ITI Examination System

Track: **Power BI**

Branch: **Mansoura**

Team Members

Shorok Mohamed **Awatef Samir**

Meran Ahmed **Aml Kamal**

Yasmeen Shamakh

Feb. 2023

Acknowledgment

First, praises and thanks to God, the Almighty, for His showers of blessing throughout our work to complete the project successfully.

The success and outcome of this project required a lot of guidance and assistance from many people, and we are extremely privileged to have got this all along the completion of our project.

All that we have done is only due to such supervision and assistance and we would not forget to thank them.

We thank **Eng. Rami Nagi** for his supervision, providing us an opportunity to do the project and giving us all support and guidance, which made us complete the project duly.

We are thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching staffs.

Project Team

Project Overview

This project was done as a graduation project for the "information technology institute" (ITI)

Examination System is an application that designed and developed for students and instructor. The system helps students to take online exam and know their evaluations. It also helps instructor to upload the questions and answers in the database and they can see the students who fail or pass the exam. The software is developed using C# programming language and database.

To use the application, the users must login with an email and password that has been inserted before in the data base.

In this application there are two users; the student and the instructor but each of them has a certain function.

The student function is to take an exam (by taking the question from data base through the stored procedure: exam generation) and answer this exam (by insertion these answers through the stored procedure: exam answer) and knowing their evaluations in courses.

The instructor function is to insert exam's questions and their choices to the data base, delete or update them, correction of exam through the stored procedure : exam corrections and knowing grads of the students.

Table Of Content

2.Tools

- Online Website "draw.io" (ER Diagram, Mapping)
- SQL Server Management Studio
- Redgate, SSIS (Filling data)
- SSRS (Reports)
- Power BI (Dashboard)

3.Business Case

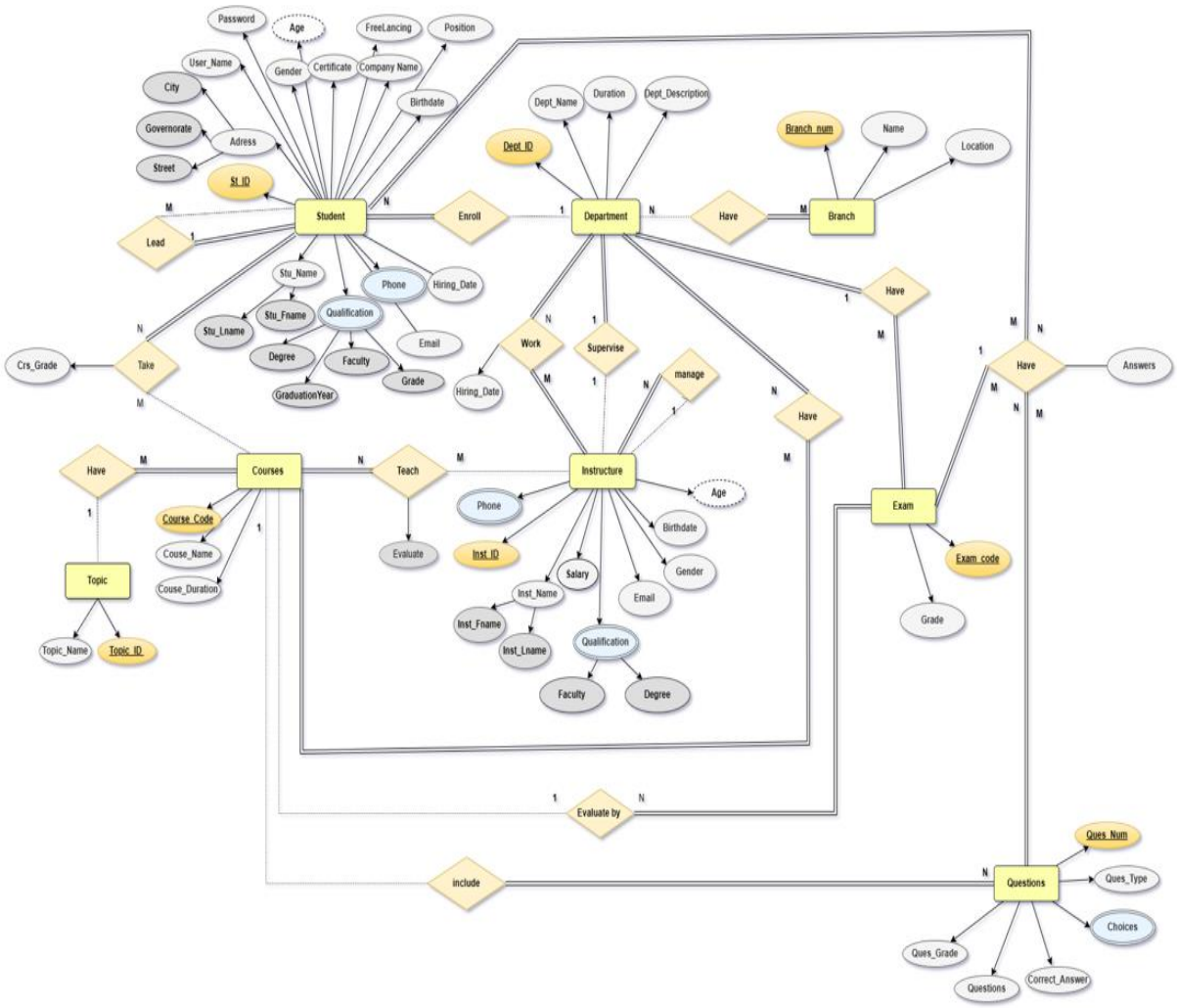
The examination system includes several entities which are the following description:

- Each student enroll in one department, studies many courses, has many exams and only one student is leader.
- Each Department is supervised by one instructor and has many courses.
- Each branch has many departments.
- Each instructor works in many departments, teaches several courses, there is always one instructor assigned to manage other instructors.
- Each course has many topics, includes many questions, and is evaluated by exams.
- Questions have different types: MCQ or True&False.

4.ERD

After determining the entities that we will need for the schema and add attributes to each entity.

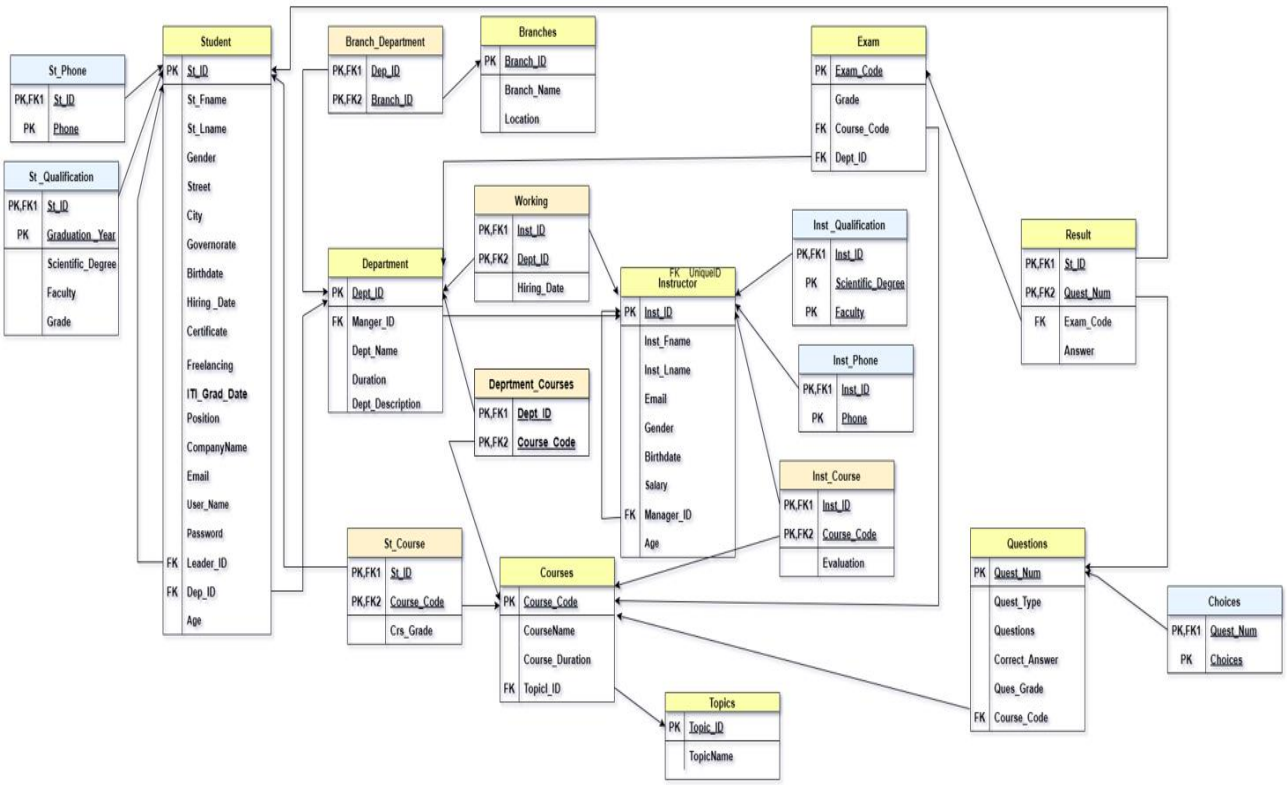
Then define the relationships between entities and add cardinality and participation to every relationship to move the next step.



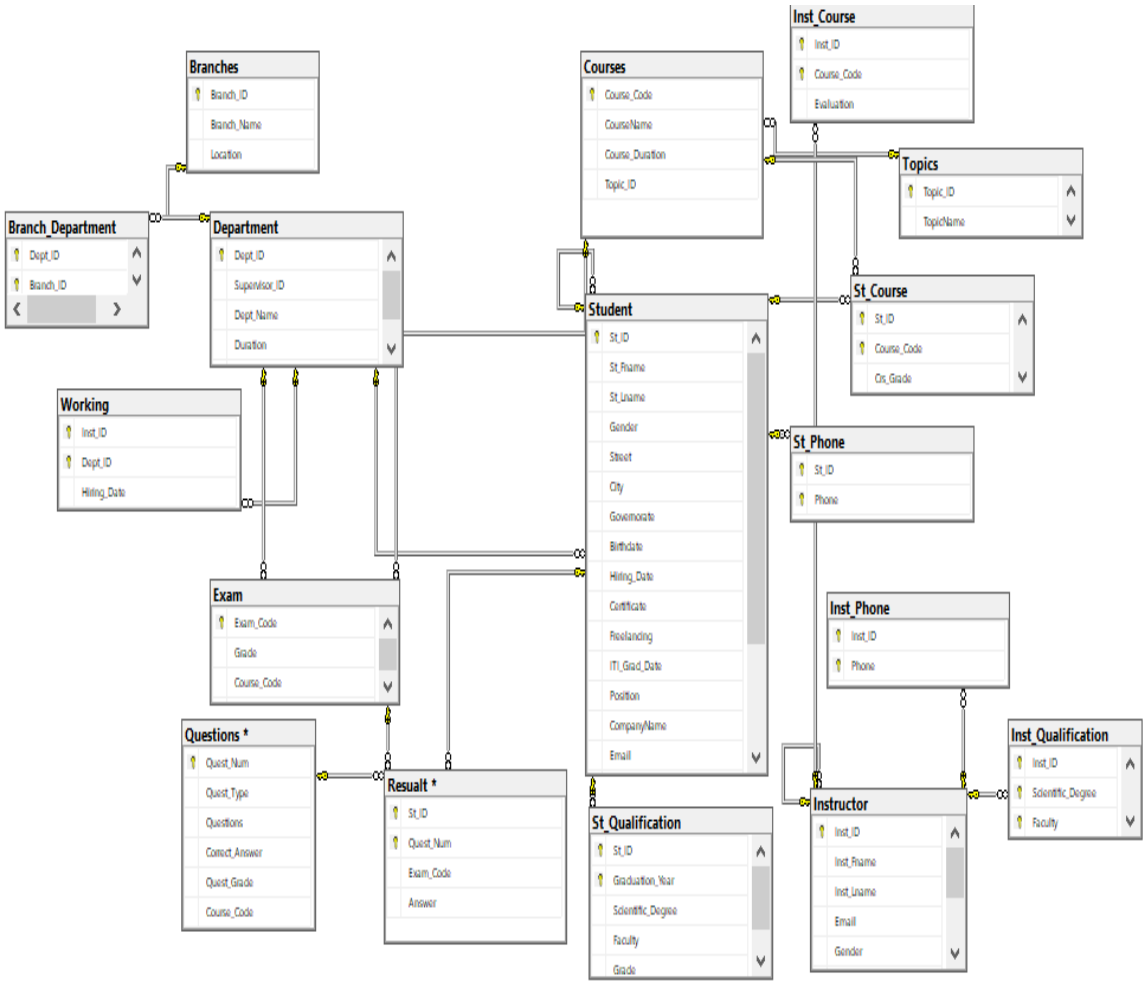
5.Mapping

The first step in data mapping is to determine which data needs to be moved or restructured, and the goal is to make your data more structured, cohesive, and accessible.

By data mapping we know the number of tables we will design and build the data base diagram through the relationship between the primary key in one table and the foreign key in another table.



6.Diagram of Database



7.Database Dictionary

7.1.Table: Student

Column Name	Data Type	Constraints
St_ID	int	primary key, not null
St_Fname	varchar(50)	Accept Null Value
St_Lname	varchar(50)	Accept Null Value
Gender	varchar(5)	check (Gender = 'M' or Gender= 'F'), Accept Null Value
Street	varchar(50)	Accept Null Value
City	varchar(50)	Accept Null Value
Governorate	varchar(50)	Accept Null Value
Birthdate	date	Accept Null Value
Hiring_Date	date	Accept Null Value
Certificate	varchar(150)	Accept Null Value
Freelancing	varchar(150)	Accept Null Value
ITI_Grad_Date	date	Accept Null Value
Position	varchar(50)	Accept Null Value
CompanyName	varchar(50)	Accept Null Value
Email	varchar(50)	Accept Null Value
User_Name	varchar(20)	Accept Null Value
Password	varchar(50)	Accept Null Value
Leader_ID	int	Foreign key, Accept Null Value
Dept_ID	int	Foreign key, Accept Null Value
Age		

- Age is derived attribute and calculated according to this equation:
  Age = (year (getdate ())-year (Birthdate))

7.2.Table: Instructor

Column Name	Data Type	Constraints
Inst_ID	int	primary key,not null
Inst_Fname	varchar(50)	Accept Null Value
Inst_Lname	varchar(50)	Accept Null Value
Email	varchar(50)	Accept Null Value
Gender	varchar(20)	check (Gender = 'M' or Gender= 'F'), Accept Null Value
Birthdate	date	Accept Null Value
Salary	int	check (Salary>2000), Accept Null Value
Manager_ID	int	Foreign key, Accept Null Value
Age		

- Age is derived attribute and calculated according to this equation:
  Age = (year (getdate ())-year (Birthdate))

7.3.Table: Department

Column Name	Data Type	Constraints
Dept_ID	int	primary key,not null
Supervisor_ID	int	Foreign key, Accept Null Value
Dept_Name	varchar(50)	Accept Null Value
Duration	varchar(20)	Accept Null Value
Dept_Description	varchar(150)	check (Duration = '3 monthes' or Duration = '9 monthes'), Accept Null Value

7.4.Table: Branches

Column Name	Data Type	Constraints
Branch_ID	int	primary key,not null
Branch_Name	varchar(50)	Accept Null Value
Location	varchar(50)	Accept Null Value

7.5.Table: Courses

Column Name	Data Type	Constraints
Course_Code	int	primary key,not null
CourseName	varchar(20)	Accept Null Value
Course_Duration	int	Accept Null Value
Topic_ID	int	Foreign key, Accept Null Value

7.6.Table: Topics

Column Name	Data Type	Constraints
Topic_ID	int	primary key,not null
TopicName	varchar(20)	Accept Null Value

7.7.Table: Exam

Column Name	Data Type	Constraints
Exam_Code	int	primary key,not null
Grade	int	Accept Null Value
Course_Code	int	Foreign key, Accept Null Value
Dept_ID	int	Foreign key, Accept Null Value

7.8.Table: Questions

Column Name	Data Type	Constraints
Quest_Num	int	primary key,not null
Quest_Type	varchar(20)	Accept Null Value
Questions	varchar(150)	Accept Null Value
Correct_Answer	varchar(50)	Accept Null Value
Quest_Grade	int	Accept Null Value
Course_Code	int	Foreign key, Accept Null Value

Tables Created Due To Relationship Between Entities

7.9.Table: St_Qualification

- According to Multivalued Attribute: Qualification

Column Name	Data Type	Constraints	
St_ID	int	Foreign key,Not null	Composite primary key
Graduation_Year	int	Not null	
Scientific_Degree	varchar(20)	Accept Null Value	
Faculty	varchar(50)	Accept Null Value	
Grade	varchar(20)	Accept Null Value	

7.10.Table: St_Phone

- According to Multivalued Attribute: Phone

Column Name	Data Type	Constraints	
St_ID	int	Foreign key,Not null	Composite primary key
Phone	char(11)	Not null	

7.11.Table: Inst_Qualification

- According to Multivalued Attribute: Qualification

Column Name	Data Type	Constraints	
Inst_ID	int	Foreign key,Not null	Composite primary key
Scientific_Degree	varchar(20)	Not null	
Faculty	varchar(50)	Not null	

7.12.Table: Inst_Phone

- According to Multivalued Attribute: Phone

Column Name	Data Type	Constraints	
Inst_ID	int	Foreign key,Not null	Composite primary key
Phone	char(11)	Not null	

7.13.Table: Working

- According To Many To Many Relationship Between Department And Instructor

Column Name	Data Type	Constraints	
Inst_ID	int	Foreign key,Not null	Composite primary key
Dept_ID	int	Foreign key,Not null	
Hiring_Date	date	Accept Null Value	

7.14.Table: Branch_Department

- According To Many To Many Relationship Between Department And Branch

Column Name	Data Type	Constraints	
Dept_ID	int	Foreign key,Not null	Composite primary key
Branch_ID	int	Foreign key,Not null	

7.15.Table: Department_Courses

- According To Many To Many Relationship Between Department And Courses

Column Name	Data Type	Constraints	
Dept_ID	int	Foreign key,Not null	Composite primary key
Course_Code	int	Foreign key,Not null	

7.16.Table: Inst_Course

- According To Many To Many Relationship Between Instructor And Courses

Column Name	Data Type	Constraints	
Inst_ID	int	Foreign key,Not null	Composite primary key
Course_Code	int	Foreign key,Not null	
Evaluation	int	Accept Null Value	

7.17.Table: St_Course

- According To Many To Many Relationship Between Student And Courses

Column Name	Data Type	Constraints	
St_ID	int	Foreign key,Not null	Composite primary key
Course_Code	int	Foreign key,Not null	
Crs_Grade	float	Accept Null Value	

7.18.Table: Choices

- According to Multivalued Attribute: Choices

Column Name	Data Type	Constraints	
Q_Number	tinyint	Foreign key,Not null	Composite primary key
Choices	nvarchar(100)	Not null	

7.19.Table: Resualt

- According To N-ary Relationship Between Student , Exam and Questions
- Answer is an Attribute on the relationship

Column Name	Data Type	Constraints	
St_ID	int	Foreign key,Not null	Composite primary key
Quest_Num	int	Foreign key,Not null	
Exam_Code	int	Foreign key, Accept Null Value	
Answer	varchar(50)	Accept Null Value	

8. Stored Procedures for Tables

8.1.Stored Procedure : Student_SP

```
create PROC Student_SP @StatementType NVARCHAR(20) = '',@St_ID int,
@St_Fname varchar(50),@St_Lname varchar(50),@Gender varchar(5),@Street
varchar(50),@City varchar(50),@Governorate varchar(50), @Birthdate date,
@Hiring_Date date,@Certificate varchar(150), @Freelancing
Varchar(150),@ITI_Grad_Date date,@Position varchar(50),
@CompanyName varchar(50), @Email varchar(50)
AS
  BEGIN try
      IF @StatementType = 'Insert'
        BEGIN
            INSERT INTO Student
                  (St_ID, St_Fname,St_Lname, Gender ,Street ,City ,
                  Governorate, Birthdate ,Hiring_Date ,Certificate ,
                  Freelancing ,ITI_Grad_Date ,Position ,
                  CompanyName , Email )
            VALUES
                  (@St_ID, @St_Fname,@St_Lname ,@Gender ,@Street ,
                   @City ,@Governorate, @Birthdate , @Hiring_Date,
                   @Certificate, @Freelancing,@ITI_Grad_Date ,@Position ,
                   @CompanyName, @Email)
         END

      IF @StatementType = 'Select'
        BEGIN
            SELECT * FROM Student WHERE St_ID=@St_ID
        END

      IF @StatementType = 'Update'
        BEGIN
            UPDATE Student
            SET   St_Fname = @St_Fname,
                  St_Lname = @St_Lname,
                  Gender= @Gender,
                  Street=@Street,
                  City =@City,
                  Governorate= @Governorate,
                  Birthdate=@Birthdate,
                  Hiring_Date=@Hiring_Date,
                  Certificate=@Certificate,
                  Freelancing =@Freelancing,
                  ITI_Grad_Date=@ITI_Grad_Date,
                  Position=@Position,
                  CompanyName=@CompanyName,
                  Email =@Email
            WHERE St_ID = @St_ID
        END
      ELSE IF @StatementType = 'Delete'
        BEGIN
```

```
            DELETE FROM Student
            WHERE St_ID = @St_ID
        END
  END try
  Begin Catch
      select 'Error'
  end catch
```

8.2.Stored Procedure : St_Phone_SP

```
create proc St_Phone_SP @StatementType NVARCHAR(20) = '',@St_ID int, @Phone
char(11)
AS
  BEGIN TRY
      IF @StatementType = 'Insert'
      BEGIN
            INSERT INTO St_Phone (St_ID, Phone)
            VALUES (@St_ID, @Phone)
      END
      IF @StatementType = 'Select'
       BEGIN
            SELECT * FROM   St_Phone where St_ID=@St_ID
       END
      IF @StatementType = 'Update'
        BEGIN
            UPDATE St_Phone
            SET Phone=@Phone Where St_ID=@St_ID
         END
      ELSE IF @StatementType = 'Delete'
        BEGIN
            DELETE FROM St_Phone WHERE St_ID=@St_ID
        END
  END TRY
  BEGIN CATCH
       Select'Error'
  END CATCH
```

8.3.Stored Procedure : St_Qualification_SP

```
create proc St_Qualification_SP @StatementType NVARCHAR(20) = '',@St_ID int
,@Graduation_Year int ,@Scientific_Degree varchar(20),@Faculty
varchar(50),@Grade Varchar(20)
AS
  BEGIN TRY
      IF @StatementType = 'Insert'
      BEGIN
         INSERT INTO St_Qualification
               (St_ID , Graduation_Year , Scientific_Degree ,Faculty ,Grade)
         VALUES (@St_ID,@Graduation_Year,@Scientific_Degree,@Faculty,@Grade)
      END
      IF @StatementType = 'Select'
       BEGIN
            SELECT * FROM St_Qualification where  St_ID=@St_ID
```

```
        END
      IF @StatementType = 'Update'
        BEGIN
            UPDATE St_Qualification
            SET Graduation_Year=@Graduation_Year,
                Scientific_Degree=@Scientific_Degree,
                Faculty=@Faculty,
                Grade=@Grade
            Where  St_ID=@St_ID
         END
      ELSE IF @StatementType = 'Delete'
        BEGIN
            DELETE FROM St_Qualification WHERE St_ID=@St_ID
        END
  END TRY
  BEGIN CATCH
       Select'Error'
  END CATCH
```

8.4.Stored Procedure : Working_SP

```
create proc Working_SP    @StatementType NVARCHAR(20) = '',
@Inst_ID int, @Dept_ID int , @Hiring_Date date
AS
  BEGIN TRY
      IF @StatementType = 'Insert'
      BEGIN
            INSERT INTO Working (Inst_ID,Dept_ID,Hiring_Date )
            VALUES (@Inst_ID,@Dept_ID,@Hiring_Date )
      END
      IF @StatementType = 'Select'
       BEGIN
            SELECT * FROM  Working Where  Inst_ID=@Inst_ID
       END

      IF @StatementType = 'Update'
        BEGIN
            UPDATE Working
            SET Dept_ID=@Dept_ID,
                Hiring_Date=@Hiring_Date
            Where Inst_ID=@Inst_ID
         END
      ELSE IF @StatementType = 'Delete'
        BEGIN
            DELETE FROM Working WHERE Inst_ID=@Inst_ID
        END
  END TRY
  BEGIN CATCH
       Select'Error'
  END CATCH
```

8.5.Stored Procedure : Branches_SP

```
Create proc Branches_SP @StatementType NVARCHAR(20) = '',@Branch_ID int ,
                        @Branch_Name varchar(50),@Location varchar(50)
AS
  BEGIN TRY
      IF @StatementType = 'Insert'
      BEGIN
            INSERT INTO Branches (Branch_ID,Branch_Name,Location)
            VALUES (@Branch_ID, @Branch_Name,@Location )
      END
      IF @StatementType = 'Select'
       BEGIN
            SELECT * FROM Branches Where Branch_ID=@Branch_ID
       END
      IF @StatementType = 'Update'
        BEGIN
            UPDATE Branches
            SET Branch_Name=@Branch_Name,
                Location=@Location
            Where Branch_ID=@Branch_ID
         END
      ELSE IF @StatementType = 'Delete'
        BEGIN
            DELETE FROM Branches WHERE Branch_ID=@Branch_ID
        END
  END TRY
  BEGIN CATCH
       Select'Error'
  END CATCH
```

8.6.Stored Procedure : Exam_SP

```
create proc Exam_SP (@StatementType NVARCHAR(20) = '',@Exam_Code int ,
@Grade INT,@Course_Code int,@Dept_ID int)
As
BEGIN TRY
      IF @StatementType = 'Insert'
      BEGIN
            INSERT INTO Exam (Exam_Code,Grade,Course_Code ,Dept_ID)
            VALUES (@Exam_Code, @Grade,@Course_Code ,@Dept_ID)
      END
      IF @StatementType = 'Select'
       BEGIN
            SELECT * FROM Exam Where Exam_Code=@Exam_Code
       END
      IF @StatementType = 'Update'
```

```
        BEGIN
            UPDATE Exam
            SET Grade=@Grade,
                Dept_ID=@Dept_ID,
                Course_Code=@Course_Code
            Where Exam_Code=@Exam_Code
         END
      ELSE IF @StatementType = 'Delete'
        BEGIN
            DELETE FROM Exam WHERE Exam_Code=@Exam_Code
        END
  END TRY
  BEGIN CATCH
       Select'Error'
  END CATCH
```

8.7.Stored Procedure : Questions_SP

```
create proc Questions_SP(@StatementType NVARCHAR(20) = '',@Quest_Num int,
@Quest_Type varchar(20), @Questions varchar(150), @Correct_Answer
Varchar(50),@Quest_Grade int ,@Course_Code int)
As
BEGIN TRY
      IF @StatementType = 'Insert'
      BEGIN
         INSERT INTO Questions
      (Quest_Num,Quest_Type,Questions,Correct_Answer,Quest_Grade,Course_Cod)
      VALUES(@Quest_Num,@Quest_Type,@Questions,@Correct_Answer,@Quest_Grade,
      @Course_Code)
      END
      IF @StatementType = 'Select'
       BEGIN
            SELECT * FROM Questions WHERE Quest_Num=@Quest_Num
       END

      IF @StatementType = 'Update'
        BEGIN
            UPDATE Questions
            SET Quest_Type=@Quest_Type,
                Questions=@Questions,
                Correct_Answer=@Correct_Answer,
                Quest_Grade=@Quest_Grade,
                Course_Code=@Course_Code
            Where Quest_Num=@Quest_Num
            End
      ELSE IF @StatementType = 'Delete'
        BEGIN
            DELETE FROM Questions WHERE Quest_Num=@Quest_Num
        END
  END TRY
  BEGIN CATCH
       Select'Error'
```

```
  END CATCH
```

8.8.Stored Procedure : Resualt_SP

```
create proc Resualt_SP (@StatementType NVARCHAR(20) = '', @St_ID int
,@Quest_Num int,@Exam_Code int,@Exam_Model int , @Answer varchar(20))
As
BEGIN TRY
      IF @StatementType = 'Insert'
      BEGIN
            INSERT INTO Resualt
                        (St_ID,Quest_Num,Exam_Code,Exam_Model,Answer)
            VALUES (@St_ID,@Quest_Num,@Exam_Code,@Exam_Model,@Answer)
      END
      IF @StatementType = 'Select'
       BEGIN
            SELECT * FROM Resualt
       END
      IF @StatementType = 'Update'
        BEGIN
            UPDATE Resualt
            SET Exam_Code=@Exam_Code,
                Exam_Model=@Exam_Model,
                Answer=@Answer
            Where St_ID=@St_ID and Quest_Num=@Quest_Num
         END
      ELSE IF @StatementType = 'Delete'
        BEGIN
            DELETE FROM Resualt WHERE St_ID=@St_ID and Quest_Num=@Quest_Num
        END
  END TRY
  BEGIN CATCH
       Select'Error'
  END CATCH
```

8.9.Stored Procedure : Instractor

```
create proc Instractor     @StatementType NVARCHAR(20) = '',@Inst_ID int ,
@Inst_Fname varchar(50),@Inst_Lname varchar(50),@Email varchar(50),@Gender
varchar(20),@Birthdate date,@Salary int,@Manager_ID int
AS
  BEGIN TRY
      IF @StatementType = 'Insert'
      BEGIN
        INSERT INTO Instructor (Inst_ID, Inst_Fname,Inst_Lname ,Email
,Gender,Birthdate, Salary,Manager_ID )
        VALUES (@Inst_ID,@Inst_Fname ,@Inst_Lname ,@Email ,@Gender,
               @Birthdate,@Salary,@Manager_ID)
      END
      IF @StatementType = 'Select'
       BEGIN
```

```
            SELECT * FROM   Instructor WHERE Inst_ID=@Inst_ID
        END
      IF @StatementType = 'Update'
        BEGIN
            UPDATE Instructor
            SET Inst_Fname = @Inst_Fname,
                Inst_Lname = @Inst_Lname,
                Email      = @Emai,
                Gender     = @Gender,
                Birthdate = @Birthdate,
                Salary     = @Salary,
                Manager_ID = @Manager_ID
            Where Inst_ID = @Inst_ID
         END
      ELSE IF @StatementType = 'Delete'
        BEGIN
            DELETE FROM Instructor WHERE Inst_ID = @Inst_ID
        END
  END TRY
  BEGIN CATCH
       Select'Error'
  END CATCH
```

8.10.Stored Procedure : Inst_Qualification_SP

```
create proc Inst_Qualification_SP @StatementType NVARCHAR(20) = '',
         @Inst_ID INT ,@Scientific_Degree varchar(20), @Faculty varchar(50)
AS
BEGIN TRY
      IF @StatementType = 'Insert'
      BEGIN
         INSERT INTO Inst_Qualification (Inst_ID,Scientific_Degree,Faculty)
         VALUES (@Inst_ID,@Scientific_Degree,@Faculty)
      END

      IF @StatementType = 'Select'
       BEGIN
            SELECT * FROM Inst_Qualification WHERE Inst_ID=@Inst_ID
       END
      IF @StatementType = 'Update'
        BEGIN
            UPDATE Inst_Qualification
            SET Scientific_Degree =@Scientific_Degree,
                Faculty=@Faculty
            Where Inst_ID =@Inst_ID
         END
      ELSE IF @StatementType = 'Delete'
        BEGIN
            DELETE FROM Inst_Qualification WHERE Inst_ID =@Inst_ID
        END
  END TRY
  BEGIN CATCH
```

```
        Select'Error'
  END CATCH
```

8.11.Stored Procedure : Inst_Phone_SP

```
create proc Inst_Phone_SP @StatementType NVARCHAR(20) = '',
                           @Inst_ID INT, @Phone char(11)
AS
BEGIN TRY
      IF @StatementType = 'Insert'
      BEGIN
            INSERT INTO Inst_Phone (Inst_ID ,Phone)
            VALUES (@Inst_ID , @Phone)
      END
      IF @StatementType = 'Select'
       BEGIN
            SELECT * FROM Inst_Phone WHERE Inst_ID=@Inst_ID
       END
      IF @StatementType = 'Update'
        BEGIN
            UPDATE Inst_Phone
            SET Phone=@Phone
            Where Inst_ID=@Inst_ID
         END
      ELSE IF @StatementType = 'Delete'
        BEGIN
            DELETE FROM Inst_Phone WHERE Inst_ID=@Inst_ID
        END
  END TRY
  BEGIN CATCH
       Select'Error'
  END CATCH
```

8.12.Stored Procedure : Department_Sp

```
create proc Department_Sp @StatementType NVARCHAR(20) = '', @Dept_ID int,
           @Manager_ID int, @Dept_Name varchar(50),@Duration varchar(20),
           @Dept_Description varchar(150)
As
BEGIN TRY
      IF @StatementType = 'Insert'
      BEGIN
        INSERT INTO Department
             (Dept_ID,Manager_ID,Dept_Name,Duration ,Dept_Description )
        VALUES (@Dept_ID,@Manager_ID,@Dept_Name,@Duration,@Dept_Description)
      END
      IF @StatementType = 'Select'
       BEGIN
            SELECT * FROM   Department where Dept_ID=@Dept_ID
```

```
        END
      IF @StatementType = 'Update'
        BEGIN
            UPDATE Department
            SET Manager_ID=@Manager_ID,
                Dept_Name=@Dept_Name,
                Duration=@Duration,
                Dept_Description=@Dept_Description
            Where Dept_ID=@Dept_ID
         END
      ELSE IF @StatementType = 'Delete'
        BEGIN
            DELETE FROM Department WHERE Dept_ID=@Dept_ID
        END
  END TRY
  BEGIN CATCH
       Select'Error'
  END CATCH
```

8.13.Stored Procedure : Branch_Department_SP

```
create proc Branch_Department_SP @StatementType NVARCHAR(20) = '',
            @Dept_ID int, @Branch_ID int
AS
BEGIN TRY
  IF @StatementType='Insert'
  BEGIN
     INSERT into Branch_Department (Dept_ID, Branch_ID)
     VALUES (@Dept_ID, @Branch_ID )
  END
  IF @StatementType='Select'
  BEGIN
      SELECT * FROM Branch_Department WHERE Dept_ID=@Dept_ID
  END

  IF @StatementType='Update'
  BEGIN
    UPDATE Branch_Department
        SET Branch_ID = @Branch_ID
        WHERE Dept_ID=@Dept_ID
  END
  ELSE IF @StatementType='Delete'
  BEGIN
     DELETE FROM Branch_Department WHERE Dept_ID=@Dept_ID
  END
END TRY
BEGIN CATCH
  SELECT'Error'
END CATCH
```

8.14.Stored Procedure : Topics_SP

```
create proc Topics_SP @Statement_Type nvarchar(20),@Topic_ID int,@TopicName
varchar(20)
AS
 BEGIN TRY
 IF @Statement_Type='Insert'
 BEGIN
   INSERT INTO Topics (Topic_ID,TopicName)
     VALUES (@Topic_ID,@TopicName)
 END
 IF @Statement_Type='Select'
 BEGIN
    SELECT * FROM Topics WHERE Topic_ID=@Topic_ID
 END
 IF @Statement_Type='Update'
 BEGIN
   UPDATE Topics
    SET TopicName=@TopicName
      WHERE Topic_ID=@Topic_ID
 END
 ELSE IF @Statement_Type='Delete'
 BEGIN
    DELETE FROM Topics WHERE Topic_ID=@Topic_ID
 END
END TRY
BEGIN CATCH
 SELECT 'Error'
END CATCH
```

8.15.Stored Procedure : Courses_SP

```
create Proc Courses_SP @Statement_Type nvarchar(20),@Course_Code int ,
           @CourseName varchar(20),@Course_Duration int ,@Topic_ID int
AS
BEGIN TRY
 IF @Statement_Type='Insert'
 BEGIN
   INSERT INTO Courses (Course_Code,CourseName,Course_Duration,Topic_ID)
            VALUES (@Course_Code,@CourseName,@Course_Duration,@Topic_ID)
 END
 IF @Statement_Type='Select'
 BEGIN
   SELECT * FROM Courses WHERE Course_Code=@Course_Code
```

```
 END
 IF @Statement_Type='Update'
  BEGIN
    UPDATE Courses
      SET CourseName=@CourseName,
          Course_Duration=@Course_Duration,
          Topic_ID=@Topic_ID
      where Course_Code=@Course_Code
  END
  ELSE IF @Statement_Type='Delete'
  BEGIN
   DELETE FROM Courses WHERE Course_Code=@Course_Code
  END
END TRY
BEGIN CATCH
 SELECT 'Error'
END CATCH
```

8.16.Stored Procedure : Inst_Course_SP

```
create proc Inst_Course_SP @Statement_Type varchar(20),
           @Inst_ID int, @Course_Code int, @Evaluation int
AS
BEGIN TRY
  IF @Statement_Type='Insert'
  BEGIN
    INSERT INTO Inst_Course (Inst_ID,Course_Code,Evaluation)
    VALUE (@Inst_ID ,@Course_Code,@Evaluation)
  END
  IF @Statement_Type='Select'
  BEGIN
   SELECT *FROM Inst_Course WHERE Inst_ID=@Inst_ID AND
Course_Code=@Course_Code
  END
  IF @Statement_Type='Update'
  BEGIN
    UPDATE Inst_Course
        SET Evaluation=@Evaluation
        WHERE Inst_ID=@Inst_ID and Course_Code=@Course_Code
  END
  ELSE IF @Statement_Type='Delete'
  BEGIN
    DELETE FROM Inst_Course
    WHERE Inst_ID=@Inst_ID and Course_Code=@Course_Code
  END
END TRY
BEGIN CATCH
  SELECT'Error'
END CATCH
```

8.17.Stored Procedure : St_Course_SP

```
create proc St_Course_SP @Statement_Type nvarchar(20),
            @St_ID int,@Course_Code int,@Crs_Grade float
AS
BEGIN TRY
  IF @Statement_Type='Insert'
  BEGIN
   INSERT INTO St_Course (St_ID,Course_Code,Crs_Grade)
   VALUES (@St_ID,@Course_Code,@Crs_Grade)
  END
  IF @Statement_Type='Select'
  BEGIN
    SELECT * FROM St_Course WHERE St_ID=@St_ID AND Course_Code=@Course_Code
  END
  IF @Statement_Type='Update'
   BEGIN
     UPDATE St_Course
          SET Crs_Grade=@Crs_Grade
          WHERE St_ID=@St_ID AND Course_Code=@Course_Code
   END
   ELSE IF @Statement_Type='Delete'
    BEGIN
       DELETE FROM St_Course WHERE St_ID=@St_ID AND Course_Code=@Course_Code
      END
END TRY
BEGIN CATCH
 SELECT'Error'
END CATCH
```

8.18.Stored Procedure : Department_Courses_SP

```
CREATE proc Department_Courses_SP @StatementType NVARCHAR(20) = '',@Dept_ID
int,@Course_Code int
As
BEGIN TRY
      IF @StatementType = 'Insert'
      BEGIN
            INSERT INTO Department_Courses (Dept_ID,Course_Code)
            VALUES (@Dept_ID,@Course_Code)
      END
      IF @StatementType = 'Select'
       BEGIN
            SELECT * FROM Department_Courses
```

```
            where Dept_ID=@Dept_ID and Course_Code=@Course_Code
        END
       ELSE IF @StatementType = 'Delete'
         BEGIN
            DELETE FROM Department_Courses
            WHERE Dept_ID=@Dept_ID and Course_Code=@Course_Code
         END
   END TRY
   BEGIN CATCH
        Select'Error'
   END CATCH
```

8.19.Stored Procedure : Choices SP

```
GO
Create proc Choices_SP  @Statement_Type varchar(20),
                        @Quest_Num int,@Choices varchar(50)
AS
BEGIN TRY
  if @Statement_Type='Insert'
  BEGIN
     INSERT INTO Choices
                          (Quest_Num ,Choices)
            VALUES          (@Quest_Num ,@Choices)
  END
  IF @Statement_Type='Select'
   BEGIN
     SELECT * FROM Choices
        where    Quest_Num = @Quest_Num AND Choices = @Choices
   END
   Else if @Statement_Type='Delete'
   begin
     DELETE FROM Choices
        where Quest_Num=@Quest_Num AND Choices=@Choices
   end
END TRY
BEGIN CATCH
  SELECT'Error'
END CATCH
```

9.Stored Procedures for Reports

9.1. Procedures for Report 1 : getstudinformation

```
create proc getstudinformation @Dept_id int
as
select * from Student
where Dept_ID in (@Dept_id)
```

9.2. Procedures for Report 2 : studentgrade

```
create proc studentgrade @st_id int
as
begin try
      select s.St_ID,St_Fname,CourseName,Crs_Grade from Student s,St_Course
      sc,Courses c where s.St_ID=sc.St_ID and sc.Course_Code=c.Course_Code
      and s.St_ID = @st_id
end try
begin catch
      select 'no student exist with this id'
end catch
```

9.3. Procedures for Report 3 : ins_courses

```
create proc ins_courses @ins_id int
as
begin try
      select i.Inst_ID ,c.CourseName,count(st_id) as StudentsNum
      from Courses c,Instructor i,Inst_Course ic,St_Course sc
      where c.Course_Code=sc.Course_Code and ic.Course_Code=c.Course_Code
      and i.Inst_ID=ic.Inst_ID and i.Inst_ID in (@ins_id) group by
      c.CourseName , i.Inst_ID
end try
begin catch
      select 'no Instructor exist with this id'
end catch
```

9.4. Procedures for Report 4 : CourseTopic

```
create proc CourseTopic @cr_code int
as
begin try
      select CourseName,TopicName from Courses c,Topics t where
t.Topic_ID=c.Topic_ID and c.Course_Code in (@cr_code)
end try
begin catch
      select 'no course exist with this code'
end catch
```

9.5. Procedures for Report 5 : exam_questions

```
create proc exam_questions @exam_code int
as
begin try
      select q.Questions,q.Quest_Num from Questions q,Resualt r
      where q.Quest_Num=r.Quest_Num and Exam_Code=@exam_code
end try
begin catch
      select 'No Exam Exist With This Code'
end catch
```

9.6. Procedures for Report 5 : choicesOfExam

```
create proc choicesOfExam
as
select Q_Number,Choices from Choices
```

9.7. Procedures for Report 6 :

```
create proc stud_ans @exam_code int,@st_id int
as
begin try
      select Questions,Answer from Questions q,Resualt r
      where q.Quest_Num=r.Quest_Num and Exam_Code=@exam_code and
St_ID=@st_id
      end try
begin catch
      select 'please enter valid data'
end catch
```

10.Exam Stored Procedure

10.1. Exam Generation Stored Procedure

```
create proc GenerateExam @st_id int ,@crs_code int,@dept_id int
as
begin  --check that enter valid information
       if exists(select s.St_ID ,sc.Course_Code ,s.Dept_ID from Student s,st_Course sc
                            where s.St_ID=sc.St_ID and s.St_ID=@st_id
                            and Course_Code=@crs_code and Dept_ID=@dept_id)
              begin
                     declare @ex_code int=(select max(Exam_Code)+1 from Exam) --auto
increment to Exam code
                     begin try
                            insert into Exam (Exam_Code,Course_Code,Dept_ID)
                            values (@ex_code,@crs_code,@dept_id)
                     end try
                     begin catch
                            select 'Error Exam'
                     end catch

                     begin try
                            insert into Resualt (Quest_Num, St_ID , Exam_Code)
                            SELECT TOP(5) Quest_Num ,@st_id ,@ex_code
                            FROM Questions
                            WHERE Course_Code=@crs_code and Quest_Type='MCQ'
                            ORDER BY NEWID()

                            insert into Resualt (Quest_Num, St_ID , Exam_Code)
                            SELECT TOP(5) Quest_Num ,@st_id ,@ex_code
                            FROM Questions
                            WHERE Course_Code=@crs_code and Quest_Type='T/F'
                            ORDER BY NEWID()
                     end try
                     begin catch
                            select 'Error selecting questions'
                     end catch
              end

       else
              print 'Please Enter Valid information'
end
```

10.2. Exam Answers Stored Procedure

```
create proc Exam_Answers @st_id int ,@quest_num int, @ex_code int , @st_ans varchar(30)
as
begin
       --check that this student already examined
       if exists (select St_ID,Quest_Num,Exam_Code from Resualt where St_ID=@st_id and
Exam_Code=@ex_code)
              begin
```

```
                    if @st_ans in (select Choices from Choices where
Quest_Num=@quest_num)
                            begin
                                    begin try
                                    update Resualt
                                    set Answer= @st_ans
                                    where St_ID=@st_id and Exam_Code=@ex_code and
Quest_Num=@quest_num
                                    end try

                                    begin catch
                                            select 'invalid insert answer'
                                    end catch
                            end
                    else
                            print('invalid answer')
                    end
        else
                print('this student did not examin')
end
```

10.3. Exam Correction Stored Procedure

```
Create proc ExamCorrrection @st_id int ,@crs_code int, @dept_id int, @ex_code int
as
begin
        begin try
                if exists (select St_ID,Exam_Code from Resualt where St_ID=@st_id)
                        begin
                                declare @total int=0
                                                SELECT @total= SUM (q.Quest_Grade)
                                                FROM Resualt r , Questions q
                                                WHERE r.Quest_Num=q.Quest_Num and
r.Exam_Code=@ex_code
                                                and r.St_ID = @st_id and Answer=Correct_Answer
                                UPDATE Exam
                                SET Grade = @total
                                WHERE Exam_Code=@ex_code and  Course_Code=@crs_code and
Dept_ID=@dept_id

                                select Grade from Exam where Exam_Code=@ex_code
                        end
                else
                        print 'invalid information'

        end try
        begin catch
                select 'Error Calc'
        end catch

end
```

11.Report (By SSRS)

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

Dept id Power BI Developer View Report

1 of 1 75% Find | Next

Students Informations of Power BI Developer Department

St ID	St Fname	St Lname	Gender	Street	City	Governorate	Birthdate	Hiring Date	Certificate	Freelancing	ITI Grad Date	Position	Company Name	Email	User Name	Password	Leader ID	Dep
47	Rana	Hazel	F	286 White Nobel Parkway	Madinat-Nasr	Cairo	6/6/2000 12:00:00 AM	2/1/2023 12:00:00 AM	Cutomer	The Red Gate Guide to SQL Server Team-based Development	12/15/2022 12:00:00 AM	BI Developer	We	Rana.1233266 7@compit.aun.edu.eg	Rana	12345	47	
48	Rewaa	Shaban	F	14 Nobel Freeway	Madinat-Nasr	Cairo	6/3/2000 12:00:00 AM	2/1/2023 12:00:00 AM		Inside the SQL Server Query Optimizer	12/15/2022 12:00:00 AM	BI Developer	We	Rewaa.183326 42@compit.aun.edu.eg	Rewaa	12345	47	
49	Reem	Mohamed	F	430 Green Clarendon Street	Madinat-Nasr	Cairo	9/1/2000 12:00:00 AM	2/1/2023 12:00:00 AM	Corporate Care		12/15/2022 12:00:00 AM	BI Developer	Vodafon	Reem.183326 22@compit.aun.edu.eg	Reem	12345	47	
50	Reem	Osama	F	677 White Oak Boulevard	Madinat-Nasr	Cairo	8/28/2000 12:00:00 AM	2/1/2023 12:00:00 AM	Marketing	SQL Server Hardware	12/15/2022 12:00:00 AM	BI Developer	Vodafon	Reem.183208 72@compit.aun.edu.eg	Reem	12345	47	
51	Renad	Asharf	F	566 Rocky Nobel Road	Madinat-Nasr	Cairo	3/28/2000 12:00:00 AM	2/1/2023 12:00:00 AM	Marketing		12/15/2022 12:00:00 AM	BI Developer	Vodafon	Renad.123326 66@compit.aun.edu.eg	Renad	12345	47	
52	Reham	Medhat	F	220 Rocky Second St.	Madinat-Nasr	Cairo	11/25/2000 12:00:00 AM	1/1/2023 12:00:00 AM			12/15/2022 12:00:00 AM	BI Developer	Nestlie	Reham.18332 652@compit.aun.edu.eg	Reham	12345	47	
53	Zaid	Hashem	M	222 Hague Boulevard	Madinat-Nasr	Cairo	6/1/1999 12:00:00 AM	2/1/2023 12:00:00 AM	Sales		12/15/2022 12:00:00 AM	Power BI Developer	Brannist	ziad.abdalateef@compit.aun.edu.eg	Zaid	12345	47	
54	Sandy	Fam	F	57 Second Parkway	Madinat-Nasr	Cairo	8/28/1999 12:00:00 AM	2/1/2023 12:00:00 AM	Cutomer		12/15/2022 12:00:00 AM	Power BI Developer	Technon	Sandy.183236 66@compit.aun.edu.eg	Sandy	12345	47	
55	Salma	Asharf	F	47 Clarendon	Madinat-Nasr	Cairo	12/16/2000	2/1/2023		SQL Server	12/15/2022	Power BI	Wala	salma.183326	Salma	12345	47	

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

ST_ID 1

1 of 1 100% Find | Next

Grades of the Student ID : 1

St ID	St Fname	Course Name	Crs Grade
1	Ibraheem	Python	90 %

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

Ins ID 1

1 of 1 100% Find | Next

The Courses that the Instructor of ID: 1 Teaches

Inst ID	Course Name	Students Num
1	Python	45

- Report that takes course ID and returns its topics

Topic ID 1

1 of 1 100% Find | Next

Courses of Programming Topic

Topic Name	Course Name
Programming	Python
Programming	C#
Programming	Java
Programming	C programming

- Report that takes exam code and returns the Questions in it and choices.

Exam Code 2

1 of 2 ? 75% Find | Next

Questions of Exam Code: 2

Which type of Programming does Python support?

a) object-oriented
b) structure programing
c) functional programing
d) all of the mentioned

Is Python case sensitive when dealing with identifiers?

a) no
b) yes
c) machine dependent
d) none of the mentioned

Which of the following is the correct extension of the Python file?

a) .python
b) .pl
c) .py
d) .p

All keywords in Python are in _____

a) Capitalized
b) lower case
c) UPPER CASE
d) None of the mentioned

Which keyword is used for function in Python language?

a) Function
b) def
c) Fun

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

exam code 2 st id 48

1 of 1 100% Find | Next

The Answers of Student ID: 48 For Exam Code:2

Questions	Answer
Which type of Programming does Python support?	a) getarg
Is Python case sensitive when dealing with identifiers?	c) .py
Which of the following is the correct extension of the Python file?	d) set()
All keywords in Python are in _____	d) Define
Which keyword is used for function in Python language?	c) { }
Which of the following character is used to give single-line comments in Python?	b) .pl
Which of the following character is used to give single-line comments in Python?	c) +
Which one of the following is not a keyword in Python language?	b) True
Which module in the python standard library parses options received from the command line?	a) True
What arithmetic operators cannot be used with strings in Python?	d) nonlocal

12.Dashboard

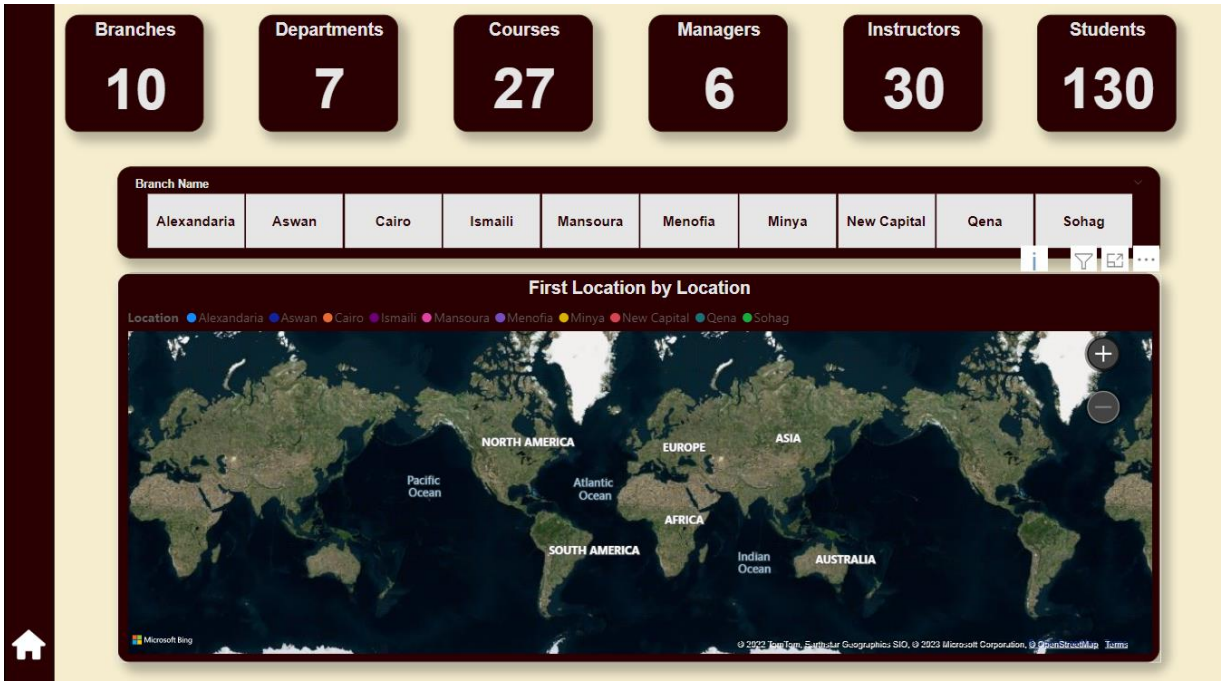
12.1 Home Page:

There are 5 Butons as Navigator that Dividede to

- **ITI Page**
- **Department Page**
- **Courses Page**
- **Students Page**

And This page have three Links Related ITI Platforms

12.2 ITI Pages:

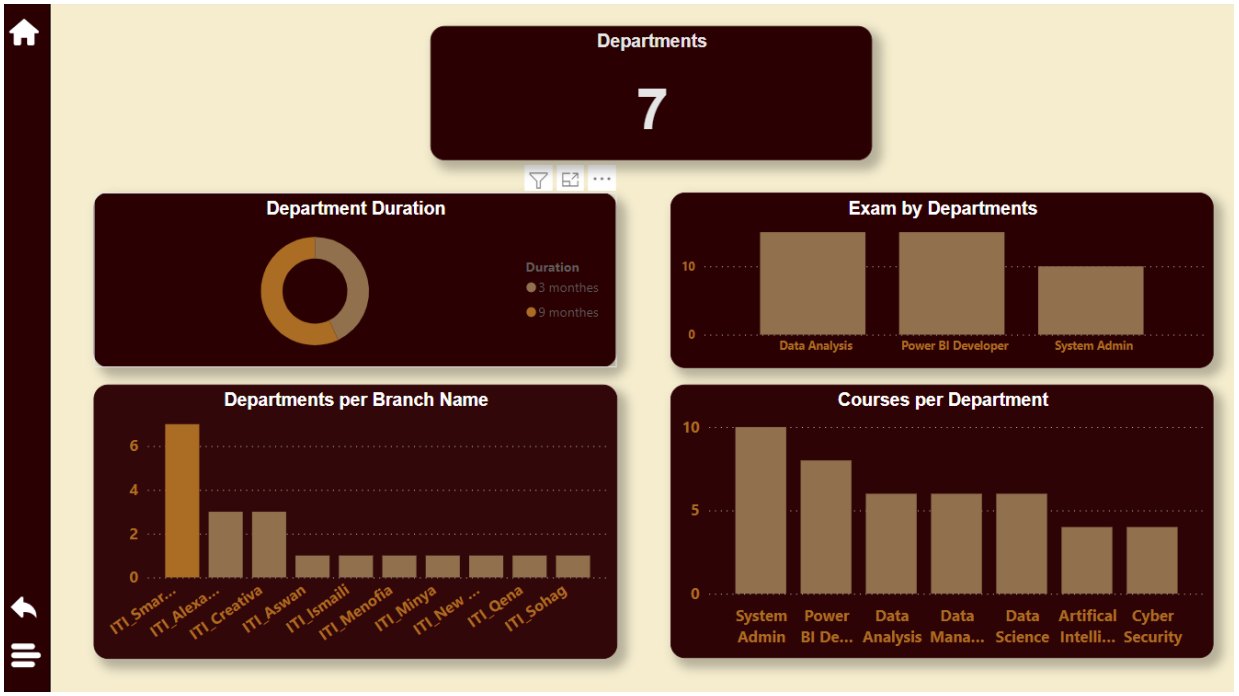


This page Have Overview in ITI Branches ,Student,Departments and Instructor.

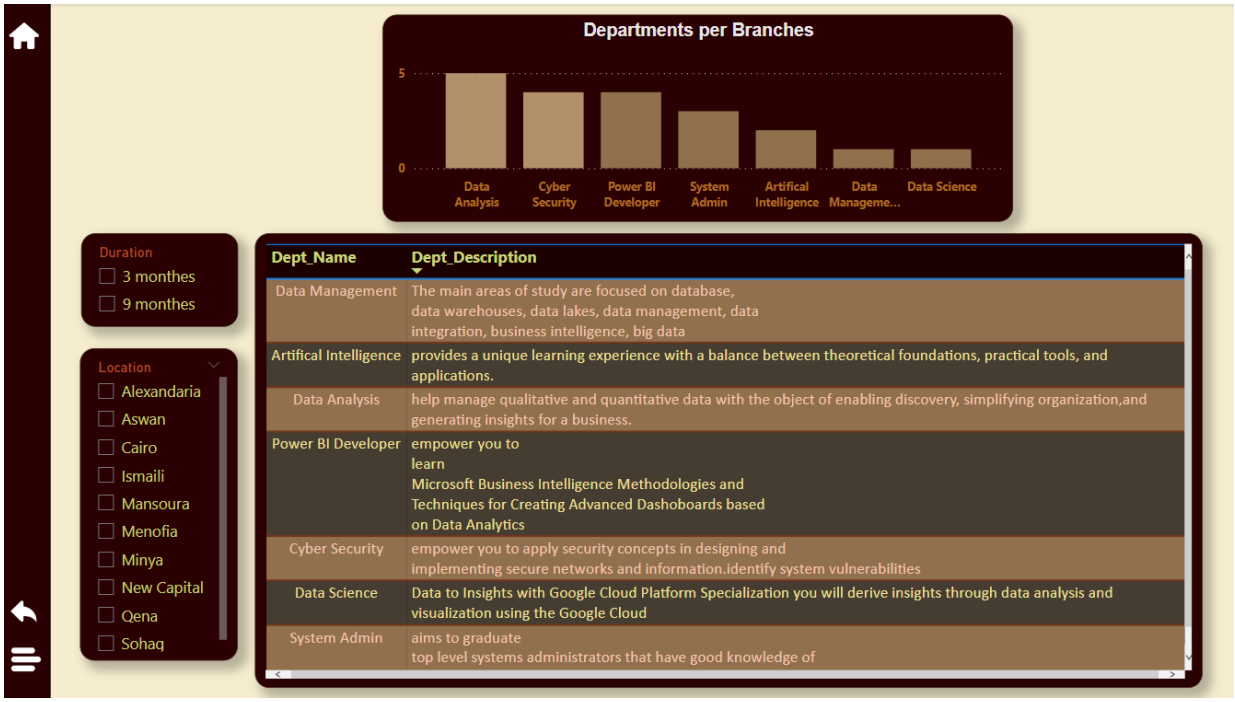
12.3 Department Pages:

This page contains of two page

Department overview:



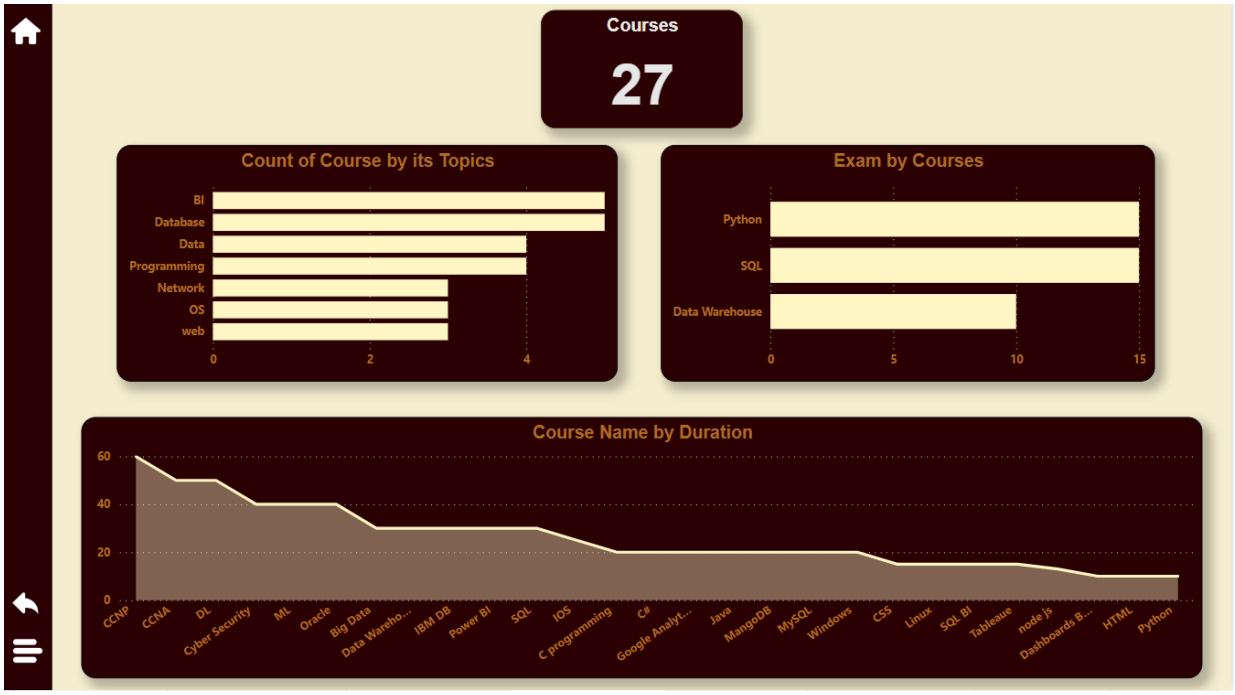
Department Details:



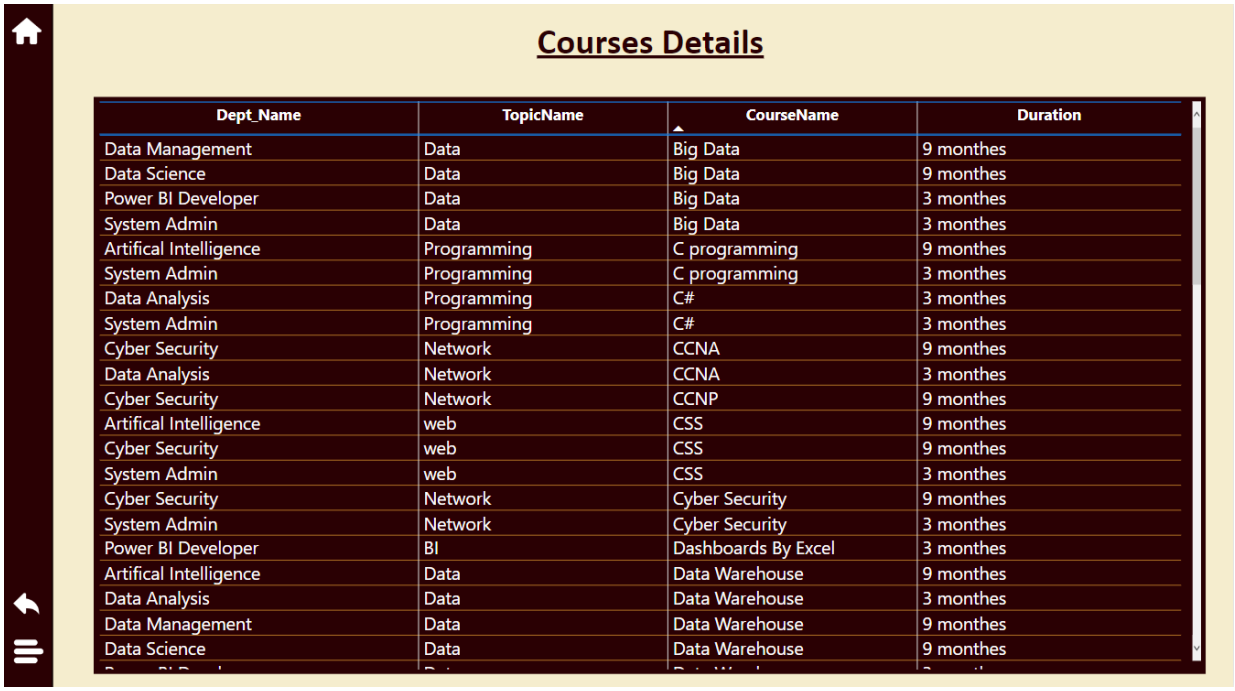
12.4 Courses Pages:

This page contains of Four pages

Courses overview:



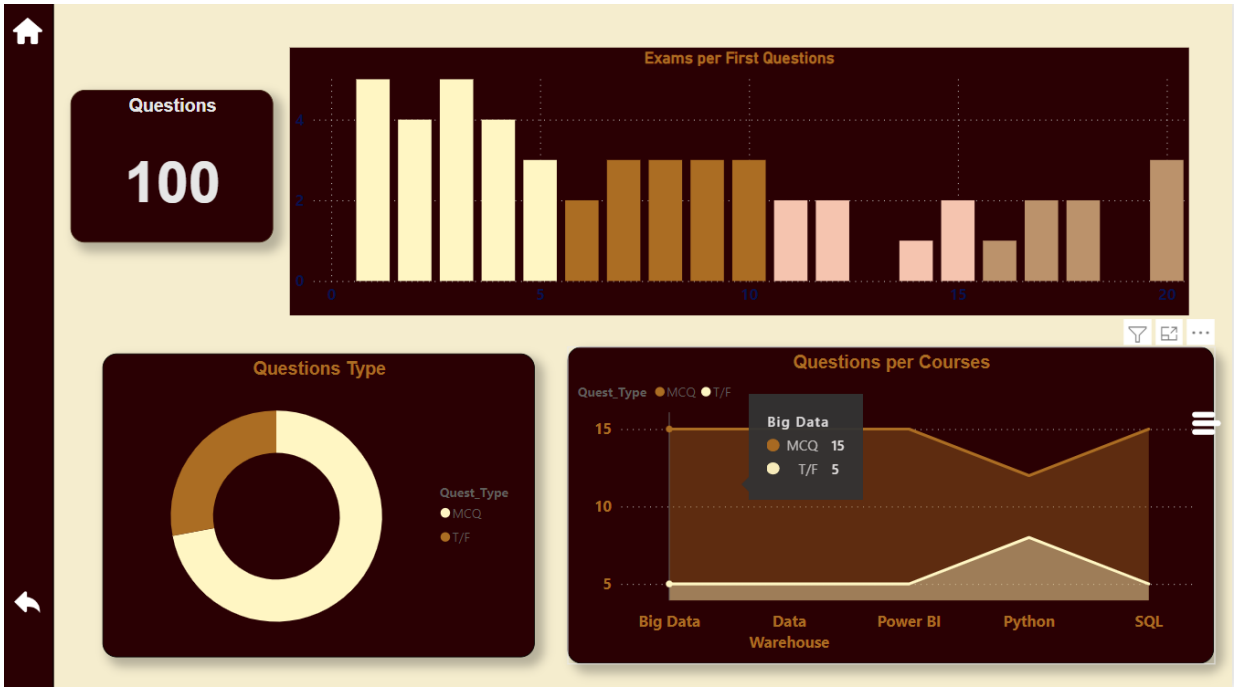
Courses Details:



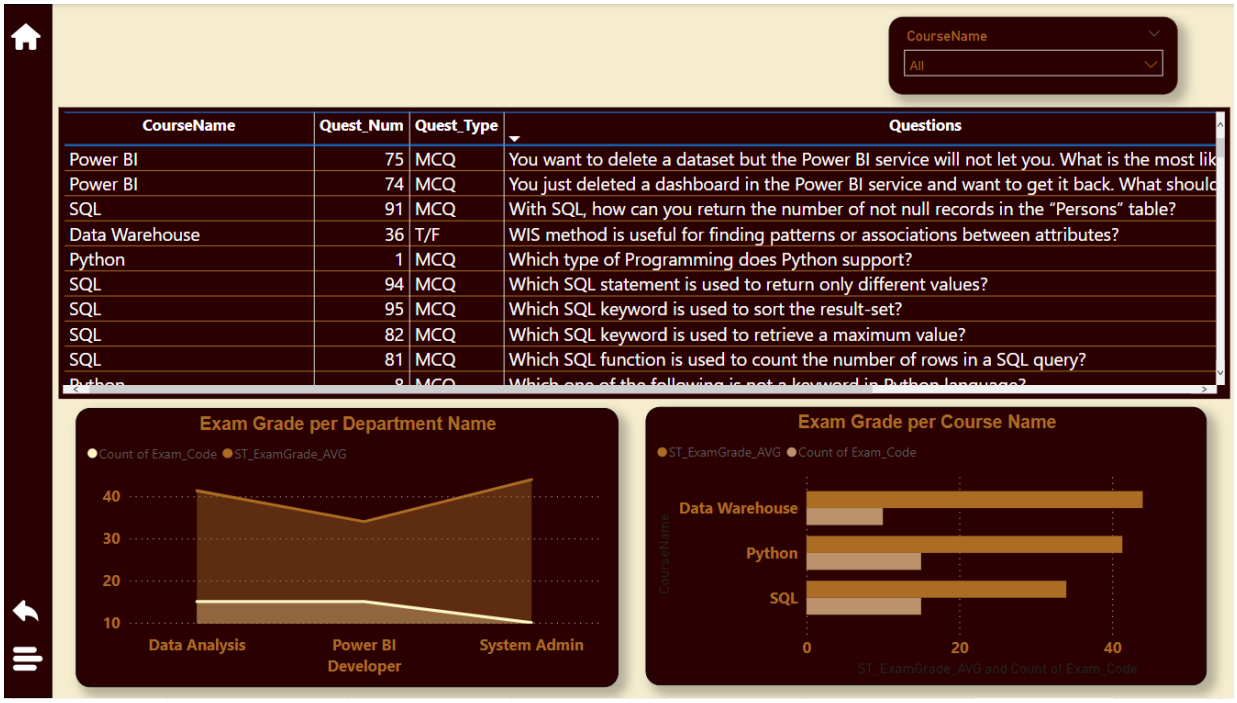
Courses Details

Dept_Name	TopicName	CourseName	Duration
Data Management	Data	Big Data	9 monthes
Data Science	Data	Big Data	9 monthes
Power BI Developer	Data	Big Data	3 monthes
System Admin	Data	Big Data	3 monthes
Artifical Intelligence	Programming	C programming	9 monthes
System Admin	Programming	C programming	3 monthes
Data Analysis	Programming	C#	3 monthes
System Admin	Programming	C#	3 monthes
Cyber Security	Network	CCNA	9 monthes
Data Analysis	Network	CCNA	3 monthes
Cyber Security	Network	CCNP	9 monthes
Artifical Intelligence	web	CSS	9 monthes
Cyber Security	web	CSS	9 monthes
System Admin	web	CSS	3 monthes
Cyber Security	Network	Cyber Security	9 monthes
System Admin	Network	Cyber Security	3 monthes
Power BI Developer	BI	Dashboards By Excel	3 monthes
Artifical Intelligence	Data	Data Warehouse	9 monthes
Data Analysis	Data	Data Warehouse	3 monthes
Data Management	Data	Data Warehouse	9 monthes
Data Science	Data	Data Warehouse	9 monthes

Courses Questions:

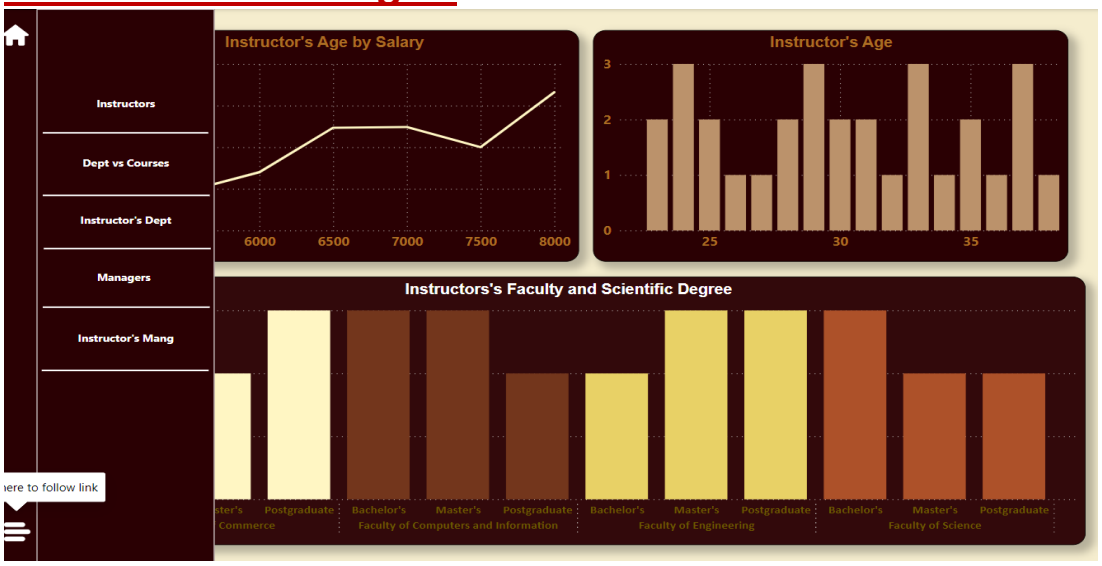


Courses Exam:

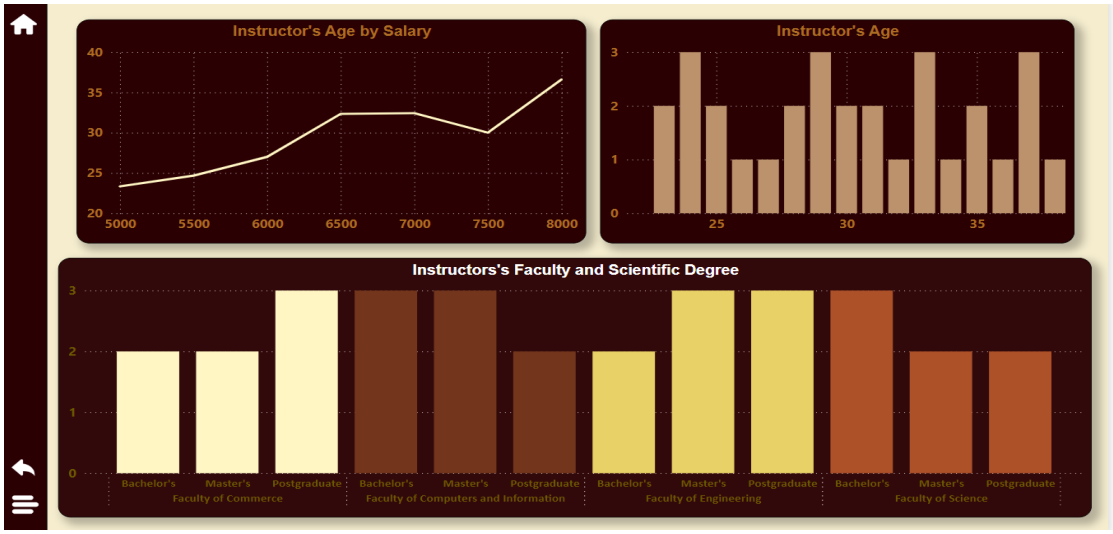


CourseName	Quest_Num	Quest_Type	Questions
Power BI	75	MCQ	You want to delete a dataset but the Power BI service will not let you. What is the most lik
Power BI	74	MCQ	You just deleted a dashboard in the Power BI service and want to get it back. What should
SQL	91	MCQ	With SQL, how can you return the number of not null records in the "Persons" table?
Data Warehouse	36	T/F	WIS method is useful for finding patterns or associations between attributes?
Python	1	MCQ	Which type of Programming does Python support?
SQL	94	MCQ	Which SQL statement is used to return only different values?
SQL	95	MCQ	Which SQL keyword is used to sort the result-set?
SQL	82	MCQ	Which SQL keyword is used to retrieve a maximum value?
SQL	81	MCQ	Which SQL function is used to count the number of rows in a SQL query?

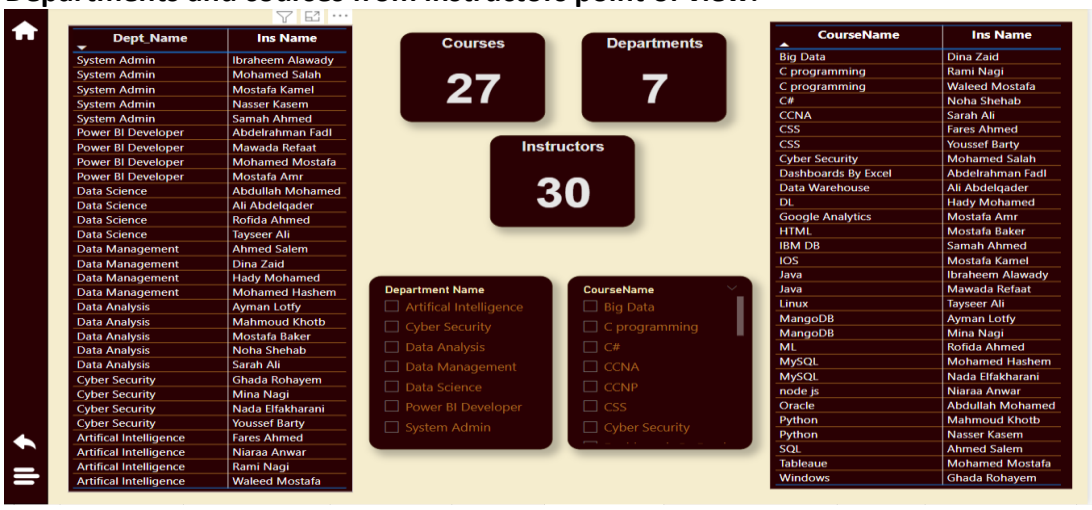
12.5 Instructors Pages:



This page contains of Four pages
Instrctors Overview:



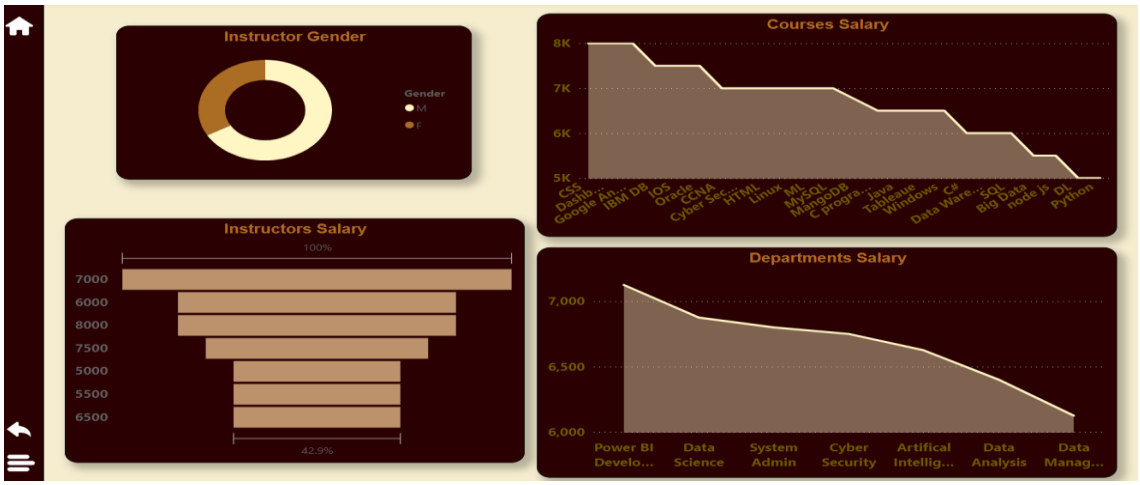
Departments and courses from instructors point of view:



Dept_Name	Ins Name
System Admin	Ibraheem Alawady
System Admin	Mohamed Salah
System Admin	Mostafa Kamel
System Admin	Nasser Kasem
System Admin	Samah Ahmed
Power BI Developer	Abdelrahman Fadl
Power BI Developer	Mawada Refaat
Power BI Developer	Mohamed Mostafa
Power BI Developer	Mostafa Amr
Data Science	Abdullah Mohamed
Data Science	Ali Abdelqader
Data Science	Rofida Ahmed
Data Science	Tayseer Ali
Data Management	Ahmed Salem
Data Management	Dina Zaid
Data Management	Hady Mohamed
Data Management	Mohamed Hashem
Data Analysis	Ayman Lotfy
Data Analysis	Mahmoud Khotb
Data Analysis	Mostafa Baker
Data Analysis	Noha Shehab
Data Analysis	Sarah Ali
Cyber Security	Ghada Rohayem
Cyber Security	Mina Nagi
Cyber Security	Nada Elfakharani
Cyber Security	Youssef Barty
Artifical Intelligence	Fares Ahmed
Artifical Intelligence	Niaraa Anwar
Artifical Intelligence	Rami Nagi
Artifical Intelligence	Waleed Mostafa

CourseName	Ins Name
Big Data	Dina Zaid
C programming	Rami Nagi
C programming	Waleed Mostafa
C#	Noha Shehab
CCNA	Sarah Ali
CSS	Fares Ahmed
CSS	Youssef Barty
Cyber Security	Mohamed Salah
Dashboards By Excel	Abdelrahman Fadl
Data Warehouse	Ali Abdelqader
DL	Hady Mohamed
Google Analytics	Mostafa Amr
HTML	Mostafa Baker
IBM DB	Samah Ahmed
IOS	Mostafa Kamel
Java	Ibraheem Alawady
Java	Mawada Refaat
Linux	Tayseer Ali
MangoDB	Ayman Lotfy
MangoDB	Mina Nagi
ML	Rofida Ahmed
MySQL	Mohamed Hashem
MySQL	Nada Elfakharani
node js	Niaraa Anwar
Oracle	Abdullah Mohamed
Python	Mahmoud Khotb
Python	Nasser Kasem
SQL	Ahmed Salem
Tableaue	Mohamed Mostafa
Windows	Ghada Rohayem

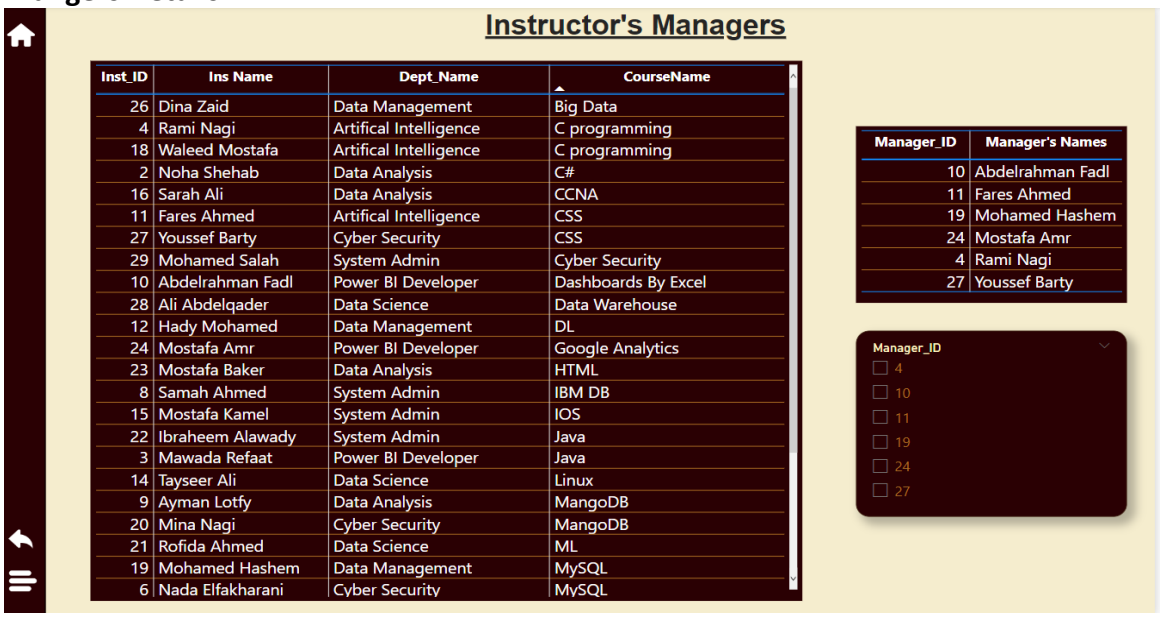
Instructors Department's



Mangers Overview

Ins Name	Sum of Salary	Age
Youssef Barty	8000	35
Rami Nagi	7500	34
Mostafa Amr	8000	37
Mohamed Hashem	8000	38
Fares Ahmed	8000	37
Abdelrahman Fadl	8000	36

Mangers Details:



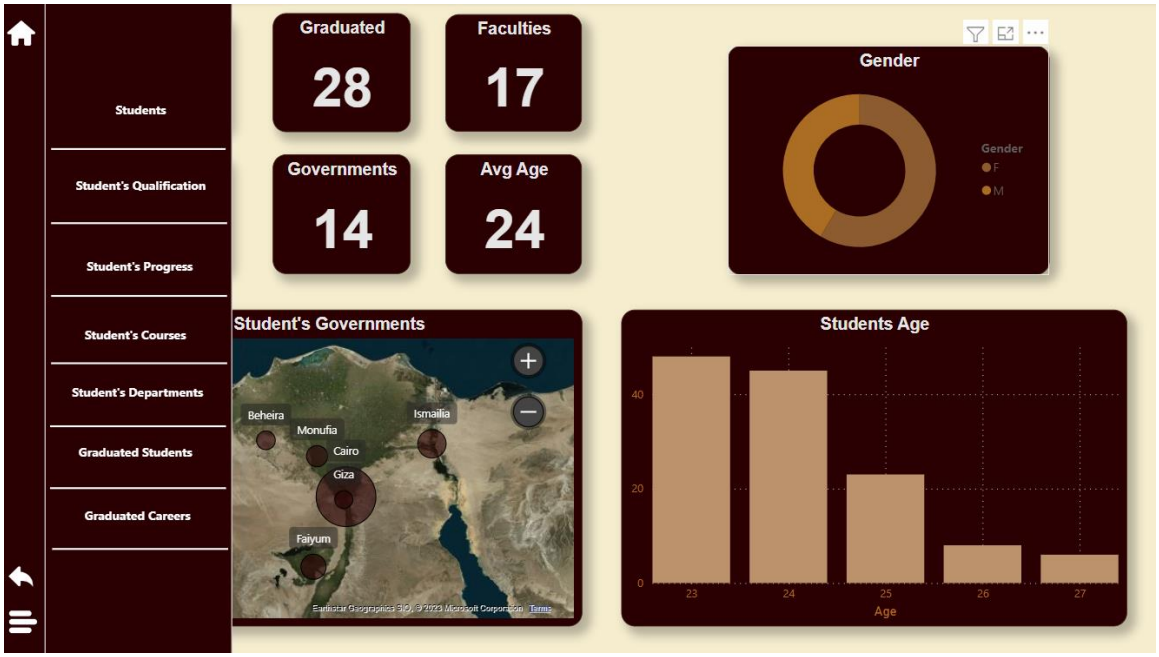
Instructor's Managers

Inst_ID	Ins Name	Dept_Name	CourseName
26	Dina Zaid	Data Management	Big Data
4	Rami Nagi	Artifical Intelligence	C programming
18	Waleed Mostafa	Artifical Intelligence	C programming
2	Noha Shehab	Data Analysis	C#
16	Sarah Ali	Data Analysis	CCNA
11	Fares Ahmed	Artifical Intelligence	CSS
27	Youssef Barty	Cyber Security	CSS
29	Mohamed Salah	System Admin	Cyber Security
10	Abdelrahman Fadl	Power BI Developer	Dashboards By Excel
28	Ali Abdelqader	Data Science	Data Warehouse
12	Hady Mohamed	Data Management	DL
24	Mostafa Amr	Power BI Developer	Google Analytics
23	Mostafa Baker	Data Analysis	HTML
8	Samah Ahmed	System Admin	IBM DB
15	Mostafa Kamel	System Admin	IOS
22	Ibraheem Alawady	System Admin	Java
3	Mawada Refaat	Power BI Developer	Java
14	Tayseer Ali	Data Science	Linux
9	Ayman Lotfy	Data Analysis	MangoDB
20	Mina Nagi	Cyber Security	MangoDB
21	Rofida Ahmed	Data Science	ML
19	Mohamed Hashem	Data Management	MySQL
6	Nada Elfakharani	Cyber Security	MySQL

Manager_ID	Manager's Names
10	Abdelrahman Fadl
11	Fares Ahmed
19	Mohamed Hashem
24	Mostafa Amr
4	Rami Nagi
27	Youssef Barty

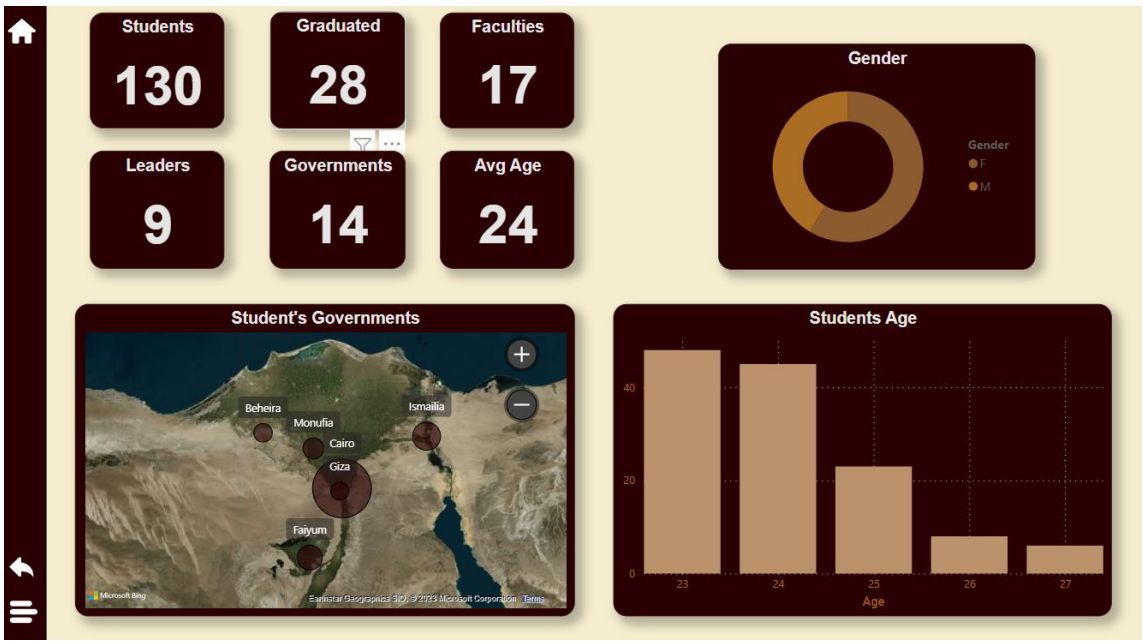
Manager_ID
4
10
11
19
24
27

12.6 Students Pages:

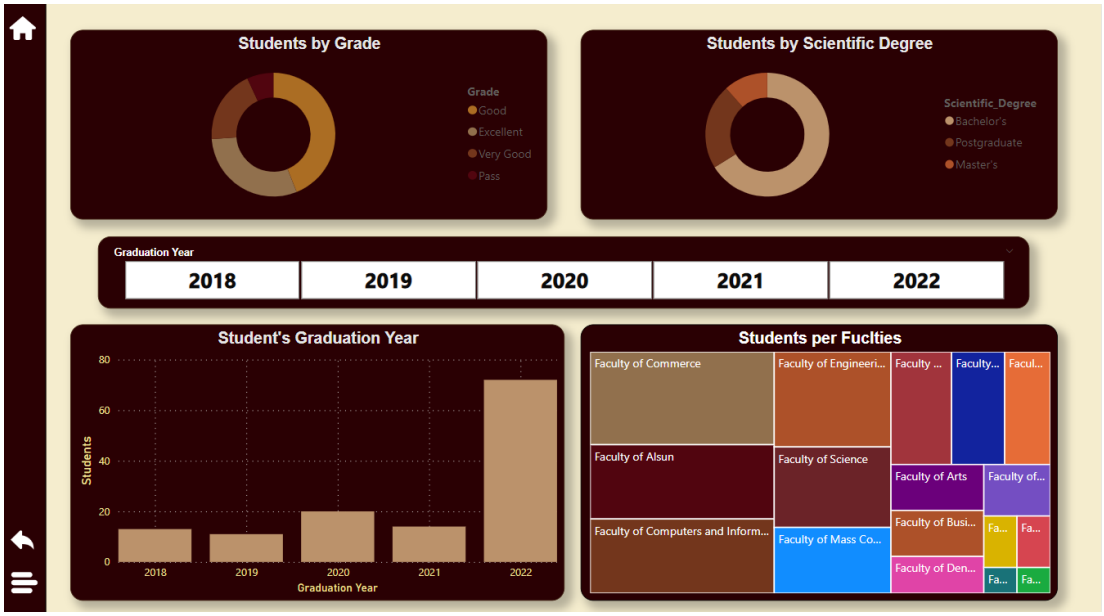


This page consists of Seven Pages:

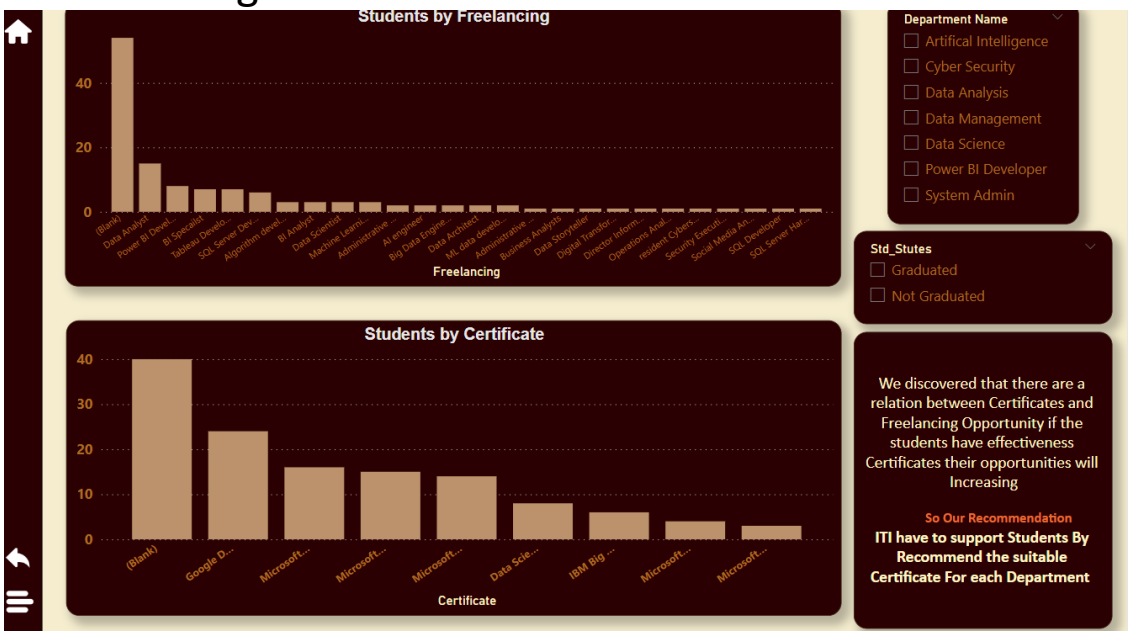
Students:



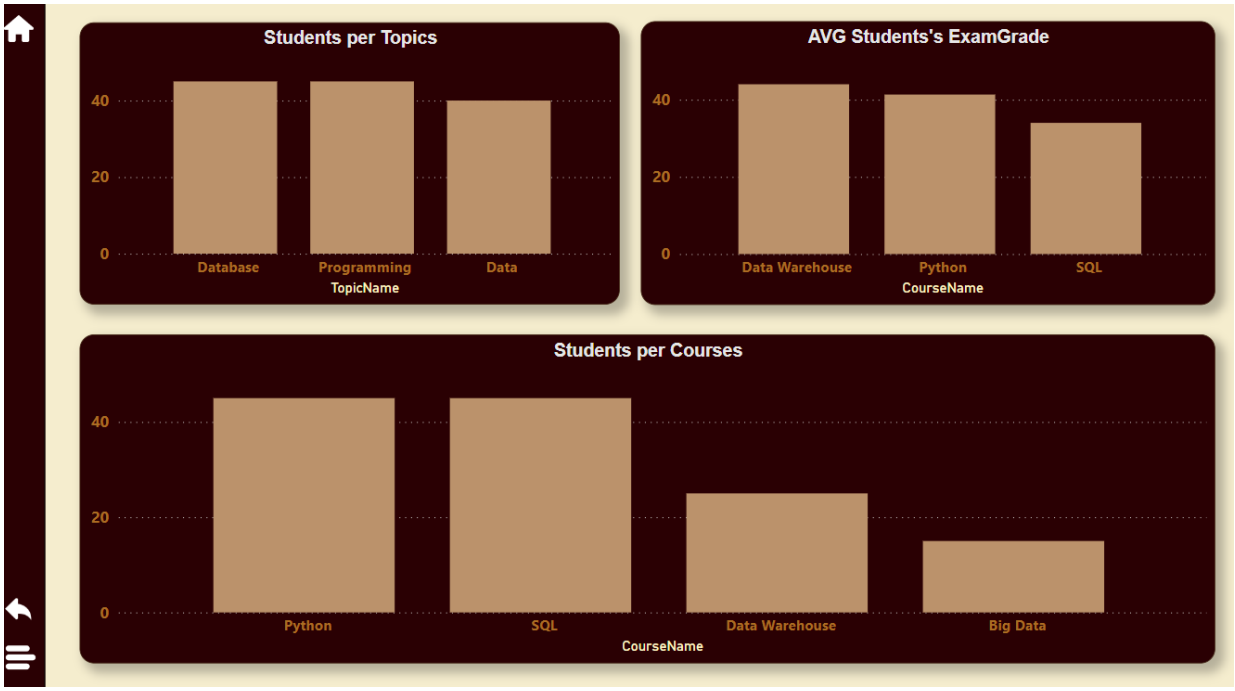
Student Qualification:



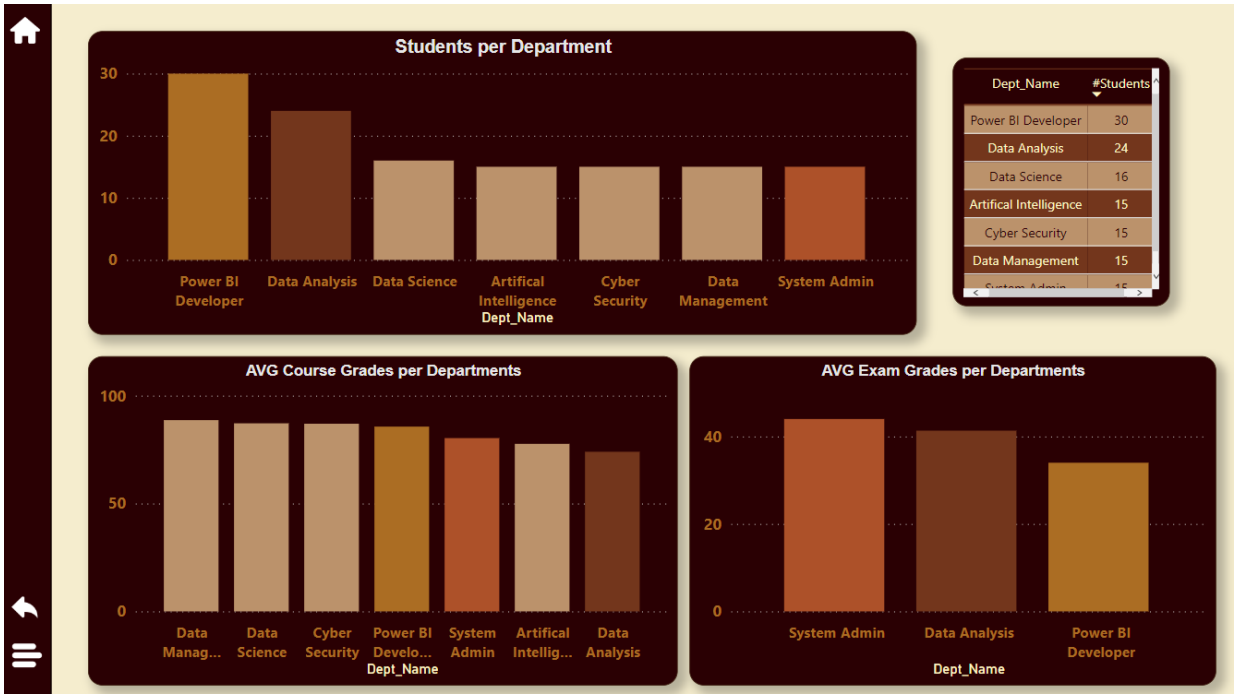
Student Progress:



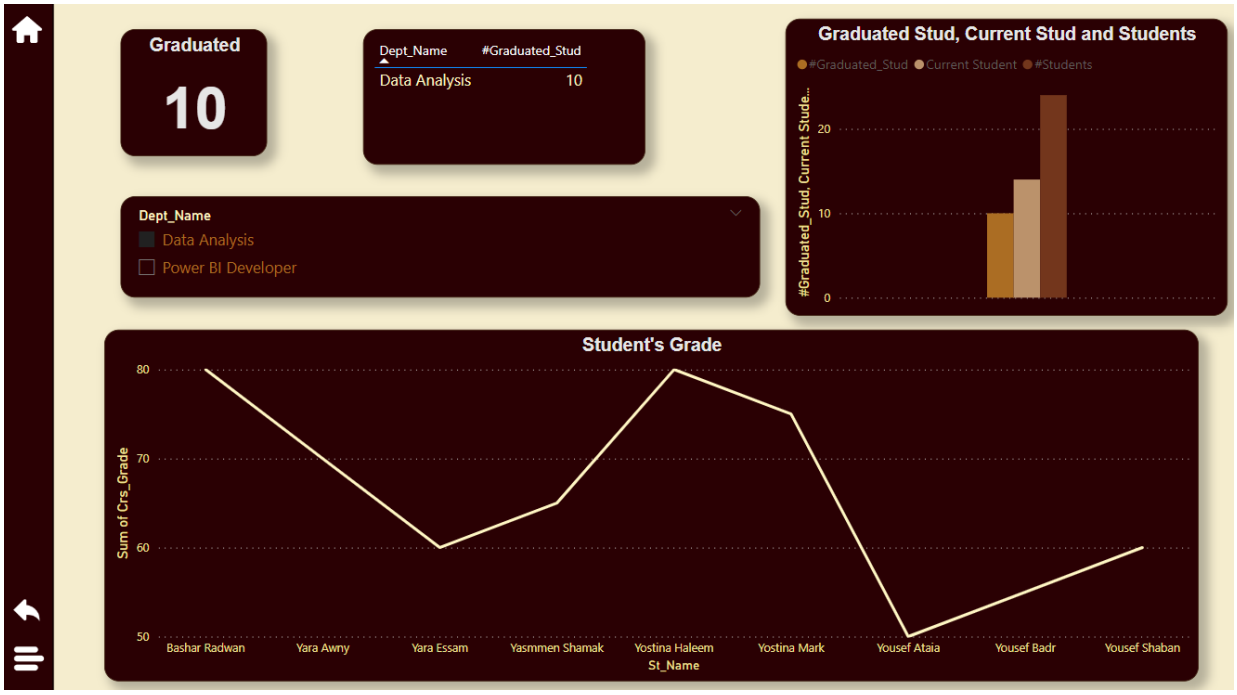
Student And Courses:



Student And Departments:



Graduated Student



Graduate Courses