Examination System

Team 5

Team members:-

1- **Hala Sherif**

2- **Heba Elsaied**

3- **Hellana Asaad**

4- **Yasmeen Abotaleb**

5- **Dina Mahmoud**

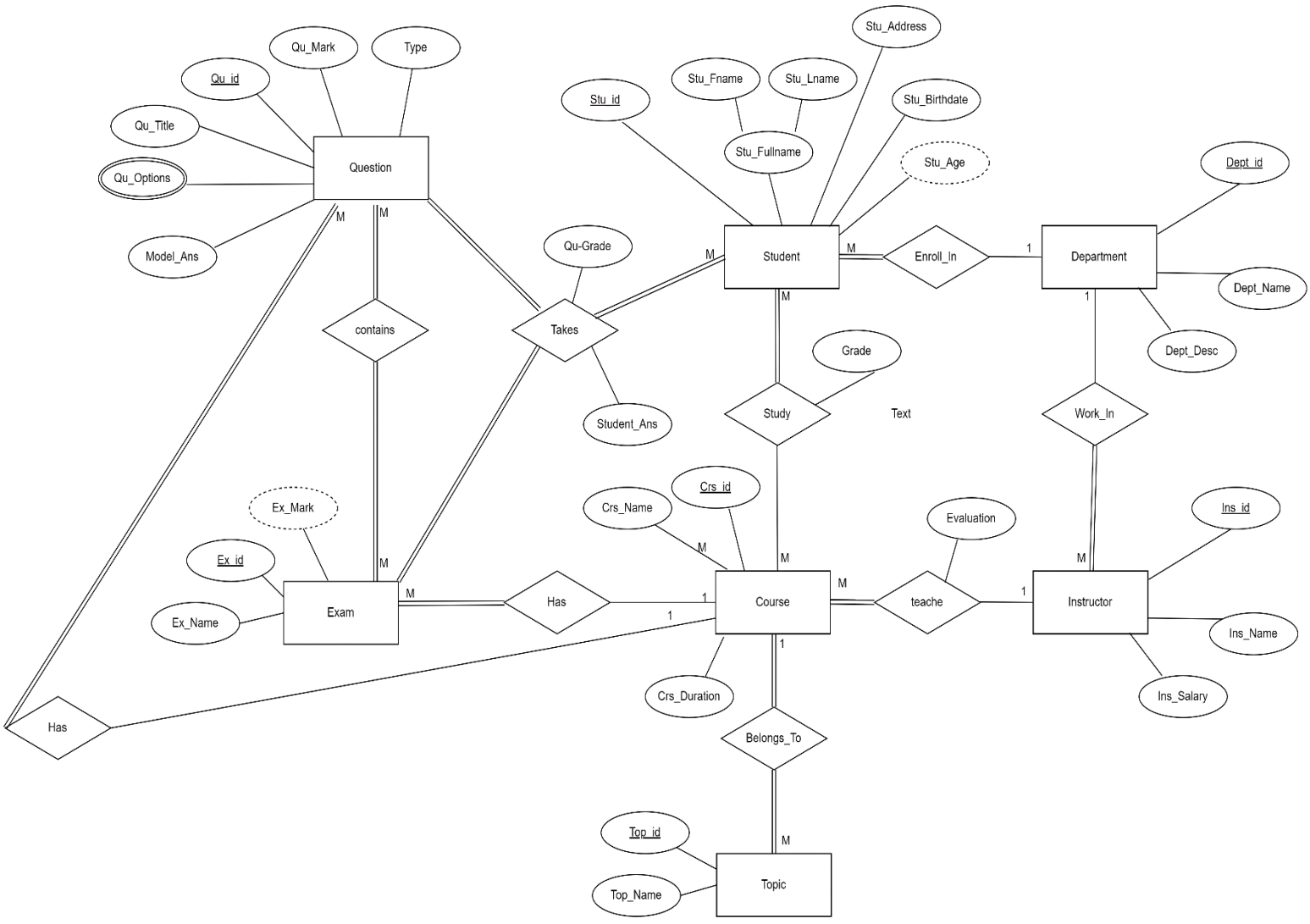
Examination System

In this Project, we aim to create an Examination System that generates random questions for students by creating our own Data Base, choosing our entities, their properties, and the relations between them, using SQL Server as our relational database management system.

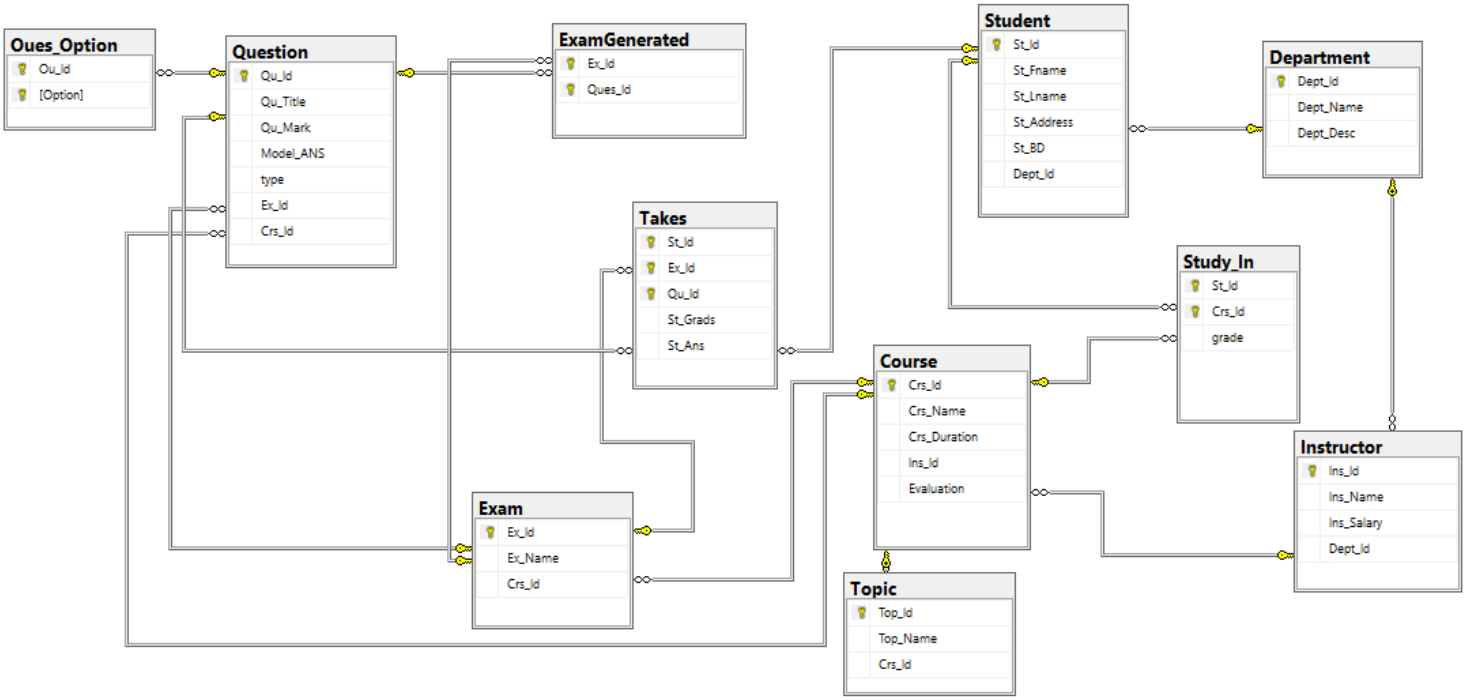
Our system consists of some students who are enrolled in a number of departments and take many courses with different topics. There are some instructors who teach these courses and have Evaluations. Our Exam contains some random questions that have two types, the first one is multiple-choice questions and the second one is True/False questions. Each course has a number of exams to evaluate every student using our exam correction system that enables students to get their grades in each course they are enrolled in.

During our implementation of this system, we went through a number of steps:

1- Drawing the Entity Relational Diagram as shown below:



2- Applying mapping our diagram into some tables, as shown below



As shown in the figure above, there are many tables that belong to two types:

- **<u>Main Tables which include:</u>**

- Each Student has a unique Id as a Primary key, First Name, Last Name, Address, and Birthdate.
- Each Instructor has a unique Id as a Primary key, Name, and Salary.
- Each Department has a unique Id as a Primary key, name, and Description.
- Each Course has a unique Id as Primary key, Name, Duration, and, Evaluation.
- Each Topic has a unique Id as a Primary key and Name.
- Each Question has a unique Id as a Primary key, Title(Body), type (MCQ, T/F), Mark, and model answer.
- Each Exam has a unique Id as a Primary key and Name.

- **<u>The tables and that resulted from the relations between the main tables:</u>**

- Students must belong to one Department and Department may include many students So we put the Department Id as a foreign key in the students' table.
- Instructors must belong to one Department and Department may include many instructors so we put the Department Id as a foreign key in the instructors' table.
- Students must take many Courses and Courses may be taken by many students which resulted in a new table called (Study in) with a new attribute (grade) that represents the grade of students in each course and two foreign keys (Course Id, Student Id).
- Many Courses must be taught by many instructors and one instructor may teach many courses so we put the Instructor Id as a foreign key in the course table.

- One course must have many topics and each topic must belong to one course so we put the Course Id as a foreign key in the Topics table.
- One course has many Exams and each Exam belongs to one course So we put the Course Id as a foreign key in the Exams Table.
- One course may have many questions and each question must belong to one course So we put the Course Id as a foreign key in the questions table.
- Many Exams must have many questions and many questions must belong to many exams which results in a new table called (Exam Generated) with two foreign keys (Exam id, Question Id).
- Many Students must take many Exams that must contain many questions which resulted in a new table called (Takes) with two new attributes (student grade, student answer) and three foreign keys (Student Id, Question Id, Exam Id).
- One question has many options and many options belong to one question which results in the table called(question option) with question Id as a foreign key in the new table.

3- Writing the procedures needed for the system which include:

1-CRUD operations for all entities and this include:

1-Student:

- **Add Student:** insert Student into Student Table by adding his/her id, First Name, Last Name, Address, Birth Date, and department Id.
- **Read Student:** Get Data for all students or specify only one student using student Id.
- **Edit Student:** Edit Student's Data using Student Id.
- **Delete Student:** Delete Student's Data using Student Id.

2-Instructor:

- **Add Instructor:** insert a new instructor into the instructor Table by adding his/her id, Name, Salary, and department Id.
- **Read Instructor:** Get Data for all instructors or specify only one instructor using instructor Id.
- **Edit Instructor:** Edit the instructor's Data using the instructor Id.
- **Delete Instructor:** Delete the instructor's Data using the instructor Id.

3-Department:

- **Add Department:** insert a new Department into Department Table by adding its id, Name, and Description.
- **Read Department:** Get Data for all Departments or specify only one department using Department Id.
- **Edit Department:** Edit Department's Data using Department Id.
- **Delete Department:** Delete Department's Data using Department Id.

4-Course:

- **Add Course:** insert a new Course into the Course table by adding its Id, Name, Duration, Evaluation, and instructor Id.
- **Read Course:** Get Data for all Courses or specify only one Course using Course Id.
- **Edit Course:** Edit the Course's Data using Course Id.
- **Delete Course:** Delete Course Data using Course Id.

5-Topic:

- **Add Topic:** insert a new Topic into the Topic table by adding its Id, Name, and Course Id.
- **Read Topic:** Get Data for all Topics or specify only one Topic using Topic Id.
- **Edit Topic:** Edit the Topic's Data using Topic Id.
- **Delete Topic:** Delete Topic Data using Topic Id.

6-Question:

- **Add Question:** insert a new Question into the Question table by adding its Id, Title, Mark, Type, Model Answer, Course Id, and Exam Id.
- **Read Question:** Get Data for all Questions or specify only one Question using Question Id.
- **Edit Question:** Edit the Question's Data using Question Id.
- **Delete Question:** Delete Question Data using Question Id.

7-Exam:

- **Add Exam:** insert a new Exam into the Exam table by adding its Id, Name, and Course Id.
- **Read Exam:** Get Data for all Exam or specify only one Exam using Question Id.
- **Edit Exam:** Edit the Exam's Data using Exam Id.
- **Delete Exam:** Delete Exam Data using Exam Id.

8-Study-in:

- **Add**: insert a new record into this table by adding Student's id, Course's Id, and Grade.
- **Read:** Get Data for all records or specify only one record using Student Id.
- **Edit:** Edit the record's Data using Student Id.
- **Delete:** Delete the record's Data using Student Id.

9-Takes:

- **Add:** insert a new record into this table by adding Student's id, Exam Id, Question Id, Student Answer, and Student Grade.
- **Read:** Get Data for all records or specify only one record using Student Id.
- **Edit:** Edit the record's Data using Student Id.
- **Delete:** Delete the record's Data using Student Id.

2-Some Procedures for the Exam Generation, Answers, and Correction and include:

1- Exam Generation:

A new exam will be generated by specifying the Course Id, Exam name, number of multiple choice questions, and number of true/false questions and the questions will be inserted randomly in the result set. This result set will appear in ExamGenertaed Table.

2- Exam Answers:

This Procedure is for inserting student answers into (Takes) Table by specifying the Student Id, Exam Id and writing the content of student answers to all questions in the exam.

3- Exam Correction

This procedure contains 3 main steps, the first one is to compare the student's answer to the model answer, the second one assigns 1 mark for the correct answer and zero marks for the false one and the third step is to calculate the sum of the answers for each student and the count of all answers and by dividing them, we will get the total grade of each student per course and will update it in (Study in) Table. And this will be specified using exam Id and student Id.

- **Some Procedures for the Report which include:**

1- Procedure that returns the student's information according to Department Id that the student is enrolled.
2- Procedure that takes the student ID and returns the grades of the student in all courses.
3- Procedure that takes the instructor ID and returns the name of the courses that he teaches and the number of students per course.
4- Procedure that takes course ID and returns its topics
5- Procedure that takes exam number and returns the Questions in it and its choices.
6- Procedure that takes exam number and the student ID then returns the Questions in this exam with the student answers.