Database documentation

Server name:	.
Database name:	Examination System
Documentation date:	3/1/2022
Database size:	16.00 MB
Database description:	

Table summary

Name	Data size	Index size	Row size (*)	Row Count
dbo.Course	8 KB	8 KB	44	4
dbo.Department	8 KB	8 KB	44	3
dbo.Instructor	8 KB	8 KB	121	2
dbo.Instructor_teaches_Courses	8 KB	8 KB	15	3
dbo.Instructor_worksIn_Department	8 KB	8 KB	15	2
dbo.Question	8 KB	8 KB	928	46
dbo.Student	8 KB	8 KB	125	4
dbo.Student_takes_Courses	8 KB	8 KB	19	6
dbo.Students_answer_Questions	8 KB	8 KB	119	1
dbo.Topic	8 KB	8 KB	44	7
dbo.User	8 KB	24 KB	170	6

* Estimated row size. It assumes that all variable-length are 100% full, except those set to MAX.

11 table(s)

dbo.Course

I F P	Column name	Data type	Nulls	Default	Description
🔑 🔑	ID	int	NO		ID column contains integer datatype - is a primary key - cannot be null
	name	varchar (25)	NO		name field: contains the name of the course - contains varchar datatype - cannot be null -must be unique - must be less than 25 char
	duration	int	YES		duration field: contains integer value /number refers to count of hours in each course

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PK_Course	ID	ASC	Yes	CLUSTERED

Referencing tables

Table name	Foreign key	Primary key or unique constraint
dbo.Instructor_teaches_Courses	FK_Instructor_teaches_Courses_Course	PK_Course
dbo.Question	FK_Question_Course	PK_Course
dbo.Student_takes_Courses	FK_Student_takes_Courses_Course	PK_Course
dbo.Topic	FK_Topic_Course	PK_Course

Objects that depend on dbo.Course

dbo.addCourse (Stored procedure)

dbo.getAllCourses (Stored procedure)

dbo.getCourseById (Stored procedure)

dbo.getCrsesThatStudentTake (Stored procedure)

dbo.getInsCoursesByInsID (Stored procedure)

dbo.getInstructorCourses (Stored procedure)

dbo.removeCourse (Stored procedure)

dbo.updateCrsData (Stored procedure)

dbo.Department

I F P	Column name	Data type	Nulls	Default	Description
🔑IX 🔑	ID	int	NO		ID field : is the primary key of the Department table - contains integer values - must be unique and not null
	Name	varchar (25)	NO		Department name - must be unique - textual data - cannot be more than 25 characters
🔑FK	mgr_id	int	NO		is a foriegn key - contains integer data refers to ID column in Instructor table -cannot be null

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PK_Department	ID	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_Department_Instructor	mgr_id	PK_Instructor (dbo.Instructor)	

Referencing tables

Table name	Foreign key	Primary key or unique constraint
dbo.Instructor_worksIn_Department	FK_Instructor_worksIn_Department_Department	PK_Department
dbo.Student	FK_Student_Department	PK_Department

Objects that depend on dbo.Department

dbo.addDept (Stored procedure)

dbo.deleteDept (Stored procedure)

dbo.getAllDepts (Stored procedure)

dbo.getDeptByID (Stored procedure)

dbo.getDeptsInsWorksIn (Stored procedure)

dbo.updateDept (Stored procedure)

dbo.Instructor

I F P	Column name	Data type	Nulls	Default	Description
IX 🔑	ID	int	NO		ID: is the primary key for instructor table - it contains integer values
	firstname	varchar (50)	NO		firstname: column contains textual data represent Instructor first name - must be less than 50 characters - cannot be null
	lastname	varchar (50)	NO		lastname field: contains instructor's last name - cannot be null - cannot be more than 50 characters
FK	user_id	int	NO		user_id field : is a foreign key that refers to the ID column in User table

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PK_Instructor	ID	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_Instructor_User	user_id	PK_User (dbo.User)	

Referencing tables

Table name	Foreign key	Primary key or unique constraint
dbo.Department	FK_Department_Instructor	PK_Instructor
dbo.Instructor_teaches_Courses	FK_Instructor_teaches_Courses_Instructor	PK_Instructor
dbo.Instructor_worksIn_Department	FK_Instructor_worksIn_Department_Instructor	PK_Instructor

Objects that depend on dbo.Instructor

dbo.addIns (Stored procedure)
dbo.getAllInstructors (Stored procedure)
dbo.getDeptsInsWorksIn (Stored procedure)
dbo.getInsById (Stored procedure)
dbo.getInsCoursesByInsID (Stored procedure)
dbo.getInstructorCourses (Stored procedure)
dbo.removeIns (Stored procedure)
dbo.updateInsFname (Stored procedure)

dbo.Instructor_teaches_Courses

I F P	Column name	Data type	Nulls	Default	Description
IX FK 🔑	ins_id	int	NO		is a foreign key refers to ID column in Instructor table - contains integer values - cannot be null
IX FK 🔑	crs_id	int	NO		is a foreign key refers to ID column in Course table - contains integer values - cannot be null

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PK_Instructor_teaches_Courses	ins_id	ASC	Yes	CLUSTERED
PK_Instructor_teaches_Courses	crs_id	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_Instructor_teaches_Courses_Course	crs_id	PK_Course (dbo.Course)	
FK_Instructor_teaches_Courses_Instructor	ins_id	PK_Instructor (dbo.Instructor)	

Objects that depend on dbo.Instructor_teaches_Courses

dbo.assignNewCrsToInstructor (Stored procedure)

dbo.changeCrsThatInsTeach (Stored procedure)

dbo.getInsCoursesByInsID (Stored procedure)

dbo.getInstructorCourses (Stored procedure)

dbo.unassignCourseToIns (Stored procedure)

dbo.Instructor_worksIn_Department

I F P	Column name	Data type	Nulls	Default	Description
IX FK 🔑	ins_id	int	NO		is a foreign key refers to ID column in Instructor table - integer values - cannot be null
IX FK 🔑	dept_id	int	NO		is a foreign key refers to ID column in Department table - integer values - cannot be null

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PK_Instructor_worksIn_Department	ins_id	ASC	Yes	CLUSTERED
PK_Instructor_worksIn_Department	dept_id	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_Instructor_worksIn_Department_Department	dept_id	PK_Department (dbo.Department)	
FK_Instructor_worksIn_Department_Instructor	ins_id	PK_Instructor (dbo.Instructor)	

Objects that depend on dbo.Instructor_worksIn_Department

dbo.assignInsToDept (Stored procedure)

dbo.chngeInsDept (Stored procedure)

dbo.getDeptsInsWorksIn (Stored procedure)

dbo.rmoveInsFromDept (Stored procedure)

dbo.Question

I	F	P	Column name	Data type	Nulls	Default	Description
IX		🔑	ID	int	NO		Primary key - contains integer data - must be unique and not null
			body	varchar (500)	NO		This column contains the body of the question - cannot be null
			correct_answer	varchar (100)	NO		This column contains one of the choices as well as the correct choice - cannot be null- contains textual data
			a2	varchar (100)	NO		contains the second choice / answer to the question - cannot be null - contains textual data
			a3	varchar (100)	YES		Contains the third choice / answer - can be null in case of (true or false) question
			a4	varchar (100)	YES		Contains the fourth choice / answer - can be null in case of (true or false) question
			type_of_question	bit	NO	((0))	Contains type of question - True for [True or False] and False for [Multiple Choice]
	FK		crs_id	int	NO		Is a foriegn key refers to ID column in Course table - contains integer value - cannot be null

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PK_Question	ID	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_Question_Course	crs_id	PK_Course (dbo.Course)	

Referencing tables

Table name	Foreign key	Primary key or unique constraint
dbo.Students_answer_Questions	FK_Students_answer_Questions_Question	PK_Question

Objects that depend on dbo.Question

dbo.addQuest (Stored procedure)

dbo.deleteQuest (Stored procedure)

dbo.getAllAboutQuestions (Stored procedure)

dbo.getAllQuestions (Stored procedure)

dbo.getQuestById (Stored procedure)

dbo.getStdsAnsToSpecQuest (Stored procedure)

dbo.updateQuest (Stored procedure)

dbo.Student

I F P	Column name	Data type	Nulls	Default	Description
IX 🔑	ID	int	NO		ID field : is the primary key of the table - contains integer unique values - cannot be null
	firstname	varchar (50)	NO		firstname field: contains student firstname, must be varchar cannot be more than 50 characters - cannot be null
	lastname	varchar (50)	NO		lastname field: contains student lastname, must be varchar cannot be more than 50 characters - cannot be null
FK	user_id	int	NO		user-id field: is a foreign key refers to ID column in User table - cannot be null - contains integer values
FK	dept_id	int	NO		dept-id field: is a foreign key refers to ID column in Department table - cannot be null - contains integer values

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PK_Student	ID	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_Student_Department	dept_id	PK_Department (dbo.Department)	
FK_Student_User	user_id	PK_User (dbo.User)	

Referencing tables

Table name	Foreign key	Primary key or unique constraint
dbo.Student_takes_Courses	FK_Student_takes_Courses_Student	PK_Student
dbo.Students_answer_Questions	FK_Students_answer_Questions_Student	PK_Student

Objects that depend on dbo.Student

dbo.addStd (Stored procedure)

dbo.deleteStd (Stored procedure)

dbo.getAllStudents (Stored procedure)

dbo.getCrsesThatStudentTake (Stored procedure)

dbo.getStdsAnsToSpecQuest (Stored procedure)

dbo.getStudentByID (Stored procedure)

dbo.updateStdFname (Stored procedure)

dbo.Student_takes_Courses

I F P	Column name	Data type	Nulls	Default	Description
IX FK 🔑	student_id	int	NO		foreign key refers to ID column in Student table
IX FK 🔑	crs_id	int	NO		foreign key refers to ID column in Course table
	grade	int	NO	((0))	grade field : contains student grade which is an integer value represent student grade at that course - default value is equal to 0 - cannot be null

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PK_Student_takes_Courses	student_id	ASC	Yes	CLUSTERED
PK_Student_takes_Courses	crs_id	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_Student_takes_Courses_Course	crs_id	PK_Course (dbo.Course)	
FK_Student_takes_Courses_Student	student_id	PK_Student (dbo.Student)	

Objects that depend on dbo.Student_takes_Courses

dbo.addNewCrsToStd (Stored procedure)

dbo.chngCrsStdTake (Stored procedure)

dbo.getCrsesThatStudentTake (Stored procedure)

dbo.removeFromCrsStdTake (Stored procedure)

dbo.Students_answer_Questions

I F P	Column name	Data type	Nulls	Default	Description
FK	Std_ID	int	NO		Std_ID: is a foreign key refers to ID of the student- takes integer values - doesn't allow null
FK	Question_ID	int	NO		Contains a foreign key that refers to ID column in Question table - integer values - doesn't allow null
	Student_Answer	varchar (100)	YES		Contains varchar datatype - consist of the student answer for a particular question

Foreign keys

Constraint name	Column name	Reference	Description
FK_Students_answer_Questions_Question	Question_ID	PK_Question (dbo.Question)	
FK_Students_answer_Questions_Student	Std_ID	PK_Student (dbo.Student)	

Objects that depend on dbo.Students_answer_Questions

dbo.addNewQuestionAnswer (Stored procedure)

dbo.getStdsAnsToSpecQuest (Stored procedure)

dbo.removeStdAnswer (Stored procedure)

dbo.updateStdAnswer (Stored procedure)

dbo.Topic

I F P	Column name	Data type	Nulls	Default	Description
IX 🔑	ID	int	NO		The primary key of the table - contains integer values - cannot be null
	name	varchar (25)	NO		The topic name - textual data - cannot be null
FK	crs_id	int	NO		is a foriegn key refers to ID column in Course table - Each Course contains number of topics

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PK_Topic	ID	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_Topic_Course	crs_id	PK_Course (dbo.Course)	

Objects that depend on dbo.Topic

dbo.addTopic (Stored procedure)

dbo.getAllTopics (Stored procedure)

dbo.getTopicByID (Stored procedure)

dbo.removeTopic (Stored procedure)

dbo.updateTopicName (Stored procedure)

dbo.User

I F P	Column name	Data type	Nulls	Default	Description
🔑 🔑	ID	int	NO		ID field: the primary key of the table contains integer datatype - cannot be null
	username	varchar (50)	NO		username field: contains varchar data type (textual data) - cannot be more than 50 chars - cannot be null
	email	varchar (50)	NO		email field: contains the user email - is unique - varchar datatype - cannot be null
	password	varchar (50)	NO		password field: contains user password - varchar datatype - cannot be more than 50 characters
	user_type	bit	NO		user_type: contains bit value represent two types of the user if 0 then it's instructor if 1 then it's student

Indexes

Index name	Column name	Sort direction	Is unique	Index type
email	ID	ASC	Yes	NONCLUSTERED
PK_User	ID	ASC	Yes	CLUSTERED

Referencing tables

Table name	Foreign key	Primary key or unique constraint
dbo.Instructor	FK_Instructor_User	PK_User
dbo.Student	FK_Student_User	PK_User

Objects that depend on dbo.User

dbo.createUser (Stored procedure)
dbo.deleteUser (Stored procedure)
dbo.getUserByID (Stored procedure)
dbo.getUsers (Stored procedure)
dbo.updateUserName (Stored procedure)

dbo.addCourse

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@id	int	IN	
@crsname	varchar	IN	
@crsduration	int	IN	

```
create proc addCourse @id int, @crsname varchar(25), @crsduration int
as
if not exists(select * from Course where ID = @id)
insert into Course values(@id, @crsname, @crsduration)
else
select 'duplicate id, cannot insert'as 'message'
```

Related objects

dbo.Course(TABLE)

dbo.addDept

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@id	int	IN	
@deptname	varchar	IN	
@mgr_id	int	IN	

```
create proc addDept @id int, @deptname varchar(25),@mgr_id int
as
if not exists(select * from Department where ID = @id)
insert into Department values(@id, @deptname, @mgr_id)
else
select 'Duplicate ID cannot insert'as 'message'
```

Related objects

dbo.Department(TABLE)

dbo.addIns

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@id	int	IN	
@fname	varchar	IN	
@lname	varchar	IN	
@userid	int	IN	

```
create proc addIns @id int, @fname varchar(50), @lname varchar(50), @userid int
as
if not exists (select * from [Instructor]
where ID = @id)
insert into [Instructor] values (@id, @fname, @lname, @userid)
else
select 'duplicate Id'
```

Related objects

dbo.Instructor(TABLE)

dbo.addNewCrsToStd

Created: 01 Mar 2022 Last updated: 01 Mar 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@stdid	int	IN	
@crsid	int	IN	

```
create proc addNewCrsToStd @stdid int, @crsid int
as
begin try
insert into Student_takes_Courses values (@stdid,@crsid,0)
end try
begin catch
select 'cannot insert invalid stdid or crsid'as 'message'
end catch
```

Related objects

dbo.Student_takes_Courses(TABLE)

dbo.addNewQuestionAnswer

Created: 01 Mar 2022 Last updated: 01 Mar 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@stdid	int	IN	
@questid	int	IN	
@stdAns	varchar	IN	

```
create proc addNewQuestionAnswer @stdid int, @questid int, @stdAns varchar(100)
as
begin try
insert into Students_answer_Questions values (@stdid,@questid,@stdAns)
end try
begin catch
select 'cannot insert invalid stdid or questionid'as 'message'
end catch
```

Related objects

dbo.Students_answer_Questions(TABLE)

dbo.addQuest

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@id	int	IN	
@questBody	varchar	IN	
@correctans	varchar	IN	
@anstwo	varchar	IN	
@ansthree	varchar	IN	
@ansfour	varchar	IN	
@questtype	bit	IN	
@crsid	int	IN	

```
create proc addQuest @id int, @questBody varchar(500), @correctans varchar(100), @anstwo varchar(100),
@ansthree varchar(100), @ansfour varchar(100), @questtype bit, @crsid int
as
if not exists (select * from Question where ID = @id)
insert into Question values(@id, @questBody, @correctans,@anstwo,@ansthree,@ansfour,@questtype,@crsid)
else
select 'Sorry, Duplicate Id'as 'message'
```

Related objects

dbo.Question(TABLE)

dbo.addStd

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@id	int	IN	
@fname	varchar	IN	
@lname	varchar	IN	
@userid	int	IN	
@deptid	int	IN	

```
create proc addStd @id int,@fname varchar(20),@lname varchar(20),@userid int,@deptid int
as
if not exists(select * from [Student] where ID = @id)
insert into Student values(@id,@fname,@lname,@userid,@deptid)
else
select 'Sorry Duplicate Id'as 'message'
```

Related objects

dbo.Student(TABLE)

dbo.addTopic

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@id	int	IN	
@topicname	varchar	IN	
@crsid	int	IN	

```
create proc addTopic @id int, @topicname varchar(25), @crsid int
as
if not exists ( select * from Topic where ID = @id )
insert into Topic values (@id, @topicname, @crsid)
else
select 'duplicate id cannot insert'
```

Related objects

dbo.Topic(TABLE)

dbo.assignInsToDept

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@deptId	int	IN	
@insid	int	IN	

```
create proc assignInsToDept @deptId int, @insid int
as
begin try
insert into Instructor_worksIn_Department values (@insid,@deptId)
end try
begin catch
select 'invalid crs id or ins id cannot insert'as 'message'
end catch
```

Related objects

dbo.Instructor_worksIn_Department(TABLE)

dbo.assignNewCrsToInstructor

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@insId	int	IN	
@crsId	int	IN	

```
create proc assignNewCrsToInstructor @insId int, @crsId int
as
begin try
insert into Instructor_teaches_Courses values (@insId, @crsId)
end try
begin catch
select 'instructor id or course id is not right'
end catch
```

Related objects

dbo.Instructor_teaches_Courses(TABLE)

dbo.changeCrsThatInsTeach

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@id	int	IN	
@newcrsid	int	IN	

```
create proc changeCrsThatInsTeach @id int, @newcrsid int
as
begin try
update Instructor_teaches_Courses set crs_id = @newcrsid
where ins_id = @id
end try
begin catch
select 'The instructor id or course id is invalid'as 'message'
end catch
```

Related objects

dbo.Instructor_teaches_Courses(TABLE)

dbo.chngCrsStdTake

Created: 01 Mar 2022 Last updated: 01 Mar 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@stdid	int	IN	
@newCrsid	int	IN	

```
create proc chngCrsStdTake @stdid int, @newCrsid int
as
if exists(select * from Student_takes_Courses where student_id = @stdid)
update Student_takes_Courses set crs_id = @newCrsid
where student_id = @stdid
else
select 'invalid student id'as 'message'
```

Related objects

dbo.Student_takes_Courses(TABLE)

dbo.chngeInsDept

Created: 01 Mar 2022 Last updated: 01 Mar 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@insid	int	IN	
@newdeptid	int	IN	

```
CREATE proc chngeInsDept @insid int, @newdeptid int
as
if exists(select * from Instructor_worksIn_Department where ins_id = @insid)
update Instructor_worksIn_Department set dept_id = @newdeptid
where ins_id = @insid
else
select 'invalid ins id or dept id'as 'message'
```

Related objects

dbo.Instructor_worksIn_Department(TABLE)

dbo.createUser

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@Id	int	IN	
@username	varchar	IN	
@email	varchar	IN	
@pass	varchar	IN	
@utype	bit	IN	

```
create proc createUser @Id int, @username varchar(50), @email varchar(50), @pass varchar(50), @utype bit
as
insert into [User] values(@Id, @username, @email, @pass, @utype)
```

Related objects

dbo.User(TABLE)

dbo.deleteDept

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@id	int	IN	

```
create proc deleteDept @id int
as
if exists(select * from Department where ID = @id)
delete from Department where ID = @id
else
select 'sorry, no such ID 'as 'message'
```

Related objects

dbo.Department(TABLE)

dbo.deleteQuest

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@id	int	IN	

```
create proc deleteQuest @id int
as
if exists(select * from Question where ID = @id)
delete from Question where ID = @id
else
select 'Sorry this id does not exist here 'as 'message'
```

Related objects

dbo.Question(TABLE)

dbo.deleteStd

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@id	int	IN	

```
create proc deleteStd @id int
as
if exists(select * from [Student] where ID = @id)
delete from [Student] where ID = @id
else
select 'Sorry this id does not exist here'as 'message'
```

Related objects

dbo.Student(TABLE)

dbo.deleteUser

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@Id	int	IN	

```
create proc deleteUser @Id int
as
delete from [User] where ID = @Id
```

Related objects

dbo.User(TABLE)

dbo.getAllAboutQuestions

Created: 28 Feb 2022 Last updated: 28 Feb 2022

```
create proc getAllAboutQuestions
as
select * from Question
```

Related objects

dbo.Question(TABLE)

dbo.getAllCourses

Created: 28 Feb 2022 Last updated: 28 Feb 2022

```
create proc getAllCourses
as
select * from Course
```

Related objects

dbo.Course(TABLE)

dbo.getAllDepts

Created: 28 Feb 2022 Last updated: 28 Feb 2022

```
create proc getAllDepts
as
select * from Department
```

Related objects

dbo.Department(TABLE)

dbo.getAllInstructors

Created: 28 Feb 2022 Last updated: 28 Feb 2022

```
create proc getAllInstructors
as
select * from [Instructor]
```

Related objects

dbo.Instructor(TABLE)

dbo.getAllQuestions

Created: 28 Feb 2022 Last updated: 28 Feb 2022

```
create proc getAllQuestions
as
select body,correct_answer,a2,a3,a4,type_of_question,crs_id from Question
```

Related objects

dbo.Question(TABLE)

dbo.getAllStudents

Created: 28 Feb 2022 Last updated: 28 Feb 2022

```
create proc getAllStudents
as
select * from [Student]
```

Related objects

dbo.Student(TABLE)

dbo.getAllTopics

Created: 28 Feb 2022 Last updated: 28 Feb 2022

```
create proc getAllTopics
as
select * from Topic
```

Related objects

dbo.Topic(TABLE)

dbo.getCourseById

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@id	int	IN	

```sql
create proc getCourseById @id int
as
if exists (select * from Course where ID = @id)
select * from Course where ID = @id
else
select 'Sorry! this id does not exist'as 'message'
```

Related objects

dbo.Course(TABLE)

dbo.getCrsesThatStudentTake

Created: 01 Mar 2022 Last updated: 01 Mar 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@stdid	int	IN	

```
create proc getCrsesThatStudentTake @stdid int
as
if exists ( select * from Student_takes_Courses where Student_takes_Courses.student_id = @stdid)
select concat(firstname,' ', lastname) as [Student Name] , name, grade
from Student inner join Student_takes_Courses
on ID = Student_takes_Courses.student_id
inner join Course
on Student_takes_Courses.crs_id = Course.ID
where Student.ID = @stdid
else
select 'no such student id 'as 'message'
```

Related objects

dbo.Student(TABLE)

dbo.Course(TABLE)

dbo.Student_takes_Courses(TABLE)

dbo.getDeptByID

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@id	int	IN	

```
create proc getDeptByID @id int
as
if exists ( select * from Department where ID = @id )
select * from Department
where ID = @id
else
select 'This id does not exist'as 'message'
```

Related objects

dbo.Department(TABLE)

dbo.getDeptsInsWorksIn

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@id	int	IN	

```
create proc getDeptsInsWorksIn @id int
as
if exists (select * from Instructor_worksIn_Department where ins_id = @id)
select concat(Instructor.firstname, '  ',Instructor.lastname) as [Instructor name], Department.Name
from Department inner join Instructor_worksIn_Department
on Department.ID = Instructor_worksIn_Department.dept_id
inner join Instructor
on Instructor.ID = Instructor_worksIn_Department.ins_id
where Instructor_worksIn_Department.ins_id = @id
else
select 'ins id or crs id is invalid'as 'message'
```

Related objects

dbo.Instructor(TABLE)

dbo.Department(TABLE)

dbo.Instructor_worksIn_Department(TABLE)

dbo.getInsById

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@id	int	IN	

```
CREATE proc getInsById @id int
as
if exists(select * from [Instructor]
where ID = @id)
select * from [Instructor]
where ID = @id
else
select 'this id doesnot exist'
```

Related objects

dbo.Instructor(TABLE)

dbo.getInsCoursesByInsID

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@id	int	IN	

```
create proc getInsCoursesByInsID @id int
as
if exists ( select * from Instructor_teaches_Courses where ins_id = @id)
select Course.name as [Course name]  , concat(Instructor.firstname,' ',Instructor.lastname) as
[Instructor Name]
from Course inner join Instructor_teaches_Courses
on Course.ID = Instructor_teaches_Courses.crs_id
inner join Instructor
on Instructor.ID = Instructor_teaches_Courses.ins_id
where ins_id = @id
```

Related objects

dbo.Instructor(TABLE)

dbo.Course(TABLE)

dbo.Instructor_teaches_Courses(TABLE)

dbo.getInstructorCourses

Created: 28 Feb 2022 Last updated: 28 Feb 2022

```
create proc getInstructorCourses
as
select Course.name  , concat(Instructor.firstname,' ',Instructor.lastname) as [Instructor Name]
from Course inner join Instructor_teaches_Courses
on Course.ID = Instructor_teaches_Courses.crs_id
inner join Instructor
on Instructor.ID = Instructor_teaches_Courses.ins_id
```

Related objects

dbo.Instructor(TABLE)

dbo.Course(TABLE)

dbo.Instructor_teaches_Courses(TABLE)

dbo.getQuestById

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@id	int	IN	

```
create proc getQuestById @id int
as
if exists (select * from Question where ID = @id)
select * from Question
where ID = @id
else
select 'Sorry this Id does not exist here'
```

Related objects

dbo.Question(TABLE)

dbo.getStdsAnsToSpecQuest

Created: 01 Mar 2022 Last updated: 01 Mar 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@stdid	int	IN	

```
create proc getStdsAnsToSpecQuest @stdid int
as
if exists(select * from Students_answer_Questions where Std_ID = @stdid)
select concat(firstname,' ',lastname) as [Student Name],body as [question body],Student_Answer
from Student inner join Students_answer_Questions
on ID = Std_ID
inner join Question
on Question_ID = Question.ID
where Std_ID = @stdid
else
select 'invalid student id'as 'message'
```

Related objects

dbo.Question(TABLE)

dbo.Students_answer_Questions(TABLE)

dbo.Student(TABLE)

dbo.getStudentByID

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@id	int	IN	

```sql
create proc getStudentByID @id int
as
if exists(select * from Student where ID = @id)
select * from [Student]
where ID = @id
else
select 'This ID is not here'
```

Related objects

dbo.Student(TABLE)

dbo.getTopicByID

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@id	int	IN	

```
create proc getTopicByID @id int
as
if exists( select * from Topic where ID = @id)
select * from Topic where ID = @id
else
select 'ID does not exist'as 'message'
```

Related objects

dbo.Topic(TABLE)

dbo.getUserByID

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@Id	int	IN	

```
create proc getUserByID @Id int
as
select * from [User]
where ID = @Id
```

Related objects

dbo.User(TABLE)

dbo.getUsers

Created: 28 Feb 2022 Last updated: 28 Feb 2022

```
create proc getUsers
as
select * from [User]
```

Related objects

dbo.User(TABLE)

dbo.removeCourse

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@id	int	IN	

```
create proc removeCourse @id int
as
if exists (select * from Course where ID = @id)
delete from Course where ID = @id
else
select 'cannot delete! This ID does not exist'as 'message'
```

Related objects

dbo.Course(TABLE)

dbo.removeFromCrsStdTake

Created: 01 Mar 2022 Last updated: 01 Mar 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@stdid	int	IN	
@crsToBeRemovedId	int	IN	

```
create proc removeFromCrsStdTake @stdid int, @crsToBeRemovedId int
as
if exists( select * from Student_takes_Courses where student_id = @stdid and crs_id = @crsToBeRemovedId)
delete from Student_takes_Courses where student_id = @stdid and crs_id = @crsToBeRemovedId
else
select 'invalid stdid or crsid'as 'message'
```

Related objects

dbo.Student_takes_Courses(TABLE)

dbo.removeIns

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@id	int	IN	

```
create proc removeIns @id int
as
if exists (select * from [Instructor]
where ID = @id)
delete from [Instructor] where ID = @id
else
select 'this Id doesnot exist in this table'
```

Related objects

dbo.Instructor(TABLE)

dbo.removeStdAnswer

Created: 01 Mar 2022 Last updated: 01 Mar 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@stdid	int	IN	
@questid	int	IN	

```
create proc removeStdAnswer @stdid int, @questid int
as
if exists(select * from Students_answer_Questions where Std_ID = @stdid and Question_ID = @questid)
delete from Students_answer_Questions where Std_ID = @stdid and Question_ID = @questid
else
select 'invalid std id or question id'as 'message'
```

Related objects

dbo.Students_answer_Questions(TABLE)

dbo.removeTopic

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@id	int	IN	

```
create proc removeTopic @id int
as
if exists ( select * from Topic where ID = @id )
delete from Topic where ID = @id
else
select 'Sorry! this id does not exist 'as 'message'
```

Related objects

dbo.Topic(TABLE)

dbo.rmoveInsFromDept

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@insid	int	IN	
@deptid	int	IN	

```
create proc rmoveInsFromDept @insid int, @deptid int
as
if exists(select * from Instructor_worksIn_Department where dept_id = @deptid)
delete from Instructor_worksIn_Department
where dept_id = @deptid and ins_id = @insid
else
select 'no such dept id or ins id'as 'message'
```

Related objects

dbo.Instructor_worksIn_Department(TABLE)

dbo.unassignCourseToIns

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@id	int	IN	
@crsid	int	IN	

```
CREATE proc unassignCourseToIns @id int, @crsid int
as
if exists(select * from Instructor_teaches_Courses where ins_id = @id and crs_id = @crsid)
delete from Instructor_teaches_Courses where ins_id = @id and crs_id = @crsid
else
select 'instructor id or course id is not valid'
```

Related objects

dbo.Instructor_teaches_Courses(TABLE)

dbo.updateCrsData

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@id	int	IN	
@Crsname	varchar	IN	
@crsduration	int	IN	

```
create proc updateCrsData @id int, @Crsname varchar(25), @crsduration int = 20
as
if exists (select * from Course where ID = @id)
update Course set name = @Crsname, duration = @crsduration
where ID = @id
else
select 'This id doesnot exist'as 'message'
```

Related objects

dbo.Course(TABLE)

dbo.updateDept

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@id	int	IN	
@name	varchar	IN	
@mgrid	int	IN	

```
create proc updateDept @id int, @name varchar(25), @mgrid int
as
if exists(select * from Department where ID = @id)
update Department set Name = @name, mgr_id = @mgrid
where ID = @id
else
select 'Sorry this id does not exist'as 'message'
```

Related objects

dbo.Department(TABLE)

dbo.updateInsFname

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@id	int	IN	
@fname	varchar	IN	

```
create proc updateInsFname @id int, @fname varchar(25)
as
begin try
update [Instructor] set firstname = @fname
where ID = @id
end try
begin catch
select 'This id does not exist here'
end catch
```

Related objects

dbo.Instructor(TABLE)

dbo.updateQuest

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@id	int	IN	
@QuestBody	varchar	IN	
@ansoone	varchar	IN	
@anstwo	varchar	IN	
@ansthree	varchar	IN	
@ansfour	varchar	IN	
@questtype	bit	IN	
@crsid	int	IN	

```
create proc updateQuest @id int , @QuestBody varchar(500), @ansoone varchar(100), @anstwo varchar(100),
@ansthree varchar(100),@ansfour varchar(100), @questtype bit ,@crsid int = 3
as
if exists (select * from Question where ID = @id)
update Question set body = @QuestBody, correct_answer = @ansoone, a2 = @anstwo, a3 = @ansthree, a4 =
@ansfour, type_of_question = @questtype
where ID = @id
else
select 'Sorry this Id does not exist here'as 'message'
```

Related objects

dbo.Question(TABLE)

dbo.updateStdAnswer

Created: 01 Mar 2022 Last updated: 01 Mar 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@stdid	int	IN	
@questid	int	IN	
@newAnswer	varchar	IN	

```
create proc updateStdAnswer @stdid int, @questid int , @newAnswer varchar(100)
as
if exists(select * from Students_answer_Questions where Std_ID = @stdid and Question_ID = @questid)
update Students_answer_Questions set Student_Answer = @newAnswer
where Std_ID = @stdid and Question_ID = @questid
else
select 'invalid student id or question id'as 'message'
```

Related objects

dbo.Students_answer_Questions(TABLE)

dbo.updateStdFname

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@id	int	IN	
@fname	varchar	IN	

```
create proc updateStdFname @id int, @fname varchar(25)
as
if exists(select * from [Student] where ID = @id)
update [Student] set firstname = @fname
where ID = @id
else
select 'Sorry this id does not exist here'as 'message'
```

Related objects

dbo.Student(TABLE)

dbo.updateTopicName

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@id	int	IN	
@topname	varchar	IN	

```sql
create proc updateTopicName @id int, @topname varchar(25)
as
if exists (select * from Topic where ID = @id)
update Topic set name = @topname
where ID = @id
else
select 'Sorry! This id does not exist'as 'message'
```

Related objects

dbo.Topic(TABLE)

dbo.updateUserName

Created: 28 Feb 2022 Last updated: 28 Feb 2022

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@email	varchar	IN	
@newusername	varchar	IN	

```
CREATE proc updateUserName @email varchar(50), @newusername varchar(50)
as
begin try
update [User]
set username = @newusername
where email = @email
end try
begin catch
select 'email doesnot exist'
end catch
```

Related objects

dbo.User(TABLE)

sys.sys

Created: 13 Apr 2009 Last updated: 22 Aug 2017

Components

COMPONENT NAME	TYPE	DERIVATION	NAMESPACE
anySimpleType	ANY_SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
anyType	ANY_TYPE	NONE	http://www.w3.org/2001/XMLSchema
base	ATTRIBUTE	NONE	http://www.w3.org/XML/1998/namespace
base64Encoded	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/sqltypes
clrTypeName	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/sqltypes
lang	ATTRIBUTE	NONE	http://www.w3.org/XML/1998/namespace
localeId	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/sqltypes
maxLength	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/sqltypes
precision	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/sqltypes
scale	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/sqltypes
space	ATTRIBUTE	NONE	http://www.w3.org/XML/1998/namespace
sqlCollationVersion	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/sqltypes
sqlCompareOptions	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/sqltypes
sqlDbType	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/sqltypes
sqlSortId	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/sqltypes
sqlTypeAlias	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/sqltypes
supplementaryCharacters	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/sqltypes
xmlSchemaCollection	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/sqltypes
xml	COMPLEX_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
ENTITIES	LIST_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
IDREFS	LIST_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
NMTOKENS	LIST_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
sqlCompareOptionsList	LIST_TYPE	NONE	http://schemas.microsoft.com/sqlserver/2004/sqltypes
NOTATION	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
QName	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
anyURI	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
base64Binary	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
boolean	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
date	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
dateTime	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
decimal	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
double	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
duration	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
float	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
gDay	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
gMonth	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
gMonthDay	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
gYear	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
gYearMonth	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
hexBinary	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
string	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
time	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
ENTITY	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema

ID	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
IDREF	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
NCName	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
NMTOKEN	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
Name	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
bigint	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
binary	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
bit	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
byte	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
char	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
date	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
datetime	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
datetime2	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
datetimeoffset	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
dbobject	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
decimal	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
float	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
hierarchyid	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
image	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
int	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
int	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
integer	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
language	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
long	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
money	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
nchar	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
negativeInteger	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
nonNegativeInteger	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
nonPositiveInteger	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
normalizedString	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
ntext	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
numeric	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
nvarchar	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
positiveInteger	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
real	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
short	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
smalldatetime	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
smallint	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
smallmoney	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
sqlCompareOptionsEnum	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
sqlDbTypeEnum	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
text	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
time	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
timestamp	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
timestampNumeric	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
tinyint	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
token	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
uniqueidentifier	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
unsignedByte	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
unsignedInt	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
unsignedLong	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
unsignedShort	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema

varbinary	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
varchar	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
xmlSpaceEnum	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/XML/1998/namespace

Microsoft.SqlServer.Types

Created: 22 Aug 2017 Last updated: 22 Aug 2017

CLR NAME:	microsoft.sqlserver.types, version=14.0.0.0, culture=neutral, publickeytoken=89845dcd8080cc91, processorarchitecture=msil
PERMISSION:	UNSAFE_ACCESS
DESCRIPTION:	

User name	Login name	Created	Roles	User description
dbo	DESKTOP-F7CTSK1\Computec	4/8/2003	db_owner	

Role name	App role	Users with this role	Role description
db_accessadmin			
db_backupoperator			
db_datareader			
db_datawriter			
db_ddladmin			
db_denydatareader			
db_denydatawriter			
db_owner		dbo	
db_securityadmin			
public			