select' id not found'
end catch
GO
```

Uses

[dbo].[student_courses]

[dbo].[select_studentExamByExam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Exam_ld	varchar(5)	5

SQL Script

```
create PROC [dbo].[select_studentExamByExam] @Exam_Id varchar(5) = '*'
AS
begin try
      if @Exam Id='*'
   begin
       exec('select * from student_Exam')
   end
   else
       if exists ( select * from student_Exam where exam_ID = CAST(@Exam_Id AS INT))
       exec('select * from student_Exam where exam_ID='+@Exam_Id)
       end
       else
       begin
       select 'Error in Exam ID'
       end
   end
end try
   begin catch
   select' id not found'
end catch
```

Uses

[dbo].[student_exam]

[dbo].[select_studentExamByExamAndStudebt]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Exam_ld	int	4
@student_ld	int	4

SQL Script

```
create PROC [dbo].[select studentExamByExamAndStudebt] @Exam Id int, @student Id int
begin try
if exists ( select * from student_Exam where exam_ID =@Exam_Id and student_ID = @student_Id )
       begin
       exec('select * from student Exam where exam ID='+@Exam Id+' and student ID = '
+@student_Id)
      end
       else
       begin
      select 'Error in Exam ID'
       end
end try
  begin catch
   select'invalid ID'
end catch
GO
```

Uses

[dbo].[student_exam]

[dbo].[select_studentExamByStudent]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Student_id	varchar(5)	5

SQL Script

```
-- 2 select
create PROC [dbo].[select studentExamByStudent] @Student id varchar(5) = "*"
begin try
       if @Student_id='*'
   begin
       exec('select * from student exam')
   end
   else
   begin
       if exists ( select * from student exam where student ID = CAST(@Student id AS
INT))
       begin
       exec('select * from student_exam where student_ID='+@Student_id)
       else
       begin
       select 'Error in student ID'
       end
   end
end try
  begin catch
   select' id not found'
end catch
GO
```

Uses

[dbo].[student_exam]

[dbo].[select_studentExamQuestionByExam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Exam_ld	varchar(5)	5

```
create PROC [dbo].[select_studentExamQuestionByExam] @Exam_Id varchar(5) = "*"
AS
begin try
   if @Exam Id='*'
       exec('select * from Students_exams_questions')
   end
   else
       if exists ( select * from Students_exams_questions where exam_ID = CAST(@Exam_-
Id AS INT))
       exec('select * from Students exams questions where exam ID='+@Exam Id)
       end
       else
       begin
       select 'Error in Exam ID'
   end
end try
   begin catch
   select' id not found'
end catch
GO
```

Project > DESKTOP-7QIRQTB\SQLEXPRESS > User databases > SQLProject > Programmability > Store	d
Procedures > dbo.select_studentExamQuestionBvExam	

Uses

[dbo].[Students_exams_questions]

[dbo].[select_studentExamQuestionByExamAndStudent]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Exam_ld	int	4
@student_ld	int	4

SQL Script

Uses

[dbo].[Students_exams_questions]

[dbo].[select_studentExamQuestionByStudent]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Student_id	varchar(5)	5

SQL Script

```
-- 2 select
create PROC [dbo].[select studentExamQuestionByStudent] @Student id varchar(5) = '*'
begin try
       if @Student_id='*'
   begin
       exec('select * from Students exams questions')
   end
   else
   begin
       if exists ( select * from Students exams questions where student ID =
CAST(@Student_id AS INT))
       begin
       exec('select * from Students exams questions where student ID='+@Student id)
       else
       begin
       select 'Error in student ID'
       end
   end
end try
  begin catch
   select' id not found'
end catch
GO
```

Uses

[dbo].[Students_exams_questions]

[dbo].[select_Topics]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@selectedData	varchar(10)	10

```
Create proc [dbo].[select_Topics] @selectedData varchar(10) = '*'
begin try
if @selectedData = '*'
begin
   exec('select' + @selectedData +'from Topics')
end
else
begin
  if exists ( select Topic ID from Topics where Topic ID= cast(@selectedData as
int))
   exec('select * from Topics where Topic ID = ' + @selectedData)
   else
   begin
   select 'Error in Topic ID'
   end
end try
begin catch
select 'enter a valid id '
end catch
GO
```

Project > DESKTOP-7QIRQTB\SQLEXPRESS > User databases > SQLProject > Programmability > Stored
Procedures > dbo.select_Topics

Uses

[dbo].[topics]

[dbo].[update_choices]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@id	int	4
@qs_desc	varchar(max)	max
@qs_isCorrect	char	1
@qs_ID	int	4

```
create proc [dbo].[update choices] @id int,@qs desc VARCHAR(max), @qs isCorrect char
,@qs_ID INT
as
begin try
if exists ( select choice ID from choices where choice ID=@id)
begin
  update choices
   set choice_description = @qs_desc,
   isCorrect = @qs_isCorrect,
   question ID = @qs ID
   where choice_ID=@id
end
else
   select 'Error in choice ID'
end
end try
begin catch
select 'Error wrong info or id '
end catch
```

Procedures > dbo.update_	_choices		

Project > DESKTOP-7QIRQTB\SQLEXPRESS > User databases > SQLProject > Programmability > Stored

Uses

[dbo].[choices]

[dbo].[update_Course]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
Create proc [dbo].[update_Course] @id int,@C_Name varchar(50)
as
begin try
update Courses set course_name=@C_Name where Course_ID=@id
end try
begin catch
select 'Error in Course ID'
end catch

GO
```

Uses

[dbo].[courses]

[dbo].[update_Department]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@id	int	4
@name	varchar(30)	30
@supervisior_ID	int	4

SQL Script

```
create
       proc [dbo].[update Department] @id int,@name varchar(30),@supervisior ID int
as
begin try
if exists ( select Department_ID from Department where Department_ID=@id)
update Department
set department name=@name,
@supervisior ID=@supervisior ID
where department ID=@id
end
begin
select 'Error in Department ID'
end
end try
begin catch
select 'Error wrond info or id '
end catch
GO
```

Uses

[dbo].[department]

[dbo].[update_Exams]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
Create proc [dbo].[update_Exams] @E_id int,@Course_id int, @time int
as
begin try
update exams set Course_ID=@Course_id, duration=@time where exam_ID= @E_id
end try
begin catch
select 'Error in Exam ID'
end catch

GO
```

Uses

[dbo].[exams]

[dbo].[update_Instructor]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@id	int	4
@name	varchar(30)	30
@username	varchar(30)	30
@pass	varchar(30)	30

SQL Script

```
create proc [dbo].[update_Instructor] @id int,@name varchar(30),@username
varchar(30),@pass varchar(30)
as
begin try
update instructor
set instructor_name=@name,
instructor_UserName=@username,
instructor_Password=@pass
where instructor_ID=@id
end try
begin catch
select 'Error wrond info or id '
end catch
GO
```

Uses

[dbo].[instructor]

[dbo].[update_questions]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@id	int	4
@qs_desc	varchar(max)	max
@qs_type	varchar(10)	10
@course_ID	int	4

```
\label{local_create} \begin{tabular}{ll} create proc [dbo].[update\_questions] @id int,@qs\_desc VARCHAR(max), @qs\_type varchar(10) ,@course_ID INT \\ \end{tabular}
begin try
if exists ( select question ID from questions where question ID=@id)
    update questions
   set qusetion desc = @qs desc,
    question_type = @qs_type,
    course ID = @course ID
    where question ID=@id
end
else
begin
   select 'Error in choice ID'
end
end try
begin catch
select 'Error wrong info or id '
end catch
```

Procedures > dbo.update_question	S		

Project > DESKTOP-7QIRQTB\SQLEXPRESS > User databases > SQLProject > Programmability > Stored

Uses

[dbo].[questions]

[dbo].[update_Student]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@id	int	4
@Fname	varchar(30)	30
@Lname	varchar(30)	30
@Username	varchar(30)	30
@pass	varchar(30)	30

```
create proc [dbo].[update_Student] @id int,@Fname varchar(30),@Lname
varchar(30),@Username varchar(30),@pass varchar(30)
as
begin try
update students set student_Fname-@Fname, student_Lname-@Lname, student_-
Username-@Username,student_Fassword-@pass where student_ID-@id
end try
begin catch
select 'Error in Student ID'
end catch
```

Project > DESKTOP-7QIRQTB\SQLEXPRESS > User databases > SQLProject > Programmability > Stored	
Procedures > dbo.update Student	

Uses

[dbo].[students]

[dbo].[update_studentExam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
create proc [dbo].[update studentExam] @Student id int,@Exam ID INT, @grade int
begin try
if exists ( select exam grade from student exam where student ID=@Student id and exam -
ID = @Exam ID)
begin
   update student_exam
   set exam_grade = @grade
   where student ID=@Student id and exam ID = @Exam ID
end
else
begin
   select 'Error in one of two IDs'
end
end try
begin catch
select 'Error wrong info or id '
end catch
GO
```

Uses

[dbo].[student_exam]

[dbo].[update_studentExamQuestion]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Student_id	int	4
@Exam_ID	int	4
@Question_ld	int	4
@student_answer	varchar(50)	50

SQL Script

```
create proc [dbo].[update_studentExamQuestion] @Student_id int,@Exam_ID INT, @Question_-
Id int, @student_answer varchar(50)
begin try
if exists ( select exam grade from student exam where student ID=@Student id and exam -
ID = @Exam_ID)
begin
   update Students exams questions
   set qustion_ID = @Question_Id, student_answer =@student_answer
   where student ID=@Student id and exam ID = @Exam ID
end
else
begin
   select 'Error in one of two IDs'
end try
begin catch
select 'Error wrong info or id '
end catch
GO
```

Uses

[dbo].[student_exam] [dbo].[Students_exams_questions]

[dbo].[update_Topics]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@E_id	int	4
@Course_id	int	4
@T_name	varchar(50)	50

SQL Script

```
Create proc [dbo].[update_Topics] @E_id int,@Course_id int, @T_name varchar(50)
as
begin try
update Topics set Course_ID=@Course_id , topic_name=@T_name where Topic_ID=@E_id
end try
begin catch
select 'Error in Topic ID'
end catch
GO
```

Uses

[dbo].[topics]



Objects

ame	
00	
uest	
ansour	
rky	
ahby	



Property	Value
Туре	SqlUser
Login Name	sa
Default Schema	dbo

Database Level Permissions

Туре	Action
CONNECT	Grant

SQL Script

GO



Property	Value
Туре	SqlUser
Default Schema	guest

SQL Script

GO			



Property	Value
Туре	SqlUser
Default Schema	dbo

Database Level Permissions

Туре	Action
CONNECT	Grant
VIEW ANY COLUMN MASTER KEY DEFINITION	Grant
VIEW ANY COLUMN ENCRYPTION KEY DEFINITION	Grant
BACKUP DATABASE	Grant
BACKUP LOG	Grant
CREATE DEFAULT	Grant
CREATE FUNCTION	Grant
CREATE PROCEDURE	Grant
CREATE RULE	Grant
CREATE TABLE	Grant
CREATE VIEW	Grant

SQL Script

```
CREATE USER [mansour] WITHOUT LOGIN
GRANT ALTER ANY MASK TO [mansour]
GRANT ALTER ANY SENSITIVITY CLASSIFICATION TO [mansour]
GRANT ALTER ANY EXTERNAL DATA SOURCE TO [mansour]
GRANT ALTER ANY EXTERNAL FILE FORMAT TO [mansour]
GRANT ALTER TO [mansour]
GRANT ALTER ANY ASYMMETRIC KEY TO [mansour]
GRANT ALTER ANY APPLICATION ROLE TO [mansour]
GRANT ALTER ANY ASSEMBLY TO [mansour]
GRANT ALTER ANY CERTIFICATE TO [mansour]
GRANT ALTER ANY DATABASE AUDIT TO [mansour]
GRANT ALTER ANY DATASPACE TO [mansour]
GRANT ALTER ANY DATABASE EVENT NOTIFICATION TO [mansour]
GRANT ALTER ANY FULLTEXT CATALOG TO [mansour]
GRANT ALTER ANY MESSAGE TYPE TO [mansour]
GRANT ALTER ANY ROLE TO [mansour]
GRANT ALTER ANY ROUTE TO [mansour]
```

```
GRANT ALTER ANY REMOTE SERVICE BINDING TO [mansour]
GRANT ALTER ANY CONTRACT TO [mansour]
GRANT ALTER ANY SYMMETRIC KEY TO [mansour]
GRANT ALTER ANY SCHEMA TO [mansour]
GRANT ALTER ANY SECURITY POLICY TO [mansour]
GRANT ALTER ANY SERVICE TO [mansour]
GRANT ALTER ANY DATABASE DDL TRIGGER TO [mansour]
GRANT ALTER ANY USER TO [mansour]
GRANT AUTHENTICATE TO [mansour]
GRANT BACKUP DATABASE TO [mansour]
GRANT BACKUP LOG TO [mansour]
GRANT CONTROL TO [mansour]
GRANT CONNECT REPLICATION TO [mansour]
GRANT CHECKPOINT TO [mansour]
GRANT CREATE AGGREGATE TO [mansour]
GRANT CREATE ASYMMETRIC KEY TO [mansour]
GRANT CREATE ASSEMBLY TO [mansour]
GRANT CREATE CERTIFICATE TO [mansour]
GRANT CREATE DEFAULT TO [mansour]
GRANT CREATE DATABASE DDL EVENT NOTIFICATION TO [mansour]
GRANT CREATE FUNCTION TO [mansour]
GRANT CREATE FULLTEXT CATALOG TO [mansour]
GRANT CREATE MESSAGE TYPE TO [mansour]
GRANT CREATE PROCEDURE TO [mansour]
GRANT CREATE QUEUE TO [mansour]
GRANT CREATE ROLE TO [mansour]
GRANT CREATE ROUTE TO [mansour]
GRANT CREATE RULE TO [mansour]
GRANT CREATE REMOTE SERVICE BINDING TO [mansour]
GRANT CREATE CONTRACT TO [mansour]
GRANT CREATE SYMMETRIC KEY TO [mansour]
GRANT CREATE SCHEMA TO [mansour]
GRANT CREATE SYNONYM TO [mansour]
GRANT CREATE SERVICE TO [mansour]
GRANT CREATE TABLE TO [mansour]
GRANT CREATE TYPE TO [mansour]
GRANT CREATE VIEW TO [mansour]
GRANT CREATE XML SCHEMA COLLECTION TO [mansour]
GRANT DELETE TO [mansour]
GRANT EXECUTE TO [mansour]
GRANT INSERT TO [mansour]
GRANT REFERENCES TO [mansour]
GRANT SELECT TO [mansour]
GRANT SHOWPLAN TO [mansour]
GRANT SUBSCRIBE QUERY NOTIFICATIONS TO [mansour]
GRANT TAKE OWNERSHIP TO [mansour]
GRANT UNMASK TO [mansour]
GRANT UPDATE TO [mansour]
GRANT VIEW ANY SENSITIVITY CLASSIFICATION TO [mansour]
GRANT VIEW DEFINITION TO [mansour]
GRANT VIEW ANY COLUMN ENCRYPTION KEY DEFINITION TO [mansour]
GRANT VIEW ANY COLUMN MASTER KEY DEFINITION TO [mansour]
GRANT VIEW DATABASE STATE TO [mansour]
```

 $\label{lem:project} Project > DESKTOP-7QIRQTB \\ SQLEXPRESS > User \ databases > SQLProject > Security > Users > mansour$

Author: Goharista



Property	Value
Туре	SqlUser
Default Schema	dbo

Database Level Permissions

Туре	Action
CONNECT	Grant

SQL Script

CREATE USER [torky] WITHOUT LOGIN
GO



Property	Value
Туре	SqlUser
Default Schema	dbo

Database Level Permissions

Туре	Action
CONNECT	Grant

SQL Script

CREATE USER [wahby] WITHOUT LOGIN
GO

A 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

Members

- mansour
- wahby

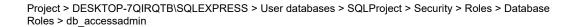
SQL Script

```
ALTER ROLE [db_accessadmin] ADD MEMBER [mansour]

GO

ALTER ROLE [db_accessadmin] ADD MEMBER [wahby]

GO
```



Uses

mansour wahby

db_backupoperator

Properties

Property	Value
Owner	dbo

Members

- mansour
- wahby

SQL Script

```
ALTER ROLE [db_backupoperator] ADD MEMBER [mansour]

GO

ALTER ROLE [db_backupoperator] ADD MEMBER [wahby]

GO
```

Uses

mansour wahby

♣ db_datareader

Properties

Property	Value
Owner	dbo

Members

- mansour
- wahby

SQL Script

```
ALTER ROLE [db_datareader] ADD MEMBER [mansour]

GO

ALTER ROLE [db_datareader] ADD MEMBER [wahby]

GO
```

Uses

mansour wahby

db_datawriter

Properties

Property	Value
Owner	dbo

Members

- mansour
- wahby

SQL Script

```
ALTER ROLE [db_datawriter] ADD MEMBER [mansour]

GO

ALTER ROLE [db_datawriter] ADD MEMBER [wahby]

GO
```

Uses

mansour

Author: Goharista

wahby

db_ddladmin

Properties

Property	Value
Owner	dbo

Members

- mansour
- wahby

SQL Script

```
ALTER ROLE [db_ddladmin] ADD MEMBER [mansour]

GO

ALTER ROLE [db_ddladmin] ADD MEMBER [wahby]

GO
```

Uses

mansour wahby

db_denydatareader

Properties

Property	Value
Owner	dbo

Project > DESKTOP-7QIRQTB\SQLEXPRESS > User databases > SQLProject > Security > Roles > Database Roles > db_denydatareader

Members

- mansour
- wahby

SQL Script

```
ALTER ROLE [db_denydatareader] ADD MEMBER [mansour]

GO

ALTER ROLE [db_denydatareader] ADD MEMBER [wahby]

GO
```

Uses

mansour wahby

db_denydatawriter

Properties

Property	Value
Owner	dbo

Members

- mansour
- wahby

SQL Script

```
ALTER ROLE [db_denydatawriter] ADD MEMBER [mansour]

GO

ALTER ROLE [db_denydatawriter] ADD MEMBER [wahby]

GO
```

Uses

mansour wahby

db_owner

Properties

Property	Value
Owner	dbo

Members

- mansour
- wahby

SQL Script

```
ALTER ROLE [db_owner] ADD MEMBER [mansour]

GO

ALTER ROLE [db_owner] ADD MEMBER [wahby]

GO
```

Uses

mansour wahby

♣ db_securityadmin

Properties

Property	Value
Owner	dbo

Members

- mansour
- wahby

$\label{lem:project} Project > DESKTOP-7QIRQTB \\ SQLEXPRESS > User \ databases > SQLProject > Security > Roles > Database \\ Roles > db_securityadmin$

SQL Script

```
ALTER ROLE [db_securityadmin] ADD MEMBER [mansour]

GO

ALTER ROLE [db_securityadmin] ADD MEMBER [wahby]

GO
```

Uses

mansour wahby

♣ public

Properties

Property	Value
Owner	dbo