



LEARNING MANAGEMENT SYSTEM DOCUMENTATION REPORT

Abdelmohimen Elmegber
Data Analyst

1. Introduction

The Learning Management System (LMS) is an essential component for educational institutions to effectively manage their resources, courses, instructors, and students. This documentation outlines the database schema designed to support the functionalities required for an efficient LMS.

2. Requirement Analysis Mapping

Requirement Analysis

The initial step in designing the Learning Management System (LMS) is to analyze the business requirements and map them to system functionalities. The following is a summary of the roles and functions that are to be performed by the LMS.

The LMS was designed to enable students, instructors, and course management. Institution managers should be able to add classes and institutions and have general access to information. Senior instructors can create, new programs-courses and create a schedule where they assign instructors to teach specific courses at specified times and classrooms. All instructors should be able to grade their students. Instructors and students should be able to view their schedules and exams too.

Institution Managers:

- Create and manage institutions, classes, and departments.
- Manage programs, courses, and topics.
- Assign courses to programs and instructors to course classes.

Senior Instructors:

- Create programs, courses, and topics.
- Create exams and assessments for each course.
- Define prerequisites and assign semesters to courses.

Instructors' Managers:

- Register instructors and assign them to institutions, departments, and supervisors.
- Manage instructor information and credentials.

Students Managers:

- Register students and assign them to programs and institutions.
- Manage student enrollment and track enrollment submissions.

Instructors:

- View personal information, assigned courses, and students.
- Manage student course assessments, results, and exam submissions.

Students:

- View personal information, course assignments, and results.
- View courses' assessments' scores and exams' submissions.
- Subscribe to available courses or classes and submit exams.

Database Mapping

The next step involves translating the identified functionalities into a database schema. Using Draw IO to design the top level database architecture and translating it to The Attached SQL DDL script outlines the tables, columns, and relationships needed for the LMS and they are as follows:

Database Schema Overview

The database schema is organized into three main schemas: Institutions, Programs, and Students, each encapsulating pertinent entities and relationships.

- Institutions Schema:
 - Manages the infrastructure of educational institutions, including institutions themselves, programs, instructors, departments, and classrooms.
- Programs Schema:
 - Focuses on academic programs, courses, assessments, and their interconnections.
- Students Schema:
 - Facilitates student enrollment, assessment scores, course results, and exam submissions.

Schema Entities

Each schema is composed of tables representing distinct entities and their attributes, fostering a structured data model conducive to efficient data management and retrieval. For instance, the Institutions schema contains tables for Institutions, Semesters, Time Slots, Departments, and Instructors, among others.

Relationships and Constraints

Foreign key constraints establish relationships between entities across schemas, ensuring data integrity and coherence. Additionally, constraints such as primary keys, unique constraints, and check constraints are enforced to maintain data consistency and validity throughout the database.

Stored Procedures

Stored procedures have been meticulously implemented to fulfill the business requirements of the Learning Management System (LMS) database, providing a structured framework for Create, Read, Update, and Delete (CRUD) operations. These procedures enhance security by enforcing controlled access to database objects, mitigating the risk of unauthorized access to sensitive data. Through role-based access control, specific permissions are assigned to distinct user roles within the LMS system. For example, Institution Managers, Senior Instructors, Instructors' Managers, and Students Managers can execute procedures tailored to their roles, such as managing institutions, courses, instructors, and students.

Additionally, stored procedures facilitate centralized management of database operations, enabling administrators to optimize query performance and ensure consistency across the system. By encapsulating database logic within stored procedures, maintenance tasks are streamlined, minimizing the need for modifications to application code. Each stored procedure is parameterized, providing flexibility in data manipulation, while error-handling mechanisms maintain transactional integrity, mitigating potential data inconsistencies.

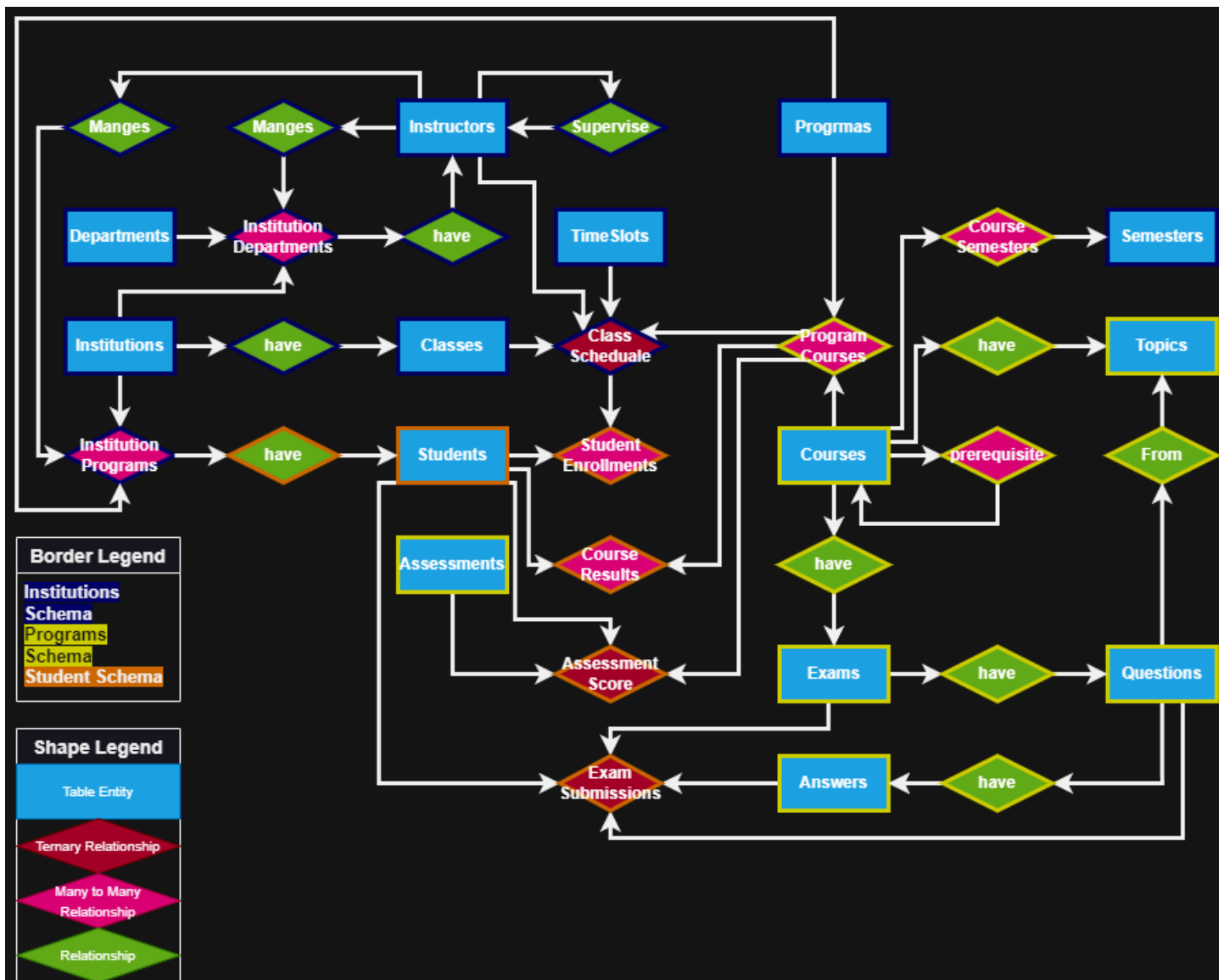


Figure 1: Top-Level Schema Design (ERD from Draw.io)

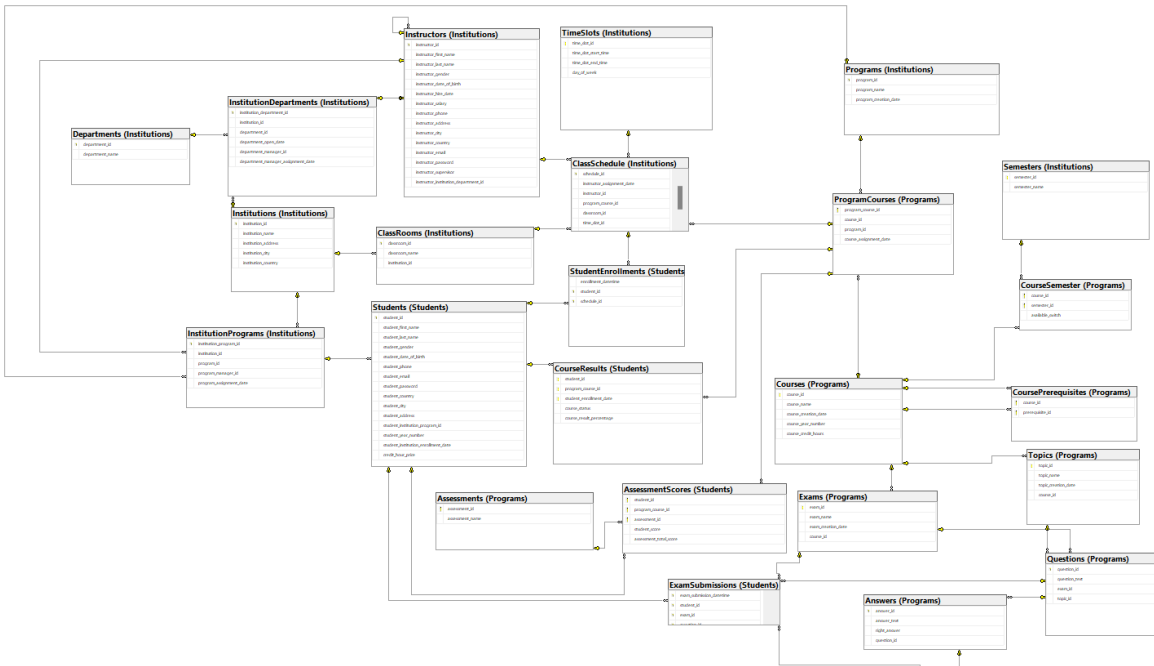


Figure 2: Detailed Schema Design (ERD from SQL Server Management Studio)

These ER diagrams visualize the database structure, including tables, columns, relationships, and constraints. The schema design aligns with the identified system functionalities, ensuring that the database can effectively support the requirements of the Learning Management System.

3. Mock Data Generation

Data Generation Strategy

To populate the core tables such as students, instructors, and courses with realistic data that complies with the business logic of the Learning Management System (LMS), various data generation tools and techniques were employed.

Data Generation Tools

Mockaroo

Mockaroo, a website that generates mock data, was used to populate core tables like students, instructors, and departments. Mockaroo provides customizable data templates, allowing users to generate randomized attributes such as names, addresses, emails, and phone numbers.

SQL Queries and Cursors

To generate simple patterns based on business logic conditions SQL queries were used. However, more complex business processes generating data based on various conditions for a large number of variables would require using database cursors.

Python Scripting

For very complex business patterns Python can be used to significantly cut down development time since we are using python to generate data only one time it is more practical to develop the data quickly without worrying about the extra dependencies.

Data Generation Process

Core Table Population

Using Mockaroo, we generate mock data for core tables such as students, instructors, departments, and institutions. The generated data includes attributes such as names, addresses, emails, and phone numbers, ensuring diversity and realism.

Generating Data Compliant with Business Logic

To generate complex entities such as courses, exams, assessments, and class schedules, I utilized a hybrid approach involving SQL queries and Python scripting:

- SQL queries were leveraged to extract pertinent data from core tables and generate new records based on predefined rules, relationships, and patterns.
- For more intricate data generation tasks, database Cursors and T-SQL were employed to execute complex queries, allowing for sophisticated data manipulation.
- Python, advanced logic was implemented using libraries such as pandas to simulate realistic scenarios. This ensured that the generated data adhered closely to the intricate business logic of the LMS, enabling the creation of datasets that accurately reflected real-world educational scenarios.

Export Format

After all the data is generated cleaned and transformed into database schema-compliant form, the data is saved into CSV files in the data staging folder.

Result Verification

The verification process ensures the integrity and accuracy of the generated data, addressing both business and database compliance requirements. Meticulous revision is conducted to verify:

- **Business Compliance:** Ensuring that exams are correctly assigned to their respective courses and programs, aligning with the established curriculum structure and academic guidelines.
- **Database Compliance:** Cross-referencing table keys to validate relationships, such as confirming that all courses maintain associations with their corresponding programs, thereby upholding data consistency and relational integrity.

4. Importing Mock Data

To Import the data while monitoring it for any discrepancy to the database, I used **SQL Server Integration Services (SSIS)**. Using an SSIS package created within SQL Server Data Tools (SSDT) I moved the data from the staging CSV files by including data flow tasks and control flow elements necessary to migrate and validate the data.

Database Reset

To validate the generated data and the successful migration of new files I created a task to truncate all data from the database and reset all Surrogate identity keys.

Source and Destination Configuration

Every Generated and Cleaned CSV file is referenced from the staging file as a data source. Additionally, a connection to the database engine is established and the LMS database is referenced as a destination.

Control Flow Steps

- Data flow steps are added so that tables that have no foreign keys are imported first then tables that depend on them are imported after.
- In the case of circular references where tables have reference each other a step is added in the control flow to disable the foreign keys constraints temporarily and then activate the constraints again after data importation ensuring constraints integrity.
- Note that by only disabling the circular reference table constraints is easier to identify errors and debug them.

Role in data Generation and Validation

To generate the data mock data and validate them SSIS played a major role. Since the data is built based on core tables then SQL/Python generated tables and all are connected. Using SSIS I iteratively generated and validated the data.

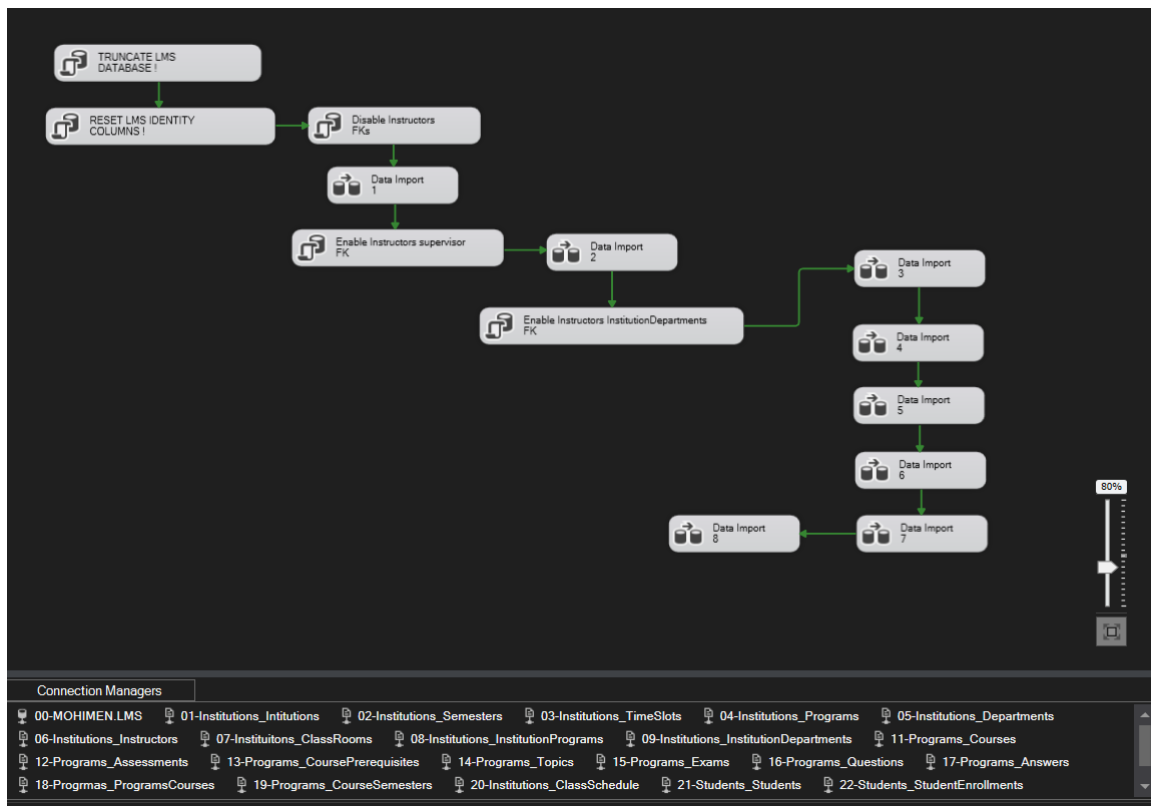


Figure 5: Student Demographics across different programs and courses

5. Reporting

To monitor business activity users must have the ability to generate automated reports to be up to date at any moment in time. This requirement can be achieved by utilizing views to join data in meaningful ways and dashboard reports.

Database Views

Three views were created to provide a top view of the learning process allowing the user to be informed about the performance of students, instructors, programs, and more.

Exam submissions

This view focuses on exam-related data; including student responses, correct answers, and exam details. Integrating this view into a dashboard enhances exam management and monitoring capabilities. This view enables Exam Performance Analysis and Further Improve Program and Course Evaluation

Course assessments

This view consolidates data related to instructors, courses, students, and assessment scores, making it an essential component of the LMS dashboard and a valuable tool for not only instructors and senior instructors but also students. This view can be utilized to track student course work and allow the user to assess the performance of students, instructors, and programs, allowing informed decisions to questions like: what courses are students struggling to pass, are there instructors that their students are not struggling, etc.

Schedule

This view provides dynamic scheduling information, including class timings, upcoming classes, and time until the next class. Integrating this view into a dashboard enhances schedule management and visibility. This view provides a Schedule Overview with Real-time Schedule Updates, Enhancing Time Management for students and instructors.

Dashboards

Utilizing Power BI I Imported the database views to create the following dashboards. The dashboards provide filters to instantly get the most granular info about a student for example, and aggregate visualizations to get a top-down view of a whole educational institution

Student Demographics

The following dashboard displays the distribution of students across different institutions programs and courses using a decomposition tree and a doughnut chart.

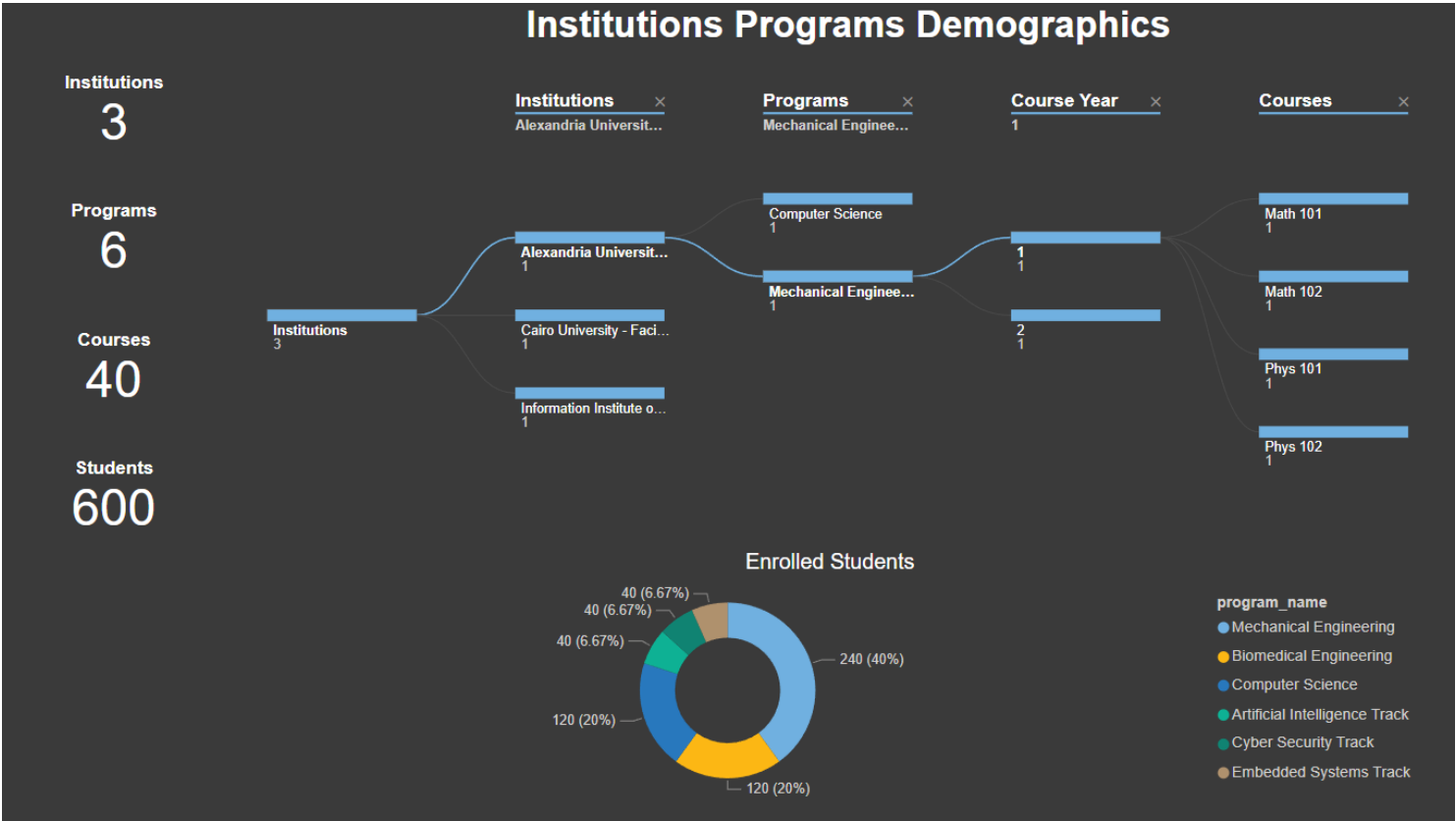


Figure 4: Student Demographics across different programs and courses

Schedule

The schedule dashboard is a dynamic schedule that aims to improve visibility and dynamically alert users to their next classes

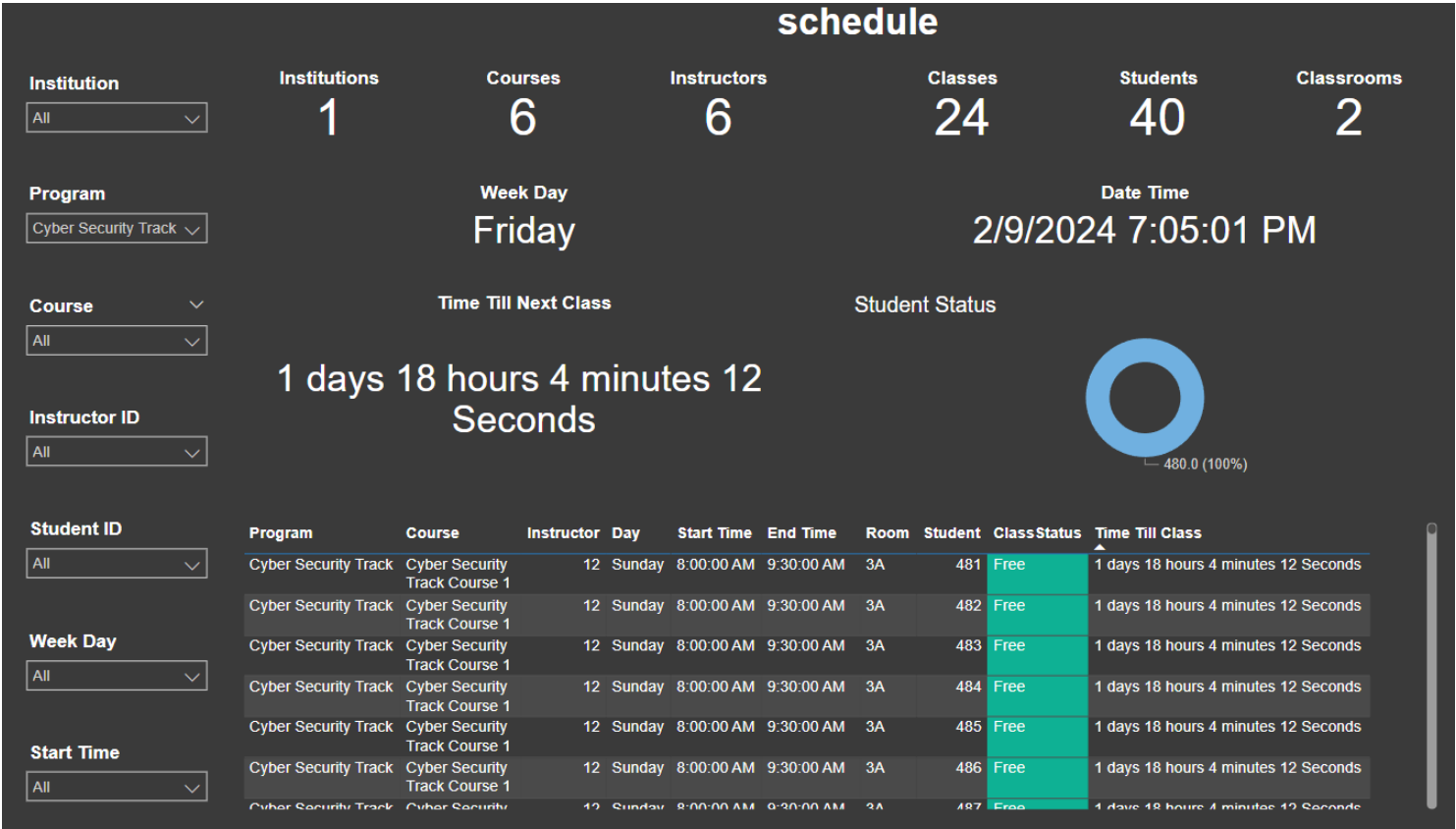


Figure 5: Student Schedule across different programs and courses

Students Assessments

The assessments dashboards show the result of each coursework assessment providing an automated communication system that keeps students informed and up to date with their performance.

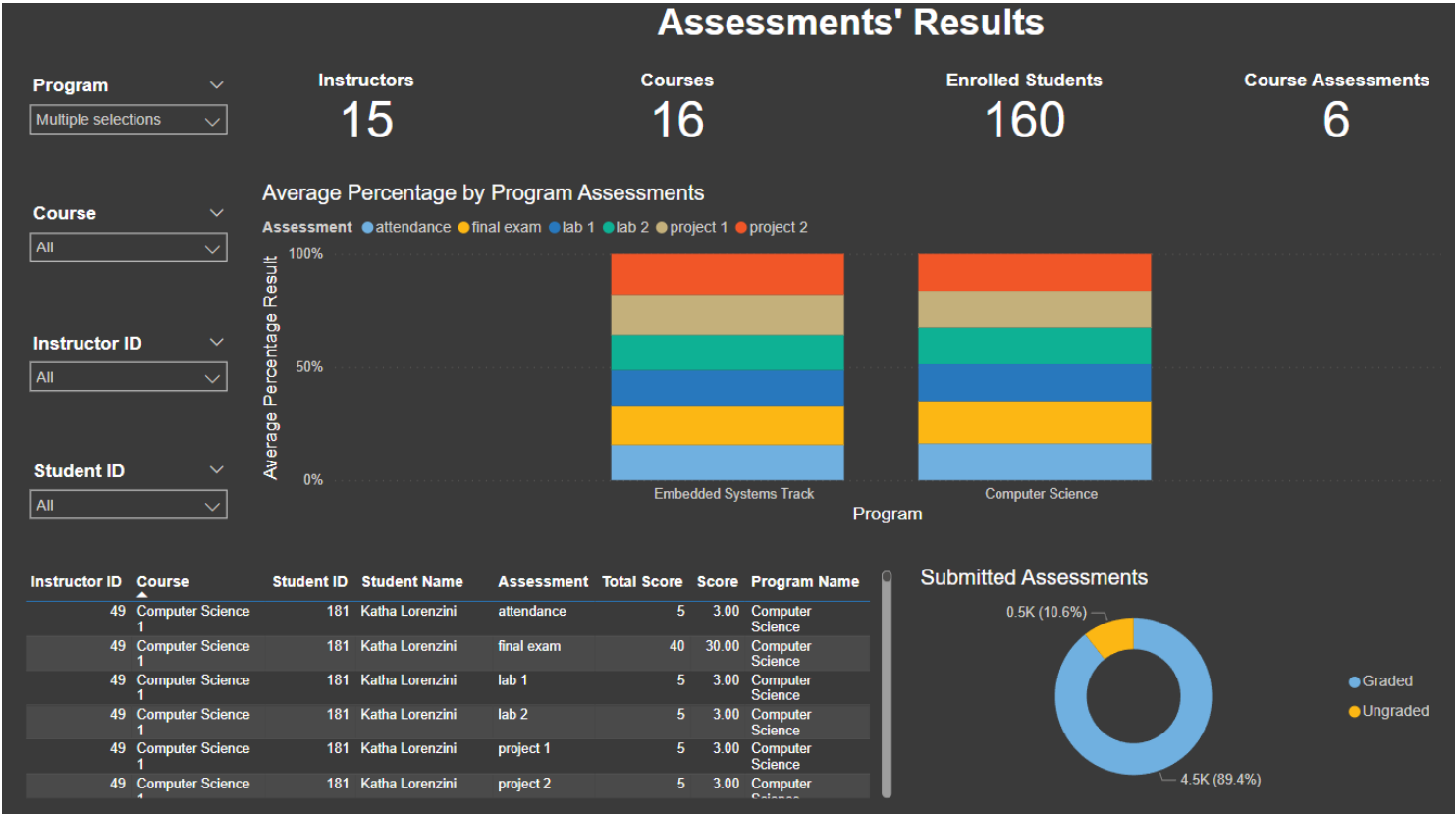


Figure 6: Student Assessments across different programs and courses

Exams

The exam dashboards automatically show students their exam submissions and refer to the text where the student can re-read the topics that they answered its question wrong allowing students to track and improve their performance.

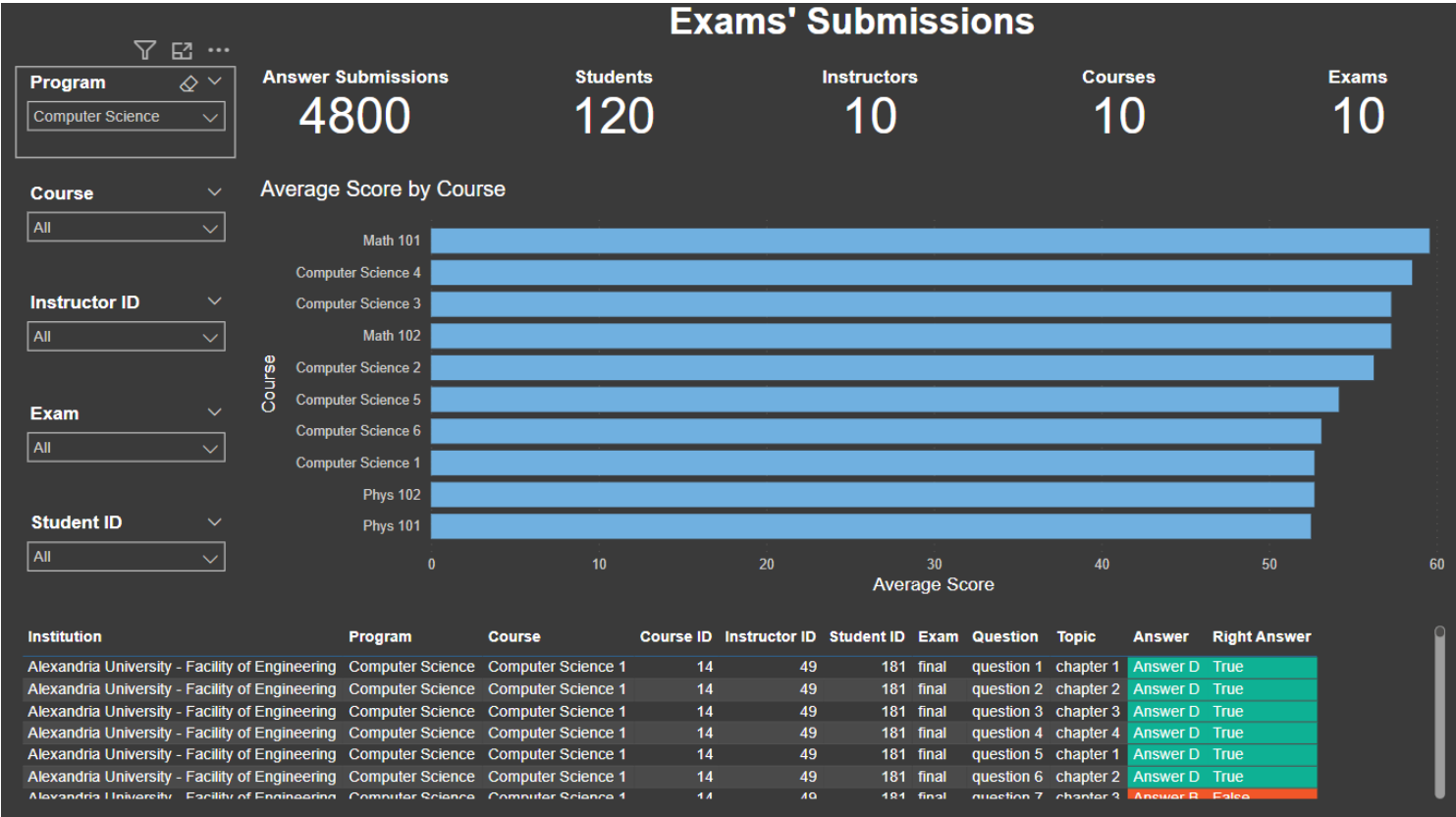


Figure 7: Student Exams across different programs and courses