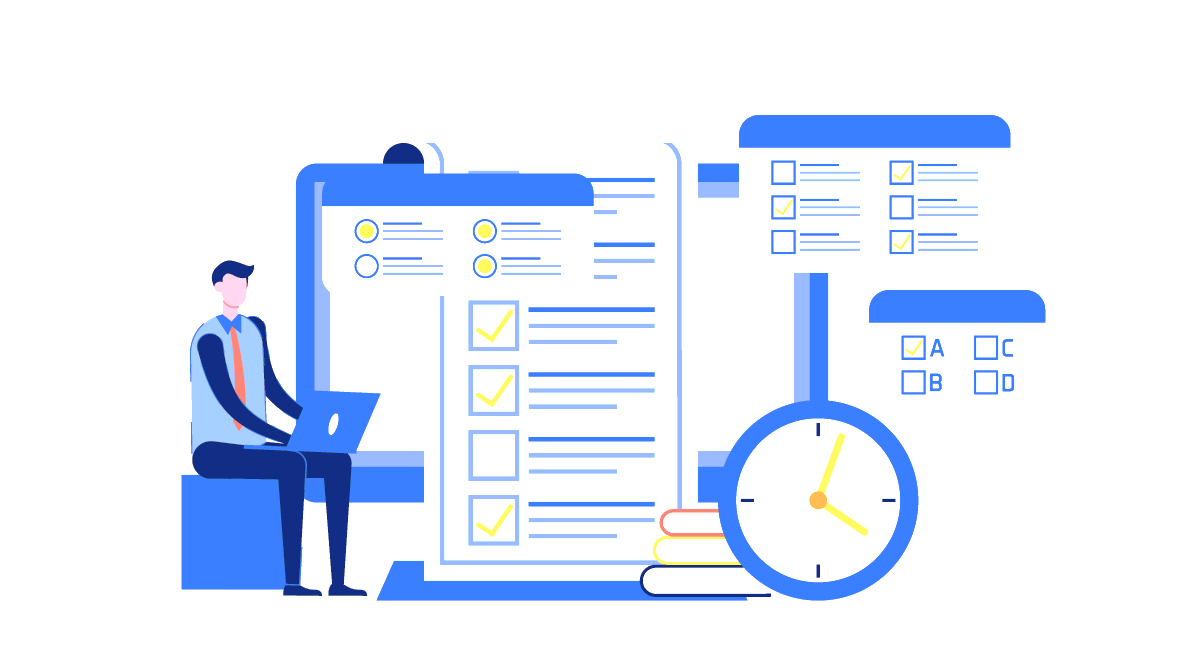


**Online Examination System Project**



**Prepared by:**

* Basant Amr
* Maryam Mohamed
* Nada Osama
* Yomna Ramadan
* Yousra Zaki

**Power BI Developer Track**

**Intensive Code Camp**

7/6/2023

* **Tables of Contents**

1. Introduction
2. System Architecture
3. Business Requirements
4. Data Model
5. Database Design
6. Reporting

* **Introduction**

The Online Examination System is a web-based application that streamlines the process of conducting exams and assessments online.

The Online Examination System not only replaces paperwork but also releases the workload of faculty.

This system provides a secure and easy-to-use platform for students to take exams, instructors to create and manage exams, and administrators to oversee the entire process.

Students can access the system from anywhere with an internet connection, and take exams at their convenience, so they can take exams without going to any physical destination.

Hence, the purpose of this system is to save effort and time of faculty.

The system is designed to be flexible and customizable, allowing instructors to create different types of exams, including multiple choice and true or false. Instructors can also set time limits and grading criteria for each exam and view the results and feedback from students.

* **System Architecture**

The system architecture consists of four main components:

1. User Interface (UI): This component provides the front-end interface for users to interact with the system. It includes web pages, forms, and other UI elements that allow users to log in, register, view exam details, take exams and show grades.
2. Application Layer: This component consists of the server-side code that processes user requests, manages exam data, and communicates with the database. It includes several modules, such as exam management, user authentication, reporting and analytics, and user management.
3. Database Layer: This component stores all the data related to the system, including user profiles, all staff details, topic and courses, questions bank for courses, exam details, exam questions, and exam results. It is implemented using a relational database management system (RDBMS), such as MS SQL Server.

The system architecture is designed to be scalable, secure, and reliable, with multiple layers of security and redundancy to protect against data loss and unauthorized access. The UI and application layers are hosted on a web server, which can be scaled horizontally to handle increased traffic and load. The database layer is hosted on a separate database server, which can be scaled vertically to handle increased data storage and processing requirements.

Overall, the system architecture provides a robust and flexible platform for managing online exams and assessments and can be customized to meet the specific needs of different organizations and institutions.

* **Business Requirement**

1. User Roles: The system should support multiple user roles, such as students, instructors, and administrators, each with their own permissions and access levels. Students should be able to view and take exams, while instructors should be able to create and manage exams.
2. Exam Creation and Management: Instructors should be able to create and manage exams, including setting up exam details such as exam duration, exam type, and grading criteria. Instructors should also be able to add, edit, and delete exam questions, and view exam results and feedback.
3. Exam Taking: Students should be able to access exams, view exam details, and take exams within the specified time limit. The system should provide a user-friendly interface for taking exams, including features such as timers, progress bars, and the ability to save and resume exams.
4. Exam Results and Feedback: The system should provide detailed feedback for each exam, including overall scores, grades, and correct/incorrect answers. Instructors should be able to view and analyze exam results and provide feedback to students.
5. Reporting and Analytics: The system should provide reporting and analytics capabilities, including the ability to generate reports and visualizations on exam performance metrics, such as average scores, pass rates, and question-level analysis. The system should also integrate with external reporting and analytics tools, such as SSRS, Power BI, and Tableau.
6. Scalability and Performance: The system should be scalable and able to handle many users and exams. The system should be designed to be highly available and performant, with minimal downtime and fast response times.
7. Students should take certificates and join a job whether it is freelancing job or hired for companies with full or part time jobs.

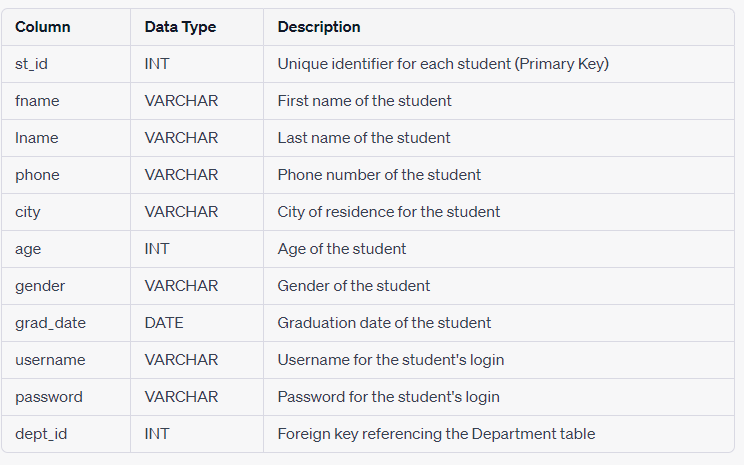
* **Data Model**

The data model includes several entities including:

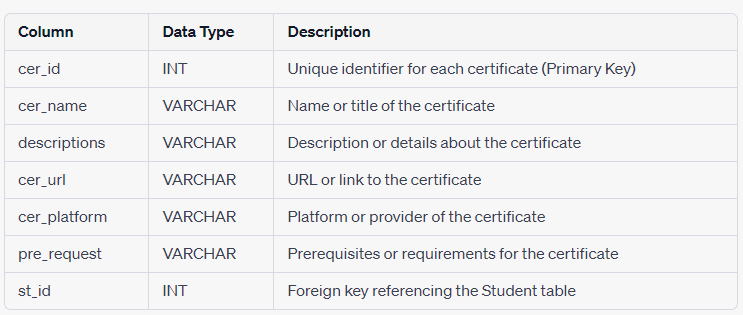
Normalization: To ensure data integrity and minimize redundancy, the database design should follow normalization rules. When we checked our database, we found that it’s already in the third normal form.

**The data model consists of several entities, including:**

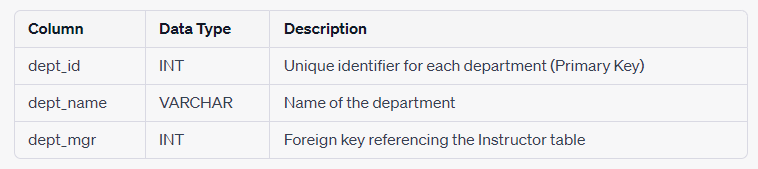
1. Student: The "Student" entity represents individual students enrolled in the system. It stores information about each student, including the following attributes:



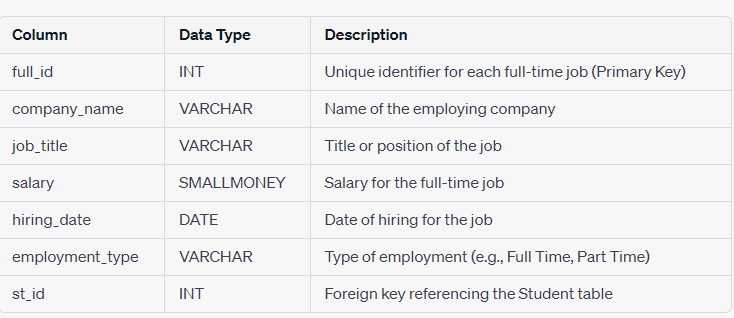
1. Certificates: The "Certificates" table stores information about the certificates obtained by students. It includes the following attributes:



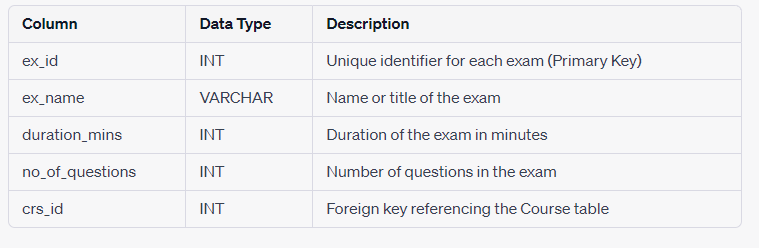
1. Department: The "Department" table represents different departments within the system. It includes the following attributes:



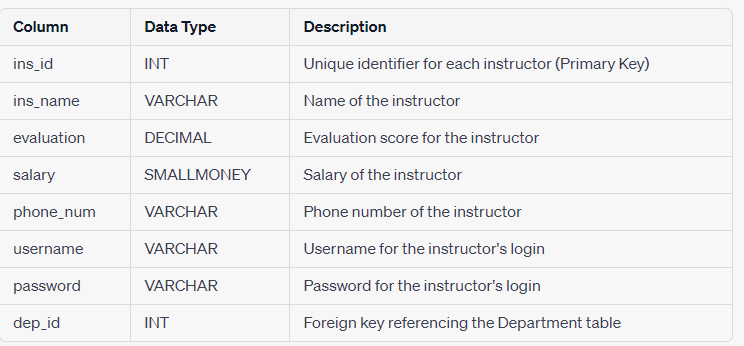
1. FullTimeJob: The "FullTimeJob" table stores information about full-time job positions. It includes the following attributes:



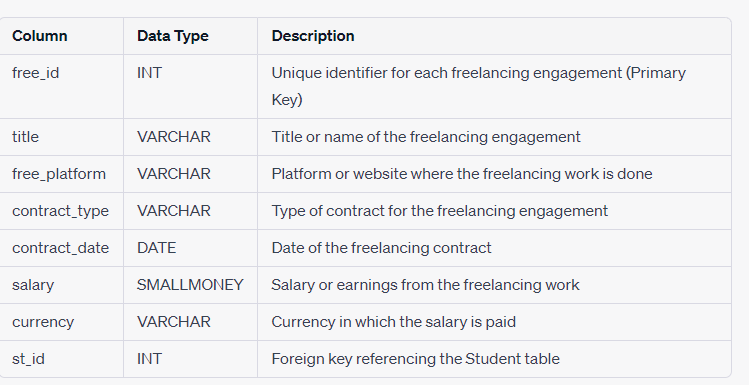
1. Exam: The "Exam" table represents exams within the system. It includes the following attributes:



1. Instructor: The "Instructor" table stores information about the instructors in the system. It includes the following attributes:



1. Freelancing: The "Freelancing" table stores information about freelancing engagements of students. It includes the following attributes:



**The relationships between the entities are as follows:**

* One-to-One Relationship:

No one-to-one relationships are explicitly defined in the provided schema.

* One-to-Many Relationship:

Department to Instructor: One department can have multiple instructors, but each instructor belongs to only one department.

Course to Instructor: One course can have multiple instructors, but each instructor can teach multiple courses.

Department to Student: One department can have multiple students, but each student belongs to only one department.

Exam to Question: One exam can have multiple questions, but each question belongs to only one exam.

* Many-to-Many Relationship:

Course to Topic: A course can be related to multiple topics, and a topic can be associated with multiple courses.

Student to Course: A student can enroll in multiple courses, and each course can have multiple students.

Instructor to Course: An instructor can teach multiple courses, and each course can have multiple instructors.

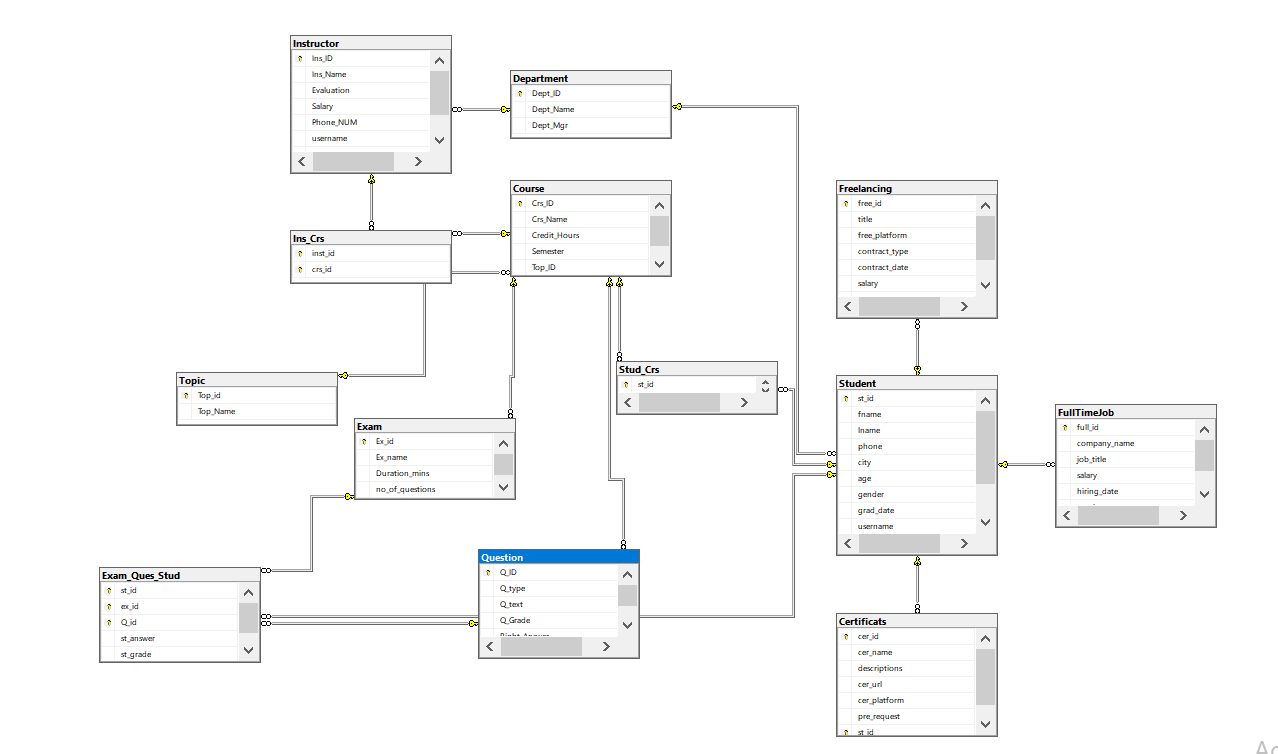
Student to Certificate: A student can have multiple certificates, and a certificate can be obtained by multiple students.

Exam to Course: An exam can be associated with multiple courses, and a course can have multiple exams.

* **Database Design**

The database design for an Online Examination System aims to efficiently store and manage the data related to exams, questions, answers and other relevant information. The design ensures proper organization, integrity, and performance of the system, enabling seamless functionality and reliable data storage.

We have implemented the database in Microsoft SQL Server Management to conduct the following diagram that shows entities and their associated attributes that can be considered for the database design and the relationship between entities:



**Components of the Database Diagram:**

Entities: The diagram illustrates the main entities involved in the Online Examination System, including Students, Instructors, Department, Course, Exams, Questions, Answers, Certificate and FullTimeJob. These entities represent the core components of the system and capture essential information related to exams, questions, and exam results.

Relationships: The diagram showcases the relationships between the entities, indicating how they are connected and interact with one another. It highlights the associations such as one-to-many and many-to-many relationships that exist within the database.

For example, the diagram showcases that a Student can participate in multiple exams, exams can have multiple questions also the user can take multiple certificates and certificates are taken by multiple students.

**Stored Procedure**

1. Insert Stored Procedure: This stored procedure is used to insert new records into the database tables. It should accept input parameters that correspond to the columns in the table and insert the values into the appropriate columns. The stored procedure should also perform validation checks to ensure that the data being inserted is valid and consistent with the database schema. For example, it should check for required fields, data type and length constraints, and referential integrity constraints. Like If we try to insert into a table which is a child for a parent table, it will be checked first that foreign key value matches one of the primary key values.
2. Update Stored Procedure: This stored procedure is used to update existing records in the database tables. It should accept input parameters that correspond to the columns in the table and update the values in the appropriate columns based on the specified condition. The stored procedure should also perform validation checks to ensure that the data being updated is valid and consistent with the database schema.
3. Delete Stored Procedure: This stored procedure is used to delete existing records from the database tables. It should accept input parameters that correspond to the columns in the table and delete the records based on the specified condition. The stored procedure should also perform validation checks to ensure that the data being deleted is valid and consistent with the database schema.  
   For example, if the constrain of foreign key is assigned that when the parent record deleted so no action, then this stored procedure will not allow to delete a record of a parent table which has a child table.
4. Select Stored Procedure: This stored procedure is used to retrieve data from the database tables. It should accept input parameters that correspond to the columns in the table and retrieve the records based on the specified condition. The stored procedure should also perform validation checks to ensure that the data being retrieved is valid and consistent with the database schema.
5. Generate Random Exam Stored Procedure: This stored procedure is used to generate a random exam for a student. It should accept input parameters that correspond to the student ID and the exam ID and generate a random set of questions from the exam question bank. The stored procedure should also ensure that the generated exam meets the specified criteria, such as the number of questions, and time limit.
6. Student Answers Stored Procedure: This stored procedure is used to record the answers submitted by a student for an exam. It should accept input parameters that correspond to the student ID, the exam ID, question ID and the answers submitted by the student. The stored procedure should also perform validation checks to ensure that the data being submitted is valid and consistent with the exam requirements.
7. Correct Exam Stored Procedure: This stored procedure is used to correct the answers submitted by a student for an exam. It should accept input parameters that correspond to the student ID, the exam ID, and the answers submitted by the student. The stored procedure should compare the submitted answers with the correct answers for each question, calculate the score for each question, and generate an overall score for the exam.   
     
   So, stored procedures are used to make the database system designed to meet the specific needs and requirements of online examination system. And make sure that the data processing and retrieval is efficient, accurate, and consistent.

**Data Entry:**

To populate the database with relevant data, we utilized the GetData application, which facilitated the insertion of various records into the system. The dataset comprises 1000 student profiles, each representing a unique individual enrolled in the system. Additionally, we incorporated information on 1000 freelancing jobs, highlighting the diverse range of opportunities available to students. Furthermore, we included 500 full-time job records, showcasing the employment prospects associated with the platform.

To enhance the academic aspect of the system, we incorporated 2000 student course records, outlining the courses undertaken by individual students. This comprehensive collection of data reflects the diverse academic interests of the student population. To ensure the courses' seamless implementation, we integrated 48 distinct courses into the system, each uniquely identified and equipped with relevant details.

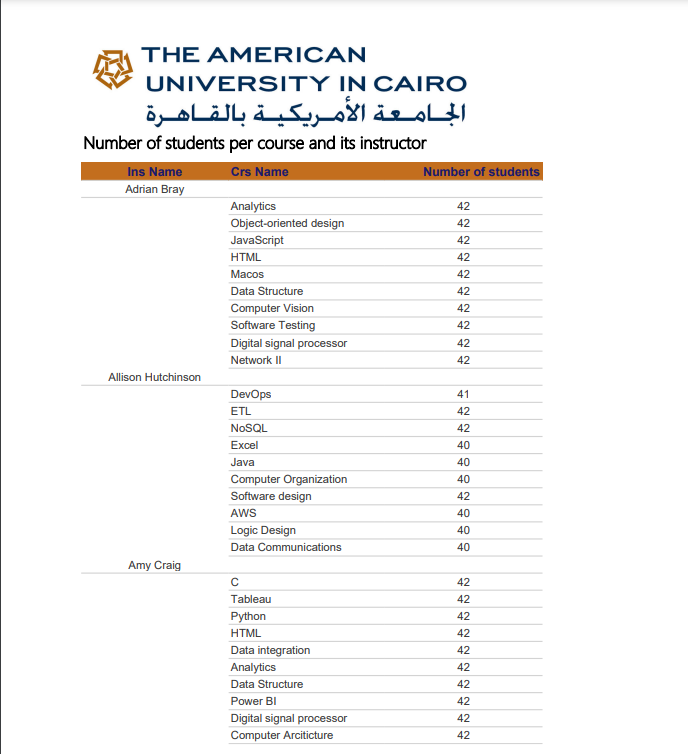
To support the learning experience, we introduced 80 instructors into the system, each associated with specific courses. The instructors' information enables effective course management and facilitates seamless interaction between students and faculty members.

By meticulously incorporating these records into the database, we have established a robust foundation for the Online Examination System project. The extensive dataset ensures the system's functionality and supports its core features, empowering students and instructors alike to engage in a comprehensive and dynamic learning environment.

* **Reporting and analysis**

The Online Examination System is equipped with robust reporting and analytics features that provide valuable insights and facilitate data-driven decision-making. This section explores the various tools and technologies employed in the system, including SSRS reports, Power BI dashboards, and Tableau visualizations."

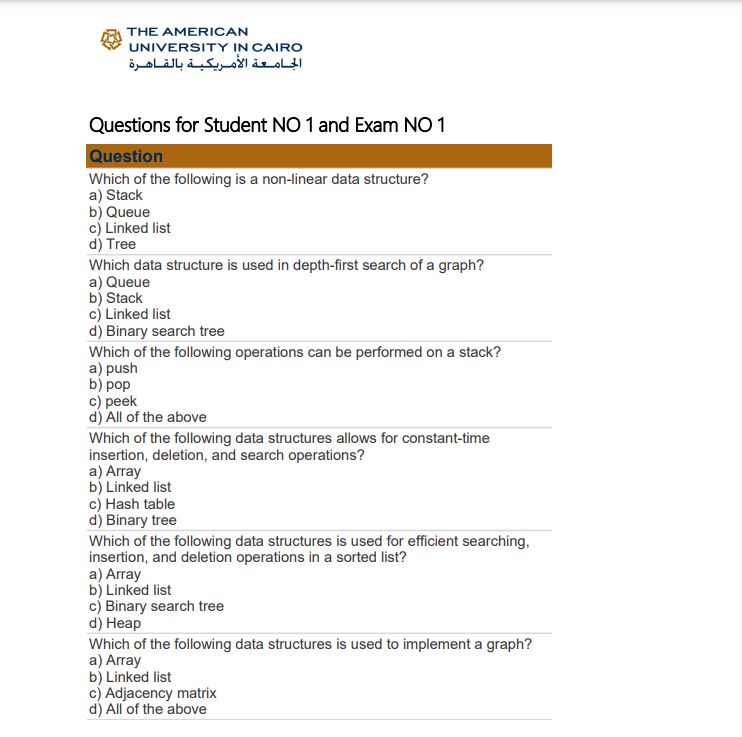
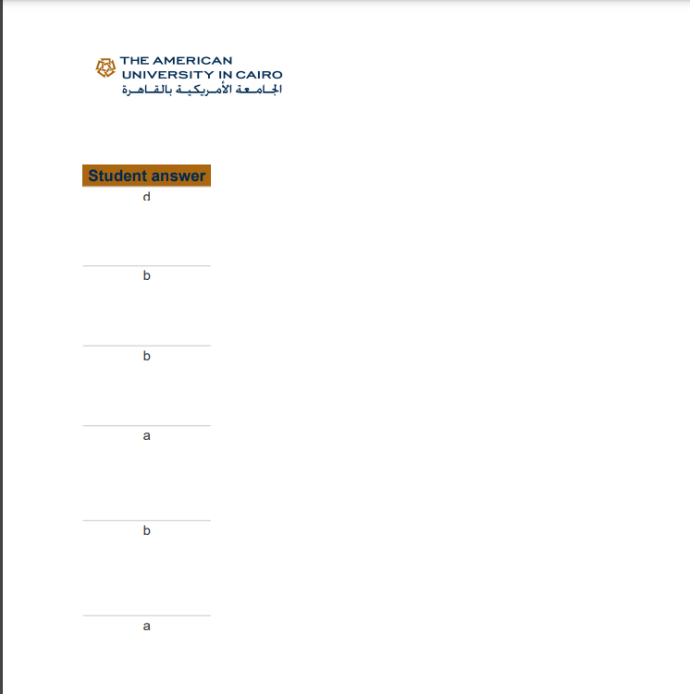
SSRS Reports: -

1. 

* Report that takes the instructor ID and returns the name of the courses that he teaches and the number of students per course.

2.

* Report that takes the student ID and returns the grades of the student in all courses. %

3.

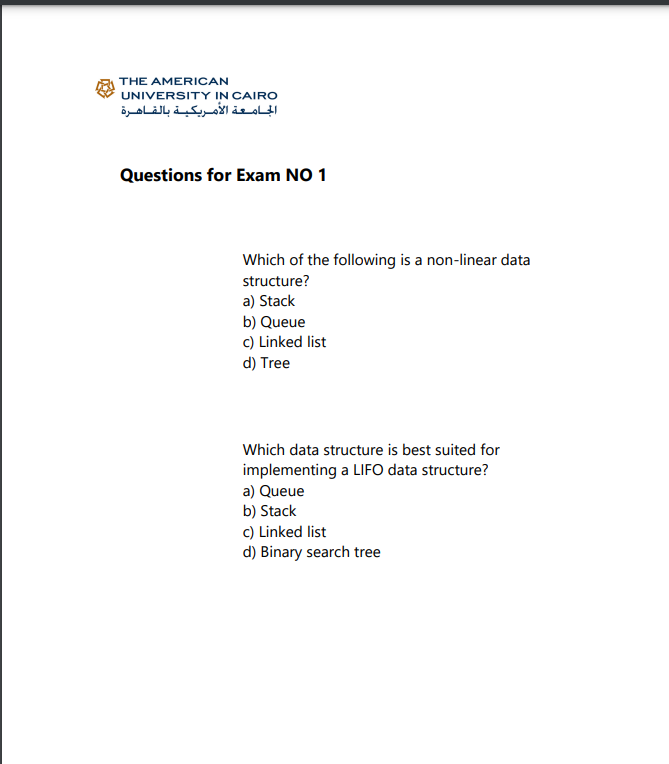
* Report that takes exam number and the student ID then returns the Questions in this exam with the student answers.

4.

* Report that takes course ID and returns its topics

5.

* Report that returns the students information according to Department No parameter.

6.

* Report that takes exam number and returns the Questions in it and chocies [freeform report]

Power BI dashboard samples: -

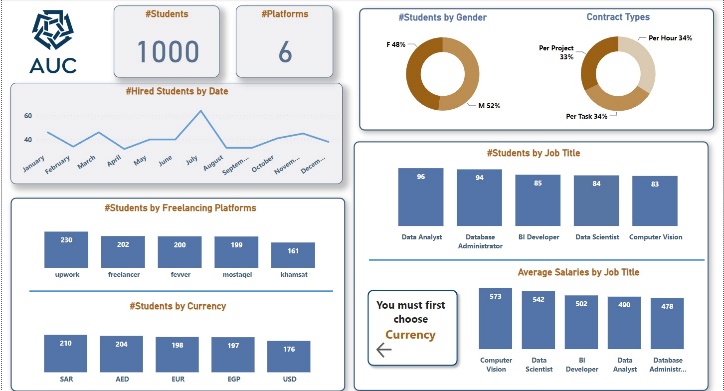




Tableau Dashboard Samples: -

