Software Requirements Specification (SRS)

# Examination System Database

## 1. Introduction

### 1.1 Purpose

The purpose of this document is to define the functional, non-functional, and technical requirements for the Examination System Database.

This system will support the management and execution of exams within an educational institution, covering exam generation, student participation, automatic and manual evaluation, and result storage.

### 1.2 Scope

This project focuses on building a SQL Server-based system to:  
- Manage users (admin, training manager, instructors, students…….).  
- Generate and conduct exams.  
- Store course, question, and exam data.  
- Automatically evaluate certain question types.  
- Provide tools for instructors to manually assess and enter marks for subjective answers.  
- Support administrative operations like managing branches, tracks, and intakes.

### 1.3 Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| Term | Definition |
| SRS | Software Requirements Specification |
| DB | Database |
| ERD | Entity Relationship Diagram |
| T/F | True/False |

## 2. Overall Description

### 2.1 Product Perspective

This system is a standalone backend solution.

No frontend/UI is in scope.

Interaction will be via stored procedures, views, and SQL accounts.

### 2.2 Product Functions

- Manage users and permissions.  
- Manage educational hierarchy (branches, tracks, intakes).  
- Manage instructors, courses, and students.  
- Create and store exams.  
- Auto-grade multiple choice and T/F questions.  
- Store and partially assess text answers using string functions and regex.  
- Record and display student results.

### 2.3 User Classes and Characteristics

|  |  |
| --- | --- |
| User | Role |
| Admin | System-level tasks and configuration |
| Training Manager | Manages intakes, branches, tracks, and student data |
| Instructor | Creates and manages exams, assesses students |
| Student | Takes exams at scheduled times |

### 2.4 Operating Environment

- Microsoft SQL Server (2022)  
- SQL Server Management Studio (SSMS) for interaction

## 3. Functional Requirements

### 3.1 User Management

- Users must be authenticated via login.  
- Four roles: Admin, Training Manager, Instructor, Student.  
- Users can only access role-specific features.

### 3.2 Course and Question Management

- Courses have: name, description, max degree, min degree.  
- Each instructor can teach multiple courses.  
- Question types: Multiple Choice, True/False, Text.  
- Question pool must support tagging and random/manual selection.

### 3.3 Exam Management

- Instructors can create exams for their courses.  
- Exams may be manual or auto-generated from the pool.  
- Exams must define metadata (type, time, course, intake, etc.).  
- Each question must have an assigned degree not exceeding the course’s max.

### 3.4 Exam Execution

- Students can only access exams at scheduled times.  
- System auto-saves and grades objective answers.  
- Instructors manually review and enter marks for text answers.

### 3.5 Result Storage

- Store student answers and results.  
- Allow display of valid/invalid subjective answers.

### 3.6 Administrative Tools

- Admin can configure user roles and system settings.  
- Training manager manages branches, tracks, intakes, and students.

## 4. Non-Functional Requirements

### 4.1 Performance

Use indexing for optimal query performance.

### 4.2 Reliability & Availability

System should be available 24/7. Daily automatic backup required.

### 4.3 Security

Role-based access control. SQL permissions assigned per user type.

### 4.4 Maintainability

Modular stored procedures, views, and functions. Clear naming conventions.

## 5. Database Design Requirements

### 5.1 Implementation Details

- Use filegroups based on data types and estimated size.  
- Use appropriate datatypes.  
- Apply constraints and triggers to ensure data integrity.

### 5.2 Stored Procedures & Functions

- CRUD operations for all main entities.  
- Exam generation and grading logic.  
- Result calculation functions.

### 5.3 Views

- Create user-specific views to hide raw queries.  
- Allow filtering/searching results by various criteria.

## 6. Deliverables

|  |  |
| --- | --- |
| Deliverable | Description |
| Requirement Sheet | Original system specification |
| ERD | Diagram of the full data model |
| Database Files | MDF, LDF with filegroups |
| SQL Scripts | Per user and full database |
| Object Descriptions | Text file documenting DB objects |
| Test Sheets | Query, expected results, comments |
| Accounts File | All DB usernames and passwords |

# Full Database Tables Summary - ITI Examination System

## Table: Department

Description: Stores department names.

|  |  |
| --- | --- |
| Column Name | Data Type |
| dept\_id | int (PK) |
| dept\_name | nvarchar(255) |

## Table: Intake

Description: Represents different student intakes.

|  |  |
| --- | --- |
| Column Name | Data Type |
| int\_ID | int (PK) |
| int\_name | nvarchar(255) |
| s\_date | date |
| e\_date | date |

## Table: User

Description: Login credentials and roles of users.

|  |  |
| --- | --- |
| Column Name | Data Type |
| acc\_ID | int (PK) |
| user\_name | nvarchar(255) |
| password | nvarchar(255) |
| role | char |

## Table: Training\_Manager

Description: Stores training manager personal info.

|  |  |
| --- | --- |
| Column Name | Data Type |
| mng\_id | int (PK) |
| mng\_fname | nvarchar(255) |
| mng\_lname | nvarchar(255) |
| mng\_email | nvarchar(255) |
| account\_id | int (FK to User) |

## Table: Instructor

Description: Contains instructor information.

|  |  |
| --- | --- |
| Column Name | Data Type |
| inst\_id | int (PK) |
| inst\_fname | nvarchar(255) |
| inst\_lname | nvarchar(255) |
| nid | char (Unique) |
| year\_o\_experince | int |
| ints\_email | nvarchar(255) |
| hire\_date | date |
| city | nvarchar(255) |
| street | nvarchar(255) |
| specialization | nvarchar(255) |
| acc\_id | int (FK to User) |

## Table: Instructor\_Phone

Description: Stores phone numbers of instructors.

|  |  |
| --- | --- |
| Column Name | Data Type |
| inst\_id | int (FK to Instructor) |
| phone | nvarchar(255) |

## Table: Course

Description: Details about courses offered.

|  |  |
| --- | --- |
| Column Name | Data Type |
| c\_id | int (PK) |
| c\_name | nvarchar(255) |
| description | nvarchar(255) |
| min\_degree | int |
| max\_degree | int |

## Table: Instructor\_Course

Description: Relationship between instructors and courses.

|  |  |
| --- | --- |
| Column Name | Data Type |
| inst\_id | int (FK to Instructor) |
| c\_id | int (FK to Course) |

## Table: Track

Description: Tracks within departments.

|  |  |
| --- | --- |
| Column Name | Data Type |
| track\_id | int (PK) |
| track\_name | nvarchar(255) |
| dept\_id | int (FK to Department) |

## Table: Branch

Description: Training branches and locations.

|  |  |
| --- | --- |
| Column Name | Data Type |
| branch\_id | int (PK) |
| branch\_name | nvarchar(255) |
| mng\_id | int (FK to Training\_Manager) |
| int\_id | int (FK to Intake) |

## Table: Branch\_Track\_Intake

Description: Links branches, tracks and intakes.

|  |  |
| --- | --- |
| Column Name | Data Type |
| track\_id | int (FK to Track) |
| branch\_id | int (FK to Branch) |
| int\_ID | int (FK to Intake) |

## Table: Student

Description: Stores student personal data and relations.

|  |  |
| --- | --- |
| Column Name | Data Type |
| std\_ID | int (PK) |
| std\_Fname | nvarchar(255) |
| std\_Lname | nvarchar(255) |
| GPA | float |
| enrol\_date | date |
| DOB | date |
| int\_ID | int (FK to Intake) |
| acc\_ID | int (FK to User) |
| track\_id | int (FK to Track) |
| branch\_id | int (FK to Branch) |

## Table: Student\_phone

Description: Stores phone numbers for students.

|  |  |
| --- | --- |
| Column Name | Data Type |
| std\_ID | int (FK to Student) |
| Phone\_Number | nvarchar(255) |

## Table: Exam

Description: Exam details including timing and course.

|  |  |
| --- | --- |
| Column Name | Data Type |
| exam\_id | int (PK) |
| a\_option | nvarchar(255) |
| year | int |
| exam\_type | nvarchar(255) |
| track\_degree | int |
| track\_time | time |
| exam\_time | time |
| std\_time | time |
| c\_id | int (FK to Course) |
| inst\_id | int (FK to Instructor) |

## Table: Intake\_Exam

Description: Exams assigned per intake.

|  |  |
| --- | --- |
| Column Name | Data Type |
| int\_id | int (FK to Intake) |
| c\_id | int (FK to Course) |

## Table: Track\_Course

Description: Courses assigned per track.

|  |  |
| --- | --- |
| Column Name | Data Type |
| track\_id | int (FK to Track) |
| exam\_id | int (FK to Exam) |

## Table: Question

Description: Questions stored for exams.

|  |  |
| --- | --- |
| Column Name | Data Type |
| que\_id | int (PK) |
| que\_type | nvarchar(255) |
| b\_a\_answer | nvarchar(255) |
| correct\_answer | nvarchar(255) |
| que\_test | nvarchar(255) |
| c\_id | int (FK to Course) |
| exam\_id | int (FK to Exam) |

## Table: MCQ

Description: Choices for multiple choice questions.

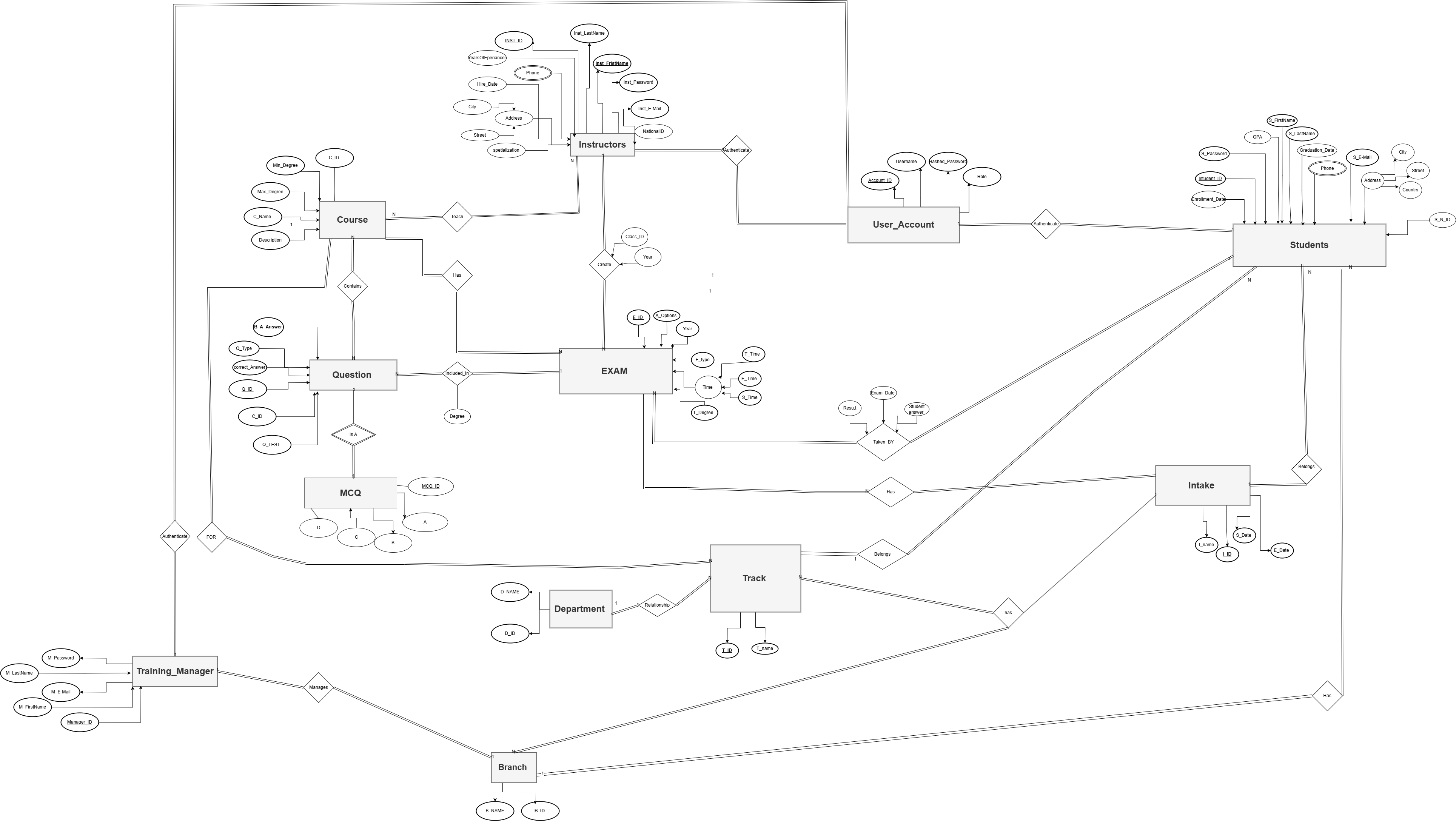
|  |  |
| --- | --- |
| Column Name | Data Type |
| mcq\_id | int |
| que\_id | int (FK to Question) |
| A | nvarchar(255) |
| B | nvarchar(255) |

## Table: Student\_Exam

Description: Stores student results per exam.

|  |  |
| --- | --- |
| Column Name | Data Type |
| std\_ID | int (FK to Student) |
| exam\_id | int (FK to Exam) |
| Result | nvarchar(255) |

ERD



Mapping

