(local)\ Documentation

Examination_System

Server	(local)\
Author	Abd-Elrahman Elsayed
Created	م 11:52:08 2023 ,فبراير 28
File Path	C:\Users\Abd-Elrahman Elsayed\Desktop\Examination_system_documentation-2023-02-28T23-52-08.docx

Table of Contents

▤ *(local)*

Databases (1)

- ▤ Examination_System

Server Properties

Property	Value
Product	Microsoft SQL Server
Version	14.0.1000.169
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	8113
Is Clustered	False
Root Directory	C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL
Collation	SQL_Latin1_General_CP1_CI_AS

Server Settings

Property	Value
Default data file path	C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\
Default backup file path	C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\Backup
Default log file path	C:\Program Files\Microsoft SQL Server\MSSQL14.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	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)

- ▤ Examination_System

⊟ Examination_System Database

Database Properties

Property	Value
SQL Server Version	SQL Server 2017
Compatibility Level	SQL Server 2017
Last backup time	02/28/2023
Last log backup time	-
Creation date	Feb 28 2023
Users	4
Database Encryption Enabled	False
Database Encryption Algorithm	None
Database size	16.00 MB
Unallocated space	3.81 MB

Database Options

Property	Value
Compatibility Level	140
Database collation	SQL_Latin1_General_CP1_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	f36650b9-6713-47b6-b295-11c0f5155dca
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-60VPK9C\Abd-Elrahman Elsayed

Files

Name	Type	Size	Maxsize	Autogrowth	File Name
Examination_-System	Data	8.00 MB	unlimited	64.00 MB	C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\Examination_-System.mdf
Examination_-System_log	Log	8.00 MB	2048.00 GB	64.00 MB	C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\Examination_-System_log.ldf

▥ *Tables*

Objects

Name
dbo.Course
dbo.Department
dbo.Exam
dbo.Exam_Questions
dbo.Instructor
dbo.Instructor_Courses
dbo.Question
dbo.Student
dbo.Student_Courses
dbo.Student_Exam
dbo.Topics

🔳 [dbo].[Course]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	6
Created	2023, فبراير 20 ص 11:58:26
Last Modified	2023, فبراير 28 م 05:57:52

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PK C	ID	int	4	NOT NULL
	Name	varchar(50)	50	NULL allowed
	Duration	int	4	NULL allowed

Indexes

Key	Name	Key Columns	Unique
PK C	PK_Course	ID	True

SQL Script

```
CREATE TABLE [dbo].[Course]
(
[ID] [int] NOT NULL,
[Name] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[Duration] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Course] ADD CONSTRAINT [PK Course] PRIMARY KEY CLUSTERED ([ID]) ON
[PRIMARY]
GO
```

Used By

[dbo].[Exam]
[dbo].[Instructor_Courses]
[dbo].[Student_Courses]
[dbo].[Topics]
[dbo].[Course_Topics]
[dbo].[courses_info_for_instrucor]
[dbo].[deleteCourse]
[dbo].[deleteInstCrsRelation]

[dbo].[deleteStdCrsRelation]
[dbo].[Generate_Exam]
[dbo].[insertCourse]
[dbo].[insertInstructorCrs]
[dbo].[insertInstructorCrss]
[dbo].[insertStudentCrs]
[dbo].[Print_Exam_questions]
[dbo].[selectCourse]
[dbo].[Student_Grades_in_each_course]
[dbo].[updateCourseName]
[dbo].[updateInstructorCrs]
[dbo].[updateStudentCrs]
[dbo].[viewInstructorsCoursec]
[dbo].[viewInstructorsCoursecs]
[dbo].[viewStudentsCoursesGrades]

🗐 [dbo].[Department]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	7
Created	2023, فبراير 21 م 09:42:51
Last Modified	2023, فبراير 28 م 05:08:15

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PK C	Dept_Id	int	4	NOT NULL
	Name	varchar(50)	50	NULL allowed
FK	M_SSN	bigint	8	NULL allowed

Indexes

Key	Name	Key Columns	Unique
PK C	PK_Department	Dept_Id	True

Foreign Keys

Name	Columns
FK_Department_Instructor	M_SSN->[dbo].[Instructor].[SSN]

SQL Script

```sql
CREATE TABLE [dbo].[Department]
(
[Dept_Id] [int] NOT NULL,
[Name] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[M_SSN] [bigint] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Department] ADD CONSTRAINT [PK Department] PRIMARY KEY CLUSTERED ([Dept -
Id]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Department] ADD CONSTRAINT [FK Department Instructor] FOREIGN KEY ([M SSN])
REFERENCES [dbo].[Instructor] ([SSN])
GO
```

Uses

[dbo].[Instructor]

Used By

[dbo].[Instructor]
[dbo].[Student]
[dbo].[deleteDeparment]
[dbo].[insertDepartment]
[dbo].[selectAllDepartments]
[dbo].[selectAllInstructors]
[dbo].[selectAllStudents]
[dbo].[selectDepartment]
[dbo].[selectInstructor]
[dbo].[selectStudent]
[dbo].[Students_info_in_Department]
[dbo].[updateDepartmentName]

⊞ [dbo].[Exam]

Properties

Property	Value
Row Count (~)	8
Created	2023 ,فبراير 24 م 11:16:09
Last Modified	2023 ,فبراير 24 م 11:51:33

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PK	ID	int	4	NOT NULL
	Date	date	3	NULL allowed
	Start_Time	time	5	NULL allowed
	Duration	float	8	NULL allowed
FK	Cr_id	int	4	NOT NULL

Indexes

Key	Name	Key Columns	Unique
PK	PK_Exam	ID	True

Foreign Keys

Name	Columns
FK_Exam_Course	Cr_id->[dbo].[Course].[ID]

SQL Script

```
CREATE TABLE [dbo].[Exam]
(
[ID] [int] NOT NULL,
[Date] [date] NULL,
[Start_Time] [time] NULL,
[Duration] [float] NULL,
[Cr_id] [int] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Exam] ADD CONSTRAINT [PK_Exam] PRIMARY KEY CLUSTERED ([ID]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Exam] ADD CONSTRAINT [FK_Exam_Course] FOREIGN KEY ([Cr_id]) REFERENCES
[dbo].[Course] ([ID])
GO
```

Uses

[dbo].[Course]

Used By

[dbo].[Exam_Questions]
[dbo].[Student_Exam]
[dbo].[correctExam]
[dbo].[deleteExam]
[dbo].[deleteExamQ]
[dbo].[deleteStdExamRelation]
[dbo].[Generate_Exam]
[dbo].[insertExam]
[dbo].[insertStudentExam]
[dbo].[insertStudentExamm]
[dbo].[Print_Exam_questions]
[dbo].[selectexam]
[dbo].[updateEExamQu]
[dbo].[updateExam]
[dbo].[updateStudentExam]
[dbo].[viewExamQuestion]
[dbo].[viewStudentsAnswer]

[dbo].[Exam_Questions]

Properties

Property	Value
Row Count (~)	70
Created	2023 ,فبراير 20 م 12:03:53
Last Modified	2023 ,فبراير 26 ص 12:38:51

Columns

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

Indexes

Key	Name	Key Columns	Unique
PK C	PK_Exam_Questions	Ex_ID, Q_ID	True

Foreign Keys

Name	Columns
FK_Exam_Questions_Exam	Ex_ID->[dbo].[Exam].[ID]
FK_Exam_Questions_Question	Q_ID->[dbo].[Question].[ID]

SQL Script

```
CREATE TABLE [dbo].[Exam_Questions]
(
[Ex_ID] [int] NOT NULL,
[Q_ID] [int] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Exam_Questions] ADD CONSTRAINT [PK_Exam_Questions] PRIMARY KEY CLUSTERED
([Ex_ID], [Q_ID]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Exam_Questions] ADD CONSTRAINT [FK_Exam_Questions_Exam] FOREIGN KEY ([Ex_ID]) REFERENCES [dbo].[Exam] ([ID])
GO
ALTER TABLE [dbo].[Exam_Questions] ADD CONSTRAINT [FK_Exam_Questions_Question] FOREIGN KEY
([Q_ID]) REFERENCES [dbo].[Question] ([ID])
GO
```

Uses

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

Used By

[dbo].[correctExam]
[dbo].[deleteExamQ]
[dbo].[Exam_Answers]
[dbo].[Generate_Exam]
[dbo].[insertStudentExamm]
[dbo].[ModelAnswer]
[dbo].[Print_Exam_questions]
[dbo].[updateEExamQu]
[dbo].[viewExamQuestion]

🖼 [dbo].[Instructor]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	4
Created	2023, فبراير 21 م 09:42:50
Last Modified	2023, فبراير 21 م 11:18:37

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PK C	SSN	bigint	8	NOT NULL
	Fname	varchar(50)	50	NULL allowed
	Lname	varchar(50)	50	NULL allowed
	Age	int	4	NULL allowed
	Gender	nchar(1)	2	NULL allowed
	City	varchar(50)	50	NULL allowed
	Street	varchar(50)	50	NULL allowed
	Building	int	4	NULL allowed
	Phone	int	4	NULL allowed
FK	Dept_Id	int	4	NULL allowed

Indexes

Key	Name	Key Columns	Unique
PK C	PK_Instructor	SSN	True

Foreign Keys

Name	Columns
FK_Instructor_Department	Dept_Id->[dbo].[Department].[Dept_Id]

SQL Script

```
CREATE TABLE [dbo].[Instructor]
(
[SSN] [bigint] NOT NULL,
[Fname] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[Lname] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[Age] [int] NULL,
[Gender] [nchar] (1) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
```

```
[City] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[Street] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[Building] [int] NULL,
[Phone] [int] NULL,
[Dept_Id] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Instructor] ADD CONSTRAINT [PK Instructor] PRIMARY KEY CLUSTERED ([SSN]) ON
[PRIMARY]
GO
ALTER TABLE [dbo].[Instructor] ADD CONSTRAINT [FK Instructor Department] FOREIGN KEY ([Dept -
Id]) REFERENCES [dbo].[Department] ([Dept_Id])
GO
```

Uses

[dbo].[Department]

Used By

[dbo].[Department]
[dbo].[Instructor_Courses]
[dbo].[courses_info_for_instrucor]
[dbo].[deleteInstCrsRelation]
[dbo].[deleteInstructor]
[dbo].[insertInstructor]
[dbo].[insertInstructorCrs]
[dbo].[insertInstructorCrss]
[dbo].[selectAllInstructors]
[dbo].[selectInstructor]
[dbo].[updateInstructorCity]
[dbo].[updateInstructorCrs]
[dbo].[viewInstructorsCoursec]
[dbo].[viewInstructorsCoursecs]

📰 [dbo].[Instructor_Courses]

Properties

Property	Value
Row Count (~)	4
Created	2023, فبراير 21 م 09:42:51
Last Modified	2023, فبراير 21 م 09:42:51

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PK FK C	In_SSN	bigint	8	NOT NULL
PK FK C	Cr_Id	int	4	NOT NULL

Indexes

Key	Name	Key Columns	Unique
PK C	PK_Instructor_Courses	In_SSN, Cr_Id	True

Foreign Keys

Name	Columns
FK_Instructor_Courses_Course	Cr_Id->[dbo].[Course].[ID]
FK_Instructor_Courses_Instructor	In_SSN->[dbo].[Instructor].[SSN]

SQL Script

```sql
CREATE TABLE [dbo].[Instructor_Courses]
(
[In_SSN] [bigint] NOT NULL,
[Cr_Id] [int] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Instructor_Courses] ADD CONSTRAINT [PK_Instructor_Courses] PRIMARY KEY
CLUSTERED ([In_SSN], [Cr_Id]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Instructor_Courses] ADD CONSTRAINT [FK_Instructor_Courses_Course] FOREIGN
KEY ([Cr_Id]) REFERENCES [dbo].[Course] ([ID])
GO
ALTER TABLE [dbo].[Instructor_Courses] ADD CONSTRAINT [FK_Instructor_Courses_Instructor]
FOREIGN KEY ([In_SSN]) REFERENCES [dbo].[Instructor] ([SSN])
GO
```

Uses

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

Used By

[dbo].[courses_info_for_instrucor]
[dbo].[deleteInstCrsRelation]
[dbo].[insertInstructorCrs]
[dbo].[insertInstructorCrss]
[dbo].[updateInstructorCrs]
[dbo].[viewInstructorsCoursec]
[dbo].[viewInstructorsCoursecs]

📼 [dbo].[Question]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	21
Created	2023, فبراير 26 ص 12:38:51
Last Modified	2023, فبراير 26 ص 12:38:51

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PK C	ID	int	4	NOT NULL
	Type	varchar(50)	50	NULL allowed
	Content	varchar(1000)	1000	NULL allowed
	Cr_Id	int	4	NULL allowed
	Correct_Answer	nchar(1)	2	NULL allowed
	Choice A	varchar(700)	700	NULL allowed
	Choice B	varchar(700)	700	NULL allowed
	Choice C	varchar(700)	700	NULL allowed
	Choice D	varchar(700)	700	NULL allowed
	Grade	float	8	NULL allowed

Indexes

Key	Name	Key Columns	Unique
PK C	PK_Question	ID	True

SQL Script

```
CREATE TABLE [dbo].[Question]
(
[ID] [int] NOT NULL,
[Type] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[Content] [varchar] (1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[Cr_Id] [int] NULL,
[Correct_Answer] [nchar] (1) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[Choice A] [varchar] (700) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[Choice B] [varchar] (700) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[Choice C] [varchar] (700) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[Choice D] [varchar] (700) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[Grade] [float] NULL
) ON [PRIMARY]
GO
```

```
ALTER TABLE [dbo].[Question] ADD CONSTRAINT [PK Question] PRIMARY KEY CLUSTERED ([ID]) ON
[PRIMARY]
GO
```

Used By

[dbo].[Exam_Questions]

[dbo].[Student_Exam]

[dbo].[correctExam]

[dbo].[deleteExamQ]

[dbo].[deleteQuetion]

[dbo].[Exam_Questions_And_Student_Answers]

[dbo].[Generate_Exam]

[dbo].[insertStudentExam]

[dbo].[insertStudentExamm]

[dbo].[insetQuestion]

[dbo].[ModelAnswer]

[dbo].[Print_Exam_questions]

[dbo].[SelectQuetions]

[dbo].[updateEExamQu]

[dbo].[updateQuetions]

[dbo].[viewExamQuestion]

⊞ [dbo].[Student]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	3
Created	2023, فبراير 21 م 10:57:36
Last Modified	2023, فبراير 28 م 05:08:15

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PK C	SSN	bigint	8	NOT NULL
	Fname	varchar(50)	50	NULL allowed
	Lname	varchar(50)	50	NULL allowed
	Age	int	4	NULL allowed
	Faculty	varchar(50)	50	NULL allowed
	Gender	nchar(1)	2	NULL allowed
	City	varchar(50)	50	NULL allowed
	Street	varchar(50)	50	NULL allowed
	Building	int	4	NULL allowed
	Phone	int	4	NULL allowed
FK	Dept_ID	int	4	NULL allowed

Indexes

Key	Name	Key Columns	Unique
PK C	PK_Student	SSN	True

Foreign Keys

Name	Columns
FK_Student_Department	Dept_ID->[dbo].[Department].[Dept_Id]

SQL Script

```
CREATE TABLE [dbo].[Student]
(
[SSN] [bigint] NOT NULL,
[Fname] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[Lname] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
```

```
[Age] [int] NULL,
[Faculty] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[Gender] [nchar] (1) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[City] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[Street] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[Building] [int] NULL,
[Phone] [int] NULL,
[Dept_ID] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Student] ADD CONSTRAINT [PK Student] PRIMARY KEY CLUSTERED ([SSN]) ON
[PRIMARY]
GO
ALTER TABLE [dbo].[Student] ADD CONSTRAINT [FK Student Department] FOREIGN KEY ([Dept ID])
REFERENCES [dbo].[Department] ([Dept_Id])
GO
```

Uses

[dbo].[Department]

Used By

[dbo].[Student_Courses]
[dbo].[Student_Exam]
[dbo].[deleteStdCrsRelation]
[dbo].[deleteStdExamRelation]
[dbo].[deleteStudent]
[dbo].[Exam_Questions_And_Student_Answers]
[dbo].[insertStudent]
[dbo].[insertStudentCrs]
[dbo].[insertStudentExam]
[dbo].[selectAllStudents]
[dbo].[selectStudent]
[dbo].[Student_Grades_in_each_course]
[dbo].[Students_info_in_Department]
[dbo].[updateStudentCrs]
[dbo].[updateStudentExam]
[dbo].[updateStudentFaculty]
[dbo].[viewStudentsAnswer]
[dbo].[viewStudentsCoursesGrades]

📇 [dbo].[Student_Courses]

Properties

Property	Value
Row Count (~)	1
Created	2023 ,فبراير 26 ص 12:29:05
Last Modified	2023 ,فبراير 26 ص 12:29:05

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PK FK C	Std_SSN	bigint	8	NOT NULL
PK FK C	Cr_Id	int	4	NOT NULL
	Grade	int	4	NULL allowed

Indexes

Key	Name	Key Columns	Unique
PK C	PK_Student_Courses_1	Std_SSN, Cr_Id	True

Foreign Keys

Name	Columns
FK_Student_Courses_Course	Cr_Id->[dbo].[Course].[ID]
FK_Student_Courses_Student	Std_SSN->[dbo].[Student].[SSN]

SQL Script

```sql
CREATE TABLE [dbo].[Student_Courses]
(
[Std_SSN] [bigint] NOT NULL,
[Cr_Id] [int] NOT NULL,
[Grade] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Student_Courses] ADD CONSTRAINT [PK_Student_Courses_1] PRIMARY KEY
CLUSTERED ([Std_SSN], [Cr_Id]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Student_Courses] ADD CONSTRAINT [FK_Student_Courses_Course] FOREIGN KEY
([Cr_Id]) REFERENCES [dbo].[Course] ([ID])
GO
ALTER TABLE [dbo].[Student_Courses] ADD CONSTRAINT [FK_Student_Courses_Student] FOREIGN KEY
([Std_SSN]) REFERENCES [dbo].[Student] ([SSN])
GO
```

Uses

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

Used By

[dbo].[correctExam]
[dbo].[courses_info_for_instrucor]
[dbo].[deleteStdCrsRelation]
[dbo].[insertStudentCrs]
[dbo].[Student_Grades_in_each_course]
[dbo].[updateStudentCrs]
[dbo].[viewStudentsCoursesGrades]

[dbo].[Student_Exam]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	20
Created	2023, فبراير 21 م 10:57:37
Last Modified	2023, فبراير 26 ص 12:38:51

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PK FK C	Std_SSN	bigint	8	NOT NULL
PK FK C	Ex_ID	int	4	NOT NULL
PK FK C	Q_ID	int	4	NOT NULL
	Std_Anwser	varchar(50)	50	NULL allowed

Indexes

Key	Name	Key Columns	Unique
PK C	PK_Student_Exam	Std_SSN, Ex_ID, Q_ID	True

Foreign Keys

Name	Columns
FK_Student_Exam_Exam	Ex_ID->[dbo].[Exam].[ID]
FK_Student_Exam_Question	Q_ID->[dbo].[Question].[ID]
FK_Student_Exam_Student	Std_SSN->[dbo].[Student].[SSN]

SQL Script

```
CREATE TABLE [dbo].[Student_Exam]
(
[Std_SSN] [bigint] NOT NULL,
[Ex_ID] [int] NOT NULL,
[Q_ID] [int] NOT NULL,
[Std_Anwser] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Student_Exam] ADD CONSTRAINT [PK_Student_Exam] PRIMARY KEY CLUSTERED
([Std_SSN], [Ex_ID], [Q_ID]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Student_Exam] ADD CONSTRAINT [FK_Student_Exam_Exam] FOREIGN KEY ([Ex_ID])
```

```
REFERENCES [dbo].[Exam] ([ID])
GO
ALTER TABLE [dbo].[Student_Exam] ADD CONSTRAINT [FK_Student_Exam_Question] FOREIGN KEY ([Q_-
ID]) REFERENCES [dbo].[Question] ([ID])
GO
ALTER TABLE [dbo].[Student Exam] ADD CONSTRAINT [FK Student Exam Student] FOREIGN KEY ([Std -
SSN]) REFERENCES [dbo].[Student] ([SSN])
GO
```

Uses

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

Used By

[dbo].[correctExam]
[dbo].[deleteStdExamRelation]
[dbo].[Exam_Answers]
[dbo].[Exam_Questions_And_Student_Answers]
[dbo].[insertStudentExam]
[dbo].[updateStudentExam]
[dbo].[viewStudentsAnswer]

📧 [dbo].[Topics]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	6
Created	2023, فبراير 28 م 05:52:20
Last Modified	2023, فبراير 28 م 05:57:52

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PK C	Id	int	4	NOT NULL
	Name	varchar(50)	50	NOT NULL
FK	Cr_Id	int	4	NULL allowed

Indexes

Key	Name	Key Columns	Unique
PK C	PK_Topics	Id	True

Foreign Keys

Name	Columns
FK_Topics_Course	Cr_Id->[dbo].[Course].[ID]

SQL Script

```sql
CREATE TABLE [dbo].[Topics]
(
[Id] [int] NOT NULL,
[Name] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
[Cr_Id] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Topics] ADD CONSTRAINT [PK Topics] PRIMARY KEY CLUSTERED ([Id]) ON
[PRIMARY]
GO
ALTER TABLE [dbo].[Topics] ADD CONSTRAINT [FK Topics Course] FOREIGN KEY ([Cr Id]) REFERENCES
[dbo].[Course] ([ID])
GO
```

Uses

[dbo].[Course]

Used By

[dbo].[Course_Topics]

📖 *Stored Procedures*

Objects

Name
dbo.correctExam
dbo.Course_Topics
dbo.courses_info_for_instrucor
dbo.deleteCourse
dbo.deleteDeparment
dbo.deleteExam
dbo.deleteExamQ
dbo.deleteInstCrsRelation
dbo.deleteInstructor
dbo.deleteQuetion
dbo.deleteStdCrsRelation
dbo.deleteStdExamRelation
dbo.deleteStudent
dbo.Exam_Answers
dbo.Exam_Questions_And_Student_Answers
dbo.Generate_Exam
dbo.insertCourse
dbo.insertDepartment
dbo.insertExam
dbo.insertInstructor
dbo.insertInstructorCrs
dbo.insertInstructorCrss
dbo.insertStudent
dbo.insertStudentCrs
dbo.insertStudentExam
dbo.insertStudentExamm
dbo.insetQuestion
dbo.ModelAnswer
dbo.Print_Exam_questions
dbo.selectAllDepartments
dbo.selectAllInstructors
dbo.selectAllStudents
dbo.selectCourse
dbo.selectDepartment
dbo.selectexam
dbo.selectInstructor
dbo.SelectQuetions

dbo.selectStudent
dbo.Student_Grades_in_each_course
dbo.Students_info_in_Department
dbo.updateCourseName
dbo.updateDepartmentName
dbo.updateEExamQu
dbo.updateExam
dbo.updateInstructorCity
dbo.updateInstructorCrs
dbo.updateQuetions
dbo.updateStudentCrs
dbo.updateStudentExam
dbo.updateStudentFaculty
dbo.viewExamQuestion
dbo.viewInstructorsCoursec
dbo.viewInstructorsCoursecs
dbo.viewStudentsAnswer
dbo.viewStudentsCoursesGrades

🖩 [dbo].[correctExam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Std_SSN	bigint	8
@Ex_Id	int	4

SQL Script

```sql
CREATE procedure [dbo].[correctExam](@Std_SSN bigint, @Ex_Id int)
as
    declare @table table (q_id int, correct nchar(1), std_anwser nchar(1))
    insert into @table
    select q.id, q.correct_answer, sx.std_anwser
    from question q, exam_questions eq, Student_Exam sx
    where q.id = eq.q_id and q.id = sx.q_id and eq.ex_id = @ex_id and sx.std_ssn = @std_ssn
    declare @correct nchar(1), @std_anwser nchar(1)

    declare @i int = 0, @grade int = 0
    while @i < 10
       begin
            select top(@i) @correct = correct, @std_anwser = std_anwser
            from @table
            if(@correct = @std_anwser)
                begin
                    set @grade += 1
                end
            set @i += 1
       end
    begin try
        declare @cr_id int
        select @cr_id = cr_id
        from exam
        where id = @ex_id
        insert into student_courses (std_ssn, cr_id, grade)
        values (@Std_SSN, @cr_id, @grade)
    end try
    begin catch
        select 'The Exam Id or Student SSN not found', error_message()
    end catch
GO
```

Uses

[dbo].[Exam]
[dbo].[Exam_Questions]
[dbo].[Question]
[dbo].[Student_Courses]
[dbo].[Student_Exam]

📄 [dbo].[Course_Topics]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE procedure [dbo].[Course_Topics] (@CrId int)
as
    declare @Course_Name varchar(50)
    select @Course_Name = name
        from course
        where id = @crid
    if @Course_Name is not null
        begin
            select @Course_Name as [Course Name] , Name as Topics_Name
            from topics
            where cr_id = @CrId
        end
    else
        begin
            select 'This couse is not found' as Error
        end
GO
```

Uses

[dbo].[Course]
[dbo].[Topics]

📰 [dbo].[courses_info_for_instrucor]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Ins_Id	bigint	8

SQL Script

```sql
CREATE PROC [dbo].[courses_info_for_instrucor]
@Ins_Id BIGINT
AS
BEGIN
    BEGIN TRY
        IF EXISTS(SELECT * FROM Instructor i WHERE i.SSN= @Ins_Id)
            BEGIN
                declare @Instructor_Name varchar(100)
                select @Instructor_Name = (i.Fname + ' ' +i.Lname)
                from instructor i
                where i.SSN = @Ins_Id

                SELECT @Instructor_Name as [Instructor Name], c.Name as [Course Name] ,
COUNT(sc.Std_SSN) AS 'Number of students'
                FROM Course c INNER JOIN Instructor_Courses ic
                ON c.ID = ic.Cr_Id  AND  ic.In_SSN = @Ins_Id
                            INNER JOIN Student_Courses sc
                            ON c.ID = sc.Cr_Id
                            GROUP BY c.Name
            END
        ELSE
            BEGIN
                SELECT 'This instructor does not exist' as Error
            END
    END TRY
    BEGIN CATCH
        SELECT 'An error occurred while processing the request' as Error
    END CATCH
END
GO
```

Uses

[dbo].[Course]
[dbo].[Instructor]
[dbo].[Instructor_Courses]
[dbo].[Student_Courses]

📄 [dbo].[deleteCourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```sql
create proc [dbo].[deleteCourse] @ID int
 as
 begin try
    delete Course
    where ID=@ID
end try
 begin catch
       select 'QueryError, Nothing Has Executed'
  end catch
GO
```

Uses

[dbo].[Course]

📄 [dbo].[deleteDeparment]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```sql
create proc [dbo].[deleteDeparment] @Name varchar(20)
as
    begin try
        delete from Department
        where Name=@Name
    end try
    begin catch
        select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

[dbo].[Department]

📄 [dbo].[deleteExam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
create proc [dbo].[deleteExam] @ID int
 as
 begin try
    delete Exam
    where ID=@ID
 end try
 begin catch
        select 'QueryError, Nothing Has Executed'
 end catch
GO
```

Uses

[dbo].[Exam]

[dbo].[deleteExamQ]

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

```sql
create proc [dbo].[deleteExamQ] @Ex_ID int,@Q_ID int
as
    begin try
        delete from Exam_Questions
        where Q_ID=(select ID from Question where ID=@Q_ID )
        and
        Ex_ID=(select ID from Exam where ID=@Ex_ID)
    end try
    begin catch
        select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

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

📄 [dbo].[deleteInstCrsRelation]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@In_SSN	bigint	8
@Cr_Id	int	4

SQL Script

```
create proc [dbo].[deleteInstCrsRelation]  @In_SSN bigint,@Cr_Id int
as
    begin try
        delete from Instructor_Courses
        where In_SSN=(select SSN from Instructor where In_SSN=@In_SSN)
        And
        Cr_Id =(select ID from Course where Cr_Id=@Cr_Id)
    end try
    begin catch
        select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

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

📄 [dbo].[deleteInstructor]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@SSNI	bigint	8

SQL Script

```sql
create proc [dbo].[deleteInstructor] @SSNI bigint
as
    begin try
        delete from Instructor
        where SSN=@SSNI
    end try
    begin catch
        select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

[dbo].[Instructor]

📄 [dbo].[deleteQuetion]

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].[deleteQuetion] @ID int
 as
 begin try
   delete
   from Question
   where ID=@ID
   end try
   begin catch
       select 'QueryError, Nothing Has Executed'
   end catch
GO
```

Uses

[dbo].[Question]

📄 [dbo].[deleteStdCrsRelation]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Std_SSN	bigint	8
@Cr_Id	int	4

SQL Script

```
create proc [dbo].[deleteStdCrsRelation] @Std_SSN bigint,@Cr_Id int
as
    begin try
        delete from Student_Courses
        where Std_SSN=(select SSN from Student where SSN=@Std_SSN)
        And
        Cr_Id =(select ID from Course where Cr_Id=@Cr_Id)
    end try
    begin catch
        select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

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

🗒 [dbo].[deleteStdExamRelation]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Std_SSN	bigint	8
@Ex_ID	int	4

SQL Script

```
create proc [dbo].[deleteStdExamRelation] @Std_SSN bigint,@Ex_ID int
as
    begin try
        delete from Student_Exam
        where Std_SSN=(select SSN from Student where SSN=@Std_SSN)
        And
        Ex_ID =(select e.ID  from Exam as e where ID=@Ex_ID)

    end try
    begin catch
        select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

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

[dbo].[deleteStudent]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@SSN	bigint	8

SQL Script

```
create proc [dbo].[deleteStudent] @SSN bigint
as
    begin try
        delete from Student
        where SSN=@SSN
    end try
    begin catch
        select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

[dbo].[Student]

[dbo].[Exam_Answers]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Std_SSN	bigint	8
@Ex_Id	int	4
@std_answer	nchar(19)	38

SQL Script

```sql
CREATE procedure [dbo].[Exam_Answers] (@Std_SSN bigint, @Ex_Id int ,@std_answer nchar(19))
as
    begin try
        declare @table1 table (JoiningID1 INT IDENTITY(1,1), anwser nchar(1))
        insert into @table1
        SELECT value
            FROM STRING_SPLIT(@std_answer, ',')

        declare @table2 table (JoiningID2 INT IDENTITY(1,1), q_id int)
        insert into @table2
        select q_id
        from exam_questions eq
        where eq.ex_id = @Ex_Id

        insert into student_exam (std_ssn, ex_id, q_id, std_anwser)
        select @std_ssn, @ex_id, t2.q_id, t1.anwser
        from @table1 t1, @table2 t2
        where JoiningID2 = JoiningID1

    end try
    begin catch
        select 'There is a problem', error_message()
    end catch
GO
```

Uses

[dbo].[Exam_Questions]
[dbo].[Student_Exam]

🖼 [dbo].[Exam_Questions_And_Student_Answers]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@ExamID	int	4
@Stu_Id	bigint	8

SQL Script

```sql
CREATE PROC [dbo].[Exam_Questions_And_Student_Answers]
(@ExamID INT , @Stu_Id BIGINT)
AS
BEGIN
    BEGIN TRY
        declare @Student_Name varchar(100)
            SELECT @Student_Name = (s.Fname + ' ' + s.Lname)
            FROM Student s, Student_Exam se
            WHERE s.SSN= @Stu_Id
                AND
                se.Ex_ID = @ExamID
                and
                s.SSN = se.Std_SSN
        IF @Student_Name is not null
            BEGIN
                SELECT @Student_Name as [Student Name] ,q.Content, se.Std_Anwser
                FROM Student_Exam se INNER JOIN Question q
                    ON se.Q_ID = q.ID
                WHERE se.Ex_ID = @ExamID AND se.Std_SSN = @Stu_Id
            END
        ELSE
            BEGIN
                SELECT 'This exam or this student does not exist' as Error
            END
    END TRY
    BEGIN CATCH
        SELECT 'An error occurred while processing the request' as Error
    END CATCH
END

GO
```

Uses

[dbo].[Question]
[dbo].[Student]
[dbo].[Student_Exam]

📄 [dbo].[Generate_Exam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@course_name	varchar(10)	10
@no_of_mcq	int	4
@no_of_t_or_f	int	4

SQL Script

```sql
CREATE PROC [dbo].[Generate_Exam] @course_name VARCHAR(10), @no_of_mcq INT, @no_of_t_or_f INT
    AS
    if @no_of_mcq + @no_of_t_or_f = 10
    begin
        DECLARE @viewedExam TABLE
                (quesID INT, question VARCHAR(1000), choice1 VARCHAR(700), choice2
VARCHAR(700), choice3 VARCHAR(700), choice4 VARCHAR(700))
                    INSERT INTO @viewedExam
                SELECT * from
                    (SELECT TOP (@no of mcq) q.ID, q.content AS QUESTION,q.[Choice A],
q.[Choice B], q.[Choice C], q.[Choice D]
                        FROM Question q, Course c
                            WHERE q.type = 'MCQ' AND q.cr_id = c.ID AND c.name = @course_name
                                ORDER BY NEWID()) MCQ
                UNION ALL
                SELECT * from
                    (SELECT TOP (@no_of_t_or_f) q.ID, q.content AS QUESTION,q.[Choice A],
q.[Choice B], q.[Choice C], q.[Choice D]
                        FROM Question q, Course c
                            WHERE q.type = 'T&F' AND q.cr_id = c.ID AND c.name = @course_name
                                ORDER BY NEWID()) TrueFalse


                begin try
                    DECLARE @examId INT
                    SELECT @examId = NEXT VALUE FOR mySeq
                    INSERT INTO Exam (ID, Duration, Date, Start_Time, cr_id)
                    VALUES (@examId, 120, Convert(date, getdate()), cast(getdate() as
time(0)), (SELECT ID FROM Course WHERE name = @course_name));
                            INSERT INTO exam_questions (ex_id, q_id)
                    SELECT TOP 10 (@examId), quesID FROM @viewedExam

                    SELECT question, choice1, choice2, choice3, choice4 FROM @viewedExam
                end try
```

```
                begin catch
                    select 'Course name is wrong'
                end catch
    end
    else
        begin
            select 'Number of questions must be 10 only'
        end
GO
```

Uses

[dbo].[Course]
[dbo].[Exam]
[dbo].[Exam_Questions]
[dbo].[Question]
[dbo].[mySeq]

📰 [dbo].[insertCourse]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@ID	int	4
@Name	varchar(50)	50
@Duration	int	4

SQL Script

```sql
create proc [dbo].[insertCourse] (
@ID int,
@Name varchar(50),
@Duration int)
as
  begin try
   insert into Course
   Values(@ID,@Name,@Duration)
  end try
  begin catch
      select 'QueryError, Nothing Has Executed'
  end catch
GO
```

Uses

[dbo].[Course]

📄 [dbo].[insertDepartment]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Dept_Id	int	4
@Name	varchar(50)	50
@M_SSN	bigint	8

SQL Script

```sql
create proc [dbo].[insertDepartment]
(
@Dept_Id int,
@Name varchar(50),
@M_SSN bigint
)
as
    begin try
        insert into Department
        values(@Dept_Id,@Name,@M_SSN)
    end try
    begin catch
        select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

[dbo].[Department]

📄 [dbo].[insertExam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@ID	int	4
@Date	date	3
@Start_Time	time	5
@Duration	float	8
@Cr_id	int	4

SQL Script

```
create proc [dbo].[insertExam]
(
@ID int,
@Date date,
@Start_Time time(7),
@Duration float ,
@Cr_id int
)
as
    begin try
    insert into Exam
    Values(@ID,@Date,@Start_Time,@Duration,@Cr_id)
    end try
    begin catch
        select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

[dbo].[Exam]

📄 [dbo].[insertInstructor]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@SSN	bigint	8
@Fname	varchar(50)	50
@Lname	varchar(50)	50
@Age	int	4
@Gender	nchar	1
@City	varchar(50)	50
@Street	varchar(50)	50
@Building	int	4
@Phone	int	4
@Dept_Id	int	4

SQL Script

```sql
create proc [dbo].[insertInstructor]
(
    @SSN bigint,
    @Fname varchar(50),
    @Lname varchar(50),
    @Age int,

    @Gender nchar(1),
    @City varchar(50),
    @Street varchar(50),
    @Building int,
    @Phone int,
    @Dept_Id int
)
as
    begin try
        insert into Instructor
        values(@SSN,@Fname,@Lname,@Age,@Gender,@City,@Street,@Building,@Phone,@Dept_Id)
    end try
    begin catch
        select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

[dbo].[Instructor]

📋 [dbo].[insertInstructorCrs]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@In_SSN	bigint	8
@Cr_Id	varchar(20)	20

SQL Script

```sql
create proc [dbo].[insertInstructorCrs]
(
@In_SSN BIGint,
@Cr_Id varchar(20)
)
as
    begin try
        insert into Instructor_Courses
        select(select SSN from Instructor where SSN= @In_SSN ),
        (select ID from Course where ID= @Cr_Id)
    end try
    begin catch
        select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

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

🖼 [dbo].[insertInstructorCrss]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@In_SSN	bigint	8
@Cr_Id	int	4

SQL Script

```
create proc [dbo].[insertInstructorCrss]
(
@In_SSN BIGint,
@Cr_Id int
)
as
    begin try
        insert into Instructor_Courses
        select(select SSN from Instructor where SSN= @In_SSN ),
        (select ID from Course where ID= @Cr_Id)
    end try
    begin catch
        select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

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

📄 [dbo].[insertStudent]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@SSN	bigint	8
@Fname	varchar(50)	50
@Lname	varchar(50)	50
@Age	int	4
@Faculty	varchar(50)	50
@Gender	nchar	1
@City	varchar(50)	50
@Street	varchar(50)	50
@Building	int	4
@Phone	int	4
@Dept_ID	int	4

SQL Script

```
create proc [dbo].[insertStudent]
(
    @SSN bigint,
    @Fname varchar(50),
    @Lname varchar(50),
    @Age int,
    @Faculty varchar(50),
    @Gender nchar(1),
    @City varchar(50),
    @Street varchar(50),
    @Building int,
    @Phone int,
    @Dept_ID int
)
as
    begin try
        insert into Student
        values(@SSN,@Fname,@Lname,@Age,@Faculty,@Gender,@City,@Street,@Building,@Phone,@Dept_-
ID)
    end try
    begin catch
        select 'QueryError, Nothing Has Executed'
```

```
    end catch
GO
```

Uses

[dbo].[Student]

[dbo].[insertStudentCrs]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Std_SSN	bigint	8
@Cr_Id	int	4
@Grade	float	8

SQL Script

```sql
create proc [dbo].[insertStudentCrs]
(
@Std_SSN bigint,
@Cr_Id INT,
@Grade FLOAT
)
as
    begin try
        insert into Student_Courses
        select(select SSN from Student where SSN= @Std_SSN ),
        (select ID from Course where ID= @Cr_Id),
        @grade
    end try
    begin catch
        select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

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

📄 [dbo].[insertStudentExam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Std_SSN	bigint	8
@Ex_ID	int	4
@Q_ID	int	4
@Std_Anwser	varchar(50)	50

SQL Script

```
create proc [dbo].[insertStudentExam]
(
@Std_SSN bigint,
@Ex_ID INT,
@Q_ID int,
@Std_Anwser varchar(50))
as
    begin try
        insert into Student_Exam
        select(select SSN from Student where SSN= @Std_SSN ),
        (select ID from Exam where ID= @Ex_ID),
        (select ID from Question where ID= @Q_ID),
        Std_Anwser= @Std_Anwser
    end try
    begin catch
        select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

[dbo].[Exam]
[dbo].[Question]
[dbo].[Student]
[dbo].[Student_Exam]

📄 [dbo].[insertStudentExamm]

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

```sql
create proc [dbo].[insertStudentExamm]
(

@Ex_ID INT,
@Q_ID int
)
as
    begin try
        insert into Exam_Questions
        select(select ID from Exam where ID= @Ex_ID ),
        (select ID from Question where ID= @Q_ID)

    end try
    begin catch
        select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

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

📄 [dbo].[insetQuestion]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@ID	int	4
@Type	varchar(50)	50
@Content	varchar(1000)	1000
@Cr_Id	int	4
@Correct_Answer	nchar	1
@Choice_A	varchar(700)	700
@Choice_B	varchar(700)	700
@Choice_C	varchar(700)	700
@Choice_D	varchar(700)	700
@Grade	float	8

SQL Script

```
create proc [dbo].[insetQuestion]
(
@ID INT,
@Type varchar(50),
@Content varchar(1000),
@Cr_Id int,
@Correct_Answer nchar(1),
@Choice_A varchar(700),
@Choice_B varchar(700),
@Choice_C varchar(700),
@Choice_D varchar(700),
@Grade float
)
 as
 begin try
  insert into Question
  Values(@ID,@Type,@Content,@Cr_Id,@Correct_Answer,@Choice_A,@Choice_B,@Choice_C,@Choice_-
D,@Grade)
  end try
  begin catch
        select 'QueryError, Nothing Has Executed'
  end catch
GO
```

Uses

[dbo].[Question]

[dbo].[ModelAnswer]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
CREATE procedure [dbo].[ModelAnswer] (@ExamId int)
as
    select q.id, q.correct_answer
    from question q, exam_questions eq
    where q.id = eq.q_id and eq.ex_id = @ExamId
GO
```

Uses

[dbo].[Exam_Questions]
[dbo].[Question]

🖹 [dbo].[Print_Exam_questions]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Exam_Id	bigint	8

SQL Script

```
CREATE PROC [dbo].[Print_Exam_questions]
@Exam_Id BIGINT
AS
BEGIN
    BEGIN TRY
        --DECLARE @Exam_Id INT = 5
        --SELECT * FROM Exam e  WHERE e.ID= @Exam_Id
        IF EXISTS(SELECT * FROM Exam e  WHERE e.ID= @Exam_Id)
            BEGIN

            SELECT c.name as [Course Name], e.date, e.start time, e.duration, NEXT VALUE FOR
Counter_Questions as [Counter Questions],Type, q.Content AS [Question Content] , q.[Choice
A],q.[Choice B],q.[Choice C],q.[Choice D]
                FROM Exam_Questions eq, Question q, exam e, course c
                where eq.ex id = @exam id and e.id = @exam id and eq.q id = q.id and c.id =
e.cr_id

            END
        ELSE
            BEGIN
                SELECT 'This exam does not exist' as Error
            END
    END TRY
    BEGIN CATCH
        SELECT 'An error occurred while processing the request' as Error
    END CATCH
END
GO
```

Uses

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

[dbo].[Question]

📄 [dbo].[selectAllDepartments]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```sql
create proc [dbo].[selectAllDepartments]
as
    begin try
        select Name as Department_Name,M_SSN as Department_M_SSN from Department
     end try
    begin catch
        select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

[dbo].[Department]

Project > (local)\ > User databases > Examination_System > Programmability > Stored Procedures > dbo.selectAll-Instructors

[dbo].[selectAllInstructors]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
create proc [dbo].[selectAllInstructors]
as
    begin try
        select
ins.SSN,ins.Fname,ins.Lname,ins.Age,ins.Gender,ins.City,ins.Street,ins.Building,ins.Phone,ins.Phone,dept.Name
        from Instructor as ins,Department as dept
        where ins.Dept_Id=dept.Dept_Id

     end try
    begin catch
        select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

[dbo].[Department]
[dbo].[Instructor]

[dbo].[selectAllStudents]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```sql
-------------------Select ALL STUDENTS-----------------------
create proc [dbo].[selectAllStudents]
as
    begin try
    select s.*
        from Student as s, Department as dept
        where s.SSN=dept.M_SSN
     end try
    begin catch
        select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

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

🗒 [dbo].[selectCourse]

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].[selectCourse] @ID int
as
 begin try
   select * from Course
   where ID=@ID
 end try
 begin catch
      select 'QueryError, Nothing Has Executed'
 end catch
GO
```

Uses

[dbo].[Course]

📄 [dbo].[selectDepartment]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
create proc [dbo].[selectDepartment] @Name varchar(50)
as
    begin try
        select * from Department where Name=@Name
     end try
    begin catch
        select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

[dbo].[Department]

📄 [dbo].[selectexam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
create proc [dbo].[selectexam] @ID int
 as
  begin try
    select * from Exam
    where ID=@ID
  end try
  begin catch
      select 'QueryError, Nothing Has Executed'
  end catch
GO
```

Uses

[dbo].[Exam]

📋 [dbo].[selectInstructor]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@SSNI	bigint	8

SQL Script

```
create proc [dbo].[selectInstructor] @SSNI bigint
as
    begin try
        select SSN,Fname,Lname,Age,Gender,City,Street,Building,Phone,Phone,dept.Name
        from Instructor as ins,Department as dept
        where SSN = @SSNI
        and  ins.Dept_Id =dept.Dept_Id
     end try
    begin catch
        select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

[dbo].[Department]
[dbo].[Instructor]

📄 [dbo].[SelectQuetions]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
create proc [dbo].[SelectQuetions] @Cr_Id int
as
begin try
select * from Question
where Cr_Id=@Cr_Id
end try
 begin catch
        select 'QueryError, Nothing Has Executed'
  end catch

GO
```

Uses

[dbo].[Question]

[dbo].[selectStudent]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@SSN	bigint	8

SQL Script

```
create proc [dbo].[selectStudent] @SSN bigint
as
    begin try
        select
s.SSN,s.Fname,s.Lname,s.Age,s.Faculty,s.Gender,s.City,s.Street,s.Building,s.Phone
        from Student as s,Department as dept
        where s.SSN=@SSN
        and s.SSN=dept.M_SSN
    end try
    begin catch
        select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

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

🖹 [dbo].[Student_Grades_in_each_course]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@St_Id	bigint	8

SQL Script

```sql
CREATE PROC [dbo].[Student_Grades_in_each_course]
@St_Id BIGINT
AS
BEGIN
    BEGIN TRY
    declare @Student_Name varchar(100)
        SELECT @Student_Name = (s.Fname + ' ' + s.Lname)
        FROM student s
        WHERE ssn = @St_Id
        IF @student_Name is not null
            BEGIN
                --DECLARE @St_Id BIGINT = 29802021200916
                SELECT @student_Name as [Student Name], c.Name as [Course Name], sc.Grade
                FROM Student_Courses sc INNER JOIN Course c
                ON   sc.Cr_Id = c.ID
                WHERE sc.Std_SSN = @St_Id
            END
        ELSE
            BEGIN
                SELECT 'This student does not exist' as Error
            END
    END TRY
    BEGIN CATCH
        SELECT 'An error occurred while processing the request' as Error
    END CATCH
END

    EXECUTE Student_Grades_in_each_course 29908091200777
GO
```

Uses

[dbo].[Course]

[dbo].[Student]
[dbo].[Student_Courses]

📖 [dbo].[Students_info_in_Department]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```sql
CREATE PROC [dbo].[Students_info_in_Department]
@Department_Id INT
AS
BEGIN
    BEGIN TRY
    declare @departmen_name varchar(50)
    SELECT @departmen_name = name
    FROM Department WHERE Dept_Id = @Department_Id
        IF @departmen_name is not null
            BEGIN
                IF EXISTS (
                    SELECT s.*
                    FROM Student s
                    WHERE s.Dept_ID = @Department_Id
                )
                    BEGIN
                        SELECT @departmen_name as [Department Name], s.*
                        FROM Student s
                        WHERE s.Dept_ID = @Department_Id
                    END
                ELSE
                    BEGIN
                        select'This department has no students' as Error
                    END
            ENd
        ELSE
            BEGIN
                SELECT 'The department id does not exist' as Error
            END
    END TRY
    BEGIN CATCH
        SELECT 'An error occurred while processing the request' as Error
    END CATCH
END
GO
```

Uses

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

📖 [dbo].[updateCourseName]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```sql
create proc [dbo].[updateCourseName] @ID int,@Name varchar(50)
as
begin try
    update Course
    set Name=@Name
    where ID=@ID
end try
 begin catch
      select 'QueryError, Nothing Has Executed'
  end catch

GO
```

Uses

[dbo].[Course]

📋 [dbo].[updateDepartmentName]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

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

SQL Script

```
create proc [dbo].[updateDepartmentName] (
@Dept_Id int, @Name varchar(50)
)
as
    begin try
        update Department
        set Name=@Name
        where Dept_Id=@Dept_Id
    end try
    begin catch
        select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

[dbo].[Department]

📇 [dbo].[updateEExamQu]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Q_ID	int	4
@Ex_ID	int	4

SQL Script

```
create proc [dbo].[updateEExamQu] @Q_ID int, @Ex_ID INT
as
    begin try
        update Exam_Questions
        set @Q_ID= Q_ID
        where Q_ID=(select ID from Question where ID=@Q_ID )
        and
        Ex_ID=(select ID from Exam where ID=@Ex_ID)
    end try
    begin catch
        select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

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

📄 [dbo].[updateExam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@ID	int	4
@Duration	float	8

SQL Script

```sql
create proc [dbo].[updateExam] @ID int,@Duration float
 as
 begin try
    update Exam
    set Duration=@Duration
    where ID=@ID
 end try
 begin catch
        select 'QueryError, Nothing Has Executed'
 end catch
GO
```

Uses

[dbo].[Exam]

🗒 [dbo].[updateInstructorCity]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@SSNI	bigint	8
@City	varchar(50)	50

SQL Script

```
create proc [dbo].[updateInstructorCity] @SSNI bigint,@City varchar(50)
as
    begin try
        update Instructor
        set City=@City
        where SSN=@SSNI
    end try
    begin catch
            select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

[dbo].[Instructor]

📖 [dbo].[updateInstructorCrs]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@In_SSN	bigint	8
@oldCrs	varchar(50)	50
@newCrs	varchar(50)	50

SQL Script

```sql
create proc [dbo].[updateInstructorCrs] @In_SSN bigint,@oldCrs varchar(50),@newCrs varchar(50)
as
    begin try
    update Instructor_Courses
    set Cr_Id=(select ID from Course where Name=@newCrs)
    where In_SSN=(select SSN from Instructor where In_SSN=@In_SSN)
    and
    Cr_Id=(select ID from Course where Name=@oldCrs)
    end try
    begin catch
        select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

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

[dbo].[updateQuetions]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Content	varchar(1000)	1000
@ID	int	4

SQL Script

```
create proc [dbo].[updateQuetions] @Content varchar(1000),@ID int
as
begin try
    update Question
    set Content=@Content
    where ID=@ID
end try
begin catch
        select 'QueryError, Nothing Has Executed'
  end catch

GO
```

Uses

[dbo].[Question]

🖼 [dbo].[updateStudentCrs]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Std_SSN	bigint	8
@newGrade	float	8
@Cr_Id	int	4

SQL Script

```
create proc [dbo].[updateStudentCrs] @Std_SSN BIGint,@newGrade float, @Cr_Id INT
as
    begin try
        update Student_Courses
        set @newGrade= Grade
        where Std_SSN=(select SSN from Student where SSN=@Std_SSN )
        and
        Cr_Id=(select ID from Course where ID=@Cr_Id)
    end try
    begin catch
        select 'QueryError, Nothing Has Executed'
    end catch
--===========DE
GO
```

Uses

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

🖿 [dbo].[updateStudentExam]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Std_SSN	bigint	8
@newStd_Anwser	varchar(50)	50
@Ex_ID	int	4

SQL Script

```
create proc [dbo].[updateStudentExam] @Std_SSN BIGint,@newStd_Anwser varchar(50), @Ex_ID INT
as
    begin try
        update Student_Exam
        set @newStd_Anwser= Std_Anwser
        where Std_SSN=(select SSN from Student where SSN=@Std_SSN )
        and
        Ex_ID=(select ID from Exam where ID=@Ex_ID)
    end try
    begin catch
        select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

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

📄 [dbo].[updateStudentFaculty]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@SSN	bigint	8
@Faculty	varchar(50)	50

SQL Script

```sql
create proc [dbo].[updateStudentFaculty] @SSN bigint,@Faculty varchar(50)
as
    begin try
        update Student
        set Faculty=@Faculty
        where SSN=@SSN
    end try
    begin catch
            select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

[dbo].[Student]

📄 [dbo].[viewExamQuestion]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
create proc [dbo].[viewExamQuestion]
as
    begin try
        select e.ID, e.Date, q.Content, q.Correct_Answer, eq.Q_ID
        from Exam as e,Question as q,Exam_Questions as eq
        where E.ID=eq.Ex_ID and Q.ID=EQ.Q_ID
     end try
    begin catch
        select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

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

📑 [dbo].[viewInstructorsCoursec]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
create proc [dbo].[viewInstructorsCoursec]
as
    begin try
        select Concat(ins.Fname,' ',ins.Lname) as Instructor_Name, cr.Name as Course_Name
        from Course as cr,Instructor as ins,Instructor_Courses as inscr
        where  cr.ID=inscr.Cr_Id and ins.SSN=inscr.In_SSN
     end try
    begin catch
        select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

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

📄 [dbo].[viewInstructorsCoursecs]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```sql
create proc [dbo].[viewInstructorsCoursecs]
as
    begin try
        select Concat(ins.Fname,' ',ins.Lname) as Instructor_Name, cr.Name as Course_Name
        from Course as cr,Instructor as ins,Instructor_Courses as inscr
        where  cr.ID=inscr.Cr_Id and ins.SSN=inscr.In_SSN
     end try
    begin catch
        select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

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

[dbo].[viewStudentsAnswer]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```sql
create proc [dbo].[viewStudentsAnswer]
as
    begin try
        select Concat(ST.Fname ,ST.Lname)as Student Name,E.ID as EXAMID,stdEX.Std Anwser as
Student_Answer
        from Student as st,Exam as E,Student_Exam as stdEX
        where st.SSN=stdEX.Std_SSN and e.ID=stdEX.Ex_ID
     end try
    begin catch
        select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

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

📋 [dbo].[viewStudentsCoursesGrades]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
create proc [dbo].[viewStudentsCoursesGrades]
as
    begin try
        select Concat(ST.Fname ,ST.Lname)as Student Name,cr.Name as Course Name,stdcr.Grade as
Student_grade
        from Student as st,Course as cr,Student_Courses as stdcr
        where st.SSN=stdcr.Std_SSN and cr.ID=stdcr.Cr_Id
     end try
    begin catch
        select 'QueryError, Nothing Has Executed'
    end catch
GO
```

Uses

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

⬒ *Sequences*

Objects

Name
dbo.mySeq

⊞ [dbo].[mySeq]

Properties

Property	Value
Owner	dbo

SQL Script

```sql
CREATE SEQUENCE [dbo].[mySeq]
AS int
START WITH 1
INCREMENT BY 1
MINVALUE 1
MAXVALUE 2147483647
NO CYCLE
CACHE
GO
```

Used By

[dbo].[Generate_Exam]

👤 *Users*

Objects

Name
dbo
guest

👤 dbo

Properties

Property	Value
Type	SqlUser
Login Name	sa
Default Schema	dbo

Database Level Permissions

Type	Action
CONNECT	Grant

SQL Script

```
GO
```

👤 guest

Properties

Property	Value
Type	SqlUser
Default Schema	guest

SQL Script

```
GO
```

👥 *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

Properties

Property	Value
Owner	dbo

👥 public

Properties

Property	Value
Owner	dbo