**Database Schema Documentation**

This document provides an overview of the database schema, detailing the structure of tables and the relationships between them. Each section includes a description of the table, its columns, data types, and relationships.

1. **Courses Table**

CourseID (PK)

CourseName (NVARCHAR)

Description (NVARCHAR)

MaxDegree (Float)

MinDegree (Float)

1. **Instructors Table**

InstructorID (PK)

FirstName (NVARCHAR)

LastName (NVARCHAR)

Email (NVARCHAR)

CourseID (Foreign key referencing Courses)

1. **TrainingManager Table**

ID (PK)

FirstName (NVARCHAR)

LastName (NVARCHAR)

Email (NVARCHAR)

1. **Departments Table**

DepartmentID (Primary Key)

DepartmentName (NVARCHAR)

1. **Branches Table**

BranchID (Primary Key)

BranchName

DepartmentID (Foreign Key referencing Departments)

1. **Tracks Table**

TrackID (Primary Key)

TrackName

BranchID (Foreign Key referencing Branches)

1. **Intakes Table**

IntakeID (Primary Key)

IntakeName

BranchID (Foreign Key referencing Branches)

1. **Students Table**

StudentID (Primary Key)

FirstName (NVARCHAR)

LastName (NVARCHAR)

Email (NVARCHAR)

IntakeID (Foreign Key referencing Intakes)

BranchID (Foreign Key referencing Branches)

TrackID (Foreign Key referencing Tracks)

CourseID (Foreign key referencing Courses)

1. **Users**

userID (PK)

username (NVARCHAR)

password (NVARCHAR)

studentID (Foreign Key referencing Intakes)

instructorID (Foreign Key referencing Intakes)

TrainingManegerID (Foreign Key referencing Intakes)

1. **Questions Table**

QuestionID (Primary Key)

QuestionText (NVARCHAR)

QuestionType (Multiple Choice, True & False, Text)

CorrectAnswer (for Multiple Choice and True & False)

MultiAnswer (INT)

TrueFalseAnswer (NVARCHAR)

TextAnswer (NVARCHAR)

BestAcceptedAnswer (NVARCHAR)

CourseID (Foreign key referencing Courses)

1. **Exams Table**

ExamID (PK)

ExamType (Exam or Corrective) (VARCHAR)

StartTime (DATETIME)

EndTime (DATETIME)

TotalTime (INT)

AllowanceOptions (TEXT)

CourseID (Foreign Key referencing Courses)

InstructorID (Foreign Key referencing Instructors)

IntakeID (Foreign Key referencing Intakes)

BranchID (Foreign Key referencing Branches)

TrackID (Foreign Key referencing Tracks)

1. **ExamQuestions Table**

ExamQuestionID (PK)

Degree

ExamID (Foreign Key referencing Exams)

QuestionID (Foreign Key referencing Questions)

1. **StudentExams Table**

StudentExamID (Primary Key)

StartTime

EndTime

StudentID (Foreign Key referencing Students)

ExamID (Foreign Key referencing Exams)

1. **StudentAnswers Table**

StudentAnswerID (PK)

StudentAnswer

IsCorrect (calculated field)

IsValidAnswer (BIT)

Degree (referencing degree in QUESTIONS TABLE)

StudentExamID (Foreign Key referencing StudentExams)

QuestionID (Foreign Key referencing Questions)

**RELATIONSHIPS:**

* An instructor can teach many courses (one-to-many).
* A course can have many exams (one-to-many).
* An exam is associated with one course and one instructor (many-to-one).
* A student can take many exams (one-to-many).
* An exam can have many students (one-to-many).
* A question can be part of many exams (many-to-many).
* An exam question is associated with one exam and one question (many-to-one).

### **Procedure Documentation and Usage Guide**

The following document provides details and usage examples for several stored procedures related to user management, exam creation, question handling, student answers, and student exams in a university database. These procedures are designed to facilitate various tasks such as adding users, creating exams, adding questions, checking student answers, and managing student exams.

#### **1. TrainingManager.sp\_department**

This procedure adds a new department to the Departments table.

**Parameters:**

* @departmentName: The name of the department to be added (type: nvarchar).

**Usage:**



#### 2. **admin.SP\_AddUsers**

This procedure adds a new user to the Admin.Users table with optional IDs for Student, Instructor, or Training Manager.

**Parameters:**

* @Username: The username of the new user (type: NVARCHAR (50)).
* @Password: The password of the new user (type: NVARCHAR (255)).
* @StudentID: Optional, the ID of the student (type: INT, default is NULL).
* @InstructorID: Optional, the ID of the instructor (type: INT, default is NULL).
* @TrainingManagerID: Optional, the ID of the training manager (type: INT, default is NULL).

**Usage:**



#### 3. **instructor.spCreateExam**

This procedure creates a new exam and ensures the instructor is assigned to the specified course.

**Parameters:**

* @CourseID: The ID of the course (type: INT).
* @InstructorID: The ID of the instructor (type: INT).
* @ExamType: The type of exam (type: VARCHAR (50)).
* @IntakeID: The ID of the intake (type: INT).
* @BranchID: The ID of the branch (type: INT).
* @TrackID: The ID of the track (type: INT).
* @StartTime: The start time of the exam (type: DATETIME).
* @EndTime: The end time of the exam (type: DATETIME).
* @TotalTime: The total duration of the exam in minutes (type: INT).
* @AllowanceOptions: Any additional allowance options (type: NVARCHAR (255)).

**Usage:**

EXEC instructor.spCreateExam 2, 2, 'Exam', 1, 1, 4, '2023-12-01 10:00:00.000', '2023-12-01 12:00:00.000', 120, NULL

EXEC instructor.spCreateExam 3, 2, 'Exam', 3, 3, 2, '2024-06-14 7:00:00.000', '2024-06-14 9:00:00.000', 120, NULL

#### 4. **instructor.spAddQuestion**

This procedure adds a question to the instructor.Questions table.

**Parameters:**

* @CourseID: The ID of the course (type: INT).
* @QuestionText: The text of the question (type: NVARCHAR (100)).
* @QuestionType: The type of question (type: VARCHAR (50)).
* @MultiAnswer: Optional, the multiple-choice answer (type: INT, default is NULL).
* @TrueFalseAnswer: Optional, the true/false answer (type: TINYINT, default is NULL).
* @TextAnswer: Optional, the text answer (type: NVARCHAR (255), default is NULL).

**Usage:**

**A black background with green text

Description automatically generated**

#### 5. **instructor.spAddQuestionToExam**

This procedure associates a question with an exam in the ExamQuestions table.

**Parameters:**

* @ExamID: The ID of the exam (type: INT).
* @QuestionID: The ID of the question (type: INT).
* @Degree: The degree assigned to the question (type: DECIMAL(5, 2)).

Usage:



#### 6. **Student.SP\_checkSutudentAnswer**

This procedure checks a student's answer and inserts the result into Student.StudentAnswers.

**Parameters:**

* @StudentID: The ID of the student (type: INT).
* @studentExamID: The ID of the student exam (type: INT).
* @QuestionID: The ID of the question (type: INT).
* @studentanswer: The student's answer (type: NVARCHAR (255)).

**Usage:**

**EXEC Student.SP\_checkSutudentAnswer 2, 2, 1, 'AA'**

#### 7. **instructor. spAddRandomQuestionToExam**

This procedure randomly selects a question from the Questions table for a given course and adds it to an exam.

**Parameters:**

* @ExamID: The ID of the exam (type: INT).
* @CourseID: The ID of the course (type: INT).
* @Degree: The degree assigned to the question (type: DECIMAL (5, 2)).

Usage:



#### 8. **instructor.SP\_calculateTotalDegree**

This procedure calculates the total degree for a student in an exam and assigns a corrective exam if the score is less than 60%.

**Parameters:**

* @studentId: The ID of the student (type: INT).
* @examId: The ID of the exam (type: INT).

**Usage:**

A black background with pink and red text

Description automatically generated

These stored procedures provide comprehensive functionality for managing departments, users, exams, questions, and student responses within the university database system. Each procedure includes parameters that allow for flexible and specific operations as needed.

**SQL Server Triggers Documentation**

**1. Trigger: Student.trg\_CheckExamTime**

Description: This trigger ensures that a student can only take an exam within the allowed exam time. If the student's start or end time is outside the permitted exam time, the transaction is rolled back.

Schema: Student

Table: StudentExams

Event: AFTER INSERT

**2. Trigger: instructor.trg\_CheckExamTotalDegree**

Description: This trigger ensures that the total degree of all questions in an exam does not exceed the maximum allowed degree for the associated course. If it does, the transaction is rolled back.

Schema: instructor

Table: ExamQuestions

Event: AFTER INSERT, UPDATE

**3. Trigger: Student.trg\_PreventTakeExamTwice**

Description: This trigger ensures that a student cannot take the same exam more than once. If a student attempts to take the same exam again, the transaction is rolled back.

Schema: Student

Table: StudentExams

Event: AFTER INSERT

**SQL Server Functions Documentation**

**1. Function: Student.GetExamDetails**

Description: This function retrieves the details of a specified exam along with its questions.

Schema: Student

Function Name: GetExamDetails

Parameters: @ExamID (INT): The ID of the exam to retrieve details for.

Returns: A table with the following columns:

QuestionID (INT): The ID of the question.

QuestionText (NVARCHAR): The text of the question.

QuestionType (NVARCHAR): The type of the question.

Degree (DECIMAL): The degree assigned to the question in the exam.

**2. Function: instructor. GetStudentDetails**

Description: This function retrieves all details of a specified student, including their exams and the total degree for each exam.

Schema: instructor

Function Name: GetStudentDetails

Parameters: @StudentID (INT): The ID of the student to retrieve details for.

Returns: A table with the following columns:

StudentID (INT): The ID of the student.

FirstName (NVARCHAR): The first name of the student.

LastName (NVARCHAR): The last name of the student.

Email (NVARCHAR): The email of the student.

IntakeName (NVARCHAR): The name of the intake.

BranchName (NVARCHAR): The name of the branch.

TrackName (NVARCHAR): The name of the track.

ExamID (INT): The ID of the exam.

CourseID (INT): The ID of the course.

InstructorID (INT): The ID of the instructor.

ExamType (NVARCHAR): The type of the exam.

StartTime (DATETIME): The start time of the exam.

EndTime (DATETIME): The end time of the exam.

TotalTime (INT): The total time of the exam.

TotalDegree (DECIMAL): The total degree of the exam.

**3. Function: instructor. ValidateStudentAnswer**

Description: This function validates a student's answer by comparing it to the best accepted answer. It returns 1 (TRUE) if the student's answer matches the best accepted answer, and 0 (FALSE) otherwise.

This function is used in PROC SP\_checkStudentAnswer.

Schema: instructor

Function Name: ValidateStudentAnswer

Parameters:

@StudentAnswer (NVARCHAR(MAX)): The student's answer.

@BestAcceptedAnswer (NVARCHAR(MAX)): The best accepted answer.

Returns: BIT (1 for valid answer, 0 for invalid answer).

**SQL Server Views Documentation**

**1. View: vwStudentResults**

Description: This view aggregates student exam results, showing each student's obtained degree and total possible degree for each exam they have taken. The results are grouped by student and exam.

View Name: vwStudentResults

Columns:

StudentID (INT): The ID of the student.

StudentName (NVARCHAR): The full name of the student (concatenation of first name and last name).

ExamID (INT): The ID of the exam.

CourseID (INT): The ID of the course.

CourseName (NVARCHAR): The name of the course.

ObtainedDegree (DECIMAL): The total degree obtained by the student in the exam (calculated as the sum of degrees for correct answers).

TotalDegree (DECIMAL): The total possible degree for the exam.

**2. View: vwExams**

Description: This view displays details about exams, including the course name, instructor name, exam type, start and end times, total time, and allowance options.

View Name: vwExams

Columns:

ExamID (INT): The ID of the exam.

CourseName (NVARCHAR): The name of the course.

InstructorName (NVARCHAR): The full name of the instructor (concatenation of first name and last name).

ExamType (NVARCHAR): The type of the exam.

StartTime (DATETIME): The start time of the exam.

EndTime (DATETIME): The end time of the exam.

TotalTime (INT): The total time allocated for the exam.

The allowance options for the exam.

**Documentation for Index Creation and Management**

Overview

Indexes are critical for optimizing database performance, particularly for speeding up data retrieval operations. However, they come at the cost of additional storage space and may slow down data modification operations like insert, update, and delete.

This script creates, rebuilds, and manages indexes for various tables in the database to ensure efficient data access.

1.Instructor Table Indexes

2.TrainingManager Table Indexes

3.Departments Table Indexes

4. Branches Table Indexes

5. Tracks Table Indexes

6. Intakes Table Indexes

7. Students Table Indexes

8. Users Table Indexes

9. Questions Table Indexes

10. Exams Table Indexes

11. ExamQuestions Table Indexes

12. StudentExams Table Indexes

13. StudentAnswers Table Indexes

To analyze the performance of a query, you can enable STATISTICS IO and STATISTICS TIME to gather detailed metrics on logical reads, physical reads, and the time taken to execute the query.

 **SET STATISTICS IO ON**: This command provides information about the amount of disk activity generated by T-SQL statements.

 SET **STATISTICS TIME ON**: This command displays the amount of time required to parse, compile, and execute each T-SQL statement.

 SET **STATISTICS IO OFF** and **SET STATISTICS TIME OFF**: These commands turn off the statistics collection after the query is executed.