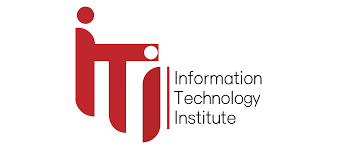
****

**Examination System**

**Submitted by**

**Ahmed Abdelbasset Aboelhamd**

**Seif Eldin Assem Shehata**

**Belal Abdelnasser Mahmoud**

**Amira Hassan Tawfik**

# Abstract

In the world of technology, we can effortlessly get data about schools, universities, training centers, and institutes services with help of desktop application. The most critical part of this application is the examination system that handles the exams generation for instructors as well as taking exams for students.

So here we develop a system for Examination that is a database-backed Windows form application for the purposes of Examination, correction and grading.

easily getting courses and its corresponding exams

The system asks for ID and password and choosing if you are instructor or student to pass our login form providing appropriate tools for them to perform their roles, whether they are professors, teachers, students, course administrators.

To find your exams that you are already enrolled in its courses.

You will be navigated by clicking on Start Exam button to a list of MCQ, T/F questions with its chooses.

Contents

[Abstract 2](#_Toc118214832)

[Introduction 4](#_Toc118214833)

[Requirements: 5](#_Toc118214834)

[Tools & Technologies 6](#_Toc118214835)

[ER-Diagram 7](#_Toc118214836)

[Mapping 7](#_Toc118214837)

[Database Diagram 9](#_Toc118214838)

[Tables 10](#_Toc118214839)

[Stored Procedures 25](#_Toc118214840)

[Windows Application 53](#_Toc118214841)

[Application Overview 53](#_Toc118214842)

[How to Use It 53](#_Toc118214843)

[Technologies Used 53](#_Toc118214844)

[Tools Used 53](#_Toc118214845)

[Application Forms 54](#_Toc118214846)

[1. Login Form 54](#_Toc118214847)

[2. Instructor View Form 55](#_Toc118214848)

[3. Insert Questions View 56](#_Toc118214849)

[4. Insert MCQ question view 57](#_Toc118214850)

[57](#_Toc118214851)

[Inserting MCQ question by typing the question body, then the correct choice and the choices of the question. 57](#_Toc118214852)

[5. Insert T/F Question view 58](#_Toc118214853)

[58](#_Toc118214854)

[6. Student View Form 59](#_Toc118214855)

[7. Exam View Form 60](#_Toc118214856)

[8. Exam Grade View Form 61](#_Toc118214857)

[SSRS Reports deployed on Power BI Report Server 62](#_Toc118214858)

[Courses Topics By Course ID 62](#_Toc118214859)

[Exam Questions by ID 63](#_Toc118214860)

[Instructor Course by ID 64](#_Toc118214861)

[Student Answers By Exam ID and Student ID 65](#_Toc118214862)

[65](#_Toc118214863)

[Student Grade By ID 66](#_Toc118214864)

[Student Department By ID 67](#_Toc118214865)

[Power BI Dashboard 68](#_Toc118214866)

# Introduction

An organization must have accurate and reliable data for effective decision making. To this end, the organization maintains records on the various facets maintaining relationships among them. Such related data are called a database. A database system is an integrated collection of related files, along with details of the interpretation of the data contained therein. Basically, database system is nothing more than a computer-based record keeping system i.e. a system whose overall purpose is to record and maintain information/data.

Also to access to this data and use it you have to use some tool to make it easily for any user tech or non tech for that we have many visualizations tools like applications and reports tools the purposes of teaching and learning that consists of a main page that can be displayed by system users. Each system user has different pages or interface depending on the different functions that he/she can use the system for. The system includes a database system to store all relevant information and data and to produce various reports and queries.

## Requirements:

Construct an Automated System that can perform Online exams and build SQL database for such system.

* ERD.
* Database Dictionary.
* Stored Procedures.
* Select Insert update and delete in the table.
* Exam generation.
* Exam Answers.
* Exam Correction.

**DB will help ITI staff to see following Reports:**

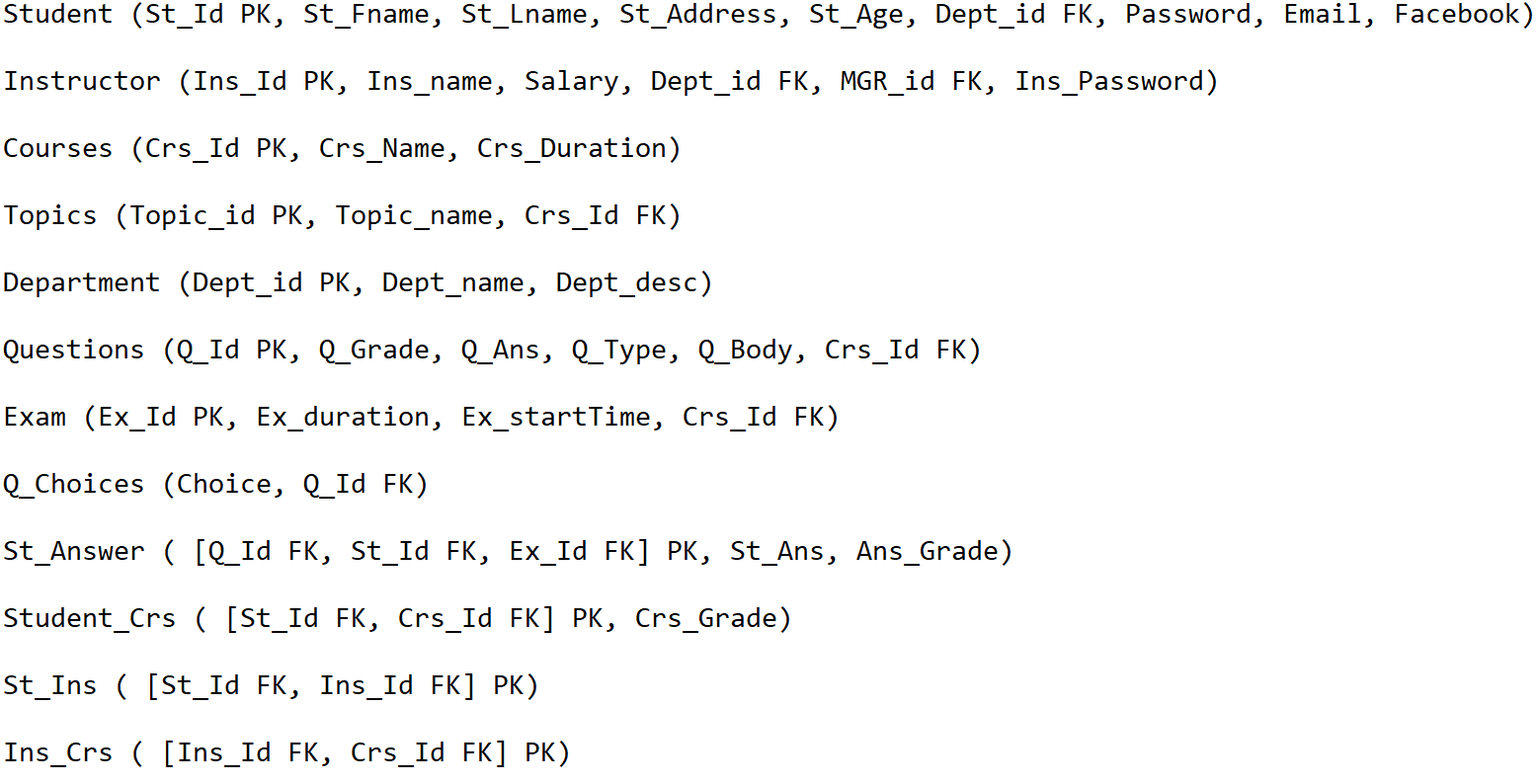
* Report that returns the students information according to Department no parameter.
* Report that takes the student ID and returns the grades of the student per the courses that he teaches and the number of students per course.
* Report that takes course ID and returns its topics.
* Report that takes Exam number and returns Questions in it.
* Report that takes Exam number and the student ID then returns the Questions in the exam with the student answers.

# Tools & Technologies

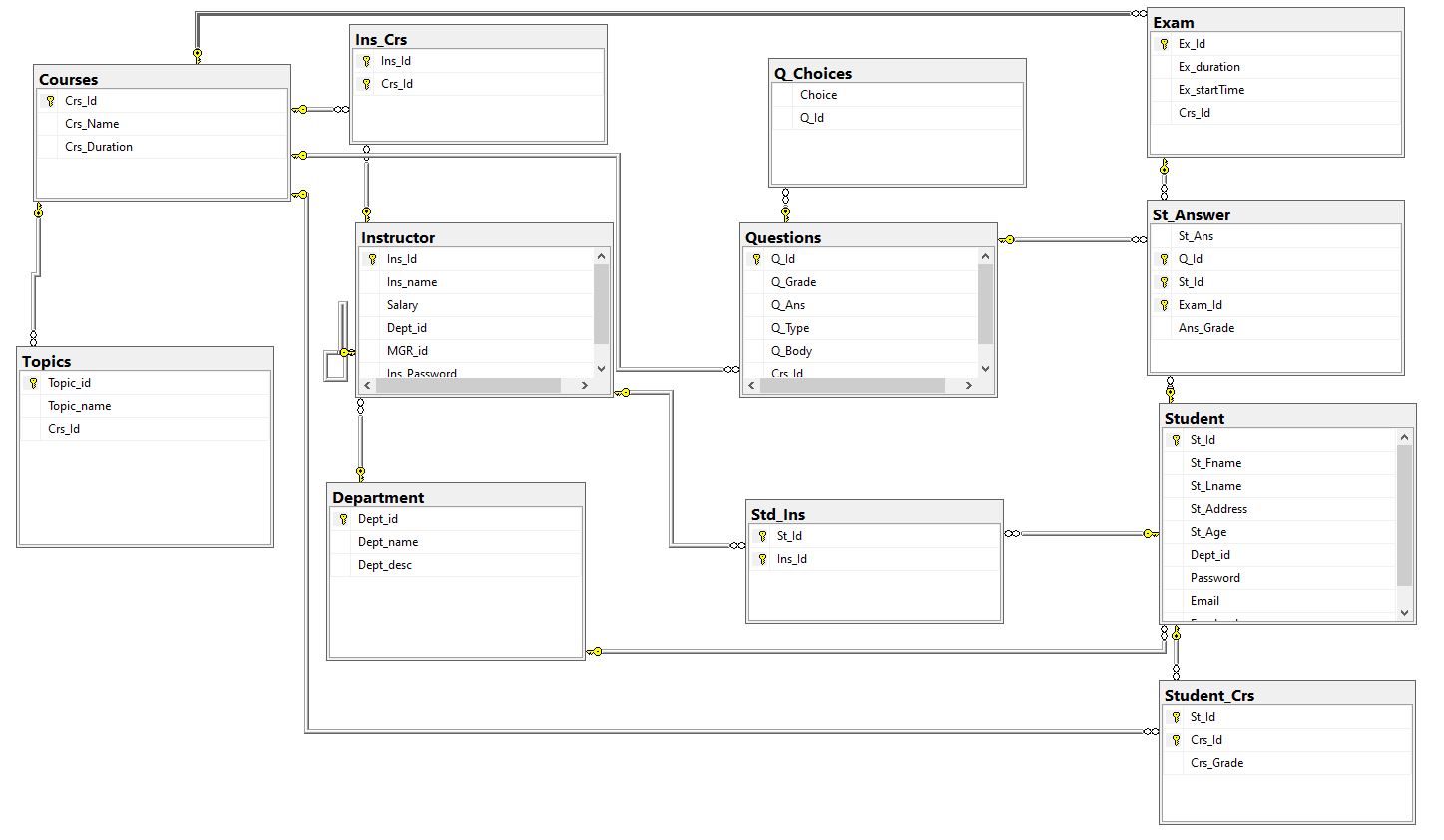
* ERD: Lucid-chart
* Database Engine: Microsoft SQL Server
* Development Environment: DotNet Core 3.1
* IDE: VS 2019
* Reporting: SQL server reporting service (SSRS)
* Visualization: Power Bi

# ER-Diagram

# Mapping



# Database Diagram



# Tables

Objects

|  |
| --- |
| Name |
| [dbo.Courses](#O4ld57VmJLF39OBaaCgMNYCthZ8=) |
| [dbo.Department](#t9zZqTRqCKuw7OCNqp8XFCQCcjk=) |
| [dbo.Exam](#z/su5780V2800FyUuN14oWvsmno=) |
| [dbo.Ins\_­Crs](#Kw0JqTOcrH1XKBGHNj2ms9MC4JA=) |
| [dbo.Instructor](#tGtu288hDPgtM36M2n52W5Dsvik=) |
| [dbo.Q\_­Choices](#SFsehF604mcj2XtKbaEYxh8LNfA=) |
| [dbo.Questions](#mMYACMcSPyiWjpWi0IukAsEwS9M=) |
| [dbo.St\_­Answer](#Nf+ZgK/zOIWL4km6mYLXPab5zck=) |
| [dbo.Std\_­Ins](#yose6ZFL7l+XGzC4GgNls4rcEVg=) |
| [dbo.Student](#v8sVyhZoeheyt+7TiEYGmFtXAgM=) |
| [dbo.Student\_­Crs](#AEVKaVadZhd/dD4HyoIE9KAzWc0=) |
| [dbo.Topics](#FNd8+Olldg3PVXuOJzKfZD3H7QY=) |

|  |
| --- |
| **[dbo].[Courses]** |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | Crs\_­Id | int | 4 | NOT NULL |
|  | Crs\_­Name | varchar(20) | 20 | NULL allowed |
|  | Crs\_­Duration | int | 4 | NULL allowed |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Courses | Crs\_­Id | True |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Courses]  (  [Crs\_­Id] [int] NOT NULL,  [Crs\_­Name] [varchar] (20) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Crs\_­Duration] [int] NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[Courses] ADD CONSTRAINT [PK\_­Courses] PRIMARY KEY CLUSTERED ([Crs\_­Id]) ON [PRIMARY]  GO |

|  |
| --- |
| **[dbo].[Department]** |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | Dept\_id | int | 4 | NOT NULL |
|  | Dept\_name | varchar(20) | 20 | NULL allowed |
|  | Dept\_desc | varchar(100) | 100 | NULL allowed |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Department | Dept\_id | True |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Department]  (  [Dept\_id] [int] NOT NULL,  [Dept\_name] [varchar] (20) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Dept\_desc] [varchar] (100) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[Department] ADD CONSTRAINT [PK\_­Department] PRIMARY KEY CLUSTERED ([Dept\_id]) ON [PRIMARY]  GO |
| **[dbo].[Exam]** |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity |
|  | Ex\_­Id | int | 4 | NOT NULL | 1 - 1 |
|  | Ex\_duration | int | 4 | NOT NULL |  |
|  | Ex\_start­Time | datetime | 8 | NULL allowed |  |
|  | Crs\_­Id | int | 4 | NOT NULL |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Exam | Ex\_­Id | True |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Exam\_­Courses | Crs\_­Id->[[dbo].[Courses].[Crs\_­Id]](#O4ld57VmJLF39OBaaCgMNYCthZ8=) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Exam]  (  [Ex\_­Id] [int] NOT NULL IDENTITY(1, 1),  [Ex\_duration] [int] NOT NULL,  [Ex\_start­Time] [datetime] NULL,  [Crs\_­Id] [int] NOT 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\_­Courses] FOREIGN KEY ([Crs\_­Id]) REFERENCES [dbo].[Courses] ([Crs\_­Id]) |

|  |
| --- |
| **[dbo].[Ins\_­Crs]** |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | Ins\_­Id | int | 4 | NOT NULL |
|  | Crs\_­Id | int | 4 | NOT NULL |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Ins\_­Crs | Ins\_­Id, Crs\_­Id | True |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Ins\_­Crs\_­Courses | Crs\_­Id->[[dbo].[Courses].[Crs\_­Id]](#O4ld57VmJLF39OBaaCgMNYCthZ8=) |
| FK\_­Ins\_­Crs\_­Instructor | Ins\_­Id->[[dbo].[Instructor].[Ins\_­Id]](#tGtu288hDPgtM36M2n52W5Dsvik=) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Ins\_­Crs]  (  [Ins\_­Id] [int] NOT NULL,  [Crs\_­Id] [int] NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[Ins\_­Crs] ADD CONSTRAINT [PK\_­Ins\_­Crs] PRIMARY KEY CLUSTERED ([Ins\_­Id], [Crs\_­Id]) ON [PRIMARY]  GO  ALTER TABLE [dbo].[Ins\_­Crs] ADD CONSTRAINT [FK\_­Ins\_­Crs\_­Courses] FOREIGN KEY ([Crs\_­Id]) REFERENCES [dbo].[Courses] ([Crs\_­Id])  GO  ALTER TABLE [dbo].[Ins\_­Crs] ADD CONSTRAINT [FK\_­Ins\_­Crs\_­Instructor] FOREIGN KEY ([Ins\_­Id]) REFERENCES [dbo].[Instructor] ([Ins\_­Id])  GO |

|  |
| --- |
| **[dbo].[Instructor]** |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | Ins\_­Id | int | 4 | NOT NULL |
|  | Ins\_name | varchar(20) | 20 | NULL allowed |
|  | Salary | money | 8 | NULL allowed |
|  | Dept\_id | int | 4 | NULL allowed |
|  | MGR\_id | int | 4 | NULL allowed |
|  | Ins\_­Password | varchar(8) | 8 | NOT NULL |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Instructor | Ins\_­Id | True |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Instructor\_­Department | Dept\_id->[[dbo].[Department].[Dept\_id]](#t9zZqTRqCKuw7OCNqp8XFCQCcjk=) |
| FK\_­Instructor\_­Instructor | MGR\_id->[[dbo].[Instructor].[Ins\_­Id]](#tGtu288hDPgtM36M2n52W5Dsvik=) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Instructor]  (  [Ins\_­Id] [int] NOT NULL,  [Ins\_name] [varchar] (20) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Salary] [money] NULL,  [Dept\_id] [int] NULL,  [MGR\_id] [int] NULL,  [Ins\_­Password] [varchar] (8) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT 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  ALTER TABLE [dbo].[Instructor] ADD CONSTRAINT [FK\_­Instructor\_­Instructor] FOREIGN KEY ([MGR\_id]) REFERENCES [dbo].[Instructor] ([Ins\_­Id])  GO |
|  |
| **[dbo].[Q\_­Choices]** |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | Choice | varchar(max) | max | NOT NULL |
|  | Q\_­Id | int | 4 | NOT NULL |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Q\_­Choices\_­Questions | Q\_­Id->[[dbo].[Questions].[Q\_­Id]](#mMYACMcSPyiWjpWi0IukAsEwS9M=) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Q\_­Choices]  (  [Choice] [varchar] (max) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Q\_­Id] [int] NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[Q\_­Choices] ADD CONSTRAINT [FK\_­Q\_­Choices\_­Questions] FOREIGN KEY ([Q\_­Id]) REFERENCES [dbo].[Questions] ([Q\_­Id])  GO |

|  |
| --- |
| **[dbo].[Questions]** |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | Q\_­Id | int | 4 | NOT NULL |
|  | Q\_­Grade | int | 4 | NOT NULL |
|  | Q\_­Ans | varchar(max) | max | NULL allowed |
|  | Q\_­Type | varchar(10) | 10 | NULL allowed |
|  | Q\_­Body | varchar(200) | 200 | NULL allowed |
|  | Crs\_­Id | int | 4 | NULL allowed |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Questions | Q\_­Id | True |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Questions\_­Courses | Crs\_­Id->[[dbo].[Courses].[Crs\_­Id]](#O4ld57VmJLF39OBaaCgMNYCthZ8=) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Questions]  (  [Q\_­Id] [int] NOT NULL,  [Q\_­Grade] [int] NOT NULL,  [Q\_­Ans] [varchar] (max) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Q\_­Type] [varchar] (10) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Q\_­Body] [varchar] (200) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Crs\_­Id] [int] NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[Questions] ADD CONSTRAINT [PK\_­Questions] PRIMARY KEY CLUSTERED ([Q\_­Id]) ON [PRIMARY]  GO  ALTER TABLE [dbo].[Questions] ADD CONSTRAINT [FK\_­Questions\_­Courses] FOREIGN KEY ([Crs\_­Id]) REFERENCES [dbo].[Courses] ([Crs\_­Id])  GO |
|  |
| **[dbo].[St\_­Answer]** |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Default |
|  | St\_­Ans | varchar(50) | 50 | NULL allowed |  |
|  | Q\_­Id | int | 4 | NOT NULL |  |
|  | St\_­Id | int | 4 | NOT NULL |  |
|  | Exam\_­Id | int | 4 | NOT NULL |  |
|  | Ans\_­Grade | int | 4 | NOT NULL | ((0)) |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­St\_­Answer | Q\_­Id, St\_­Id, Exam\_­Id | True |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­St\_­Answer\_­Exam | Exam\_­Id->[[dbo].[Exam].[Ex\_­Id]](#z/su5780V2800FyUuN14oWvsmno=) |
| FK\_­St\_­Answer\_­Questions | Q\_­Id->[[dbo].[Questions].[Q\_­Id]](#mMYACMcSPyiWjpWi0IukAsEwS9M=) |
| FK\_­St\_­Answer\_­Student | St\_­Id->[[dbo].[Student].[St\_­Id]](#v8sVyhZoeheyt+7TiEYGmFtXAgM=) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[St\_­Answer]  (  [St\_­Ans] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Q\_­Id] [int] NOT NULL,  [St\_­Id] [int] NOT NULL,  [Exam\_­Id] [int] NOT NULL,  [Ans\_­Grade] [int] NOT NULL CONSTRAINT [DF\_­St\_­Answer\_­Ans\_­Grade] DEFAULT ((0))  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[St\_­Answer] ADD CONSTRAINT [PK\_­St\_­Answer] PRIMARY KEY CLUSTERED ([Q\_­Id], [St\_­Id], [Exam\_­Id]) ON [PRIMARY]  GO  ALTER TABLE [dbo].[St\_­Answer] ADD CONSTRAINT [FK\_­St\_­Answer\_­Exam] FOREIGN KEY ([Exam\_­Id]) REFERENCES [dbo].[Exam] ([Ex\_­Id])  GO  ALTER TABLE [dbo].[St\_­Answer] ADD CONSTRAINT [FK\_­St\_­Answer\_­Questions] FOREIGN KEY ([Q\_­Id]) REFERENCES [dbo].[Questions] ([Q\_­Id])  GO  ALTER TABLE [dbo].[St\_­Answer] ADD CONSTRAINT [FK\_­St\_­Answer\_­Student] FOREIGN KEY ([St\_­Id]) REFERENCES [dbo].[Student] ([St\_­Id])  GO |
| **[dbo].[Std\_­Ins]** |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | St\_­Id | int | 4 | NOT NULL |
|  | Ins\_­Id | int | 4 | NOT NULL |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Std\_­Ins | St\_­Id, Ins\_­Id | True |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Std\_­Ins\_­Instructor | Ins\_­Id->[[dbo].[Instructor].[Ins\_­Id]](#tGtu288hDPgtM36M2n52W5Dsvik=) |
| FK\_­Std\_­Ins\_­Student | St\_­Id->[[dbo].[Student].[St\_­Id]](#v8sVyhZoeheyt+7TiEYGmFtXAgM=) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Std\_­Ins]  (  [St\_­Id] [int] NOT NULL,  [Ins\_­Id] [int] NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[Std\_­Ins] ADD CONSTRAINT [PK\_­Std\_­Ins] PRIMARY KEY CLUSTERED ([St\_­Id], [Ins\_­Id]) ON [PRIMARY]  GO  ALTER TABLE [dbo].[Std\_­Ins] ADD CONSTRAINT [FK\_­Std\_­Ins\_­Instructor] FOREIGN KEY ([Ins\_­Id]) REFERENCES [dbo].[Instructor] ([Ins\_­Id])  GO  ALTER TABLE [dbo].[Std\_­Ins] ADD CONSTRAINT [FK\_­Std\_­Ins\_­Student] FOREIGN KEY ([St\_­Id]) REFERENCES [dbo].[Student] ([St\_­Id])  GO |
|  |
| **[dbo].[Student]** |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | St\_­Id | int | 4 | NOT NULL |
|  | St\_­Fname | varchar(20) | 20 | NULL allowed |
|  | St\_­Lname | varchar(20) | 20 | NULL allowed |
|  | St\_­Address | varchar(50) | 50 | NULL allowed |
|  | St\_­Age | int | 4 | NULL allowed |
|  | Dept\_id | int | 4 | NOT NULL |
|  | Password | varchar(8) | 8 | NOT NULL |
|  | Email | varchar(50) | 50 | NULL allowed |
|  | Facebook | varchar(max) | max | 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]](#t9zZqTRqCKuw7OCNqp8XFCQCcjk=) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Student]  (  [St\_­Id] [int] NOT NULL,  [St\_­Fname] [varchar] (20) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [St\_­Lname] [varchar] (20) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [St\_­Address] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [St\_­Age] [int] NULL,  [Dept\_id] [int] NOT NULL,  [Password] [varchar] (8) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Email] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Facebook] [varchar] (max) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS 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 |
|  |
| **[dbo].[Student\_­Crs]** |

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\_­Student\_­Crs | St\_­Id, Crs\_­Id | True |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Student\_­Crs\_­Courses | Crs\_­Id->[[dbo].[Courses].[Crs\_­Id]](#O4ld57VmJLF39OBaaCgMNYCthZ8=) |
| FK\_­Student\_­Crs\_­Student | St\_­Id->[[dbo].[Student].[St\_­Id]](#v8sVyhZoeheyt+7TiEYGmFtXAgM=) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Student\_­Crs]  (  [St\_­Id] [int] NOT NULL,  [Crs\_­Id] [int] NOT NULL,  [Crs\_­Grade] [int] NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[Student\_­Crs] ADD CONSTRAINT [PK\_­Student\_­Crs] PRIMARY KEY CLUSTERED ([St\_­Id], [Crs\_­Id]) ON [PRIMARY]  GO  ALTER TABLE [dbo].[Student\_­Crs] ADD CONSTRAINT [FK\_­Student\_­Crs\_­Courses] FOREIGN KEY ([Crs\_­Id]) REFERENCES [dbo].[Courses] ([Crs\_­Id])  GO  ALTER TABLE [dbo].[Student\_­Crs] ADD CONSTRAINT [FK\_­Student\_­Crs\_­Student] FOREIGN KEY ([St\_­Id]) REFERENCES [dbo].[Student] ([St\_­Id])  GO |
| **[dbo].[Topics]** |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | Topic\_id | int | 4 | NOT NULL |
|  | Topic\_name | varchar(20) | 20 | NULL allowed |
|  | Crs\_­Id | int | 4 | NULL allowed |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Topics | Topic\_id | True |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Topics\_­Courses | Crs\_­Id->[[dbo].[Courses].[Crs\_­Id]](#O4ld57VmJLF39OBaaCgMNYCthZ8=) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Topics]  (  [Topic\_id] [int] NOT NULL,  [Topic\_name] [varchar] (20) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Crs\_­Id] [int] NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[Topics] ADD CONSTRAINT [PK\_­Topics] PRIMARY KEY CLUSTERED ([Topic\_id]) ON [PRIMARY]  GO  ALTER TABLE [dbo].[Topics] ADD CONSTRAINT [FK\_­Topics\_­Courses] FOREIGN KEY ([Crs\_­Id]) REFERENCES [dbo].[Courses] ([Crs\_­Id])  GO |

# Stored Procedures

Objects

|  |
| --- |
| Name |
| [dbo.ADD\_­Salary30perc](#yudeC/3vzyu/pnx2dGmVV8s+hGU=) |
| [dbo.Answer­Correction](#DTCaDSE+OeoDEy8rlgkzHbCeiAw=) |
| [dbo.answer­Question](#sBPUOxLdgbLqdmxkq5BG+yNY3fE=) |
| [dbo.Crs\_­Insert](#aLYz5x8JVjJ7YktvhpsCYoxir50=) |
| [dbo.Delete­Crs\_by­ID](#SgWEnjSjIXtRb4glWClB7ZY1Te0=) |
| [dbo.Delete­Inst\_by­ID](#zn6alDJ+y3Vof0XQT6t4yNUmFwo=) |
| [dbo.Delete­St\_by­ID](#Lczdkte7bGfDP1r8Inq95/wq9DQ=) |
| [dbo.Delete­Top\_by­ID](#QB+icESud8qAym71kMcCRYIvrew=) |
| [dbo.Dept\_­Insert](#h1o/l2Thkrpmgw+8vIEfr+1iRKQ=) |
| [dbo.Exam­Create](#5ZXsM5t7NDopuulinZPadWjCPb0=) |
| [dbo.Exam­Generation](#2aB8Hki2r/zWtkJ0ByclfWBnfOs=) |
| [dbo.Get­Crs­BYIns](#o7UI5o7bLTaP0RzEXu8WZIrE5+k=) |
| [dbo.get­Crs­Grade](#nhTc9kx1HAAAD+SApz8uQGjYDjM=) |
| [dbo.get­Exam­Duration](#vZgDVOFuuGEglIcINDbFJxp7jfI=) |
| [dbo.Get­Ques­BYEx](#wDYTYPkCDG8GMmx+PfVIneP39OQ=) |
| [dbo.Get­Ques­Nnd­Answer­BYEx](#zeGXwbqi/aZMYpA5amlaWRgpgSM=) |
| [dbo.get­Question­List](#RJ5LOS8C1O8AJiVz5xhjNHf50K8=) |
| [dbo.Get­St­Grades](#D5a0XLxQMdbf3QXzYqoBJeHpt+4=) |
| [dbo.Get­Top­BYCrs](#736NYK+eE2esT6YT1K7UVZ/rWBg=) |
| [dbo.Ins\_­Crs\_­Insert](#2O7NXCo8Aceue/j0ci/9TsB760k=) |
| [dbo.Insert\_­Inst](#zcLHcKZ9xpxXRBco11kDwOGI8F8=) |
| [dbo.insert­Choices](#vmoxOB6oW6woQ+KGT1hXNb/tVZU=) |
| [dbo.insert­Data­Exam](#ZXhjnE4v9G+MXyHtrIUqOlICSAc=) |
| [dbo.insert­Data­Question](#WITFe+8z7Y8BHtjlWjCq6hhc2RQ=) |
| [dbo.Instructor­Crs](#mLDJ4MJlP/83hSkJYAj3Qn/o9xY=) |
| [dbo.Instructor­Login](#ZdU7+JqKNb9AqnqBeTlMZDprd44=) |
| [dbo.question­Choices](#kLPrv+WEYf7AfJezLokp+LpSsYQ=) |
| [dbo.select­Exam­Data](#w4GVHoDp8/esad5+eeMf7vvOfMc=) |
| [dbo.Std\_­Crs\_­Insert](#LGM0V87iVLIuIoA1V//VzpIXjlk=) |
| [dbo.Std\_­Insert](#/b/c23k0DCQ/QkY6kqxW8vwOh8I=) |
| [dbo.Std\_­Inst\_­Insert](#5pBkRGlVi/glWXgctWRGqPdGhYQ=) |
| [dbo.student­Courses](#nFRDVEQh+U9jK7bK6hHncuL4ecY=) |
| [dbo.Student­Login](#P67vncHs3REn5elnWEWNdnLOM5I=) |
| [dbo.Top\_­Insert](#S0WkhcCWXURiIrBk7dO8/wcUd0c=) |
| [dbo.Update\_­Crs­Duration](#/FNxozP3zEXFJ8anL3ruIUpvfYk=) |
| [dbo.Update\_­St­Addr](#T3w6J2lrozdFtezbsBk3kzKBVNQ=) |
| [dbo.Update\_­St­Dept](#nS8uwyR4BXvRXSXst8GaC2TBYsA=) |
| [dbo.Update\_­Top­Name](#232qcC2y2QNNSZ+Unpz2Qwlw7WM=) |

|  |
| --- |
| **[dbo].[ADD\_­Salary30perc]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @inst­ID | int | 4 |

SQL Script

|  |
| --- |
| ----------------------------------------------------------------------  create proc [dbo].[ADD\_­Salary30perc] @inst­ID int  with encryption  as  Update Instructor set Salary = Salary\*1.3 where Ins\_­Id = @inst­ID  GO |

|  |
| --- |
| **[dbo].[Answer­Correction]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @SID | int | 4 |
| @CID | int | 4 |

SQL Script

|  |
| --- |
| create proc [dbo].[Answer­Correction] @SID int ,@CID int  as  Update St\_­Answer  set Ans\_­Grade = Q.Q\_­Grade  from St\_­Answer SA inner join Questions Q  on SA.Q\_­Id = Q.Q\_­Id  where SA.St\_­Ans = Q.Q\_­Ans  update Student\_­Crs set Crs\_­Grade =  (select Sum(Ans\_­Grade) as sumof­Q  from St\_­Answer SA, Questions Q  where SA.St\_­Id=St\_­Id and Q.Crs\_­Id = Crs\_­Id and Q.Q\_­Id = SA.Q\_­Id  AND St\_­Id=@SID and Crs\_­Id=@CID  )  where St\_­Id=@SID and Crs\_­Id=@CID  GO |

|  |
| --- |
| **[dbo].[answer­Question]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @QID | int | 4 |
| @SID | int | 4 |
| @CID | int | 4 |
| @Ans | varchar(50) | 50 |

SQL Script

|  |
| --- |
| create proc [dbo].[answer­Question](@QID int, @SID int, @CID int, @Ans varchar(50))  as  update St\_­Answer set St\_­Ans=@Ans where Q\_­Id=@QID  and Q\_­Id=(select Q\_­Id from Questions where Q\_­Id =@QID and Crs\_­Id=@CID)  and St\_­Id=@SID  GO |

|  |
| --- |
| **[dbo].[Crs\_­Insert]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @id | int | 4 |
| @name | varchar(20) | 20 |
| @duration | int | 4 |

SQL Script

|  |
| --- |
| create proc [dbo].[Crs\_­Insert]  (  @id int,  @name varchar(20),  @duration int  )  with encryption  AS  insert into Courses values(@id,@name,@duration)  GO |

|  |
| --- |
| **[dbo].[Delete­Crs\_by­ID]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @crs­ID | int | 4 |

SQL Script

|  |
| --- |
| ----------------------------------------------------------------------  create proc [dbo].[Delete­Crs\_by­ID] @crs­ID int  with encryption  as  delete Courses where Crs\_­Id = @crs­ID  GO |

|  |
| --- |
| **[dbo].[Delete­Inst\_by­ID]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @inst­ID | int | 4 |

SQL Script

|  |
| --- |
| ----------------------------------------------------------------------  create proc [dbo].[Delete­Inst\_by­ID] @inst­ID int  with encryption  as  delete from Instructor where Ins\_­Id = @inst­ID  GO |

|  |
| --- |
| **[dbo].[Delete­St\_by­ID]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @St­ID | int | 4 |

SQL Script

|  |
| --- |
| ----------------------------------------------------------------------  create proc [dbo].[Delete­St\_by­ID] @St­ID int  with encryption  as  delete from Student where St\_­Id = @St­ID  GO |

|  |
| --- |
| **[dbo].[Delete­Top\_by­ID]** |

Parameters

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

SQL Script

|  |
| --- |
| ----------------------------------------------------------------------  create proc [dbo].[Delete­Top\_by­ID] @ID int  with encryption  as  delete Topics where Topic\_id = @ID  GO |

|  |
| --- |
| **[dbo].[Dept\_­Insert]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @id | int | 4 |
| @name | varchar(20) | 20 |
| @desc | varchar(100) | 100 |

SQL Script

|  |
| --- |
| CREATE proc [dbo].[Dept\_­Insert]  (  @id int,  @name VARCHAR(20),  @desc VARCHAR(100)  )  with encryption  AS  BEGIN  insert into Department values( @id, @name, @desc)  END  GO |

|  |
| --- |
| **[dbo].[Exam­Create]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @exam­Duration | int | 4 |
| @Crs­ID | int | 4 |

SQL Script

|  |
| --- |
| create proc [dbo].[Exam­Create] @exam­Duration int , @Crs­ID int  with encryption  as  insert into Exam(Ex\_duration,Ex\_start­Time,Crs\_­Id)values(@exam­Duration, GETDATE(), @Crs­ID)  GO |

|  |
| --- |
| **[dbo].[Exam­Generation]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @SID | int | 4 |
| @CID | int | 4 |

SQL Script

|  |
| --- |
| create proc [dbo].[Exam­Generation] @SID int, @CID int  with encryption  as  declare @EID int =(  select top(1) Ex\_­Id from Exam  where Crs\_­Id = @CID  order by NEWID() )  insert into St\_­Answer (Q\_­Id,St\_­Id,Exam\_­Id)  SELECT \*  FROM  (  SELECT TOP(5) Q\_­Id , St\_­Id , Ex\_­Id  FROM Questions Q, Student S, Exam E  where Q\_­Type = 'MCQ' AND Q.Crs\_­Id=@CID  AND S.St\_­Id = @SID AND E.Ex\_­Id = @EID  ORDER BY NEWID()  ) MCQ  UNION ALL  SELECT \* FROM  (  SELECT TOP(5) Q\_­Id ,St\_­Id, Ex\_­Id  FROM Questions Q, Student S ,Exam E  where Q\_­Type = 'T/F' AND Q.Crs\_­Id=@CID  AND S.St\_­Id = @SID AND E.Ex\_­Id = @EID  ORDER BY NEWID()  ) TF  GO |

|  |
| --- |
| **[dbo].[Get­Crs­BYIns]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @ins\_id | int | 4 |

SQL Script

|  |
| --- |
| create proc [dbo].[Get­Crs­BYIns] @ins\_id int as  SELECT Instructor.Ins\_name, Courses.Crs\_­Name, Student\_­Crs.St\_­Id  FROM Ins\_­Crs INNER JOIN  Instructor ON Ins\_­Crs.Ins\_­Id = Instructor.Ins\_­Id INNER JOIN  Student\_­Crs ON Ins\_­Crs.Crs\_­Id = Student\_­Crs.Crs\_­Id INNER JOIN  Courses ON Ins\_­Crs.Crs\_­Id = Courses.Crs\_­Id AND Student\_­Crs.Crs\_­Id = Courses.Crs\_­Id  where Instructor.Ins\_­Id=@ins\_id  GO |

|  |
| --- |
| **[dbo].[get­Crs­Grade]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @SID | int | 4 |
| @CID | int | 4 |

SQL Script

|  |
| --- |
| create proc [dbo].[get­Crs­Grade] @SID int, @CID int  as  select Crs\_­Grade from Student\_­Crs where St\_­Id=@SID AND Crs\_­Id=@CID  GO |

|  |
| --- |
| **[dbo].[get­Exam­Duration]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @CID | int | 4 |

SQL Script

|  |
| --- |
| create proc [dbo].[get­Exam­Duration] @CID int  as  select Ex\_duration,Crs\_­Id from Exam where Crs\_­Id=@CID  GO |

|  |
| --- |
| **[dbo].[Get­Ques­BYEx]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Exam\_id | int | 4 |

SQL Script

|  |
| --- |
| create proc [dbo].[Get­Ques­BYEx] @Exam\_id int as  SELECT Exam.Ex\_­Id, Questions.Q\_­Body  FROM Exam INNER JOIN  St\_­Answer ON Exam.Ex\_­Id = St\_­Answer.Exam\_­Id INNER JOIN  Questions ON St\_­Answer.Q\_­Id = Questions.Q\_­Id  where Exam.Ex\_­Id=@Exam\_id  GO |

|  |
| --- |
| **[dbo].[Get­Ques­Nnd­Answer­BYEx]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Exam\_id | int | 4 |
| @ST\_id | int | 4 |

SQL Script

|  |
| --- |
| create proc [dbo].[Get­Ques­Nnd­Answer­BYEx] @Exam\_id int, @ST\_id int as  SELECT Student.St\_­Id, Student.St\_­Fname, Exam.Ex\_­Id, Questions.Q\_­Body, St\_­Answer.St\_­Ans  FROM Student INNER JOIN  St\_­Answer ON Student.St\_­Id = St\_­Answer.St\_­Id INNER JOIN  Questions ON St\_­Answer.Q\_­Id = Questions.Q\_­Id INNER JOIN  Exam ON St\_­Answer.Exam\_­Id = Exam.Ex\_­Id  where Student.St\_­Id=@ST\_id and Exam.Ex\_­Id=@Exam\_id  GO |

|  |
| --- |
| **[dbo].[get­Question­List]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @SID | int | 4 |
| @CID | int | 4 |

SQL Script

|  |
| --- |
| CREATE proc [dbo].[get­Question­List] @SID int, @CID int  as  select Q.Q\_­Id, Q\_­Body from Questions Q , St\_­Answer SA  where Q.Q\_­Id = SA.Q\_­Id AND SA.St\_­Id=@SID AND Q.Crs\_­Id=@CID  order by SA.Q\_­Id  select Choice from Questions Q ,Q\_­Choices C, St\_­Answer SA  where Q.Q\_­Id = SA.Q\_­Id AND C.Q\_­Id = Q.Q\_­Id  order by Q.Q\_­Id  GO |

|  |
| --- |
| **[dbo].[Get­St­Grades]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @ST\_id | int | 4 |

SQL Script

|  |
| --- |
| create proc [dbo].[Get­St­Grades] @ST\_id int as  SELECT Student.St\_­Id, Student.St\_­Fname, Student.St\_­Lname, Student\_­Crs.Crs\_­Grade, Courses.Crs\_­Name  FROM Student\_­Crs INNER JOIN  Student ON Student\_­Crs.St\_­Id = Student.St\_­Id INNER JOIN  Courses ON Student\_­Crs.Crs\_­Id = Courses.Crs\_­Id  where Student.St\_­Id=@ST\_id  GO |

|  |
| --- |
| **[dbo].[Get­Top­BYCrs]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Crs\_id | int | 4 |

SQL Script

|  |
| --- |
| create proc [dbo].[Get­Top­BYCrs] @Crs\_id int as  SELECT Courses.Crs\_­Id, Courses.Crs\_­Name, Topics.Topic\_name  FROM Topics INNER JOIN  Courses ON Topics.Crs\_­Id = Courses.Crs\_­Id  where Courses.Crs\_­Id=@Crs\_id  GO |

|  |
| --- |
| **[dbo].[Ins\_­Crs\_­Insert]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @iid | int | 4 |
| @cid | int | 4 |

SQL Script

|  |
| --- |
| create proc [dbo].[Ins\_­Crs\_­Insert]  (  @iid int,  @cid int  )  with encryption  AS  BEGIN  insert into Ins\_­Crs values(@iid,@cid)  END  GO |

|  |
| --- |
| **[dbo].[Insert\_­Inst]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @id | int | 4 |
| @name | varchar(20) | 20 |
| @salary | money | 8 |
| @deptid | int | 4 |
| @mgr | int | 4 |
| @pass | varchar(8) | 8 |

SQL Script

|  |
| --- |
| CREATE proc [dbo].[Insert\_­Inst]  (  @id int,  @name VARCHAR(20),  @salary money,  @deptid int,  @mgr int,  @pass varchar(8)  )  with encryption  AS  BEGIN  insert into Instructor values( @id, @name, @salary,@deptid, @mgr,@pass)  END  GO |

|  |
| --- |
| **[dbo].[insert­Choices]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @choice | varchar(max) | max |
| @id | int | 4 |

SQL Script

|  |
| --- |
| ----------------------------------------------------------------------  CREATE proc [dbo].[insert­Choices]  (  @choice varchar(MAX),  @id int  )  with encryption  as  insert into Q\_­Choices values(@choice,@id)  GO |
|  |
| **[dbo].[insert­Data­Exam]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @duration | int | 4 |
| @start\_­Time | datetime | 8 |
| @course­Id | int | 4 |

SQL Script

|  |
| --- |
| create proc [dbo].[insert­Data­Exam] @duration int, @start\_­Time datetime, @course­Id int  with encryption  as  insert into Exam values(@duration, @start\_­Time, @course­Id)  GO |

|  |
| --- |
| **[dbo].[insert­Data­Question]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @id | int | 4 |
| @grade | int | 4 |
| @answer | varchar(max) | max |
| @type | varchar(10) | 10 |
| @body | varchar(200) | 200 |
| @Crs\_­Id | int | 4 |

SQL Script

|  |
| --- |
| Create proc [dbo].[insert­Data­Question] @id int, @grade int, @answer varchar(max), @type varchar(10), @body varchar(200),@Crs\_­Id int  with encryption  as  insert into Questions values(@id, @grade, @answer, @type, @body,@Crs\_­Id)  GO |

|  |
| --- |
| **[dbo].[Instructor­Crs]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Ins­ID | int | 4 |

SQL Script

|  |
| --- |
| CREATE proc [dbo].[Instructor­Crs] @Ins­ID int  as  select C.Crs\_­Id, Crs\_­Name from Courses C , Ins\_­Crs IC  where C.Crs\_­Id = IC.Crs\_­Id AND IC.Ins\_­Id = @Ins­ID  GO |

|  |
| --- |
| **[dbo].[Instructor­Login]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @ID | int | 4 |
| @PW | varchar(8) | 8 |

SQL Script

|  |
| --- |
| create proc [dbo].[Instructor­Login]  (@ID int , @PW varchar(8))  as  select Count(\*) from Instructor where Ins\_­Id = @ID and Ins\_­Password = @PW  GO |

|  |
| --- |
| **[dbo].[question­Choices]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @QID | int | 4 |

SQL Script

|  |
| --- |
| create proc [dbo].[question­Choices] @QID int  as  select Choice, C.Q\_­Id from Q\_­Choices C, Questions Q, St\_­Answer SA  where Q.Q\_­Id=C.Q\_­Id AND SA.Q\_­Id=C.Q\_­Id AND C.Q\_­Id=@QID  group by C.Q\_­Id,Choice  GO |

|  |
| --- |
| **[dbo].[select­Exam­Data]** |

SQL Script

|  |
| --- |
| CREATE proc [dbo].[select­Exam­Data]  with encryption  as  select Ex\_duration,Crs\_­Id from Exam  GO |

|  |
| --- |
| **[dbo].[Std\_­Crs\_­Insert]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @sid | int | 4 |
| @cid | int | 4 |

SQL Script

|  |
| --- |
| create proc [dbo].[Std\_­Crs\_­Insert]  (  @sid int,  @cid int    )  with encryption  AS  BEGIN  declare @grade as int = 0;  insert into Student\_­Crs values(@sid,@cid,@grade)  END  GO |

|  |
| --- |
| **[dbo].[Std\_­Insert]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @id | int | 4 |
| @fname | varchar(20) | 20 |
| @lname | varchar(20) | 20 |
| @addr | varchar(50) | 50 |
| @age | int | 4 |
| @deptid | int | 4 |
| @pass | varchar(8) | 8 |
| @email | varchar(50) | 50 |

SQL Script

|  |
| --- |
| CREATE proc [dbo].[Std\_­Insert]  (  @id int,  @fname VARCHAR(20),  @lname VARCHAR(20),  @addr varchar(50),  @age int,  @deptid int,  @pass varchar(8),  @email varchar(50)  )  with encryption  AS  BEGIN  insert into Student values( @id, @fname, @lname, @addr, @age, @deptid, @pass, @email)  END  GO |

|  |
| --- |
| **[dbo].[Std\_­Inst\_­Insert]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @sid | int | 4 |
| @iid | int | 4 |

SQL Script

|  |
| --- |
| CREATE proc [dbo].[Std\_­Inst\_­Insert]  (  @sid int,  @iid int  )  with encryption  AS  BEGIN  insert into Std\_­Ins values(@sid,@iid)  END  GO |

|  |
| --- |
| **[dbo].[student­Courses]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @St­ID | int | 4 |

SQL Script

|  |
| --- |
| create proc [dbo].[student­Courses] @St­ID int  as  select distinct S.St\_­Id, C.Crs\_­Id, Crs\_­Name from Student S, Courses C, Student\_­Crs SC,Exam E  where S.St\_­Id = SC.St\_­Id AND C.Crs\_­Id = SC.Crs\_­Id  AND E.Crs\_­Id = C.Crs\_­Id  AND not exists (  select St\_­Id from Student\_­Crs where St\_­Id=S.St\_­Id AND Crs\_­Id=C.Crs\_­Id AND Crs\_­Grade!=0  )  AND S.St\_­Id=@St­ID  GO |

|  |
| --- |
| **[dbo].[Student­Login]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @ID | int | 4 |
| @PW | varchar(8) | 8 |

SQL Script

|  |
| --- |
| create proc [dbo].[Student­Login]  (@ID int , @PW varchar(8))  as  select Count(\*) from Student where St\_­Id = @ID and Password = @PW  GO |

|  |
| --- |
| **[dbo].[Top\_­Insert]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @id | int | 4 |
| @name | varchar(20) | 20 |
| @cid | int | 4 |

SQL Script

|  |
| --- |
| create proc [dbo].[Top\_­Insert]  (  @id int,  @name varchar(20),  @cid int  )  with encryption  AS  BEGIN  insert into Topics values(@id,@name,@cid)  END  GO |

|  |
| --- |
| **[dbo].[Update\_­Crs­Duration]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Crs­Id | int | 4 |
| @Crs­Duration | int | 4 |

SQL Script

|  |
| --- |
| create proc [dbo].[Update\_­Crs­Duration] @Crs­Id int , @Crs­Duration int  with encryption  as  Update Courses set Crs\_­Duration=@Crs­Duration where Crs\_­Id=@Crs­Id  GO |

|  |
| --- |
| **[dbo].[Update\_­St­Addr]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @st­ID | int | 4 |
| @st­Add | varchar(50) | 50 |

SQL Script

|  |
| --- |
| create proc [dbo].[Update\_­St­Addr] @st­ID int , @st­Add varchar(50)  with encryption  as  Update Student set St\_­Address=@st­Add where St\_­Id=@st­ID  GO |

|  |
| --- |
| **[dbo].[Update\_­St­Dept]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @st­ID | int | 4 |
| @Dept­Id | int | 4 |

SQL Script

|  |
| --- |
| create proc [dbo].[Update\_­St­Dept] @st­ID INT , @Dept­Id int  with encryption  as  Update Student set Dept\_id=@Dept­Id WHERE St\_­Id=@st­ID  GO |

|  |
| --- |
| **[dbo].[Update\_­Top­Name]** |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @top­Id | int | 4 |
| @topname | varchar(20) | 20 |

SQL Script

|  |
| --- |
| ----------------------------------------------------------------------  create proc [dbo].[Update\_­Top­Name] @top­Id int , @topname varchar(20)  with encryption  as  Update Topics set Topic\_name = @topname where Topic\_id = @top­Id  GO |

# Windows Application

## Application Overview

A desktop application for examination system where instructor can generate exam and students take exams and get their grade instantly

### How to Use It

1- Sign in as student/ instructor

2- IDs and Passwords to try Student: ID (1) Password (1230) Instructor: ID (202) Password(c#123) (for more kindly check the database)

3- Login as instructor first to add an exam (insert duration and select one of the courses that this instructor teach, database would be updated in grid)

4- Logout then login as student (student would only find in the exam list the courses which he is enrolled in that has an exam)

5- Press button to get started (a random exam model (if more than 1 exam is created) is generated and random questions are generated)

6- choose the correct answer from the combo box beside each question (if duration (in minutes) has passed and student didn't submit the button, the exam is submitted and grade reviewed)

7- press submit to get redirected to the grade page

### Technologies Used

C#, ADO, SQL Server, Windows Form Application

### Tools Used

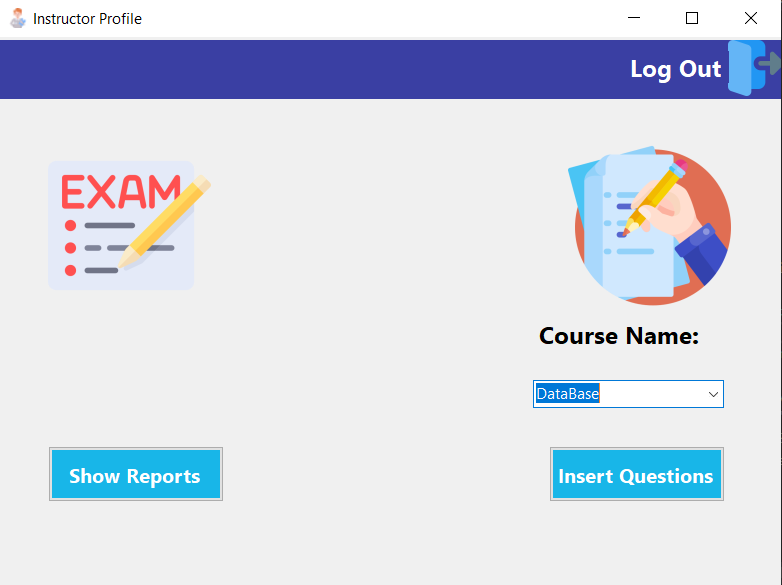
Desktop Application: Windows Form App (.NET) C# Reports: Report Builder Dictionary: SQL Data Dictionary ERD and Mapping: Draw.io

## Application Forms

### Login Form

Authentication form that takes the ID and Password of the student or instructor and depends on the login selection it will move you to the next form (Student or Instructor) if the ID and Password exists in the database

### Instructor View Form



The instructor form enables him to insert questions in an exam of a selected course (that he gives) when clicking insert questions will transform the instructor to inserting question form and by click Show Reports button will direct him to power bi services showing reports

### Insert Questions View

Insert Question View makes instructor to chose which type of questions you wants to add in his course if its MCQ or T/F

### Insert MCQ question view

### 

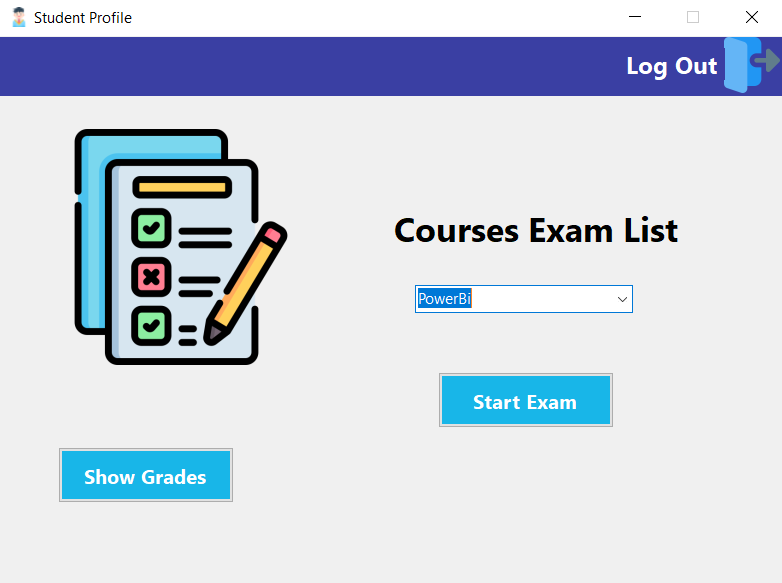
### Inserting MCQ question by typing the question body, then the correct choice and the choices of the question.

### Insert T/F Question view

### 

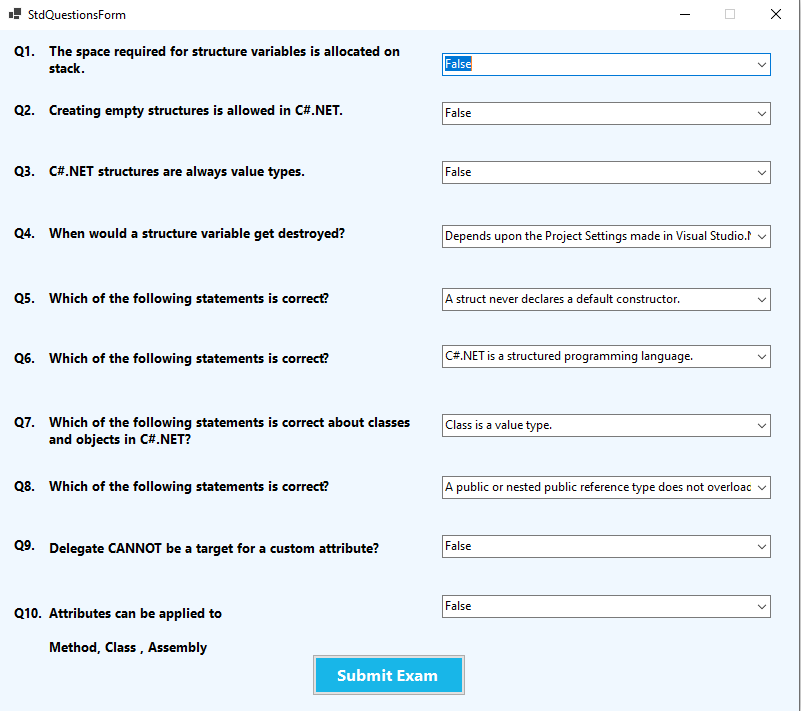
Inserting MCQ question by typing the question body, then the correct choice and the choices of the question.

### Student View Form



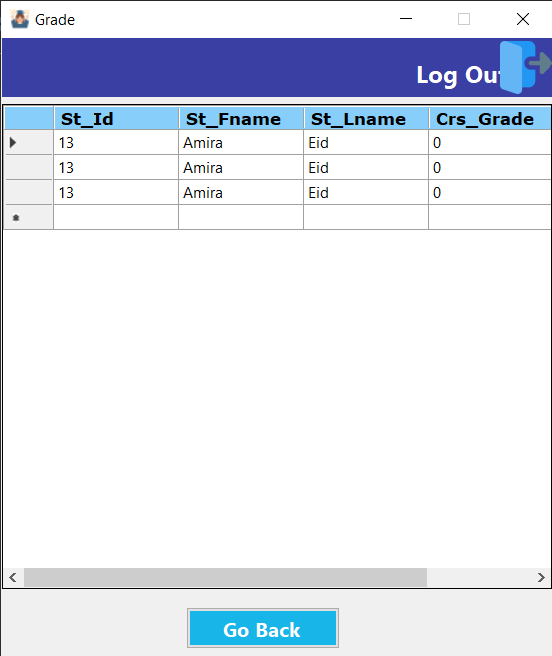
The student form enables him to take an exam of a selected course (that he takes) once he clicked Start Exam Button, he gets transferred to his exam questions and start to answer the questions, and showing him his old grades in every course he already enrolled

### Exam View Form



The exam which the student takes, the student from the drop-down menu the answer of each question then submits his answers to get his grade in the grade form

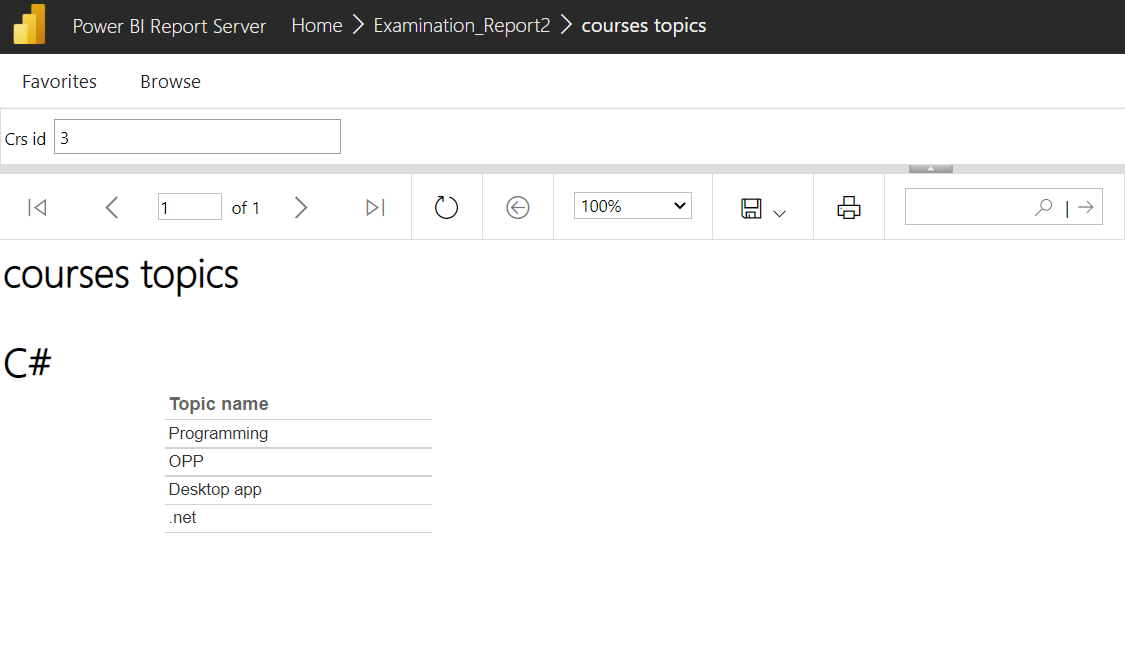
### Exam Grade View Form



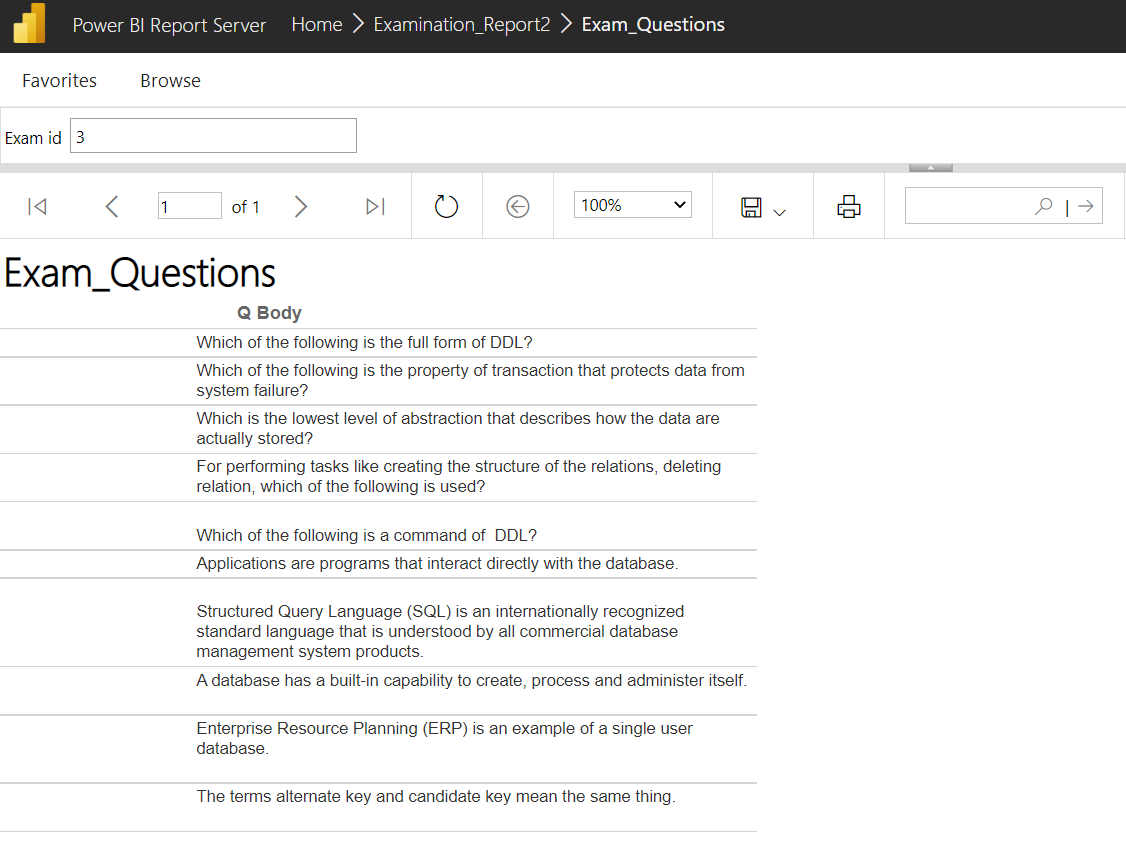
The grade form shows the student’s degree in every course he/she already enrolled

# SSRS Reports deployed on Power BI Report Server

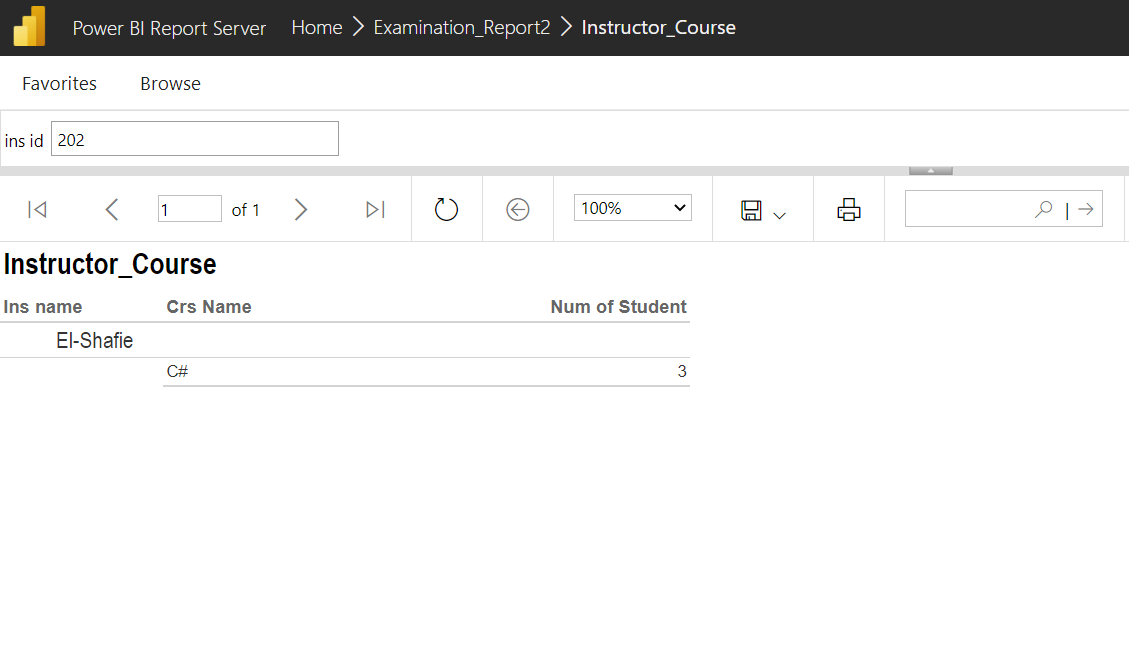
## Courses Topics By Course ID



## Exam Questions by ID



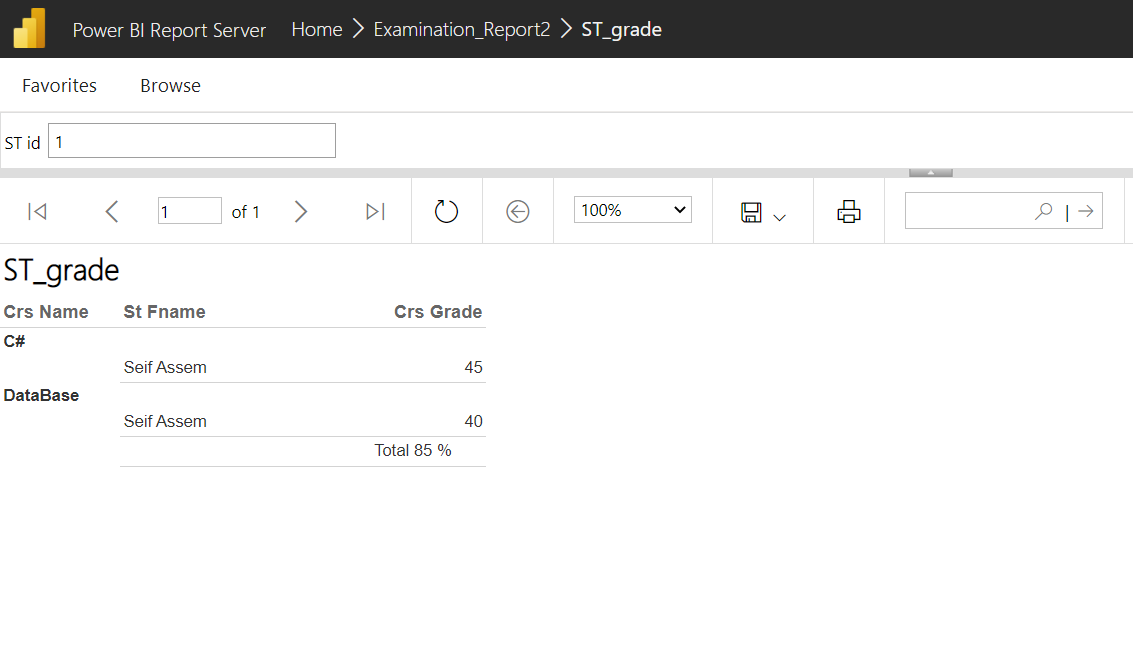
## Instructor Course by ID



## Student Answers By Exam ID and Student ID

## 

## Student Grade By ID



## Student Department By ID

# Power BI Dashboard