



# Documentation for Examination System

## **Description :**

A desktop application used by students and Instructor that allows the student to know the subjects in which he has registered and their grades and allows them to take exams and correct them automatically based on the model answer stored in the database.

Instructors are allowed to add students, add different questions for each subject, and view the different reports of the system

## **Used Technologies and Tools :**

- Advanced SQL.
- SQL server
- SSRS
- Power BI
- Facebook API

## **Users:**

- Instructor :
  - Add Questions in database for each Course
  - Add Student in database
  - Show Reports for a system


































- Student :
  - Show her/his Grade
  - Take Exam






































## **About the System :**

- Each course contains several questions.
- Exams were created for each course and stored in the Database.
- each exam is generated by chose number of MCQ questions and number of T/F Questions and choose this number randomly form the question table.
- When the student login to the system, he finds her/his courses, and if he has taken his exam, the result will appear to him.
- if the student did not take the exam for any course he can take the exam, the exam will correct automatically in the database and in the next login he find the grade.
- The instructor add new students to the database or add new questions for each course.
- The instructor also can show reports generated by the database.

# Database documentation


## Table of Contents

Table of Contents.....	4
 .....	6
 User databases .....	8
 Examination System Database .....	9
 Tables .....	12
 [dbo].[Choices] .....	13
 [dbo].[Course] .....	15
 [dbo].[Department] .....	17
 [dbo].[Exam] .....	18
 [dbo].[Exam_Question] .....	20
 [dbo].[Ins_course] .....	22
 [dbo].[Instructor] .....	24
 [dbo].[Question] .....	26
 [dbo].[Stud_Course] .....	28
 [dbo].[Student] .....	30
 [dbo].[Student_Exam] .....	32
 [dbo].[Topic] .....	34
 Stored Procedures .....	35
 [dbo].[All_Ins_Data] .....	36
 [dbo].[All_St_Data] .....	37
 [dbo].[All_St_Data_byEmail] .....	38
 [dbo].[All_St_Grade] .....	39
 [dbo].[All_St_Grade_byEmail] .....	40
 [dbo].[all_St_in_Dep#] .....	41
 [dbo].[Delete_Department] .....	42
 [dbo].[Delete_Question] .....	43
 [dbo].[Delete_St_Crs] .....	44
 [dbo].[Exam_by_CrsName] .....	45
 [dbo].[Exam_corr] .....	46
 [dbo].[Exam_for_Course] .....	47
 [dbo].[Exam_Questions] .....	48
 [dbo].[Exam_St_answer] .....	49
 [dbo].[Generat_Exam] .....	51
 [dbo].[Grade_by_Stid_Crs] .....	53

	[dbo].[Ins_Crs_#St] .....	54
	[dbo].[Insert_department] .....	55
	[dbo].[Insert_ins] .....	56
	[dbo].[insert_ins_Crs] .....	57
	[dbo].[insert_Question] .....	58
	[dbo].[Insert_St] .....	60
	[dbo].[insert_St_Crs] .....	61
	[dbo].[Insert_topic] .....	62
	[dbo].[Instructor_Login] .....	63
	[dbo].[St_Exam_Q] .....	64
	[dbo].[Student_Login] .....	65
	[dbo].[TopicForCrs] .....	66
	[dbo].[update_instructor] .....	67
	[dbo].[update_Question] .....	69
	[dbo].[update_Student] .....	70
	[dbo].[update_Topic] .....	72
	Users .....	73
	dbo .....	74
	guest .....	75
	Database Roles .....	76
	db_accessadmin .....	76
	db_backupoperator .....	76
	db_datareader .....	77
	db_datawriter .....	77
	db_ddladmin .....	77
	db_denydatareader .....	78
	db_denydatawriter .....	78
	db_owner .....	78
	db_securityadmin .....	78
	Examination System application .....	80
	Login Form .....	81
	Student Login .....	81
	Exam Form .....	82
	Instructor Login .....	83
	-Insert Student Form .....	84
	Insert Question Form .....	84
	Report Form .....	85



## Databases (1)

-  Examination System

## Server Properties

Property	Value
Product	Microsoft SQL Server
Version	14.0.2042.3
Language	English (United States)
Platform	NT x64
Edition	Developer Edition (64-bit)
Engine Edition	3 (Enterprise)
Processors	12
OS Version	6.3 (19044)
Physical Memory	16228
Is Clustered	False
Root Directory	C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL
Collation	SQL_Latin1_General_CP1_CI_AS

## Server Settings

Property	Value
Default data file path	C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\
Default backup file path	C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\Backup
Default log file path	C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\
Recovery Interval (minutes)	0
Default index fill factor	0
Default backup media retention	0
Compress Backup	False

## Advanced Server Settings


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

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

## **User databases**

### Databases (1)

---

-  Examination System



## Examination System Database

### Database Properties

Property	Value
SQL Server Version	SQL Server 2017
Compatibility Level	SQL Server 2017
Last backup time	10/30/2022
Last log backup time	-
Creation date	Oct 29 2022
Users	4
Database Encryption Enabled	False
Database Encryption Algorithm	None
Database size	16.00 MB
Unallocated space	3.43 MB

### Database Options

Property	Value
Compatibility Level	140
Database collation	SQL_Latin1_General_CP1_CI_AS
Restrict access	MULTI_USER
Is read-only	False
Auto close	False
Auto shrink	False
Database status	ONLINE
In standby	False
Cleanly shutdown	False
Supplemental logging enabled	False
Snapshot isolation state	OFF
Read committed snapshot on	False
Recovery model	FULL
Page verify option	CHECKSUM
Auto create statistics	True
Auto update statistics	True
Auto update statistics asynchronously	False
ANSI NULL default	False
ANSI NULL enabled	False
ANSI padding enabled	False

ANSI warnings enabled	False
Arithmetic abort enabled	False
Concatenating NULL yields NULL	False
Numeric roundabort enabled	False
Quoted Identifier On	False
Recursive triggers enabled	False
Close cursors on commit	False
Local cursors by default	False
Fulltext enabled	True
Trustworthy	False
Database chaining	False
Forced parameterization	False
Master key encrypted by server	False
Published	False
Subscribed	False
Merge published	False
Is distribution database	False
Sync with backup	False
Service broker GUID	039ffe3e-3c99-4a56-b929-015707e05ef5
Service broker enabled	False
Log reuse wait	NOTHING
Date correlation	False
CDC enabled	False
Encrypted	False
Honor broker priority	False
Default language	English
Default fulltext language LCID	1033
Nested triggers enabled	True
Transform noise words	False
Two-digit year cutoff	2049
Containment	NONE
Target recovery time	60
Database owner	DESKTOP-O4872J7\DELL

## Files

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

					System_log.ldf
--	--	--	--	--	----------------

## **Tables**

### Objects

---




Name
dbo.Choices
dbo.Course
dbo.Department
dbo.Exam
dbo.Exam_Question
dbo.Ins_course
dbo.Instructor
dbo.Question
dbo.Stud_Course
dbo.Student
dbo.Student_Exam
dbo.Topic

## [dbo].[Choices]


### Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	380
Created	4:49:08 PM Friday, October 21, 2022
Last Modified	8:25:35 PM Saturday, October 22, 2022

### Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
 	Q_ID	int	4	NOT NULL
	Choice	nvarchar(255)	510	NOT NULL

### Indexes

Key	Name	Key Columns	Unique
	PK_Choices	Q_ID, Choice	True

### Foreign Keys

Name	Columns
FK_Choices_Question	Q_ID->[dbo].[Question].[Q_ID]

### SQL Script

```
CREATE TABLE [dbo].[Choices]
(
  [Q_ID] [int] NOT NULL,
  [Choice] [nvarchar] (255) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Choices] ADD CONSTRAINT [PK Choices] PRIMARY KEY CLUSTERED ([Q_ID], [Choice])
ON [PRIMARY]
GO
ALTER TABLE [dbo].[Choices] ADD CONSTRAINT [FK_Choices_Question] FOREIGN KEY ([Q_ID]) REFERENCES
[dbo].[Question] ([Q_ID])
GO
```

## Uses

---

[dbo].[Question]

## Used By

---

[dbo].[Delete\_Question]

[dbo].[Exam\_for\_Course]



[dbo].[insert\_Question]

## [dbo].[Course]


### Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	8
Created	8:22:39 PM Saturday, October 22, 2022
Last Modified	11:50:58 PM Saturday, October 29, 2022

### Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
	Crs_id	int	4	NOT NULL
	Crs_Name	nvarchar(50)	100	NULL allowed
	Crs_Duration	int	4	NULL allowed
	Top_id	int	4	NULL allowed

### Indexes

Key	Name	Key Columns	Unique
	PK_Course	Crs_id	True

### Foreign Keys

Name	Columns
FK_Course_Topic	Top_id->[dbo].[Topic].[Top_id]

### SQL Script

```
CREATE TABLE [dbo].[Course]
(
    [Crs_id] [int] NOT NULL,
    [Crs_Name] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [Crs_Duration] [int] NULL,
    [Top_id] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Course] ADD CONSTRAINT [PK_Course] PRIMARY KEY CLUSTERED ([Crs_id]) ON [PRIMARY]
GO
```

```
ALTER TABLE [dbo].[Course] ADD CONSTRAINT [FK_Course_Topic] FOREIGN KEY ([Top_id]) REFERENCES  
[dbo].[Topic] ([Top_id])  
GO
```

## Uses

---

[dbo].[Topic]

## Used By

---

[dbo].[Exam]  
[dbo].[Ins\_course]  
[dbo].[Question]  
[dbo].[Stud\_Course]  
[dbo].[All\_St\_Grade]  
[dbo].[All\_St\_Grade\_byEmail]  
[dbo].[Exam\_by\_CrsName]  
[dbo].[Exam\_for\_Course]  
[dbo].[Generat\_Exam]  
[dbo].[Grade\_by\_Stid\_Crs]  
[dbo].[Ins\_Crs\_#St]  
[dbo].[insert\_Question]  
[dbo].[TopicForCrs]




## [dbo].[Department]


### Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	5
Created	8:22:21 PM Saturday, October 22, 2022
Last Modified	8:25:48 PM Saturday, October 22, 2022

### Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
	Dept_id	int	4	NOT NULL
	Dept_Name	nvarchar(50)	100	NOT NULL

### Indexes

Key	Name	Key Columns	Unique
	PK_Department	Dept_id	True

### SQL Script

```
CREATE TABLE [dbo].[Department]
(
    [Dept_id] [int] NOT NULL,
    [Dept_Name] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Department] ADD CONSTRAINT [PK_Department] PRIMARY KEY CLUSTERED ([Dept_id])
ON [PRIMARY]
GO
```

### Used By



[dbo].[Instructor]  
[dbo].[Student]  
[dbo].[all\_St\_in\_Dep#]  
[dbo].[Delete\_Department]  
[dbo].[Insert\_department]

## [dbo].[Exam]


### Properties

Property	Value
Row Count (~)	10
Created	8:23:51 PM Saturday, October 22, 2022
Last Modified	8:23:51 PM Saturday, October 22, 2022

### Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
	Ex_id	int	4	NOT NULL
	No_TF	int	4	NULL allowed
	No_MCQ	int	4	NULL allowed
	Crs_id	int	4	NULL allowed

### Indexes

Key	Name	Key Columns	Unique
	PK_Exam	Ex_id	True

### Foreign Keys

Name	Columns
FK_Exam_Course	Crs_id->[dbo].[Course].[Crs_id]

### SQL Script

```
CREATE TABLE [dbo].[Exam]
(
    [Ex_id] [int] NOT NULL,
    [No_TF] [int] NULL,
    [No_MCQ] [int] NULL,
    [Crs_id] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Exam] ADD CONSTRAINT [PK_Exam] PRIMARY KEY CLUSTERED ([Ex_id]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Exam] ADD CONSTRAINT [FK_Exam_Course] FOREIGN KEY ([Crs_id]) REFERENCES
[dbo].[Course] ([Crs_id])
```

GO

## Uses

---

[dbo].[Course]

## Used By

---

[dbo].[Exam\_Question]

[dbo].[Student\_Exam]

[dbo].[Exam\_by\_CrsName]

[dbo].[Exam\_corr]

[dbo].[Exam\_for\_Course]

[dbo].[Exam\_Questions]

[dbo].[Exam\_St\_answer]





[dbo].[Generat\_Exam]

## [dbo].[Exam\_Question]


### Properties

Property	Value
Row Count (~)	64
Created	7:06:23 PM Tuesday, October 18, 2022
Last Modified	8:25:35 PM Saturday, October 22, 2022

### Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
 	Ex_id	int	4	NOT NULL
 	Ques_id	int	4	NOT NULL

### Indexes

Key	Name	Key Columns	Unique
	PK_Exam_Question	Ex_id, Ques_id	True

### Foreign Keys

Name	Columns
FK_Exam_Question_Exam	Ex_id->[dbo].[Exam].[Ex_id]
FK_Exam_Question_Question	Ques_id->[dbo].[Question].[Q_ID]

### SQL Script

```
CREATE TABLE [dbo].[Exam_Question]
(
  [Ex_id] [int] NOT NULL,
  [Ques_id] [int] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Exam_Question] ADD CONSTRAINT [PK_Exam_Question] PRIMARY KEY CLUSTERED
([Ex_id], [Ques_id]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Exam_Question] ADD CONSTRAINT [FK_Exam_Question_Exam] FOREIGN KEY ([Ex_id])
REFERENCES [dbo].[Exam] ([Ex_id])
GO
ALTER TABLE [dbo].[Exam_Question] ADD CONSTRAINT [FK_Exam_Question_Question] FOREIGN KEY
([Ques_id]) REFERENCES [dbo].[Question] ([Q_ID])
```

GO

### Uses

---

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

### Used By

---





[dbo].[Exam\_for\_Course]  
[dbo].[Exam\_Questions]  
[dbo].[Generat\_Exam]

## [dbo].[Ins\_course]



### Properties

Property	Value
Row Count (~)	23
Created	11:03:10 PM Friday, October 21, 2022
Last Modified	8:25:48 PM Saturday, October 22, 2022

### Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
  C	Ins_id	int	4	NOT NULL
  C	Crs_id	int	4	NOT NULL

### Indexes

Key	Name	Key Columns	Unique
 	PK_Ins_course	Ins_id, Crs_id	True

### Foreign Keys

Name	Columns
FK_Ins_Course_Course	Crs_id->[dbo].[Course].[Crs_id]
FK_Ins_Course_Instructor	Ins_id->[dbo].[Instructor].[Ins_id]

### SQL Script

```
CREATE TABLE [dbo].[Ins_course]
(
  [Ins_id] [int] NOT NULL,
  [Crs_id] [int] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Ins_course] ADD CONSTRAINT [PK_Ins_course] PRIMARY KEY CLUSTERED ([Ins_id],
[Crs_id]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Ins_course] ADD CONSTRAINT [FK_Ins_Course_Course] FOREIGN KEY ([Crs_id])
REFERENCES [dbo].[Course] ([Crs_id])
GO
ALTER TABLE [dbo].[Ins_course] ADD CONSTRAINT [FK_Ins_Course_Instructor] FOREIGN KEY ([Ins_id])
REFERENCES [dbo].[Instructor] ([Ins_id])
```

GO

### Uses

---

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

### Used By

---



[dbo].[Ins\_Crs\_#St]  
[dbo].[insert\_ins\_Crs]

## [dbo].[Instructor]


### Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	22
Created	8:25:48 PM Saturday, October 22, 2022
Last Modified	8:25:48 PM Saturday, October 22, 2022

### Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
	Ins_id	int	4	NOT NULL
	Ins_FName	nvarchar(50)	100	NULL allowed
	Ins_LName	nvarchar(50)	100	NULL allowed
	Ins_Phone	int	4	NULL allowed
	Ins_Email	nvarchar(50)	100	NULL allowed
	Ins_Address	nvarchar(50)	100	NULL allowed
	Dept_id	int	4	NULL allowed
	Ins_Password	int	4	NULL allowed
	Ins_Age	int	4	NULL allowed

### Indexes

Key	Name	Key Columns	Unique
	PK_Instructor	Ins_id	True

### Foreign Keys

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

### SQL Script

```
CREATE TABLE [dbo].[Instructor]
(
  [Ins_id] [int] NOT NULL,
  [Ins_FName] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
```



```

[Ins_LName] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[Ins_Phone] [int] NULL,
[Ins_Email] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[Ins_Address] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[Dept_id] [int] NULL,
[Ins_Password] [int] NULL,
[Ins_Age] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Instructor] ADD CONSTRAINT [PK_Instructor] PRIMARY KEY CLUSTERED ([Ins_id]) ON
[PRIMARY]
GO
ALTER TABLE [dbo].[Instructor] ADD CONSTRAINT [FK_Instructor_Department] FOREIGN KEY ([Dept_id])
REFERENCES [dbo].[Department] ([Dept_id])
GO

```

## Uses

---

[dbo].[Department]

## Used By

---



[dbo].[Ins\_course]  
[dbo].[All\_Ins\_Data]  
[dbo].[Ins\_Crs\_#St]  
[dbo].[Insert\_ins]  
[dbo].[Instructor\_Login]  
[dbo].[update\_instructor]

## [dbo].[Question]


### Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	117
Created	8:25:34 PM Saturday, October 22, 2022
Last Modified	6:59:16 PM Sunday, October 23, 2022

### Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
	Q_ID	int	4	NOT NULL
	Question_Bady	nvarchar(max)	max	NULL allowed
	Q_Type	nvarchar(50)	100	NULL allowed
	Model_Answer	nvarchar(max)	max	NULL allowed
	Point	int	4	NULL allowed
	Crs_ID	int	4	NULL allowed

### Indexes

Key	Name	Key Columns	Unique
	PK_Question	Q_ID	True

### Foreign Keys

Name	Columns
FK_Question_Course	Crs_ID->[dbo].[Course].[Crs_id]

### SQL Script

```
CREATE TABLE [dbo].[Question]
(
    [Q_ID] [int] NOT NULL,
    [Question_Bady] [nvarchar] (max) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [Q_Type] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [Model_Answer] [nvarchar] (max) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [Point] [int] NULL,
    [Crs_ID] [int] NULL
)
```

```
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Question] ADD CONSTRAINT [PK_Question] PRIMARY KEY CLUSTERED ([Q_ID]) ON
[PRIMARY]
GO
ALTER TABLE [dbo].[Question] ADD CONSTRAINT [FK_Question_Course] FOREIGN KEY ([CrS_ID])
REFERENCES [dbo].[Course] ([CrS_id])
GO
```

## Uses

---

[dbo].[Course]

## Used By

---



[dbo].[Choices]  
[dbo].[Exam\_Question]  
[dbo].[Student\_Exam]  
[dbo].[Delete\_Question]  
[dbo].[Exam\_corr]  
[dbo].[Exam\_for\_Course]  
[dbo].[Exam\_Questions]  
[dbo].[Generat\_Exam]  
[dbo].[insert\_Question]  
[dbo].[St\_Exam\_Q]  
[dbo].[update\_Question]

## [dbo].[Stud\_Course]


### Properties

Property	Value
Row Count (~)	73
Created	11:50:58 PM Saturday, October 29, 2022
Last Modified	11:50:58 PM Saturday, October 29, 2022

### Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
	St_id	int	4	NOT NULL
	Crs_id	int	4	NOT NULL
	Crs_Grade	int	4	NULL allowed

### Indexes

Key	Name	Key Columns	Unique
	PK_Stud_Course	St_id, Crs_id	True

### Foreign Keys

Name	Columns
FK_Stud_Course_Course	Crs_id->[dbo].[Course].[Crs_id]
FK_Stud_Course_Student	St_id->[dbo].[Student].[St_id]

### SQL Script

```
CREATE TABLE [dbo].[Stud_Course]
(
    [St_id] [int] NOT NULL,
    [Crs_id] [int] NOT NULL,
    [Crs_Grade] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Stud_Course] ADD CONSTRAINT [PK_Stud_Course] PRIMARY KEY CLUSTERED ([St_id],
[Crs_id]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Stud_Course] ADD CONSTRAINT [FK_Stud_Course_Course] FOREIGN KEY ([Crs_id])
REFERENCES [dbo].[Course] ([Crs_id])
```

```
GO
ALTER TABLE [dbo].[Stud_Course] ADD CONSTRAINT [FK_Stud_Course_Student] FOREIGN KEY ([St_id])
REFERENCES [dbo].[Student] ([St_id])
GO
```

## Uses

---

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

## Used By

---



[dbo].[All\_St\_Grade]  
[dbo].[All\_St\_Grade\_byEmail]  
[dbo].[Delete\_St\_Crs]  
[dbo].[Exam\_corr]  
[dbo].[Grade\_by\_Stid\_Crs]  
[dbo].[Ins\_Crs\_#St]  
[dbo].[insert\_St\_Crs]

## [dbo].[Student]


### Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	113
Created	8:25:02 PM Saturday, October 22, 2022
Last Modified	11:50:58 PM Saturday, October 29, 2022

### Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
	St_id	int	4	NOT NULL
	St_FName	nvarchar(50)	100	NULL allowed
	St_LName	nvarchar(50)	100	NULL allowed
	St_Email	nvarchar(50)	100	NULL allowed
	St_Address	nvarchar(50)	100	NULL allowed
	St_Age	int	4	NULL allowed
	St_Phone	int	4	NULL allowed
	Dept_id	int	4	NULL allowed
	St_Gender	nvarchar(50)	100	NULL allowed
	St_Password	int	4	NULL allowed
	St_DOB	date	3	NULL allowed

### Indexes

Key	Name	Key Columns	Unique
	PK_Student	St_id	True

### Foreign Keys

Name	Columns
FK_Student_Department	Dept_id->[dbo].[Department].[Dept_id]

### SQL Script

```
CREATE TABLE [dbo].[Student]
```

```

(
[St_id] [int] NOT NULL,
[St_FName] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[St_LName] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[St_Email] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[St_Address] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[St_Age] [int] NULL,
[St_Phone] [int] NULL,
[Dept_id] [int] NULL,
[St_Gender] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[St_Password] [int] NULL,
[St_DOB] [date] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Student] ADD CONSTRAINT [PK_Student] PRIMARY KEY CLUSTERED ([St_id]) ON
[PRIMARY]
GO
ALTER TABLE [dbo].[Student] ADD CONSTRAINT [FK_Student_Department] FOREIGN KEY ([Dept_id])
REFERENCES [dbo].[Department] ([Dept_id])
GO

```

## Uses

---

[dbo].[Department]

## Used By

---







[dbo].[Stud\_Course]  
 [dbo].[Student\_Exam]  
 [dbo].[All\_St\_Data]  
 [dbo].[All\_St\_Data\_byEmail]  
 [dbo].[All\_St\_Grade]  
 [dbo].[All\_St\_Grade\_byEmail]  
 [dbo].[all\_St\_in\_Dep#]  
 [dbo].[Exam\_St\_answer]  
 [dbo].[Insert\_St]  
 [dbo].[Student\_Login]  
 [dbo].[update\_Student]

## [dbo].[Student\_Exam]


### Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	12
Created	7:10:13 PM Tuesday, October 18, 2022
Last Modified	8:25:35 PM Saturday, October 22, 2022

### Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
 	St_id	int	4	NOT NULL
 	Ex_id	int	4	NOT NULL
 	Ques_id	int	4	NOT NULL
	Grade	int	4	NULL allowed
	Date	date	3	NULL allowed
	Time	time	5	NULL allowed
	St_Answer	nvarchar(max)	max	NULL allowed

### Indexes

Key	Name	Key Columns	Unique
	PK_Student_Exam	St_id, Ex_id, Ques_id	True

### Foreign Keys

Name	Columns
FK_Student_Exam_Exam	Ex_id->[dbo].[Exam].[Ex_id]
FK_Student_Exam_Question	Ques_id->[dbo].[Question].[Q_ID]
FK_Student_Exam_Student	St_id->[dbo].[Student].[St_id]

### SQL Script

```
CREATE TABLE [dbo].[Student_Exam]
(
  [St_id] [int] NOT NULL,
  [Ex_id] [int] NOT NULL,
```



```

[Ques_id] [int] NOT NULL,
[Grade] [int] NULL,
[Date] [date] NULL,
[Time] [time] NULL,
[St_Answer] [nvarchar] (max) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Student_Exam] ADD CONSTRAINT [PK_Student_Exam] PRIMARY KEY CLUSTERED ([St_id],
[Ex_id], [Ques_id]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Student_Exam] ADD CONSTRAINT [FK_Student_Exam_Exam] FOREIGN KEY ([Ex_id])
REFERENCES [dbo].[Exam] ([Ex_id])
GO
ALTER TABLE [dbo].[Student_Exam] ADD CONSTRAINT [FK_Student_Exam_Question] FOREIGN KEY
([Ques_id]) REFERENCES [dbo].[Question] ([Q_ID])
GO
ALTER TABLE [dbo].[Student_Exam] ADD CONSTRAINT [FK_Student_Exam_Student] FOREIGN KEY ([St_id])
REFERENCES [dbo].[Student] ([St_id])
GO

```

## Uses

---

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

## Used By

---


[dbo].[Exam\_corr]  
[dbo].[Exam\_St\_answer]  
[dbo].[St\_Exam\_Q]

## [dbo].[Topic]

### Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	5
Created	8:24:15 PM Saturday, October 22, 2022
Last Modified	8:24:15 PM Saturday, October 22, 2022

### Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
	Top_id	int	4	NOT NULL
	Top_Name	nvarchar(50)	100	NULL allowed

### Indexes

Key	Name	Key Columns	Unique
	PK_Topic	Top_id	True

### SQL Script

```
CREATE TABLE [dbo].[Topic]
(
  [Top_id] [int] NOT NULL,
  [Top_Name] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Topic] ADD CONSTRAINT [PK_Topic] PRIMARY KEY CLUSTERED ([Top_id]) ON [PRIMARY]
GO
```

### Used By

[dbo].[Course]  
[dbo].[Insert\_topic]  
[dbo].[TopicForCrS]  
[dbo].[update\_Topic]

## **Stored Procedures**

### Objects

---

Name
dbo.All_Ins_Data
dbo.All_St_Data
dbo.All_St_Data_byEmail
dbo.All_St_Grade
dbo.All_St_Grade_byEmail
dbo.all_St_in_Dep#
dbo.Delete_Department
dbo.Delete_Question
dbo.Delete_St_Crs
dbo.Exam_by_CrsName
dbo.Exam_corr
dbo.Exam_for_Course
dbo.Exam_Questions
dbo.Exam_St_answer
dbo.Generat_Exam
dbo.Grade_by_Stid_Crs
dbo.Ins_Crs_#St
dbo.Insert_department
dbo.Insert_ins
dbo.insert_ins_Crs
dbo.insert_Question
dbo.Insert_St
dbo.insert_St_Crs
dbo.Insert_topic
dbo.Instructor_Login
dbo.St_Exam_Q
dbo.Student_Login
dbo.TopicForCrs
dbo.update_instructor
dbo.update_Question
dbo.update_Student
dbo.update_Topic

## [dbo].[All\_Ins\_Data]

### Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

### Parameters

Name	Data Type	Max Length (Bytes)
@Ins_Email	nvarchar(max)	max

### SQL Script

```
CREATE PROC [dbo].[All_Ins_Data] @Ins_Email NVARCHAR(MAX)
AS
BEGIN
    IF EXISTS(SELECT Ins_Email FROM dbo.Instructor WHERE Ins_Email = @Ins_Email)
        BEGIN
            SELECT * FROM dbo.Instructor
            WHERE Ins_Email = @Ins_Email
        END
    ELSE
        SELECT 'this Instructor does not exist'
END
GO
```

### Uses

[dbo].[Instructor]

## [dbo].[All\_St\_Data]

### Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

### Parameters

Name	Data Type	Max Length (Bytes)
@St_Email	nvarchar(max)	max

### SQL Script

```
CREATE proc [dbo].[All_St_Data] @St_Email nvarchar(max)
as
begin
    if exists(select St_Email from Student where St_Email = @St_Email)
        begin
            select * from Student
            where St_Email = @St_Email
        end
    else
        select 'this Student does not exist'
end
GO
```

### Uses

[dbo].[Student]

## [dbo].[All\_St\_Data\_byEmail]

### Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

### Parameters

Name	Data Type	Max Length (Bytes)
@St_email	nvarchar(50)	100

### SQL Script

```
CREATE proc [dbo].[All_St_Data_byEmail] @St_email nvarchar(50)
as
begin
    if exists(select St_Email from Student where St_Email = @St_Email)
        begin
            select * from Student
            where St_Email = @St_Email
        end
    else
        select 'this Student does not exist'
end
GO
```

### Uses

[dbo].[Student]

## [dbo].[All\_St\_Grade]

### Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

### Parameters

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

### SQL Script

```
CREATE proc [dbo].[All_St_Grade] @St_id int
as
begin
    if exists(select St_id from Student where St_id = @St_id)
        begin
            select sc.St_id , s.St_FName, s.St_Lname ,c.Crs_Name ,sc.Crs_Grade from Stud_Course
sc ,
            Student s ,Course c where sc.St_id = s.St_id and c.Crs id = sc.Crs id and sc.St id
= @St_id
        end
    else
        select 'this Student does not exist'
    end
GO
```

### Uses

[dbo].[Course]  
[dbo].[Stud\_Course]  
[dbo].[Student]

## [dbo].[All\_St\_Grade\_byEmail]

### Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

### Parameters

Name	Data Type	Max Length (Bytes)
@St_Email	nvarchar(50)	100


### SQL Script

```
CREATE proc [dbo].[All_St_Grade_byEmail] @St_Email nvarchar(50)
as
begin
    if exists(select St_Email from Student where St_Email = @St_Email)
        begin
            select sc.St_id , s.St_FName, s.St_Lname ,c.Crs_Name ,sc.Crs_Grade from Stud_Course
sc ,
            Student s ,Course c where sc.St_id = s.St_id and c.Crs id = sc.Crs id and s.St -
Email = @St_Email
        end
    else
        select 'this Student does not exist'
    end
GO
```

### Uses

[dbo].[Course]  
[dbo].[Stud\_Course]  
[dbo].[Student]



 [dbo].[all\_St\_in\_Dep#]

## Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

## Parameters

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

## SQL Script

```
create proc [dbo].[all_St_in_Dep#] @Dep_id int
as
begin
    if exists(select Dept_id from Department where Dept_id = @Dep_id)
        begin
            select * from Student where Dept_id = @Dep_id
        end
    else
        select 'this department does not exist'
end
GO
```

## Uses

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

## [dbo].[Delete\_Department]

### Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

### Parameters

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

### SQL Script

```
create proc [dbo].[Delete_Department] @Dept_Id int
as
begin try
    if exists(select Dept_id from Department where Dept_id = @Dept_Id )
        begin
            delete from Department
            where Dept_id = @Dept_Id
        end
    else
        select 'this Department doe not exist'
end try
begin catch
    select ERROR_MESSAGE()
end catch
GO
```

### Uses

[dbo].[Department]

## [dbo].[Delete\_Question]

### Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

### Parameters

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

### SQL Script

```
create proc [dbo].[Delete_Question] @Q_Id int
as
begin try
    if exists(select Q_ID from Question where Q_ID = @Q_Id )
        begin
            delete from Choices
            where Q_ID = @Q_Id

            delete from Question
            where Q_ID = @Q_Id
        end
    else
        select 'this Question doe not exist'
end try
begin catch
    select ERROR_MESSAGE()
end catch
GO
```

### Uses

[dbo].[Choices]  
[dbo].[Question]

## [dbo].[Delete\_St\_Crs]

### Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

### Parameters

Name	Data Type	Max Length (Bytes)
@St_Id	int	4
@Crs_id	int	4

### SQL Script

```
CREATE proc [dbo].[Delete_St_Crs] @St_Id int , @Crs_id int
as
begin try
    if exists(select St_id , Crs_id from Stud_Course where St_id = @St_Id and Crs_id =@Crs_id)
        begin
            delete from Stud_Course
            where St_id = @St_Id and Crs_id =@Crs_id
        end
    else
        select 'this course & this student doe not exist'
end try
begin catch
    select 'error'
end catch
GO
```

### Uses

[dbo].[Stud\_Course]

## [dbo].[Exam\_by\_CrsName]

### Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

### Parameters

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

### SQL Script

```
CREATE PROC [dbo].[Exam_by_CrsName] @CrS_name VARCHAR(50)
AS
BEGIN
    IF EXISTS(SELECT Crs_Name FROM dbo.Course WHERE Crs_Name = @CrS_name)
        BEGIN
            DECLARE @CrS_id int = (SELECT Crs_id FROM Course WHERE Crs_Name=@CrS_name)

            SELECT TOP(1) [Ex_id]
            FROM dbo.Exam
            WHERE Crs_id = @CrS_id

        END
    ELSE
        SELECT 'this Exam does not exist'
END
GO
```

### Uses

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

## [dbo].[Exam\_corr]

### Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

### Parameters

Name	Data Type	Max Length (Bytes)
@St_id	int	4
@Ex_id	int	4

### SQL Script

```
CREATE proc [dbo].[Exam_corr] @St_id int , @Ex_id int
as
begin
    declare @No_Q int = (select count(Ques_id) from Student_Exam WHERE St_id = @St_id and Ex_id =
@Ex_id )

    declare @Right_Q float = (select count(s.Ques_id) from Student_Exam s join Question q on
s.Ques_id=q.Q_ID
    where s.St_id = 1 and s.Ex_id = 1 and s.St_Answer = q.Model_Answer )

    declare @Grade float
    select @Grade = (@Right_Q/@No_Q) *100

    update Stud_Course
    set [Crs_Grade] = ( str(@Grade) + '%' )
    where St_id= @St_id and Crs_id = (select Crs_id from Exam where Ex_id = @Ex_id)

end
GO
```

### Uses

[dbo].[Exam]  
[dbo].[Question]  
[dbo].[Stud\_Course]  
[dbo].[Student\_Exam]

## [dbo].[Exam\_for\_Course]

### Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

### Parameters

Name	Data Type	Max Length (Bytes)
@CrS_name	nvarchar(50)	100

### SQL Script

```
CREATE PROC [dbo].[Exam_for_Course] @CrS_name NVARCHAR(50)
AS
BEGIN
    IF EXISTS(SELECT Crs_Name FROM Course WHERE Crs_Name = @CrS_name)
    BEGIN
        DECLARE @CrS_id INT = (SELECT Crs_id FROM dbo.Course WHERE Crs_Name = @CrS_name )
        DECLARE @Ex_id INT = (SELECT TOP(1) Ex_id FROM dbo.Exam WHERE Crs_id = @CrS_id )

        SELECT q.Q_ID , e.Ex_id , q.Question_Body , q.Q_Type , q.Point , q.Model_Answer ,
c.Choice
        FROM Question q , Exam_Question eq , Exam e , dbo.Choices c

        WHERE q.Q_ID = eq.Ques_id AND e.Ex_id = eq.Ex_id AND q.Q_ID = c.Q_ID
            AND e.Crs_id = @CrS_id AND e.Ex_id = @Ex_id

    END
    ELSE
        SELECT 'this Exam does not exist'
END
GO
```

### Uses

[dbo].[Choices]  
[dbo].[Course]  
[dbo].[Exam]  
[dbo].[Exam\_Question]  
[dbo].[Question]

## [dbo].[Exam\_Questions]

### Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

### Parameters

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

### SQL Script

```
create proc [dbo].[Exam_Questions]  @Ex_id int
as
begin
    if exists(select Ex_id from Exam where Ex_id = @Ex_id)
        begin
            select q.Question_Bady from Question q , Exam_Question eq where q.Q_ID = eq.Ques_id
and Ex_id = @Ex_id
        end
    else
        select 'this Exam does not exist'
end
GO
```

### Uses

[dbo].[Exam]  
[dbo].[Exam\_Question]  
[dbo].[Question]



## [dbo].[Exam\_St\_answer]

### Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

### Parameters

Name	Data Type	Max Length (Bytes)
@St_id	int	4
@Ex_id	int	4
@Q_id	int	4
@St_Answer	varchar(50)	50

### SQL Script

```
CREATE PROC [dbo].[Exam_St_answer] @St_id INT , @Ex_id INT , @Q_id INT , @St_Answer VARCHAR(50)
AS
BEGIN
    IF EXISTS (SELECT St_id FROM Student WHERE St_id = @St_id)
        BEGIN
            IF EXISTS(SELECT Ex_id FROM Exam WHERE Ex_id=@Ex_id)
                BEGIN

                    INSERT INTO Student_Exam ([St_id],[Ex_id],[Ques_id],[Date],[St_Answer])
                    VALUES (@St_id , @Ex_id , @Q_id , GETDATE() , @St_Answer )

                END
            ELSE
                SELECT 'this Exam does not exist'
        END
    ELSE
        SELECT 'this student ID does not exist'
END
GO
```

### Uses

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

[dbo].[Student\_Exam]

## [dbo].[Generat\_Exam]

### Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

### Parameters

Name	Data Type	Max Length (Bytes)
@Exam_id	int	4
@Course_name	nvarchar(50)	100
@No_TF	int	4
@No_MCQ	int	4

### SQL Script

```
CREATE proc [dbo].[Generat_Exam] @Exam_id int , @Course_name nvarchar(50) , @No_TF int , @No_MCQ
int
as
begin
    if exists(select Ex_id from Exam where Ex_id = @Exam_id)
        select 'this exam id already exast '
    else
        begin
            if exists (select Crs_Name from Course where Crs_Name = @Course_name)
                begin
                    declare @Crs id int = (select Crs id from Course where Crs Name =
@Course_name)

                    insert into Exam ([Ex_id],[No_TF],[No_MCQ],[Crs_id])
                    values (@Exam_id , @No_TF , @No_MCQ , @Crs_id )

                    insert into Exam_Question (Ex_id , Ques_id )
                    select top(@No_TF) @Exam_id , Q_id
                    from Question
                    where Q_Type = 'TF' and Crs_ID = @Crs_id
                    order by newid()

                    insert into Exam_Question (Ex_id , Ques_id )
                    select top(@No_MCQ) @Exam_id , Q_id
                    from Question
                    where Q_Type = 'MCQ' and Crs_ID = @Crs_id
                    order by newid()
```

```
        end
    else
        select 'this course is not exist'
    end
end
GO
```

## Uses

---

[dbo].[Course]  
[dbo].[Exam]  
[dbo].[Exam\_Question]  
[dbo].[Question]

## [dbo].[Grade\_by\_Stid\_Crs]

### Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

### Parameters

Name	Data Type	Max Length (Bytes)
@St_id	int	4
@Crs_name	nvarchar(50)	100

### SQL Script

```
CREATE PROC [dbo].[Grade_by_Stid_Crs] @St_id INT , @Crs_name NVARCHAR(50)
AS
BEGIN
    DECLARE @Crs_id INT = (SELECT Crs_id FROM dbo.Course WHERE Crs_Name = @Crs_name )

    SELECT Crs_Grade FROM dbo.Stud_Course WHERE Crs_id = @Crs_id AND St_id = @St_id

END
GO
```

### Uses

[dbo].[Course]  
[dbo].[Stud\_Course]

## [dbo].[Ins\_Crs\_#St]

### Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

### Parameters

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

### SQL Script

```
CREATE proc [dbo].[Ins_Crs_#St] @ins_id int
as
begin
    if exists(select Ins_id from Instructor where Ins_id = @ins_id)
        begin
            select c.Crs_Name , COUNT(sc.St_id) NumberOfSt from Course c , Stud_Course sc ,
Ins_course ic
            where C.Crs_id = SC.Crs_id
            and c.Crs_id=ic.Crs_id and Ins_id = @ins_id
            group by c.Crs_Name
        end
    else
        select 'this instructor does not exist'
end
GO
```

### Uses

[dbo].[Course]  
[dbo].[Ins\_course]  
[dbo].[Instructor]  
[dbo].[Stud\_Course]

## [dbo].[Insert\_department]

### Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

### Parameters

Name	Data Type	Max Length (Bytes)
@dept_id	int	4
@dept_Name	nvarchar(50)	100

### SQL Script

```
create proc [dbo].[Insert_department] @dept_id int , @dept_Name nvarchar(50)
as
begin
    if not exists (select dept_id from Department where dept_id = @dept_id )
        begin
            insert into Department([dept_id],[dept_Name])
            values (@dept_id , @dept_Name)
        end
    else
        select 'this Department already exists'
    end
end
GO
```

### Uses

[dbo].[Department]

## [dbo].[Insert\_ins]

### Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

### Parameters

Name	Data Type	Max Length (Bytes)
@id	int	4
@fname	nvarchar(50)	100
@lname	nvarchar(50)	100
@dept_id	int	4
@email	nvarchar(50)	100
@password	int	4

### SQL Script

```
CREATE proc [dbo].[Insert_ins] @id int , @fname nvarchar(50) , @lname nvarchar(50) , @dept_id int
,@email nvarchar(50) , @password int
as
    begin try
        insert into Instructor ([Ins_id],[Ins_FName],[Ins_LName],[Dept_id],[Ins_Email],[Ins_
Password])
        values (@id , @fname , @lname , @dept_id , @email , @password)
    end try
    begin catch
        select 'error'
    end catch
GO
```

### Uses

[dbo].[Instructor]



## [dbo].[insert\_ins\_Crs]

### Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

### Parameters

Name	Data Type	Max Length (Bytes)
@ins_Id	int	4
@Crs_id	int	4

### SQL Script

```
create proc [dbo].[insert_ins_Crs] @ins_Id int , @Crs_id int
as
begin try
    if exists(select ins_id , Crs_id from Ins_course where ins_id = @ins_Id and Crs_id =@Crs_id)
        begin
            select 'this course is already assigned to this instructor '
        end
    else
        insert into ins_Course (ins_id , Crs_id)
        values (@ins_Id , @Crs_id)
end try
begin catch
    select 'error'
end catch
GO
```

### Uses

[dbo].[Ins\_course]

## [dbo].[insert\_Question]

### Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

### Parameters

Name	Data Type	Max Length (Bytes)
@Q_id	int	4
@Q_Type	nvarchar(50)	100
@Q_body	nvarchar(max)	max
@Crs_id	int	4
@Model_Answer	nvarchar(max)	max
@choice_1	nvarchar(max)	max
@choice_2	nvarchar(max)	max
@choice_3	nvarchar(max)	max
@choice_4	nvarchar(max)	max

### SQL Script

```
CREATE proc [dbo].[insert_Question] @Q_id int ,@Q_Type nvarchar(50) , @Q_body nvarchar(max) ,
@Crs_id int , @Model_Answer nvarchar(max) , @choice_1 nvarchar(max), @choice_2 nvarchar(max),
@choice_3 nvarchar(max),@choice_4 nvarchar(max)

as
begin
    if exists (select Crs_id from Course where Crs_id = @Crs_id )
        begin
            if not exists (select Q_ID from Question where Q_ID = @Q_id)
                begin
                    insert into Question ([Q_ID],[Question_Body],[Q_Type] , [Model_Answer],[Crs_
ID])
                    values(@Q_id , @Q_body ,@Q_Type , @Model_Answer , @Crs_id )

                    if @Q_Type = 'TF'
                        begin
                            insert into Choices ([Q_ID],[Choice])
                            values (@Q_id,@choice_1),(@Q_id,@choice_2)
                        end
                    else

```

```
begin
    insert into Choices ([Q_ID],[Choice])
    values
(@Q_id,@choice_1), (@Q_id,@choice_2), (@Q_id,@choice_3), (@Q_id,@choice_4)

end

else
    select 'this Question id already exist'
end

else
    select 'this course doen not exist'
end
GO
```

## Uses

---

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

## [dbo].[Insert\_St]

### Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

### Parameters

Name	Data Type	Max Length (Bytes)
@id	int	4
@fname	nvarchar(50)	100
@lname	nvarchar(50)	100
@dept_id	int	4
@email	nvarchar(50)	100
@password	int	4

### SQL Script

```
CREATE proc [dbo].[Insert_St] @id int , @fname nvarchar(50) , @lname nvarchar(50) , @dept_id int
,@email nvarchar(50) , @password int
as
    begin try
        insert into Student ([St_id],[St_FName],[St_LName],[Dept_id],[St_Email],[St_Password])
        values (@id , @fname , @lname , @dept_id ,@email, @password)
    end try
    begin catch
        select 'error'
    end catch
GO
```

### Uses

[dbo].[Student]

## [dbo].[insert\_St\_Crs]

### Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

### Parameters

Name	Data Type	Max Length (Bytes)
@St_Id	int	4
@Crs_id	int	4

### SQL Script

```
create proc [dbo].[insert_St_Crs] @St_Id int , @Crs_id int
as
begin try
    if exists(select St_id , Crs_id from Stud_Course where St_id = @St_Id and Crs_id =@Crs_id)
        begin
            select 'this course is already assigned to this student '
        end
    else
        insert into Stud_Course (St_id , Crs_id)
        values (@St_Id , @Crs_id)
end try
begin catch
    select 'error'
end catch
GO
```

### Uses

[dbo].[Stud\_Course]

## [dbo].[Insert\_topic]

### Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

### Parameters

Name	Data Type	Max Length (Bytes)
@Topic_id	int	4
@Topic_Name	nvarchar(50)	100

### SQL Script

```
CREATE proc [dbo].[Insert_topic] @Topic_id int , @Topic_Name nvarchar(50)
as
begin
    if not exists (select Top_id from Topic where Top_id = @Topic_id )
    begin
        insert into topic ([Top_id],[Top_Name])
        values (@Topic_id , @Topic_Name)
    end
else
    select 'this topic already exists'
end
GO
```

### Uses

[dbo].[Topic]

## [dbo].[Instructor\_Login]

### Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

### Parameters

Name	Data Type	Max Length (Bytes)
@Email	nvarchar(max)	max
@Pass	int	4

### SQL Script

```
create proc [dbo].[Instructor_Login] @Email nvarchar(max) , @Pass int
as
    if exists ( select Ins_Email , Ins_Password from Instructor where Ins_Email = @Email and
Ins_Password = @Pass )
        begin
            select 1
        end
    else
        select 0
GO
```

### Uses

[dbo].[Instructor]

## [dbo].[St\_Exam\_Q]

### Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

### Parameters

Name	Data Type	Max Length (Bytes)
@Ex_id	int	4
@St_id	int	4

### SQL Script

```
CREATE proc [dbo].[St_Exam_Q] @Ex_id int , @St_id int
as
begin
    if exists(select Ex_id , St_id from Student_Exam where Ex_id = @Ex_id and St_id=@St_id )
        begin
            select q.Question_Bady , Se.St Answer from Student Exam se , Question q where q.Q ID
= se.Ques_id
            and Ex_id = @Ex_id and St_id = @St_id
        end
    else
        select 'Error'
end
GO
```

### Uses

[dbo].[Question]  
[dbo].[Student\_Exam]



## [dbo].[Student\_Login]

### Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

### Parameters

Name	Data Type	Max Length (Bytes)
@Email	nvarchar(max)	max
@Pass	int	4

### SQL Script

```
create proc [dbo].[Student_Login] @Email nvarchar(max) , @Pass int
as
    if exists ( select St_Email , St_Password from Student where St_Email = @Email and St_
Password = @Pass )
        begin
            select 1
        end
    else
        select 0
GO
```

### Uses

[dbo].[Student]

## [dbo].[TopicForCrs]

### Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

### Parameters

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

### SQL Script

```
CREATE proc [dbo].[TopicForCrs] @Crs_id int
as
begin
    if exists(select Crs_id from Course where Crs_id = @Crs_id)
        begin
            select Top_Name from Topic t , Course c where t.Top_id = c.Top_id and Crs_id =
@Crs_id
        end
    else
        select 'this Course does not exist'
end
GO
```

### Uses

[dbo].[Course]

[dbo].[Topic]

## [dbo].[update\_instructor]

### Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

### Parameters

Name	Data Type	Max Length (Bytes)
@ins_id	int	4
@ins_Fname	nvarchar(50)	100
@ins_Lname	nvarchar(50)	100
@ins_Email	varchar(100)	100
@ins_password	int	4
@dept_id	int	4
@ins_Address	nvarchar(50)	100

### SQL Script

```
create Proc [dbo].[update_instructor] @ins_id int , @ins_Fname nvarchar(50) , @ins_Lname
nvarchar(50) , @ins_Email varchar(100),
@ins_password int , @dept_id int , @ins_Address nvarchar(50)
as
begin try
    if exists (select Ins_id from Instructor where Ins_id = @ins_id )
        begin
            update Instructor
                set Ins_FName = @ins_Fname , ins_Lname = @ins_Lname , ins_Email = @ins_Email
                , ins_Password = @ins_password , dept_id = @dept_id , ins_Address = @ins_Address
            where ins_id = @ins_id
        end
    else
        select 'instuctor Id is not found'
end try
begin catch
    Select ERROR_MESSAGE()
end catch
GO
```

## Uses

---

[dbo].[Instructor]

## [dbo].[update\_Question]

### Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

### Parameters

Name	Data Type	Max Length (Bytes)
@Q_Id	int	4
@Q_Bady	nvarchar(max)	max
@Q_Type	nvarchar(20)	40
@Model_Ans	nvarchar(250)	500
@Point	int	4
@Crs_ID	int	4

### SQL Script

```
CREATE Proc [dbo].[update_Question] @Q_Id int , @Q_Bady nvarchar(max) , @Q_Type nvarchar(20) ,
@Model_Ans nvarchar(250) , @Point int , @Crs_ID int
as
begin try
    if exists (select Q_ID from Question where Q_ID = @Q_Id )
        begin
            update Question
            set [Q_Type] = @Q_Type , [Question_Bady] = @Q_Bady , [Model_Answer] = @Model_Ans ,
                [Point] = @Point, [Crs_ID] = @Crs_ID
            where Q_ID = @Q_Id
        end
    else
        select 'Question ID Not Found'
    end try
begin catch
    Select ERROR_MESSAGE()
end catch
GO
```

### Uses

[dbo].[Question]

## [dbo].[update\_Student]

### Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

### Parameters

Name	Data Type	Max Length (Bytes)
@st_id	int	4
@st_Fname	nvarchar(50)	100
@st_Lname	nvarchar(50)	100
@st_Email	varchar(100)	100
@st_password	int	4
@dept_id	int	4
@St_Address	nvarchar(50)	100

### SQL Script

```
CREATE Proc [dbo].[update_Student] @st_id int , @st_Fname nvarchar(50) , @st_Lname nvarchar(50) ,
@st_Email varchar(100),
@st_password int , @dept_id int , @St_Address nvarchar(50)
as
begin try
    if exists (select st_id from Student where St_id = @st_id )
        begin
            update Student
                set St_FName = @st_Fname , St_Lname = @st_Lname , St_Email = @st_Email
                    , St_Password = @st_password , dept_id = @dept_id , St_Address = @St_Address
                where St_id = @st_id
        end
    else
        select 'Student Id is not found'
end try
begin catch
    Select ERROR_MESSAGE()
end catch
GO
```

## Uses

---

[dbo].[Student]

## [dbo].[update\_Topic]

### Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

### Parameters

Name	Data Type	Max Length (Bytes)
@Topic_Id	int	4
@Topic_Name	nvarchar(50)	100

### SQL Script

```
-- update Topic

create Proc [dbo].[update_Topic] @Topic_Id int , @Topic_Name nvarchar(50)
as
begin try
    if exists (select Top_id from Topic where Top_id=@Topic_Id )
        begin
            update Topic
            set Top_Name = @Topic_Name
            where Top_id=@Topic_Id
        end
    else
        select 'Topic Id not found'
end try
begin catch
    Select ERROR_MESSAGE()
end catch
GO
```

### Uses

[dbo].[Topic]



## **Users**

### Objects

---

Name
dbo
guest

 **dbo**

### Properties


Property	Value
Type	WindowsUser
Login Name	DESKTOP-O4872J7\DELL
Default Schema	dbo

### Database Level Permissions

Type	Action
CONNECT	Grant

### SQL Script

GO

 guest

### Properties

Property	Value
Type	SqlUser
Default Schema	guest

### SQL Script

GO

## **Database Roles**

### Objects

Name
db_accessadmin
db_backupoperator
db_datareader
db_datawriter
db_ddladmin
db_denydatareader
db_denydatawriter
db_owner
db_securityadmin
public

## **db\_accessadmin**

### Properties

Property	Value
Owner	dbo

## **db\_backupoperator**


### Properties

Property	Value
Owner	dbo

 **db\_datareader**

#### Properties

Property	Value
Owner	dbo

 **db\_datawriter**


#### Properties

Property	Value
Owner	dbo

 **db\_ddladmin**


#### Properties

Property	Value
Owner	dbo

 **db\_denydatareader**


#### Properties

Property	Value
Owner	dbo

 **db\_denydatawriter**

#### Properties

Property	Value
Owner	dbo

 **db\_owner**

#### Properties

Property	Value
Owner	dbo

 **db\_securityadmin**

## Properties

---

Property	Value
Owner	dbo

 public

## Properties

---

Property	Value
Owner	dbo

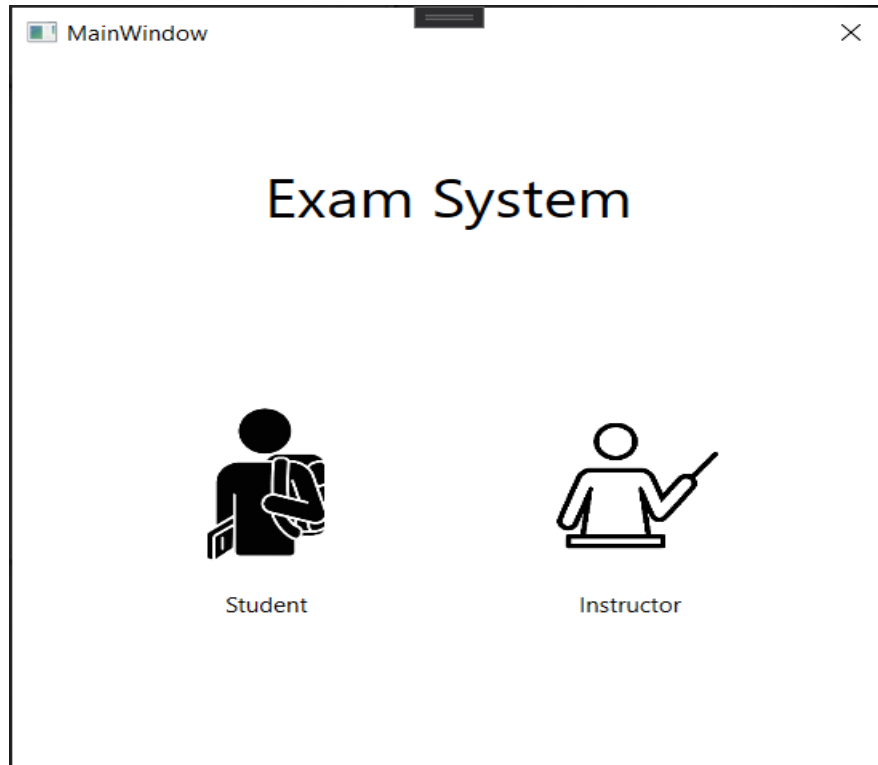
# Desktop Application Documentation

The desktop application has two different types of users, each user can log in by Email and Password saved in a database.

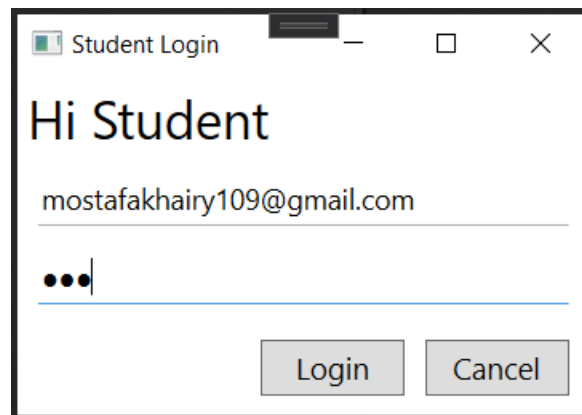
- Student :
  - Can show grades and perform exam.
- Instructor :
  - can add new student.
  - can add new Question.
  - show reports.



## 1-Login Form



## 2-Student Login



Welcome

Hi, Mostafa

Information

Full Name: **Mostafa Khairy** Gender: **Male**  
Student ID: **111** Department ID: **2**  
Email: **mostafakhairy109@gmail.com** Address: **Sohag**

Courses

Number of Courses: **3**

Courses

Results

☐ You still haven't taken this course yet.

Grade:

Start Exam

### 3-Exam Form

Exam

< Question 1 of 7 >

SQL Server has mainly how many types of views?

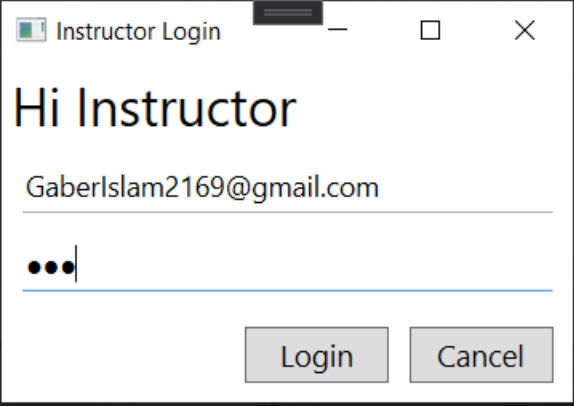
☐ four  
☐ one  
☐ three  
☐ two

Overall Exam Progress Time Elapsed: 00:00:00

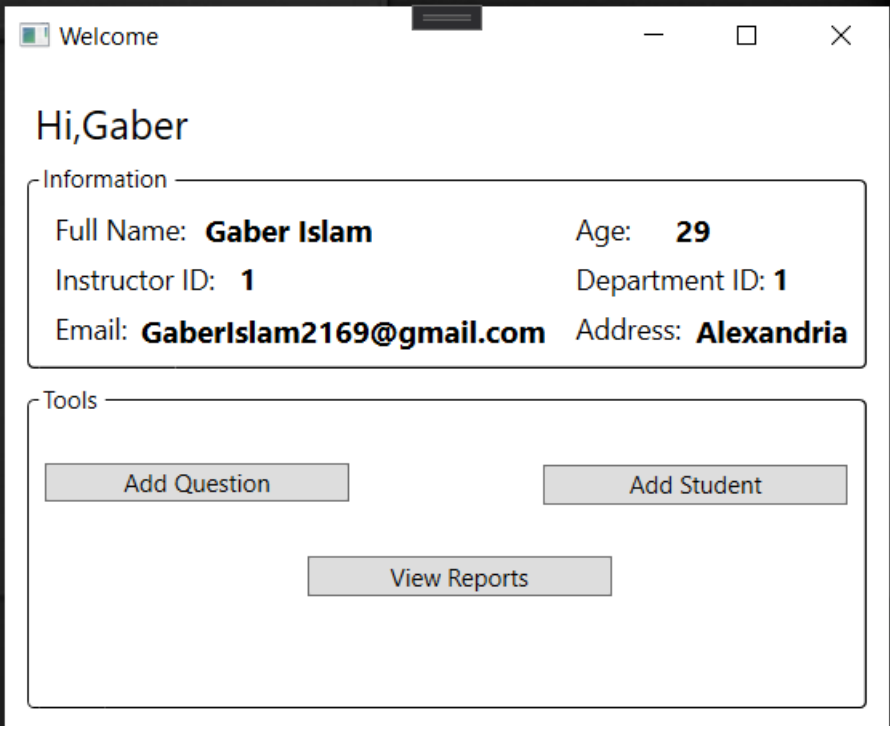
☐

Finish Cancel

## 4-Instructor Login



A screenshot of a web application window titled "Instructor Login". The window has a standard Windows-style title bar with a close button (X) on the right. The main content area displays "Hi Instructor" in a large, bold font. Below this, the email address "GaberIslam2169@gmail.com" is entered into a text field. A password field is shown with three black dots and a cursor. At the bottom right, there are two buttons: "Login" and "Cancel".



A screenshot of a web application window titled "Welcome". The window has a standard Windows-style title bar with a close button (X) on the right. The main content area displays "Hi,Gaber" in a large, bold font. Below this, there is a section titled "Information" which contains a table of user details. Below the information section, there is a section titled "Tools" which contains three buttons: "Add Question", "Add Student", and "View Reports".

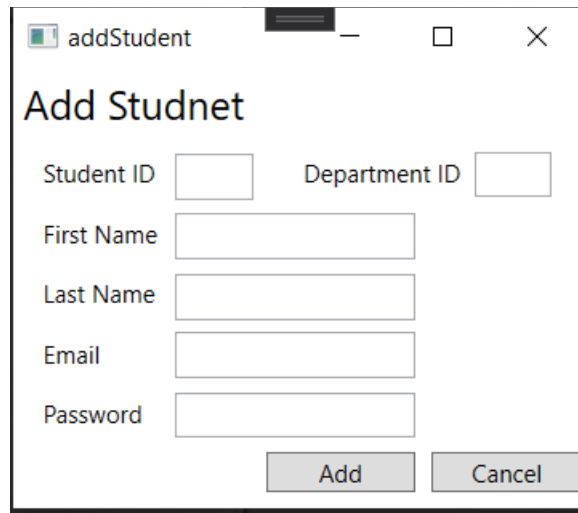
Information	
Full Name: <b>Gaber Islam</b>	Age: <b>29</b>
Instructor ID: <b>1</b>	Department ID: <b>1</b>
Email: <b>GaberIslam2169@gmail.com</b>	Address: <b>Alexandria</b>

Tools

Add Question Add Student

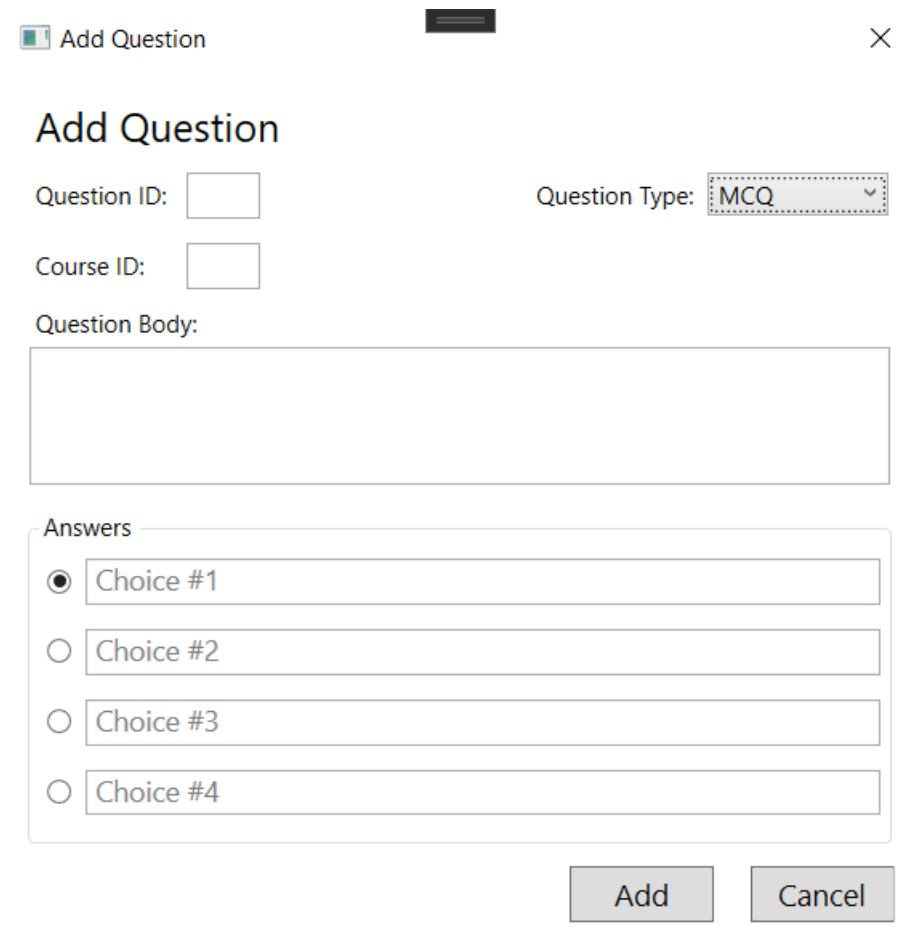
View Reports

## 5-Insert Student Form



The screenshot shows a window titled 'addStudent' with a standard Windows title bar (minimize, maximize, close buttons). The main heading is 'Add Studnet'. Below the heading are five input fields: 'Student ID', 'Department ID', 'First Name', 'Last Name', 'Email', and 'Password'. The 'Student ID' and 'Department ID' fields are small, while the others are larger. At the bottom right, there are two buttons: 'Add' and 'Cancel'.

## 6-Insert Question Form



The screenshot shows a window titled 'Add Question' with a standard Windows title bar. The main heading is 'Add Question'. Below the heading are three input fields: 'Question ID:', 'Course ID:', and 'Question Body:'. The 'Question ID:' and 'Course ID:' fields are small, while the 'Question Body:' field is a large text area. To the right of the 'Question ID:' field is a 'Question Type:' dropdown menu with 'MCQ' selected. Below the 'Question Body:' field is a section titled 'Answers' containing four radio button options: 'Choice #1', 'Choice #2', 'Choice #3', and 'Choice #4'. The 'Choice #1' radio button is selected. At the bottom right, there are two buttons: 'Add' and 'Cancel'.

## 7-Report Form

The screenshot shows a web application window titled "reports\_Form". At the top, there are six buttons labeled "Report 1", "Report 2", "Report 3", "Report 4", "Report 5", and "Report 6". Below these buttons is a navigation bar containing several icons for navigation and document management, followed by a text input field containing "of", a dropdown menu showing "100%", and buttons for "Find" and "Next". The main content area of the window is currently blank.