

# **Examination System Database**

## Servers

Servers 1

DESKTOP-E6VB8AG

## DESKTOP-E6VB8AG

## User databases

# examination\_sys

## Properties

Name	Value
Creation date	6/5/2022 12:27:13 AM

## Object Types 9

- Tables
- Views
- Stored Procedures
- Table-Valued Functions
- User-Defined Table Types
- Rules
- Users
- Database Roles
- Schemas

## Database Files 5

Name	Type	Size	Max size	Autogrowth	File Name
exam_sys_Primary	Rows	10,240 KB	N/A	10 Percent	C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\exam_sys_Primary.mdf
exam_sys_Log	Log	5,120 KB	2,147,483,648 KB	10 Percent	C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\exam_sys_Log.ldf
group_one_FG	Rows	5,120 KB	N/A	5 Percent	C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\group_one_FG.ndf
group_two_FG	Rows	5,120 KB	N/A	5 Percent	C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\group_two_FG.ndf
group_three_FG	Rows	5,120 KB	N/A	5 Percent	C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\group_three_FG.ndf

## Tables

### Objects 24

Name	Description
audit_admin.audit1_class	Table To Audit by Admin User any Change on Class Table
audit_admin.audit2_class_instruct	Table To Audit by Admin User any Change on Class_Instructor Table
audit_admin.audit3_course	Table To Audit by Admin User any Change on CourseTable
audit_admin.audit4_course_student	Table To Audit by Admin User any Change on Course_Student Table
audit_admin.audit4_exam	Table To Audit by Admin User any Change on ExamTable
audit_admin.audit4_instructor	Table To Audit by Admin User any Change on InstructorTable
audit_admin.audit5_instructor_student_exam	Table To Audit by Admin User any Change on Instructor_Student_Exam Table
audit_admin.audit6_multi_choice	Table To Audit by Admin User any Change on Multi_choice Table
audit_admin.audit6_question_exam	Table To Audit by Admin User any Change on Questio_Exam Table
audit_admin.audit6_question_pool	Table To Audit by Admin User any Change on Question_Pool Table
audit_admin.audit7_student	Table To Audit by Admin User any Change on Student Table
audit_admin.audit7_student_question_exam	Table To Audit by Admin User any Change on Student_Question_Exam Table
dbo.class	Class table Contain Class_Id and Class_Name (Class_ID Primary Key )
dbo.class_instructor_course	Class_Instructor_Name Table Relased For Ternary Relationship Between Class , instructor And Course (Class_id ,Course_id , Instructor_id , Year Primary key)
dbo.course	Course Table Contain Course_name , Course_description , Course_MaxDegree , Course_MinDegree (Course_id Primary key)
dbo.course_student	Course_student Table relased from Relationship between student and Courses He Studied and Final Result in every course he studied (Course_ID and Student_id as Primary Key)
dbo.exam	Exam Table Contain Start_time , End_time , type (Corrective , Exam) , Exam_Date and Exam_Total_degree (Exam_id Primary key) and (Course_id , Instructor_id as Foreign Key)
dbo.instructor	Instructor Table Contain (Name , Birthdate , Phone , City and Street ) and (Instructor_ID primary key)
dbo.instructor_student_exam	Instructor_Student_Exam Table Relased For Ternary Relationship Between Instructor, Student And Exam (Student_id ,Exam_id , Instructor_id , Year Primary key)
dbo.multi_choice	Multi_choice table contain other choices of multi-choice question (Choice_text and Question_ID as Primary Key )
dbo.question_exam	Question_Exam table relased from relation between Exam And Questions Contain questions for every Exam and Question Degree (Exam_ID and Question_ID as Primary Key)
dbo.question_pool	Question_Pool contain Questions , question type , Model_Answer (Question_ID as Primary Key ) and (Course_ID and Instructor_ID as Foreign Key)
dbo.student	Student Table Contain Name , E-Mail , Birthdate , phone , City and Street (Student_ID as Primary Key ) And (Course_ID as Foreign Key)
dbo.student_question_exam	Student_Question_Exam Table Relased For Ternary Relationship Between Student, Questions And Exam (Student_id ,Question_id and Exam_id as Primary key)

## audit\_admin.audit1\_class

### Columns

Key	Name	Data Type	Length	Precision	Scale	Not Null	Identity	Rule	Default	Computed	Persisted	Description
	class_id	int	4	10	0	False				False	False	
	class_name	nvarchar	0	0	0	False				False	False	
	users_name	nvarchar	0	0	0	False				False	False	
	modifieddate	date	3	10	0	False				False	False	
	old_class_name	nvarchar	0	0	0	False				False	False	
	inserted_class_name	nvarchar	0	0	0	False				False	False	

### SQL Script

```

CREATE TABLE audit_admin.audit1_class (
    class_id int NULL,
    class_name nvarchar(max) NULL,
    users_name nvarchar(max) NULL,
    modifieddate date NULL,
    old_class_name nvarchar(max) NULL,
    inserted_class_name nvarchar(max) NULL
)
ON [PRIMARY]
TEXTIMAGE_ON [PRIMARY]
GO

```

### Used By

No items found

## audit\_admin.audit2\_class\_instruct

### Columns

Key	Name	Data Type	Length	Precision	Scale	Not Null	Identity	Rule	Default	Computed	Persisted	Description
	class_id	int	4	10	0	False				False	False	
	course_id	int	4	10	0	False				False	False	
	instruct_id	nvarchar		0	0	False				False	False	
	users_name	nvarchar		0	0	False				False	False	
	modifieddate	date	3	10	0	False				False	False	
	old_year	date	3	10	0	False				False	False	
	new_year	date	3	10	0	False				False	False	

### SQL Script

```

CREATE TABLE audit_admin.audit2_class_instruct (
    class_id int NULL,
    course_id int NULL,
    instruct_id nvarchar(max) NULL,
    users_name nvarchar(max) NULL,
    modifieddate date NULL,
    old_year date NULL,
    new_year date NULL
)
ON [PRIMARY]
TEXTIMAGE_ON [PRIMARY]
GO

```

### Used By

No items found

## audit\_admin.audit3\_course

### Columns

Key	Name	Data Type	Length	Precision	Scale	Not Null	Identity	Rule	Default	Computed	Persisted	Description
	course_id	int	4	10	0	False				False	False	
	users_name	nvarchar		0	0	False				False	False	
	modifieddate	date	3	10	0	False				False	False	
	old_course_name	nvarchar		0	0	False				False	False	
	new_course_name	nvarchar		0	0	False				False	False	
	old_course_description	nvarchar		0	0	False				False	False	
	new_course_description	nvarchar		0	0	False				False	False	
	old_course_maxDegree	int	4	10	0	False				False	False	
	new_course_maxDegree	int	4	10	0	False				False	False	
	old_course_minDegree	int	4	10	0	False				False	False	
	new_course_minDegree	int	4	10	0	False				False	False	

### SQL Script

```

CREATE TABLE audit_admin.audit3_course (
    course_id int NULL,
    users_name nvarchar(max) NULL,
    modifieddate date NULL,
    old_course_name nvarchar(max) NULL,
    new_course_name nvarchar(max) NULL,
    old_course_description nvarchar(max) NULL,
    new_course_description nvarchar(max) NULL,
    old_course_maxDegree int NULL,
    new_course_maxDegree int NULL,
    old_course_minDegree int NULL,
    new_course_minDegree int NULL
)
ON [PRIMARY]
TEXTIMAGE_ON [PRIMARY]
GO

```

### Used By

No items found

## audit\_admin.audit4\_course\_student

### Columns

Key	Name	Data Type	Length	Precision	Scale	Not Null	Identity	Rule	Default	Computed	Persisted	Description
	course_id	int	4	10	0	False				False	False	
	users_name	nvarchar	0	0	0	False				False	False	
	modifieddate	date	3	10	0	False				False	False	
	old_final_result	int	4	10	0	False				False	False	
	new_final_result	int	4	10	0	False				False	False	

### SQL Script

```

CREATE TABLE audit_admin.audit4_course_student (
    course_id int NULL,
    users_name nvarchar(max) NULL,
    modifieddate date NULL,
    old_final_result int NULL,
    new_final_result int NULL
)
ON [PRIMARY]
TEXTIMAGE_ON [PRIMARY]
GO

```

### Used By

No items found

## audit\_admin.audit4\_exam

### Columns

Key	Name	Data Type	Length	Precision	Scale	Not Null	Identity	Rule	Default	Computed	Persisted	Description
	exam_id	int	4	10	0	False				False	False	
	users_name	nvarchar		0	0	False				False	False	
	modifieddate	date	3	10	0	False				False	False	
	old_exam_starttime	time	5	16	7	False				False	False	
	new_exam_starttime	time	5	16	7	False				False	False	
	old_exam_endtime	time	5	16	7	False				False	False	
	new_exam_endtime	time	5	16	7	False				False	False	
	old_exam_type	varchar		0	0	False				False	False	
	new_exam_type	varchar		0	0	False				False	False	
	old_exam_date	date	3	10	0	False				False	False	
	new_exam_date	date	3	10	0	False				False	False	
	old_exam_total_degree	int	4	10	0	False				False	False	
	new_exam_total_degree	int	4	10	0	False				False	False	
	course_id	int	4	10	0	False				False	False	
	instruct_id	char	14	0	0	False				False	False	

### SQL Script

```

CREATE TABLE audit_admin.audit4_exam (
    exam_id int NULL,
    users_name nvarchar(max) NULL,
    modifieddate date NULL,
    old_exam_starttime time NULL,
    new_exam_starttime time NULL,
    old_exam_endtime time NULL,
    new_exam_endtime time NULL,
    old_exam_type varchar(max) NULL,
    new_exam_type varchar(max) NULL,
    old_exam_date date NULL,
    new_exam_date date NULL,
    old_exam_total_degree int NULL,
    new_exam_total_degree int NULL,
    course_id int NULL,
    instruct_id char(14) NULL
)
ON [PRIMARY]
TEXTIMAGE_ON [PRIMARY]
GO

```

### Used By

No items found

## audit\_admin.audit4\_instructor

### Columns

Key	Name	Data Type	Length	Precision	Scale	Not Null	Identity	Rule	Default	Computed	Persisted	Description
	instruct_id	int	4	10	0	False				False	False	
	users_name	nvarchar		0	0	False				False	False	
	modifieddate	date	3	10	0	False				False	False	
	instruct_name	nvarchar		0	0	False				False	False	
	instruct_birthdate	date	3	10	0	False				False	False	
	instruct_phone	char	11	0	0	False				False	False	
	instruct_city	nvarchar		0	0	False				False	False	
	instruct_streat	nvarchar		0	0	False				False	False	

### SQL Script

```

CREATE TABLE audit_admin.audit4_instructor (
    instruct_id int NULL,
    users_name nvarchar(max) NULL,
    modifieddate date NULL,
    instruct_name nvarchar(max) NULL,
    instruct_birthdate date NULL,
    instruct_phone char(11) NULL,
    instruct_city nvarchar(max) NULL,
    instruct_streat nvarchar(max) NULL
)
ON [PRIMARY]
TEXTIMAGE_ON [PRIMARY]
GO

```

### Used By

No items found

## audit\_admin.audit5\_instructor\_student\_exam

### Columns

Key	Name	Data Type	Length	Precision	Scale	Not Null	Identity	Rule	Default	Computed	Persisted	Description
	student_id	int	4	10	0	False				False	False	
	users_name	nvarchar		0	0	False				False	False	
	modifieddate	date	3	10	0	False				False	False	
	instruct_id	char	14	0	0	False				False	False	
	exam_result	int	4	10	0	False				False	False	

### SQL Script

```
CREATE TABLE audit_admin.audit5_instructor_student_exam (
    student_id int NULL,
    users_name nvarchar(max) NULL,
    modifieddate date NULL,
    instruct_id char(14) NULL,
    exam_result int NULL
)
ON [PRIMARY]
TEXTIMAGE_ON [PRIMARY]
GO
```

### Used By

No items found

## audit\_admin.audit6\_multi\_choice

### Columns

Key	Name	Data Type	Length	Precision	Scale	Not Null	Identity	Rule	Default	Computed	Persisted	Description
	choice_text	varchar		0	0	False				False	False	
	users_name	nvarchar		0	0	False				False	False	
	modifieddate	date	3	10	0	False				False	False	
	question_id	int	4	10	0	False				False	False	

### SQL Script

```
CREATE TABLE audit_admin.audit6_multi_choice (
    choice_text varchar(max) NULL,
    users_name nvarchar(max) NULL,
    modifieddate date NULL,
    question_id int NULL
)
ON [PRIMARY]
TEXTIMAGE_ON [PRIMARY]
GO
```

### Used By

No items found

## audit\_admin.audit6\_question\_exam

### Columns

Key	Name	Data Type	Length	Precision	Scale	Not Null	Identity	Rule	Default	Computed	Persisted	Description
	exam_id	int	4	10	0	False				False	False	
	users_name	nvarchar		0	0	False				False	False	
	modifieddate	date	3	10	0	False				False	False	
	question_degree	int	4	10	0	False				False	False	

### SQL Script

```
CREATE TABLE audit_admin.audit6_question_exam (
    exam_id int NULL,
    users_name nvarchar(max) NULL,
    modifieddate date NULL,
    question_degree int NULL
)
ON [PRIMARY]
TEXTIMAGE_ON [PRIMARY]
GO
```

### Used By

No items found

## audit\_admin.audit6\_question\_pool

### Columns

Key	Name	Data Type	Length	Precision	Scale	Not Null	Identity	Rule	Default	Computed	Persisted	Description
	question_id	int	4	10	0	False				False	False	
	users_name	nvarchar		0	0	False				False	False	
	modifieddate	date	3	10	0	False				False	False	
	old_question_text	varchar		0	0	False				False	False	
	new_question_text	varchar		0	0	False				False	False	
	old_question_type	varchar		0	0	False				False	False	
	new_question_type	varchar		0	0	False				False	False	
	old_model_answer	varchar		0	0	False				False	False	
	new_model_answer	varchar		0	0	False				False	False	
	course_id	int	4	10	0	False				False	False	
	instruct_id	char	14	0	0	False				False	False	

### SQL Script

```

CREATE TABLE audit_admin.audit6_question_pool (
    question_id int NULL,
    users_name nvarchar(max) NULL,
    modifieddate date NULL,
    old_question_text varchar(max) NULL,
    new_question_text varchar(max) NULL,
    old_question_type varchar(max) NULL,
    new_question_type varchar(max) NULL,
    old_model_answer varchar(max) NULL,
    new_model_answer varchar(max) NULL,
    course_id int NULL,
    instruct_id char(14) NULL
)
ON [PRIMARY]
TEXTIMAGE_ON [PRIMARY]
GO

```

### Used By

No items found

## audit\_admin.audit7\_student

### Columns

Key	Name	Data Type	Length	Precision	Scale	Not Null	Identity	Rule	Default	Computed	Persisted	Description
	student_id	int	4	10	0	False				False	False	
	users_name	nvarchar		0	0	False				False	False	
	modifieddate	date	3	10	0	False				False	False	
	student_name	nvarchar		0	0	False				False	False	
	student_email	nvarchar		0	0	False				False	False	
	student_birthDate	date	3	10	0	False				False	False	
	student_phone	char	11	0	0	False				False	False	
	student_city	nvarchar		0	0	False				False	False	
	student_street	nvarchar		0	0	False				False	False	
	class_id	int	4	10	0	False				False	False	

### SQL Script

```

CREATE TABLE audit_admin.audit7_student (
    student_id int NULL,
    users_name nvarchar(max) NULL,
    modifieddate date NULL,
    student_name nvarchar(max) NULL,
    student_email nvarchar(max) NULL,
    student_birthDate date NULL,
    student_phone char(11) NULL,
    student_city nvarchar(max) NULL,
    student_street nvarchar(max) NULL,
    class_id int NULL
)
ON [PRIMARY]
TEXTIMAGE_ON [PRIMARY]
GO

```

### Used By

No items found

## audit\_admin.audit7\_student\_question\_exam

### Columns

Key	Name	Data Type	Length	Precision	Scale	Not Null	Identity	Rule	Default	Computed	Persisted	Description
	student_id	int	4	10	0	False				False	False	
	users_name	nvarchar		0	0	False				False	False	
	modifieddate	date	3	10	0	False				False	False	
	question_answer	nvarchar		0	0	False				False	False	

### SQL Script

```
CREATE TABLE audit_admin.audit7_student_question_exam (
    student_id int NULL,
    users_name nvarchar(max) NULL,
    modifieddate date NULL,
    question_answer nvarchar(max) NULL
)
ON [PRIMARY]
TEXTIMAGE_ON [PRIMARY]
GO
```

### Used By

No items found

## dbo.class

### Columns

Key	Name	Data Type	Length	Precision	Scale	Not Null	Identity	Rule	Default	Computed	Persisted	Description
	class_id	int	4	10	0	True	1 - 1			False	False	
	class_name	nvarchar	20	0	0	False				False	False	

### SQL Script

```
CREATE TABLE dbo.class (
    class_id int IDENTITY,
    class_name nvarchar(20) NULL,
    PRIMARY KEY CLUSTERED (class_id) ON exam_S_F_G_1
)
ON exam_S_F_G_1
GO
```

### Used By 4

-  [dbo.class\\_instructor\\_course](#)
-  [dbo.student](#)
-  [dbo.Instrucotr\\_Class\\_CourseInfo](#)
-  [dbo.Student\\_info](#)

## dbo.class\_instructor\_course

### Columns

Key	Name	Data Type	Length	Precision	Scale	Not Null	Identity	Rule	Default	Computed	Persisted	Description
	class_id	int	4	10	0	True				False	False	
	course_id	int	4	10	0	True				False	False	
	instruct_id	char	14	0	0	True				False	False	
	year	date	3	10	0	True			(getdate())	False	False	

### Foreign Keys

Name	Columns	Description
class_inst_course__instructor_FK	instruct_id	
class_inst_course_class_FK	class_id	
class_inst_course_course_FK	course_id	

### SQL Script

```

CREATE TABLE dbo.class_instructor_course (
    class_id int NOT NULL,
    course_id int NOT NULL,
    instruct_id char(14) NOT NULL,
    year date NOT NULL DEFAULT (getdate()),
    CONSTRAINT class_instructor_course_PK PRIMARY KEY CLUSTERED (class_id, course_id, instruct_id, year),
    CONSTRAINT class_inst_course_unique_uq UNIQUE (class_id, course_id, year)
)
ON [PRIMARY]
GO

ALTER TABLE dbo.class_instructor_course
    ADD CONSTRAINT class_inst_course__instructor_FK FOREIGN KEY (instruct_id) REFERENCES dbo.instructor (instruct_id)
GO

ALTER TABLE dbo.class_instructor_course
    ADD CONSTRAINT class_inst_course_class_FK FOREIGN KEY (class_id) REFERENCES dbo.class (class_id)
GO

ALTER TABLE dbo.class_instructor_course
    ADD CONSTRAINT class_inst_course_course_FK FOREIGN KEY (course_id) REFERENCES dbo.course (course_id)
GO

```

### Used By 7

- [dbo.Course\\_info](#)
- [dbo.Instrucotr\\_Class\\_CourseInfo](#)
- [dbo.Instructor](#)
- [dbo.class](#)
- [dbo.course](#)
- [dbo.instructor](#)
- [dbo.user](#)

 dbo.InsTructorDeatils  
 dbo.Student\_info\_course  
 dbo.assignExamForStudent  
 dbo.instruct\_exam  
 dbo.InstructorCourseByld

## dbo.course

### Columns

Key	Name	Data Type	Length	Precision	Scale	Not Null	Identity	Rule	Default	Computed	Persisted	Description
	course_id	int	4	10	0	True	1 - 1			False	False	
	course_name	nvarchar	20	0	0	True				False	False	
	course_description	nvarchar	60	0	0	False				False	False	
	course_maxDegree	int	4	10	0	True				False	False	
	course_minDegree	int	4	10	0	True				False	False	

### Check Constraints

Name	Columns	Condition	Description
course_Max_Min_Degree_ch	course_maxDegree, course_minDegree	([course_maxDegree]> [course_minDegree])	

### SQL Script

```
CREATE TABLE dbo.course (
    course_id int IDENTITY,
    course_name nvarchar(20) NOT NULL,
    course_description nvarchar(60) NULL,
    course_maxDegree int NOT NULL,
    course_minDegree int NOT NULL,
    PRIMARY KEY CLUSTERED (course_id) ON exam_S_F_G_1,
    CONSTRAINT course_Max_Min_Degree_ch CHECK ([course_maxDegree]>[course_minDegree])
)
ON exam_S_F_G_1
GO
```

### Used By

-  dbo.class\_instructor\_course
-  dbo.course\_student
-  dbo.exam
-  dbo.question\_pool
-  dbo.Course\_info
-  dbo.Course\_Questions\_Pool
-  dbo.Instrucotr\_Class\_CourseInfo
-  dbo.InsTructorDeatils
-  dbo.Student\_courseInfo
-  dbo.Student\_Exam\_Result
-  dbo.Student\_Exams
-  dbo.Student\_info\_course
-  dbo.StudentCoursesInfo
-  dbo.Generat\_exam\_manually
-  dbo.Generat\_exam\_random
-  dbo.calcFinalResCourse
-  dbo.Proc\_Course\_Exams
-  dbo.Show\_student\_courses

 dbo.showExamToAnswer  
 dbo.InstructorCourseById

## dbo.course\_student

### Columns

Key	Name	Data Type	Length	Precision	Scale	Not Null	Identity	Rule	Default	Computed	Persisted	Description
	course_id	int	4	10	0	True				False	False	
	student_id	int	4	10	0	True				False	False	
	final_result	int	4	10	0	False				False	False	

### Foreign Keys

Name	Columns	Description
course_student_course_FK	course_id	
course_student_student_FK	student_id	

### SQL Script

```

CREATE TABLE dbo.course_student (
    course_id int NOT NULL,
    student_id int NOT NULL,
    final_result int NULL,
    CONSTRAINT course_student_PK PRIMARY KEY CLUSTERED (course_id, student_id)
)
ON [PRIMARY]
GO

ALTER TABLE dbo.course_student
ADD CONSTRAINT course_student_course_FK FOREIGN KEY (course_id) REFERENCES dbo.course (course_id)
GO

ALTER TABLE dbo.course_student
ADD CONSTRAINT course_student_student_FK FOREIGN KEY (student_id) REFERENCES dbo.student (student_id)
GO

```

### Used By 6

- [dbo.Student\\_courseInfo](#)
- [dbo.StudentCoursesInfo](#)
- [dbo.AssAllStdForExaCourseByld](#)
- [dbo.assignExamForStudent](#)
- [dbo.Proc\\_Course\\_Exams](#)
- [dbo.Show\\_student\\_courses](#)

## dbo.exam

### Columns

Key	Name	Data Type	Length	Precision	Scale	Not Null	Identity	Rule	Default	Computed	Persisted	Description
	exam_id	int	4	10	0	True	1 - 1			False	False	
	exam_startTime	time	5	16	7	True				False	False	
	exam_endTime	time	5	16	7	True				False	False	
	exam_type	nvarchar	20	0	0	True				False	False	
	exam_date	date	3	10	0	True				False	False	
	exam_total_degree	int	4	10	0	True				False	False	
	course_id	int	4	10	0	True				False	False	
	instruct_id	char	14	0	0	True				False	False	

### Check Constraints

Name	Columns	Condition	Description
check_type_exam	exam_type	([exam_type]='corrective' OR [exam_type]='exam')	
exam_exam_date_ch	exam_date	([exam_date]>getdate())	
exam_start_end_ch	exam_startTime, exam_endTime	([exam_startTime]<[exam_endTime])	

### Foreign Keys

Name	Columns	Description
course_exam_FK	course_id	
exam_instructor_FK	instruct_id	

### SQL Script

```

CREATE TABLE dbo.exam (
    exam_id int IDENTITY,
    exam_startTime time NOT NULL,
    exam_endTime time NOT NULL,
    exam_type nvarchar(20) NOT NULL,
    exam_date date NOT NULL,
    exam_total_degree int NOT NULL,
    course_id int NOT NULL,
    instruct_id char(14) NOT NULL,
    PRIMARY KEY CLUSTERED (exam_id) ON exam_S_F_G_2,
    CONSTRAINT check_type_exam CHECK ([exam_type]='corrective' OR [exam_type]='exam'),
    CONSTRAINT exam_exam_date_ch CHECK ([exam_date]>getdate()),
    CONSTRAINT exam_start_end_ch CHECK ([exam_startTime]<[exam_endTime])
)
ON exam_S_F_G_2
GO

```

```

ALTER TABLE dbo.exam
ADD CONSTRAINT course_exam_FK FOREIGN KEY (course_id) REFERENCES dbo.course (course_id)

```

GO

```
ALTER TABLE dbo.exam  
    ADD CONSTRAINT exam_instructor_FK FOREIGN KEY (instruct_id) REFERENCES dbo.instructor (instruct_id)
```

GO

## Used By 14

-  [dbo.instructor\\_student\\_exam](#)
-  [dbo.question\\_exam](#)
-  [dbo.student\\_question\\_exam](#)
-  [dbo.Exam\\_Content\\_Details](#)
-  [dbo.Student\\_Exam\\_Rslt](#)
-  [dbo.Student\\_Exams](#)
-  [dbo.Generat\\_exam\\_manually](#)
-  [dbo.Generat\\_exam\\_random](#)
-  [dbo.AssAllStdForExaCourseByld](#)
-  [dbo.assignExamForStudent](#)
-  [dbo.calcFinalResCourse](#)
-  [dbo.showExamToAnswer](#)
-  [dbo.instruct\\_exam](#)
-  [dbo.StudentExamByld](#)

## dbo.instructor

### Columns

Key	Name	Data Type	Length	Precision	Scale	Not Null	Identity	Rule	Default	Computed	Persisted	Description
	instruct_id	char	14	0	0	True				False	False	
	instruct_name	nvarchar	20	0	0	True				False	False	
	instruct_birthDate	date	3	10	0	True				False	False	
	instruct_phone	char	11	0	0	True				False	False	
	instruct_city	nvarchar	20	0	0	True				False	False	
	instruct_street	nvarchar	20	0	0	False				False	False	

### SQL Script

```
CREATE TABLE dbo.instructor (
    instruct_id char(14) NOT NULL,
    instruct_name nvarchar(20) NOT NULL,
    instruct_birthDate date NOT NULL,
    instruct_phone char(11) NOT NULL,
    instruct_city nvarchar(20) NOT NULL,
    instruct_street nvarchar(20) NULL,
    PRIMARY KEY CLUSTERED (instruct_id) ON exam_S_F_G_1
)
ON exam_S_F_G_1
GO
```

### Used By 10

-  [dbo.class\\_instructor\\_course](#)
-  [dbo.exam](#)
-  [dbo.instructor\\_student\\_exam](#)
-  [dbo.question\\_pool](#)
-  [dbo.Course\\_info](#)
-  [dbo.Course\\_Questions\\_Pool](#)
-  [dbo.Instrucotr\\_Class\\_CourseInfo](#)
-  [dbo.InsTructorDeatils](#)
-  [dbo.assignExamForStudent](#)
-  [dbo.InstructorCourseByld](#)

## dbo.instructor\_student\_exam

### Columns

Key	Name	Data Type	Length	Precision	Scale	Not Null	Identity	Rule	Default	Computed	Persisted	Description
	student_id	int	4	10	0	True				False	False	
	exam_id	int	4	10	0	True				False	False	
	instruct_id	char	14	0	0	True				False	False	
	exam_result	int	4	10	0	False				False	False	
	done	char	1	0	0	False		('0')		False	False	

### Foreign Keys

Name	Columns	Description
inst_std_exam_exam_FK	exam_id	
inst_std_exam_instructor_FK	instruct_id	
inst_std_exam_student_FK	student_id	

### SQL Script

```

CREATE TABLE dbo.instructor_student_exam (
    student_id int NOT NULL,
    exam_id int NOT NULL,
    instruct_id char(14) NOT NULL,
    exam_result int NULL,
    done char(1) NULL DEFAULT ('0'),
    CONSTRAINT instructor_student_exam_PK PRIMARY KEY CLUSTERED (student_id, instruct_id, exam_id)
)
ON [PRIMARY]
GO

ALTER TABLE dbo.instructor_student_exam
ADD CONSTRAINT inst_std_exam_exam_FK FOREIGN KEY (exam_id) REFERENCES dbo.exam (exam_id)
GO

ALTER TABLE dbo.instructor_student_exam
ADD CONSTRAINT inst_std_exam_instructor_FK FOREIGN KEY (instruct_id) REFERENCES dbo.instructor (instruct_id)
GO

ALTER TABLE dbo.instructor_student_exam
ADD CONSTRAINT inst_std_exam_student_FK FOREIGN KEY (student_id) REFERENCES dbo.student (student_id)
GO

```

### Used By 8

- [dbo.Student\\_Exam\\_Rslt](#)
- [dbo.AssAllStdForExaCourseByld](#)
- [dbo.assignExamForStudent](#)
- [dbo.calcFinalResCourse](#)

 dbo.countExamResultForStudent  
 dbo.Proc\_Course\_Exams  
 dbo.showExamToAnswer  
 dbo.StudentExamById

## dbo.multi\_choice

### Columns

Key	Name	Data Type	Length	Precision	Scale	Not Null	Identity	Rule	Default	Computed	Persisted	Description
	choise_text	varchar	100	0	0	True				False	False	
	question_id	int	4	10	0	True				False	False	

### Foreign Keys

Name	Columns	Description
multi_choice_question_FK	question_id	

### SQL Script

```

CREATE TABLE dbo.multi_choice (
    choise_text varchar(100) NOT NULL,
    question_id int NOT NULL,
    CONSTRAINT multi_choice_PK PRIMARY KEY CLUSTERED (choise_text, question_id)
)
ON [PRIMARY]
GO

ALTER TABLE dbo.multi_choice
ADD CONSTRAINT multi_choice_question_FK FOREIGN KEY (question_id) REFERENCES dbo.question_pool (question_id)
GO

```

### Used By

No items found

## dbo.question\_exam

### Columns

Key	Name	Data Type	Length	Precision	Scale	Not Null	Identity	Rule	Default	Computed	Persisted	Description
	exam_id	int	4	10	0	True				False	False	
	question_id	int	4	10	0	True				False	False	
	question_degree	int	4	10	0	False				False	False	

### Foreign Keys

Name	Columns	Description
question_exam_exam_FK	exam_id	
question_exam_question_FK	question_id	

### SQL Script

```

CREATE TABLE dbo.question_exam (
    exam_id int NOT NULL,
    question_id int NOT NULL,
    question_degree int NULL,
    CONSTRAINT question_exam_PK PRIMARY KEY CLUSTERED (question_id, exam_id)
)
ON [PRIMARY]
GO

ALTER TABLE dbo.question_exam
ADD CONSTRAINT question_exam_exam_FK FOREIGN KEY (exam_id) REFERENCES dbo.exam (exam_id)
GO

ALTER TABLE dbo.question_exam
ADD CONSTRAINT question_exam_question_FK FOREIGN KEY (question_id) REFERENCES dbo.question_pool (question_id)
GO

```

### Used By 4

- [dbo.Exam\\_Content\\_Details](#)
- [dbo.Generat\\_exam\\_manaully](#)
- [dbo.Generat\\_exam\\_random](#)
- [dbo.countExamResultForStudent](#)

## dbo.question\_pool

### Columns

Key	Name	Data Type	Length	Precision	Scale	Not Null	Identity	Rule	Default	Computed	Persisted	Description
	question_id	int	4	10	0	True	1 - 1			False	False	
	question_text	varchar		0	0	True				False	False	
	question_type	varchar	20	0	0	True				False	False	
	model_answer	varchar	100	0	0	True				False	False	
	course_id	int	4	10	0	True				False	False	
	instruct_id	char	14	0	0	True				False	False	

### Check Constraints

Name	Columns	Condition	Description
question_type_CH	question_type	([question_type]='choise' OR [question_type]='text' OR [question_type]='true & false')	

### Foreign Keys

Name	Columns	Description
question_course_FK	course_id	
question_instructor_FK	instruct_id	

### SQL Script

```

CREATE TABLE dbo.question_pool (
    question_id int IDENTITY,
    question_text varchar(max) NOT NULL,
    question_type varchar(20) NOT NULL,
    model_answer varchar(100) NOT NULL,
    course_id int NOT NULL,
    instruct_id char(14) NOT NULL,
    PRIMARY KEY CLUSTERED (question_id) ON exam_S_F_G_2,
    CONSTRAINT question_type_CH CHECK ([question_type]='choise' OR [question_type]='text' OR [question_type]='true & false')
)
ON exam_S_F_G_2
TEXTIMAGE_ON exam_S_F_G_2
GO

ALTER TABLE dbo.question_pool
ADD CONSTRAINT question_course_FK FOREIGN KEY (course_id) REFERENCES dbo.course (course_id)
GO

ALTER TABLE dbo.question_pool
ADD CONSTRAINT question_instructor_FK FOREIGN KEY (instruct_id) REFERENCES dbo.instructor (instruct_id)
GO

```

### Used By 9

- [dbo.multi\\_choice](#)
- [dbo.question\\_exam](#)

dbo.student\_question\_exam  
dbo.Course\_Questions\_Pool  
dbo.Exam\_Content\_Details  
dbo.Student\_Exams  
dbo.Generat\_exam\_manually  
dbo.Generat\_exam\_random  
dbo.countExamResultForStudent

## dbo.student

### Columns

Key	Name	Data Type	Length	Precision	Scale	Not Null	Identity	Rule	Default	Computed	Persisted	Description
PK	student_id	int	4	10	0	True				False	False	
	student_name	nvarchar	20	0	0	True				False	False	
	student_email	nvarchar	50	0	0	False				False	False	
	student_birthDate	date	3	10	0	True				False	False	
	student_phone	char	11	0	0	True				False	False	
	student_city	nvarchar	20	0	0	True				False	False	
	student_street	nvarchar	20	0	0	False				False	False	
FK	class_id	int	4	10	0	True				False	False	

### Foreign Keys

Name	Columns	Description
student_class_FK	class_id	

### SQL Script

```

CREATE TABLE dbo.student (
    student_id int NOT NULL,
    student_name nvarchar(20) NOT NULL,
    student_email nvarchar(50) NULL,
    student_birthDate date NOT NULL,
    student_phone char(11) NOT NULL,
    student_city nvarchar(20) NOT NULL,
    student_street nvarchar(20) NULL,
    class_id int NOT NULL,
    PRIMARY KEY CLUSTERED (student_id) ON exam_S_F_G_2
)
ON exam_S_F_G_2
GO

ALTER TABLE dbo.student
ADD CONSTRAINT student_class_FK FOREIGN KEY (class_id) REFERENCES dbo.class (class_id)
GO

```

### Used By 14

-  [dbo.course\\_student](#)
-  [dbo.instructor\\_student\\_exam](#)
-  [dbo.student\\_question\\_exam](#)
-  [dbo.Student\\_courseInfo](#)
-  [dbo.Student\\_Exam\\_Result](#)
-  [dbo.Student\\_Exams](#)
-  [dbo.Student\\_info](#)
-  [dbo.Student\\_info\\_course](#)
-  [dbo.StudentCoursesInfo](#)
-  [dbo.AssAllStdForExaCourseByld](#)

 dbo.assignExamForStudent  
 dbo.calcFinalResCourse  
 dbo.Show\_student\_courses  
 dbo.StudentExamById

## dbo.student\_question\_exam

### Columns

Key	Name	Data Type	Length	Precision	Scale	Not Null	Identity	Rule	Default	Computed	Persisted	Description
	student_id	int	4	10	0	True				False	False	
	question_id	int	4	10	0	True				False	False	
	exam_id	int	4	10	0	True				False	False	
	student_answer	nvarchar	100	0	0	False				False	False	

### Foreign Keys

Name	Columns	Description
std_quest_exam_exam_FK	exam_id	
std_quest_exam_question_FK	question_id	
std_quest_exam_student_FK	student_id	

### SQL Script

```

CREATE TABLE dbo.student_question_exam (
    student_id int NOT NULL,
    question_id int NOT NULL,
    exam_id int NOT NULL,
    student_answer nvarchar(100) NULL,
    CONSTRAINT student_question_exam_PK PRIMARY KEY CLUSTERED (student_id, question_id, exam_id)
)
ON [PRIMARY]
GO

ALTER TABLE dbo.student_question_exam
ADD CONSTRAINT std_quest_exam_exam_FK FOREIGN KEY (exam_id) REFERENCES dbo.exam (exam_id)
GO

ALTER TABLE dbo.student_question_exam
ADD CONSTRAINT std_quest_exam_question_FK FOREIGN KEY (question_id) REFERENCES dbo.question_pool (question_id)
GO

ALTER TABLE dbo.student_question_exam
ADD CONSTRAINT std_quest_exam_student_FK FOREIGN KEY (student_id) REFERENCES dbo.student (student_id)
GO

```

### Used By 2

- [dbo.Student\\_Exams](#)
- [dbo.countExamResultForStudent](#)

 **Views**
**Objects (11)**

Name	Description
dbo.Course_info	View Show Instructor Name , His Courses And Course Description
dbo.Course_Questions_Pool	View Show instructor_ID , Instructor_Name , Course_Name , Questions , Question_type and Model Answer for every Course
dbo.Exam_Content_Details	View Show Exam_ID , Exam_type , Exam_Date , Exam_Questions , Question_type , Queestion_Degree and Model Answer
dbo.Instrucotr_Class_CourseInfo	View Show instructor_ID , Instructor_Name , Class_Name , and Course_Name
dbo.InsTructorDeatils	View Show instructor_ID , Instructor_Name , Ins_Phone , Address and Courses
dbo.Student_coursesInfo	View Show Student_ID , Student_Name , Course_ID , Course_Name and Course_Final_Result
dbo.Student_Exam_Result	View Show Exam_ID , Student_Name , Course_Name , Exam_type Exam_Date and Exam_Result
dbo.Student_Exams	View Show Student_ID , Student_Name , Exam_ID , Exam_Type , Course_Name , Questions , Student_Answer and Model Answer
dbo.Student_info	View Show Student_Name , address , E-Mail , Phone And Name of his class
dbo.Student_info_course	View Show Student_Name , Course_Name and Education Year
dbo.StudentCoursesInfo	View Show Student_ID , Student_Name , Course_Name ,Course_MaxDegree and Course_MinDegree

## dbo.Course\_info

### Columns

Key	Name	Description
	Instructor Name	
	Course Name	
	Course Description	

### SQL Script

```
SET QUOTED_IDENTIFIER, ANSI_NULLS ON
GO
create OR ALTER view dbo.Course_info([Instructor Name],[Course Name],[Course Description])
as
(
    select i.instruct_name,co.course_name,co.course_description
    from instructor i,class_instructor_course cic,course co
    where i.instruct_id=cic.instruct_id and co.course_id=cic.course_id
)
GO
```

### Used By

No items found

## dbo.Course\_Questions\_Pool

### Columns

Key	Name	Description
	Ins_ID	
	Instrucotr Name	
	Course Name	
	Question Type	
	Question	
	Model Answer	

### SQL Script

```
SET QUOTED_IDENTIFIER, ANSI_NULLS ON
GO
Create OR ALTER View dbo.Course_Questions_Pool ([Ins_ID] , [Instrucotr Name] , [Course Name] , [Question Type] ,
[Question] , [Model Answer])
as
select I.instruct_id ,I.instruct_name, Cr.course_name , QP.question_type , QP.question_text , QP.model_answer
from question_pool QP , Course Cr , Instructor I
where Cr.course_id = QP.course_id AND I.instruct_id = QP.instruct_id
)
GO
```

### Used By

No items found

## dbo.Exam\_Content\_Details

### Columns

Key	Name	Description
	Exam ID	
	Exam Type	
	Exam Date	
	Exam Questions	
	Question Type	
	Question Degree	
	Model Answer	

### SQL Script

```
SET QUOTED_IDENTIFIER ANSI_NULLS ON
GO
CREATE OR ALTER VIEW dbo.Exam_Content_Details([Exam ID] , [Exam Type] , [Exam Date] , [Exam Questions],
[Question Type] , [Question Degree] , [Model Answer])
AS
(
    SELECT e.exam_id , E.exam_type , exam_date , Qp.question_text , Qp.question_type , QE.question_degree , Qp.
    model_answer
    FROM Exam E , question_exam QE , question_pool QP
    WHERE E.exam_id = QE.exam_id AND QE.question_id = QP.question_id
)
GO
```

### Used By ②

-  [dbo.CreatedExam](#)
-  [dbo.showExamToAnswer](#)

## dbo.Instrucotr\_Class\_CourseInfo

### Columns

Key	Name	Description
	ID	
	Instrucotr Name	
	Class Name	
	Course Name	

### SQL Script

```
SET QUOTED_IDENTIFIER, ANSI_NULLS ON
GO
create OR ALTER View dbo.Instrucotr_Class_CourseInfo ([ID] , [Instrucotr Name] , [Class Name] , [Course Name] )
as
(
select I.instruct_id, I.instruct_name , C.class_name ,Cr.course_name
from course Cr, instructor I , class C , class_instructor_course CI
where I.instruct_id = CI.instruct_id
AND C.class_id = Ci.class_id
AND Cr.course_id = CI.course_id
)
GO
```

### Used By

No items found

## dbo.InsTructorDeatils

### Columns

Key	Name	Description
	INS ID	
	Instructor Name	
	Phone	
	Address	
	Course	

### SQL Script

```
SET QUOTED_IDENTIFIER, ANSI_NULLS ON
GO
Create OR ALTER View dbo.InsTructorDeatils ([INS ID] , [Instructor Name],[Phone] , [Address] , [Course])
as
(
select I.instruct_id , I.instruct_name , I.instruct_phone ,I.instruct_city +' '+i.instruct_street , C.course_name
from instructor I , course C , class_instructor_course CI
where I.instruct_id = CI.instruct_id And C.course_id = CI.course_id
)
GO
```

### Used By

No items found

## dbo.Student\_courseInfo

### Columns

Key	Name	Description
	ID	
	Student Name	
	Course ID	
	Course Name	
	Course Final Result	

### SQL Script

```
SET QUOTED_IDENTIFIER, ANSI_NULLS ON
GO
Create OR ALTER View dbo.Student_courseInfo ([ID] , [Student Name] , [Course ID] , [Course Name] ,
[Course Final Result])
as
(
select S.student_id, S.student_name , Cr.course_id, Cr.course_name ,Cs.final_result
from student S , course Cr , course_student CS
where S.student_id = CS.student_id AND Cr.course_id = CS.course_id
)
GO
```

### Used By

No items found

## dbo.Student\_Exam\_Result

### Columns

Key	Name	Description
	ID	
	Student Name	
	Course Name	
	Exam Type	
	Exam Date	
	Exam Result	

### SQL Script

```
SET QUOTED_IDENTIFIER, ANSI_NULLS ON
GO
create OR ALTER View dbo.Student_Exam_Result ( [ID], [Student Name] , [Course Name] ,[Exam Type] , [Exam Date] ,
[Exam Result] )
as
select S.student_id, S.student_name , Cr.course_name , E.exam_type , E.exam_date , ISE.exam_result
from instructor_student_exam ISE , exam E , Student S , course Cr
where S.student_id = ISE.student_id AND Cr.course_id = E.course_id
AND E.exam_id = ISE.exam_id
)
GO
```

### Used By

No items found

## dbo.Student\_Exams

### Columns

Key	Name	Description
	Student ID	
	Student Name	
	Exam ID	
	Exam Type	
	Course Name	
	Questions	
	Student Answer	
	Model Answer	

### SQL Script

```
SET QUOTED_IDENTIFIER, ANSI_NULLS ON
GO
Create OR ALTER View dbo.Student_Exams ([Student ID],[Student Name] , [Exam ID] , [Exam Type] , [Course Name] ,
[Questions] , [Student Answer] , [Model Answer])
as
(
select S.student_id , S.student_name , E.exam_id , E.exam_type, Cr.course_name , QP.question_text , SQE.
student_answer , Qp.model_answer
From Student S , student_question_exam SQE , exam E , question_pool QP , course Cr
where S.student_id = SQE.student_id AND E.exam_id = SQE.exam_id
And Qp.question_id = SQE.question_id AND E.course_id = Cr.course_id
)
GO
```

### Used By ①

 [dbo.Proc\\_Course\\_Exams](#)

## dbo.Student\_info

### Columns

Key	Name	Description
	Student Name	
	Student Address	
	Email	
	Phone	
	Class Name	

### SQL Script

```
SET QUOTED_IDENTIFIER, ANSI_NULLS ON
GO
create OR ALTER view dbo.Student_info([Student Name] , [Student Address] , [Email] , [Phone] , [Class Name])
as
(
    select S.student_name,s.student_city +' '+s.student_street, s.student_email ,s.student_phone,  C.class_name
    from student S,class c
    where c.class_id=s.class_id
)
GO
```

### Used By

No items found

## dbo.Student\_info\_course

### Columns

Key	Name	Description
	Student Name	
	Course Name	
	Education Year	

### SQL Script

```
SET QUOTED_IDENTIFIER, ANSI_NULLS ON
GO
CREATE OR ALTER VIEW dbo.Student_info_course([Student Name],[Course Name],[Education Year])
AS
(
    SELECT s.student_name, co.course_name, cic.year
    FROM student S, class_instructor_course cic, course co
    WHERE s.class_id=cic.class_id AND co.course_id=cic.course_id
)
GO
```

### Used By

No items found

## dbo.StudentCoursesInfo

### Columns

Key	Name	Description
	ID	
	Student Name	
	Course Name	
	Course Max Degree	
	Course MIn Degree	

### SQL Script

```
SET QUOTED_IDENTIFIER, ANSI_NULLS ON
GO
Create OR ALTER view dbo.StudentCoursesInfo (ID , [Student Name] , [Course Name] , [Course Max Degree] ,
[Course MIn Degree] )
as
(
select S.student_id , S.student_name , C.course_name , C.course_maxDegree , C.course_minDegree
from student S , course C ,course_student CS
where S.student_id = CS.student_id AND c.course_id = CS.course_id
)
GO
```

### Used By

No items found

## Programmability

### Objects 4

 Stored Procedures

 Functions

 Types

 Rules

## Stored Procedures

### Objects 14

Name	Description
dbo.AssAllStdForExaCourseByld	Procedure to assign all student in a specific Course In this Course Exam
dbo.assignExamForMStudent	Procedure to assign Exam for Multi student (Take a table of students ) in a specific Course
dbo.assignExamForStudent	Procedure to assign Exam for Specific student in a specific Course
dbo.calcFinalResCourse	Procedure to Calculate final result in course For student
dbo.countExamResultForStudent	Procedure Return Exam Result For Specific Exam
dbo.CreatedExam	Procedure to view exam after created
dbo.differentalbackup	Procedure to Make Differential Backup
dbo.fullbackup	Procedure to Make Full Backup
dbo.Generat_exam_manually	Procedure to generate Exam Manually
dbo.Generat_exam_random	Procedure to generate Exam Randomly
dbo.logbackup	Procedure to Make Transaction log Backup
dbo.Proc_Course_Exams	Procedure Return student Exams in a specific Course
dbo.Show_student_courses	Procedure Return student Course he attend
dbo.showExamToAnswer	Procedure Show Exam For Student To attend this Exam

## dbo.AssAllStdForExaCourseById

### Parameters

Name	Data Type	Length	Description
@exam_id	int	4	
@course_id	int	4	
@instruct_id	char	14	

### SQL Script

```

SET QUOTED_IDENTIFIER, ANSI_NULLS ON
GO
create OR ALTER proc dbo.AssAllStdForExaCourseById @exam_id int ,@course_id int ,@instruct_id char(14)
as
begin
begin tran
begin try
if @exam_id = (select exam_id from exam where course_id = @course_id and instruct_id=@instruct_id)
begin
declare @student_id int;
declare @index int=1;
declare @count int= (select count(s.student_id) from student s,course_student cs where s.student_id = cs.student_id and cs.course_id =@course_id);
while @index < @count
begin
with tableAllStudent AS
(select s.student_id,ROW_NUMBER() over(order by s.student_id) rownum from student s,course_student cs where s.student_id = cs.student_id and cs.course_id =@course_id)
select @student_id=tas.student_id from tableAllStudent tas where rownum = @index;
print @student_id;
if @student_id is not null
begin
insert into instructor_student_exam(student_id,exam_id,instruct_id) values(@student_id,@exam_id,@instruct_id);
end
else
begin
raiserror('not exist student in this courses or the data not correct',16,10);
rollback tran;
end
set @index = @index+1;
end
commit tran
end
else
begin
rollback tran;
end
end try
begin catch
rollback;
throw;
select ERROR_MESSAGE();
end catch
end
GO

```

### Used By

No items found

## dbo.assignExamForMStudent

### Parameters

Name	Data Type	Length	Description
@examStudent	assigExamStudent_type	max	

### SQL Script

```

SET QUOTED_IDENTIFIER ANSI_NULLS ON
GO
CREATE OR ALTER PROC dbo.assignExamForMStudent @examStudent assigExamStudent_type readonly
AS
BEGIN
    BEGIN TRY
        DECLARE @index INT = 1;
        DECLARE @lastIndex INT = (SELECT COUNT(*) FROM @examStudent);
        DECLARE @student INT;
        DECLARE @exam INT;
        DECLARE @instruct CHAR(14);
        WHILE @index <= @lastIndex
        BEGIN
            WITH cte AS ( SELECT *, ROW_NUMBER() OVER( ORDER BY student_id) AS ROW_NUM
                FROM @examStudent)
            SELECT @student = student_id, @exam = exam_id, @instruct = instruct_id
            FROM cte WHERE ROW_NUM = @index;
            EXEC assignExamForStudent @student, @exam, @instruct;
            SET @index = @index + 1;
        END
        COMMIT;
    END TRY
    BEGIN CATCH
        ROLLBACK;
        SELECT ERROR_MESSAGE();
    END CATCH
    END;
GO

```

### Used By

No items found

## dbo.assignExamForStudent

### Parameters

Name	Data Type	Length	Description
@student_id	int	4	
@exam_id	int	4	
@instruct_id	char	14	

### SQL Script

```

SET QUOTED_IDENTIFIER, ANSI_NULLS ON
GO
create OR ALTER proc dbo.assignExamForStudent @student_id int,@exam_id int,@instruct_id char(14)
as
begin
begin tran
begin try
if exists(select e.exam_id,s.student_id,i.instruct_id from exam e,course_student cs,student s,instructor i,
class_instructor_course cic
where e.course_id=cs.course_id and e.exam_id = @exam_id and cs.student_id=@student_id
and cs.student_id = s.student_id and i.instruct_id = cic.instruct_id and i.instruct_id = @instruct_id and cic.
course_id=cs.course_id)
begin
insert into instructor_student_exam(student_id,exam_id,instruct_id) values(@student_id,@exam_id,@instruct_id);
end
else
begin
raiserror('student and exam not exist in the same course',16,10);
rollback tran;
end
commit tran;
end try
begin catch
select ERROR_MESSAGE() as 'Message Error';
rollback tran;
end catch
end
GO

```

### Used By ①

 dbo.assignExamForMStudent

## dbo.calcFinalResCourse

### Parameters

Name	Data Type	Length	Description
@course_id	int	4	
@student_id	int	4	

### SQL Script

```

SET QUOTED_IDENTIFIER, ANSI_NULLS ON
GO
create OR ALTER proc dbo.calcFinalResCourse @course_id int, @student_id int
as
begin
begin tran
begin try
declare @final_Res int ;
select @final_Res=sum(ise.exam_result)
from course c,exam e,instructor_student_exam ise,student s
where c.course_id = e.course_id and ise.exam_id = e.exam_id and ise.student_id = s.student_id and s.student_id=
@student_id
and c.course_id = @course_id
if @final_Res is not null
begin
print @final_Res
--update course_student
--set final_result = @final_Res
--where student_id = @student_id
commit tran
end
else
begin
raiserror('data not correct',16,10);
rollback tran
end
end try
begin catch
select ERROR_MESSAGE() as 'Error Message';
rollback tran
end catch
end
GO

```

### Used By

No items found.

# dbo.countExamResultForStudent

## Parameters

Name	Data Type	Length	Description
@student_id	int	4	
@exam_id	int	4	

## SQL Script

```

SET QUOTED_IDENTIFIER, ANSI_NULLS ON
GO
create OR ALTER proc dbo.countExamResultForStudent @student_id int,@exam_id int
as
begin
declare @index int=1;
declare @lastIndex int=(select count(*) from question_exam where exam_id = @exam_id);
declare @question int;
declare @degree int;
declare @total int=0;
declare @answer nvarchar(max);
declare @answer_std nvarchar(max);
while @index<=@lastIndex
begin
WITH cte AS ( SELECT    qe.question_id,qe.question_degree,sqe.student_answer,ROW_NUMBER()
              OVER( order BY qe.exam_id) AS ROW_NUM
        FROM question_exam qe,student_question_exam sqe
        where qe.exam_id = sqe.exam_id and qe.exam_id = @exam_id and sqe.student_id=@student_id)
SELECT @question = question_id,@degree=question_degree,@answer_std=student_answer
FROM   cte WHERE  ROW_NUM = @index;
select @answer=model_answer from question_pool where question_id=@question;
if(@answer=@answer_std)
begin
  set @total = @total+@degree
end
end
if(@total>0)
begin
  update instructor_student_exam
  set exam_result = @total
  where student_id = @student_id and exam_id = @exam_id
end
end
GO

```

## Used By

No items found

## dbo.CreatedExam

### Parameters

Name	Data Type	Length	Description
@examID	int	4	

### SQL Script

```
SET QUOTED_IDENTIFIER, ANSI_NULLS ON
GO
CREATE OR ALTER proc dbo.CreatedExam @examID int
as
begin
    select * from [dbo].[Exam_Content_Details]
    where [Exam ID] = @examID
end
GO
```

### Used By 2

-  [dbo.Generat\\_exam\\_manaully](#)
-  [dbo.Generat\\_exam\\_random](#)

## dbo.differentialbackup

### SQL Script

```
SET QUOTED_IDENTIFIER, ANSI_NULLS ON
GO
--#####
create OR ALTER proc dbo.differentialbackup
as
begin
    BACKUP DATABASE examination_sys
    TO DISK = 'examination_sys.bak'
    WITH DIFFERENTIAL;
end
GO
```

### Used By

No items found

## dbo.fullbackup

### SQL Script

```
SET QUOTED_IDENTIFIER, ANSI_NULLS ON
GO
create OR ALTER proc dbo.fullbackup
as
begin
    BACKUP DATABASE examination_sys
    TO DISK = 'examination_sys.bak'
end
GO
```

### Used By

No items found

## dbo.Generat\_exam\_maually

### Parameters

Name	Data Type	Length	Description
@tableType	table_exam_DT	max	
@collectionExam	tableExamC	max	

### SQL Script

```

SET QUOTED_IDENTIFIER, ANSI_NULLS ON
GO
create OR ALTER proc dbo.Generat_exam_maually @tableType table_exam_DT readonly,@collectionExam tableExamC
readonly
as
begin
begin tran
begin try
declare @totalDegree int;
declare @maxDegree int;
declare @lastIndex int;
declare @index int=1;
select question_id into #collectionExam from question_pool where course_id = (select course_id from @tableType);
select @totalDegree=sum(question_degree),@lastIndex=count(*) from @collectionExam;
select @maxDegree=course_maxDegree from course where course_id=(select course_id from @tableType);
if @totalDegree = (select total_degree from @tableType)
begin
if @maxDegree <@totalDegree
begin
raiserror('total degree for exam greater than max degree of this course the exam not save and question not
save',16,10);
throw 50005,'total degree for exam greater than max degree of this course the exam not save and question
not save',1;
end
else
begin
INSERT INTO exam SELECT * FROM @tableType;
declare @examID int = (select IDENT_CURRENT('exam'));
declare @question int,@degree int;
while @index <= @lastIndex
begin
with collectionExa as (select *,ROW_NUMBER() over(order by question_id) as rownum from @collectionExam)
select @question=CE.question_id,@degree=CE.question_degree from collectionExa CE where rownum = @index;
if Exists(select question_id from #collectionExam where question_id = @question)
begin
insert into question_exam values(@examID,@question,@degree);
end
else
begin
raiserror('this question not exist in this course',16,10);
throw 50005,'this question not exist in this course',1;
end
set @index = @index+1;
end
end
commit tran;
drop table #collectionExam;
exec CreatedExam @examID
end
else
begin
raiserror('total degree is not equal sum of question which you inserted it',16,10);
throw 50005,'total degree is not equal sum of question which you inserted it',1;
end
end try
begin catch
rollback tran
select ERROR_MESSAGE();
end

```

```
    end catch  
end  
GO
```

## Used By

No items found

## dbo.Generat\_exam\_random

### Parameters

Name	Data Type	Length	Description
@tableType	table_exam_DT	max	
@questionNo	quest_type_no	max	
@questionDeg	quest_type_degree	max	

### SQL Script

```

SET QUOTED_IDENTIFIER, ANSI_NULLS ON
GO
create OR ALTER proc dbo.Generat_exam_random @tableType table_exam_DT readonly,
    @questionNo quest_type_no readonly,
    @questionDeg quest_type_degree readonly
AS
BEGIN
begin tran
    begin try
        declare @FT int=(select FT_no from @questionNo);
        declare @text int=(select text_no from @questionNo);
        declare @mcq int=(select mcq_no from @questionNo);
        declare @FT_D int=(select FT_degree from @questionDeg);
        declare @text_D int=(select text_degree from @questionDeg);
        declare @mcq_D int=(select mcq_degree from @questionDeg);
        select * into #question_pool_temp from question_pool where course_id in (16)
        --select course_id from @tableType;
        declare @sumTF int,@sumTxt int,@sumMcq int;
        select @sumTxt=COUNT(*) from #question_pool_temp where question_type='text'
        select @sumTF=COUNT(*) from #question_pool_temp where question_type='true & false'
        select @sumMcq=COUNT(*) from #question_pool_temp where question_type='choise'
        declare @maxDegreeCourse int =(select course_maxDegree from course where course_id =(select course_id from @tableType));
        if (@FT*@FT_D+@text*@text_D+@mcq*@mcq_D) = (select total_degree from @tableType)
            begin
                print concat('@FT',@FT,' T' ,@sumTF)
                print concat('@text',@text , ' T',@sumTxt)
                print concat('@mcq',@mcq , ' T',@sumMcq)
                if (@sumMcq>=@mcq and @sumTF>=@FT and @sumTxt>=@text)
                    begin
                        if ((@FT*@FT_D+@text*@text_D+@mcq*@mcq_D) <= @maxDegreeCourse)
                            and (select course_maxDegree from course where course_id =(select course_id from @tableType)) is not null
                            --- and (select * from #question_pool_temp) is not null
                        begin
                            INSERT INTO exam SELECT * FROM @tableType;
                            declare @lastIndexClass int=(SELECT IDENT_CURRENT('exam'))
                            declare @currentQuesId int;
                            while @FT>0
                                begin
                                    declare @ques_id_FT int =(SELECT TOP 1 question_id FROM #question_pool_temp where question_type = 'true & false' ORDER BY
                                    NewId())
                                    insert into dbo.question_exam values
                                    (
                                    (@lastIndexClass),
                                    (@ques_id_FT),
                                    (select FT_degree from @questionDeg)
                                    )
                                    delete #question_pool_temp
                                    where question_id = @ques_id_FT;
                                    set @FT = @FT -1;
                                end
                            while @text>0
                                begin
                                    declare @ques_id_txt int =(SELECT TOP 1 question_id FROM #question_pool_temp where question_type = 'text' ORDER BY
                                    NewId())
                                    insert into dbo.question_exam values
                                    (
                                    (@lastIndexClass),
                                    (@ques_id_txt),
                                    (select text_degree from @questionDeg)
                                    )
                                    delete #question_pool_temp
                                    where question_id = @ques_id_txt;
                                    set @text = @text -1;
                                end
                        end
                    end
                end
            end
        end
    end try
    begin catch
        rollback tran
    end catch
END

```

```
NewId()
insert into dbo.question_exam values
(
    (@lastIndexClass),
    (@ques_id_txt),
    (select FT_degree from @questionDeg)
)
delete #question_pool_temp
where question_id = @ques_id_txt;
set @text = @text-1;
end
while @mcq>0
begin
declare @ques_id_mcq int =(SELECT TOP 1 question_id FROM #question_pool_temp where question_type ='choise' ORDER BY
NewId())
insert into dbo.question_exam values
(
    (@lastIndexClass),
    (@ques_id_mcq),
    (select FT_degree from @questionDeg)
)
delete #question_pool_temp
where question_id = @ques_id_mcq;
set @mcq = @mcq-1;
end
commit ;
exec CreatedExam @lastIndexClass;
end
else
begin
rollback
RAISERROR('The Total Degree For This Exam Exceed Max Degree For This Course',16,10);
end
end
else
begin
RAISERROR('The Number Of Question Error',16,10);
end
end
else
begin
    RAISERROR('sum of degree of question less than total degree',16,10);
end
end try
begin catch
rollback
select ERROR_MESSAGE() as 'ERROR MESSAGE';
end catch
END
GO
```

## Used By

No items found

## dbo.logbackup

### SQL Script

```
SET QUOTED_IDENTIFIER, ANSI_NULLS ON
GO
--#####
create OR ALTER proc dbo.logbackup
as
begin
    BACKUP LOG [examination_sys] TO DISK = 'examination_sys.bak';
end
GO
```

### Used By

No items found

## dbo.Proc\_Course\_Exams

### Parameters

Name	Data Type	Length	Description
@St_id	int	4	
@Crs_id	int	4	

### SQL Script

```

SET QUOTED_IDENTIFIER, ANSI_NULLS ON
GO
create OR ALTER proc dbo.Proc_Course_Exams @St_id int , @Crs_id int
as
begin
declare @st_Crs_E int =(select exam_id
    from instructor_student_exam , course
    where student_id = @St_id AND course_id = @Crs_id AND done=1 )

if Exists (select student_id from course_student where student_id =@St_id AND course_id = @Crs_id)
begin
    if exists(select @st_Crs_E)
begin
    select [Course Name] , [Exam ID] , [Questions] , [Student Answer] , [Model Answer] , exam_result 'Exam
    Result' ,
    final_result 'Course Final Result'
    from [dbo].[Student_Exams] , instructor_student_exam , course_student c
    where exam_id = [Exam ID] And c.course_id = @Crs_id
    And c.student_id = @St_id
end
Else
    raiserror('You not attended any Exam In this Course yet ',16,10);
end
else
    raiserror('YOU Not Attend in Course You Entered ',16,10);
end
GO

```

### Used By

No items found

## dbo.Show\_student\_courses

### Parameters

Name	Data Type	Length	Description
@id	int	4	
@name	varchar	50	

### SQL Script

```
SET QUOTED_IDENTIFIER, ANSI_NULLS ON
GO
create OR ALTER proc dbo.Show_student_courses @id int,@name varchar(50)
as
begin try
    if exists(select * from student where student_id = @id and student_name =@name)
    begin
        select s.student_name,c.course_name,ct.final_result
        from course c,student s,course_student ct
        where ct.student_id =s.student_id
        and c.course_id = ct.course_id
        and s.student_id =@id
        and s.student_name =@name
    end
    else
        raiserror('not exist data for this id and name',16,10);
    end try
    begin catch
        select ERROR_MESSAGE() as 'ERROR MESSAGE';
    end catch
GO
```

### Used By

No items found

## dbo.showExamToAnswer

### Parameters

Name	Data Type	Length	Description
@student_id	int	4	
@exam_id	int	4	
@course_id	int	4	

### SQL Script

```

SET QUOTED_IDENTIFIER, ANSI_NULLS ON
GO
create OR ALTER proc dbo.showExamToAnswer @student_id int ,@exam_id int ,@course_id int
as
begin
declare @Stime time,@Etime time,@Dexam date;
select @Stime=e.exam_startTime,@Etime=e.exam_endTime,@Dexam=e.exam_date
from exam e, course c,instructor_student_exam ise
where e.course_id = c.course_id and ise.exam_id=e.exam_id
and ise.student_id = 1 and ise.exam_id = 10 and c.course_id = 4
if @Stime is null and @Etime is null and @Dexam is null
begin
raiserror('exam not exist',16,10)
end
else
begin
if @Dexam < GETDATE()
begin
select E.* from exam E where E.exam_id = @exam_id;
end
else if @Dexam > GETDATE()
begin
select 'not allow ';
end
else if @Dexam = GETDATE() and @Stime > convert(varchar, getdate(), 8)
begin
select CONCAT('the exam will start in ',@Stime)
end
else if @Dexam = GETDATE() and @Stime < convert(varchar, getdate(), 8) and convert(varchar, getdate(), 8) <
@Etime
begin
select ecd.[Exam Questions],ecd.[Exam Type],ecd.[Question Degree],ecd.[Question Type] from Exam_Content_Details
ecd where [Exam ID] = 10;
end
else if @Dexam = GETDATE() and convert(varchar, getdate(), 8) > @Etime
begin
select 'time out for this exam';
end
end
end
end
end
GO

```

### Used By

No items found

## Functions

### Objects ①



## Table-Valued Functions

### Objects 3

Name	Description
dbo.instruct_exam	
dbo.InstructorCourseByld	Function to View Instructor Name And His Courses
dbo.StudentExamByld	Function to View Student Name And His Exams

## dbo.instruct\_exam

### Parameters

Name	Data Type	Length	Description
@year	date	3	

### SQL Script

```
SET QUOTED_IDENTIFIER, ANSI_NULLS ON
GO
create OR ALTER function dbo.instruct_exam(@year date)
returns table
as
return
(
    select x.exam_id ,x.course_id,x.instruct_id,cl.class_id from dbo.exam x inner join dbo.class_instructor_course
    cl on x.instruct_id=cl.instruct_id and x.course_id=cl.course_id and year(@year)=year(x.exam_date)
)
GO
```

### Used By

No items found

## fx dbo.InstructorCourseById

### Parameters

Name	Data Type	Length	Description
@instruct_id	char	14	

### SQL Script

```
SET QUOTED_IDENTIFIER, ANSI_NULLS ON
GO
create OR ALTER function dbo.InstructorCourseById(@instruct_id char(14))
returns table
as
return(select i.instruct_name , c.course_name from course c,class_instructor_course cic,instructor i
where c.course_id=cic.course_id and i.instruct_id =cic.instruct_id and i.instruct_id = @instruct_id)
GO
```

### Used By

No items found

## dbo.StudentExamById

### Parameters

Name	Data Type	Length	Description
@student_id	int	4	

### SQL Script

```
SET QUOTED_IDENTIFIER, ANSI_NULLS ON
GO
create OR ALTER function dbo.StudentExamById(@student_id int)
returns table
as
return
(
    select s.student_name,e.* ,ise.exam_result,ise.done  from exam e,instructor_student_exam sqe,student s,
instructor_student_exam ise
    where s.student_id=sqe.student_id and e.exam_id =sqe.exam_id and s.student_id = @student_id
    and ise.student_id = @student_id and ise.exam_id = e.exam_id
)
GO
```

### Used By

No items found

## Types

### Objects ①

 [User-Defined Table Types](#)

## User-Defined Table Types

### Objects 5

Name	Description
dbo.assigExamStudent_type	Table data Type Contain student_ID , Instructor_ID and Exam_ID
dbo.quest_type_degree	Table data Type Contain Degree Of every Question type
dbo.quest_type_no	Table data Type Contain Number Of every Question type
dbo.table_exam_DT	Table data Type Contain Exam Info for exam to be created
dbo.tableExamC	Table data Type Contain Question_ID and Question_Degree

## dbo.assigExamStudent\_type

### Columns

Key	Name	Data Type	Length	Precision	Scale	Not Null	Identity	Default	Computed	Persisted	Description
	student_id	int	4	10	0	False			False	False	
	exam_id	int	4	10	0	False			False	False	
	insrtuct_id	char	14	0	0	False			False	False	

### SQL Script

```
CREATE TYPE dbo.assigExamStudent_type AS TABLE (
    student_id int NULL,
    exam_id int NULL,
    insrtuct_id char(14) NULL
)
GO
```

### Used By 1

 [dbo.assignExamForMStudent](#)

## dbo.quest\_type\_degree

### Columns

Key	Name	Data Type	Length	Precision	Scale	Not Null	Identity	Default	Computed	Persisted	Description
	FT_degree	int	4	10	0	False			False	False	
	text_degree	int	4	10	0	False			False	False	
	mcq_degree	int	4	10	0	False			False	False	

### SQL Script

```
CREATE TYPE dbo.quest_type_degree AS TABLE (
    FT_degree int NULL,
    text_degree int NULL,
    mcq_degree int NULL
)
GO
```

### Used By 1

 [dbo.Generat\\_exam\\_random](#)

## dbo.quest\_type\_no

### Columns

Key	Name	Data Type	Length	Precision	Scale	Not Null	Identity	Default	Computed	Persisted	Description
	FT_no	int	4	10	0	False			False	False	
	text_no	int	4	10	0	False			False	False	
	mcq_no	int	4	10	0	False			False	False	

### SQL Script

```
CREATE TYPE dbo.quest_type_no AS TABLE (
    FT_no int NULL,
    text_no int NULL,
    mcq_no int NULL
)
GO
```

### Used By 1

 [dbo.Generat\\_exam\\_random](#)

## dbo.table\_exam\_DT

### Columns

Key	Name	Data Type	Length	Precision	Scale	Not Null	Identity	Default	Computed	Persisted	Description
	startTime	time	5	16	7	False			False	False	
	endTime	time	5	16	7	False			False	False	
	exam_type	varchar	20	0	0	False			False	False	
	exam_date	date	3	10	0	False			False	False	
	total_degree	int	4	10	0	False			False	False	
	course_id	int	4	10	0	False			False	False	
	instruct_id	char	14	0	0	False			False	False	

### SQL Script

```
CREATE TYPE dbo.table_exam_DT AS TABLE (
    startTime time NULL,
    endTime time NULL,
    exam_type varchar(20) NULL,
    exam_date date NULL,
    total_degree int NULL,
    course_id int NULL,
    instruct_id char(14) NULL
)
GO
```

### Used By 2

-  [dbo.Generat\\_exam\\_manaully](#)
-  [dbo.Generat\\_exam\\_random](#)

## dbo.tableExamC

### Columns

Key	Name	Data Type	Length	Precision	Scale	Not Null	Identity	Default	Computed	Persisted	Description
	question_id	int	4	10	0	False			False	False	
	question_degree	int	4	10	0	False			False	False	

### SQL Script

```
CREATE TYPE dbo.tableExamC AS TABLE (
    question_id int NULL,
    question_degree int NULL
)
GO
```

### Used By 1

 dbo.Generat\_exam\_manually

## Rules

### Objects ②

Name	Description
dbo.personal_identity	
dbo.rule_phone	

## dbo.personal\_identity

### SQL Script

```
SET QUOTED_IDENTIFIER OFF
GO
***** Object: Rule [personal_identity] Script Date: 6/5/2022 12:09:09 AM *****/
CREATE RULE dbo.personal_identity AS
len(@id) = 14
GO
```

### Used By

No items found

## dbo.rule\_phone

### SQL Script

```
SET QUOTED_IDENTIFIER OFF
GO
***** Object: Rule [rule_phone] Script Date: 6/5/2022 12:09:09 AM *****/
CREATE RULE dbo.rule_phone AS
len(@id) = 11
GO
```

### Used By

No items found

## 📁 Security

### Objects 3

-  Users
-  Roles
-  Schemas

## Users

### Objects ②

Name	Description
student_sys	
instructor_sys	

## student\_sys

### Database Level Permissions

Type	Action
Grant	CONNECT

### Owned Schemas 1



dbo

### Role Memberships

Name
db_owner
student_role

### SQL Script

```
CREATE USER student_sys  
WITHOUT LOGIN  
GO
```

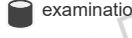
### Used By 3



db\_owner



student\_role



examination\_sys

## instructor\_sys

### Database Level Permissions

Type	Action
Grant	CONNECT

### Owned Schemas 1



dbo

### Role Memberships

Name
db_owner
instructor_role

### SQL Script

```
CREATE USER instructor_sys  
WITHOUT LOGIN  
GO
```

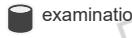
### Used By 3



db\_owner



instructor\_role



examination\_sys

## Roles

### Objects 1

 Database Roles

## Database Roles

### Objects 12

Name	Description
db_accessadmin	
db_backupoperator	
db_datareader	
db_datawriter	
db_ddladmin	
db_denydatareader	
db_denydatawriter	
db_owner	
db_securityadmin	
instructor_role	
public	
student_role	

## db\_accessadmin

### Members

No items found

### SQL Script

```
CREATE ROLE db_accessadmin  
GO
```

### Used By 1



db\_accessadmin

## db\_backupoperator

### Members

No items found

### SQL Script

```
CREATE ROLE db_backupoperator  
GO
```

### Used By 1



db\_backupoperator

## db\_datareader

### Members

No items found

### SQL Script

```
CREATE ROLE db_datareader  
GO
```

### Used By 1

 db\_datareader

## db\_datawriter

### Members

No items found

### SQL Script

```
CREATE ROLE db_datawriter  
GO
```

### Used By 1

 db\_datawriter

## db\_ddladmin

### Members

No items found

### SQL Script

```
CREATE ROLE db_ddladmin  
GO
```

### Used By 1



db\_ddladmin

## db\_denydatareader

### Members

No items found

### SQL Script

```
CREATE ROLE db_denydatareader  
GO
```

### Used By 1

 db\_denydatareader

## db\_denydatawriter

### Members

No items found

### SQL Script

```
CREATE ROLE db_denydatawriter  
GO
```

### Used By 1

 db\_denydatawriter

## db\_owner

### Members ③

dbo  
instructor\_sys  
student\_sys

### SQL Script

```
CREATE ROLE db_owner
GO

EXEC sp_addrolemember N'db_owner', N'dbo'
GO

EXEC sp_addrolemember N'db_owner', N'instructor_sys'
GO

EXEC sp_addrolemember N'db_owner', N'student_sys'
GO
```

### Used By ①

db\_owner

## db\_securityadmin

### Members

No items found

### SQL Script

```
CREATE ROLE db_securityadmin  
GO
```

### Used By 1



db\_securityadmin

## instructor\_role

### Members ①

 instructor\_sys

### SQL Script

```
CREATE ROLE instructor_role  
GO  
  
EXEC sp_addrolemember N'instructor_role', N'instructor_sys'  
GO
```

### Used By

No items found

## public

### Members

No items found

### Database Level Permissions

Type	Action
Grant	VIEW ANY COLUMN ENCRYPTION KEY DEFINITION
Grant	VIEW ANY COLUMN MASTER KEY DEFINITION

### SQL Script

```
CREATE ROLE [public]  
GO
```

### Used By

No items found

## student\_role

### Members ①

 student\_sys

### SQL Script

```
CREATE ROLE student_role
GO

EXEC sp_addrolemember N'student_role', N'student_sys'
GO
```

### Used By

No items found

## Schemas

### Objects 14

Name	Description
audit_admin	
db_accessadmin	
db_backupoperator	
db_datareader	
db_datawriter	
db_ddladmin	
db_denydatareader	
db_denydatawriter	
db_owner	
db_securityadmin	
dbo	
guest	
INFORMATION_SCHEMA	
sys	

## audit\_admin

### SQL Script

```
CREATE SCHEMA audit_admin AUTHORIZATION dbo  
GO
```

### Used By 12

-  [audit\\_admin.audit1\\_class](#)
-  [audit\\_admin.audit2\\_class\\_instruct](#)
-  [audit\\_admin.audit3\\_course](#)
-  [audit\\_admin.audit4\\_course\\_student](#)
-  [audit\\_admin.audit4\\_exam](#)
-  [audit\\_admin.audit4\\_instructor](#)
-  [audit\\_admin.audit5\\_instructor\\_student\\_exam](#)
-  [audit\\_admin.audit6\\_multi\\_choice](#)
-  [audit\\_admin.audit6\\_question\\_exam](#)
-  [audit\\_admin.audit6\\_question\\_pool](#)
-  [audit\\_admin.audit7\\_student](#)
-  [audit\\_admin.audit7\\_student\\_question\\_exam](#)

## db\_accessadmin

### SQL Script

```
CREATE SCHEMA db_accessadmin AUTHORIZATION db_accessadmin  
GO
```

### Used By

No items found

## db\_backupoperator

### SQL Script

```
CREATE SCHEMA db_backupoperator AUTHORIZATION db_backupoperator  
GO
```

### Used By

No items found

## db\_datareader

### SQL Script

```
CREATE SCHEMA db_datareader AUTHORIZATION db_datareader  
GO
```

### Used By

No items found

## db\_datawriter

### SQL Script

```
CREATE SCHEMA db_datawriter AUTHORIZATION db_datawriter  
GO
```

### Used By

No items found

## db\_ddladmin

### SQL Script

```
CREATE SCHEMA db_ddladmin AUTHORIZATION db_ddladmin  
GO
```

### Used By

No items found

## db\_denydatareader

### SQL Script

```
CREATE SCHEMA db_denydatareader AUTHORIZATION db_denydatareader  
GO
```

### Used By

No items found

## db\_denydatawriter

### SQL Script

```
CREATE SCHEMA db_denydatawriter AUTHORIZATION db_denydatawriter  
GO
```

### Used By

No items found

## db\_owner

### SQL Script

```
CREATE SCHEMA db_owner AUTHORIZATION db_owner  
GO
```

### Used By

No items found

## db\_securityadmin

### SQL Script

```
CREATE SCHEMA db_securityadmin AUTHORIZATION db_securityadmin  
GO
```

### Used By

No items found

## dbo

### SQL Script

```
CREATE SCHEMA dbo AUTHORIZATION dbo  
GO
```

### Used By

No items found

## guest

### SQL Script

```
CREATE SCHEMA guest AUTHORIZATION guest  
GO
```

### Used By

No items found

## INFORMATION\_SCHEMA

### SQL Script

```
CREATE SCHEMA INFORMATION_SCHEMA AUTHORIZATION INFORMATION_SCHEMA  
GO
```

### Used By

No items found

## sys

### SQL Script

```
CREATE SCHEMA sys AUTHORIZATION sys  
GO
```

### Used By

No items found