Examination System

Supervisor by Eng. Rami Nagi

Presented by:
Hagar Mohamed ahmed
Mohamed Amer hamed
Hend Mohamed abdelfatah
Rewaa Srour Abdeltawab
Mohamed Ashraf Lotfy

Table of contents

1-Introduction

Welcome to the documentation for the Examination System, a comprehensive platform designed to facilitate the management of educational assessments, student data, and reporting functionalities. This system caters to the diverse needs of educational institutions, instructors, and students, providing robust features to streamline examination processes, track student progress, and generate insightful reports.

OVERVIEW:

The Examination System is built upon a relational database model, encapsulating various entities, relationships, and attributes essential for efficient data management. Key components of the system include:

- Entities and Relationships: The system incorporates entities such as Student, Intake, Branch, Department, Instructor, Courses, Questions, Exam, Freelance, Certificate, and Company, each interconnected through well-defined relationships. These entities represent fundamental aspects of the educational ecosystem, facilitating seamless data organization and retrieval.

- Stored Procedures: To enhance database functionality, stored procedures have been implemented to handle common operations such as selecting, updating, and deleting records for each entity. Additionally, specialized procedures have been developed for exam random generation, student answer submission, and exam correction, ensuring smooth execution of critical examination processes.

- Reporting and Analytics: The system offers robust reporting capabilities, leveraging SQL Server Reporting Services (SSRS) for generating comprehensive reports on student performance, examination outcomes, and other relevant metrics. Moreover, interactive dashboards powered by Power BI enable stakeholders to gain actionable insights through intuitive visualizations and data analysis tools.

- Desktop App: the system features a complementary desktop application built using Python and the Tkinter library. This desktop application provides users with a convenient and intuitive interface for accessing key features and functionalities of the Examination System.
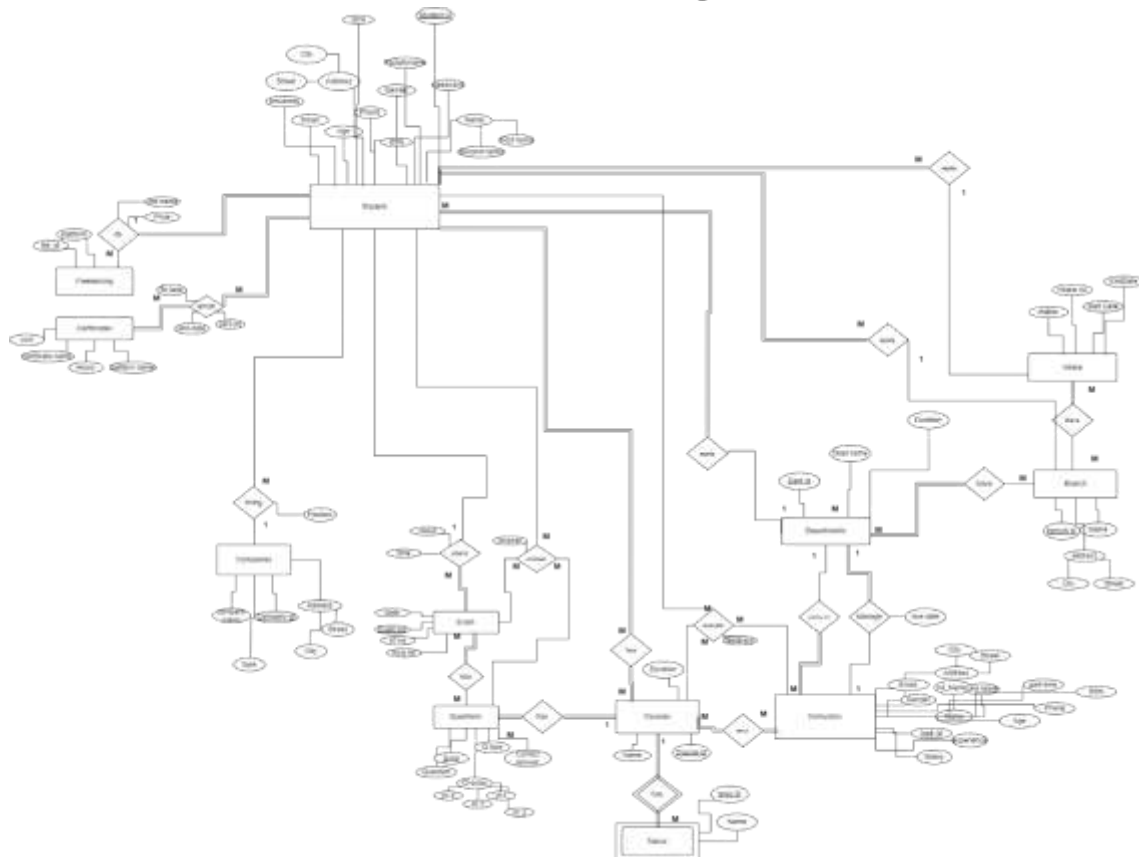
2-Implementation life cycle

The implementation life cycle is an iterative process needed to ensure the system meets the requirements that outline the phases a system goes through, from entity relationship diagrams to interactive dashboards.

3-Entity relationship diagram

ENTITIES:

Student:
- Represents students enrolled in the examination system.
- Has a many-to-one (M:1) relationship with entities Intake, Branch, and Department, indicating that each student is associated with one intake, one branch, and one department.

Intake:
- Represents the intake sessions within the system.
- Participates in a many-to-many (M:M) relationship with the Branch entity, indicating that multiple branches can operate in different intake sessions and vice versa.

Branch:
- Represents branches within the system.
- Has a many-to-many (M:M) relationship with the Department entity, signifying that a branch can offer multiple departments and vice versa.

Department:
- Represents departments within branches.
- Participates in a one-to-many (1:M) relationship with the Instructor entity, indicating that one department can have multiple instructors.
- Participates in a one-to-one (1:1) relationship with the Instructor entity, indicating that one department can be managed by one instructors.

Instructor:
- Represents instructors teaching courses.
- Has a one-to-many (1:M) relationship with the Department entity, signifying that an instructor can work in one department.

Courses:
- Represents courses offered within the system.
- Participates in a many-to-many (M:M) relationship with the Instructor entity, indicating that multiple instructors can teach a course and vice versa.
- Contains a weak entity called Topics, with a one-to-many (1:M) relationship, signifying that each course consists of multiple topics.

Questions:
- Represents questions related to courses.
- Participates in a one-to-many (1:M) relationship with the Courses entity, indicating that each course contains multiple questions.

Exam:

- Represents exams conducted within the system.
- Participates in a many-to-many (M:M) relationship with the Questions entity, signifying that an exam comprises multiple questions and vice versa.

Freelance:

- Represents freelance jobs undertaken by students.
- Has a one-to-many (1:M) relationship with the Student entity, indicating that each student can undertake multiple freelance jobs.

Certificate:

- Represents certificates earned by students.
- Participates in a many-to-many (M:M) relationship with the Student entity, indicating that a student can earn multiple certificates and vice versa.

Company:

- Represents companies where students work after graduation.
- Has a many-to-one (M:1) relationship with the Student entity, indicating that each student may work in one company after graduation, but multiple students may work for the same company.
- This addition enhances the ERD by providing a link between students and their respective employers post-graduation. It allows the system to track students' career paths and employment status, providing valuable information for academic and career counseling purposes.

RELATIONSHIPS:

- There is a ternary relationship between Student, Question, and Exam entities to store student answers for each question in a generated exam.
- Another ternary relationship exists between Student, Courses, and Instructor entities to save feedback for each course with the respective instructor.
- There's a many-to-many (M:M) relationship between Student and Course entities with a Grade attribute to track student grades in each course.
- This ERD provides a comprehensive overview of the examination system's database structure, detailing the entities, their relationships, and attributes. It ensures data integrity and facilitates efficient management of student, course, exam, and instructor information within the system.

4-Database Creation & Mapping

Database Creation

During the development of the Examination System database on SQL Server, comprehensive constraints and features were implemented to ensure data integrity, track changes, and facilitate future data warehouse creation. Here's a detailed summary of the constraints and features incorporated:

Unique Constraints:

Unique constraints were applied to the Course Name, Topic Name, and Intake Name columns, guaranteeing that each entry in these columns remains distinct and prevents duplicates.

Primary Key Constraints and Identity:

Each table in the database has a primary key constraint to ensure unique identification of records.Identity columns were utilized for most tables to automatically generate unique primary key values, excluding the Instructor and Student tables.

Referential Integrity Constraints:

Referential integrity constraints were established to maintain relationships between parent and child tables.
On delete actions were configured:
For some relationships, the "set null" action was specified, ensuring that child records are set to null if their parent records are deleted.
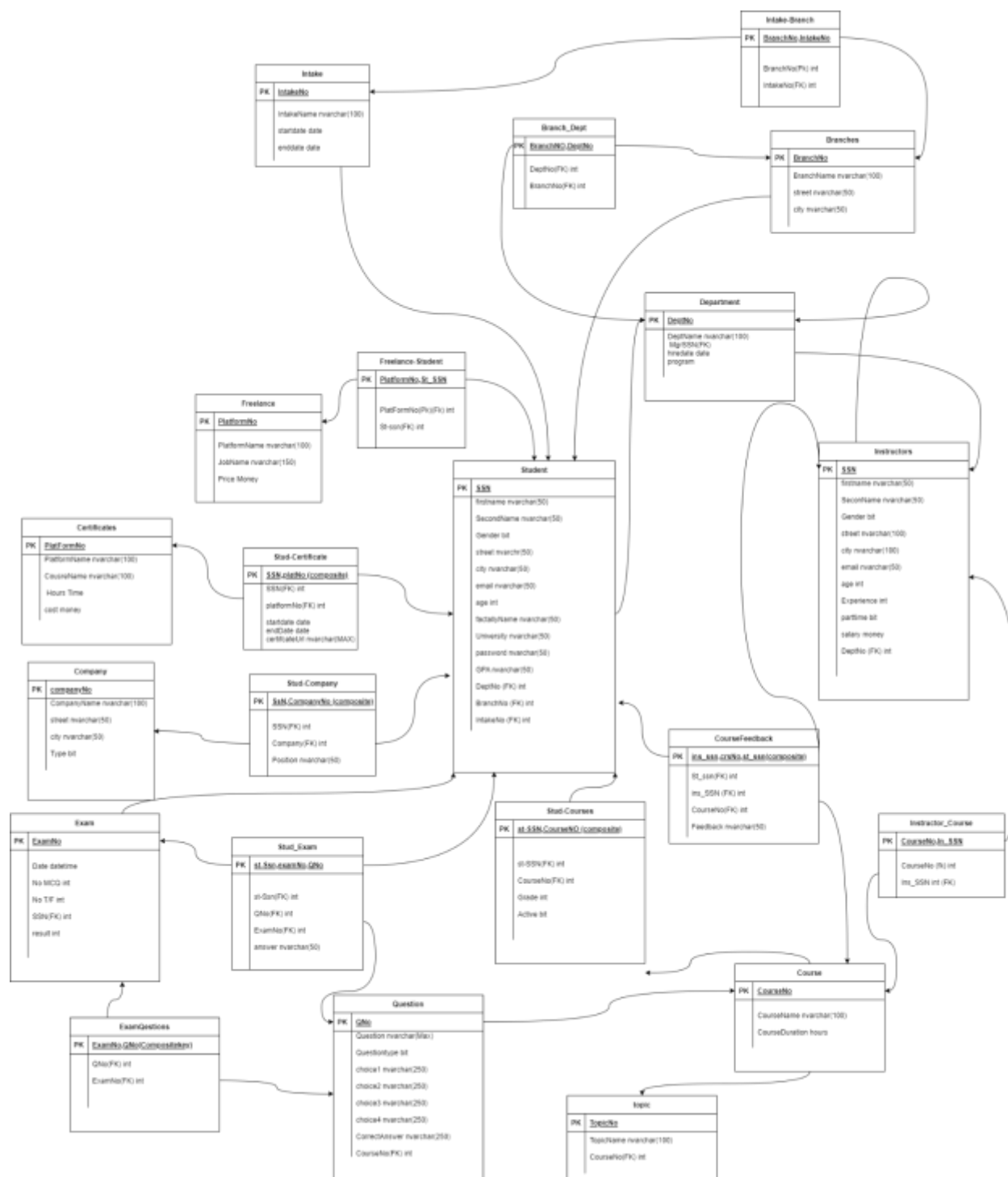For others, the "cascade" action was chosen, facilitating automatic deletion or updating of related child records when corresponding parent records are deleted or updated.

Last Modified Date:

A "Last Modified Date" column was added to each table, capturing the timestamp of the last update made to a row.A default constraint using the GETDATE() function was applied to automatically populate this column with the current timestamp upon row creation or modification.

Data Warehouse Preparation:

The inclusion of the "Last Modified Date" column with default values aids in tracking changes and will be invaluable during the creation of a data warehouse.This feature ensures that the data warehouse can efficiently identify and extract updated records, facilitating timely and accurate reporting and analysis.

Database Entity-Relationship Diagram

Intake-Branch
PK	BranchNo,IntakeNo
	BranchNo(FK) int
	IntakeNo(FK) int

Intake
PK	IntakeNo
	IntakeName nvarchar(100)
	startdate date
	enddate date

Branch_Dept
PK	BranchNO,DeptNo
	DeptNo(FK) int
	BranchNo(FK) int

Branches
PK	BranchNo
	BranchName nvarchar(100)
	street nvarchar(50)
	city nvarchar(50)

Department
PK	DeptNo
	DeptName nvarchar(100)
	Mgr/SSN(FK)
	hiredate date
	program

Freelance-Student
PK	PlatformNo,St_SSN
	PlatFormNo(Pk)(FK) int
	St-ssn(FK) int

Freelance
PK	PlatformNo
	PlatformName nvarchar(100)
	JobName nvarchar(150)
	Price Money

Student
PK	SSN
	firstname nvarchar(50)
	SecondName nvarchar(50)
	Gender bit
	street nvarchar(50)
	city nvarchar(50)
	email nvarchar(50)
	age int
	facletyName nvarchar(50)
	University nvarchar(50)
	password nvarchar(50)
	GPA nvarchar(50)
	DeptNo (FK) int
	BranchNo (FK) int
	IntakeNo (FK) int

Instructors
PK	SSN
	firstname nvarchar(50)
	SecondName nvarchar(50)
	Gender bit
	street nvarchar(100)
	city nvarchar(100)
	email nvarchar(50)
	age int
	Experience int
	parttime bit
	salary money
	DeptNo (FK) int

Certificates
PK	PlatFormNo
	PlatformName nvarchar(100)
	CourseName nvarchar(100)
	Hours Time
	cost money

Stud-Certificate
PK	SSN,platNo (composite)
	SSN(FK) int
	platformNo(FK) int
	startdate date
	endDate date
	certificateUrl nvarchar(MAX)

Company
PK	companyNo
	CompanyName nvarchar(100)
	street nvarchar(50)
	city nvarchar(50)
	Type bit

Stud-Company
PK	SsN,CompanyNo (composite)
	SSN(FK) int
	Company(FK) int
	Position nvarchar(50)

CourseFeedback
PK	ins_ssn,crsNo,st_ssn(composite)
	St_ssn(FK) int
	ins_SSN (FK) int
	CourseNo(FK) int
	Feedback nvarchar(50)

Instructor_Course
PK	CourseNo,In_SSN
	CourseNo (fk) int
	ins_SSN int (FK)

Exam
PK	ExamNo
	Date datetime
	No MCQ int
	No T/F int
	SSN(FK) int
	result int

Stud_Exam
PK	st_Ssn,examNo,QNo
	st-Ssn(FK) int
	QNo(FK) int
	ExamNo(FK) int
	answer nvarchar(50)

Stud-Courses
PK	st_SSN,CourseNO (composite)
	st-SSN(FK) int
	CourseNo(FK) int
	Grade int
	Active bit

Course
PK	CourseNo
	CourseName nvarchar(100)
	CourseDuration hours

ExamQestions
PK	ExamNo,QNo(Compositekey)
	QNo(FK) int
	ExamNo(FK) int

Question
PK	QNo
	Question nvarchar(Max)
	Questiontype bit
	choice1 nvarchar(250)
	choice2 nvarchar(250)
	choice3 nvarchar(250)
	choice4 nvarchar(250)
	CorrectAnswer nvarchar(250)
	CourseNo(FK) int

topic
PK	TopicNo
	TopicName nvarchar(100)
	CourseNo(FK) int

5-Important Stored Procedures

_RANDOMEXAMGENERATOR:

Purpose: This stored procedure generates a random exam based on the specified parameters, including the number of multiple-choice questions (MCQ), true/false questions (TF), and single select questions (SSN) for a given course.

Parameters:

- @MCQ: Number of multiple-choice questions.
- @TF: Number of true/false questions.
- @SSN: Number of single select questions.
- @CrsName: Name of the course for which the exam is generated.

Functionality:

- Inserts a new exam record into the exam table.
- Retrieves random questions from the Question table based on the specified course and question types.
- Inserts selected questions into the ExamQuestions table associated with the generated exam.

```sql
11  CREATE Proc [dbo].[_RandomExamGenerator] @MCQ int ,@TF int, @SSN int ,@CrsName
        VARCHAR(100)
12  AS
13      declare @ExamNo int
14      insert into exam (Num_MCQ,Num_TF,SSN)
15      values(@MCQ,@TF,@SSN)
16      set @ExamNo=SCOPE_IDENTITY()
17      insert into ExamQuestions (ExamNo,QNo)
18      select @ExamNo,QNo from (
19      SELECT TOP(@TF)
            q.QNo,q.Question,q.QType,q.Choice1,q.Choice2,q.Choice3,q.Choice4,q.CorrectA
            nswer FROM dbo.Question q,dbo.Course c
20              WHERE c.CourseNo=q.CourseNo AND c.CourseName=@CrsName AND  q.QType=0
21              ORDER BY NEWID()) as t1
22      union all
23      select @ExamNo,QNo from (
24      SELECT TOP(@MCQ)
            q.QNo,q.Question,q.QType,q.Choice1,q.Choice2,q.Choice3,q.Choice4,q.CorrectA
            nswer FROM dbo.Question q,dbo.Course c
25              WHERE c.CourseNo=q.CourseNo AND c.CourseName=@CrsName  AND  q.QType=1
26              ORDER BY NEWID()) as t2
27
28  GO
29
```

_EXAMANSWERS:

Purpose: This stored procedure records a student's answer to a specific question in an exam.

Parameters:

@SSN: Student's social security number.
@ExamNo: Exam number.
@QNo: Question number.
@Answer: Student's answer to the question.

Functionality:

Inserts the student's answer into the Stud_Exam table, associating it with the corresponding student, exam, and question.

```sql
10
11    CREATE Proc [dbo].[_ExamAnswers] @SSN int , @ExamNo int , @QNo
      int ,@Answer varchar(50)
12   as
13   Declare @Comb bit
14   select @Comb=case when exists (select 1 from Stud_Exam where St_SSN
15    =@SSN and ExamNo=@ExamNo and QNo=@QNo)
16    then 1 else 0 end
17   if @Comb=1
18    print 'Student already answered this Question'
19    else
20   if exists (select * from Exam where SSN=@SSN and ExamNo=@ExamNo)
21   and exists (select * from ExamQuestions where ExamNo=@ExamNo and
       QNo=@QNo)
22   insert into Stud_Exam (St_SSN,ExamNo,QNo,Answer)
23    values(@SSN,@ExamNo,@QNo,@Answer)
24    else
25    print 'This Student can not answer this question'
26    GO
27
28
```

10 %

_CORRELATION_EXAM:

Purpose: This stored procedure checks the correctness of a student's answer to a specific question.

Parameters:

- @QNo: Question number.
- @Answer: Student's answer to the question.

Functionality:

- Compares the student's answer with the correct answer and returns 'Correct' if they match, otherwise returns 'Wrong'.
- Useful for evaluating student responses and providing feedback on exam performance

```sql
/****** Object:  StoredProcedure [dbo].[_correlation_exam]      Script Date:
  2/26/2024 7:17:53 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

ALTER PROC [dbo].[_correlation_exam] @QNo INT, @answer VARCHAR(MAX)
AS
BEGIN
    DECLARE @Correct VARCHAR(MAX)
    SELECT @Correct=Q.CorrectAnswer FROM dbo.Question Q
    WHERE Q.QNo=@QNo
    IF @Correct=@answer
        BEGIN
            SELECT 'Correct'
        END
    ELSE
        SELECT 'Wrong'
END
GO
```

_UPDATEEXAMRESULT:

Purpose: This stored update exam result after exam corretion

Parameters:

- @ExamNo:Exam number.
- @result: Student's result.

Functionality:

- Update exam result after exam correction

```
9   GO
10
11  create Proc [dbo].[_UpdateExamResult] @result int ,@ExamNo int
12
13  as
14  if Exists (select 1 from Exam where ExamNo=@ExamNo)
15  update Exam
16  set Result=@result , last_modified_date=GETDATE()
17  where ExamNo=@ExamNo
18  else
19  print('This Exam number does not exitsts to update')
20  GO
21
22
```

133 %

Connected. (1/1)   (local) (16.0 RTM) HAGARPC\PC (67) ITI_GP 00:00:00 0 rows

Ln 23   Col 1   Ch 1   INS

23°C صافي غالبًا   ENG   7:16 PM 2/26/2024

5-Reports (SSRS)

Students Information by Department:

Student Grades in All Courses:

Instructor's Courses and Student Enrollment:

Instructor ID 1001

|◁ < 1 of 1 > ▷| ↻ 100% ⌄ 💾 ⌄ 🖨

Instructor Name: Anwar Basm

Course Name	Number Of Students
Data Visualization	23
Data Warehousing	23
SQL for Business Intelligence	23

Course Topics:

Course ID 1

|◁ < 1 of 1 > ▷| ↻ 100% ⌄ 💾 ⌄ 🖨 Find | Next

Couser Name: Java Fundamentals

Topic Name
OOP in Java
introduction to programming with java

Exam Questions and Choices (Freeform Report):

Exam Number 1166

Exam Report

Student Name : Fahad Jamal   **Course Name : Data Manipulation and Analysis**

Which statistical measure provides the middle value of a dataset?

(1) Mean
(2) Median
(3) Mode
(4) Range

What is the role of data visualization in data analysis?

(1) To hide data complexity
(2) To make data more difficult to understand
(3) To present data in a graphical format for easier interpretation
(4) To limit access to sensitive data

Student Answers in Exam

SSN 1006   Exam No 1166

Fahad Jamal Answers in Data Manipulation and Analysis

In data analysis, what does ETL stand for?

a-Extract, Transform, Load   b-Encrypt, Transfer, Log

c-Evaluate, Test, Learn   d-Execute, Track, Log

Student answer: Extract, Transform, Load   Correct

Which language is commonly used for querying and manipulating databases?

a-Java   b-Python

c-SQL   d-Ruby

Student answer: SQL   Correct

6-Dashboards (powerbi)

Our Dashboard Consists of :

- Student (Overview , Details, Related KPI)

- Instructor (Overview , Details, Related KPI)

- Courses (Overview , Details, Exam details)

- Department (Overview , Details, Distribution)

- Branch((Overview , Details, distribution)

- Certificates((Overview, details)

- Company(Overview, details)

- Freelance (Overview, details)

Student Dashboards:

- Display number of active students, overall student performance.
- Show students distribution over branches

Student details
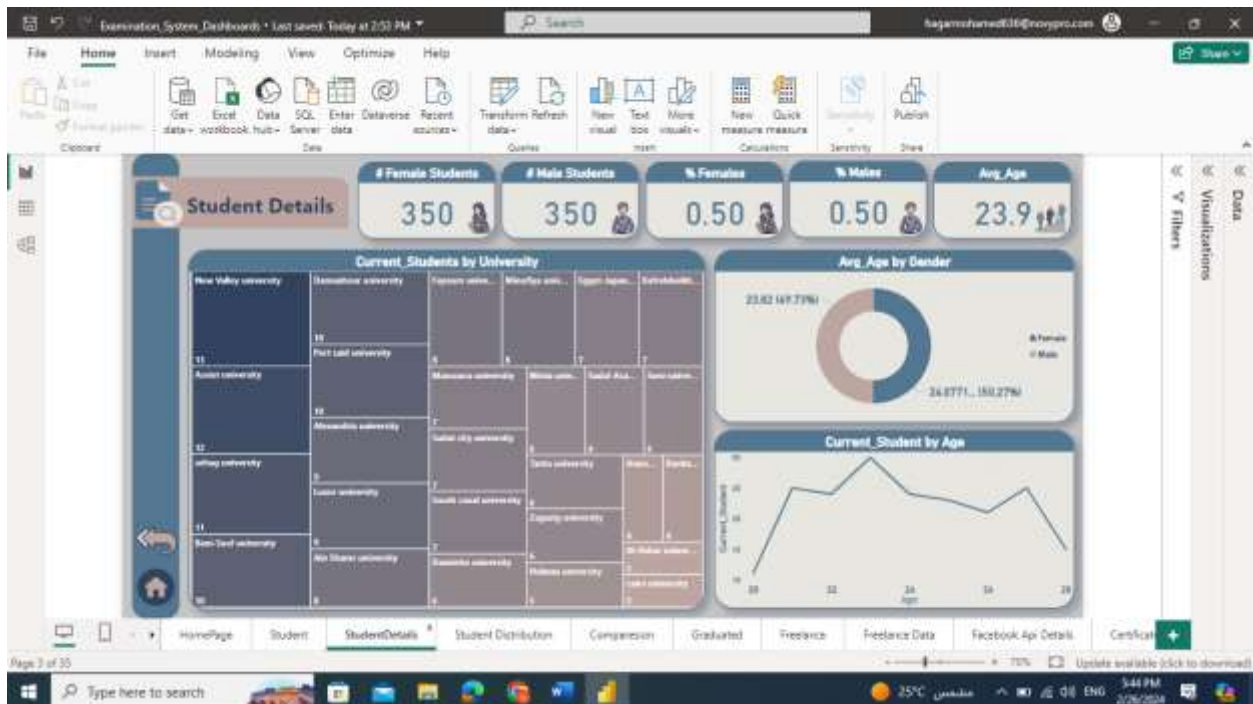
Student distribution
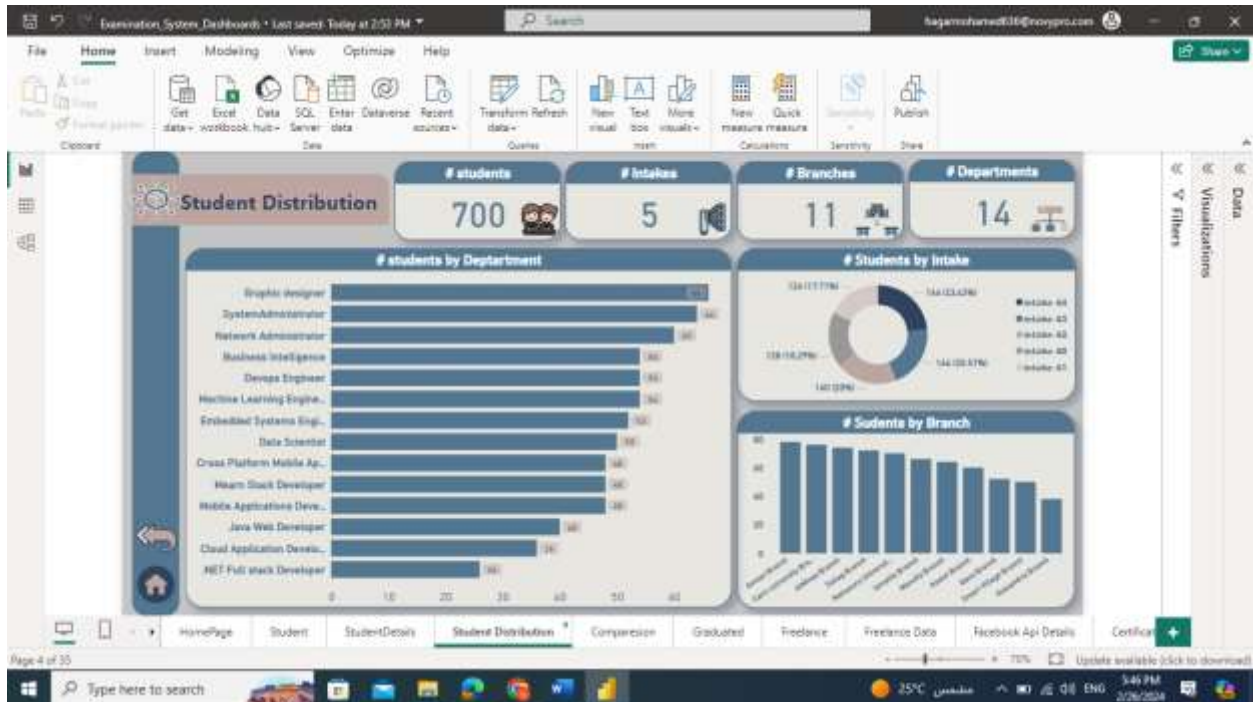
Graduated Student :

Tracks the progress of graduates.

Freelance:

Highlight the number of students engaged in freelancing jobs and their success stories.
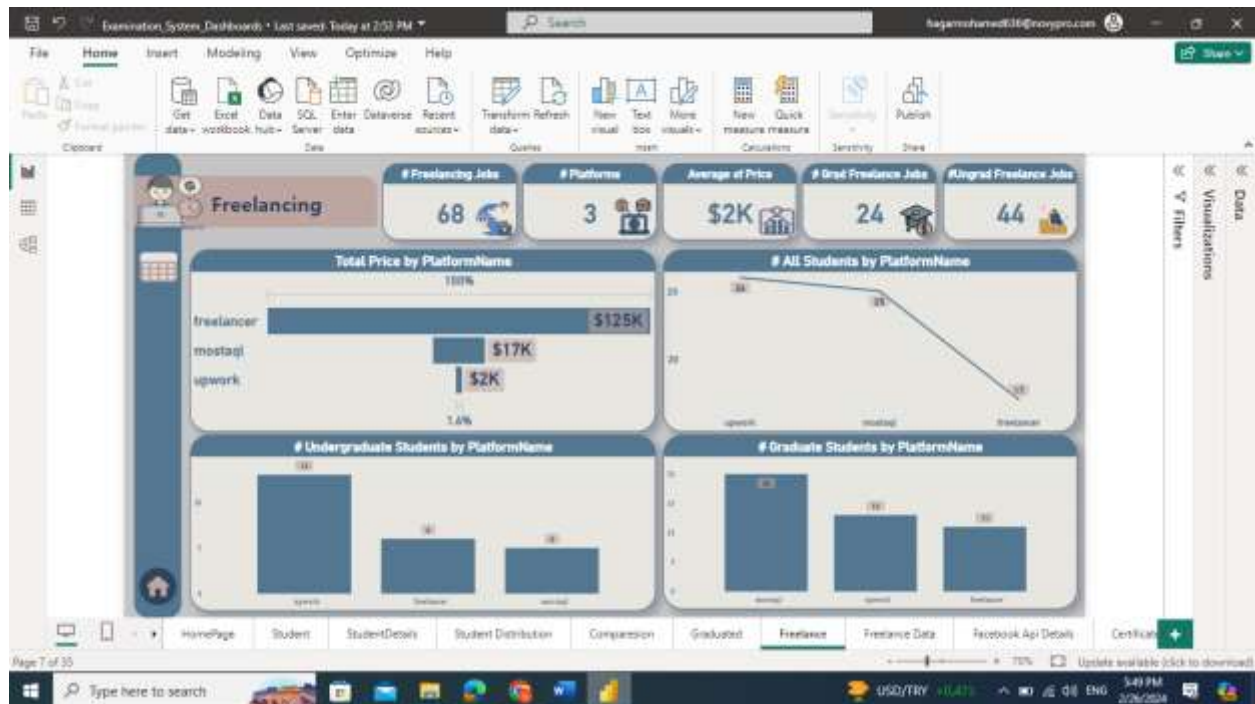
Certificates :

Showcase the number of certificates earned by students and popular certification programs.

Certificate Details
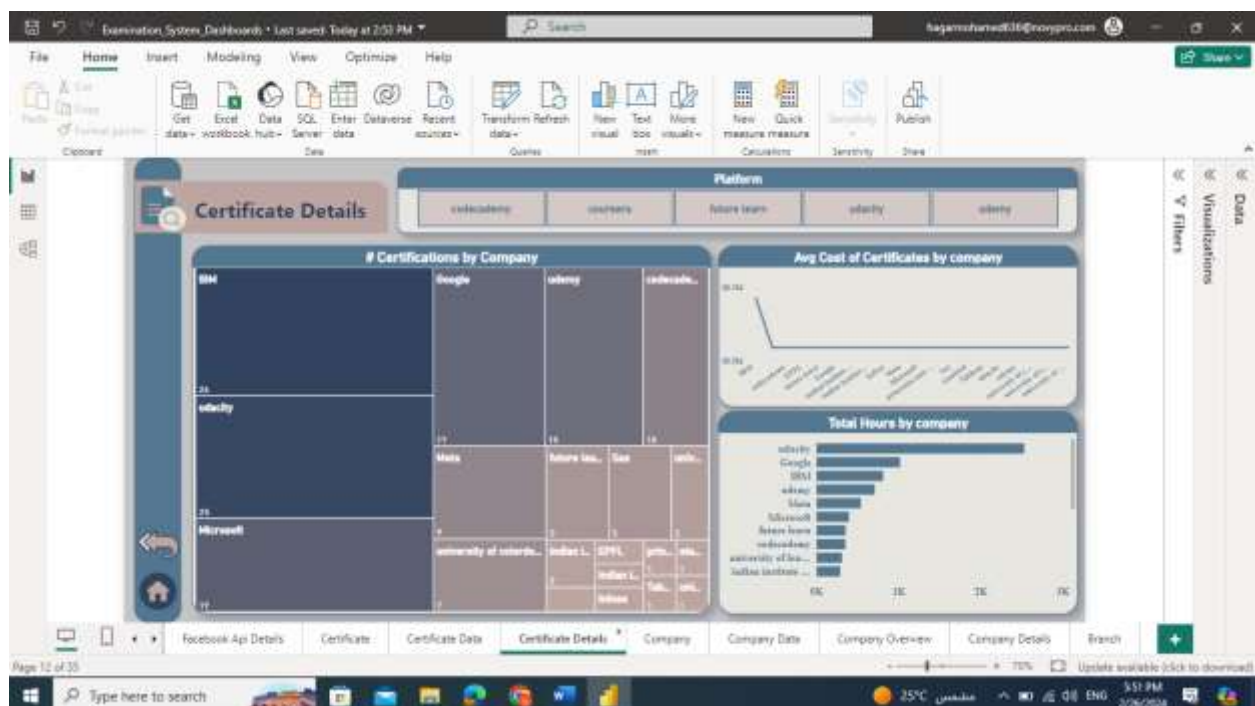
Company :

Provide information on companies recruiting students, available positions, and hiring trends.

Company overview
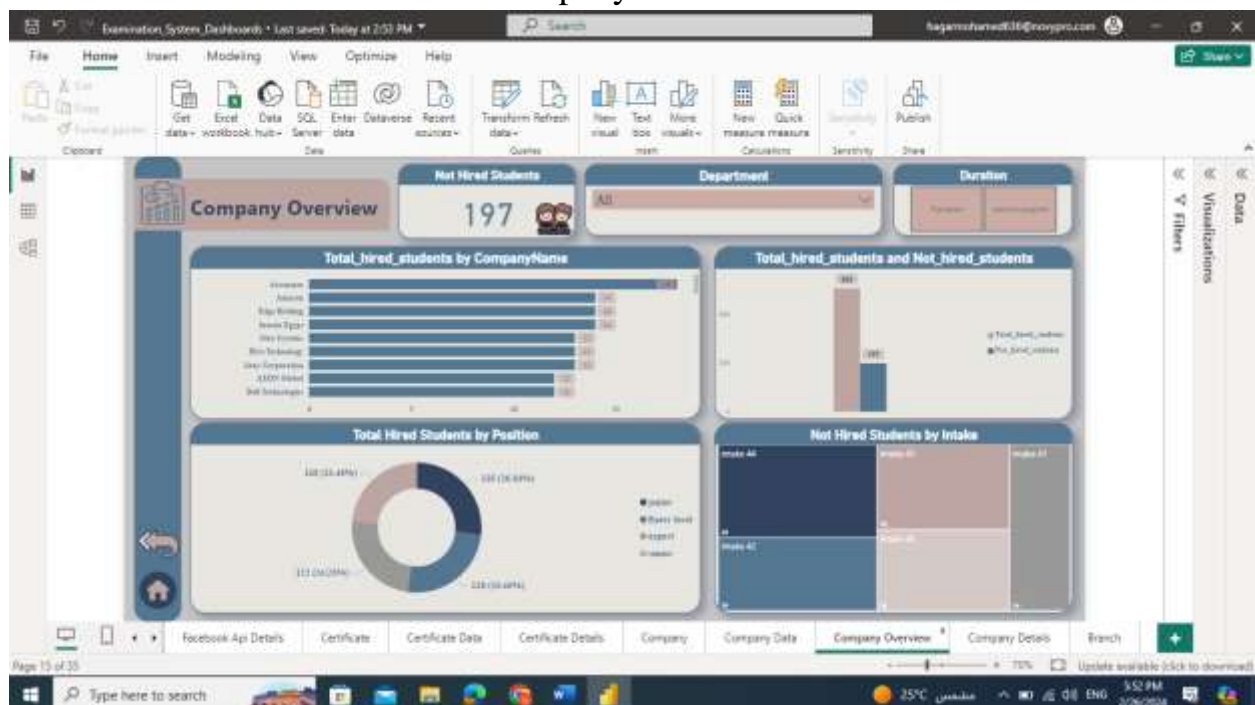
Company Details

Branch :

Show branch details based on metrics like show top student in each branch

Branch details
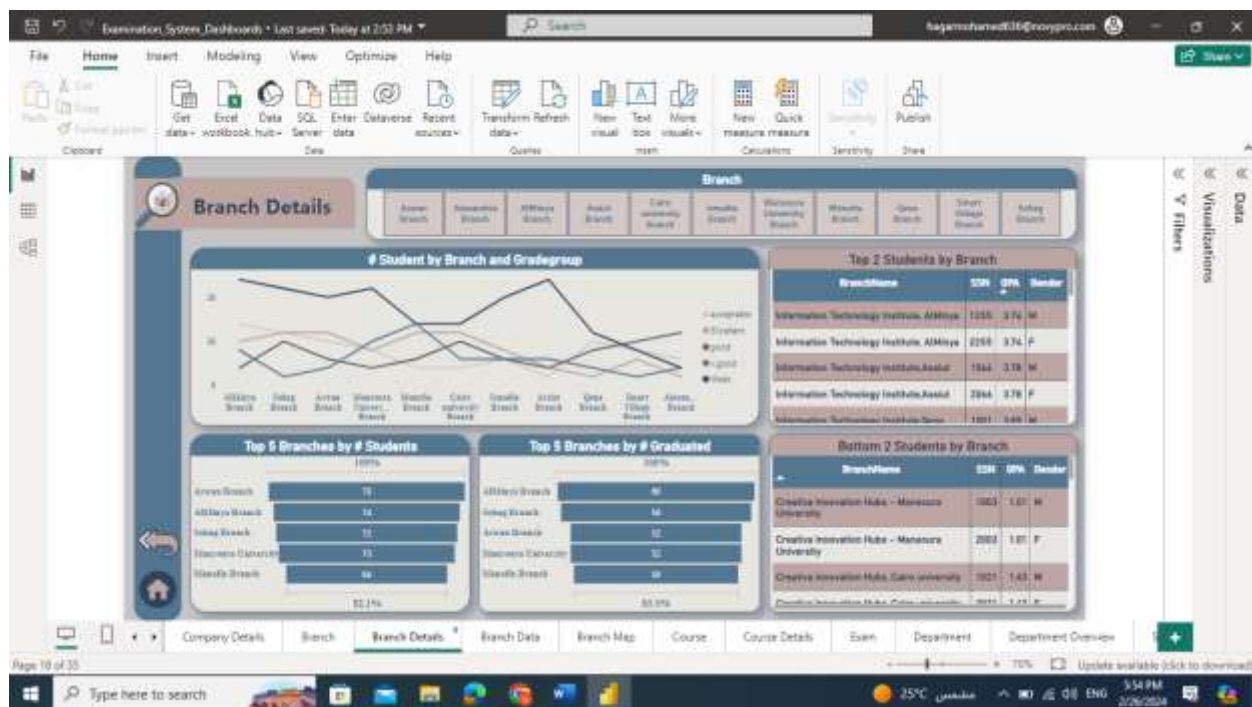
Branch Map
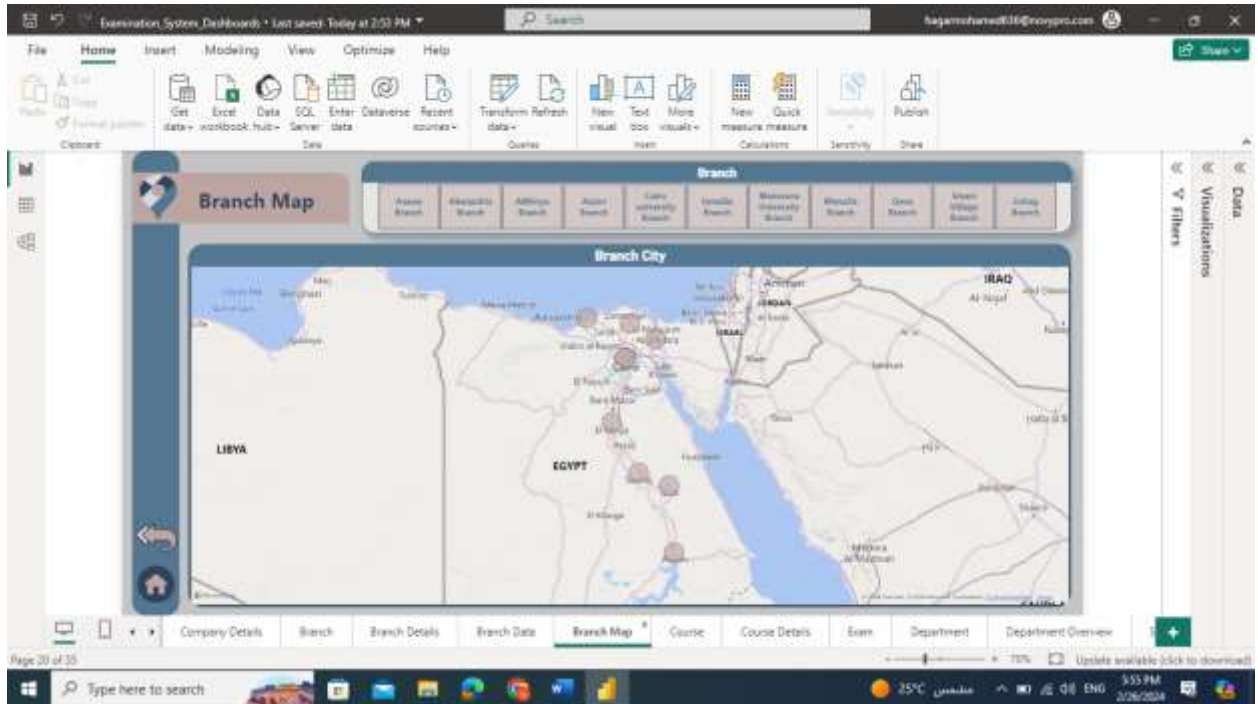
Course

Displays the number of topics and names of courses , the performance of different courses based on metrics like average grades and courses by student's grade.

Course details
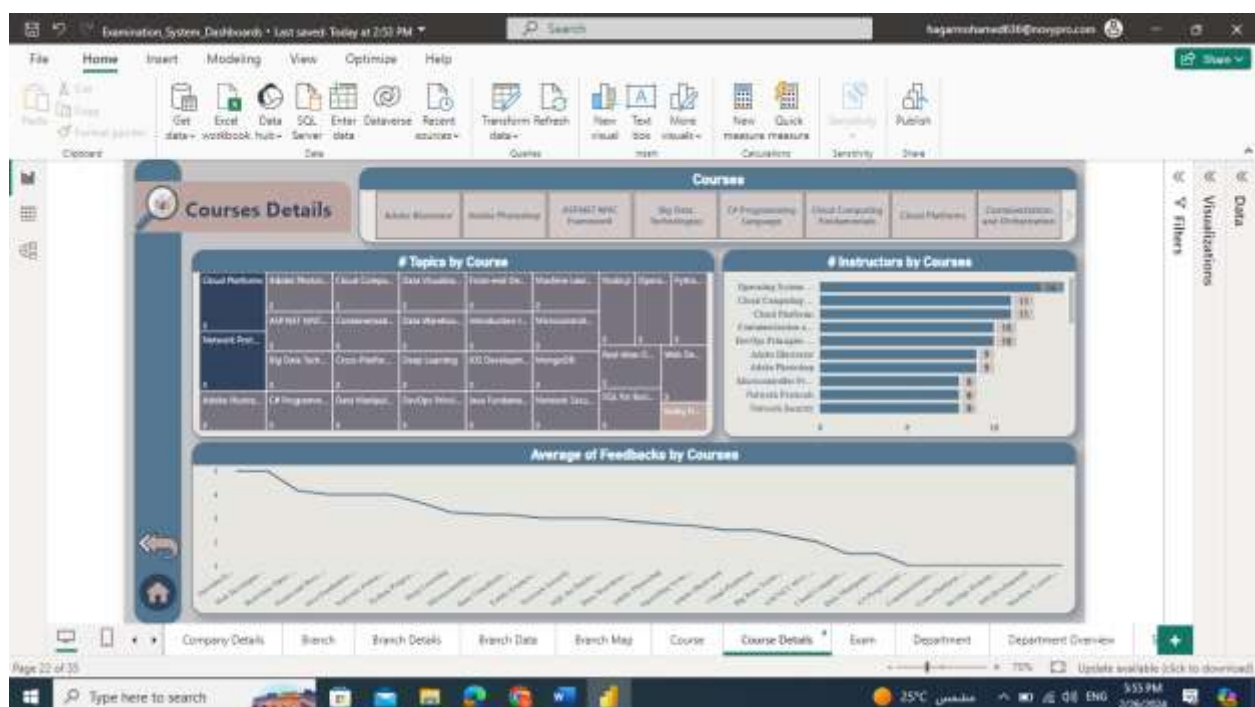
Exam:

Shows the student performance in the exam and number of exams created , average scores, and performance trends over time.
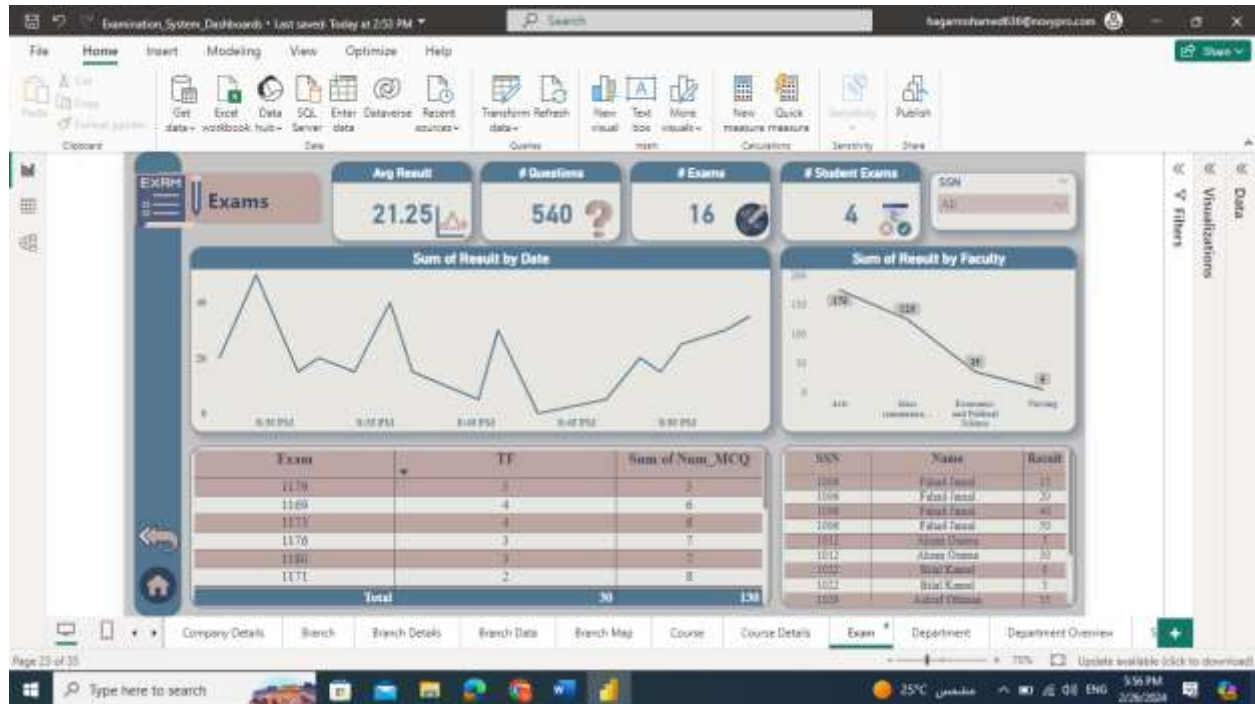
Department:

Displays the number of departments, number of student in each department ,
Shows number of department by duration
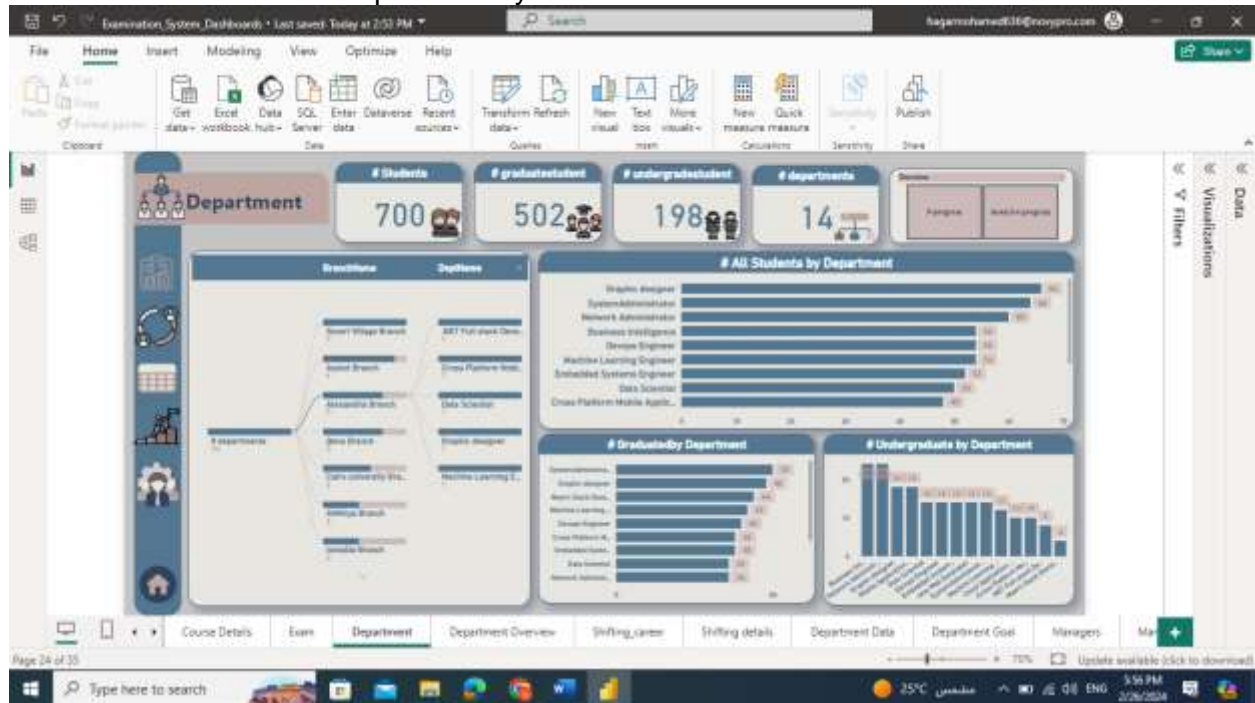
Department overview

Shifted career:

Number of student who shifted their career in each department and how they are distributed over the departments

Shifted details
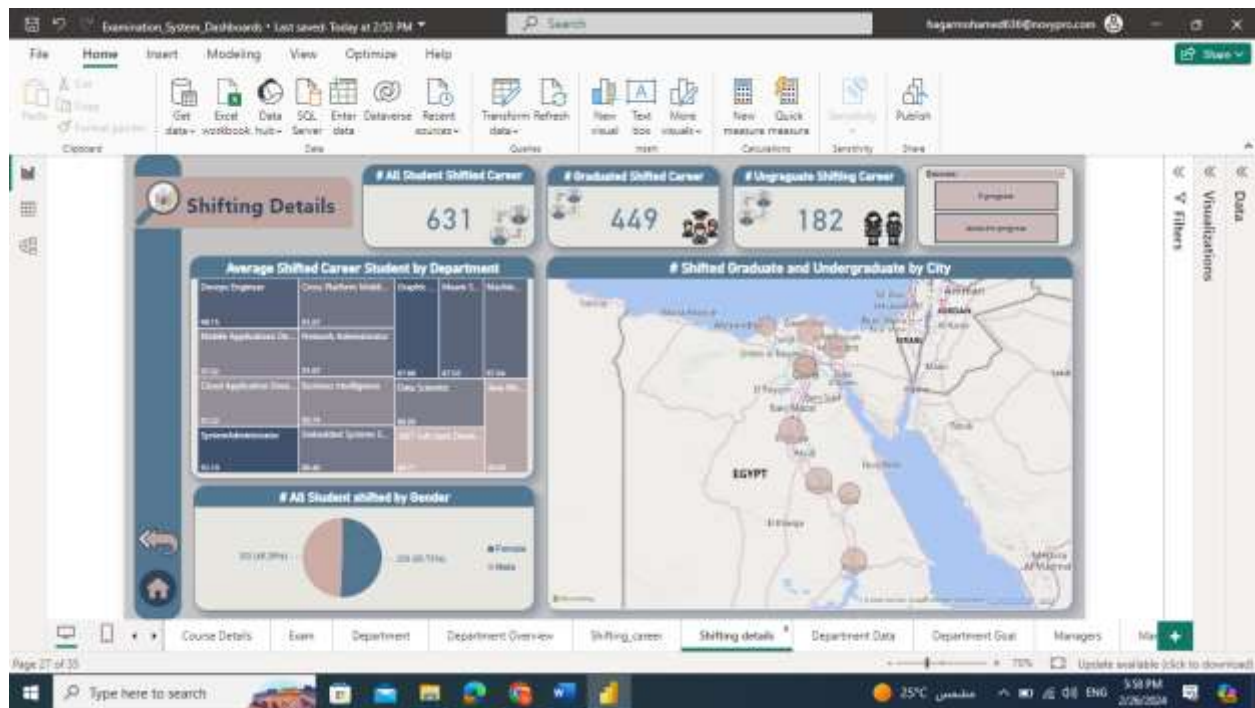
Department goal

Managers
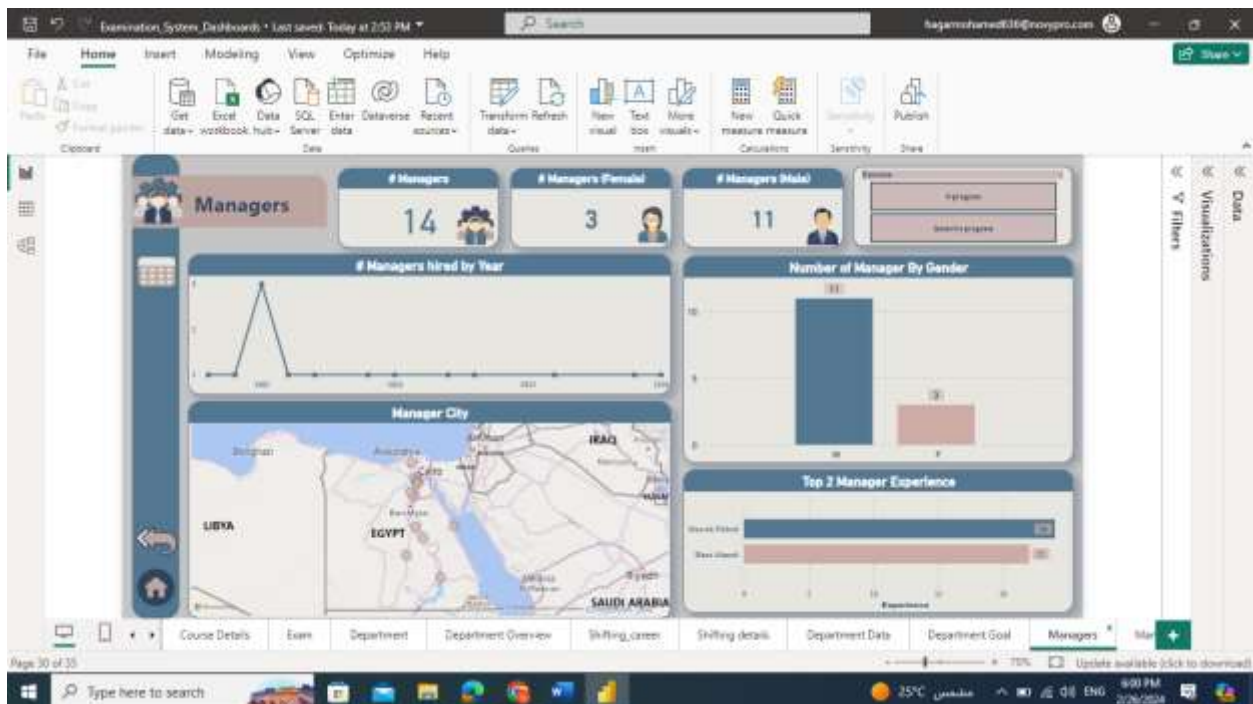
Instructors:

Provide information on instructor effectiveness, course popularity.
Show instructors distribution over branches

Instructor details

Facebook_API for the team :

7-Datawarehouse creation

The Data Warehouse component of the Examination System facilitates comprehensive data analysis and reporting for ITI staff. Utilizing a Galaxy schema approach, the warehouse consolidates data from various source tables into a structured format optimized for reporting and analysis. Below are the key components and processes involved in populating the Data Warehouse:
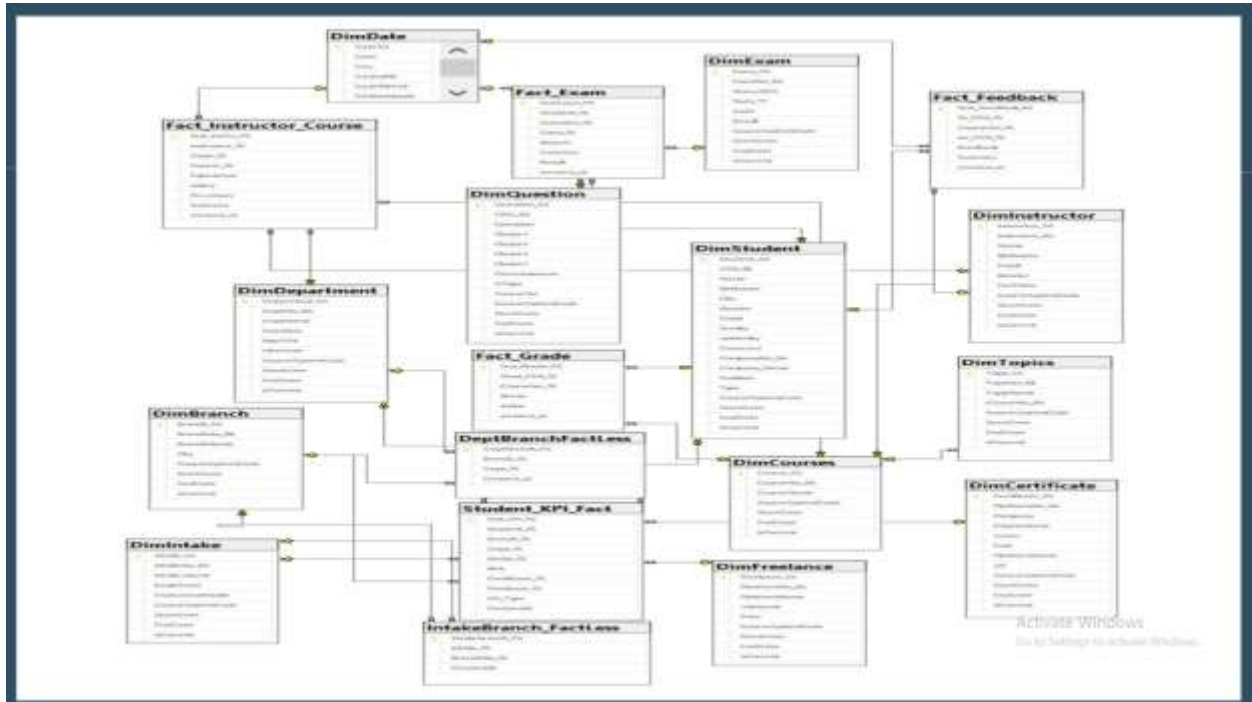
Dimension Tables:

DimStudent, DimDepartment, DimBranch, DimIntake, DimFreelance, DimCertificate, DimCourses, DimTopics, DimInstructor: These tables capture descriptive attributes related to students, departments, branches, intakes, freelancers, certificates, courses, topics, and instructors, respectively.

Fact Tables:

IntakeBranch_FactLess, DeptBranchFactLess, Student_KPI_Fact, Fact Instructor_Course, Fact_Feedback, Fact_Grade, Fact_Exam: These tables store quantitative measures and performance indicators associated with student performance, instructor-course relationships, feedback, grades, and exam results.

Source Query Execution:

The provided SQL queries retrieve data from the source database (ITI_GP) to populate the warehouse tables. These queries fetch data related to students, departments, branches, intakes, freelancers, certificates, courses, topics, instructors, feedback, grades, questions, exams, and exam results.

```sql
1   use iti_gp
2
3   -------------------DimStudent
4   select s.SSN as SSN_Bk,
5   s.Name,s.Birthdate,s.City,s.Gender,s.Email,s.Faculty
6   ,s.University,s.Password,c.CompanyNo AS ComapnyNO_BK,
       c.CompanyName,c.Type,sc.Position
7   from Student as S
8   lEFT jOIN Stud_Company AS sc ON sC.SSN=S.SSN
9   LEFT jOIN Company AS c ON C.CompanyNo=SC.CompanyNo
10  ---------------Dim Department
11  Select D.DeptNo as DeptNo_BK,D.DeptName,d.Duration,d.MgrSSN,d.HireDate
12  from Department as D
13  --------------------Dim Branch
14  Select b.BranchNo as BranchNo_Bk,b.BranchName,b.City
15  from Branches as B
16  -----------------Dim Intake
17  select I.intak_id as IntakeNo_BK , i.intak_name,i.startDate
18  ,i.endDate
19  from Intake as I
20  ------------------ IntakeBranch_FactLess
21  select  intak_id as Intake_FK,BranchNo as BranchNo_FK ,
       last_modified_date
22  from Branch_has_Intakes
23  where last_modified_date >= --user last load day
24  -------------------DeptBranchFactLess
25  Select BranchNo as Branch_FK , DeptNo as Dept_FK , last_modified_date
26  from Branch_Dept
```

110 %

ETL Process:

Extract, Transform, Load (ETL) processes extract data from the source tables, apply transformations as necessary, and load it into the corresponding tables in the Data Warehouse (ITI_GP_DW).The ETL process ensures data consistency, integrity, and alignment with the dimensional model, enabling accurate reporting and analysis.

Student dimension ETL
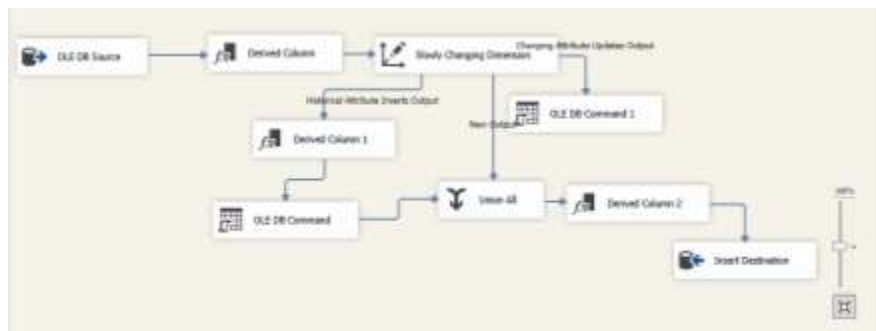
Student_KPI_Fact
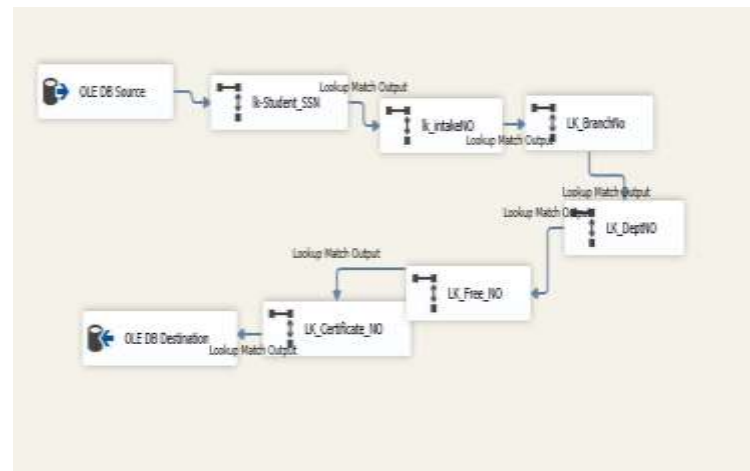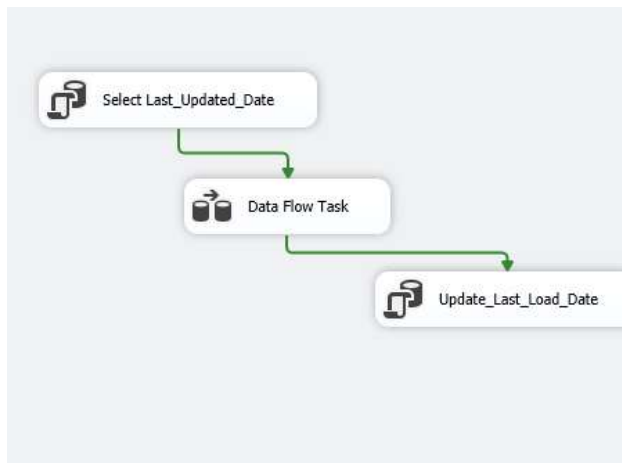
Audit Trail:

Audit tables such as audit_IntakeBranch track the last load date of data into specific warehouse tables. This information aids in monitoring data freshness and integrity, ensuring that reports are based on the most recent data available
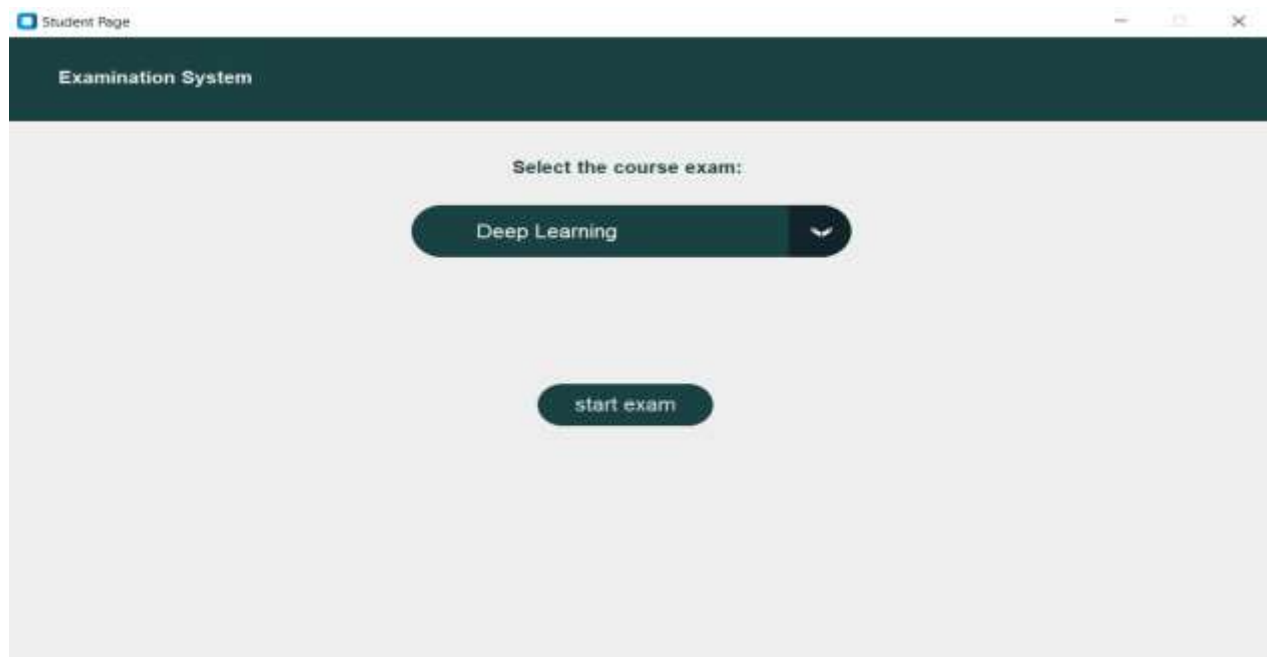
9-Desktop application

Correct student answers:

Examination System

2 . What is the purpose of the backpropagation algorithm in deep learning?   0\5

○ Initializing neural network weights
○ Forward pass computation
○ Optimizing hyperparameters
○ Updating weights based on prediction errors

incorrect
correct answer : Updating weights based on prediction errors

3 . Which deep learning architecture is commonly used for image recognition tasks?   0\5

○ Recurrent Neural Network (RNN)
○ Long Short-Term Memory (LSTM)
○ Convolutional Neural Network (CNN)
○ Autoencoder

incorrect
correct answer : Convolutional Neural Network (CNN)

View grade

Examination System

bad luck Fahad Jamal

Total exam grade : 50

ActualScore : 0

To pass exam must get score >= 25

The number of correct answe : 0

The number of incorrect answer : 10

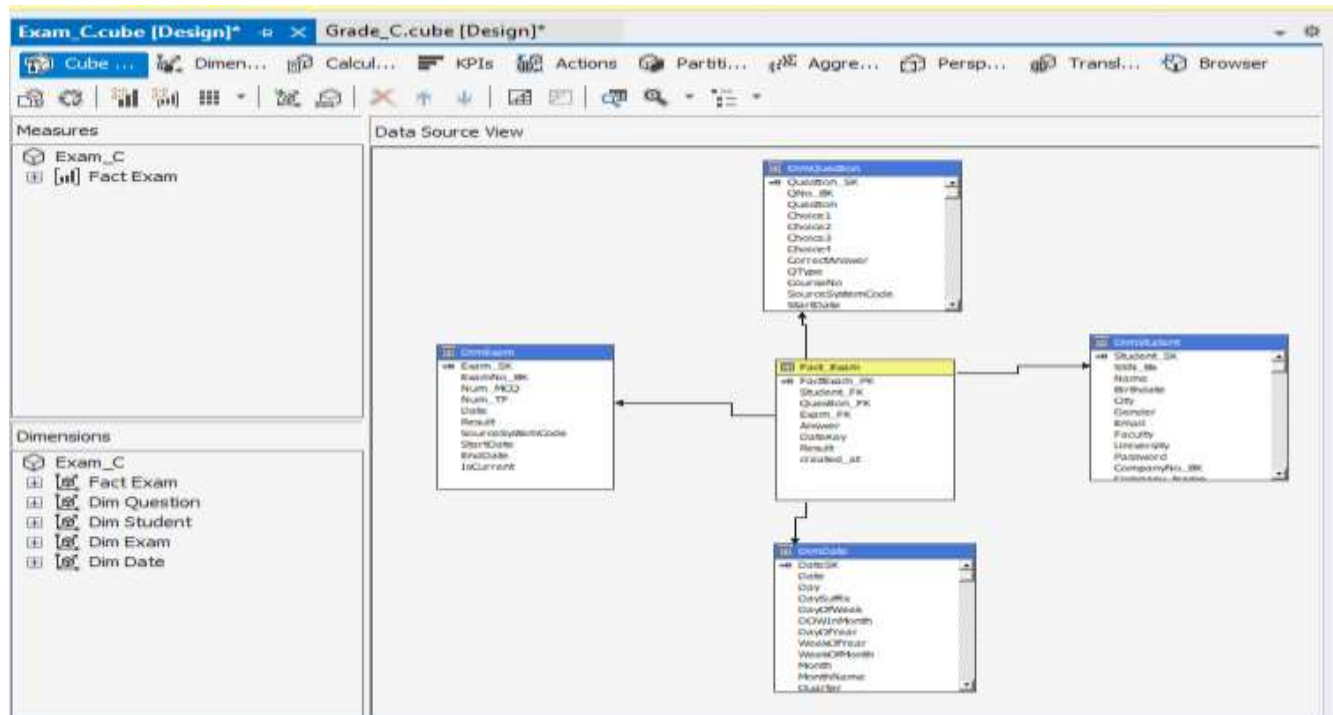1 . Which activation function is commonly used in the hidden layers of a neural network for deep learning?   0\5

○ Sigmoid
○ Tanh
○ ReLU (Rectified Linear Unit)
○ Softmax

incorrect
correct answer : ReLU (Rectified Linear Unit)

10-Cube (SSAS)

Exam Cube

Student Grade

File Home Insert Page Layout Formulas Data Review View Automate Developer Help   💬 Comments ⌸ Share ▾

Row Labels	▾ Grade
Adobe Illustrator	383
Adobe Photoshop	381
ASP.NET MVC Framework	166
Big Data Technologies	280
C# Programming Language	184
Cloud Computing Fundamentals	207
Cloud Platforms	221
Containerization and Orchestration	305
Cross-Platform Development	233
Data Manipulation and Analysis	298
Data Visualization	463
Data Warehousing	433
Deep Learning	296
DevOps Principles and Practices	296
Entity Framework	173
Front-end Development	363
Introduction to Mobile Development	340

Grade

Total

Gender

F

M

Unknown

Course Name ▾

tmpBCD2 +