



Power BI Developer

Mansoura Branch

ITI Examination System

Graduation Project

Prepared by:

Eman Ashraf Barakat Shatat

Mohamed Ramadan Nouredin

Akram Mohamed Marzouk

Muhammad Ahmed Almursii

Elbialy Mohamed Elbialy Abdelazeez

Technical Supervisors:

Eng. Rami Abo Nagy

Eng. Nadia Saleh

Contents

- ❖ Abstract
- ❖ Introduction
- ❖ Scenario
- ❖ Entities
- ❖ Entity Relationship Diagram (ERD)
- ❖ Mapping
- ❖ SQL Server Integration Service (SSIS)
- ❖ Stored Procedures
- ❖ SQL Server Reporting Service (SSRS)
- ❖ Power BI Dashboards

Abstract:

The ITI Examination System is a culmination of a graduation project meticulously crafted to streamline the examination process for students enrolled in a diverse range of courses at the esteemed Information Technology Institute (ITI). Within this system, an array of vital entities including students, courses, tracks, and instructors collaboratively harmonize to deliver a seamless and effective experience for each student. The fundamental objective of this project is to devise a design for the ITI Examination System that intricately caters to the unique needs of students and the precise requisites of their respective courses.

This system empowers students to engage in examinations pertinent to the courses they are enrolled in, while affording instructors the authority to craft and oversee these examinations. A pivotal feature of the ITI Examination System lies in its automated capacity to generate bespoke exam questions tailored for each course. Grounded in the specific learning outcomes and objectives of every course, the system produces distinct exam questions tailored to individual students.

In summation, the ITI Examination System emerges as a holistic and sophisticated solution engineered to navigate the intricacies of examination management within a dynamic educational landscape. By endowing students with a dependable and efficient platform for examination endeavors, this system stands poised to revolutionize the examination process at ITI.

Introduction:

The ITI Examination System stands as a culmination of a rigorous graduation project, traversing through multiple pivotal phases from meticulous analysis and design to the ultimate implementation. This project's core objective was to establish a robust framework for creating examinations and formulating questions tailored for students. It encompassed a spectrum of critical stages, encompassing ERD diagram development, database design and deployment, implementation of stored procedures, SSRS reporting, and the integration of Power BI dashboards.

The genesis of this endeavor initiated with a comprehensive analysis phase, where the prerequisites and requisites of the system were meticulously discerned. This phase also encompassed the delineation of entities and the intricate relationships interconnecting them.

Subsequently, the database design phase took center stage, characterized by the inception of an ERD diagram. This graphical representation provided an invaluable visual blueprint of the system's architecture. It not only aided in the identification of imperative database tables but also facilitated the seamless transition from ERD to an operational database schema. This schema encompassed pivotal tables catering to students, courses, tracks, instructors, and a repository of exam questions, among other crucial components.

The implementation phase witnessed the actualization of the database using the SQL Server Management Studio (SSMS). It also involved the creation of indispensable stored procedures, instrumental in the manipulation and retrieval of data. Furthermore, this phase encompassed the development of SSRS reports, empowering both instructors and students to peruse and dissect exam results and other pertinent data.

The crowning achievement of the project culminated in the creation of Power BI dashboards, offering an intuitive and visually compelling representation of the data repository within the system. These dashboards not only facilitated rapid and facile data visualization but also proffered invaluable insights for potential enhancements and refinements.

In summation, the ITI Examination System, traversing a comprehensive journey of analysis, design, and implementation, emerges as a holistic solution poised to revolutionize the examination process within an educational milieu.

Scenario

The ITI Examination System is a comprehensive platform overseeing various aspects of ITI's educational programs. This system is deployed across multiple branches, each distinguished by a unique ID and a distinct name. Within every branch, a dedicated manager takes charge, is identified by a unique ID, and provides essential details including their name, gender, salary, age, hiring date.

Tracks are an integral part of the educational framework, available in one or more branches. Each track possesses a unique ID, alongside a name. Courses, the building blocks of education, are hosted within tracks, with the flexibility to exist in multiple tracks simultaneously. Each course is characterized by its own unique ID, name, and duration.

Instructors play a pivotal role in the educational process, responsible for imparting knowledge. They are associated with one or more courses, and a course may have multiple instructors. Instructors have their own unique ID, and their profiles encompass additional details like name, gender, city, salary, age, phone, and hiring date.

Topics provide a structured approach to the curriculum. A course has many topics, and topics linked with the course. Each topic boasts a unique ID and a name. Exams are the means through which courses are evaluated. They are assigned a unique ID and feature details such as the exam's full mark, duration, and the number of questions.

Questions form the core of assessments, featuring a unique ID, title, type, question level, correct answer, and choices. Student answers are meticulously recorded for each question in an exam, facilitating a thorough evaluation process.

The system manages an extensive database of students, each with a unique ID and a comprehensive profile including their first name, last name, gender, faculty, phone, graduation year, city, and moreover, students are provided with access to social media platforms, each linked to their respective social IDs, profile photos, and web links. This feature enhances connectivity and enables students to establish their professional online presence. Students enroll in a specific track and may engage in various activities such as freelancing jobs, obtaining certificates, and hiring in companies.

Freelancing jobs offer students a platform to gain practical experience. Each job is uniquely identified and encompasses details such as title, cost, start date, end date, and platform. Certificates, a testament to a student's accomplishments, are granted upon meeting specific criteria. Each certificate features a certificate verification, name, platform, and duration.

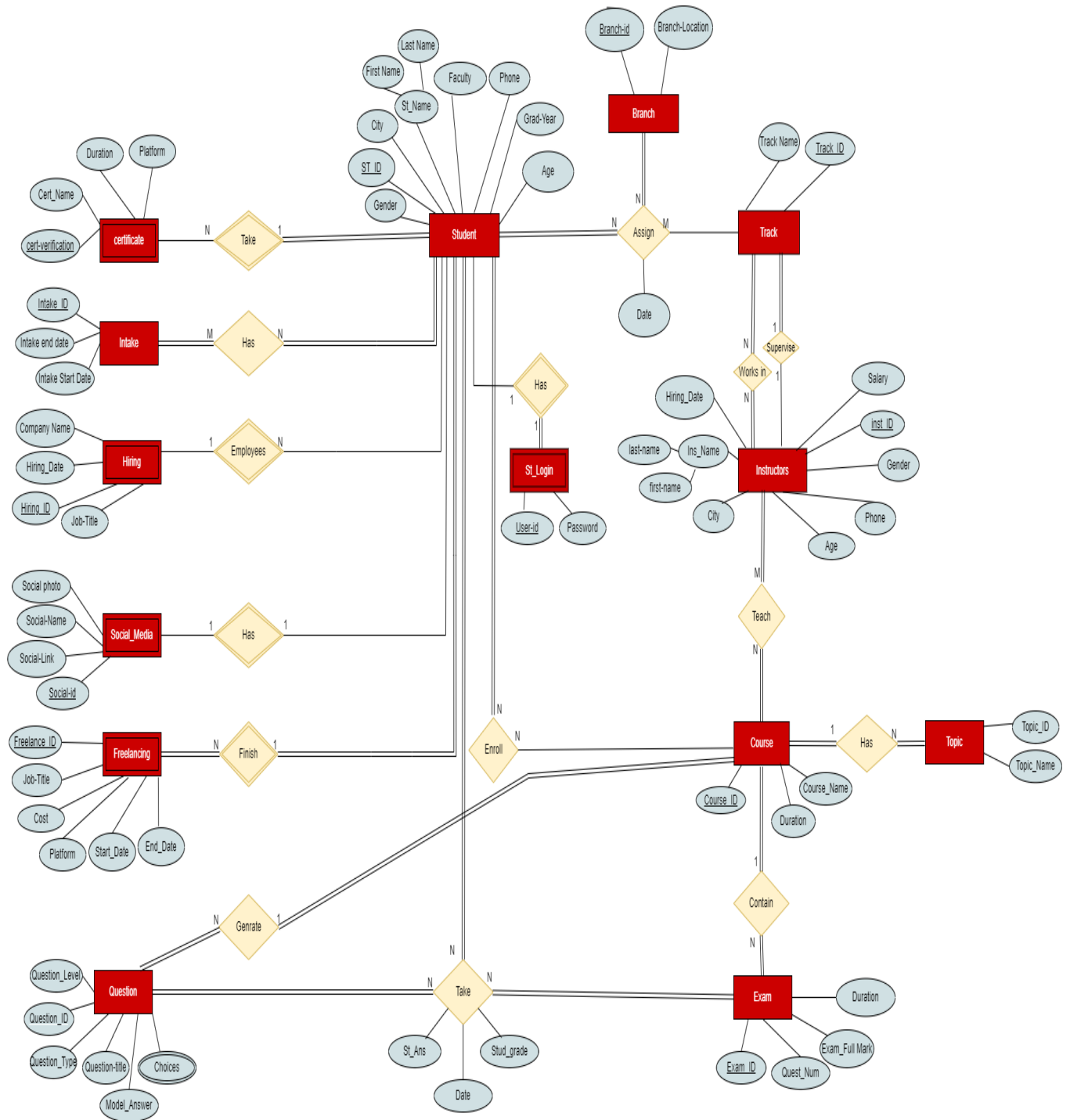
As an integral part of their educational journey, select students will have the opportunity to secure positions within various companies. These positions will encompass distinct job titles, offering students a chance to apply their acquired skills in real-world settings. Each position will come with an associated salary, reflecting the recognition of their contributions. Additionally, the hiring process will be marked by a specific date, signifying the commencement of their professional journey.

This comprehensive system streamlines ITI's educational offerings across branches, ensuring a seamless experience for students, instructors, and administrators alike. The platform covers a wide array of functionalities, ranging from course management to exam evaluations, providing a holistic educational experience.

Entities

- ❖ Tracks
- ❖ Branches
- ❖ Intake
- ❖ Courses
- ❖ Topics
- ❖ Questions
- ❖ Exam
- ❖ Students
- ❖ Instructors
- ❖ Login
- ❖ Freelance
- ❖ Certificates
- ❖ Social Media
- ❖ Hiring

Entity Relationship Diagram (ERD):



Mapping

Student (St-id →PK,city,fname,lname,gender,faculty,phone,age)

Track (track-id→PK,track-name,instid→fk)

Branch(Branch-id→PK,Branch-Location)

Instructors(inst-id→PK,first-name,last-name,gender,salary,phone,age,city,Hiring-date)

Inst-Track ((inst-id→fk,Track-id→fk) →pk)

course(course-id,course-name,Duration)

Inst-Course((inst-id→fk,course-id→fk)pk)

Std-Course ((Std-id→fk,course-id→fk)pk)

Topic (Topic-id→PK,Topic-name,course-id→fk)

Exam (Exam-id→PK,Duration,Exam-Full Mark,Question-Num,course-id→fk)

Question (Question -id→PK, Question-title , Question -Type, Model-Answer ,
Question -Level,course-id→fk)

Question-Choices ((Choices,Question-id→fk) →pk)

Intake (Intake-num→PK,,intake-start,intake-end)

St-intake((st-id→fk,intake-num→fk) →pk)

St-Login ((user id,st-id→fk) →pk,password,Username)

Certification ((cert-verification,st-id→fk) →pk),cert-name,duration,Platform)

Hiring ((hiring-id,st-id→fk) →pk),company-name,hiring-date,job-title)

Social-Media ((Social-id,st-id→fk) →pk,Social-Photo,Social-Name,Social-Link)

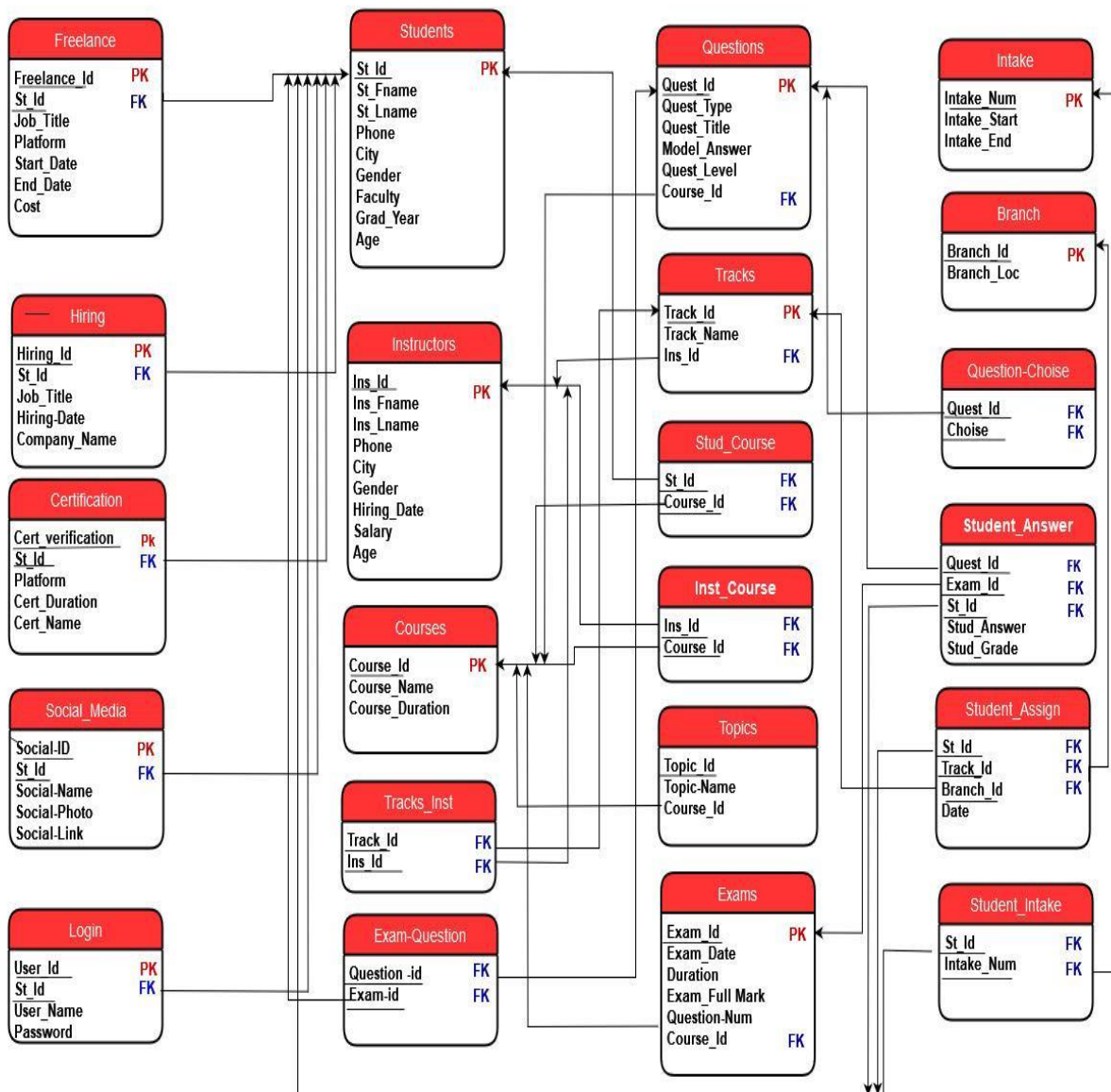
Freelancing ((freelance-id,st-id→fk)→pk,job-title,platform,start_date,End_date,cost)

Std-Answer ((Std-id→fk,Exam-id→fk,Question-id→fk) → pk),St-ans,St_Grad,date)

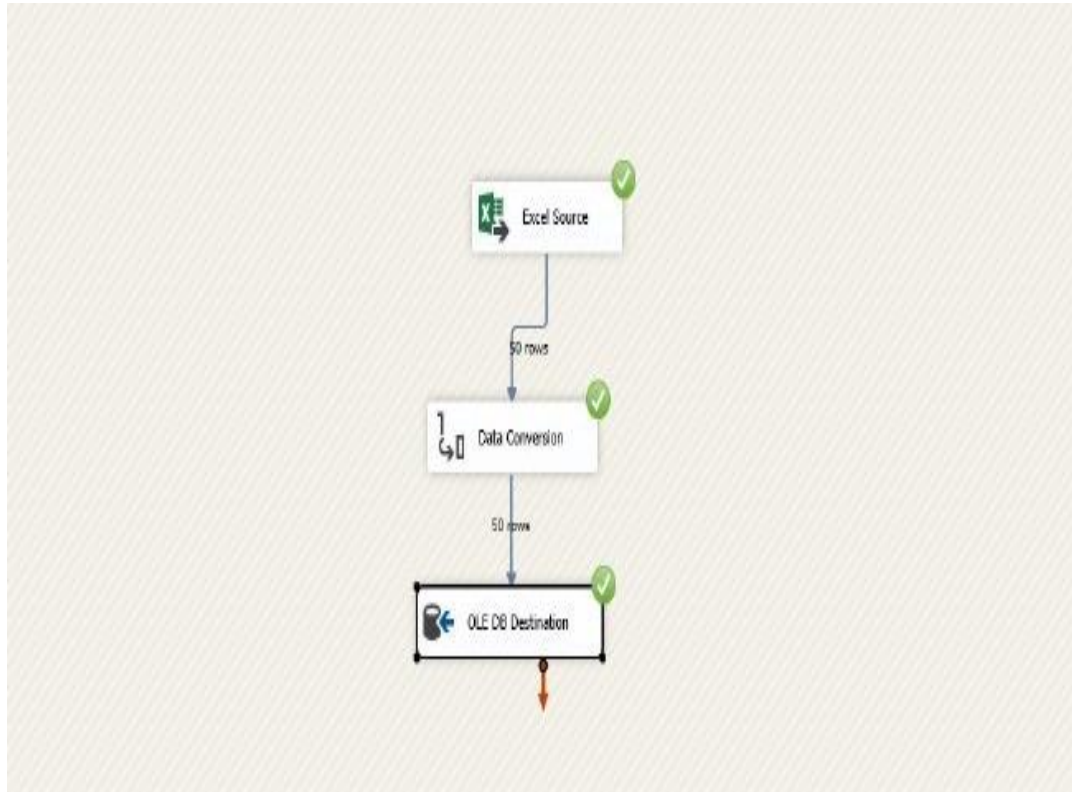
Std-Assign ((St-id→fk,track-id→fk,Branch-id→fk) →pk),Date)

Exam-Question(Question -id, →fk,,Exam-id→fk) →pk)

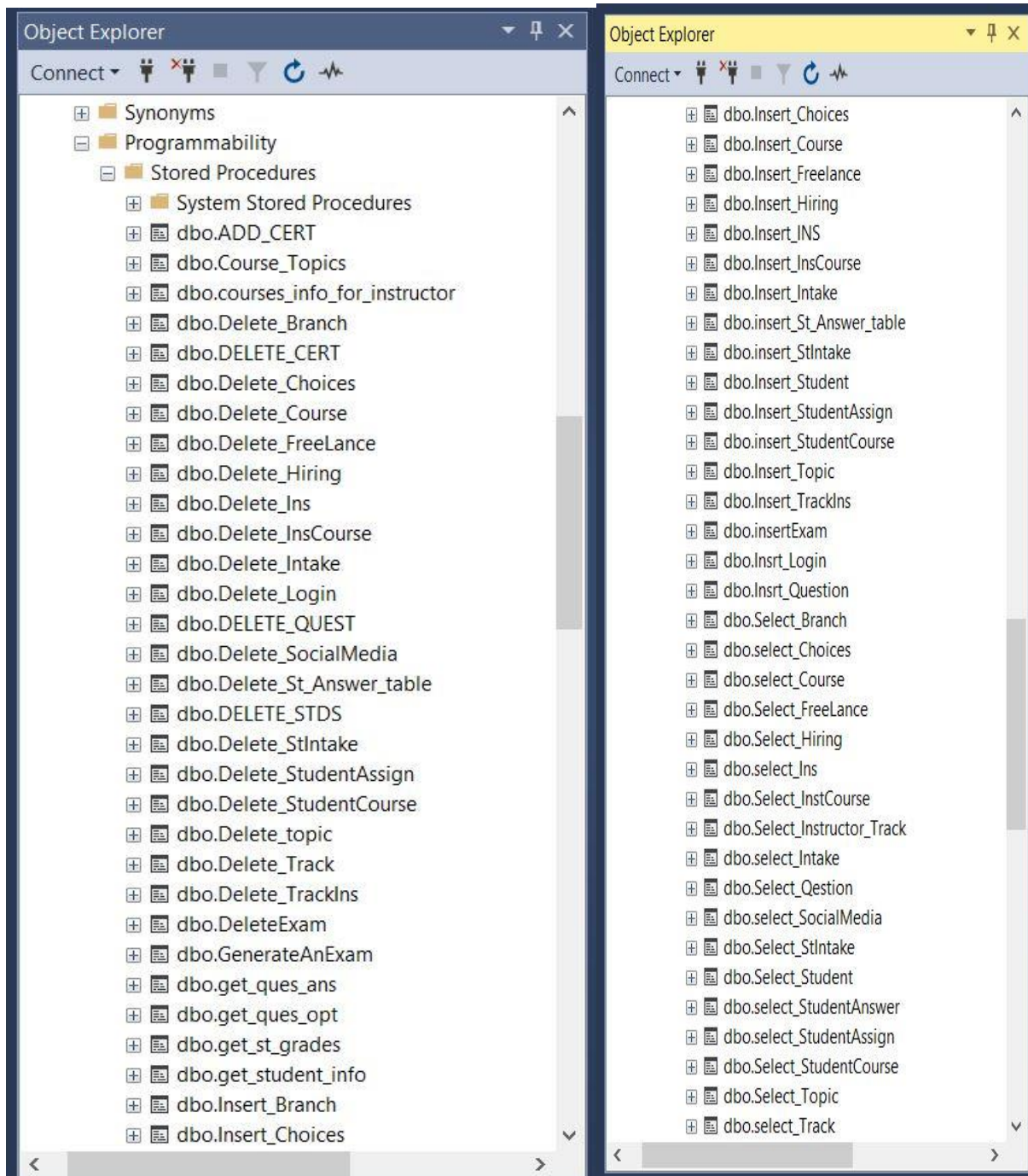
ERD



SQL Server Integration Service (SSIS)



Stored Procedures



Insert,Update, Delete, and Select For any table

EX:Upadte Track

```
USE [Examination_System]
GO
/***** Object:  StoredProcedure [dbo].[Update_Track]    Script Date: 10/21/2023 4:11:54 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[Update_Track]
(
    @Track_Id INT,
    @Track_Name VARCHAR(50),
    @Ins_Id INT
)
AS
BEGIN TRY
    UPDATE Tracks
    SET
        Track_Name = @Track_Name,
        Ins_Id = @Ins_Id
    WHERE Track_Id =@Track_Id ;
END TRY

BEGIN CATCH
    SELECT ERROR_MESSAGE() AS Errors;
END CATCH;
```

Select Track

```
USE [Examination_System]
GO
/***** Object:  StoredProcedure [dbo].[select_Track]    Script Date: 10/21/2023 4:11:11 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-----Track SELECT Proc-----
ALTER PROCEDURE [dbo].[select_Track](@Track_Id INT)
AS
BEGIN
    IF @Track_Id = 0
    BEGIN
        SELECT * FROM Tracks
    END
    ELSE
    SELECT *
    FROM Tracks
    WHERE Track_Id = @Track_Id
END
```

Delete Track

```
USE [Examination_System]
GO
/***** Object: StoredProcedure [dbo].[Delete_Track]    Script Date: 10/21/2023 4:02:06 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[Delete_Track]
(
    @Track_Id INT
)
AS
BEGIN TRY
    DELETE FROM Tracks
    WHERE Track_Id = @Track_Id;
END TRY

BEGIN CATCH
    SELECT ERROR_MESSAGE() as Errors;
END CATCH;
```

Insert Track

```
USE [Examination_System]
GO
/***** Object: StoredProcedure [dbo].[Insert_Track]    Script Date: 10/21/2023 11:43:40 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROC [dbo].[Insert_Track]
@T_NAME VARCHAR(50),
@INS_ID INT
AS
BEGIN
    BEGIN TRY
        INSERT INTO Tracks VALUES(@T_NAME,@INS_ID)
    END TRY
    BEGIN CATCH
        SELECT 'ERROR'
    END CATCH
END
```


Main Stored Procedures

```
Go
Alter PROCEDURE GenerateAnExam
    @Exam_id INT,
    @CrS_ID INT,
    @Ex_Duration INT,
    @Exam_Fullmark int ,
    @Exam_Questionnum int ,
    @Num_TF_Questions INT,
    @Num_MCQ_Questions INT
AS
BEGIN TRY
    IF NOT EXISTS (SELECT * FROM Courses WHERE Course_Id = @CrS_ID)
    BEGIN
        SELECT 'The course does not exist' AS 'ErrorMessage'
    END
    ELSE
    BEGIN
        -- Generate a random exam with 10 questions
        INSERT INTO Exams(Exam_id, Exam_Duration,Exam_FullMark,Question_Num, Course_Id)
        VALUES (@Exam_id, @Ex_Duration,@Exam_Fullmark,@Exam_Questionnum, @CrS_ID);

        -- Select True/False questions related to the same course
        INSERT INTO Exam_Question(Exam_id, Quest_id)
        SELECT TOP (@Num_TF_Questions) @Exam_id, Q.Quest_Id
        FROM Questions Q
        WHERE Q.Course_Id = @CrS_ID AND Q.Quest_Type = 't&f'
        ORDER BY NEWID();

        -- Select Multiple Choice questions related to the same course
        INSERT INTO Exam_Question(Exam_id, Quest_id)
        SELECT TOP (@Num_MCQ_Questions) @Exam_id, Q.Quest_Id
        FROM Questions Q , Question_Choise qc
        WHERE Q.Course_Id = @CrS_ID AND Q.Quest_Type = 'mcq'
        ORDER BY NEWID();

        -- Select exam model
        SELECT q.Quest_id , Quest_Title, Choices
        FROM Exam_Question eq, Questions Q, Exams E ,Question_Choise qc
        WHERE eq.Exam_Id = E.Exam_Id AND eq.Quest_id = Q.Quest_Id AND eq.Exam_Id = @Exam_id
        and qc.Quest_Id = q.Quest_Id

    END
END TRY
BEGIN CATCH
    SELECT ERROR_MESSAGE() AS errorMessage
END CATCH

GenerateAnExam @Exam_id = 2 , @CrS_ID = 2 ,@Exam_Fullmark=100, @Ex_Duration=10 , @Exam_Questionnum = 10 ,
@Num_TF_Questions=5,@Num_MCQ_Questions = 5
```

```

GO
alter PROCEDURE StudentExam_Answers
    @exam_id INT,
    @Std_ID INT,
    @Quest_Id varchar(150),
    @Std_Ans Varchar(200)
AS
BEGIN TRY
    IF NOT EXISTS (SELECT * FROM Exams WHERE Exam_Id = @exam_id)
        BEGIN
            SELECT 'The exam does not exist' AS 'ErrorMessage'
        END
    ELSE IF NOT EXISTS (SELECT * FROM Questions WHERE Quest_Id = @Quest_Id)
        BEGIN
            SELECT 'The question does not exist' AS 'ErrorMessage'
        END
    ELSE IF NOT EXISTS (SELECT * FROM Students WHERE St_Id = @Std_ID)
        BEGIN
            SELECT 'The student does not exist' AS 'ErrorMessage'
        END
    ELSE
        BEGIN
            INSERT INTO Student_Answer(St_Id, Exam_Id, Quest_Id, Date, Stud_Answer)
            VALUES (@Std_ID, @exam_id, @Quest_Id, GETDATE(), @Std_Ans);
        END
    END TRY

BEGIN CATCH
    SELECT ERROR_MESSAGE() AS ErrorMessage
END CATCH

```



```

create PROCEDURE StudentModelAnswertest
    @Std_ID INT,
    @Ex_No INT
AS
BEGIN TRY
    IF NOT EXISTS (SELECT * FROM Exams WHERE Exam_Id = @Ex_No)
        BEGIN
            SELECT 'The exam does not exist' AS 'ErrorMessage'
        END
    ELSE IF NOT EXISTS (SELECT * FROM Students WHERE St_Id = @Std_ID)
        BEGIN
            SELECT 'The student does not exist' AS 'ErrorMessage'
        END
    ELSE
        BEGIN
            -- Update Std_Score based on the student's answers
            UPDATE SE
            SET Stud_Grade = CASE WHEN q.Model_Answer = Stud_Answer THEN 1 ELSE 0 END
            FROM Student_Answer SE
            INNER JOIN Questions q ON SE.Quest_Id = q.Quest_Id
            WHERE SE.Exam_Id = @Ex_No AND SE.St_Id = @Std_ID

            -- Calculate the total number of correct answers
            DECLARE @TotalCorrectAnswers INT;
            SELECT @TotalCorrectAnswers = SUM(Stud_Grade)
            FROM Student_Answer
            WHERE Exam_Id = @Ex_No AND St_Id = @Std_ID;

            -- Calculate the grade in percentage
            DECLARE @GradePercentage DECIMAL(5, 1);
            SELECT @GradePercentage = (@TotalCorrectAnswers * 100.0) / COUNT(Quest_Id)
            FROM Student_Answer
            WHERE Exam_Id = @Ex_No AND St_Id = @Std_ID

            -- Return the grade percentage
            SELECT @GradePercentage AS GradePercentage;
        END
    END TRY
    BEGIN CATCH
        SELECT ERROR_MESSAGE() AS errorMessage
    END CATCH

    -- Test --
    GO
    StudentModelAnswertest @Ex_No = 1, @Std_ID = 13

```

Reports Stored Procedures

Report1

```
USE [Examination_System]
GO
/***** Object: StoredProcedure [dbo].[get_student_info]    Script Date: 10/21/2023 4:00:45 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-----
ALTER PROCEDURE [dbo].[get_student_info]
    @tr_id INT
AS
BEGIN
    BEGIN TRY
        SELECT *
        FROM Students
        WHERE St_Id IN (
            SELECT St_Id
            FROM Student_Assign
            WHERE track_id = @tr_id
        );
    END TRY
    BEGIN CATCH
        SELECT 'Check Your Values' AS ErrorMessage;
    END CATCH;
END;
```

Report2

```
USE [Examination_System]
GO
/***** Object: StoredProcedure [dbo].[get_st_grades]    Script Date: 10/21/2023 4:00:15 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROC [dbo].[get_st_grades] (@st_id INT)
AS
BEGIN TRY
    SELECT
        c.Course_Name,
        FORMAT(AVG(CAST((sa.Stud_Grade * 100.0 / e.Exam_FullMark) AS DECIMAL(5, 2))), '0.00%') AS Avg_Grade_Percentage
    FROM Students s
    JOIN Student_Answer sa ON sa.ST_Id = s.St_ID
    JOIN Exams e ON e.Exam_Id = sa.Exam_Id
    JOIN Courses c ON e.Course_Id = c.Course_Id
    WHERE s.ST_ID = @st_id
    GROUP BY c.Course_Name
    ORDER BY c.Course_Name
END TRY
BEGIN CATCH
    SELECT 'An error occurred. Please check your values.' AS ErrorMessage
END CATCH
```

Report3

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROC [dbo].[courses_info_for_instructor]
@Ins_Id BIGINT
AS
BEGIN
    BEGIN TRY
        IF EXISTS (SELECT * FROM Instructors i WHERE i.Ins_Id = @Ins_Id)
        BEGIN
            SELECT
                Ins_Fname + ' ' + Ins_Lname as 'Instructor' ,
                c.Course_Name as [Course Name],
                COUNT(sc.St_Id) AS 'Number of students'
            FROM
                Courses c
                INNER JOIN Inst_Course ic ON c.Course_Id = ic.Course_Id AND ic.Ins_Id = @Ins_Id
                INNER JOIN Student_Course sc ON c.Course_Id = sc.Course_Id
                INNER JOIN Instructors i on i.Ins_Id = ic.Ins_Id
            GROUP BY
                c.Course_Name , Ins_Fname , Ins_Lname
        END
        ELSE
        BEGIN
            SELECT 'This instructor does not exist' as Error
        END
    END TRY
    BEGIN CATCH
        SELECT 'An error occurred while processing the request' as Error
    END CATCH
END
```

Report4

```
USE [Examination_System]
GO
/***** Object: StoredProcedure [dbo].[Course_Topics]    Script Date: 10/21/2023 3:56:32 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[Course_Topics] (@CrId INT)
AS
BEGIN
    DECLARE @Course_Name VARCHAR(50)

    SELECT @Course_Name = Course_Name
    FROM Courses
    WHERE Course_Id = @CrId

    IF @Course_Name IS NOT NULL
    BEGIN
        SELECT @Course_Name AS [Course Name], Topic_Name AS [Topic Name]
        FROM Topics
        WHERE Course_Id = @CrId
    END
    ELSE
    BEGIN
        SELECT 'This course is not found' AS Error
    END
END
```

Report5

```
SQLQuery32.sql -...U5UUHQ\Magic (67)  SQLQuery31.sql -...U5UUHQ\Magic (53)  SQLQuery26.sql -...U5UUHQ\Magic (66)*
USE [Examination_System]
GO
/***** Object: StoredProcedure [dbo].[get_ques_opt]    Script Date: 10/21/2023 3:59:35 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[get_ques_opt]
    @ex_id INT
AS
BEGIN
    -- Try to select all questions and their options for the specified exam, with the choices aggregated into a single string.
    BEGIN TRY
        SELECT
            DISTINCT Q.Quest_Title,
            STRING_AGG(QC.Choices,char(10)) AS Choices
        FROM
            Questions Q
        INNER JOIN
            Question_Choise QC ON Q.Quest_Id = QC.Quest_Id
        INNER JOIN
            Student_Answer SA ON Q.Quest_Id = SA.Quest_Id
        INNER JOIN
            Exams ON SA.Exam_Id = Exams.Exam_Id
        WHERE
            Exams.Exam_Id = @ex_id
        GROUP BY
            Q.Quest_Title
        ORDER BY
            Q.Quest_Title;
    END TRY
    BEGIN CATCH
        SELECT 'Check Your Values';
    END CATCH;
END;
```

Report 6

```
USE [Examination_System]
GO
/***** Object: StoredProcedure [dbo].[get_ques_ans]    Script Date: 10/21/2023 3:58:45 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[get_ques_ans]
    @ex_id INT,
    @st_id INT
AS
BEGIN
    BEGIN TRY
        SELECT
            DISTINCT Q.Quest_Title,
            STRING_AGG(QC.Choices, char(10)) AS Choices,
            SA.Stud_Answer
        FROM
            Questions Q
        INNER JOIN
            Question_Choise QC ON Q.Quest_Id = QC.Quest_Id
        INNER JOIN
            Student_Answer SA ON Q.Quest_Id = SA.Quest_Id
        INNER JOIN
            Exams ON SA.Exam_Id = Exams.Exam_Id
        WHERE
            SA.Exam_Id = @ex_id AND
            SA.St_Id = @st_id
        GROUP BY
            Q.Quest_Title,
            SA.Stud_Answer
        ORDER BY
            Q.Quest_Title;
    END TRY
    BEGIN CATCH
```

SQL Server Reporting Service (SSRS)

Students Information by Track

Phone	Faculty	Grad_Year	Age
1148572510	FacultyofComputerandInformatics	2021	24
1006542728	Commerce	2019	27
1116651109	ComputerScience	2022	27
1116111102	DataSciene	2022	26
1151101220	ComputerofScience	2022	23
1055589154	Engineering	2022	24
1055563612	Engineering	2021	25
1055589791	Engineering	2022	24
1055579137	computerscience	2023	23
1055539989	computerscience	2023	23
1155518013	DataSciene	2018	27
1055569045	computerscience	2019	26
1055565718	computerscience	2023	23
1255589512	FacultyofComputerandInformatics	2021	25
1255578729	computerscience	2019	26
1255510779	FacultyofComputerandInformatics	2019	26
1255507442	computerscience	2018	27
1055566590	Engineering	2019	26
1055545621	computerscience	2022	24
1155505219	Engineering	2023	23
1055598130	FacultyofComputerandInformatics	2023	23
1255503595	computerscience	2018	27
1055579621	computerscience	2019	26

Courses by Instructors



Instructor	Course Name
Ramy Mohamed	DataMining
Ramy Mohamed	XML



Topics by Courses

Topic Name
XML Syntax
Elements and Attributes
XML Declaration
Well-Formed XML vs. Valid XML
XML Schema (XSD)

Students Grade

Course Name	Grade
XML	90

Report5

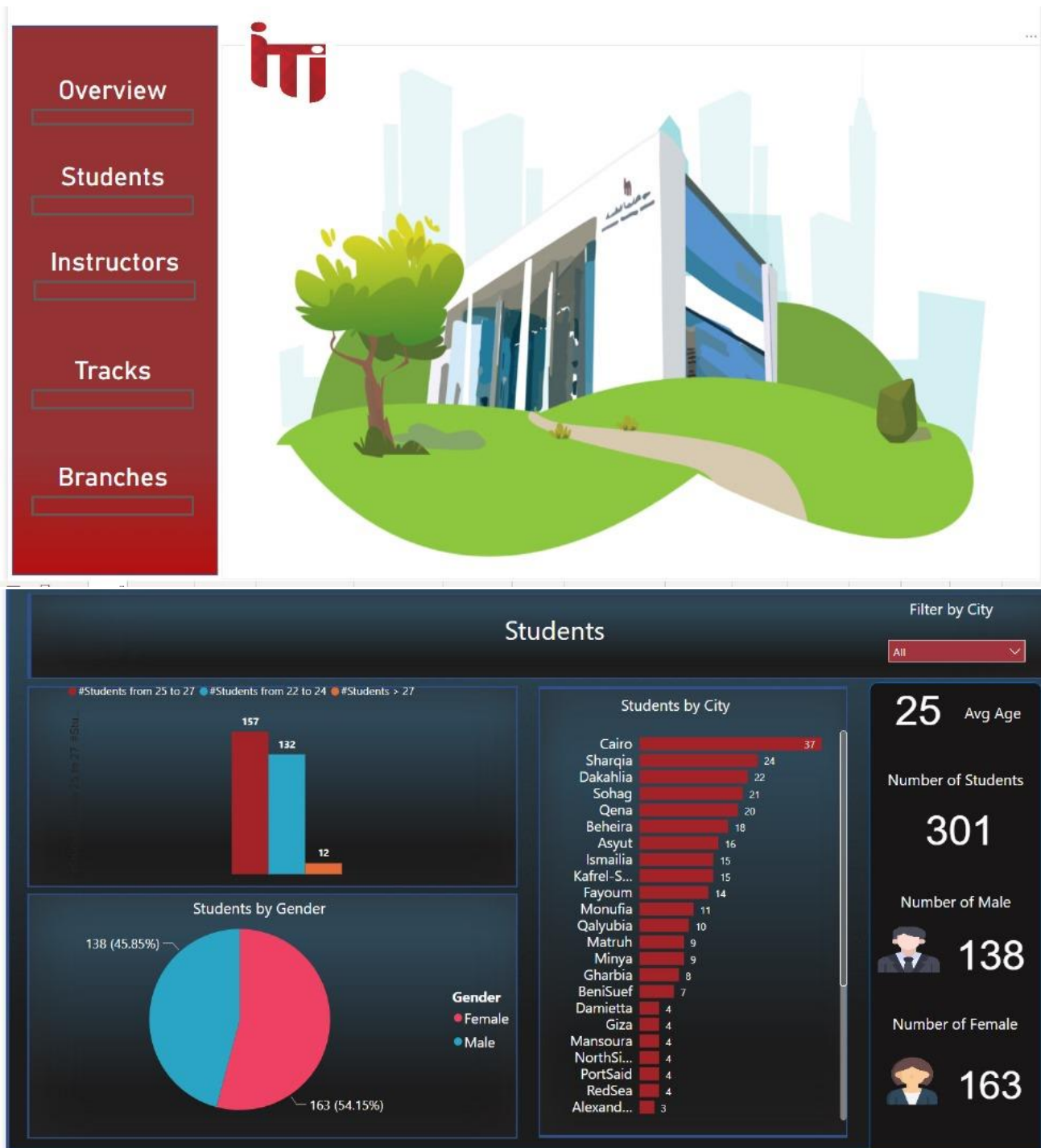
Question Num	Question Title	Choices
1	Business intelligence equips enterprises to gain business advantage from data	A) True B) False
2	BI helps in identifying and understanding the relationship between business processes and performance?	a) True b) False
3	Big Data refers to the storage of data in traditional relational databases?	a) True b) False

Report 6



Quest Title	Choices	Stud Answer
Business intelligence equips enterprises to gain business advantage from data	A) True B) False	True
BI helps in identifying and understanding the relationship between business processes and performance?	a) True b) False	True
Big Data refers to the storage of data in traditional relational databases?	a) True b) False	False
Business intelligence (BI) is a broad category of application programs which includes ?	A) Decision support B) Data Mining C) OLAP D) All of the above	All of the above
Classification is	a) A subdivision of a set of examples into a number of classes b) A measure of the accuracy, of the classification of a concept that is given by a certain theory c) The task of assigning a classification to a set of examples d) None of these	The task of assigning a classification to a set of examples
Data Mining can only be performed using specialized software and tools	a) True b) False	True
Data warehouse architecture is based on	a) RDBMS b) Sybase c) SQL Server d) DBMS	DBMS
KPI stands for?	A) Key Performance Indicators B) Key Performance Identifier C) Key Processes Identifier D) Key Processes Indicators	Key Performance Indicators
MapReduce can best be described as a programming model used to develop Hadoop-based applications that can process massive amounts of data.	a) True b) False	False
Reports use Data Management to retrieve data for a report when a query runs or when the report is processed.	a) True B) False	False

Power BI Dashboards



OverView

#Students

301

#Instructors

58

#Intakes

4

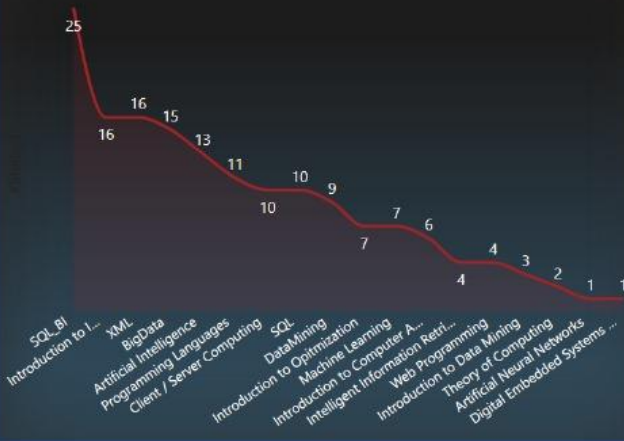
#Tracks

13

#Branch

8

Students by Courses



Students and Instructors by Branch

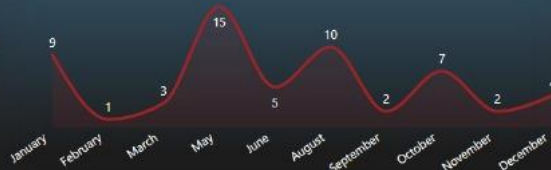


Instructors

Filter by Hiring Date

All

Instructors by Hiring



Instructors Name	Sum of Salary
Ahmed Adel	40000
Khaled Ahmed	40000
Mohamed Ammar	40000
Noha Shehab	40000
Ramy Mohamed	40000
Bassant Mamdoh	30000
Total	754000

13K Avg Salary

Number of Instructors

58

Number of Male



32

Number of Female



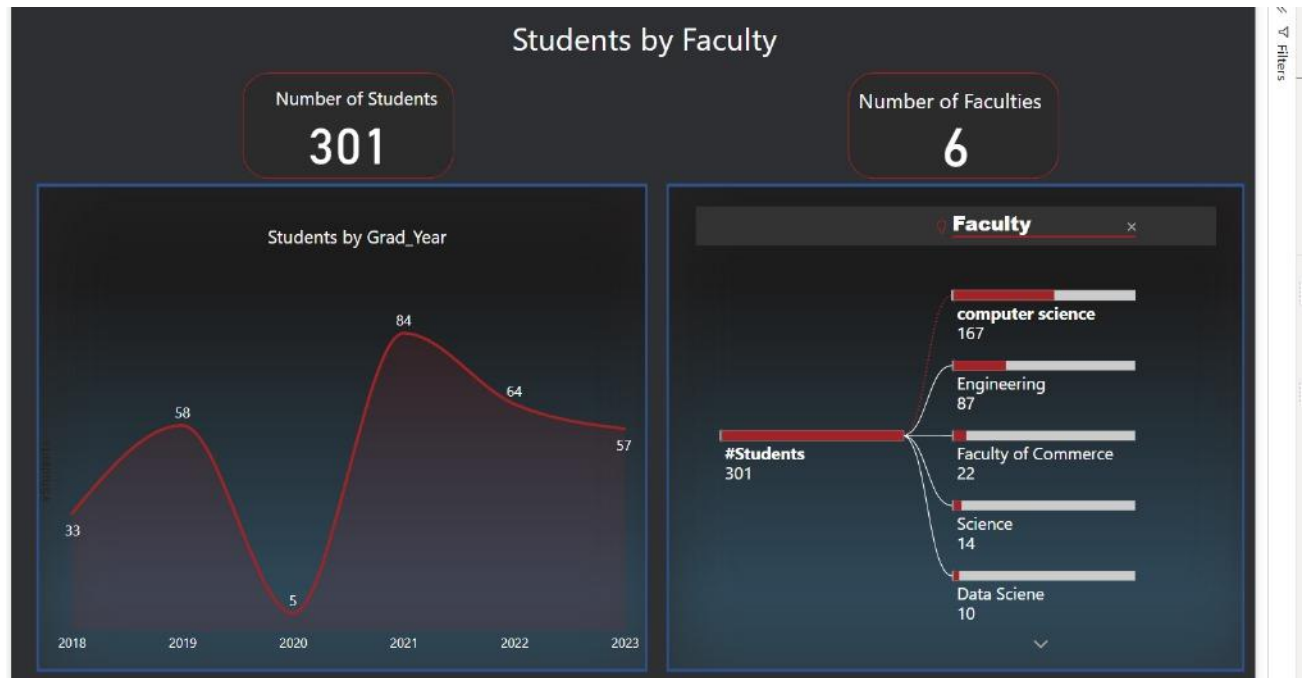
26

Instructors by Courses

Instructors Name	Course Name	Sum of Course Duration
Ahmed Adel	Artificial Intelligence	18
Ahmed Adel	Client / Server Computing	36
Ahmed Montasr	Introduction to Data Mining	24
Bassant Mamdoh	SQL_BI	18
Bassant Mamdoh	Web Programming	12
Ereen Gawesh	Programming Languages	18
Ereen Gawesh	SQL_BI	18
Khaled Ahmed	DataMining	6

Instructors by City





Students Sheet

St_Id	Student Name	Gender	City	Phone	Grad_Year	Faculty	Age	Branch Location	
19	Rana Emad	Female	Alex	01116651109	2022	computer science	27	Menya	Power Bi Devel
23	Omar Kamel	Male	Alexandria	01010946068	2020	Engineering	25	Alex	Cyber Security
245	Sadaf Nabeel	Female	Alexandria	01155540376	2022	Engineering	24		
255	Adara Fazli	Female	Alexandria	01055595891	2021	computer science	25		
13	Eman Ashraf	Female	ALMahalla AlKubra	01287595062	2022	computer science	25	Mansoura	Social Media M
176	Safa Hatem	Female	Aswan	01255520520	2021	computer science	25	Alex	Data Science
227	Hana Amir	Female	Aswan	01055529408	2021	computer science	25		
29	Dawoud Louay	Male	Asyut	01055515730	2021	Engineering	25	Mansoura	Data Engineeri
90	Nasser Hassan	Male	Asyut	01255509411	2021	computer science	25	Mansoura	Linux System A
99	Rami Ja'far	Male	Asyut	01255515224	2021	computer science	25	Mansoura	Software Archi
101	Rayan Monur	Male	Asyut	01155561370	2019	computer science	26	Mansoura	Software Archi
112	Abid Odai	Male	Asyut	01255534478	2022	computer science	24	Mansoura	Software Testir
121	Shadi Abbas	Male	Asyut	01255508255	2019	computer science	26	Smart Village	Software Archi
125	Taher Yasser	Male	Asyut	01155556859	2018	Engineering	28	Smart Village	Web & Mobile
129	Tawfiq Ehsan	Male	Asyut	01255550394	2022	Engineering	24	Alex	Data Engineeri
133	Yahya Hussein	Male	Asyut	01255571104	2019	computer science	26	Alex	Web & Mobile
138	Zaid Jamal	Male	Asyut	01055558770	2021	computer science	25	Mansoura	Data Science
157	Muhsina Faye	Female	Asyut	01155543670	2018	Engineering	27	Menoufia	Power Bi Devel
163	Salwa Abd	Female	Asyut	01155531360	2019	computer science	22	Mansoura	Cyber Security

Filter by City

All

Filter by Age

All

Filter by Track

All

Filter by Gender

All

Filter by Faculty

All



