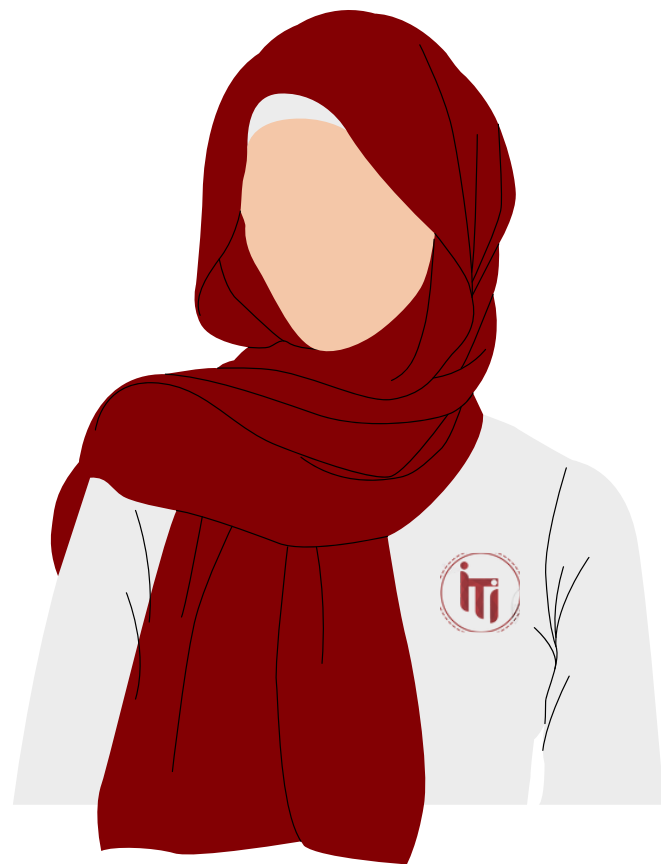


Examination System

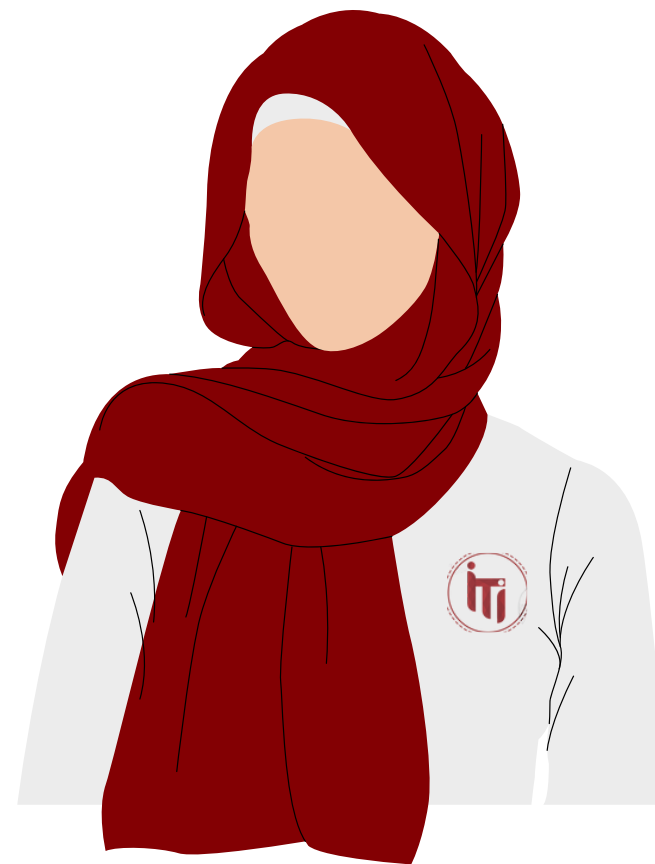
| 2025



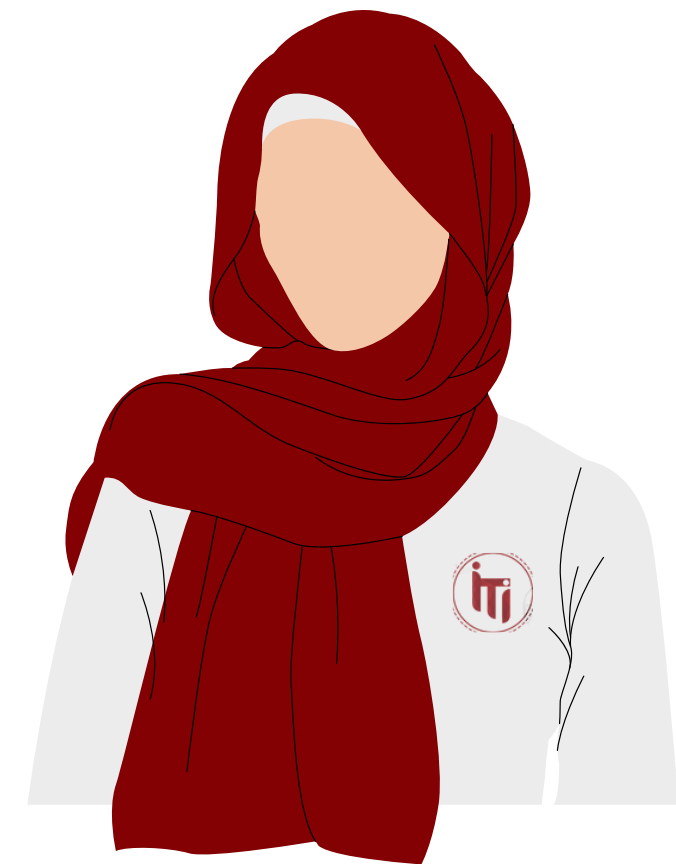
Team Members



Manar Hamada



Gharam Mokhtar

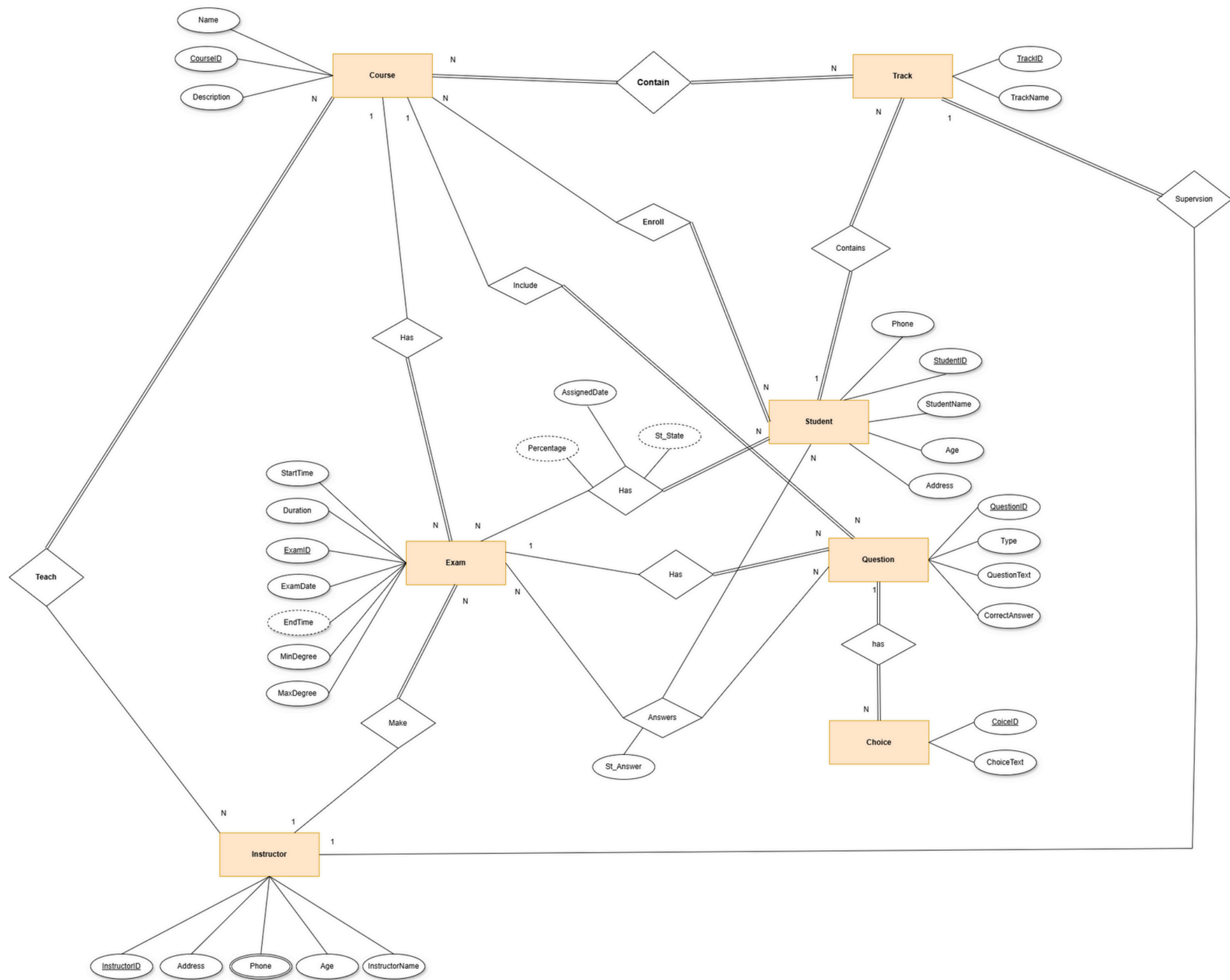


Mariam Hossam

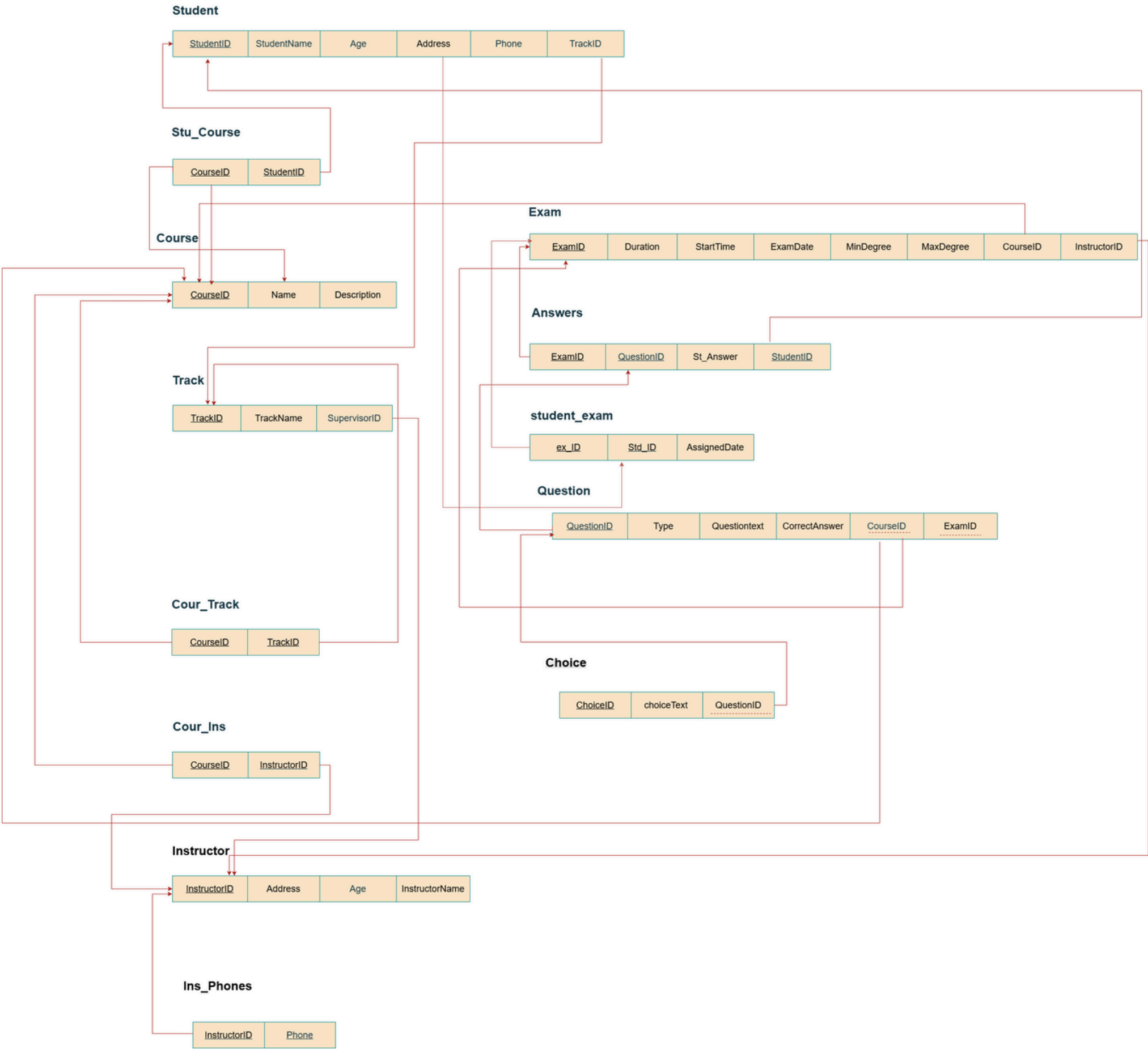
Requirements

- The system includes multiple tracks. Each track (TrackID, TrackName) contains one or more courses and is supervised by one instructor.
- Each course has a unique identifier, Name and Description.
- One or more instructors teach each course, and each instructor(InstructorID, InstructorName, Phone, Address, Age) may teach multiple courses.
- Each course(ExamID, ExamDate, StartTime, EndTime, Duration, MinDegree, MaxDegree) includes one or more exams.
- Each exam is prepared by one (and only one) instructor.
- Each exam contains one or more questions. A single question (QuestionID, Type, QuestionText, CorrectAnswer) may appear in more than one exam.
- Each question has one or more choices(ChoiceID, ChoiceText)
- Students (StudentID, StudentName, Age, Address, Phone) can enrol in one or more courses, and each course may have multiple students.
- Each student answers the questions in an exam and for each answer, the system stores:
 - The student's answer,
 - The grade the student achieved for the question.

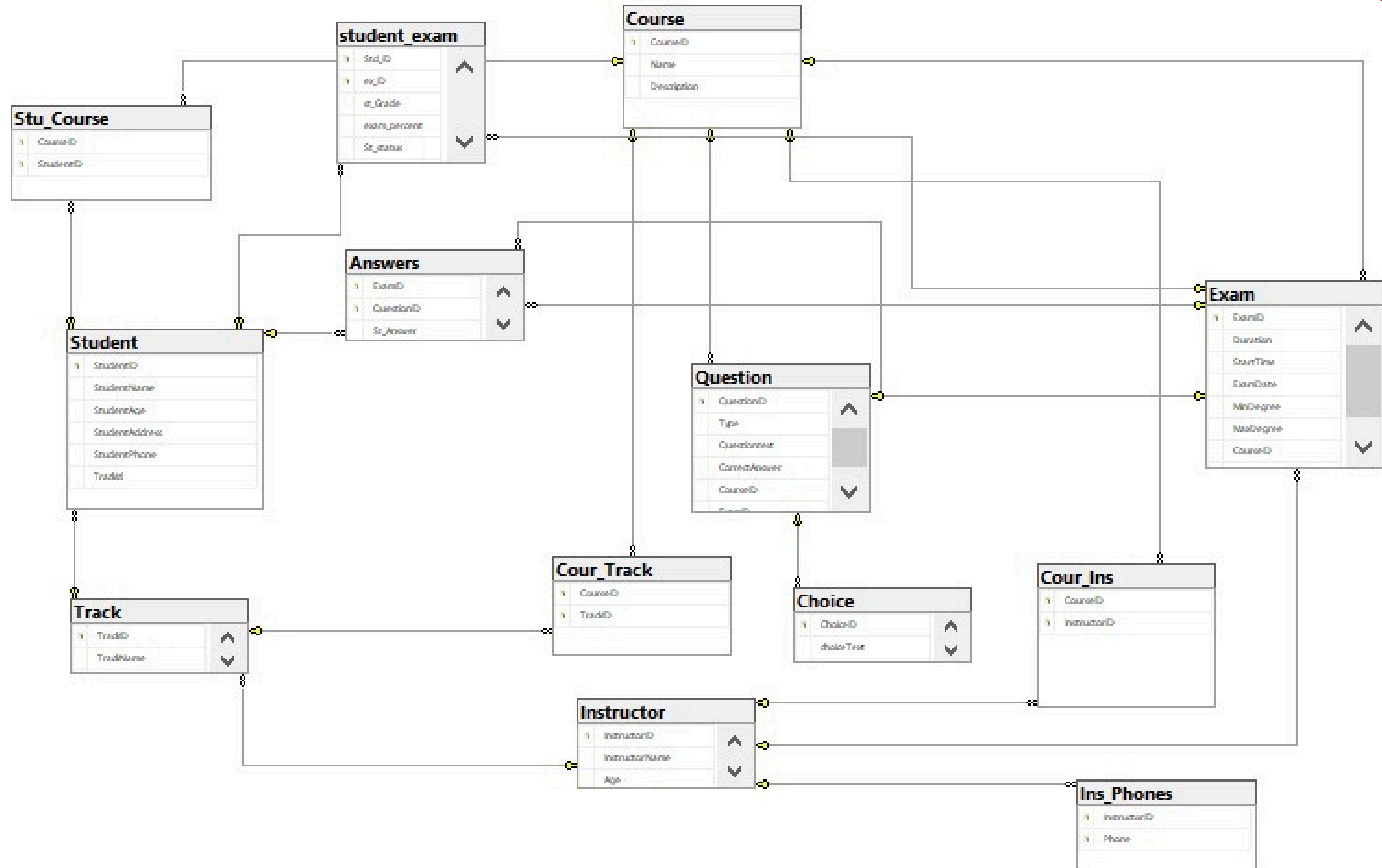
ERD



Mapping



Diagram



CreateExam PROCEDURE

```
CREATE PROCEDURE CreateExam
    @ExamID INT,
    @Duration INT,
    @StartTime TIME,
    @ExamDate DATE,
    @MinDegree INT,
    @MaxDegree INT,
    @CourseID INT,
    @InstructorID INT
AS
BEGIN
    INSERT INTO Exam (ExamID, Duration, StartTime, ExamDate, MinDegree, MaxDegree, CourseID, InstructorID)
    VALUES (@ExamID, @Duration, @StartTime, @ExamDate, @MinDegree, @MaxDegree, @CourseID, @InstructorID);
END

EXEC CreateExam
    @ExamID = 1,
    @Duration = 90,
    @StartTime = '11:00:00',
    @ExamDate = '2025-05-20',
    @MinDegree = 5,
    @MaxDegree = 10,
    @CourseID = 1,
    @InstructorID = 1;
select * from Exam
```

Add_Choices_To_Question PROCEDURE

```
CREATE PROCEDURE Add_Choices_To_Question
    @ChoiceID INT,
    @ChoiceText VARCHAR(50),
    @QuestionID INT
AS
BEGIN
    INSERT INTO Choice (ChoiceID, choiceText, QuestionID)
    VALUES (@ChoiceID, @ChoiceText, @QuestionID)
END
EXEC Add_Question_To_Exam
    @QuestionID = 3,
    @Type = 'MCQ',
    @QuestionText = 'What is the capital of France?',
    @CorrectAnswer = 'B',
    @CourseID = 1,
    @ExamID = 1;

EXEC Add_Choices_To_Question @ChoiceID = 5, @ChoiceText = 'Berlin', @QuestionID = 3;
EXEC Add_Choices_To_Question @ChoiceID = 2, @ChoiceText = 'Paris', @QuestionID = 3;
EXEC Add_Choices_To_Question @ChoiceID = 3, @ChoiceText = 'Rome', @QuestionID = 3;
EXEC Add_Choices_To_Question @ChoiceID = 4, @ChoiceText = 'Madrid', @QuestionID = 3;
```


Add_Question_To_Exam PROCEDURE

```
CREATE PROCEDURE Add_Question_To_Exam
```

```
    @QuestionID INT,  
    @Type VARCHAR(50),  
    @QuestionText VARCHAR(500),  
    @CorrectAnswer CHAR(1),  
    @CourseID INT,  
    @ExamID INT
```

```
AS
```

```
BEGIN
```

```
    INSERT INTO Question (QuestionID, Type, Questiontext, CorrectAnswer, CourseID, ExamID)  
    VALUES (@QuestionID, @Type, @QuestionText, @CorrectAnswer, @CourseID, @ExamID)
```

```
END
```

```
EXEC Add_Question_To_Exam
```

```
    @QuestionID = 2,  
    @Type = 'True/False',  
    @QuestionText = 'SQL is a programming language.',  
    @CorrectAnswer = 'F',  
    @CourseID = 1,  
    @ExamID = 1;
```

```
select * from Question
```

```
EXEC Add_Question_To_Exam @QuestionID = 1, @Type = 'MCQ', @QuestionText = 'What is the capital of USA?', @CorrectAnswer = 'B', @CourseID = 1, @ExamID = 1;
```

```
EXEC Add_Choices_To_Question @ChoiceID = 1, @ChoiceText = 'Washington DC', @QuestionID = 1;
```

```
EXEC Add_Choices_To_Question @ChoiceID = 2, @ChoiceText = 'New York', @QuestionID = 1;
```

```
EXEC Add_Choices_To_Question @ChoiceID = 3, @ChoiceText = 'Los Angeles', @QuestionID = 1;
```

```
EXEC Add_Choices_To_Question @ChoiceID = 4, @ChoiceText = 'Chicago', @QuestionID = 1;
```

```
EXEC Add_Question_To_Exam @QuestionID = 2, @Type = 'True/False', @QuestionText = 'The Earth is flat.', @CorrectAnswer = 'F', @CourseID = 1, @ExamID = 1;
```

Add_Student_Answer PROCEDURE

```
-- CREATE PROCEDURE Add_Student_Answer
    @ExamID INT,
    @QuestionID INT,
    @StudentID INT,
    @St_Answer CHAR(1)
AS
-- BEGIN
--     INSERT INTO Answers (ExamID, QuestionID, StudentID, St_Answer)
--     VALUES (@ExamID, @QuestionID, @StudentID, @St_Answer);
-- END;

EXEC Add_Student_Answer @ExamID = 1, @QuestionID = 1, @StudentID = 2, @St_Answer = 'B';
EXEC Add_Student_Answer @ExamID = 1, @QuestionID = 2, @StudentID = 2, @St_Answer = 'F';
EXEC Add_Student_Answer @ExamID = 1, @QuestionID = 3, @StudentID = 2, @St_Answer = 'A';
EXEC Add_Student_Answer @ExamID = 1, @QuestionID = 4, @StudentID = 2, @St_Answer = 'T';
EXEC Add_Student_Answer @ExamID = 1, @QuestionID = 5, @StudentID = 2, @St_Answer = 'C';
EXEC Add_Student_Answer @ExamID = 1, @QuestionID = 6, @StudentID = 2, @St_Answer = 'F';
EXEC Add_Student_Answer @ExamID = 1, @QuestionID = 7, @StudentID = 2, @St_Answer = 'B';
EXEC Add_Student_Answer @ExamID = 1, @QuestionID = 8, @StudentID = 2, @St_Answer = 'T';
EXEC Add_Student_Answer @ExamID = 1, @QuestionID = 9, @StudentID = 2, @St_Answer = 'D';
```

Exam_Correction PROCEDURE

```
ALTER PROCEDURE Exam_Correction
    @StudentID INT,
    @ExamID INT
AS
BEGIN
    DECLARE @TotalQuestions INT;
    DECLARE @CorrectAnswers INT;
    DECLARE @Grade INT;
    DECLARE @Percentage FLOAT;
    DECLARE @MinDegree INT;
    DECLARE @Duration INT;
    DECLARE @assigndate DATETIME;
    DECLARE @Studentname varchar(30);

    SELECT @TotalQuestions = COUNT(*)
    FROM Question
    WHERE ExamID = @ExamID
    IF @TotalQuestions = 0
    BEGIN
        RAISERROR('No questions found for this exam.', 16, 1);
        RETURN;
    END

    IF EXISTS (
        SELECT 1 FROM student_exam WHERE Std_ID = @StudentID AND ex_ID = @ExamID
    )
    BEGIN
        RAISERROR('Result for this student and exam already exists.', 16, 1);
        RETURN;
    END
END
```

```
SELECT @CorrectAnswers = COUNT(*)
FROM Answers A
INNER JOIN Question Q ON A.QuestionID = Q.QuestionID
WHERE A.StudentID = @StudentID AND A.ExamID = @ExamID
AND A.St_Answer = Q.CorrectAnswer;
SET @Grade = @CorrectAnswers;

SET @Percentage = (CAST(@CorrectAnswers AS FLOAT) / @TotalQuestions) * 100;

SELECT @MinDegree = MinDegree ,
    @Duration = duration,
    @assigndate = CAST(CONCAT(CONVERT(VARCHAR(10), ExamDate, 120), ' ', CONVERT(VARCHAR(8), StartTime, 108)) AS DATETIME)
FROM Exam WHERE ExamID = @ExamID;

SELECT @Studentname = StudentName
from Student
WHERE StudentID=@StudentID

INSERT INTO student_exam (Std_ID, StudentName, ex_ID, st_Grade, exam_percentage, St_status, AssignedDate, Duration)
VALUES (
    @StudentID,
    @Studentname,
    @ExamID,
    @Grade,
    @Percentage,
    CASE WHEN @Grade >= @MinDegree THEN 'Passed' ELSE 'Failed' END,
    @assigndate,
    @Duration
);
END;
EXEC Exam_Correction @StudentID = 1, @ExamID = 1;
```

student_exam PROCEDURE

```
CREATE TABLE student_exam (
    Std_ID INT,
    StudentName varchar(30),
    ex_ID INT,
    st_Grade INT,
    exam_percentage FLOAT,
    St_status VARCHAR(10),
    AssignedDate DATETIME DEFAULT GETDATE(),
    Duration INT,
    PRIMARY KEY (Std_ID, ex_ID),
    FOREIGN KEY (Std_ID) REFERENCES Student(StudentID),
    FOREIGN KEY (ex_ID) REFERENCES Exam(ExamID)
);
```

362 | SELECT * FROM student_exam

100 %

Results Messages

	Std_ID	StudentName	ex_ID	st_Grade	exam_percentage	St_status	AssignedDate	Duration
1	1	Ahmed Ali	1	2	14.2857142857143	Failed	2025-05-20 11:00:00.000	90
2	2	ahmed	1	9	64.2857142857143	Passed	2025-05-20 11:00:00.000	90
3	3	Maram Mohammed	1	6	42.8571428571429	Passed	2025-05-20 11:00:00.000	90
4	4	Fatma Ibrahim	1	6	42.8571428571429	Passed	2025-05-20 11:00:00.000	90
5	5	Galal Ahmed	1	6	42.8571428571429	Passed	2025-05-20 11:00:00.000	90
6	6	Farida Elsayed	1	6	42.8571428571429	Passed	2025-05-20 11:00:00.000	90
7	7	Amany Osama	1	8	57.1428571428571	Passed	2025-05-20 11:00:00.000	90
8	8	Karam Ali	1	8	57.1428571428571	Passed	2025-05-20 11:00:00.000	90


Data Dictionary

	TABLE_SCHEMA	TABLE_NAME	ORDINAL_POSITION	COLUMN_NAME	Full_Data_Type	Nullability	Key	FK_Constraint	Referenced_Table	Referenced_Column
1	dbo	Answers	1	ExamID	int	NOT NULL	PK	c6	Exam	ExamID
2	dbo	Answers	2	QuestionID	int	NOT NULL	PK	c7	Question	QuestionID
3	dbo	Answers	3	St_Answer	char(1)	NULL		NULL	NULL	NULL
4	dbo	Answers	4	StudentID	int	NOT NULL	PK	c8	Student	StudentID
5	dbo	Choice	1	ChoiceID	int	NOT NULL	PK	NULL	NULL	NULL
6	dbo	Choice	2	choiceText	varchar(50)	NULL		NULL	NULL	NULL
7	dbo	Choice	3	QuestionID	int	NULL		c5	Question	QuestionID
8	dbo	Cour_Ins	1	CourseID	int	NOT NULL	PK	FK__Cour_Ins__Course__619B8048	Course	CourseID
9	dbo	Cour_Ins	2	InstructorID	int	NOT NULL	PK	FK__Cour_Ins__Instru__628FA481	Instructor	InstructorID
10	dbo	Cour_Track	1	CourseID	int	NOT NULL	PK	FK__Cour_Trac__Cours__5DCAEF64	Course	CourseID
11	dbo	Cour_Track	2	TrackID	int	NOT NULL	PK	FK__Cour_Trac__Track__5EBF139D	Track	TrackID
12	dbo	Course	1	CourseID	int	NOT NULL	PK	NULL	NULL	NULL
13	dbo	Course	2	Name	nvarchar(30)	NULL		NULL	NULL	NULL
14	dbo	Course	3	Description	nvarchar(50)	NULL		NULL	NULL	NULL
15	dbo	Exam	1	ExamID	int	NOT NULL	PK	NULL	NULL	NULL
16	dbo	Exam	2	Duration	int	NULL		NULL	NULL	NULL
17	dbo	Exam	3	StartTime	time	NULL		NULL	NULL	NULL
18	dbo	Exam	4	ExamDate	date	NULL		NULL	NULL	NULL
19	dbo	Exam	5	MinDegree	int	NULL		NULL	NULL	NULL
20	dbo	Exam	6	MaxDegree	int	NULL		NULL	NULL	NULL
21	dbo	Exam	7	CourseID	int	NULL		c1	Course	CourseID
22	dbo	Exam	8	InstructorID	int	NULL		c2	Instructor	InstructorID
23	dbo	Ins_Phones	1	InstructorID	int	NOT NULL	PK	FK__Ins_Phone__Instr__5441852A	Instructor	InstructorID
24	dbo	Ins_Phones	2	Phone	varchar(20)	NOT NULL	PK	NULL	NULL	NULL
25	dbo	Instructor	1	InstructorID	int	NOT NULL	PK	NULL	NULL	NULL
26	dbo	Instructor	2	InstructorName	varchar(50)	NULL		NULL	NULL	NULL
27	dbo	Instructor	3	Age	int	NULL		NULL	NULL	NULL

PowerBI

TrackName

DE



Student according department

StudentID	StudentName	StudentPhone	StudentAddress	StudentAge	TrackName	TrackID
1	Ahmed Ali	01234567890	Cairo	21	DE	1
2	ahmed	01134567895	Alex	23	DE	1
3	Maram Mohammed	01159801236	Suez	22	DE	1
4	Fatma Ibrahim	01234897888	Banha	24	DE	1
5	Galal Ahmed	01025634897	Giza	25	DE	1
6	Farida Elsayed	01559874756	Alex	23	DE	1
7	Amany Osama	01224123698	Alex	22	DE	1
8	Karam Ali	01054789648	Cairo	24	DE	1
17	Kareem	01098765432	Cairo	22	DE	1
18	Laila	01234567890	Giza	21	DE	1
19	Omar	01156789345	Alexandria	23	DE	1
20	Hana	01567893421	Mansoura	22	DE	1
21	Youssef	01023456789	Tanta	25	DE	1
22	Farah	01187654321	Aswan	22	DE	1
23	Samir	01265437890	Port Said	20	DE	1
24	Noor	01534567892	Suez	24	DE	1
25	Hassan	01076543289	Fayoum	23	DE	1
26	Salma	01298765431	Ismailia	24	DE	1
27	Mahmoud	01123456788	Luxor	24	DE	1

student_info x

Exam Question

Instructor

Student answer

Stuent_grades

+

PowerBI

TrackName

DE



Instructor according Track

InstructorID	InstructorName	Address	Age	CourseID	Name
3	Ali Ibrahim	Alex	32	2	OS
1	Mohsen Mohammed	Alex	40	1	Networks
4	Osama Ahmed	Cairo	30	3	Docker
4	Osama Ahmed	Cairo	30	4	Linux

PowerBI

ExamID

All



Exam Questions

Questiontext

☐ **What does RAM stand for?**

Random Access Memory

Readily Available Memory

Remote Access Memory

Run Access Mode

☐ **What is the capital of USA?**

Chicago

Los Angeles

New York

Washington DC

☐ **What is the primary function of a firewall?**

Block unauthorized access

Increase internet speed

Manage user passwords

Store data securely

☐ **Which country is the largest by land area?**

Canada

Russia



student_info

Exam Questions

Instructor_according_track

Student answer

Stuent_grades



PowerBI

ExamID

1

StudentID

3



Student Answers

StudentID	False	True
3		
1		B
2	T	
3		A
4		T
5		C
6		F
7	A	
8		T
9	C	



ExamID
1

StudentID
3

Student Grades

StudentID	ExamID	St_status	st_Grade	exam_percentage
6	1	Failed	3	21.43
5	1	Failed	4	28.57
4	1	Passed	5	35.71
3	1	Passed	8	57.14
9	1	Passed	9	64.29
12	1	Passed	9	64.29
2	1	Passed	12	85.71
7	1	Passed	12	85.71
1	1	Passed	14	100.00
8	1	Passed	14	100.00



student_info

Exam Questions

Instructor_according_track

Student answer

Stuent_grades

