

Exam System DBMS

Database Documentation

Schemas

1. Training Schema

Stores data related to the structure of the organization, including students, courses, branches, and tracks.

- **Tables:**

- Student
- Course
- student_course
- Branch
- Track
- branch_track
- Track_Course
- Intake
- Instructor

- **Stored Procedures (SPs):**

- SP_AddStudent
- SP_AddCourse
- SP_AddBranch
- SP_AddTrack
- SP_AddTrackToBranch
- SP_AddCourseToTrack

- `SP_AddInstructor`
 - `SP_AddIntake`
 - `SP_AssignInstructorToCourse`
-

2. Exam Schema

Handles data related to exams, questions, and results.

- **Tables:**

- `Exam`
- `exam_questions`
- `Question`
- `question_types`
- `mcq_choices`
- `exam_results`

- **Stored Procedures (SPs):**

- `SP_AddQuestion`
 - `SP_AddMcqChoice`
 - `SP_CreateRandomExam`
 - `SP_AddExamQuestions`
 - `SP_AssignStudentsToExam`
 - `SP_CorrectAnswers`
 - `SP_CalculateExamResults`
 - `SP_GetQuestionsByCourse`
 - `SP_ShowStudentInfo`
 - `SP_GetExamResults`
-

3. Student Schema

Tracks student actions and results.

- **Tables:**
 - `Student_Answers`
 - **Stored Procedures (SPs):**
 - `SP_ViewExams`
 - `SP_GetExamQuestions`
 - `SP_SolveQuestion`
-

Roles and Authorization

Roles:

1. `admin` :

- **Permissions:** Full control over all schemas and objects.
- **Grant Script:**

```
grant control on database::exam_system to admin;
```

2. `training_manager` :

- **Permissions:** Manage training-related data, including adding courses, branches, tracks, and intakes.
- **Grant Script:**

```
grant select, insert, update, delete on schema::training to training_manager;
grant select on schema::exam to training_manager;
grant select on schema::student to training_manager;
```

3. `instructor` :

- **Permissions:** View and manage their assigned courses and exams.
- **Grant Script:**

```
grant select, insert, update, delete on schema::exam to
instructor;
grant select on schema::training to instructor;
grant select on schema::student to instructor;
```

4. **student**:

- **Permissions:** Limited access to view exams and answer questions.
- **Grant Script:**

```
grant insert on schema::student to student;
grant select on schema::exam to student
```

Tables

1. Training Schema

1. **Branch**:

- Columns: `branch_id`, `branch_name`, `location`

2. **Instructor**:

- Columns: `inst_id`, `inst_name`, `email`, `phone`, `branch_id`

3. **Course**:

- Columns: `crs_id`, `crs_name`, `description`, `max_degree`, `min_degree`, `inst_id`

4. **Track**:

- Columns: `track_id`, `track_name`

5. **Track_Course**:

- Columns: `track_id`, `crs_id`

6. **Branch_Track** :
 - Columns: `track_id`, `branch_id`
 7. **Intake** :
 - Columns: `intake_id`, `date`, `track_id`
 8. **Student** :
 - Columns: `std_id`, `fname`, `lname`, `email`, `phone`, `age`, `address`, `intake_id`, `branch_id`, `track_id`
 9. **Student_Course** :
 - Columns: `std_id`, `crs_id`
-

2. Exam Schema

1. **Question_Types** :
 - Columns: `type_id`, `type_name`
 2. **Question** :
 - Columns: `question_id`, `question_text`, `correct_answer`, `crs_id`, `question_type`
 3. **MCQ_Choices** :
 - Columns: `choice_id`, `choice_text`, `is_correct`, `question_id`
 4. **Exam** :
 - Columns: `exam_id`, `no_mcq_ques`, `no_text_ques`, `no_tf_ques`, `degree_mcq_ques`, `degree_text_ques`, `degree_tf_ques`, `exam_type`, `start_time`, `end_time`, `total_time`, `crs_id`, `inst_id`, `intake_id`, `branch_id`, `track_id`
 5. **Exam_Questions** :
 - Columns: `exam_question_id`, `degree`, `exam_id`, `question_id`
 6. **Exam_Results** :
 - Columns: `result_id`, `total_degree`, `exam_id`, `std_id`
-

3. Student Schema

1. **Student_Answers :**

- Columns: `answer_id` , `text_answer` , `is_correct` , `exam_id` , `std_id` , `question_id` , `mcq_answer`
-

Stored Procedures and Functions

1. Training Schema

Stored Procedures

SP_AddCourse

- **Parameters:** `@crs_name` , `@description` , `@max_degree` , `@min_degree` , `@inst_id`
- **Purpose:** Adds a new course to the `Course` table.

SP_AddInstructor

- **Parameters:** `@inst_name` , `@email` , `@phone` , `@branch_id`
- **Purpose:** Adds a new instructor to the `Instructor` table.

SP_AssignInstructorToCourse

- **Parameters:** `@course_id` , `@instructor_id`
- **Purpose:** Assigns an instructor to a course by updating the `Course` table.

SP_AddBranch

- **Parameters:** `@branch_name` , `@location`
- **Purpose:** Adds a new branch to the `Branch` table.

SP_AddTrackToBranch

- **Parameters:** `@track_id` , `@branch_id`
- **Purpose:** Links a track to a branch in the `Branch_Track` table.

SP_AddTrack

- **Parameters:** @track_name
- **Purpose:** Adds a new track to the Track table.

SP_AddCourseToTrack

- **Parameters:** @crs_id , @track_id
- **Purpose:** Links a course to a track in the Track_Course table.

SP_AddIntake

- **Parameters:** @date , @track_id
- **Purpose:** Adds a new intake for a track in the Intake table.

SP_AddStudent

- **Parameters:** @fname , @lname , @email , @phone , @age , @address , @intake_id , @branch_id , @track_id
- **Purpose:** Adds a new student to the Student table and enrolls them in courses linked to their track.

Functions

get_course_count_for_instructor

- **Parameters:** @instructor_id
- **Purpose:** Returns the total number of courses assigned to a specific instructor.

get_instructor_name

- **Parameters:** @instructor_id
- **Purpose:** Retrieves the name of a specific instructor based on their ID.

2. Exam Schema

Stored Procedures

SP_AddQuestion

- **Parameters:** @question_text , @correct_answer , @crs_id , @question_type
 - **Purpose:** Adds a new question to the exam.Question table, validating that the question text is not null.
-

SP_AddMcqChoices

- **Parameters:** @choice_text , @is_correct , @question_id
 - **Purpose:** Inserts a choice for a specific MCQ question into the exam.mcq_choices table.
-

SP_CreateRandomExam

- **Parameters:** @crs_id , @no_mcq , @degree_mcq_ques , @no_tf , @degree_tf_ques , @no_text , @degree_text_ques , @exam_type , @start_time , @end_time , @total_time
 - **Purpose:** Creates a randomized exam with questions pulled from the database based on the specified course and question criteria.
-

SP_GetExamResults

- **Parameters:** @exam_id
 - **Purpose:** Retrieves the total score for each student who took a specific exam, along with their names.
-

SP_GetQuestionsByCourse

- **Parameters:** @crs_id
 - **Purpose:** Fetches all questions for a specified course, including their text and type.
-

SP_AssignStudentsToExam

- **Parameters:** @exam_id , @student_ids
 - **Purpose:** Assigns a list of students to an exam and creates corresponding records in exam.exam_results .
-

SP_AddExamQuestions

- **Parameters:** @exam_id , @question_id , @degree
 - **Purpose:** Adds a question to a specific exam, including its assigned degree.
-

SP_CorrectAnswers

- **Parameters:** @exam_id
 - **Purpose:** Updates the correctness of students' answers for a given exam based on the predefined answers in the database.
-

SP_CalculateExamResults

- **Parameters:** @exam_id
 - **Purpose:** Calculates and updates the total score for each student who took a specific exam.
-

SP_ShowStudentInfo

- **Parameters:** @std_id
 - **Purpose:** Retrieves detailed information about a student, including name, email, phone, age, address, and related academic details (intake, branch, track).
-

Functions

get_no_available_question

- **Parameters:** @question_type
 - **Purpose:** Returns the number of available questions in the database for a specific question type.
-

3. Student Schema

Stored Procedures

SP_ViewExams

- **Parameters:** @std_id
 - **Purpose:** Retrieves a list of exams assigned to a student, including course name, start time, and end time.
-

SP_GetExamQuestions

- **Parameters:** @std_id , @exam_id
 - **Purpose:** Fetches the questions for a specific exam assigned to a student, categorized by question type (MCQ, True/False, and Text). Validates the exam's accessibility and ensures the student is authorized to view it.
-

SP_SolveQuestion

- **Parameters:** @std_id , @exam_id , @question_id , @answer
- **Purpose:** Allows a student to submit an answer for a specific question in an exam. Validates if the student has already answered the question and determines the correct field for the answer based on the question type.