* **Triggers**
* **Name** 🡪 trg\_CheckCourseDegree
* **Description 🡪** This trigger will ensure that any new course inserted has valid Min\_Degree and should be greater than or equal 50 and Max\_Degree values should be greater than or equal to Min\_Degree.
* **Name 🡪** trg\_CheckInstructorEmail
* **Description** 🡪 This trigger will ensure that the email for a new instructor or at update existence instructor is unique.
* **Name🡪** trg\_CheckStudentEmailAndSupervisor
* **Description** 🡪 This trigger will ensure that the email for a new student or updated student is unique and the supervisor is a valid student ID.
* **Name** 🡪trg\_AssignInstructorToCourse
* **Description** 🡪 This trigger ensures that each instructor can teach one or more courses, and each course may be taught by one instructor per class.
* **Name** 🡪 trg\_CheckExamTime
* **Description** 🡪 this trigger used to check on exam start time and exam end time
* **Views**
* **Name** 🡪 Courses\_Teached\_ByInstructor
* **Description** 🡪 This view displays courses that teached by instructors.
* **Name🡪**Updates\_and\_AddsInBranch\_ByTriningManager
* **Description** 🡪 This view shows branches that added and updated in each department by each Training Manager.
* **Name🡪**Updates\_and\_AddsInTrack\_ByTriningManager
* **Description** 🡪 This view shows Tracks that added and updated in each department by each Training Manager.
* **Name** 🡪 Student\_Questions\_Answers
* **Description** 🡪 This view shows Questions and Question’s correct answers and Student answers for this question.
* **Name** 🡪 Student\_andExamsThat\_Tooken
* **Description** 🡪 This view shows students information that take part in Exam
* **Functions**
* **Name** 🡪GetIntakesByManager
* **Description 🡪** This inline table function displays intakes that added by each training manager.
* **Name** 🡪 GetStudentsByManager
* **Description 🡪** This inline functions display students that added by training manager in a system.
* **Name** 🡪 GetCourseDetails
* **Description 🡪** This function Take course id as parameter and get Course Name and Description for this course for this id.
* **Name** 🡪 GetStudentNameBYID
* **Description 🡪** Thisfunction takes student id as a parameter to get student full name for this id.
* **Name** 🡪 GetInstructorNameByID
* **Description 🡪** Thisfunction takes instructor id from user to get the instructor’s full name for this id.
* **Name** 🡪 GetManagerNameByID
* **Description 🡪** This function takes training manager id as a parameter and return Training Manager Name based on manager id.
* **Name** 🡪 GetExamDetails
* **Description 🡪**This function takes exam id as a parameter andreturns a table containing all details about a specific exam, including the questions.
* **Indexes**

**1. Department Indexes**

* **Name** 🡪 Dept\_name
* **Description 🡪** This indexer used to search for department name.
* **Name** 🡪 Dept\_location
* **Description 🡪** This indexer used for filtering data by department location.

**2. Training\_Manager Indexes**

* **Name** 🡪 Email
* **Description 🡪** used for efficient lookups by email.
* **Name** 🡪 Manager\_Fname
* **Description 🡪** used for searching by training manager’s first name.

**3. Branch Indexes**

* **Name** 🡪 Dept\_id
* **Description 🡪** used for joining with the Department table.
* **Name** 🡪 Barnch\_name
* **Description 🡪** used for searching for branch by name.

**4. Student Indexes**

* **Name** 🡪 Email
* **Description 🡪** For efficient lookups by student’s email.
* **Name** 🡪 Intake\_ID
* **Description 🡪** used for joining and filtering.
* **Name** 🡪 Track\_ID
* **Description 🡪** used for joining and filtering.
* **Name** 🡪 Branch\_ID
* **Description 🡪** used for joining and filtering.
* **Name** 🡪 Manager\_ID
* **Description 🡪** used for joining and filtering
* **Name** 🡪 Dept\_ID
* **Description 🡪** used for joining and filtering.

**5. Instructor Indexes**

* **Name** 🡪 Email
* **Description 🡪** used for efficient lookups by email.
* **Name** 🡪 Dept\_ID
* **Description 🡪** used for joining with the Department table.

**6. Instructor\_Courses Indexes**

* **Name** 🡪 Composite index on `Instructor\_ID` and `Course\_ID
* **Description 🡪** used for efficient lookups and joins.

**7. Question Indexes**

* **Name** 🡪 Q\_Text
* **Description 🡪** used for keyword searches.
* **Name** 🡪 INS\_ID
* **Description 🡪** used for filtering instructor by his id.

**8. Exam Indexes**

* **Name** 🡪 Course\_ID
* **Description 🡪** used for filtering courses by course id.
* **Name** 🡪 Instructor\_ID
* **Description 🡪** used for filtering by instructor.
* **Name** 🡪 Start\_Time
* **Description 🡪** usedfor filtering by time range.
* **Name** 🡪 End\_Time
* **Description 🡪** usedfor filtering by time range.

**9. Exam\_Question Indexes**

* **Name** 🡪 Exam\_ID
* **Description 🡪** used for filtering by exam
* **Name** 🡪 Q\_ID
* **Description 🡪** used for filtering by question.
* **Stored Procedures**
* **Name** 🡪 Insert Department
* **Description 🡪** used to insert data into department table.
* **Name** 🡪 Insert TrainngManger
* **Description 🡪** used to insert data into table Training manager.
* **Name** 🡪 AddNewBranch
* **Description 🡪** used to insert new branch into table branch and also insert new branch id and training manager id that add new branch into table Update\_adds\_inbranch.
* **Name** 🡪 UpdateBranch
* **Description 🡪** use to make any update in table branch and store training manager id that make update on branch and branch id that updated into table Update\_adds\_inbranch.
* **Name** 🡪 AddNewIntake
* **Description 🡪** insert new intake into table Intake by Training manager and also store manager id and intake id that added by training manager into table Intake\_addedby\_manager.
* **Name** 🡪 AddNewTrack
* **Description 🡪** used to add new track into table Track by Training manager and also store the manager id that add new track and track id that manager added into table Update\_adds\_inTrack.
* **Name** 🡪 UpdateTrack
* **Description 🡪** use to make update into table track by training manager and store update information like training manager id and track id which updated by training manager into table Update\_adds\_inTrack.
* **Name** 🡪 InsertCourse
* **Description 🡪** used to insert courses into table course.
* **Name**🡪 AddNewStudent
* **Description 🡪** use to insert new student into student table.
* **Name**🡪 InsertInstructor
* **Description 🡪** used to insert instructor’s data into instructor table.
* **Name**🡪 AssignInstructorToCourse
* **Description 🡪** used to insert courses that teached by specific instructor into table Instructor\_Courses.
* **Name**🡪 InsertQuestionWithAnswer
* **Description 🡪** Inserts a new question with its correct answer into the `Question` table.
* **Name**🡪 CreateExamWithQuestions
* **Description 🡪** Creates a new exam, allowing the instructor to select questions either manually or randomly. Ensures total degrees do not exceed the course's maximum degree.
* **Name**🡪 SP\_StudentExam
* **Description 🡪** This stored procedure insert into table StudentExams information about students that take specific exams.
* **Name**🡪 InsertStudentAnswer
* **Description 🡪** This stored procedure insert into table StudentAnswer information about Answers of students on exams.
* **Name**🡪 SP\_calculateTotalDegree
* **Description 🡪** This procedure used for calculate Student's Total Degree