Information Technology Institute

Branch Ismailia

Examination System SQL Server

Examination System SQL Server

SD TRACK 2024-2025

Project Team

Alaa Eisa
Amr Hani
Alaa Ahmed
Marina Mansour
Veronica Wageh

Supervisor

Eng Rami

2024/2025

Table of Contents

1.Project Abstract

The **Examination System Database** is a robust and dynamic platform designed to manage and automate academic assessments for educational institutions. It integrates various entities and relationships to ensure seamless operation and organization. The primary focus is on supporting diverse academic workflows, ensuring secure access, and providing efficient data management for all stakeholders, including administrators, instructors, and students.

Core Features:

1. **Dynamic Question Pool**:

    - The database supports a centralized repository for questions categorized by type (Multiple Choice, True/False, Text).
    - Questions include associated metadata like grades, descriptions, and model answers for efficient evaluation.

2. **Course and Track Management**:

    - Administrators can organize courses into tracks, which are further associated with branches. This hierarchical structure ensures that courses align with institutional goals and are available in designated locations.

3. **Exam Construction**:

    - Instructors can create exams by selecting questions from the pool while adhering to predefined course requirements.
    - Exams are designed with specific details such as start/end times, duration, and associated tracks or branches.

4. **Student Management**:

    - Students can enroll in tracks and courses, with their personal details, academic progress, and exam participation meticulously recorded.
    - The system ensures that only eligible students can participate in specific exams, enforcing proper administration.

5. **Result Calculation and Evaluation**:

    - Exam results are automatically calculated based on student responses and question grades.
    - Advanced evaluation mechanisms, including text-based assessments with regular expressions, are used to ensure fair and accurate grading.

6. **Secure Access and Role Management**:

    - The system supports role-based access for Training Managers, Instructors, and Students.
    - A secure login mechanism ensures that users can only perform tasks within their defined roles.

Implementation:

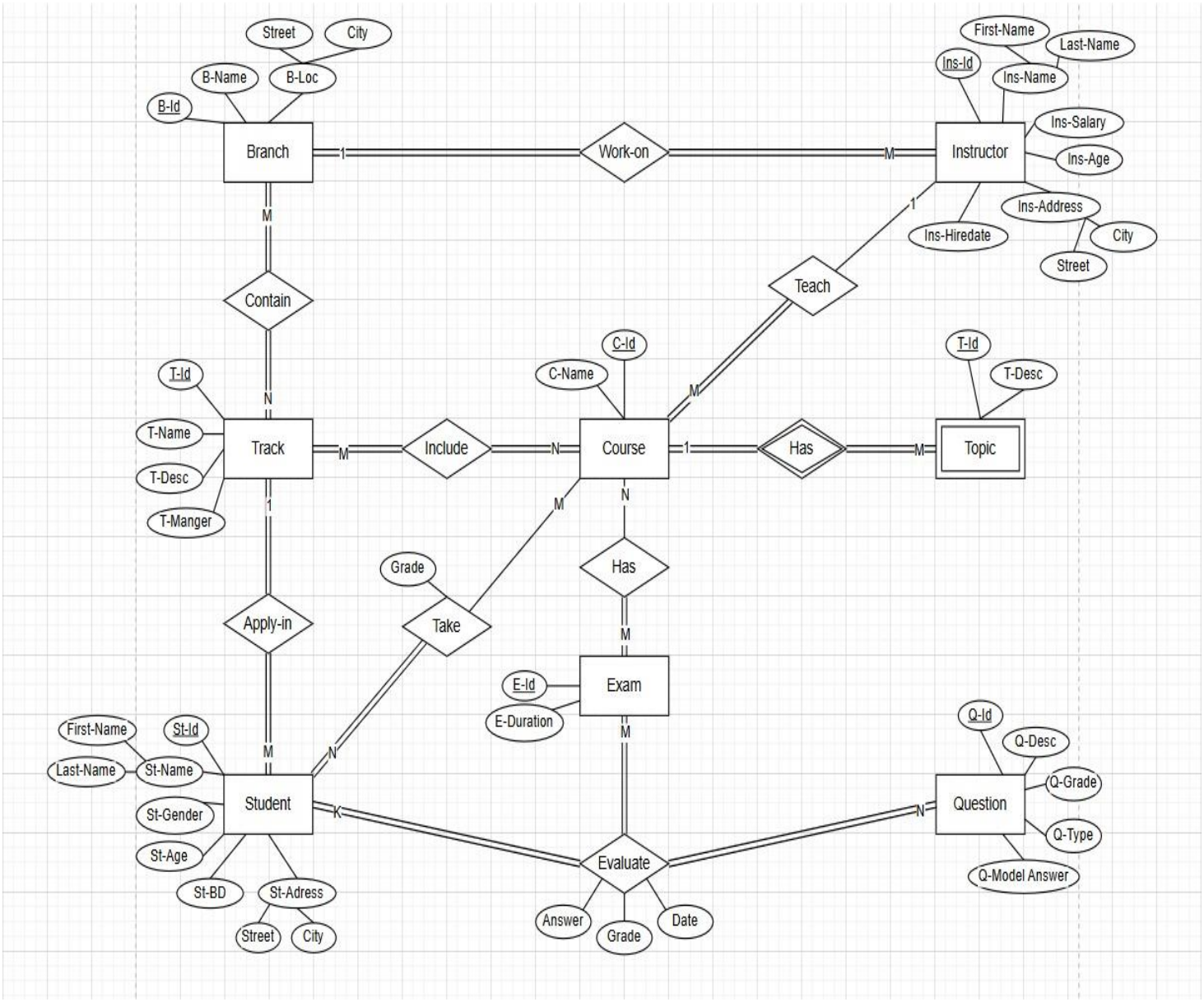
The system is built entirely using **SQL Server**, leveraging its powerful relational database capabilities for managing complex relationships. Entities such as Branches, Tracks, Courses, Exams, and Students are linked through well-defined relationships, ensuring a seamless data flow.

Benefits:

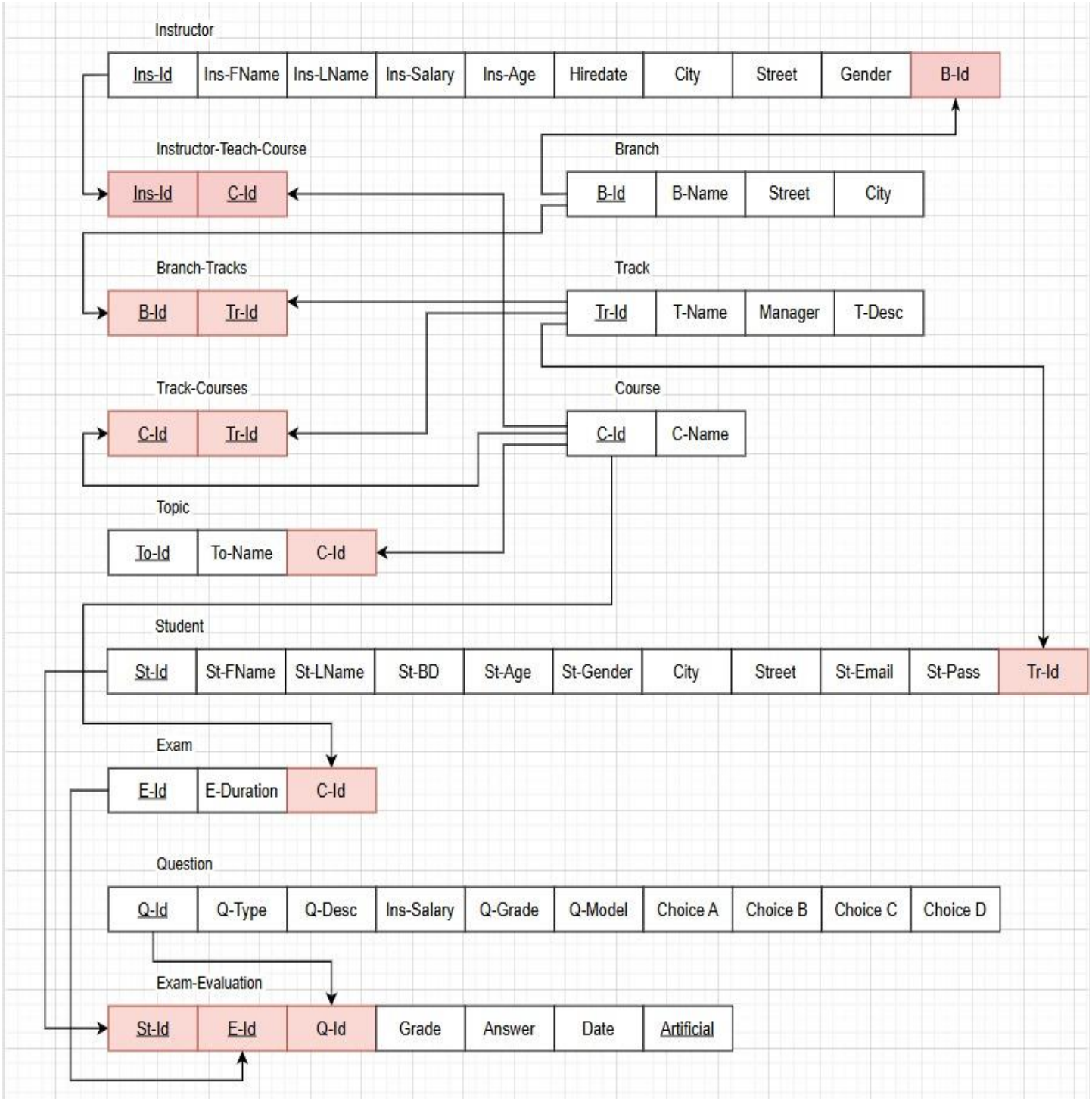
- **Efficiency:** Automates various academic and administrative processes, reducing manual workload.
- **Scalability:** Supports the addition of new branches, tracks, courses, and students without significant redesign.
- **Accuracy:** Ensures precise result tracking and evaluation using advanced database functionalities.
- **User-Friendly:** Provides clear roles for administrators, instructors, and students, simplifying operations.

By integrating these features, the Examination System Database establishes itself as a comprehensive solution for educational institutions to manage the complexities of academic assessments efficiently and securely.

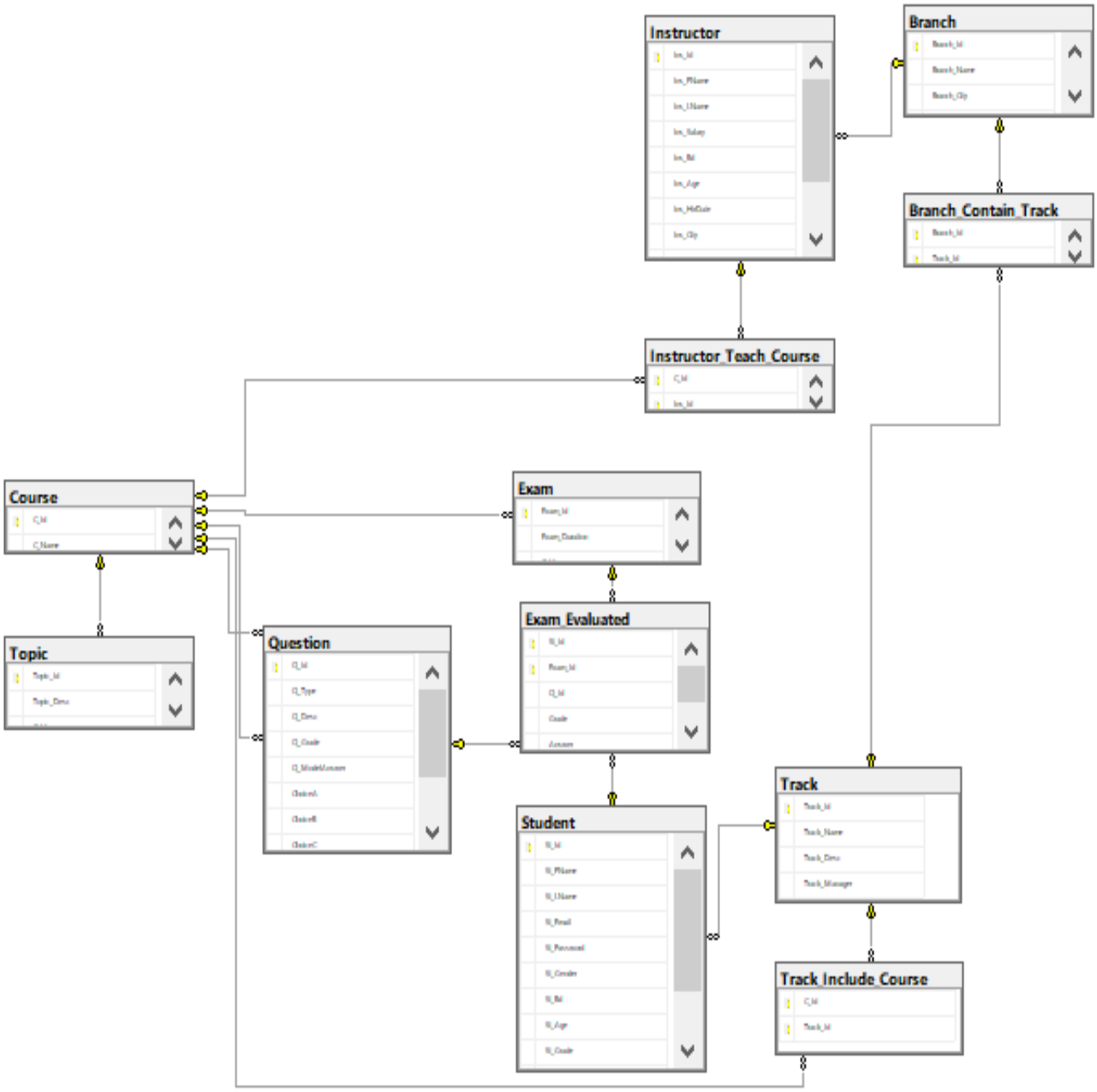
2.ERD


Street
City
B-Name
B-Loc
B-Id
Branch
1
Work-on
M
Instructor
First-Name
Last-Name
Ins-Id
Ins-Name
Ins-Salary
Ins-Age
Ins-Address
City
Street
Ins-Hiredate
1
Teach
M
M
Contain
N
T-Id
T-Name
T-Desc
T-Manger
Track
M
Include
N
Course
C-Id
C-Name
1
Has
M
Topic
T-Id
T-Desc
1
M
N
Has
M
Grade
Take
Apply-in
M
N
Exam
E-Id
E-Duration
M
First-Name
St-Id
Last-Name
St-Name
St-Gender
St-Age
St-BD
St-Adress
Street
City
Student
K
N
Evaluate
Answer
Grade
Date
Q-Id
Q-Desc
Q-Grade
Q-Type
Q-Model Answer
Question

3.Mapping


Instructor
Ins-Id
Ins-FName
Ins-LName
Ins-Salary
Ins-Age
Hiredate
City
Street
Gender
B-Id
Instructor-Teach-Course
Ins-Id
C-Id
Branch
B-Id
B-Name
Street
City
Branch-Tracks
B-Id
Tr-Id
Track
Tr-Id
T-Name
Manager
T-Desc
Track-Courses
C-Id
Tr-Id
Course
C-Id
C-Name
Topic
To-Id
To-Name
C-Id
Student
St-Id
St-FName
St-LName
St-BD
St-Age
St-Gender
City
Street
St-Email
St-Pass
Tr-Id
Exam
E-Id
E-Duration
C-Id
Question
Q-Id
Q-Type
Q-Desc
Ins-Salary
Q-Grade
Q-Model
Choice A
Choice B
Choice C
Choice D
Exam-Evaluation
St-Id
E-Id
Q-Id
Grade
Answer
Date
Artificial

4.Schema


Instructor
Branch
Branch_Contain_Track
Instructor_Teach_Course
Course
Exam
Topic
Question
Exam_Evaluated
Student
Track
Track_Include_Course

5.Table

- *[dbo].[Branch]*
- *[dbo].[Course]*
- *[dbo].[Branch_Contain_Track]*
- *[dbo].[Exam]*
- *[dbo].[Exam_Evaluated]*
- *[dbo].[Instructor]*
- *[dbo].[Instructor_Teach_Course]*
- *[dbo].[Student_Take_Course]*
- *[dbo].[Topic]*
- *[dbo].[Question]*
- *[dbo].[Student]*
- *[dbo].[Track_Include_Course]*

6. Stored Procedure

6.1 dbo.InsertNewBranch

- It takes input parameters such as:
@branch_id, @branch_name, @branch_city, @branch_street parameters.
- And check about @branch_id, if it exists print message “Branch Id already Exists”.
- Else create new branch.

6.2 dbo.InsertBranchContainTrack

- It takes input parameters such as:
@branch_id, @track_id parameters.
- And check about @branch_id and @track_id together, if they exist print message “This Branch-Track combination already exists”.
- Else check about @branch_id only and @track_id only and if they do not exist create new record.

6.3 dbo.UpdateBranch

- It takes input parameters such as:
@branch_id, @branch_name, @branch_city, @branch_street parameters.
- And check about @branch_id, if it exists update this branch.
- Else print message “Branch Id is not Exists”.

6.4 dbo. UpdateBranchContainTrack

- It takes input parameters such as:
@branch_id old, @track_id old, @branch_id new, @track_id new as parameters.
- And check about @branch_id old and @track_id old together, if they do not exist print message “The Branch-Track combination to update does not exist”.
- Else check about @branch_id new only and @track_id new only and if @branch_id new and @track_id new are exist together create new record.

6.5 dbo.DeleteBranch

- It takes input parameter @branch_id.
- And check about @branch_id, if it exists and have instructors print message “This branch have Instructor .. not deleted”.
- Else delete this branch.

6.6 dbo.DeleteBranchContainTrack

- It takes input parameters such as:
  @branch_id, @track_id parameters.
- And check about @branch_id and @track_id together, if they don't exist print message "This Branch-Track combination does not exist, cannot delete".
- Else delete this branch.

6.7 dbo.GetBranch

- It takes input parameter @branch_id.
- And check about @branch_id, if it exists display table.
- Else print message "This Branch does not exist".

6.8 dbo.GetBranchContainTrack

- It takes input parameter @branch_id.
- And check about @branch_id, if it exists display table.
- Else print message "This Branch does not exist".

6.9 dbo.InsertNewCourse

- It takes input parameters such as:
  @C_Id, @C_Name
- And check about @C_Id, if it exists print message "Course ID already Exists".
- Else create new course.

6.10 dbo.UpdateCourse

- It takes input parameters such as:
  @C_Id, @C_Name
- And check about @C_Id, if it exists update the name of this course.
- Else print message "Course doesn't Exists".

6.11 dbo. DeleteCourse

- It takes input parameters @C_Id.
- And check about @C_Id, if it exists delete this course,
- else print message “Course doesn’t Exists .. cannot deleted”.

6.12 dbo.GetCourse

- It takes input parameters @C_Id.
- And check about @C_Id, if it exists display this course,
- Else print message “Course doesn’t Exists”.

6.13 dbo.Insertinstractour

- It takes input parameters such as:
  @ins_id, @ins_fname, @ins_lname, @ins_salary, @ins_bd, @ins_city, @ins_street, @branch_id, @ins_gender.
- And check about @ins_id, if it exists print message “Instructor ID already exists”,
- Else create new Instructor.

6.14 dbo.InsertInstractourTeachCourse

- It takes input parameters such as:
  @ins_id, @C_id.
- And check about @ins_id only, if it doesn’t exist print message “Instructor ID does not exist”.
- And check about @C_id only, if it doesn’t exist print message “Course ID does not exist”.
- And check about @ins_id and @c_id together if they exist print message “This Instructor is already teaching this course”.

6.15 dbo.UpdateInstractor

- It takes input parameters such as:
  @ins_id, @ins_fname, @ins_lname, @ins_salary, @ins_bd, @ins_city, @ins_street, @branch_id, @ins_gender.
- And check about @ins_id, if it exists update instructor Information.
- Else print message “this id does not exist”.

6.16 dbo.UpdateInstractorteachCourse

- It takes input parameters such as:
  @ins_id old, @C_id old, @ins_id new, @C_id new.
- And check about @ins_id old and @C_id old together, if they don't exist print message "The Course-Instructor combination to update does not exist".
- Else check about @C_id new only, if it doesn't exist print message "The new Course ID does not exist".
- Else check about @ins_id new only, if it doesn't exist print message "The new Instructor ID does not exist".
- Else check about @ins_id new and @c_id new together if they exist print message "The new Course-Instructor combination already exists".
- Else update Instructor id and course id.

6.17 dbo.DeleteInstarctor

- It takes input parameters @ins_id.
- And check about @ins_id, if it exists delete it.
- Else print message "this id does not exist".

6.18 dbo. DeleteInstarctorTeachCourse

- It takes input parameters such as:
- @ins_id, @C_id.
- And check about @ins_id only, if it doesn't exist print message "Instructor ID does not exist".
- Else check about @C_id only, if it doesn't exist print message "Course ID does not exist".
- Else check about @ins_id and @C_id together, if they don't exist print message "This assignment does not exist".
- Else check about @ins_id and @c_id together if they exist, delete from table.

6.19 dbo.GetInstractour

- It takes input parameters @ins_id.
- And check about @ins_id, if it exists display it, else print message “this Instructor does not exist”.

6.20 dbo.GetInstractourTeachCourse

- It takes input parameters such as:
- @ins_id, @C_id.
- And check about @ins_id and @C_id, if they don’t exist print message “This Instructor-Course combination does not exist”.
- Else display the output.

6.21 dbo.InsertNewTopic

- It takes input parameters such as:
- @Topic_id, @Topic_desc, @C_id.
- And check about @Topic_id, if it exists print message “Topic ID already exists”.
- Else check about @C_id, if it doesn’t exist print message “This course ID does not exist”.
- Else insert new Topic and Description of this topic.

6.22 dbo.UpdateTopic

- It takes input parameters such as:
- @Topic_id, @Topic_desc, @C_id.
- And check about @Topic_id, if it exists:
- Check about @C_id, if it exists update @Topic_Desc and @C_id.
- Else print message “This course id does not exist”.
- Else if @Topic_id doesn’t exist, print message “Topic ID does not exist”.

6.23 dbo.DeleteTopic

- It takes input parameters @Topic_id.
- And check about @Topic_id, if it exists, delete this topic.
- Else print message “Topic ID does not exist”.

6.24 dbo.GetTopic

- It takes input parameters @Topic_id.
- And check about @Topic_id, if it exists display the output.
- Else print message “Topic ID does not exist”.

6.25 dbo.InsertNewTrack

- It takes input parameters such as:
- @Track_Id, @Track_Name, @Track_Desc, @Track_Manager.
- And check about @Track_id, if it exists, print message “Track ID already exists”.
- Else Insert new track id, track name, track desc and track manager.

6.26 dbo.UpdateTrack

- It takes input parameters such as:
- @Track_Id, @Track_Name, @Track_Desc, @Track_Manager.
- And check about @Track_id, if it exists, update track name, track desc and track manager.
- Else print message “Track ID already exists”.

6.27 dbo.DeleteTrack

- It takes input parameters @Track_id.
- And check about @Track_id, if it exists delete this track.
- Else print message “Track ID does not exist”.

6.28 dbo.GetTrack

- It takes input parameters @Track_id.
- And check about @Track_id, if it exists display the output.
- Else print message “Track ID does not exist”.

6.29 dbo.InsertTrackIncludeCourse

- It takes input parameters such as:
  @Track_id, @C_id.
- And check about @C_id only, if it doesn’t exist print message “Course ID does not exist”.
- Else check about @Track_id only, if it doesn’t exist print message “Track ID does not exist”.
- Else check about @Track_id and @C_id together, if they exist print message “The Course-Track combination already exists”.
- Else Insert new Track id and new Course id.

6.30 dbo.UpdateTrackIncludeCourse

- It takes input parameters such as:
  @Track_id old, @C_id old, @Track_id new, @C_id new.
- And check about @Track_id old and @C_id old together, if they don’t exist print message “The Course-Track combination to update does not exist”.
- Else check about @C_id new only, if it doesn’t exist print message “The new Course ID does not exist”.
- Else check about @Track_id new only, if it doesn’t exist print message “The new Track ID does not exist”.
- Else check about @Track_id new and @C_id new together if they exist print message “The new Course-Track combination already exists”.
- Else update Track id and Course id.

6.31 dbo.DeleteTrackIncludeCourse

- It takes input parameters such as:
  @Track_id, @C_id.
- And check about @Track_id and @C_id together, if they don't exist print message "The Course-Track combination to delete does not exist".
- Else it exists, delete from table.

6.32 dbo.GetTrackIncludeCourse

- It takes input parameters such as:
  @Track_id, @C_id.
- And check about @Track_id and @C_id together, if they don't exist print message "The Course-Track combination does not exist".
- Else display the table.

6.33 dbo.InsertNewStudent

- It takes input parameters such as:

  @st_id ,@st_fname, @st_lname, @st_email,@st_pass, @st_gender,

  @st_bd, @st_city,@st_stret, @t_id

- And check about @st_id, if it exists print message "student id is exists".
- Else insert new student.

6.34 dbo.UpdateStudent

- It takes input parameters such as:
  @st_id,@st_fname,@st_lname,@st_email,@st_pass,@st_gender,@st_bd,
  @st_city,@st_street , @t_id
- And Check if the student exists if not exist print to user “Student ID does not exist”.
- if exist check if track id is true if true Update the student record
- if not print to user “print track not exist”.

6.35 dbo.DeleteStudent

- It takes input parameters @id.
- And check if student is exist delete the record.
- if not print to user “this id does not exists”.

6.36 dbo.GetStudent

- It takes input parameters @id.
- And check if student is exist select his information.
- If not print to user “this dose not exists”.

6.37 dbo.InsertStudentTackCourse

- It takes input parameters such as:
  @C_Id , @St_Id , @Grade.
- And Check if the Course ID exists , Check if the Student ID exists , and finally
- Check if the Course-Student combination already exists if all are exist print
- 'The Course-Student combination already exists'
- Else Insert the new Course-Student combination with parameter information

6.38 dbo.UpdateStudentTackCourse

- It takes input parameters such as:
  @Old_C_Id , @Old_St_Id , @New_C_Id , @New_St_Id , @New_Grade.
- And first Check if the old Course-Student combination exists if not exist print to user “The Course-Student combination to update does not exist”.
- And if exist Check if the new Course ID exists.
- If not exist print to user “The new Course ID does not exist”.
- If exist Check if the new Student ID exists.
- If not print to user “The new Student ID does not exist”.
- If exist Check if the new Course-Student combination already exists.
- If exist print to user “The new Course-Student combination already exists”.
- If not Update the Course-Student combination and Grade.

7.39 dbo.DeleteStudentTackCourse

- It takes input parameters such as:
  @C_Id , @St_Id .
- And Check if the Course-Student combination exists.
- If exist delete the record.
- If not print to user “The Course-Student combination to delete does not exist”.

7.40 dbo.GetStudentTackCourse

- It takes input parameters such as:
  @C_Id , @St_Id .
- And Check if the Course-Student combination exists delete the record.
- If not print to user “The Course-Student combination does not exist”.

7.41 dbo.InsertQuestion

- It takes input parameters such as:
  @Q_Id , @Q_Type , @Q_Desc , @Q_Grade , @Q_ModelAnswer ,

@ChoiceA , @ChoiceB
@ChoiceC , @ChoiceD , @C_Id.

- And Check if the question already exists print to user 'Question already exists',
- If not check the question type ,
- If MCQ put answers , if not (T-F) set @ChoiceA= @ChoiceB= @ChoiceC= @ChoiceD=NULL,
- If other type insert print 'Wrong question type' .
- Finally if type is true insert the record .

6.42 dbo.UpdateQuestion

- It takes input parameters such as:
  @Q_Id , @Q_Type , @Q_Desc , @Q_Grade , @Q_ModelAnswer , @ChoiceA , @ChoiceB
  @ChoiceC , @ChoiceD , @C_Id.
- And check if the question exists Update it ,
- if not print 'Question ID does not exist'

6.43 dbo.DeleteQuestion

- It takes input parameters such as:
  @Q_Id .
- And check if the question exists delete it ,
- if not print 'This Question does not exist, cannot delete' .

6.44 dbo.GetQuestion

- It takes input parameters such as:
  @Q_Id .
- And check if the question exists get it ,
- If not print 'This Question does not exist' .

6.45 dbo.InsertNewExam

- It takes input parameters such as:
  @Exam_Id , @Exam_Duration, @C_Id.
- And Check if Exam ID already exists print 'Exam ID already exists',
- If not Check if Course ID exists if not exist print 'This course ID does not exist' ,
- And if course exist Insert new Exam .

6.46 dbo.InsertExamEvaluated

- It takes input parameters such as:
  @St_Id , @Exam_Id , @Q_Id , @Grade , @Answer , @Date .
- And check if student , exam and question exist insert new record with grade ,
- If any one of them not exist print 'This ID does not exist' .

6.47 dbo.UpdateExam

- It takes input parameters such as:

@Exam_Id ,@Exam_Duration , @C_Id .

- And check if exam exist check if course exist and update the record ,
- If course not exist print 'this course does not exist ' ,
- And if exam not exist print 'Exam does not exist' .

6.48 dbo.UpdateExamEvaluated

- It takes input parameters such as:
  @St_Id , @Exam_Id , @Q_Id , @Grade ,@Answer ,@Date , @artificial.
- And check if student , exam and questions are exist update the record ,
- If not print 'This record does not exist' .

6.49 dbo.DeleteExam

- It takes input parameters such as:
  @Exam_Id .
- And check if exam is exist delete it ,
- If not print 'This exam does not exist, cannot delete' .

6.50 dbo.DeleteExamEvaluated

- It takes input parameters such as:
  @St_Id ,@Exam_Id ,@Artificial .
- And Check if the record exists delete the record ,
- If not print 'This evaluation record does not exist, cannot delete' .

6.51 dbo.GetExam

- It takes input parameters such as:
  @Exam_Id .
- And check if exam is exist display it ,
- If not print 'This exam does not exist' .

6.52 dbo.GetExamEvaluated

- It takes input parameters such as:
  @St_Id , @Exam_Id , Artificial .
- And Check if the record exists display it ,
- If not print 'This evaluation record does not exist' .

6.53 dbo.Return_Std_Info

- it does not take parameter and return the students information according to Department

6.54 dbo.Std_Grad_by_Sid

- It takes input parameters such as:
- @id .
- And return the grades of the student in all courses .

6.55 dbo.Ins_Teach_Course

- It takes input parameters such as:
  @id .
- And return the name of the courses that he teaches
- and the number of student per course.

6.56 dbo.Course_Topics

- It takes input parameters such as:
  @id .
- And return the course topics .

6.57 dbo.Ques_In_Exam

- It takes input parameters such as:
  @exNum .
- And return the Questions in it and choices .

6.58 dbo.Std_Ans

- It takes input parameters such as:
  @exNum , @sId .
- And return the Questions in this exam with the student answers .

6.59 dbo. GenerateExam

- It takes input parameters such as:
  @id, @duration ,@mcq,@t_f,@ ,@sid
- And check if course is exist and id exam exists does not generate new Exam and if not check if student is exist get random questions from questions table Which belong to this course

6.60 dbo. ExamAnswers

- It takes input parameters such as:
  @eid , @sid , @answer1 ,@answer2 ,@answer3 , @answer4 , @answer5 , @answer6 , @answer7 , @answer8 , @answer9 , @answer10 .
- And set student answer in exam_Evalauted

6.61 dbo. ExamCorrection

- It takes input parameters such as:

@eid ,@sid

- And compare the student answer from exam in table exam with the correct answer of question in question table , if the answer is true sum the grade if not don’t sum