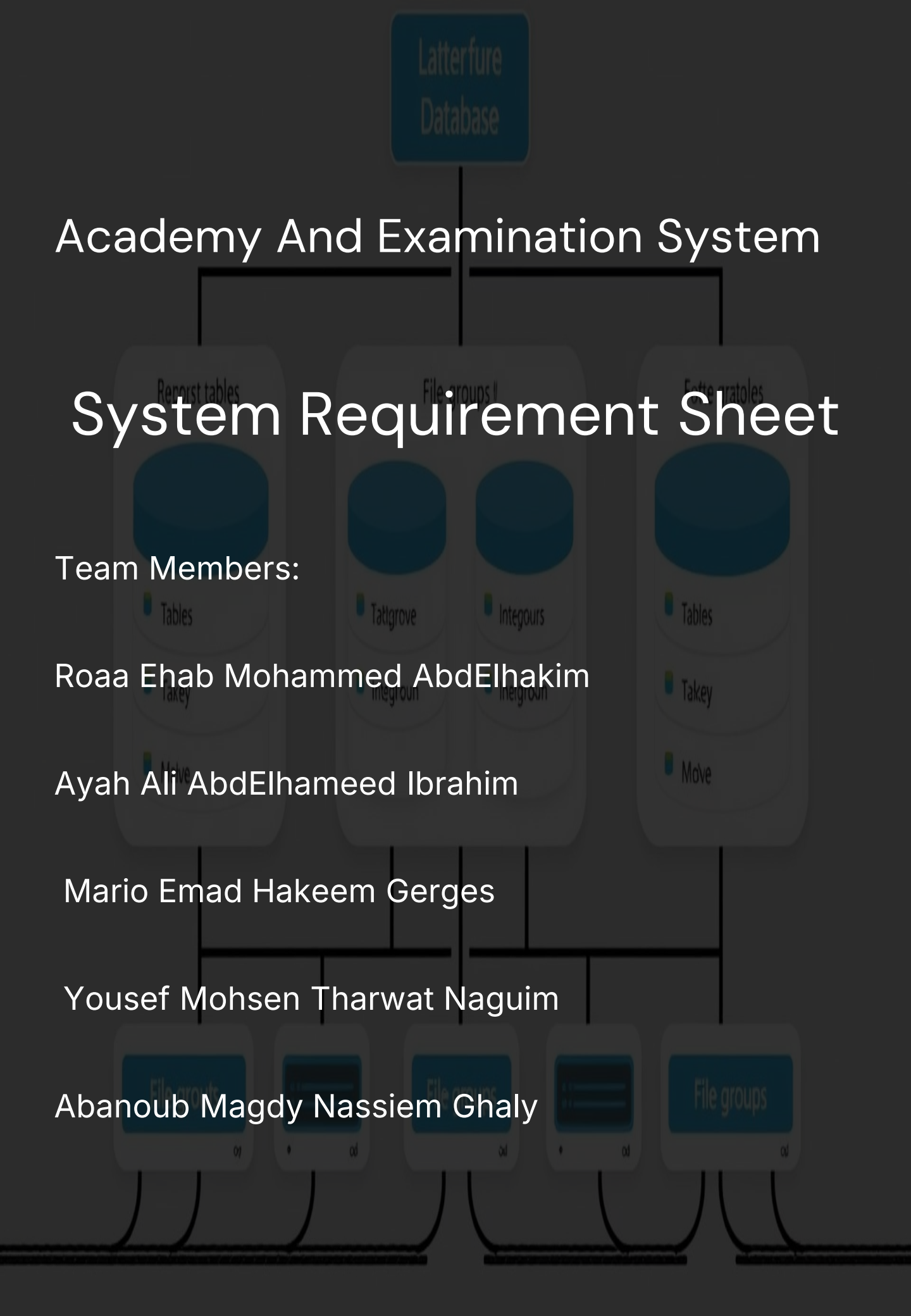
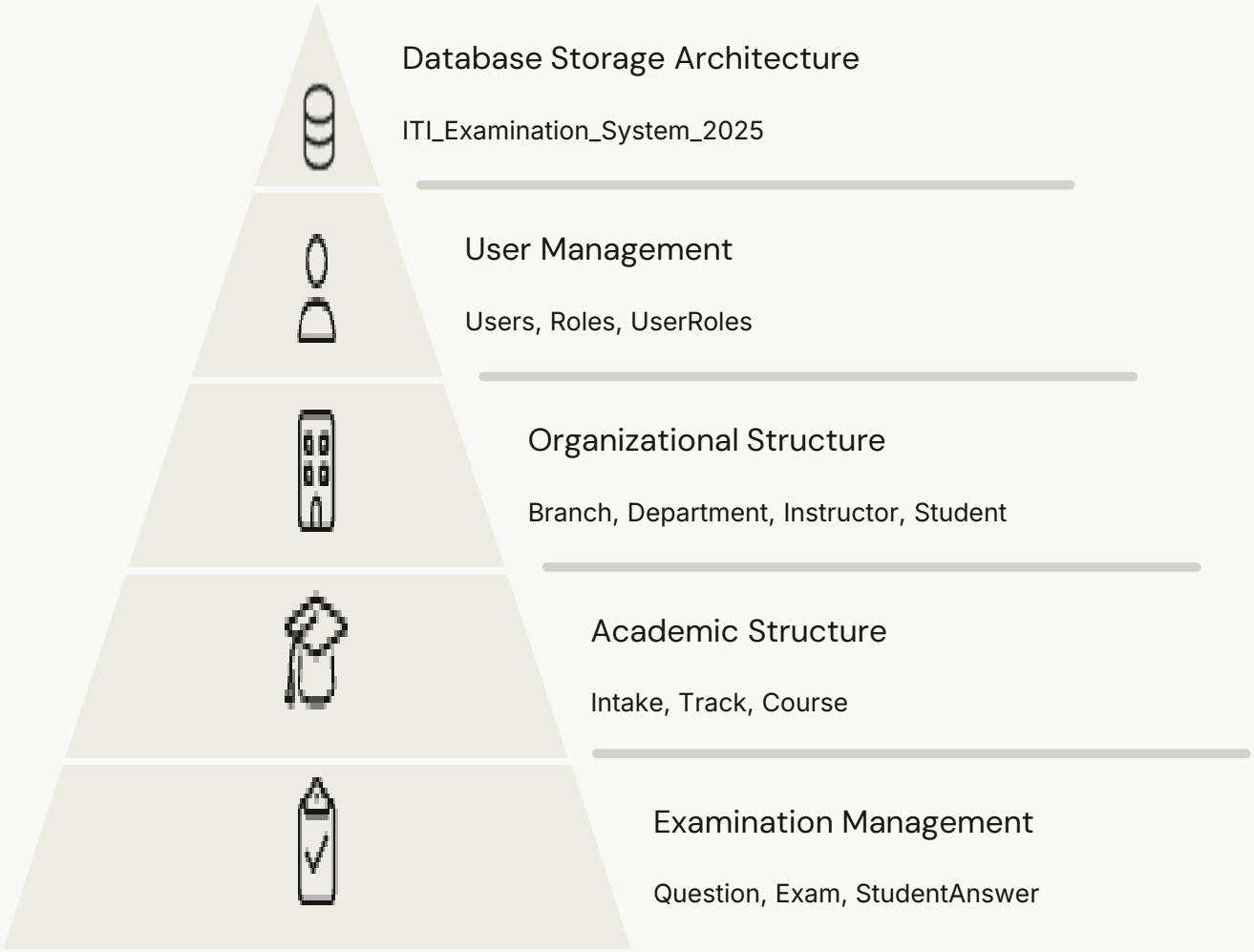

Latterfure
Database
Academy And Examination System
System Requirement Sheet
Team Members:
Roaa Ehab Mohammed AbdElhakim
Ayah Ali AbdElhameed Ibrahim
Mario Emad Hakeem Gerges
Yousef Mohsen Tharwat Naguim
Abanoub Magdy Nassiem Ghaly

Data System Overview



Database Storage Architecture

- Database Name: ITI_Examination_System_2025

- Filegroups:

Primary → Core data storage (mdf) •

Users → Contains user-related data (e.g., Users, Roles, Instructors, Students) •

CourseDetails → Courses, CourseInstructors •

Structure → Branches, Departments, Tracks, Intakes •

ExamDetails → Exams, Questions, Answers, Results •

- Transaction Log: Stored separately for durability and recovery

User and Organizational Structure

User Management

- Users Table: Stores common data (name, gender, phone, etc.) for all system actors
- Roles: Defines role types (Admin, TrainingManager , etc.)
- UserRoles: Supports many-to-many mapping between Users and Roles

Organizational Structure

- Branch: Physical locations
- BranchManager: User responsible for a branch
- Department: Logical academic units (e.g., Software, Networks)
- Instructor: A user who teaches courses, tied to a Department and Branch
- Student: A user enrolled in a track, intake, and branch

Academic and Examination Structure

Academic Structure

- Intake: Represents a specific cohort (start/end dates, year)
- Track: Academic specializations (e.g., .NET, Web, AI)
- IntakeTrack: Many-to-many relationship between Tracks and Intakes
- Course: Tied to a Track, with grading thresholds
- CourseInstructor: Many-to-many between instructors and courses

Examination Management

- Question: Linked to a Course, includes metadata (type, difficulty, default mark)
- Types: Text, MCQ, True/False
- Answer Tables: McqAnswer, TrueAndFalseAnswer, TextAnswer
- Exam: Tied to a Course, with timing, type, and total degree
- Types: exam, corrective
- ExamQuestion: Maps questions to exams with custom marks and order
- ExamStudent: Which students are assigned to which exams
- StudentAnswer: Stores actual answers, correctness, and marks
- StudentExamResult: Final grading with pass/fail, percentage, and letter grade

Data Management System (DMS) Components

Schema Management

- Defines database structure
- Tables are grouped logically by purpose and placed in specific filegroups
- Constraints: Primary Keys, Foreign Keys, Check Constraints

Query Processing & Optimization

- Normalized Design: Entities are split into separate tables to reduce redundancy
- Indexed by Default: Primary keys are automatically indexed
- Join Paths: Clearly defined with foreign keys for efficient querying

Data Integrity & Validation

- Data types (INT, NVARCHAR, BIT, etc.)
- CHECK constraints for enums
- UNIQUE constraints on columns like Email, Phone, Location, etc.
- DEFAULT values like IsDepartmentManager BIT DEFAULT 0

Core Technologies Used

Microsoft SQL Server

- Version: SQL Server 2022 (MSSQL16.MSSQLSERVER as seen in file paths)

- Purpose: Relational Database Management System (RDBMS) for storing and querying structured data.

Features used:

- Filegroups: Used for physical data segregation and performance optimization.
- T-SQL (Transact-SQL): Used for DDL and DML operations.
- Constraints: Check, Foreign Key, Unique, Identity, Default.

Core Technologies Used .count

Data Modeling Techniques

- Normalized relational design (3NF+)
- Many-to-many relationships via junction tables (e.g., UserRoles, CourseInstructor)
- Role-based access modeling via Roles and UserRoles tables
- Computed columns and constraints to enforce business logic

Analytics Tables

CourseInstructor Table

Table Definition: The CourseInstructor table defines the relationship between instructors and the courses they are assigned to. A course can have multiple instructors, and an instructor can teach multiple courses.

Table Constraints:

- Primary Key Constraint: Ensures that the combination of CourseId and InstructorId is unique.
- CourseId Foreign Key: Ensures that each CourseId in CourseInstructor exists in the Course table.
- InstructorId Foreign Key: Ensures that each InstructorId in CourseInstructor exists in the Instructor table.

IntakeTrack Table

Table Definition: The IntakeTrack table is a junction table used to map relationships between:

- Tracks (educational specializations, e.g., AI, Web, Mobile)
- Branches (geographic or organizational divisions)
- Intakes (cohorts or academic sessions)

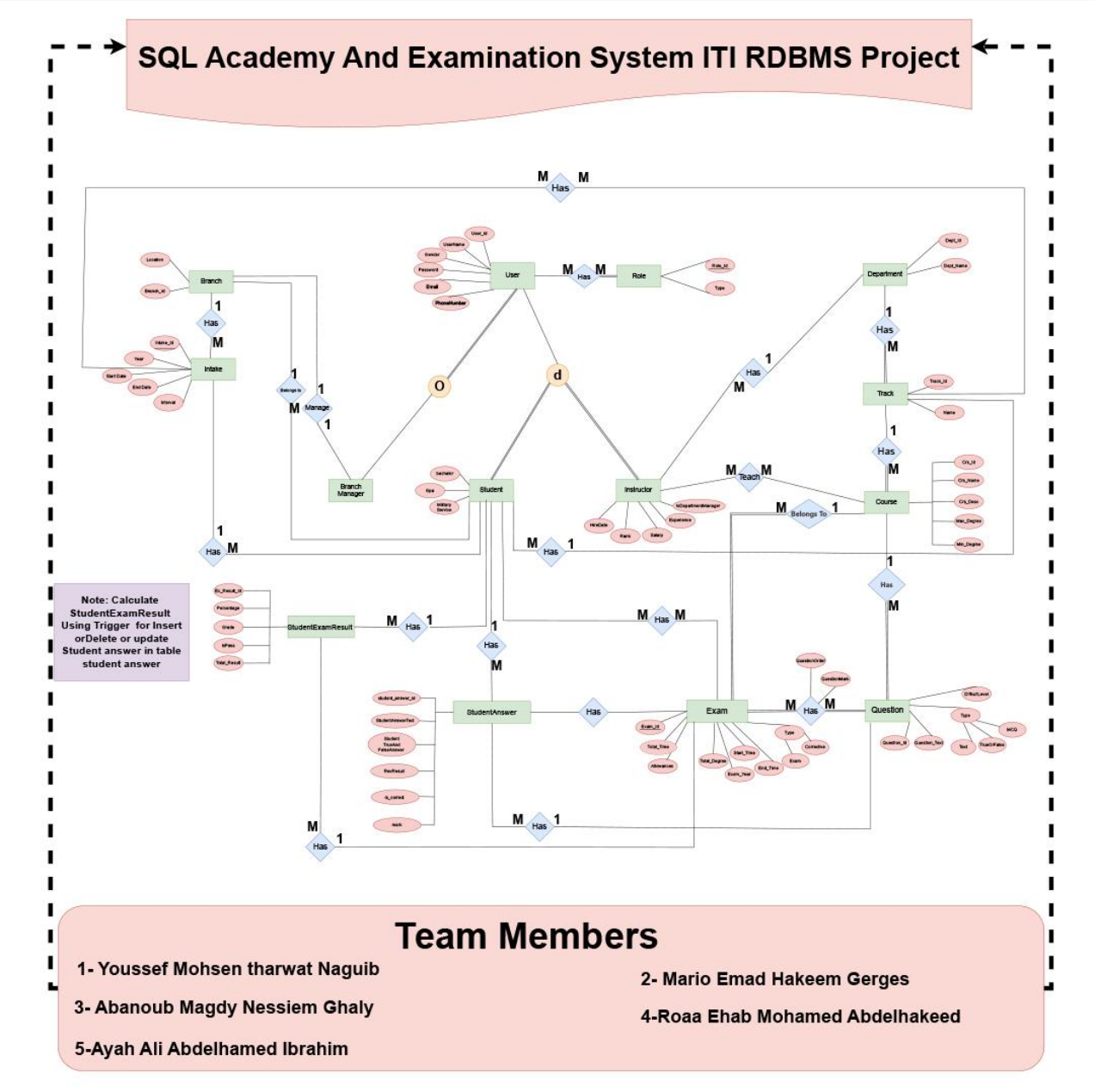
This table defines which track is offered in which branch and intake.

Table Constraints:

- PRIMARY KEY (TrackId, BranchId, IntakeId): Ensures each combination of track, branch, and intake is unique. Prevents duplicate entries for the same offering.
- FOREIGN KEY (TrackId) → Track(TrackId): Ensures the referenced track exists.
- FOREIGN KEY (BranchId, IntakeId) → Intake(BranchId, IntakeId): Ensures the referenced branch-intake combination exists.

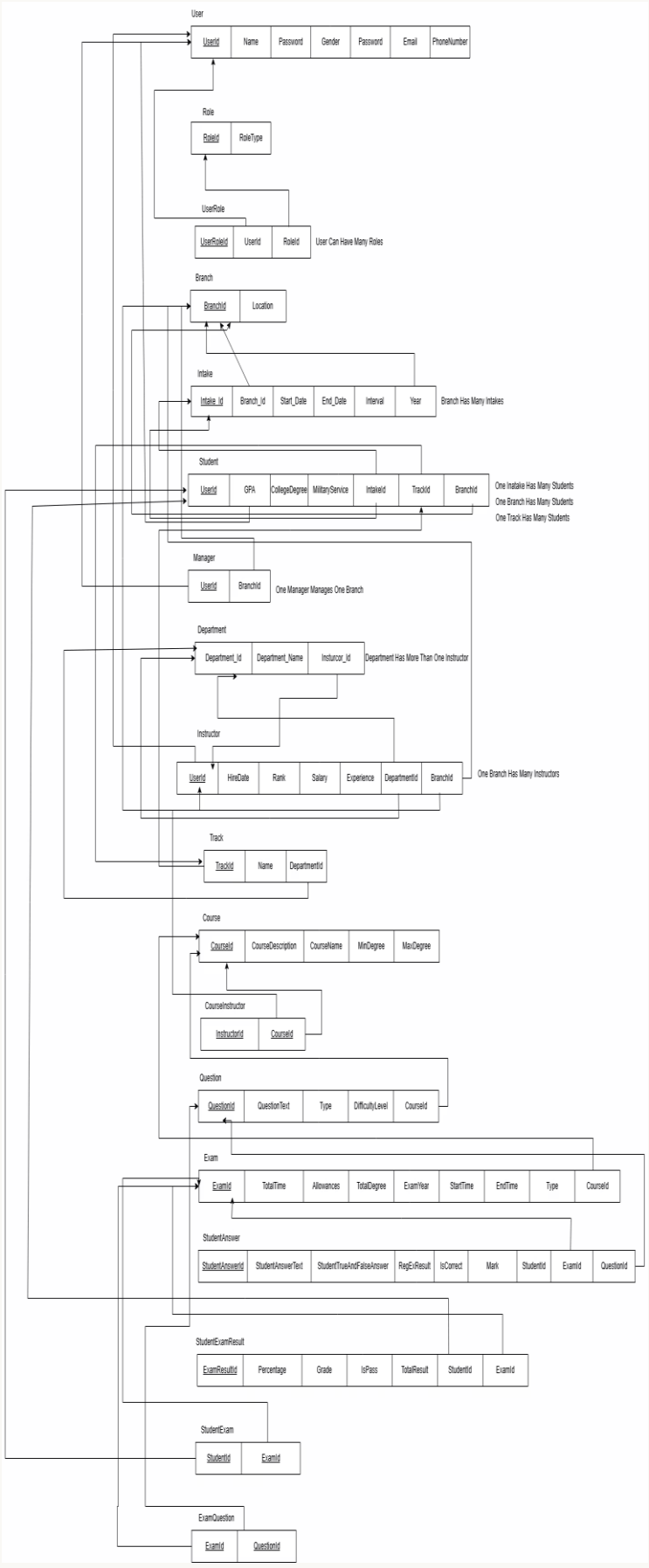
Entity Relationship Diagram and Mapping

ERD (Entity Relationship Diagram)



The Entity Relationship Diagram shows the complete database structure with all tables and their relationships.

Mapping



The mapping diagram illustrates how different entities in the system relate to each other and how they are implemented in the database schema.