ITI Examination System

🎓 Project Summary

Our project is an **Automated Examination System** designed to manage, conduct, and analyze online exams efficiently.

The system is built using **SQL Server** as the main database engine and enhanced with **Power BI dashboards and reports** to provide valuable insights for ITI staff.

Our database is designed to be comprehensive — it holds all essential academic data, including **students, instructors, intakes, courses, tracks, branches**, and more.

This structure ensures that ITI can easily manage exams, track student performance, and generate detailed analytical reports — all from a single integrated system

🔧 Objectives

- Create a reliable and secure **SQL database** to store and manage exam data.
- Automate the full **exam lifecycle** — from generation and answering to correction and grading.
- Provide **dynamic reports and dashboards** to assist instructors and administrators in decision-making.

🧱 System Components

1. Database Layer (SQL Server):

- Designed a comprehensive **database dictionary** covering all essential entities, including **students, instructors, courses, exams, questions, and answers**.
- Implemented **stored procedures** to perform the core database operations — SELECT, INSERT, UPDATE, and DELETE — ensuring data consistency and efficiency.
- Developed **automated exam generation logic** that creates a **unique set of random questions for each student**, enhancing exam fairness and reducing repetition.
- The system's database also **stores each student's answers** securely for further **evaluation, grading, and analytical reporting**.

- Integrated **exam correction procedures** that automatically compare student answers with correct solutions to calculate grades.

2. Reporting and Analytics Layer (Power BI & SSRS):

- Developed a set of **SQL-based reports** and **Power BI dashboards** to help ITI staff easily monitor and analyze exam-related data.
- Designed **reports** that allow users to:
    - View student information by department.
    - Display student grades across all courses.
    - Show instructor course assignments and the number of students per course.
    - Retrieve course topics and related exam details.
    - Display exam questions and corresponding student answers for any selected exam.
- Integrated **Power BI** with SQL Server to visualize performance trends, student distributions, and course statistics through **interactive dashboards**.
- Used **slicers, KPIs, and filters** in Power BI to provide a dynamic experience for users — allowing real-time data exploration and analysis.
- Enabled **data-driven decision-making** by transforming raw SQL data into clear, actionable visual insights.

Outcomes

This system enables **ITI** to:

- Digitize examination processes.
- Reduce manual errors in grading.
- Improve transparency in student performance tracking.
- Empower staff with actionable insights through interactive Power BI dashboards.
- **Web and Mobile Interface:** Allow students to take exams online through a secure portal.

Conclusion:

The **Automated Examination System** successfully achieves its goal of managing and analyzing the examination process digitally.
By utilizing **SQL Server** as the core database engine, the system provides a structured and secure way to store all academic data — including students, instructors, courses, exams, and answers.
Through the integration with **Power BI**, ITI staff can easily visualize and analyze results, monitor performance, and generate detailed reports in real time.
The system improves accuracy, reduces manual workload, and ensures fairness through **automated exam generation** and **randomized question distribution** for each student.

Future Work:

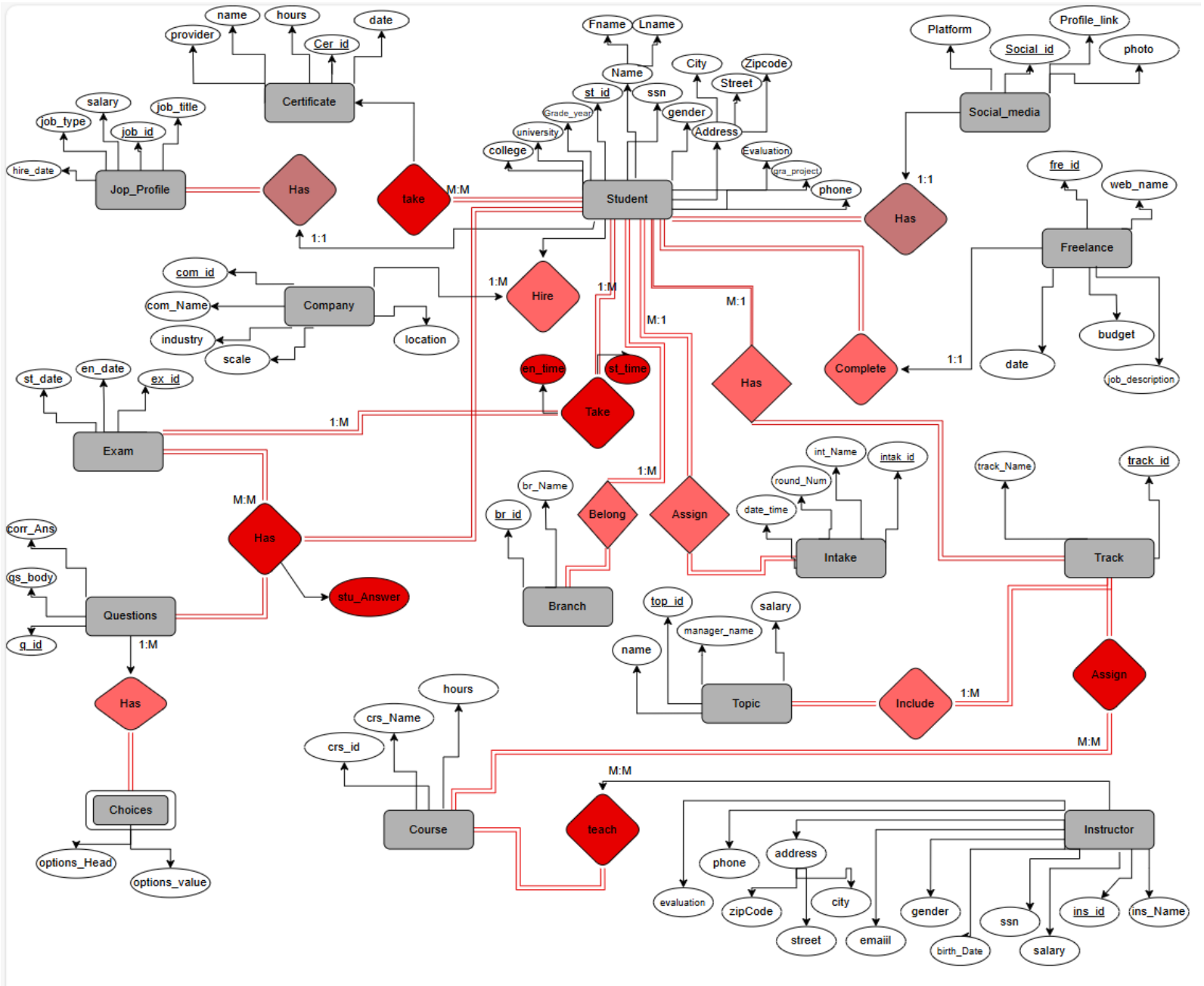
To further enhance the system, several extensions can be implemented:

- **AI-Based Analytics:** Predict student performance and identify learning gaps.
- **Enhanced Security:** Implement authentication layers and encryption for exam data.
- **Integration with Learning Systems:** Connect with LMS (Learning Management System) platforms for continuous data sharing.
- **Automated Notifications:** Send exam results or feedback through email or integrated social media APIs.

Database Design

ERD (Entity Relationship Diagram)

The backbone of whole system.



Main Entities

Entity (Table)	Purpose
Student	Stores all student information personal and academic data.
Instructor	Stores instructor data
Course	Contains course details
Track	Defines learning tracks (like Data, Web, AI, etc.)
Branch	Stores ITI branch information
Intake	Defines student intake or batch
Exam	Represents each exam event and Exam details
Question	Stores the exam questions
Choices	Stores option headers and values
Topic	Stores main departments and their managers
Project	Stores projects' data
Job_Profile	Contains student's jobs for students which had jobs
Certificate	Stores certificates earned by students
Freelance	Stores Freelance projects or work done by students
Company	Stores companies where students work or intern
Social_Media	Stores students' social media or professional links

Relationships

Entities Involved	Relationship	Type (Cardinality)	Description
Student – Track	Enroll	Many-to-One (M–1)	Each student is enrolled in one track; a track can have many students.
Student – Certificate	Take	Many-to-Many	A student can take multiple certificates, and each certificate can be taken by multiple students.
Student – Freelance	Complete	One-to-One	A student can complete One freelance and Freelance completed by one student
Student – Project	Belong	Many-to-Many	Students can belong to multiple projects, and each project may have several students.
Intake – Student	Assign	One-to-Many (1–M)	Each intake can include many students, but each student is enrolled in only one intake.
Student – SocialMedia	Has (social media)	One-to- One (1–1)	Each student is assigned to one social media platform or professional profile.
Job_Profile – Student	Has	One-to-One (1–1)	Every Job Assigned to one student, and a student works for one company at a time.
Student – Exam	Take	Many-to-Many	Students can take multiple exams, and each exam can be taken by many students.
Branch – Student	Belong	1 : M (One-to-Many)	Each branch can have many students, but each student belongs to only one branch.
Track – Branch	Has	Many to-Many	Each track is associated with one or more branches, while a branch can have multiple tracks.
Topic – Track	Include	One-to-Many	A topic includes several tracks.
Track – Course	Enroll	Many-to-Many	A course can is assigned to multiple tracks, and track contains multiple courses.
Project – Track	Belong	One-to-One	Every project belongs to one track

Instructor – Track	Supervises	One-to-Many (1–M)	One instructor supervises multiple tracks.
Instructor – Course	Assign	Many-to-Many (M–N)	Instructors can teach multiple courses, and courses can have multiple instructors.
Course – Exam	Has (Exam)	One-to-Many (1–M)	Each course can have several exams.
Exam – Question	Has (Questions)	One-to-Many (1–M)	Each exam consists of several questions.
Question – Choices	Has (Options)	One-to-Many (1–M)	Each question has multiple options (Choices).

Entities' Attributes

1.student

Student

- Primary Key: **ST_ID**

- Attributes: ST_ID, SSN, Fname, Lname, Gender, Birth_Date, Age, Address (City, Street, Zip_Code), Phone, University, College, Grade_Year, Grade_project, Evaluation, Email, Password

2. Instructor

Instructor

- Primary Key: **Ins_ID**

- Attributes: Ins_ID, SSN, Ins_FName, Ins_LName, Birth_Date, Gender, Email, Phone, Address(City, Street, Zip_Code), Salary , Evaluation

3. Course

Course

- Primary Key: **Crs_ID**
- Attributes: Crs_ID, Crs_Name, Hours

4. Track

Track

- Primary Key: **Track_ID**
- Attributes: Track_ID, Track_Name

5. Branch

Branch

- Primary Key: **Branch_ID**
- Attributes: Branch_ID, Branch_Name

6. Intake

Intake

- Primary Key: **Int_ID**
- Attributes: Int_ID, Start_date, End_date , Int_Name, Round_num

7.Exam

Exam

- Primary Key: **Ex_ID**
- Attributes: Ex_ID, Start_Time , End_Time, Exam_Date

8. Questions

Questions

- Primary Key: **Qs_ID**
- Attributes: Qs_ID, Qs_Body, Correct_Answer, Correct_Answer, Crs_Id, Q_Type

9. Certificate

Certificate

- Primary Key: **Cer_ID**
- Attributes: Cer_ID, Name, Hours, Provider

10. Freelance

Freelance

- Primary Key: **Fre_ID**
- Attributes: Fre_ID, Website_name, Date, Budget, Job_description

11. Company

Company

- Primary Key: **Com_ID**
- Attributes: Com_ID, Com_name, Location, Industry, Scale

12. Social_Media

Social_Media

- Primary Key: **Social_ID**
- Attributes: Social_ID, Platform_Name, Profile_Link, Photo

13. topic

Topic

- Primary Key: **Topic_ID**
- Attributes: Topic_ID, Topic_Name, Manager_Name, Salary

14. Project

Project

- Primary Key: **Pro_ID**
- Attributes: Pro_ID, Pro_Name

ERD Mapping & Implementation

- **Student** (ST_ID **P.K**, SSN, Fname, Lname, Gender, Birth_Date, Age, Address (City, Street, Zip_Code), Phone, University, College, Grade_Year, Grade_project, Evaluation, Email, Password, com_Id **F.K**, pro_Id **F.K**, track_Id **F.K**, intake_Id **F.K** , Branch_Id **F.K**) MAIN
- **Company** (Com_ID **P.K**, Com_name, Location, Industry, Scale)
- **Project** (Pro_ID **P.K**, Pro_Name track_Id **F.K**)
- **Freelance** (Fre_ID **P.K**, Website_name, Date, Budget, Job_description, ST_ID **F.K**,)
- **Course** (Crs_ID **P.K**, Crs_Name, Hours)
- **Student_Certificate** (st_Id **F.K**, cer_Id **F.K) P.K**
- **Student_Exam** ((ex_Id **F.K**, st_Id **F.K) P.K**, start_Time, end_Time, score, Grade, Percentage)
- **Stu_Exam_Ques** ((EX_Id **F.K**, Qs_ID **F.K**,St_ID **F.K) P.K**, Stu_Answer, Question_Order)
- Exam (EX_Id **P.K**, Start_Time, End_Time, Crs_ID **F.K**)
- **Questions** (Qs_ID **P.K**, Qs_Body, Correct_answer, Q_Type, Crs_ID **F.K**)
- **Choices** (Qs_Id **F.K**, Options_Header, Option_Value)
- **Ins_Course** (crs_Id **F.K**, Ins_Id **F.K) P.K**
- **Track_Course** (Track_Id **F.K**, Crs_Id **F.K) P.K**
- **Branch_Track**(Track_Id **F.K**, Branch_Id **F.K) P.K**
- **Track**(Track_Id **P.K**, Name, Ins_Id **F.K**,topic_Id **F.K**)
- **Intake**(Intake_Id **P.K**, Start_Date, End_Date,Int_name,Round_num)
- **Certificate**(Cer_Id **P.K**, Name, Hours, Provider, date)
- **Social_Media**(Social_Id **P.K**, Platform, Profile_link, Photo,st_Id **F.K**)
- **Branch**(Branch_Id **P.K**, Br_Name)
- **Instructor** (Ins_ID **P.K**, Ins_Name, Salary, SSN, Birth_Date, Gender, Email, City, Street, ZIP_Code,PHONE,Evaluation)
- **Topic**(Topic_ID **P.K**, Name,Manager_Name, Salary)

Building Database In SQL

First of All, We Started by Creating Database itself, after that We will Create Tables and constrains that like relationships between tables.

Building the Database in SQL

In this phase, the database for the Examination System was designed and implemented using **Microsoft SQL Server**. The process began with creating the main database structure, followed by defining all required **tables** that represent the system's core entities such as Students, Instructors, Courses, Exams, and Tracks.

Each table was carefully constructed with appropriate **constraints**, including **primary keys**, **foreign keys**, and **check constraints**, to ensure **data integrity**, **consistency**, and **accurate relationships** between entities. This stage established the foundation for efficient data handling and integration with **Power BI** for analysis and visualization.

Create Table Student

```sql
-- Table: Student
-- Description: Stores student personal and academic information.
-- Linked to Branch, Track, and Intake tables through foreign keys.

CREATE TABLE Student (
    ST_id INT IDENTITY(1,1) PRIMARY KEY,  -- Unique student identifier (auto-increment)
    SSN CHAR(14),
    FName NVARCHAR(50) NOT NULL,
    LName NVARCHAR(50) NOT NULL,
    Gender CHAR(1) CHECK (Gender IN ('M','F')),
    Birth_Date DATE,
    City NVARCHAR(100),
    Street NVARCHAR(100),
    Zip_Code NVARCHAR(10),
    Phone NVARCHAR(11),
    Grade_Year NVARCHAR(10),
    Pro_ID INT,
    Track_ID INT,
    Intake_ID INT,
    Social_ID INT,
    Branch_ID INT,
    College NVARCHAR(100),
    University NVARCHAR(100),
    Age INT,
    Grade_Project NVARCHAR(20),
    Evaluation NVARCHAR(100),
    Email NVARCHAR(100),
    Password NVARCHAR(100),
    job_ID INT
)
```

110 % No issues found

Messages

Completion time: 2025-10-28T19:51:49.0996467+03:00

```
SELECT * FROM Student
```

No issues found

Results | Messages

ST_id	SSN	FName	LName	Gender	Birth_Date	City	Street	Zip_Code	Phone	Grade_Year	Pro_ID	Track_ID	Intake_ID	Social_ID	Branch_ID	College	University	Age	Grade_Project	Evaluation	Email	Password	job_ID

Create Table Company

```
CREATE TABLE Company (
    Com_ID INT IDENTITY(1,1) PRIMARY KEY,
    Com_name NVARCHAR(100) NOT NULL,
    Location VARCHAR(50),
    Industry NVARCHAR(100),
    Scale NVARCHAR(50)
    )
```

No issues found

Messages

```
Commands completed successfully.
```

```
SELECT * FROM Company
```

No issues found

Results | Messages

Com_ID	Com_name	Location	Industry	Scale

Create Table Project

```
-- Table: Project
-- Description: Stores information about student or track-based projects.
-- Each project is associated with a specific Track.

CREATE TABLE Project (
    Pro_ID INT IDENTITY(1,1) PRIMARY KEY,
    Pro_Name NVARCHAR(100) NOT NULL,
    Track_ID INT )
```

No issues found

Messages

```
Commands completed successfully.
```

```sql
SELECT * FROM Project
```

110 % No issues found

Results Messages

Pro_ID	Pro_Name	Track_ID

Alter Table Project

```sql
-----------------------------------------------------------------------------------
-- Table: project
-- Description: Add Contraint

ALTER TABLE Project
ADD
    CONSTRAINT FK_Project_Track FOREIGN KEY (Track_ID) REFERENCES Track(Track_ID)
```

110 % 1 0

Messages

Commands completed successfully.

Create Table Freelance

```sql
-----------------------------------------------------------------------------
-- Table: Freelance
-- Description: Stores information about freelance projects or online jobs completed by students.

CREATE TABLE Freelance (
    Fre_ID INT IDENTITY(1,1) PRIMARY KEY,
    Website_name NVARCHAR(100),
    Date DATE,
    Budget INT,
    Job_description NVARCHAR(500),
    ST_ID INT )
```

110 % No issues found

Messages

Commands completed successfully.

Alter Table Freelance

```
384  ------------------------------------------------------------------------------
385  -- Table: Freelance
386  -- Description: Add Contraint
387
388  ALTER TABLE Freelance
389  ADD
390      CONSTRAINT FK_Freelance_Student FOREIGN KEY (ST_ID) REFERENCES Student(St_ID)
391
392
393
394
```

110 % 1 0

Messages

Commands completed successfully.

Create Table Course

```
94   ------------------------------------------------------------------------------
95   -- Table: Course
96   -- Description: Stores information about the courses offered in each track.
97
98   CREATE TABLE Course (
99       Crs_ID INT IDENTITY(1,1) PRIMARY KEY,
100      Crs_Name VARCHAR(50) NOT NULL,
101      Hours INT )
102
103
104
105
106
```

110 % No issues found

Messages

Commands completed successfully.

Create Table Certificate

```
101      Hours INT )
102  ------------------------------------------------------------------------------
103  -- Table: Certificate
104  -- Description: Stores information about professional or academic certificates
105  -- obtained by students.
106
107  CREATE TABLE Certificate (
108      Cer_ID INT IDENTITY(1,1) PRIMARY KEY,
109      Name NVARCHAR(100) NOT NULL,
110      Hours INT CHECK (Hours > 10),              -- Training hours must be greater than 10
111      Provider NVARCHAR(100)
112  )
113
114
115
```

110 % No issues found

Messages

Commands completed successfully.

Create Table Student_Certificate & Its Constraints

```sql
-----------------------------------------------------------------------------
-- Table: Student_Certificate
-- Description: Junction table (bridge) to handle the many-to-many relationship
-- between Students and Certificates.

CREATE TABLE Student_Certificate (
    St_ID INT NOT NULL,      -- References Student table
    Cer_ID INT NOT NULL,     -- References Certificate table

    -- Composite Primary Key ensures each student-certificate pair is unique
    CONSTRAINT PK_Student_Certificate PRIMARY KEY (St_ID, Cer_ID),

    -- Foreign Keys to maintain referential integrity
    CONSTRAINT FK_Student_Certificate_ST FOREIGN KEY (St_ID) REFERENCES Student(St_ID),
    CONSTRAINT FK_Student_Certificate_Cer FOREIGN KEY (Cer_ID) REFERENCES Certificate(Cer_ID)
)
```

110 % 1 0

Messages

```
Commands completed successfully.
```

Create Table Exam & Its Constraints

```sql
-----------------------------------------------------------------------------
-- Table: Exam
-- Description: Stores information about course exams including start and end times.
-- Each exam is associated with a specific course.

CREATE TABLE Exam (
    Ex_ID INT IDENTITY(1,1) PRIMARY KEY,
    Start_Time TIME,
    End_Time TIME,
    Crs_ID INT,                          -- References Course table (exam belongs to one course)

    -- Define foreign key relationship
    CONSTRAINT FK_Exam_Course FOREIGN KEY (Crs_ID) REFERENCES Course(Crs_ID)
)
```

Create Table Questions & Its Constraints

```sql
-----------------------------------------------------------------------------
-- Table: Questions
-- Description: Stores all exam questions for each course, including the question body and correct answer.

CREATE TABLE Questions (
    Qs_ID INT IDENTITY(1,1) PRIMARY KEY,
    Qs_Body VARCHAR(100) NOT NULL,
    Correct_Answer VARCHAR(60) NOT NULL,
    Crs_ID INT,                          -- References the course this question belongs to

    -- Define foreign key constraint to maintain referential integrity
    CONSTRAINT FK_Questions_Course FOREIGN KEY (Crs_ID) REFERENCES Course(Crs_ID)
)
```

110 % No issues found

Messages

```
Commands completed successfully.
```

Create Table Student_Exam & Its Constraints

```sql
144  ----------------------------------------------------------------------------------
145  -- Table: Student_Exam
146  -- Description: Junction table (bridge) representing the many-to-many relationship
147  -- between Students and Exams. Stores each student's participation and exam timing.
148
149  CREATE TABLE Student_Exam (
150      St_ID INT NOT NULL,                          -- References Student table
151      Ex_ID INT NOT NULL,                          -- References Exam table
152      Start_Time TIME,
153      End_Time TIME,
154
155      -- Composite primary key to ensure each student-exam pair is unique
156      CONSTRAINT PK_Student_Exam PRIMARY KEY (St_ID, Ex_ID),
157
158      -- Foreign key linking to Student table
159      CONSTRAINT FK_Student_Exam_Student FOREIGN KEY (St_ID) REFERENCES Student(ST_ID),
160
161      -- Foreign key linking to Exam table
162      CONSTRAINT FK_Student_Exam_Exam FOREIGN KEY (Ex_ID) REFERENCES Exam(Ex_ID)
163  )
164
```

110 % No issues found

Messages

```
Commands completed successfully.
```

Create Table Questions & Its Constraints

```sql
165  ----------------------------------------------------------------------------
166  -- Table: Questions
167  -- Description: Stores all exam questions for each course, including the question body and correct answer.
168
169  CREATE TABLE Questions (
170      Qs_ID INT IDENTITY(1,1) PRIMARY KEY,
171      Qs_Body VARCHAR(100) NOT NULL,
172      Correct_Answer VARCHAR(60) NOT NULL,
173      Crs_ID INT,                              -- References the course this question belongs to
174
175      -- Define foreign key constraint to maintain referential integrity
176      CONSTRAINT FK_Questions_Course FOREIGN KEY (Crs_ID) REFERENCES Course(Crs_ID)
177  )
```

110 % No issues found

Messages

```
Commands completed successfully.
```

Create Table Choices & Its Constraints

```sql
-----------------------------------------------------------------------------
-- Table: Choices
-- Description: Stores multiple-choice options for each question.
-- Each question can have several options (A, B, C, D, etc.).

CREATE TABLE Choices (
    Qs_ID INT NOT NULL,
    Options_Header CHAR(1) NOT NULL,        -- Option label (e.g., 'A', 'B', 'C', 'D')
    Options_Value VARCHAR(60) NOT NULL,     -- The text or content of the option

    -- Define foreign key relationship to maintain integrity
    CONSTRAINT FK_Choices_Questions FOREIGN KEY (Qs_ID) REFERENCES Questions(Qs_ID)
)
```

110 % 2 0

Messages

Commands completed successfully.

Create Table Stu_Exam_Qs & Its Constraints

```sql
-----------------------------------------------------------------------------
-- Table: Stu_Exam_Qs
-- Description: Bridge table that records each student's answer to each question
-- in a specific exam, along with the grade awarded.

CREATE TABLE Stu_Exam_Qs (
    Stu_ID INT NOT NULL,
    Ex_ID INT NOT NULL,
    Qs_ID INT NOT NULL,
    Grade VARCHAR(20),
    Stu_Answers VARCHAR(100),
    -- Composite primary key ensures one record per student, exam, and question
    CONSTRAINT PK_Student_Exam_Questions PRIMARY KEY (Stu_ID, Ex_ID, Qs_ID),

    -- Foreign key linking to Student table
    CONSTRAINT FK_Student_Exam_Questions_Student FOREIGN KEY (Stu_ID)
        REFERENCES Student(ST_ID),

    -- Foreign key linking to Exam table
    CONSTRAINT FK_Student_Exam_Questions_Exam FOREIGN KEY (Ex_ID)
        REFERENCES Exam(Ex_ID),

    -- Foreign key linking to Questions table
    CONSTRAINT FK_Student_Exam_Questions_Question FOREIGN KEY (Qs_ID)
        REFERENCES Questions(Qs_ID)
)
```

110 % No issues found

Messages

Commands completed successfully.

Create Table Instructor & Its Constraints

```sql
-- ------------------------------------------------------------------------------
-- Table: Instructor
-- Description: Stores personal and professional details of instructors.
-- Includes constraints for data validation and integrity.

CREATE TABLE Instructor (
    Ins_ID INT IDENTITY(1,1) PRIMARY KEY,
    SSN CHAR(14) NOT NULL,
    Ins_FName NVARCHAR(100) NOT NULL,
    Ins_LName NVARCHAR(100) NOT NULL,
    Birth_Date DATE,
    Gender NVARCHAR(10),
    Email NVARCHAR(100),
    Phone VARCHAR(11),
    City NVARCHAR(50),
    Street NVARCHAR(100),
    ZIP_Code NVARCHAR(10),

    -- Sets default salary value if user didn't insert it
    Salary INT CONSTRAINT DF_Instructor_Salary DEFAULT 15000,

    -- Constraints Section

    -- Ensure SSN is unique across all instructors
    CONSTRAINT S_UNQ UNIQUE (SSN),
    -- Ensures  Email is unique across all instructors
    CONSTRAINT E_UNQ UNIQUE (Email),
)
```

110 % No issues found

Messages

```
Commands completed successfully.
```

Create Table Ins_Course & Its Constraints

```sql
-- ------------------------------------------------------------------------------
-- Table: Ins_Course
-- Description: Bridge table that links Instructors and Courses
-- (Many-to-Many relationship).

CREATE TABLE Ins_Course (
    Ins_ID INT NOT NULL,                    -- References Instructor table
    Crs_ID INT NOT NULL,                    -- References Course table
    -- Define constraints
    CONSTRAINT PK_Ins_Course PRIMARY KEY (Crs_ID, Ins_ID),   -- Composite Primary Key
    CONSTRAINT FK_InsCourse_Course FOREIGN KEY (Crs_ID) REFERENCES Course(Crs_ID),
    CONSTRAINT FK_InsCourse_Instructor FOREIGN KEY (Ins_ID) REFERENCES Instructor(Ins_ID)
)
```

110 % No issues found

Messages

```
Commands completed successfully.
```

Create Table Topic

```
259  ----------------------------------------------------------------------------
260  -- Table: Topic
261  -- Description: Stores topics covered in each course.
262  CREATE TABLE Topic (
263      Topic_ID INT IDENTITY(1,1) PRIMARY KEY,
264      Topic_Name NVARCHAR(100) NOT NULL,
265      Manager_Name VARCHAR(30),
266      Salary INT
267      )
     ----------------------------------------------------------------------------
```

110 % — No issues found

Messages

```
Commands completed successfully.
```

Create Table Branch

```
265  ----------------------------------------------------------------------------
266  -- Table: Branch
267  -- Description: Stores branch ID and Branch name.
268
269  CREATE TABLE Branch (
270      Branch_ID INT IDENTITY(1,1) PRIMARY KEY,
271      Branch_Name VARCHAR(50) NOT NULL )
272
273
274
```

110 % — No issues found

Messages

```
Commands completed successfully.
```

Create Table Track & Its Constraints

```
273  ----------------------------------------------------------------------------
274  -- Table: Track
275  -- Description: Stores training tracks, linked to the instructor who manages it
276  -- and the main topic covers it.
277
278  CREATE TABLE Track(
279      Track_ID INT IDENTITY(1,1) PRIMARY KEY,
280      Track_Name VARCHAR(50) NOT NULL,
281      Ins_ID INT,                                  -- References Instructor table
282      Topic_ID INT,                                -- References Topic table
283
284      -- Define constraints
285      CONSTRAINT FK_Track_Instructor FOREIGN KEY (Ins_ID) REFERENCES Instructor(Ins_ID),
286      CONSTRAINT FK_Track_Topic FOREIGN KEY (Topic_ID) REFERENCES Topic(Topic_ID)
287  )
```

110 % — No issues found

Messages

```
Commands completed successfully.
```

Create Table Track_Course & Its Constraints

```sql
-- ------------------------------------------------------------------------------
-- Table: Track_Course
-- Description: Bridge table linking Tracks and Courses
-- (Many-to-Many relationship).

CREATE TABLE Track_Course (
    Track_ID INT NOT NULL,                          -- References Track table
    Crs_ID INT NOT NULL,                            -- References Course table

    -- Define constraints
    CONSTRAINT PK_Track_Course PRIMARY KEY (Track_ID, Crs_ID),   -- Composite Primary Key
    CONSTRAINT FK_TrackCourse_Track FOREIGN KEY (Track_ID) REFERENCES Track(Track_ID),
    CONSTRAINT FK_TrackCourse_Course FOREIGN KEY (Crs_ID) REFERENCES Course(Crs_ID)
)
```

110 % No issues found

Messages

```
Commands completed successfully.
```

Create Table Branch_Track & Its Constraints

```sql
-- ------------------------------------------------------------------------------
-- Table: Branch_Track
-- Description: Bridge table linking Branches and Tracks
-- (Many-to-Many relationship).

CREATE TABLE Branch_Track (
    Track_ID INT NOT NULL,                          -- References Track table
    Branch_ID INT NOT NULL,                         -- References Branch table

    -- Define constraints
    CONSTRAINT PK_Branch_Track PRIMARY KEY (Track_ID, Branch_ID),   -- Composite primary key
    CONSTRAINT FK_Branch_Track_Track FOREIGN KEY (Track_ID) REFERENCES Track(Track_ID),
    CONSTRAINT FK_Branch_Track_Branch FOREIGN KEY (Branch_ID) REFERENCES Branch(Branch_ID)
)
```

110 % No issues found

Messages

```
Commands completed successfully.
```

Create Table Intake

```sql
-- ------------------------------------------------------------------------------
-- Table: Intake
-- Description: Stores information about each intake or batch period.

CREATE TABLE Intake (
    Int_ID INT IDENTITY(1,1) PRIMARY KEY,
    Start_Date DATE,
    End_Date DATE,
    Int_Name VARCHAR(50),
    Round_num INT )
```

110 % No issues found

Messages

```
Commands completed successfully.
```

Create Table Social_Media & Its Constraints

```sql
-- ------------------------------------------------------------------------------
-- Table: Social_Media
-- Description: Stores students' social media profiles and related information.

CREATE TABLE Social_Media (
    Social_ID INT IDENTITY(1,1) PRIMARY KEY,
    Platform_Name VARCHAR(20),
    Profile_Link VARCHAR(200) UNIQUE,
    Photo NVARCHAR(200),
    Stu_ID INT,                                    -- References Student table

    -- Define foreign key constraint
    CONSTRAINT FK_SocialMedia_Student FOREIGN KEY (Stu_ID) REFERENCES Student(St_ID)
)
```

110 % No issues found

Messages

Commands completed successfully.

Create Table Job_Profile & Its Constraints

```sql
-- ------------------------------------------------------------------------------
-- Table: Job_Profile
-- Description: Stores employment details for students,
-- including company, job title, type, salary, and hire date.

CREATE TABLE Job_Profile (
    Job_ID INT IDENTITY(1,1) PRIMARY KEY,
    ST_ID INT,                                     -- References Student table
    Com_ID INT,                                    -- References Company table
    Job_Title NVARCHAR(200),
    Salary INT,
    Job_Type NVARCHAR(50),
    Hire_Date DATE,

    -- Define Constraints
    CONSTRAINT FK_JobProfile_Company FOREIGN KEY (Com_ID) REFERENCES Company(Com_ID),
    CONSTRAINT FK_JobProfile_Student FOREIGN KEY (ST_ID) REFERENCES Student(St_ID)
)
```

110 % No issues found

Messages

Commands completed successfully.

Alter Table Student Add Constraints

```
363  ------------------------------------------------------------------------------
364  -- Table: Student
365  -- Description: Add Contraints
366  ALTER TABLE Student
367  ADD
368      CONSTRAINT UQ_Student_SSN UNIQUE (SSN),    -- Ensures SSN is unique
369      CONSTRAINT CHK_Student_ZipCode CHECK (Zip_Code LIKE '[0-9][0-9][0-9][0-9][0-9]'),
370      CONSTRAINT FK_Student_Project FOREIGN KEY (Pro_ID) REFERENCES Project(Pro_ID),
371      CONSTRAINT FK_Student_Track FOREIGN KEY (Track_ID) REFERENCES Track(Track_ID),
372      CONSTRAINT FK_Student_Intake FOREIGN KEY (Intake_ID) REFERENCES Intake(Int_ID),
373      CONSTRAINT FK_Student_Social FOREIGN KEY (Social_ID) REFERENCES Social_Media(Social_ID),
374      CONSTRAINT FK_Student_Branch FOREIGN KEY (Branch_ID) REFERENCES Branch(Branch_ID),
375      CONSTRAINT FK_Student_JobProfile FOREIGN KEY (job_ID) REFERENCES Job_Profile(Job_ID)
376
377
110 %   1   0
Messages
    Commands completed successfully.
```

Data Generation and Preparation

The dataset used in this project was initially generated using AI-based data generation tools such as Mockaroo, which allowed us to create realistic and diverse records that reflect real-world scenarios. After generation, the data was reviewed, refined, and upgraded manually to ensure accuracy, completeness, and logical consistency across all tables. This combination of automated generation and manual enhancement provided a reliable dataset for building, testing, and analyzing the entire examination system.

An overview of our data

Attribute	Data Size
Students	985
colleges	14
universites	38
Branches	22
Certificates	50
Choises	900
Companies	17
Courses	72
exams	15
freelance	985
Instructors	50
Ins_courses	110
Intakes	6
Jobs	585
Questions	335
Student_certeficate	1,471
Student_exam	380
Topics	9
Tracks	59
Branch_tracks	164
Track_course	1,111
Stu_exam_Qs	1,410

Student:

```
SELECT COUNT(*) AS 'NUMBER OF RAWS' FROM Student
SELECT * FROM Student
```

No issues found

NUMBER OF RAWS
985

ST_id	SSN	Fname	Lname	Gender	Birth_Date	City	Street	ZipCode	Phone	Grade_Year	Pro_ID	Track_ID	Intake_ID	Social_ID	Branch_ID	College	University	Age
1	3040914022929	Yousef	Ashraf	M	2004-09-14	Tanta	Saeed Street	31737	01032890313	2023	12	12	6	9	19	Faculty of Computers and Artificial Intelligence	Cairo University	21
2	3010924087572	Yousef	Sabal	M	2001-09-24	Assiut	El Gomhoureya Street	71511	01252333743	2022	47	47	2	1	8	Faculty of Computers and Artificial Intelligence	German University in Cairo (GUC)	24
3	3021125264784	Ahmed	Medhat	M	2002-11-25	Suez	El Geish Street	43518	01130440268	2023	41	41	4	12	11	Faculty of Media & Mass Comm	Aswan University	23
4	2980218146689	Raafat	Elrais	M	1998-02-18	Tanta	El Bahr Street	31511	01227759689	2019	26	26	1	7	7	Faculty of Media & Mass Comm	Future University in Egypt	27
5	2990519096552	Omar	Zahran	M	1999-05-19	Minya	Port Said Street	61519	01599350687	2020	46	46	6	8	1	Faculty of Computer & Information Technology	Arab Academy for Science and Technology	26
6	3030310054470	Tamer	Mostafa	M	2003-03-10	Mansoura	Suez Canal Street	35511	01138420214	2024	4	4	3	NULL	6	Faculty of Computers and Artificial Intelligence	Cairo University	22
7	3000320136746	Aya	Mahmoud	F	2000-03-20	Alexandria	Gamal Abdel Nasser Street	21526	01165750083	2021	47	47	4	NULL	16	Faculty of Engineering	Suez Canal University	25
8	3010103239169	Nour	Mahmoud	F	2001-01-03	Cairo	El Mirghani Street	11341	01015498473	2022	1	1	3	NULL	6	Faculty of Business	Cairo University	24
9	3000508129753	Sara	Hussein	F	2000-05-08	Tanta	El Nahas Street	31511	01533080274	2021	4	4	5	NULL	5	Faculty of Media & Mass Comm	Future University in Egypt	25
10	2980608039327	Karim	Fahmy	M	1998-06-08	Mansoura	El Gomhoureya Street	35511	01290378326	2019	26	26	4	NULL	1	Faculty of Computers and Artificial Intelligence	Ain Shams University	27

Instructor:

```
SELECT COUNT(*) AS 'NUMBER OF RAWS' FROM Instructor
SELECT * FROM Instructor
```

No issues found

NUMBER OF RAWS
50

Ins_ID	SSN	Ins_FName	Ins_LName	Birth_Date	Gender	Email	Phone	City	Street	ZIP_Code	Salary	Evaluation_Instructor
1	28501100100123	Ahmed	Ali	1985-01-10	Male	ahmed.ali@gmail.com	01006599897	Giza	15 Al Haram St.	12559	28500	Good
2	29203151600456	Fatma	Ibrahim	1992-03-15	Female	fatma.ibrahim@yahoo.com	01007486849	Cairo	45 Ramses St.	11511	26000	Very Good
3	28807200200789	Mahmoud	Said	1988-07-20	Male	mahmoud.said@outlook.com	01032188777	Alexandria	10 Corniche Rd.	21526	27000	Good
4	29405011600123	Mariam	Taha	1994-05-01	Female	mariam.taha@gmail.com	01035858577	Cairo	90 Tahrir St.	11518	24500	Good
5	28311250100321	Mostafa	Kamal	1983-11-25	Male	mostafa.kamal@yahoo.com	01055374569	Giza	8 Nile St.	12211	31000	Very Good
6	29102201600124	Nour	Hassan	1991-02-20	Female	nour.hassan@outlook.com	01056074268	Mansoura	3 El Geish St.	35511	25000	Good
7	28608150100451	Omar	Abdelrahman	1986-08-15	Male	omar.abdelrahman@gmail.com	01070334062	Cairo	22 Abbas El Akkad	11765	29000	Good
8	29006050200553	Youssef	Khaled	1990-06-05	Male	youssef.khaled@yahoo.com	01077069668	Alexandria	5 Safia Zaghloul St.	21521	28000	Very Good
9	29304121600876	Aya	Mahmoud	1993-04-12	Female	aya.mahmoud@gmail.com	01080880522	Cairo	112 El Merghany St.	11341	27500	Good
10	28209302100331	Ali	Adel	1982-09-30	Male	ali.adel@outlook.com	01081452950	Giza	3 Faisal St.	12111	32000	Good

Course:

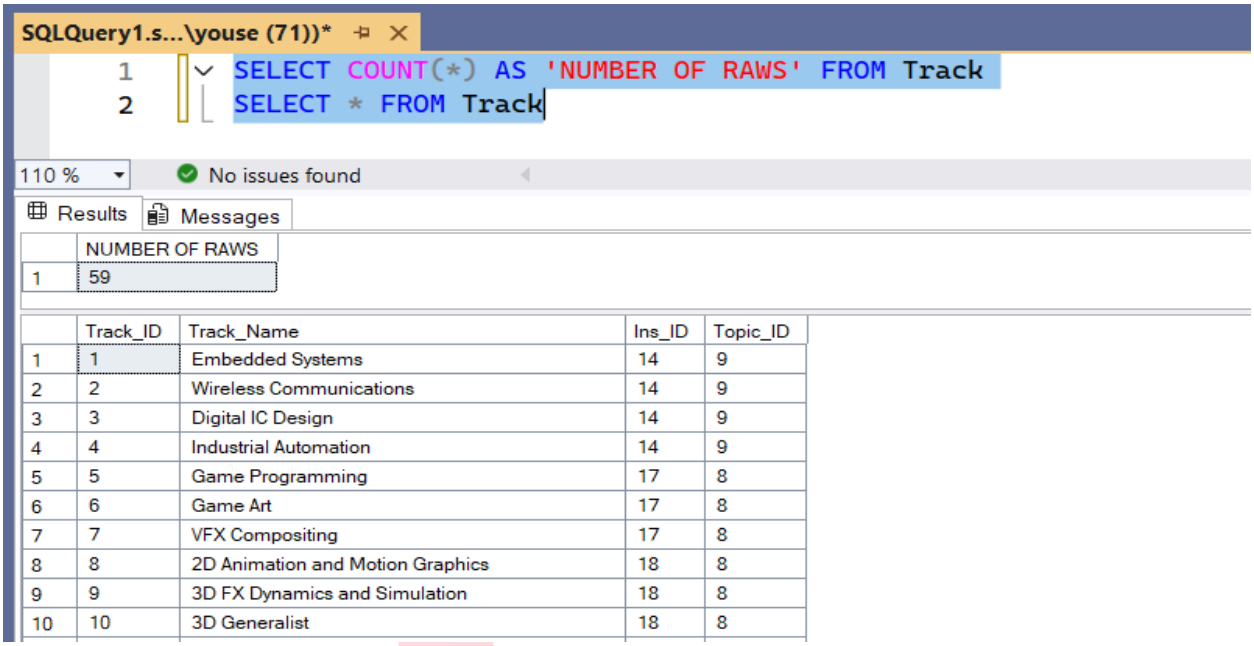
```
SELECT COUNT(*) AS 'NUMBER OF RAWS' FROM Course
SELECT * FROM Course
```

No issues found

NUMBER OF RAWS
72

Crs_ID	Crs_Name	Hours
1	Database	60
2	Operating System	12
3	Python Fundamentals	30
4	Software Testing	18
5	Power BI	36
6	Excel	12
7	SQL BI	18
8	Source Control	12
9	Data Warehouse	18
10	No SQL	18

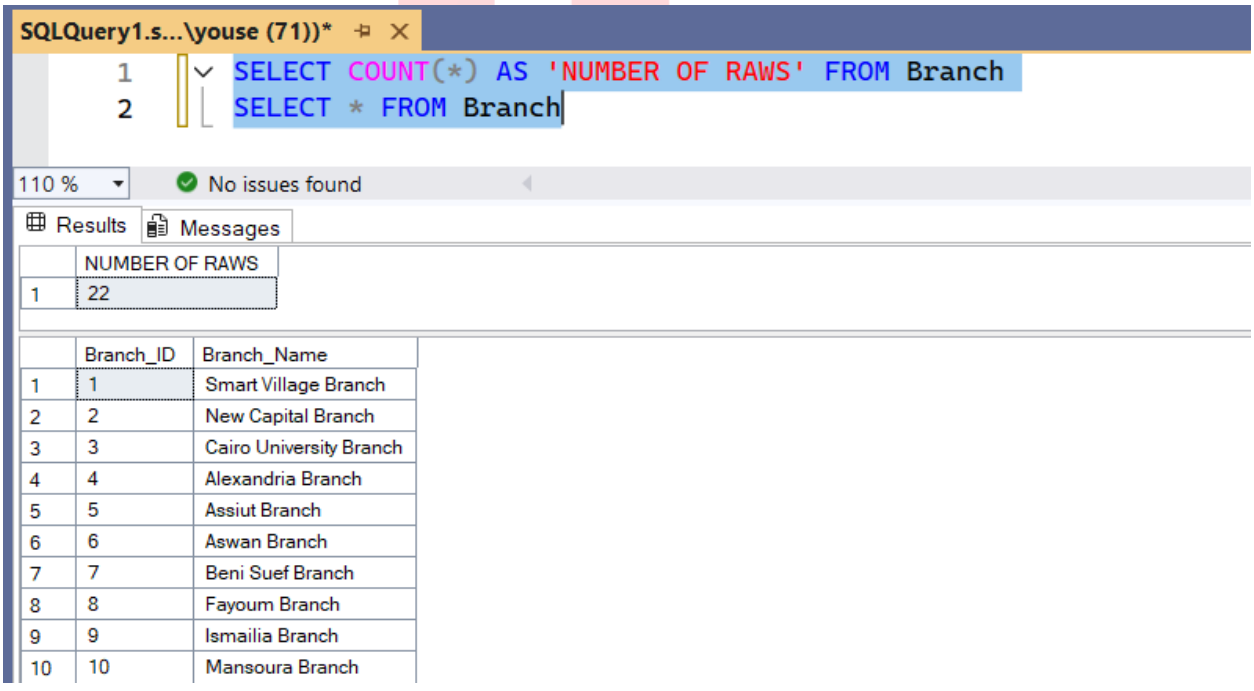
Track:



	NUMBER OF RAWS
1	59

	Track_ID	Track_Name	Ins_ID	Topic_ID
1	1	Embedded Systems	14	9
2	2	Wireless Communications	14	9
3	3	Digital IC Design	14	9
4	4	Industrial Automation	14	9
5	5	Game Programming	17	8
6	6	Game Art	17	8
7	7	VFX Compositing	17	8
8	8	2D Animation and Motion Graphics	18	8
9	9	3D FX Dynamics and Simulation	18	8
10	10	3D Generalist	18	8

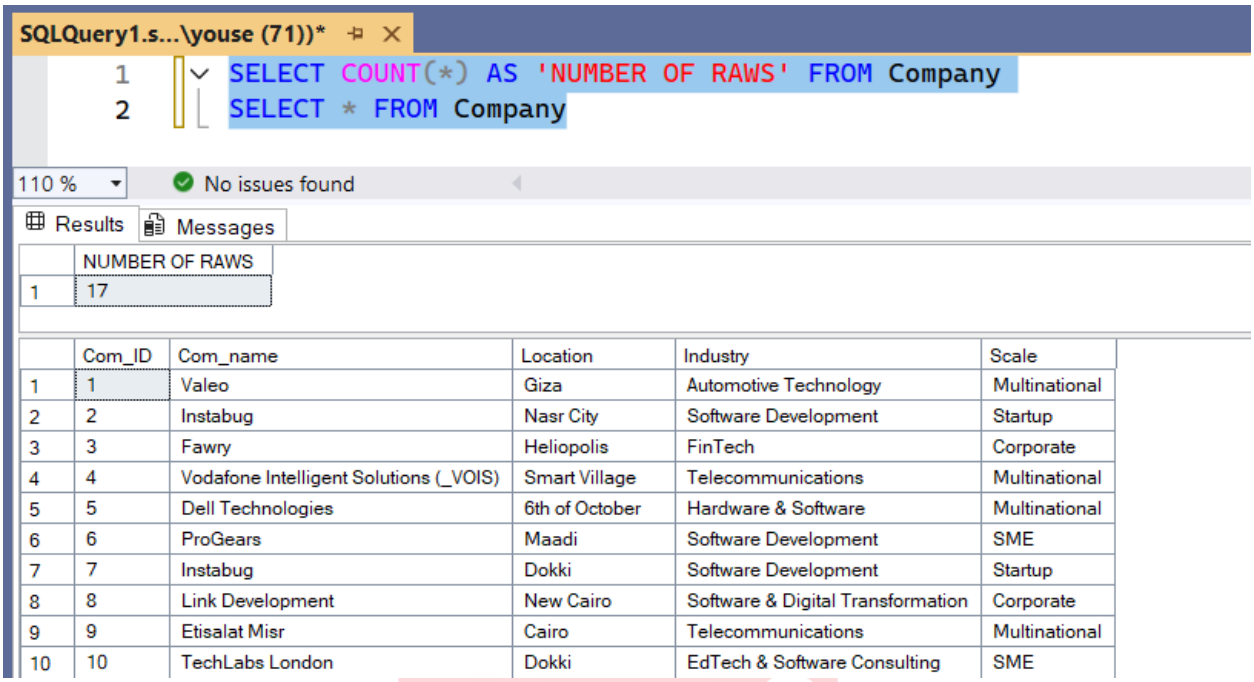
Branch:



	NUMBER OF RAWS
1	22

	Branch_ID	Branch_Name
1	1	Smart Village Branch
2	2	New Capital Branch
3	3	Cairo University Branch
4	4	Alexandria Branch
5	5	Assiut Branch
6	6	Aswan Branch
7	7	Beni Suef Branch
8	8	Fayoum Branch
9	9	Ismailia Branch
10	10	Mansoura Branch

Company:



```
SELECT COUNT(*) AS 'NUMBER OF RAWS' FROM Company
SELECT * FROM Company
```

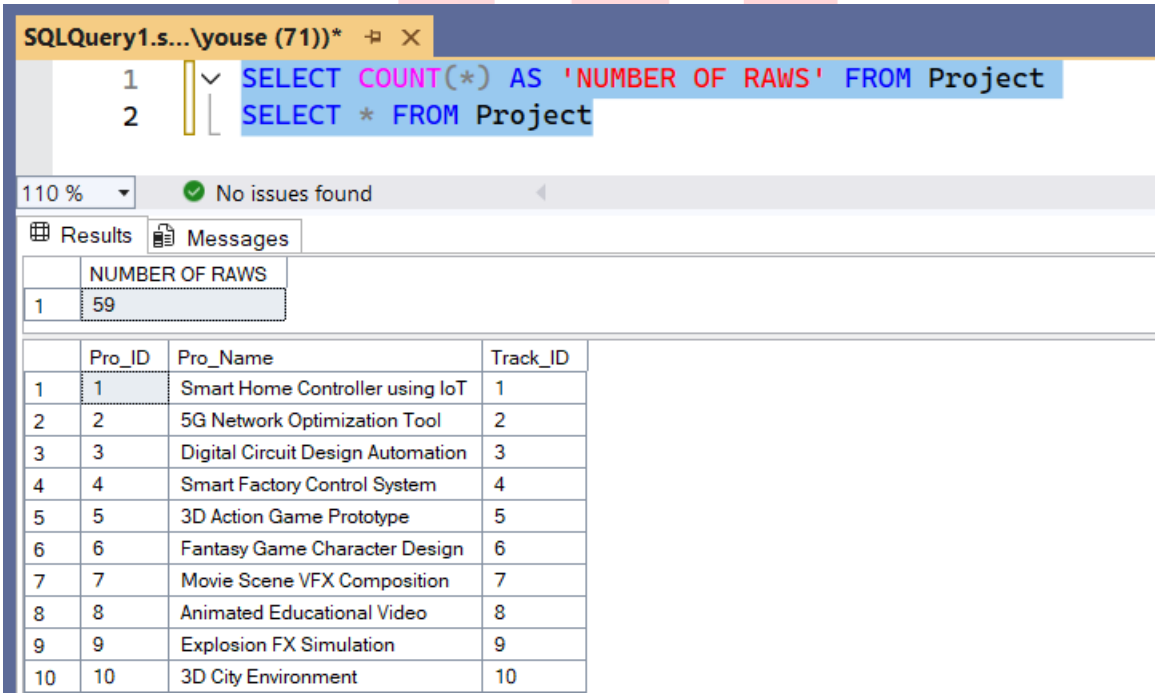
110 % — No issues found

Results | Messages

	NUMBER OF RAWS
1	17

	Com_ID	Com_name	Location	Industry	Scale
1	1	Valeo	Giza	Automotive Technology	Multinational
2	2	Instabug	Nasr City	Software Development	Startup
3	3	Fawry	Heliopolis	FinTech	Corporate
4	4	Vodafone Intelligent Solutions (_VOIS)	Smart Village	Telecommunications	Multinational
5	5	Dell Technologies	6th of October	Hardware & Software	Multinational
6	6	ProGears	Maadi	Software Development	SME
7	7	Instabug	Dokki	Software Development	Startup
8	8	Link Development	New Cairo	Software & Digital Transformation	Corporate
9	9	Etisalat Misr	Cairo	Telecommunications	Multinational
10	10	TechLabs London	Dokki	EdTech & Software Consulting	SME

Project:



```
SELECT COUNT(*) AS 'NUMBER OF RAWS' FROM Project
SELECT * FROM Project
```

110 % — No issues found

Results | Messages

	NUMBER OF RAWS
1	59

	Pro_ID	Pro_Name	Track_ID
1	1	Smart Home Controller using IoT	1
2	2	5G Network Optimization Tool	2
3	3	Digital Circuit Design Automation	3
4	4	Smart Factory Control System	4
5	5	3D Action Game Prototype	5
6	6	Fantasy Game Character Design	6
7	7	Movie Scene VFX Composition	7
8	8	Animated Educational Video	8
9	9	Explosion FX Simulation	9
10	10	3D City Environment	10

Freelance:

SQLQuery1.s...\youse (71))*

```
SELECT COUNT(*) AS 'NUMBER OF RAWS' FROM Freelance
SELECT * FROM Freelance
```

110 % — No issues found

Results | Messages

	NUMBER OF RAWS
1	985

	Fre_ID	Website_name	Date	Budget	Job_description	ST_ID
1	1	ServiceScape	2023-04-25	45	Set up CI/CD pipeline using GitHub Actions and deplo...	460
2	2	Mostaql	2025-01-24	124	Run a Facebook ad campaign setup with audience ta...	679
3	3	Upwork	2024-08-27	452	Create an interactive Node.js dashboard to track mont...	969
4	4	We Work Remotely	2024-09-08	47	Develop a recommendation system using Python and...	830
5	5	SimplyHired	2024-07-12	449	Convert design files (Figma) to a pixel-perfect respon...	946
6	6	Wellfound	2025-06-09	26	Dockerize the application and create Kubernetes ma...	680
7	7	Linkedin	2023-10-12	179	Create a full-stack web application with React fronten...	689
8	8	Khamsat	2024-01-03	887	Create a native Android app with Kotlin for booking ap...	89
9	9	Fiver	2025-01-17	490	Develop a recommendation system using Python and...	86
10	10	Upwork	2024-01-30	410	Create a prototype computer vision model to detect d...	542

Certificate:

SQLQuery1.s...\youse (71))*

```
SELECT COUNT(*) AS 'NUMBER OF RAWS' FROM Certificate
SELECT * FROM Certificate
```

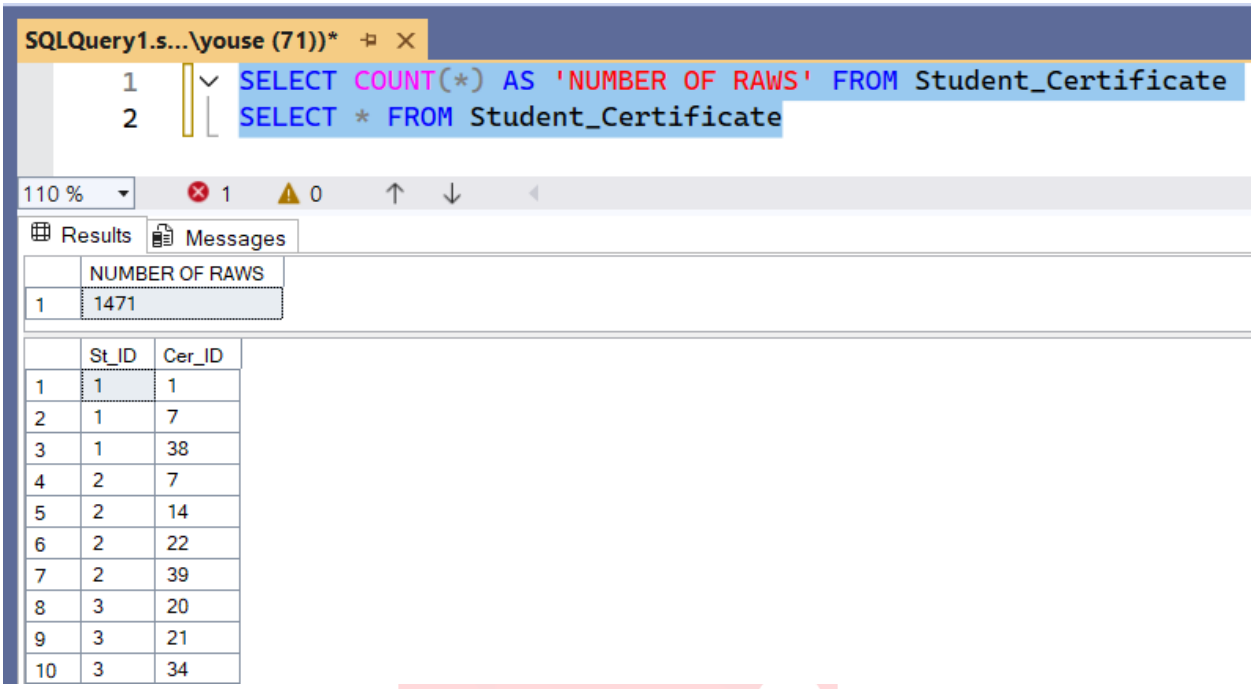
110 % — No issues found

Results | Messages

	NUMBER OF RAWS
1	50

	Cer_ID	Name	Hours	Provider
1	1	Cleaning Data in Python	12	DataCamp
2	2	Cleaning Data in R	12	DataCamp
3	3	Data Manipulation in R with dplyr	14	DataCamp
4	4	Cleaning Data with PySpark	13	DataCamp
5	5	Data Manipulation in SQL	15	DataCamp
6	6	Intermediate SQL	15	DataCamp
7	7	Introduction to SQL	13	DataCamp
8	8	Database Design	17	DataCamp
9	9	Cleaning Data in PostgreSQL Databases	14	DataCamp
10	10	Importing & Cleaning Data in R (Track)	16	DataCamp

Student_certeficate:



```
SQLQuery1.s...\youse (71))*
1  SELECT COUNT(*) AS 'NUMBER OF RAWS' FROM Student_Certificate
2  SELECT * FROM Student_Certificate
```

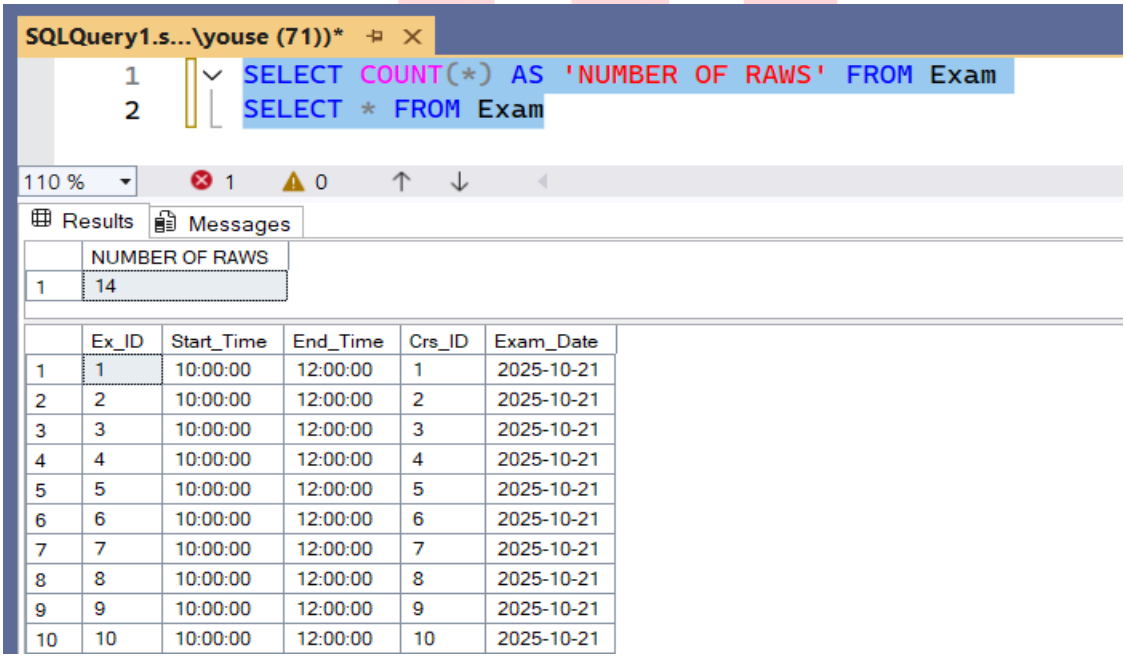
110 % 1 0

Results Messages

	NUMBER OF RAWS
1	1471

	St_ID	Cer_ID
1	1	1
2	1	7
3	1	38
4	2	7
5	2	14
6	2	22
7	2	39
8	3	20
9	3	21
10	3	34

Exam:



```
SQLQuery1.s...\youse (71))*
1  SELECT COUNT(*) AS 'NUMBER OF RAWS' FROM Exam
2  SELECT * FROM Exam
```

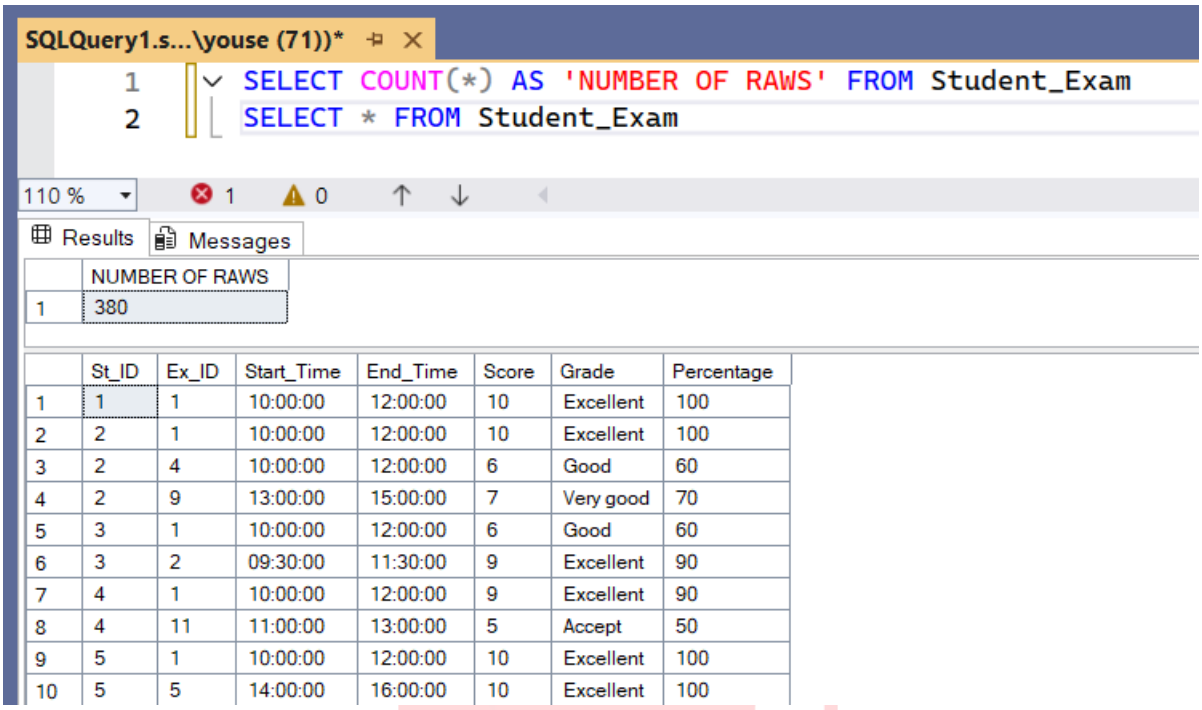
110 % 1 0

Results Messages

	NUMBER OF RAWS
1	14

	Ex_ID	Start_Time	End_Time	Crs_ID	Exam_Date
1	1	10:00:00	12:00:00	1	2025-10-21
2	2	10:00:00	12:00:00	2	2025-10-21
3	3	10:00:00	12:00:00	3	2025-10-21
4	4	10:00:00	12:00:00	4	2025-10-21
5	5	10:00:00	12:00:00	5	2025-10-21
6	6	10:00:00	12:00:00	6	2025-10-21
7	7	10:00:00	12:00:00	7	2025-10-21
8	8	10:00:00	12:00:00	8	2025-10-21
9	9	10:00:00	12:00:00	9	2025-10-21
10	10	10:00:00	12:00:00	10	2025-10-21

Student_Exam:



```
SQLQuery1.s...\youse (71))*
1  SELECT COUNT(*) AS 'NUMBER OF RAWS' FROM Student_Exam
2  SELECT * FROM Student_Exam
```

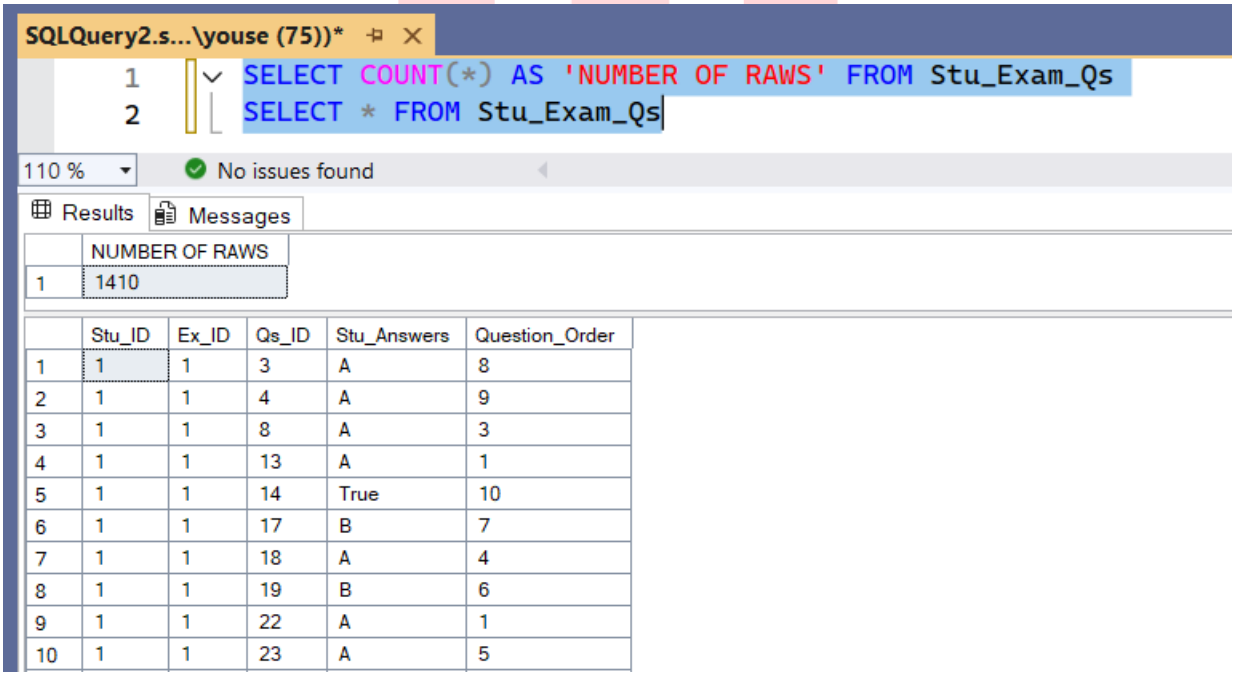
110 % | 1 | 0

Results | Messages

	NUMBER OF RAWS
1	380

	St_ID	Ex_ID	Start_Time	End_Time	Score	Grade	Percentage
1	1	1	10:00:00	12:00:00	10	Excellent	100
2	2	1	10:00:00	12:00:00	10	Excellent	100
3	2	4	10:00:00	12:00:00	6	Good	60
4	2	9	13:00:00	15:00:00	7	Very good	70
5	3	1	10:00:00	12:00:00	6	Good	60
6	3	2	09:30:00	11:30:00	9	Excellent	90
7	4	1	10:00:00	12:00:00	9	Excellent	90
8	4	11	11:00:00	13:00:00	5	Accept	50
9	5	1	10:00:00	12:00:00	10	Excellent	100
10	5	5	14:00:00	16:00:00	10	Excellent	100

Stu_Exam_Qs:



```
SQLQuery2.s...\youse (75))*
1  SELECT COUNT(*) AS 'NUMBER OF RAWS' FROM Stu_Exam_Qs
2  SELECT * FROM Stu_Exam_Qs
```

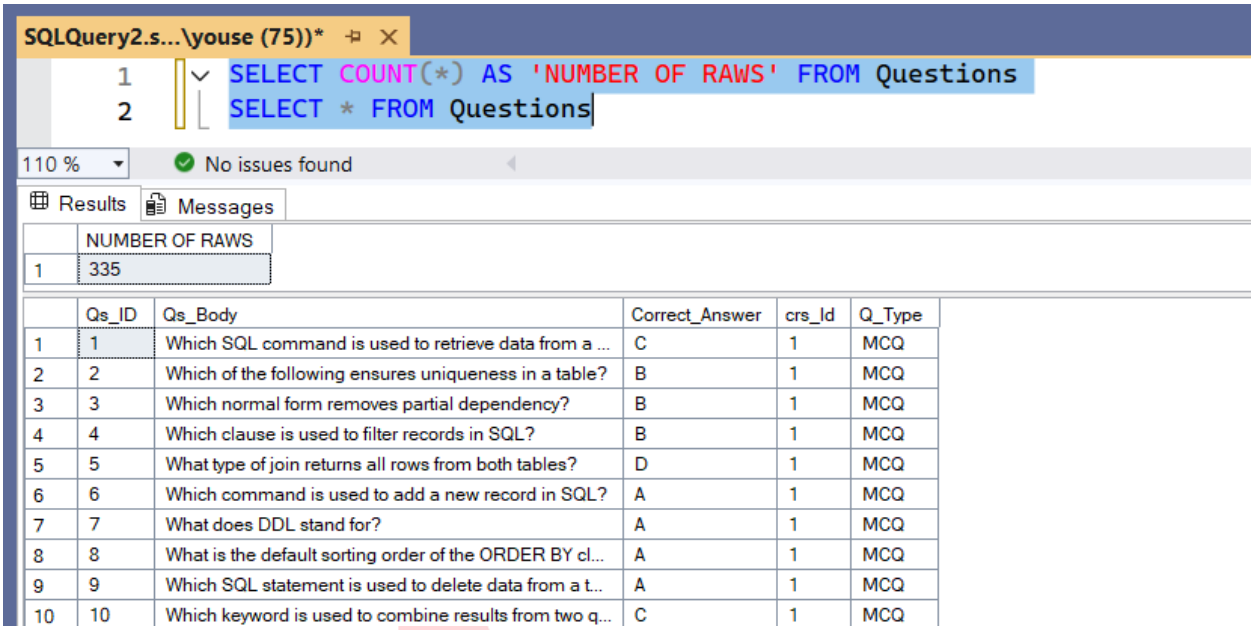
110 % | No issues found

Results | Messages

	NUMBER OF RAWS
1	1410

	Stu_ID	Ex_ID	Qs_ID	Stu_Answers	Question_Order
1	1	1	3	A	8
2	1	1	4	A	9
3	1	1	8	A	3
4	1	1	13	A	1
5	1	1	14	True	10
6	1	1	17	B	7
7	1	1	18	A	4
8	1	1	19	B	6
9	1	1	22	A	1
10	1	1	23	A	5

Questions:



```
SELECT COUNT(*) AS 'NUMBER OF RAWS' FROM Questions
SELECT * FROM Questions
```

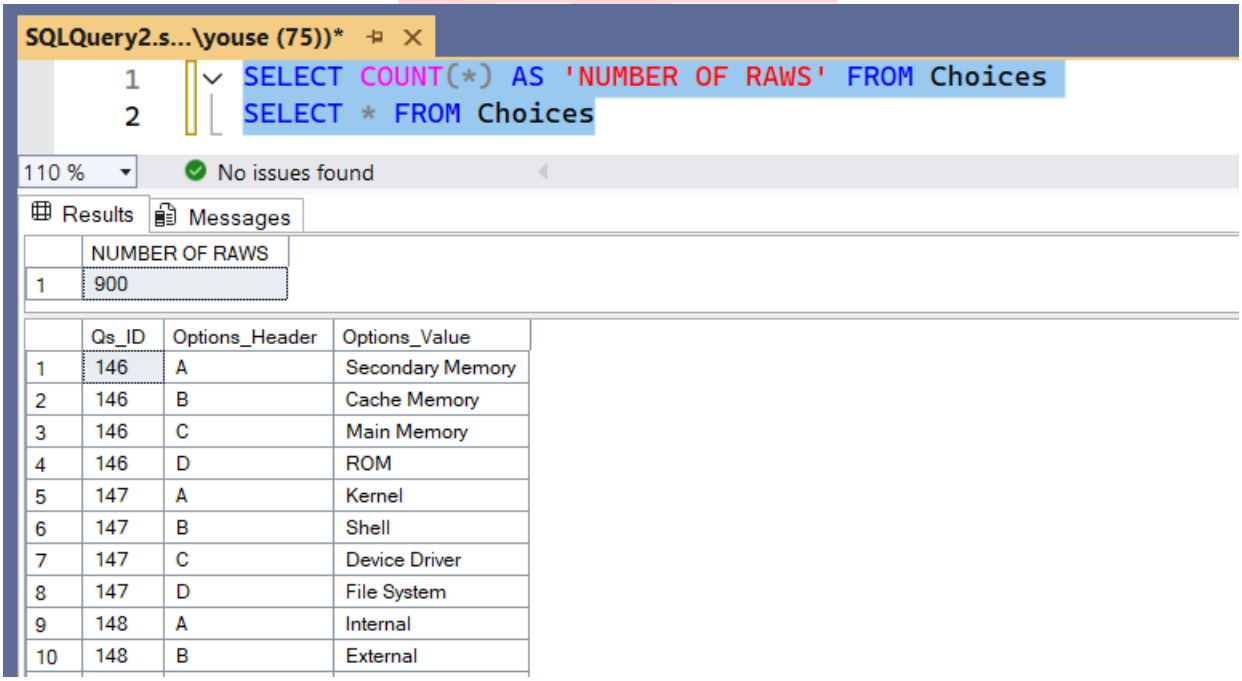
No issues found

Results | Messages

	NUMBER OF RAWS
1	335

	Qs_ID	Qs_Body	Correct_Answer	crs_Id	Q_Type
1	1	Which SQL command is used to retrieve data from a ...	C	1	MCQ
2	2	Which of the following ensures uniqueness in a table?	B	1	MCQ
3	3	Which normal form removes partial dependency?	B	1	MCQ
4	4	Which clause is used to filter records in SQL?	B	1	MCQ
5	5	What type of join returns all rows from both tables?	D	1	MCQ
6	6	Which command is used to add a new record in SQL?	A	1	MCQ
7	7	What does DDL stand for?	A	1	MCQ
8	8	What is the default sorting order of the ORDER BY cl...	A	1	MCQ
9	9	Which SQL statement is used to delete data from a t...	A	1	MCQ
10	10	Which keyword is used to combine results from two q...	C	1	MCQ

Choices:



```
SELECT COUNT(*) AS 'NUMBER OF RAWS' FROM Choices
SELECT * FROM Choices
```

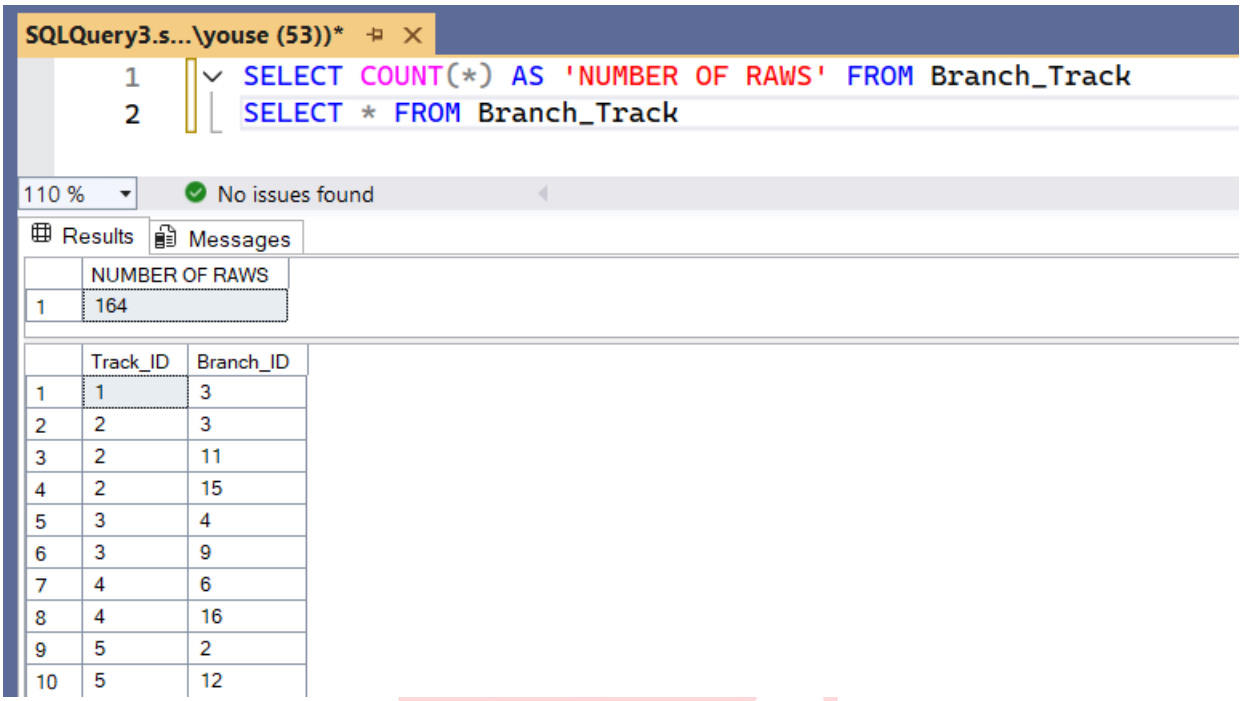
No issues found

Results | Messages

	NUMBER OF RAWS
1	900

	Qs_ID	Options_Header	Options_Value
1	146	A	Secondary Memory
2	146	B	Cache Memory
3	146	C	Main Memory
4	146	D	ROM
5	147	A	Kernel
6	147	B	Shell
7	147	C	Device Driver
8	147	D	File System
9	148	A	Internal
10	148	B	External

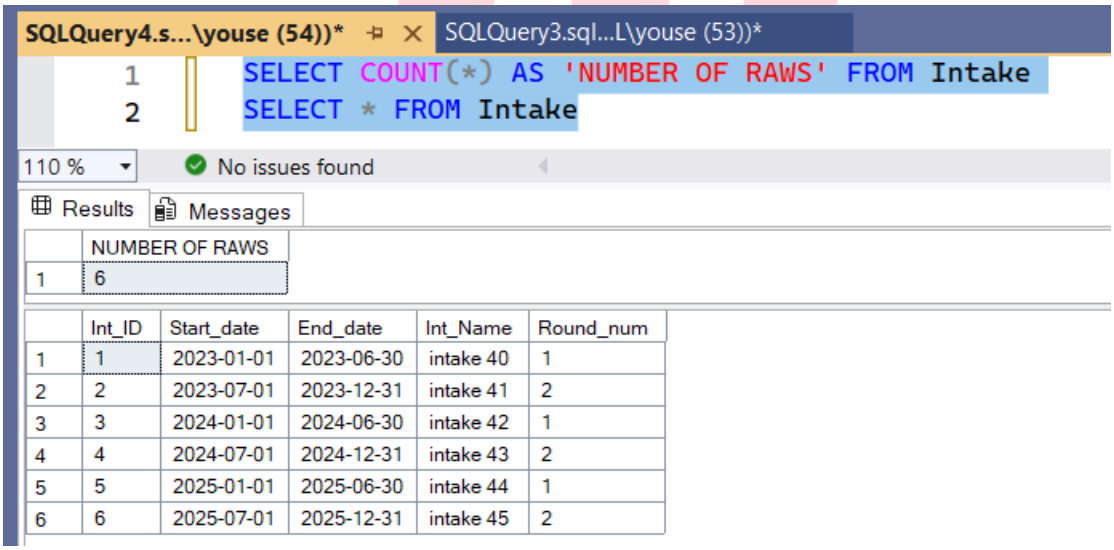
Branch_Track:



	NUMBER OF RAWS
1	164

	Track_ID	Branch_ID
1	1	3
2	2	3
3	2	11
4	2	15
5	3	4
6	3	9
7	4	6
8	4	16
9	5	2
10	5	12

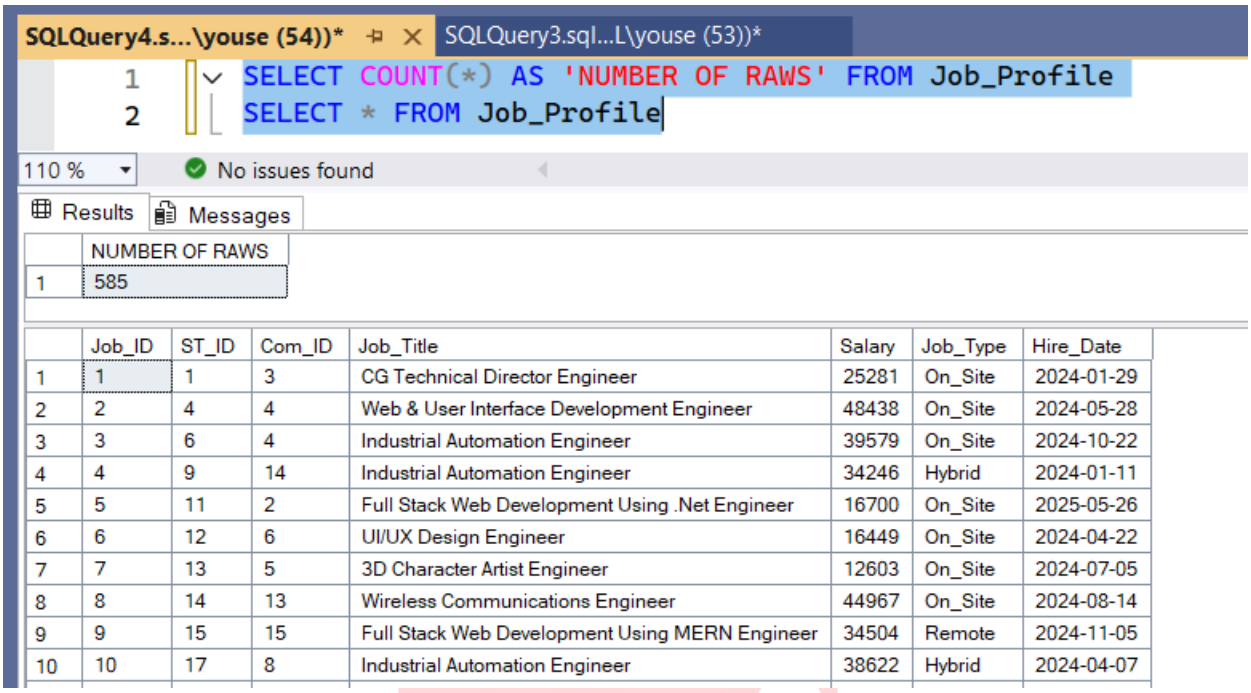
Intake:



	NUMBER OF RAWS
1	6

	Int_ID	Start_date	End_date	Int_Name	Round_num
1	1	2023-01-01	2023-06-30	intake 40	1
2	2	2023-07-01	2023-12-31	intake 41	2
3	3	2024-01-01	2024-06-30	intake 42	1
4	4	2024-07-01	2024-12-31	intake 43	2
5	5	2025-01-01	2025-06-30	intake 44	1
6	6	2025-07-01	2025-12-31	intake 45	2

Job_Profile:



SQLQuery4.s...\youse (54))* | SQLQuery3.sql...L\youse (53))*

```
SELECT COUNT(*) AS 'NUMBER OF RAWS' FROM Job_Profile
SELECT * FROM Job_Profile
```

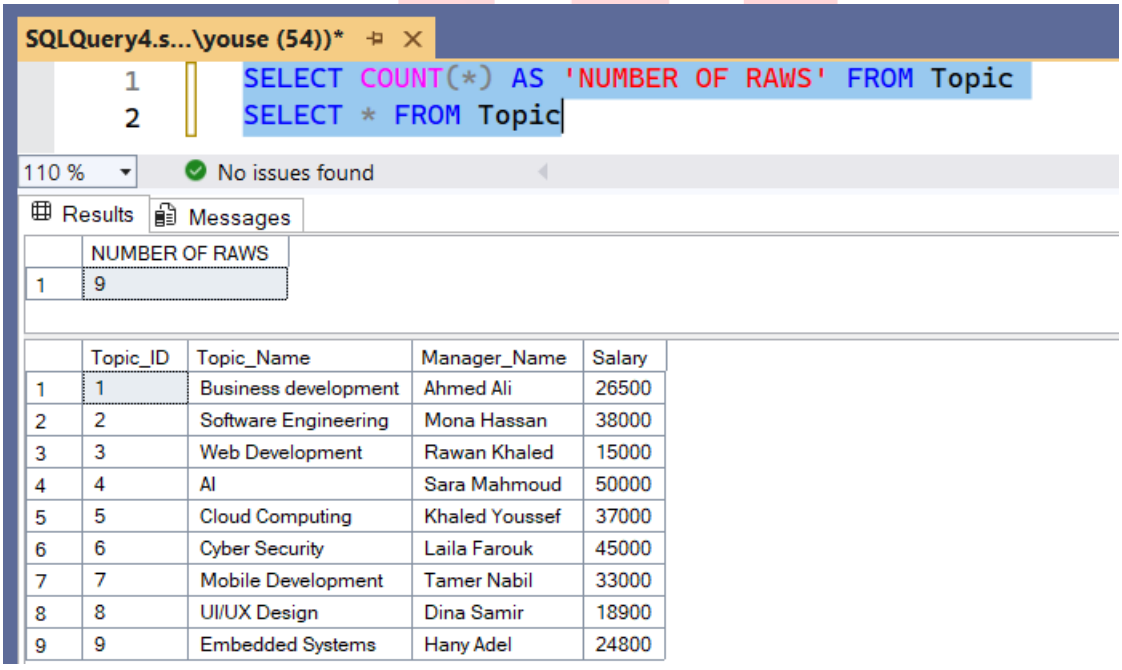
110 % — No issues found

Results | Messages

	NUMBER OF RAWS
1	585

	Job_ID	ST_ID	Com_ID	Job_Title	Salary	Job_Type	Hire_Date
1	1	1	3	CG Technical Director Engineer	25281	On_Site	2024-01-29
2	2	4	4	Web & User Interface Development Engineer	48438	On_Site	2024-05-28
3	3	6	4	Industrial Automation Engineer	39579	On_Site	2024-10-22
4	4	9	14	Industrial Automation Engineer	34246	Hybrid	2024-01-11
5	5	11	2	Full Stack Web Development Using .Net Engineer	16700	On_Site	2025-05-26
6	6	12	6	UI/UX Design Engineer	16449	On_Site	2024-04-22
7	7	13	5	3D Character Artist Engineer	12603	On_Site	2024-07-05
8	8	14	13	Wireless Communications Engineer	44967	On_Site	2024-08-14
9	9	15	15	Full Stack Web Development Using MERN Engineer	34504	Remote	2024-11-05
10	10	17	8	Industrial Automation Engineer	38622	Hybrid	2024-04-07

Topic:



SQLQuery4.s...\youse (54))*

```
SELECT COUNT(*) AS 'NUMBER OF RAWS' FROM Topic
SELECT * FROM Topic
```

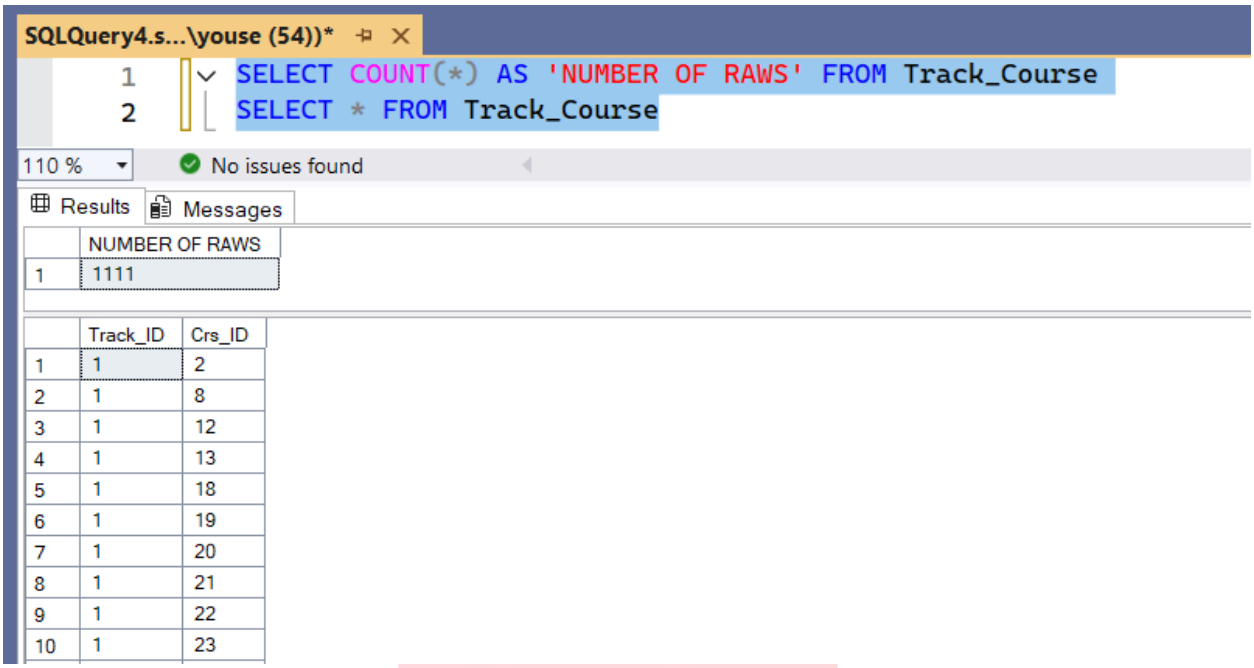
110 % — No issues found

Results | Messages

	NUMBER OF RAWS
1	9

	Topic_ID	Topic_Name	Manager_Name	Salary
1	1	Business development	Ahmed Ali	26500
2	2	Software Engineering	Mona Hassan	38000
3	3	Web Development	Rawan Khaled	15000
4	4	AI	Sara Mahmoud	50000
5	5	Cloud Computing	Khaled Youssef	37000
6	6	Cyber Security	Laila Farouk	45000
7	7	Mobile Development	Tamer Nabil	33000
8	8	UI/UX Design	Dina Samir	18900
9	9	Embedded Systems	Hany Adel	24800

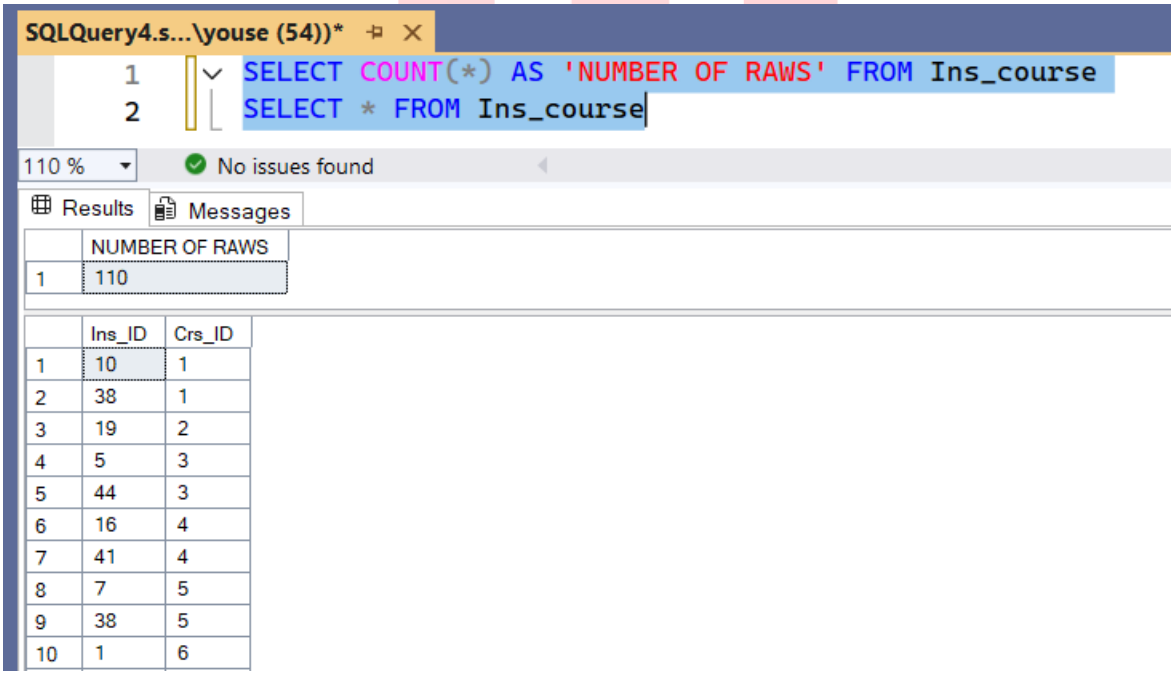
Track_Course:



	NUMBER OF RAWS
1	1111

	Track_ID	Crs_ID
1	1	2
2	1	8
3	1	12
4	1	13
5	1	18
6	1	19
7	1	20
8	1	21
9	1	22
10	1	23

Ins_Course:



	NUMBER OF RAWS
1	110

	Ins_ID	Crs_ID
1	10	1
2	38	1
3	19	2
4	5	3
5	44	3
6	16	4
7	41	4
8	7	5
9	38	5
10	1	6

Social_Media:

SQLQuery4.s...\youse (54))*

```
1  SELECT COUNT(*) AS 'NUMBER OF RAWS' FROM Social_Media
2  SELECT * FROM Social_Media
```

110 % — No issues found

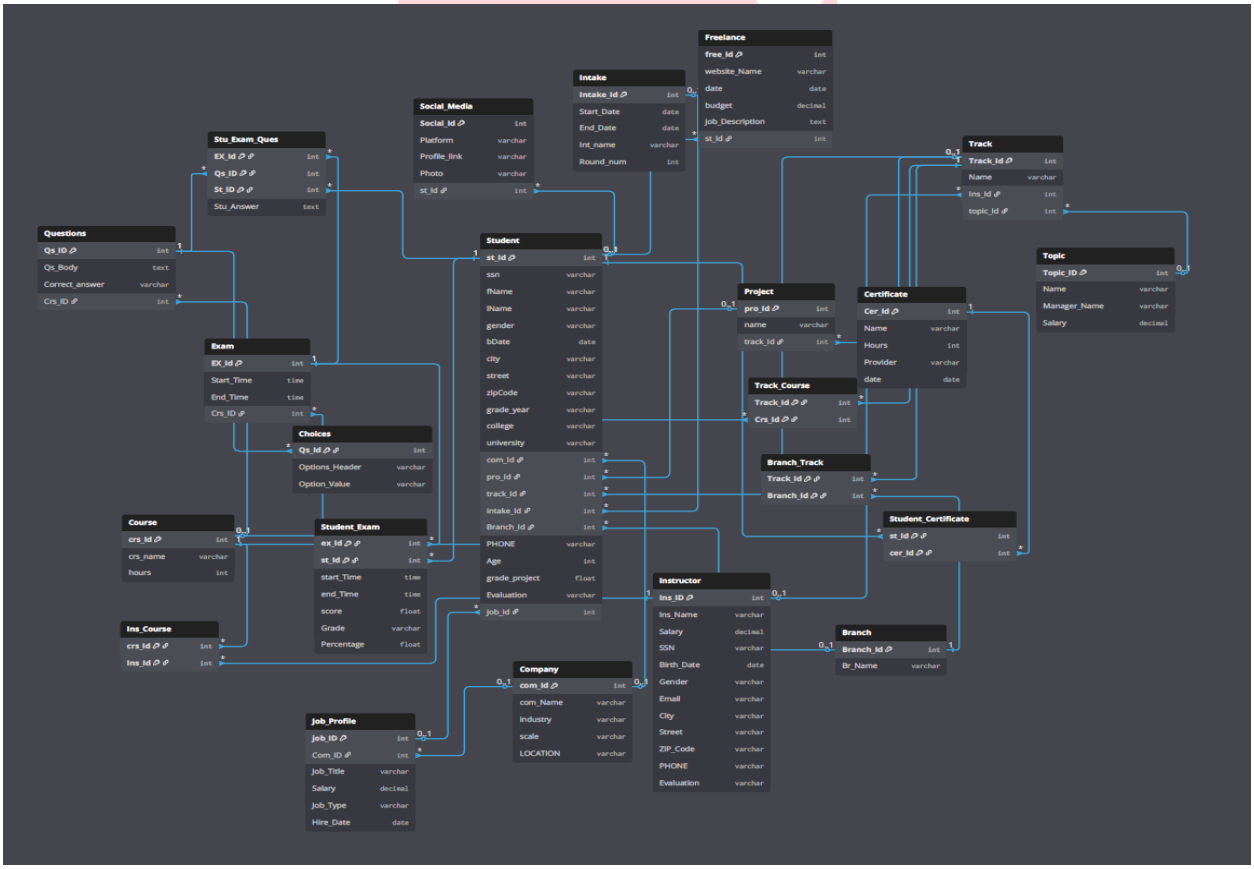
Results | Messages

	NUMBER OF RAWS
1	5

	Social_ID	Platform_Name	Profile_Link	Photo	Stu_ID
1	1	Facebook	https://www.facebook.com/profile.php?id=100093847...	https://example.com/images/profile1.jpg	2
2	7	Facebook	https://www.facebook.com/rafat.elrays	https://www.facebook.com/photo/?fbid=258125100531...	4
3	8	Facebook	https://www.facebook.com/OmarZahran72	https://www.facebook.com/photo/?fbid=167581606303...	5
4	9	Facebook	https://www.facebook.com/share/1DoLGrYV9F/?mibe...	https://www.facebook.com/photo?fbid=299240043094...	1
5	12	Facebook	https://www.facebook.com/share/1D9L8YPWHg/	NULL	3

Database Diagram

We used dbdiagram.io to design and organize our Entity-Relationship Diagram (ERD), ensuring that all database relationships were clearly structured and visually well-organized before implementation in SQL Server.



Stored Procedures

In this stage, we developed a comprehensive set of 63 stored procedures in SQL Server to manage and automate database operations efficiently. These procedures handle all core actions across the system's tables, including SELECT, INSERT, UPDATE, and DELETE statements.

The use of stored procedures ensured data consistency, reusability, and enhanced security, while also improving performance by reducing repetitive code. Each procedure was carefully designed to manage specific entities such as Students, Instructors, Courses and Exams.

Table Name	Insert	Update	Delete	Select
Student	☑	☑	☑	✗
Company	☑	☑	☑	✗
Project	☑	☑	☑	✗
Freelance	☑	☑	☑	✗
Course	☑	☑	☑	✗
Student_Certificate	☑	☑	✗	✗
Student_Exam	☑	☑	☑	✗
Stu_Exam_Qs	☑	☑	☑	✗
Questions	☑	☑	☑	✗
Choices	☑	☑	☑	✗
Ins_Course	☑	☑	☑	☑
Track_Course	☑	✗	☑	☑
Branch_Track	☑	☑	☑	✗
Track	☑	☑	☑	✗
Intake	☑	✗	☑	✗
Certificate	☑	☑	☑	✗
Social_Media	☑	☑	☑	✗
Branch	☑	☑	☑	☑
Instructor	☑	☑	☑	✗
Topic	☑	☑	☑	✗
Job_Profile	☑	☑	☑	✗

Exam procedures

Add Exam

```sql
CREATE PROCEDURE [dbo].[Add_Exam]
    @Start_Time TIME,
    @End_Time TIME,
    @Crs_ID INT
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @Today DATETIME = CAST(GETDATE() AS DATE);

    INSERT INTO Exam ( Start_Time, End_Time, Crs_ID,Exam_Date)
    VALUES ( @Start_Time, @End_Time, @Crs_ID,@Today);

    DECLARE @New_Ex_ID INT = SCOPE_IDENTITY();
    PRINT 'Exam Added Successfully. Exam ID: ' + CAST(@New_Ex_ID AS NVARCHAR(10));
END;

GO
```

Register Student

```sql
CREATE   PROCEDURE [dbo].[Register_Student] @St_ID INT , @Ex_ID INT
AS
BEGIN
    Set Nocount on;
    If Not Exists (Select 1 from Student_Exam where St_ID = @St_ID and Ex_ID = @Ex_ID)
    begin
    insert into Student_Exam (St_ID , Ex_ID , Start_Time , End_Time)
    select @St_ID , @Ex_ID , Start_time , End_time
    From Exam
    where Ex_ID = @Ex_ID
        Print 'Student Registered For Exam'
    end
    else
    begin
        Print 'Student Already Registered'
end
end;
GO
```

Exam Generation

```sql

CREATE PROCEDURE [dbo].[Generate_And_Show_Exam]
    @St_ID INT,
    @Ex_ID INT,
    @Crs_ID INT
AS
BEGIN
    SET NOCOUNT ON;

    IF NOT EXISTS (SELECT 1 FROM Student_Exam WHERE St_ID = @St_ID AND Ex_ID = @Ex_ID)
    BEGIN
        PRINT 'Student not registered for this exam.';
        RETURN;
    END

    CREATE TABLE #temp_questions
    (
        Temp_ID INT IDENTITY(1,1) PRIMARY KEY,
        Qs_ID INT,
        Qs_Body VARCHAR(1000)
    );

    INSERT INTO #temp_questions (Qs_ID, Qs_Body)
    SELECT TOP 10 Qs_ID, Qs_Body
    FROM Questions
    WHERE Crs_ID = @Crs_ID
    ORDER BY NEWID();

    INSERT INTO Stu_Exam_Qs (Stu_ID, Ex_ID, Qs_ID, Question_Order)
    SELECT @St_ID, @Ex_ID, Qs_ID, Temp_ID
    FROM #temp_questions;

    SELECT Temp_ID AS QuestionNumber, Qs_Body AS Question
    FROM #temp_questions
    ORDER BY Temp_ID;

    PRINT 'Exam Questions Generated and Shown to Student';
END;
GO
```

Submitting Answers

```sql
CREATE PROCEDURE [dbo].[SubmitExamAnswers]
    @St_ID INT,
    @Ex_ID INT,
    @Answers NVARCHAR(MAX)
AS
BEGIN
    SET NOCOUNT ON;

    ;WITH StudentAnswers AS (
        SELECT
            value AS Stu_Answer,
            ROW_NUMBER() OVER (ORDER BY (SELECT NULL)) AS rn
        FROM STRING_SPLIT(@Answers, ',')
    )
    UPDATE S
    SET S.Stu_Answers = SA.Stu_Answer
    FROM Stu_Exam_Qs S
    INNER JOIN StudentAnswers SA
        ON SA.rn = S.Question_Order
    WHERE S.Stu_ID = @St_ID AND S.Ex_ID = @Ex_ID;

    DECLARE @Correct INT;
    SELECT @Correct = COUNT(*)
    FROM Stu_Exam_Qs S
    JOIN Questions Q ON S.Qs_ID = Q.Qs_ID
    WHERE S.Stu_ID = @St_ID AND S.Ex_ID = @Ex_ID
      AND S.Stu_Answers = Q.Correct_Answer;

    UPDATE Student_Exam
    SET
    Score = @Correct,
    Percentage = (CAST(@Correct AS FLOAT) / 10) * 100
    WHERE ST_ID = @St_ID AND Ex_ID = @Ex_ID;


    PRINT 'Exam submitted. Correct Answers: ' + CAST(@Correct AS VARCHAR(10));
END;
GO
```

Building Reports in

SQL Server Reporting Services (SSRS)

In this phase, SQL Server Reporting Services (SSRS) was used to design and generate a set of interactive reports that provide insights into various aspects of the examination system. Each report was directly connected to the SQL database to ensure accurate and real-time results.

The developed reports cover multiple perspectives within the system. For instance, one report displays student information based on department number, while another retrieves a specific student's grades across all courses. Additional reports were created to show instructor details with the courses they teach and the number of enrolled students, course topics by course ID, and exam details including related questions.

Moreover, a comprehensive report was implemented that accepts both exam number and student ID to display each question alongside the student's submitted answers. Through these reports, the system enables clear performance tracking, better academic evaluation, and informed decision-making by ITI staff and administrators.

SSRS Procedures

Get Instructor Information

```
ALTER proc [dbo].[GetInstructorsInfo] @ins_id int
as
begin

select i.Ins_FName + ' ' + i.Ins_LName as "FullName" , c.crs_name , count(s.st_id) As StudentCount
from Ins_course ie inner join Instructor i on ie.Ins_ID = i.Ins_ID
                  inner join Course c on ie.Crs_ID = c.Crs_ID
                  inner join Track_Course tc on c.Crs_ID = tc.Crs_ID
                  inner join track t on t.Track_ID = tc.Track_ID
                  inner join Student s on s.Track_ID = t.Track_ID
where i.Ins_ID = @ins_id
group by c.Crs_Name , I.Ins_FName , i.Ins_LName
end
```

Get Student by Topic

```
ALTER Proc [dbo].[GetStudentsByTopic] @Topic_Id int
as
begin
    set nocount on
    select
    s.fname ,
    S.Lname ,
    s.gender,
    s.city,
    s.Age,
    s.college ,
    s.university,
    s.grade_year,
    tc.topic_name

    from student s join track t on s.track_id = t.track_Id
    join topic tc on tc.topic_id = t.Topic_ID
    where tc.Topic_ID = @Topic_id
End
```

Get Topic by Course

```
ALTER Proc [dbo].[GetTopicByCourses] @crs_id int
as
begin

select distinct t.topic_name , c.Crs_Name
from Course c inner join Track_Course tc on c.Crs_ID = tc.Crs_ID
              inner join Track k on tc.Track_ID = k.Track_ID
              inner join topic t on t.Topic_ID = k.Topic_ID
where c.Crs_ID = @crs_id
end
```

Get Questions

```
ALTER Proc [dbo].[GetQuestions] @Exam_ID int
as
begin
SELECT
    Qs_ID,
    Qs_Body,
    [A] AS [Option A],
    [B] AS [Option B],
    [C] AS [Option C],
    [D] AS [Option D]
FROM
(
    SELECT
        q.Qs_ID,
        q.Qs_Body,
        c.Options_Header,
        c.Options_Value,
        q.Correct_Answer
    FROM Stu_Exam_Qs eq
    INNER JOIN Questions q
        ON eq.Qs_ID = q.Qs_ID
    INNER JOIN Choices c
        ON q.Qs_ID = c.Qs_ID
    WHERE eq.Ex_ID = @Exam_ID
) AS SourceTable
PIVOT
(
    MAX(Options_Value)
    FOR Options_Header IN ([A], [B], [C], [D])
) AS PivotTable
ORDER BY Qs_ID;
end
```

Get Student Answers

```
ALTER Proc [dbo].[GetStudentAnswers] @Exam_ID int , @Student_ID int
 as
begin
SELECT
     q.Qs_Body,
     eq.Stu_Answers as "Student Answer"
 FROM Stu_Exam_Qs eq
 INNER JOIN Questions q
     ON eq.Qs_ID = q.Qs_ID
     AND eq.Stu_ID = @Student_ID
     AND eq.Ex_ID = @Exam_ID
 WHERE eq.Ex_ID = @Exam_ID
 ORDER BY q.Qs_ID;
 end
```

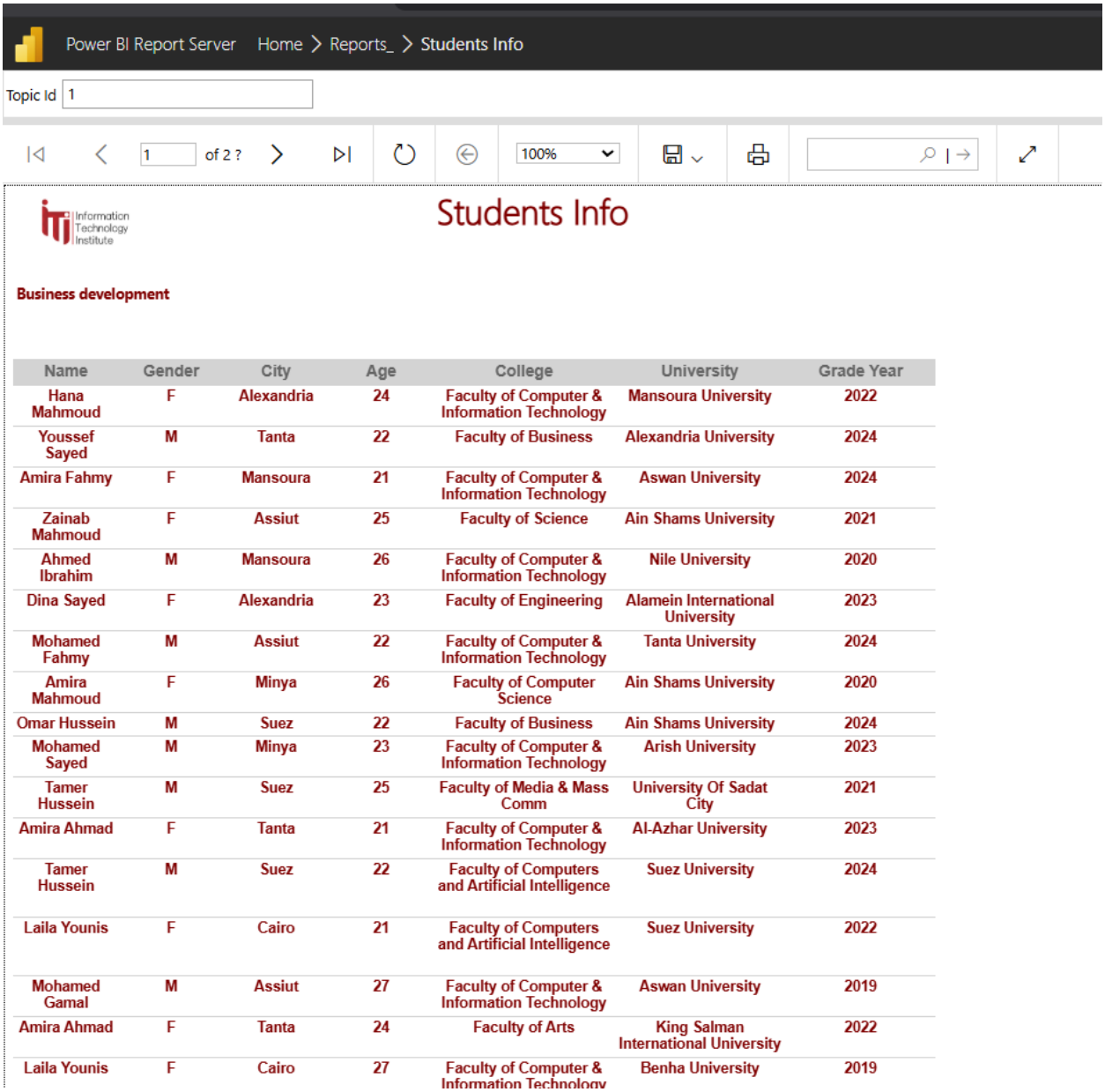
Get Student Grade

```
ALTER   proc [dbo].[GetStudentGrade] @St_id int
as
begin

select s.st_id , s.fname + ' ' + s.Lname as "Full Name" , c.crs_id , c.Crs_name , se.score , 10 as TotalMark,
cast (se.score * 100.0 / 10 as decimal (5,2)) as Percentage
from student s inner join Student_Exam se on s.ST_id = se.St_ID
               inner join Exam e on e.Ex_ID = se.Ex_ID
               inner join Course c on c.Crs_ID = e.Crs_ID
               where s.ST_id = @St_id
end
```

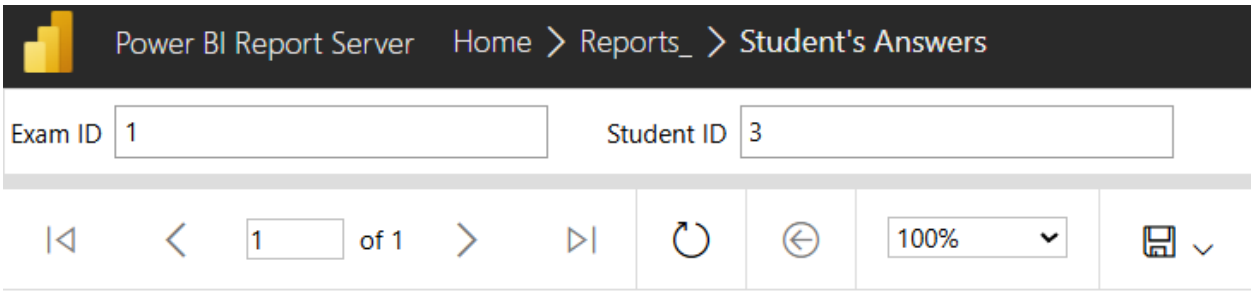
SSRS Reports

Report that returns the student information according to department NO parameter



Name	Gender	City	Age	College	University	Grade Year
Hana Mahmoud	F	Alexandria	24	Faculty of Computer & Information Technology	Mansoura University	2022
Youssef Sayed	M	Tanta	22	Faculty of Business	Alexandria University	2024
Amira Fahmy	F	Mansoura	21	Faculty of Computer & Information Technology	Aswan University	2024
Zainab Mahmoud	F	Assiut	25	Faculty of Science	Ain Shams University	2021
Ahmed Ibrahim	M	Mansoura	26	Faculty of Computer & Information Technology	Nile University	2020
Dina Sayed	F	Alexandria	23	Faculty of Engineering	Alamein International University	2023
Mohamed Fahmy	M	Assiut	22	Faculty of Computer & Information Technology	Tanta University	2024
Amira Mahmoud	F	Minya	26	Faculty of Computer Science	Ain Shams University	2020
Omar Hussein	M	Suez	22	Faculty of Business	Ain Shams University	2024
Mohamed Sayed	M	Minya	23	Faculty of Computer & Information Technology	Arish University	2023
Tamer Hussein	M	Suez	25	Faculty of Media & Mass Comm	University Of Sadat City	2021
Amira Ahmad	F	Tanta	21	Faculty of Computer & Information Technology	Al-Azhar University	2023
Tamer Hussein	M	Suez	22	Faculty of Computers and Artificial Intelligence	Suez University	2024
Laila Younis	F	Cairo	21	Faculty of Computers and Artificial Intelligence	Suez University	2022
Mohamed Gamal	M	Assiut	27	Faculty of Computer & Information Technology	Aswan University	2019
Amira Ahmad	F	Tanta	24	Faculty of Arts	King Salman International University	2022
Laila Younis	F	Cairo	27	Faculty of Computer & Information Technology	Benha University	2019

Report that takes the exam number and the student ID then return the questions in this exam with the student answer.



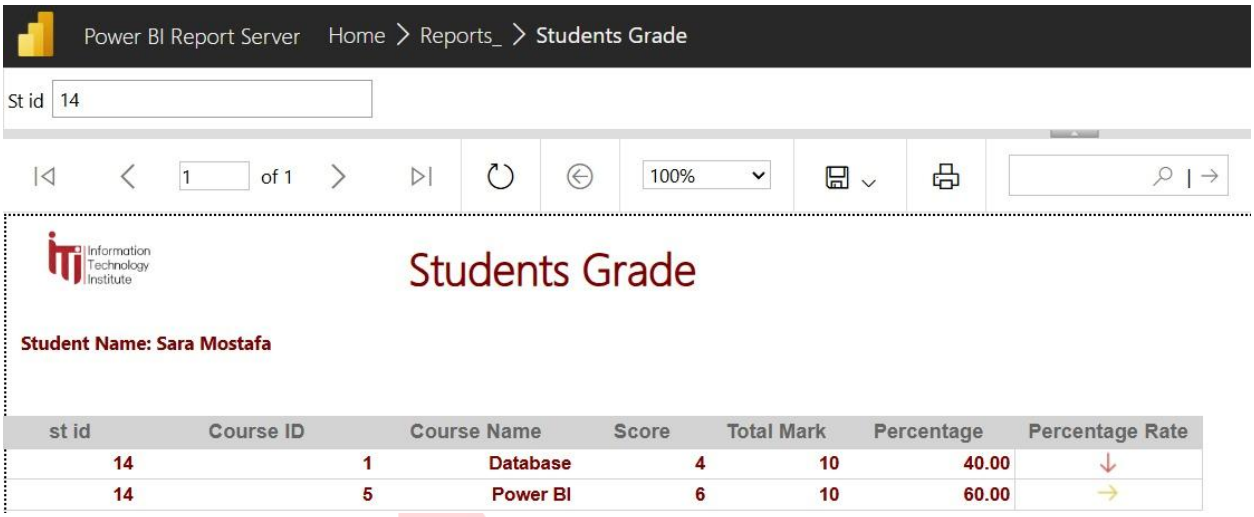
Information Technology Institute

Student's Answers

Exam ID: 1 | Student ID: 3

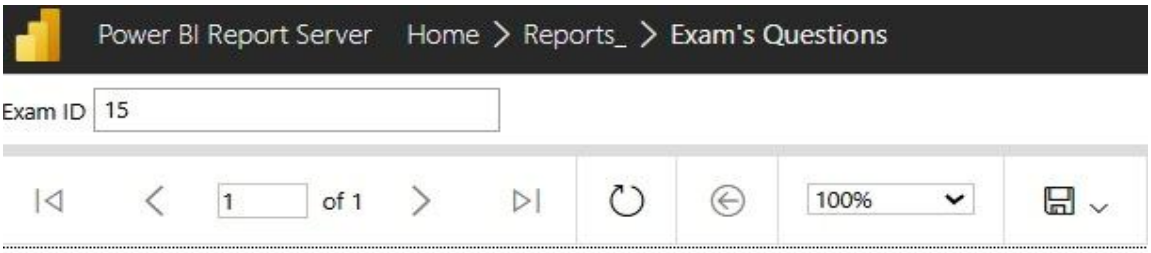
Qs Body	Student Answer
Which of the following ensures uniqueness in a table?	B
Which clause is used to filter records in SQL?	A
Which command is used to add a new record in SQL?	B
Which operator is used to test for a range of values?	True
Which SQL statement is used to rename a table?	A
Which of the following is a valid SQL data type?	A
Which SQL function counts distinct values?	A
Which statement is true about primary key?	A
Normalization helps reduce data redundancy.	A
JOIN is used to combine rows from multiple tables.	A

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



st id	Course ID	Course Name	Score	Total Mark	Percentage	Percentage Rate
14	1	Database	4	10	40.00	↓
14	5	Power BI	6	10	60.00	→

Report that takes exam number and returns questions in it.

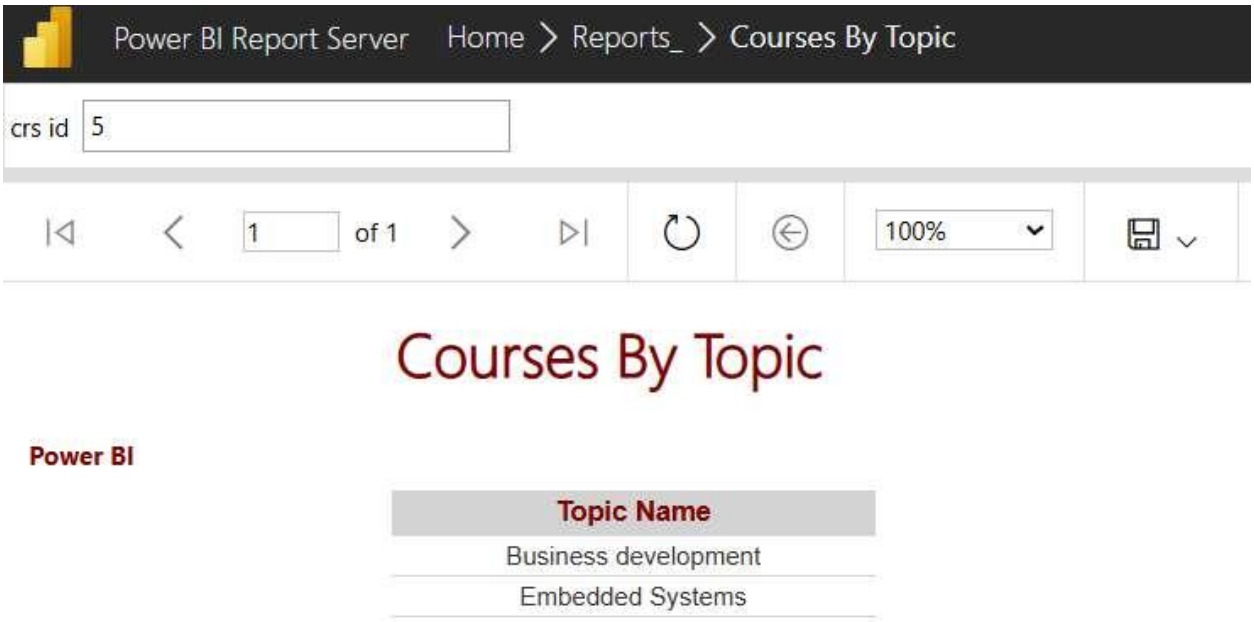


Information Technology Institute

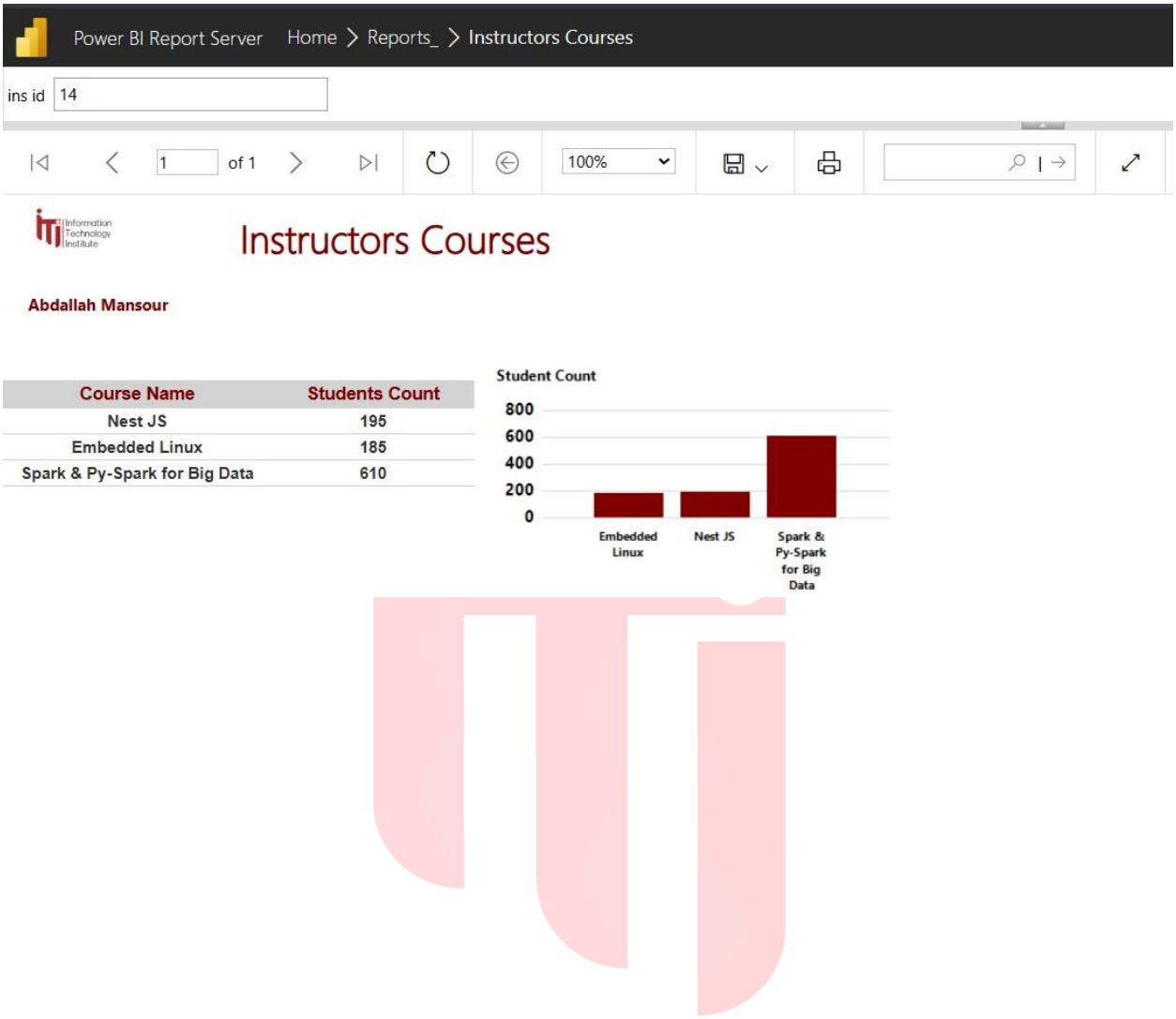
Exam's Questions

Qs ID	Qs Body	Option A	Option B	Option C	Option D
201	Which keyword is used to define a function in Python?	def	function	lambda	method
204	What symbol is used for comments in Python?	#	//	--	%
205	Which collection type allows duplicate elements?	list	set	tuple	dictionary
207	Which function displays output to the console?	print()	display()	echo()	show()
210	Which method removes the last element from a list?	remove()	pop()	delete()	clear()
211	Python is a statically typed language.	True	False		
212	Tuples in Python are immutable.	True	False		
213	The “elif” keyword is used for loops.	True	False		
214	The “in” keyword checks for membership in a	True	False		

Report that takes the course ID and returns its topics.

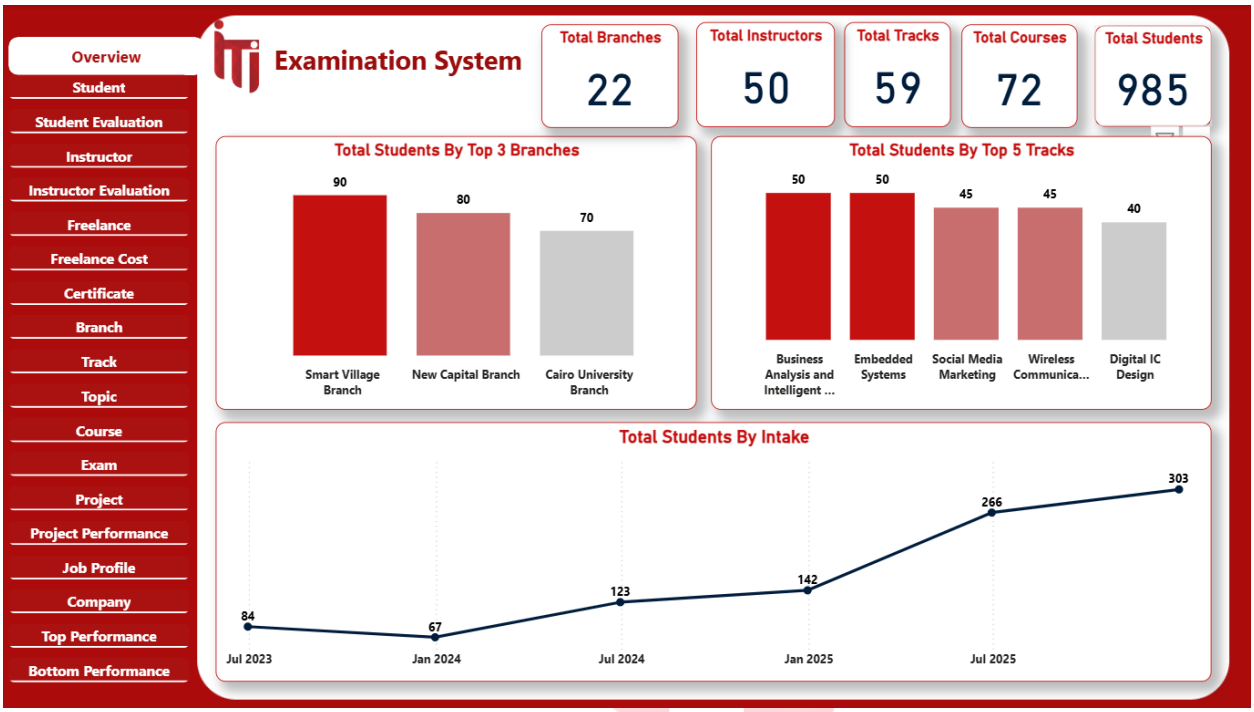


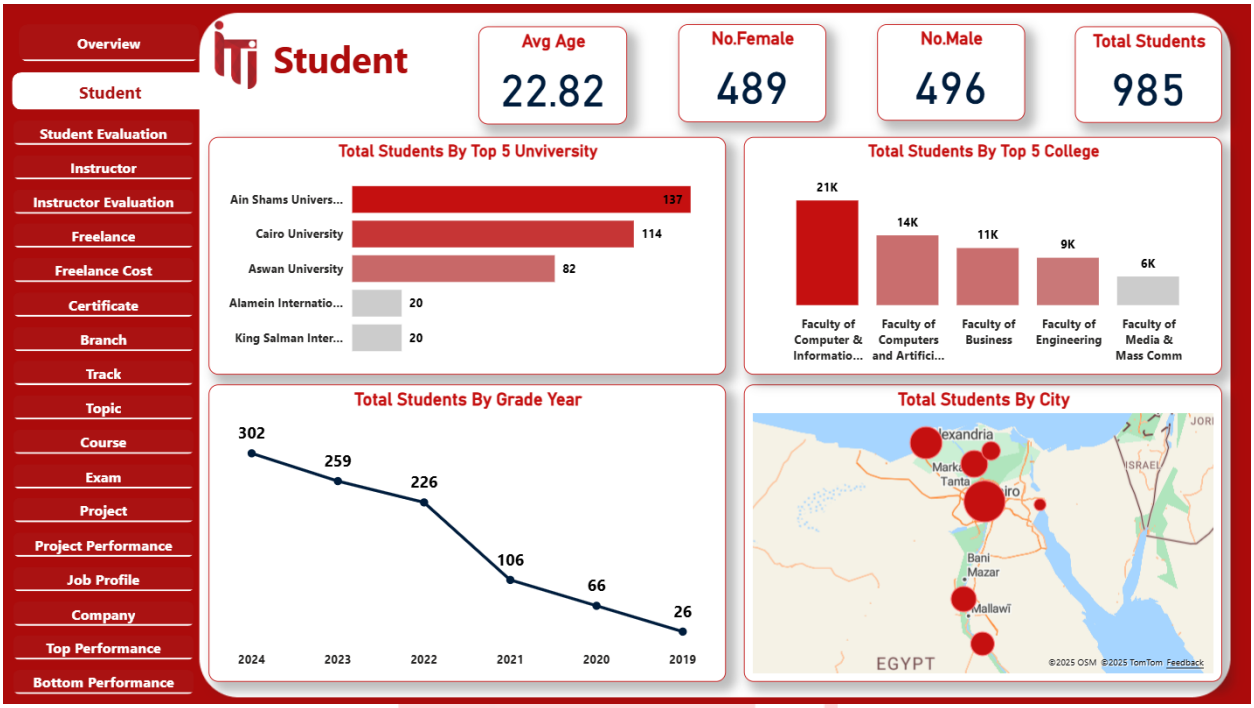
Report that takes the instructor's ID and returns the name of the course that he teaches and the number of students per course.

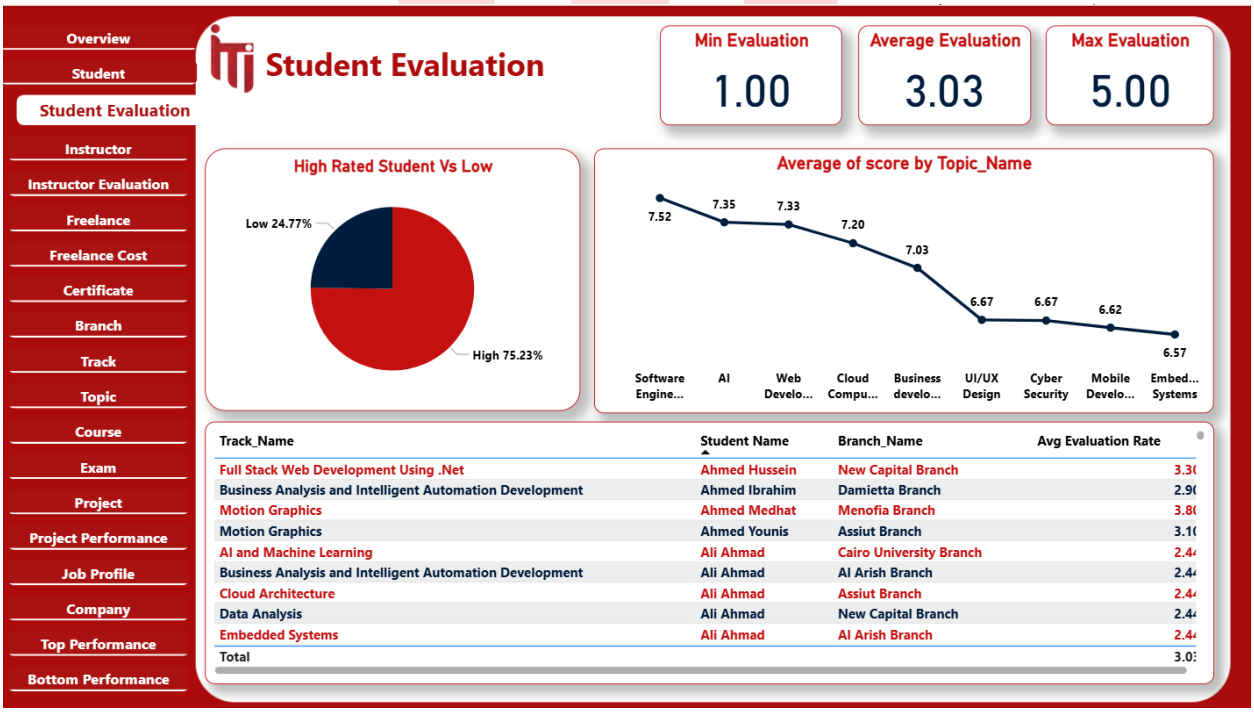


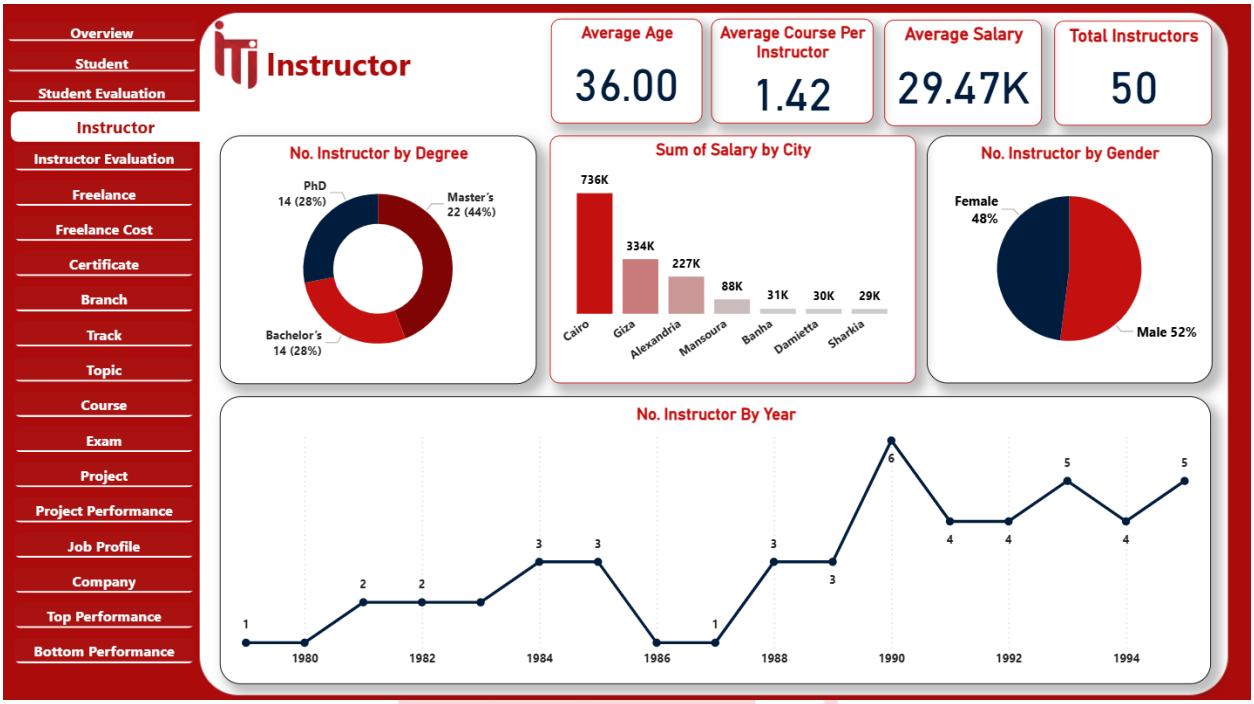
Power BI Dashboards

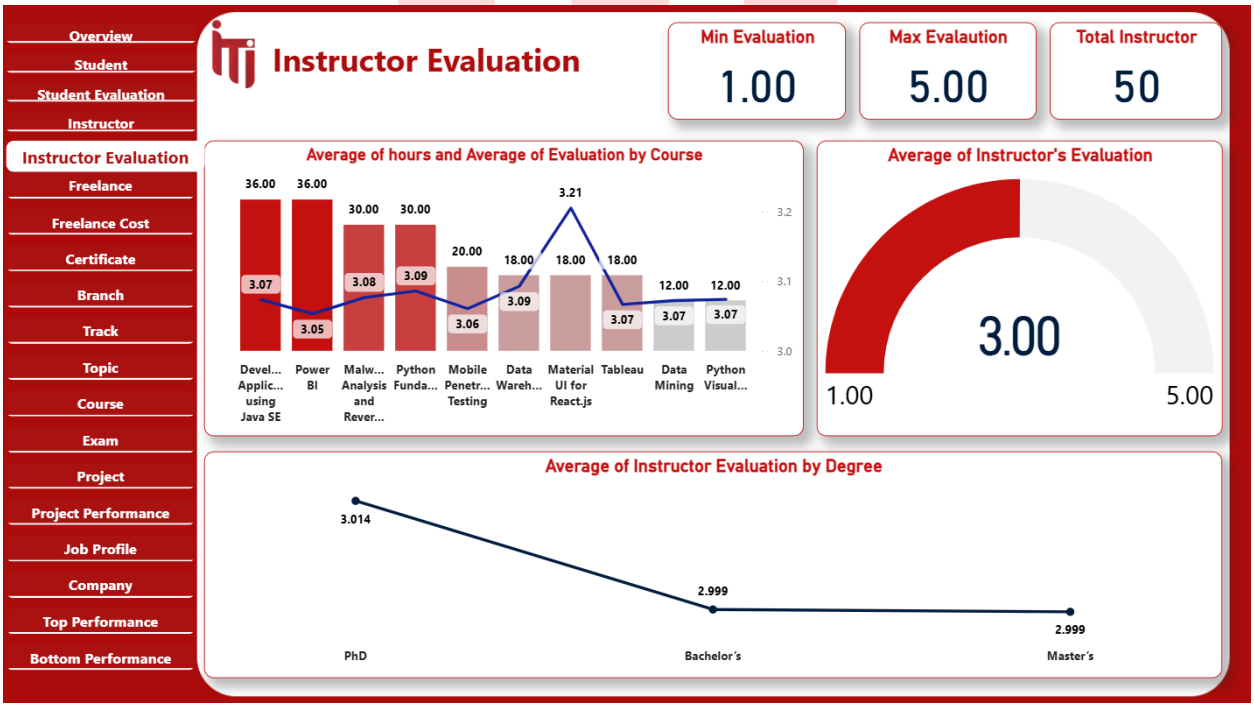
In this phase, Power BI was used to visualize data from the SQL Server database through interactive dashboards. These dashboards display real-time insights into student performance, exam results, instructor activity, and course outcomes. Using dynamic filtering and drill-through features, ITI staff can easily explore data, compare results, and monitor trends. This stage transformed complex database information into clear, actionable insights, empowering data-driven decision-making across the institution.

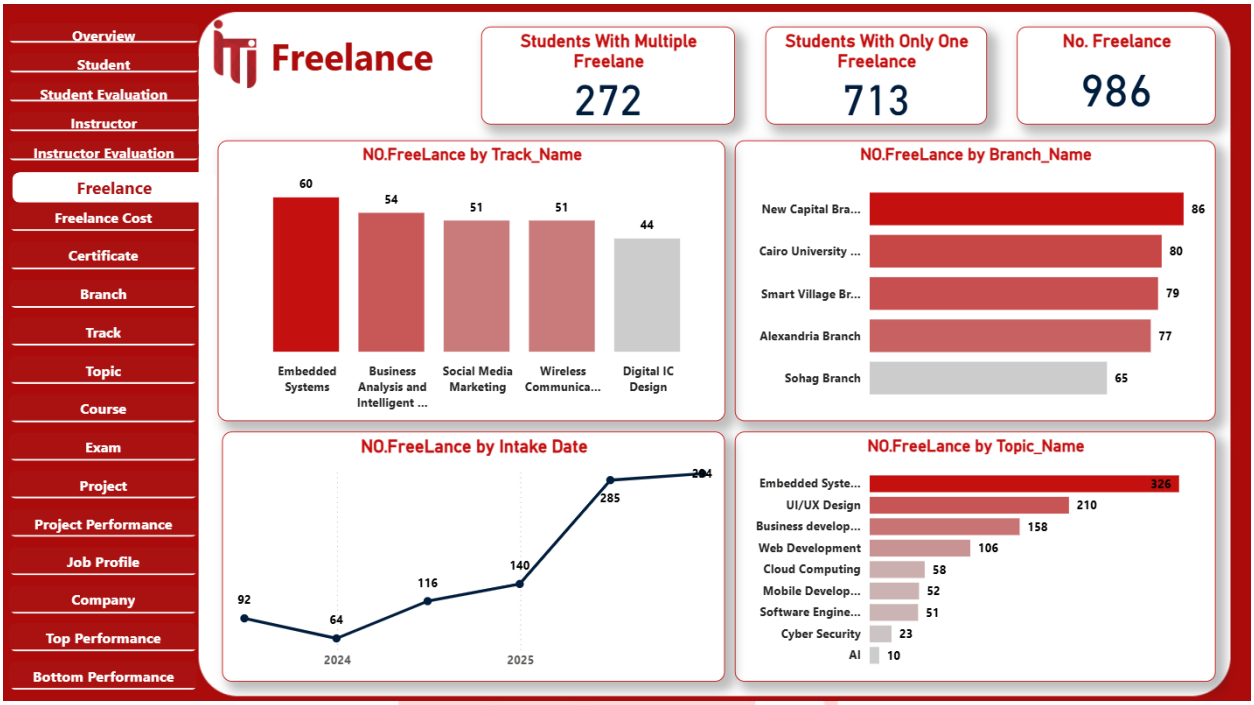


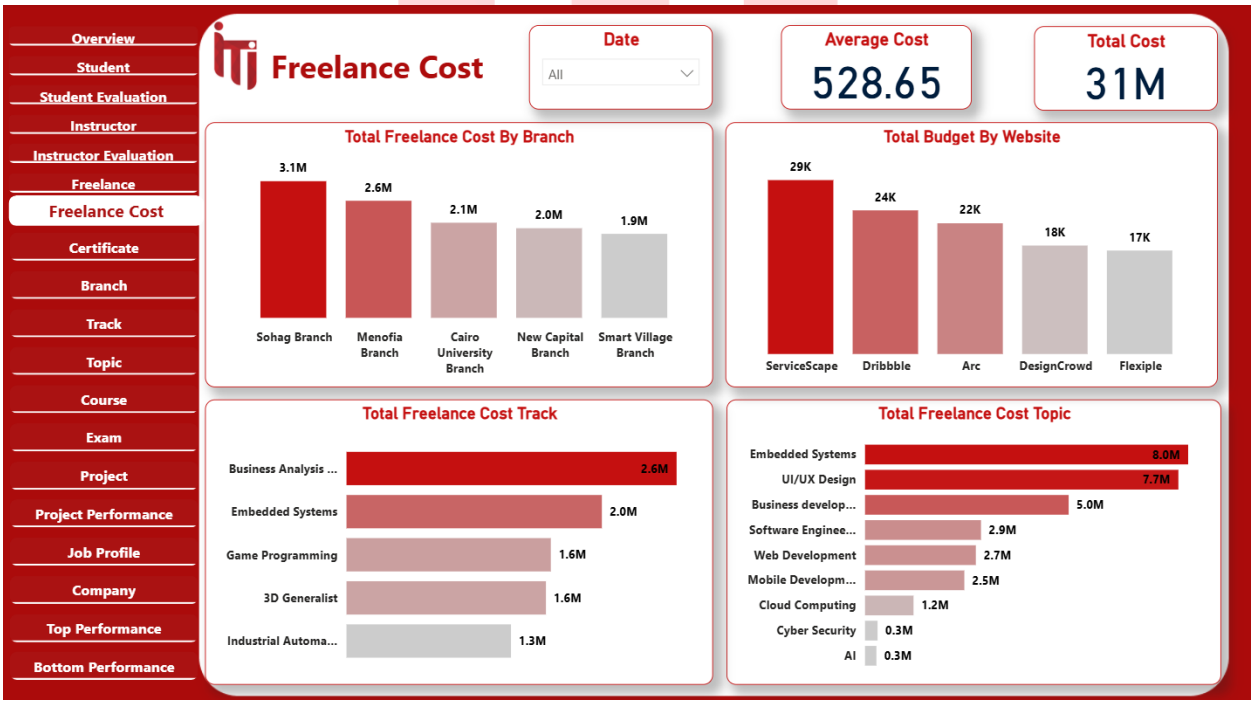

Overview
Student
Student Evaluation
Instructor
Instructor Evaluation
Freelance
Freelance Cost
Certificate
Branch
Track
Topic
Course
Exam
Project
Project Performance
Job Profile
Company
Top Performance
Bottom Performance
Student
Avg Age
22.82
No.Female
489
No.Male
496
Total Students
985
Total Students By Top 5 Unviversity
Ain Shams Univers...
137
Cairo University
114
Aswan University
82
Alamein Internatio...
20
King Salman Inter...
20
Total Students By Top 5 College
21K
14K
11K
9K
6K
Faculty of Computer & Informatio...
Faculty of Computers and Artifici...
Faculty of Business
Faculty of Engineering
Faculty of Media & Mass Comm
Total Students By Grade Year
302
259
226
106
66
26
2024
2023
2022
2021
2020
2019
Total Students By City
EGYPT
©2025 OSM ©2025 TomTom Feedback

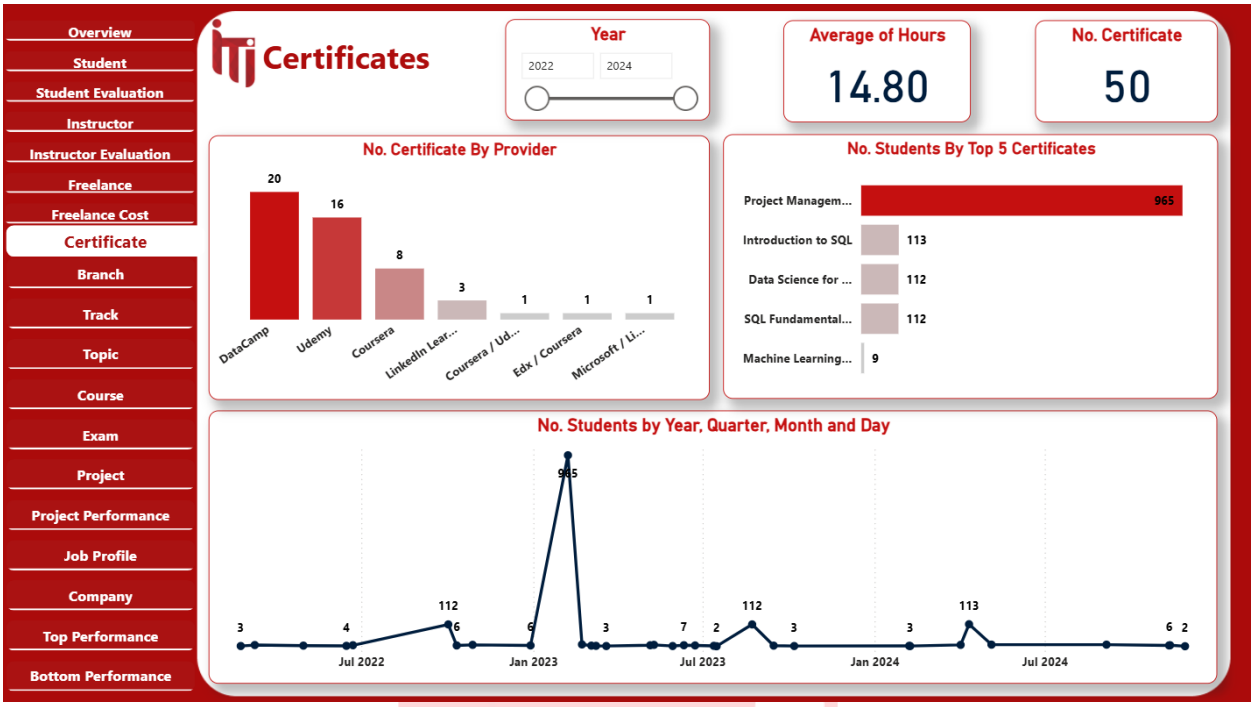

Overview
Student
Student Evaluation
Instructor
Instructor Evaluation
Freelance
Freelance Cost
Certificate
Branch
Track
Topic
Course
Exam
Project
Project Performance
Job Profile
Company
Top Performance
Bottom Performance
Student Evaluation
Min Evaluation
1.00
Average Evaluation
3.03
Max Evaluation
5.00
High Rated Student Vs Low
Low 24.77%
High 75.23%
Average of score by Topic_Name
7.52
7.35
7.33
7.20
7.03
6.67
6.67
6.62
6.57
Software Engine...
AI
Web Develo...
Cloud Compu...
Business develo...
UI/UX Design
Cyber Security
Mobile Develo...
Embed... Systems
Track_Name
Student Name
Branch_Name
Avg Evaluation Rate
Full Stack Web Development Using .Net
Ahmed Hussein
New Capital Branch
3.3(
Business Analysis and Intelligent Automation Development
Ahmed Ibrahim
Damietta Branch
2.9(
Motion Graphics
Ahmed Medhat
Menofia Branch
3.8(
Motion Graphics
Ahmed Younis
Assiut Branch
3.1(
AI and Machine Learning
Ali Ahmad
Cairo University Branch
2.4(
Business Analysis and Intelligent Automation Development
Ali Ahmad
Al Arish Branch
2.4(
Cloud Architecture
Ali Ahmad
Assiut Branch
2.4(
Data Analysis
Ali Ahmad
New Capital Branch
2.4(
Embedded Systems
Ali Ahmad
Al Arish Branch
2.4(
Total
3.0

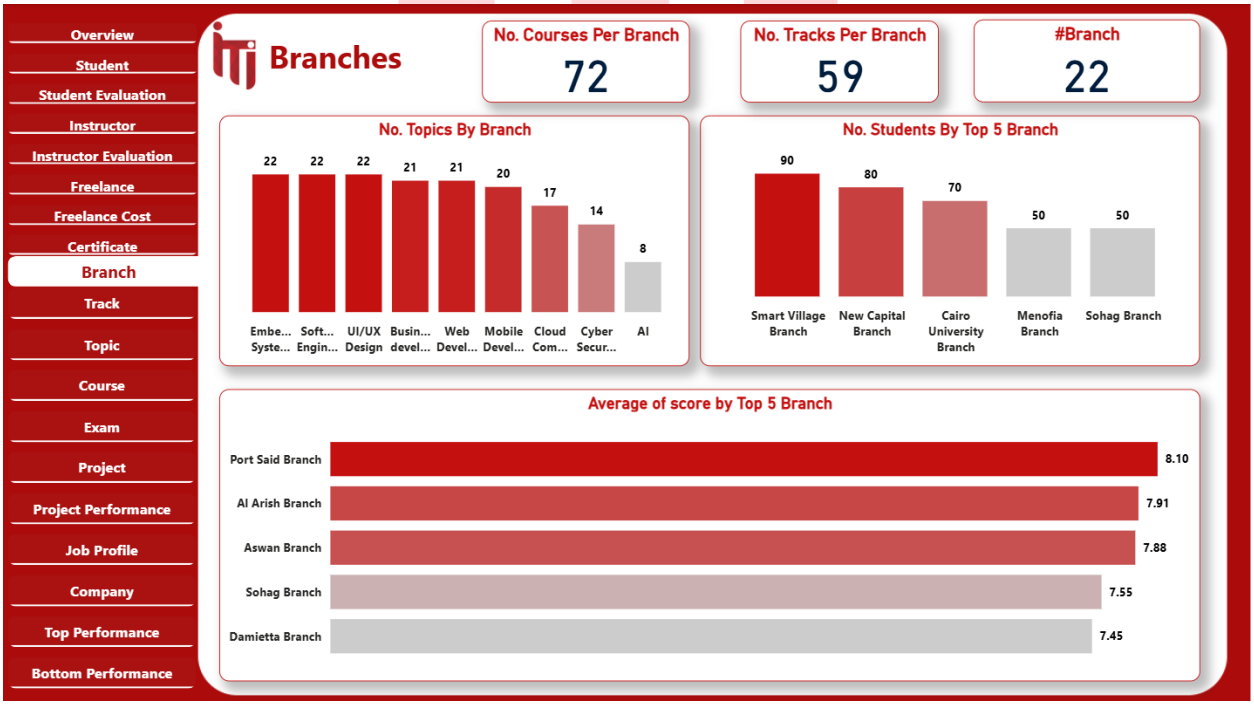

Overview
Student
Student Evaluation
Instructor
Instructor Evaluation
Freelance
Freelance Cost
Certificate
Branch
Track
Topic
Course
Exam
Project
Project Performance
Job Profile
Company
Top Performance
Bottom Performance
Instructor
Average Age
36.00
Average Course Per Instructor
1.42
Average Salary
29.47K
Total Instructors
50
No. Instructor by Degree
PhD 14 (28%)
Master's 22 (44%)
Bachelor's 14 (28%)
Sum of Salary by City
736K
334K
227K
88K
31K
30K
29K
Cairo
Giza
Alexandria
Mansoura
Banha
Damietta
Sharkia
No. Instructor by Gender
Female 48%
Male 52%
No. Instructor By Year
1
2
2
3
3
1
3
3
6
4
4
5
4
5
1980
1982
1984
1986
1988
1990
1992
1994

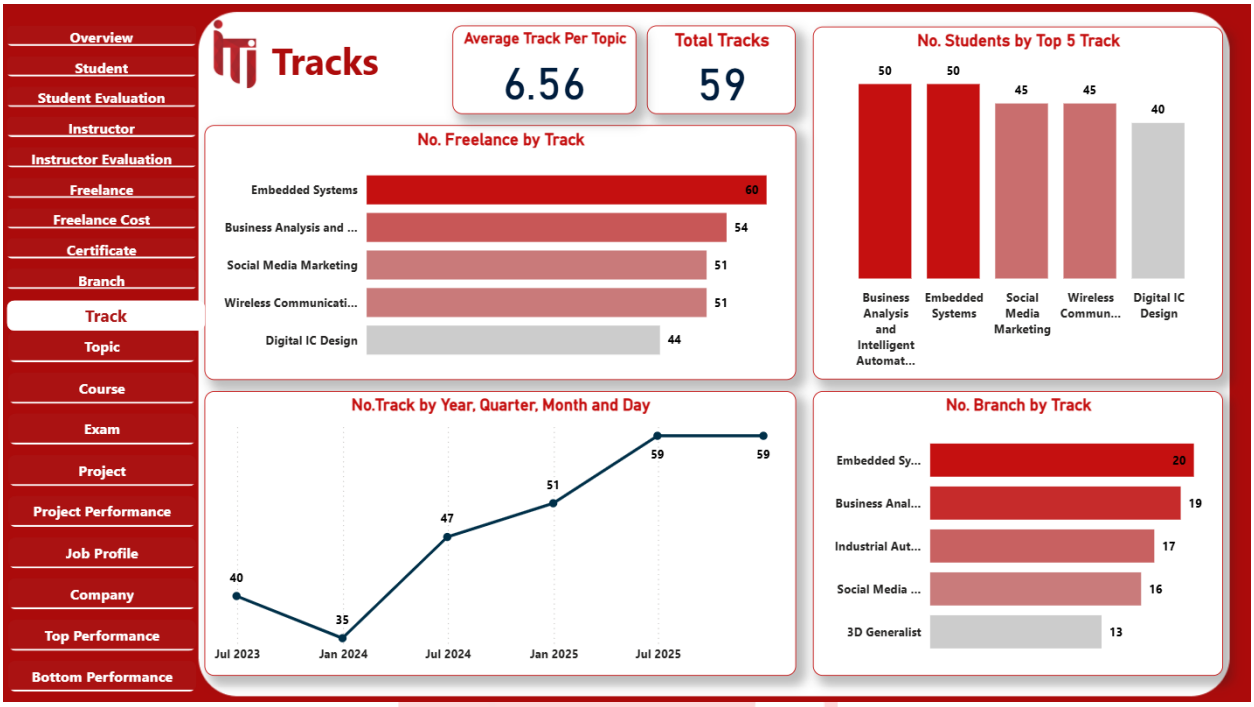

Overview
Student
Student Evaluation
Instructor
Instructor Evaluation
Freelance
Freelance Cost
Certificate
Branch
Track
Topic
Course
Exam
Project
Project Performance
Job Profile
Company
Top Performance
Bottom Performance
Instructor Evaluation
Min Evaluation
1.00
Max Evalaution
5.00
Total Instructor
50
Average of hours and Average of Evaluation by Course
36.00
36.00
30.00
30.00
20.00
18.00
3.21
18.00
18.00
12.00
12.00
3.07
3.05
3.08
3.09
3.06
3.09
3.07
3.07
3.07
3.2
3.1
3.0
Devel... Applic... using Java SE
Power BI
Malw... Analysis and Rever...
Python Funda...
Mobile Penetr... Testing
Data Wareh...
Material UI for React.js
Tableau
Data Mining
Python Visual...
Average of Instructor's Evaluation
3.00
1.00
5.00
Average of Instructor Evaluation by Degree
3.014
2.999
2.999
PhD
Bachelor's
Master's

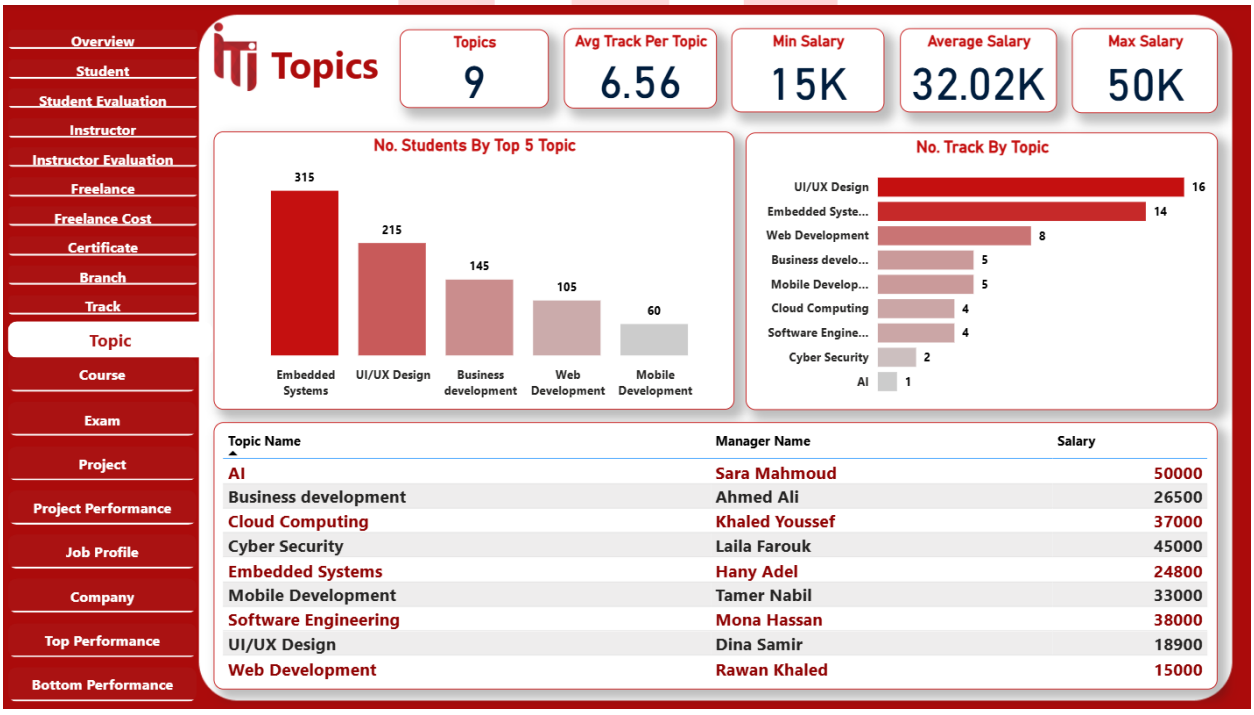

Overview
Student
Student Evaluation
Instructor
Instructor Evaluation
Freelance
Freelance Cost
Certificate
Branch
Track
Topic
Course
Exam
Project
Project Performance
Job Profile
Company
Top Performance
Bottom Performance
Freelance
Students With Multiple Freelane
272
Students With Only One Freelance
713
No. Freelance
986
NO.FreeLance by Track_Name
60
54
51
51
44
Embedded Systems
Business Analysis and Intelligent ...
Social Media Marketing
Wireless Communica...
Digital IC Design
NO.FreeLance by Branch_Name
New Capital Bra... 86
Cairo University ... 80
Smart Village Br... 79
Alexandria Branch 77
Sohag Branch 65
NO.FreeLance by Intake Date
92
64
116
140
285
304
2024
2025
NO.FreeLance by Topic_Name
Embedded Syste... 326
UI/UX Design 210
Business develop... 158
Web Development 106
Cloud Computing 58
Mobile Develop... 52
Software Engine... 51
Cyber Security 23
AI 10


Overview
Student
Student Evaluation
Instructor
Instructor Evaluation
Freelance
Freelance Cost
Certificate
Branch
Track
Topic
Course
Exam
Project
Project Performance
Job Profile
Company
Top Performance
Bottom Performance
Freelance Cost
Date
All
Average Cost
528.65
Total Cost
31M
Total Freelance Cost By Branch
3.1M
2.6M
2.1M
2.0M
1.9M
Sohag Branch
Menofia Branch
Cairo University Branch
New Capital Branch
Smart Village Branch
Total Budget By Website
29K
24K
22K
18K
17K
ServiceScape
Dribbble
Arc
DesignCrowd
Flexiple
Total Freelance Cost Track
Business Analysis ... 2.6M
Embedded Systems 2.0M
Game Programming 1.6M
3D Generalist 1.6M
Industrial Automa... 1.3M
Total Freelance Cost Topic
Embedded Systems 8.0M
UI/UX Design 7.7M
Business develop... 5.0M
Software Enginee... 2.9M
Web Development 2.7M
Mobile Developm... 2.5M
Cloud Computing 1.2M
Cyber Security 0.3M
AI 0.3M

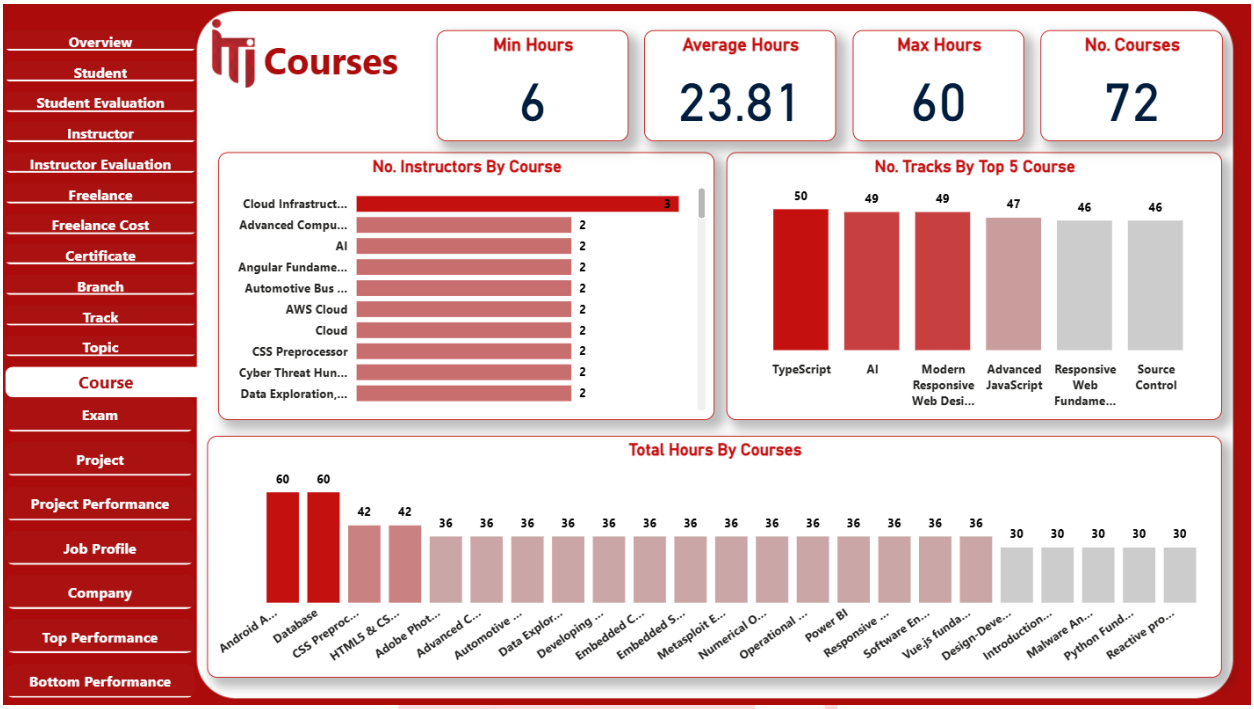

Overview
Student
Student Evaluation
Instructor
Instructor Evaluation
Freelance
Freelance Cost
Certificate
Branch
Track
Topic
Course
Exam
Project
Project Performance
Job Profile
Company
Top Performance
Bottom Performance
Certificates
Year
2022
2024
Average of Hours
14.80
No. Certificate
50
No. Certificate By Provider
20
16
8
3
1
1
1
DataCamp
Udemy
Coursera
LinkedIn Lear...
Coursera / Ud...
Edx / Coursera
Microsoft / Li...
No. Students By Top 5 Certificates
Project Managem...
965
Introduction to SQL
113
Data Science for ...
112
SQL Fundamental...
112
Machine Learning...
9
No. Students by Year, Quarter, Month and Day
3
4
112
6
6
965
3
7
2
112
3
3
113
6
2
Jul 2022
Jan 2023
Jul 2023
Jan 2024
Jul 2024

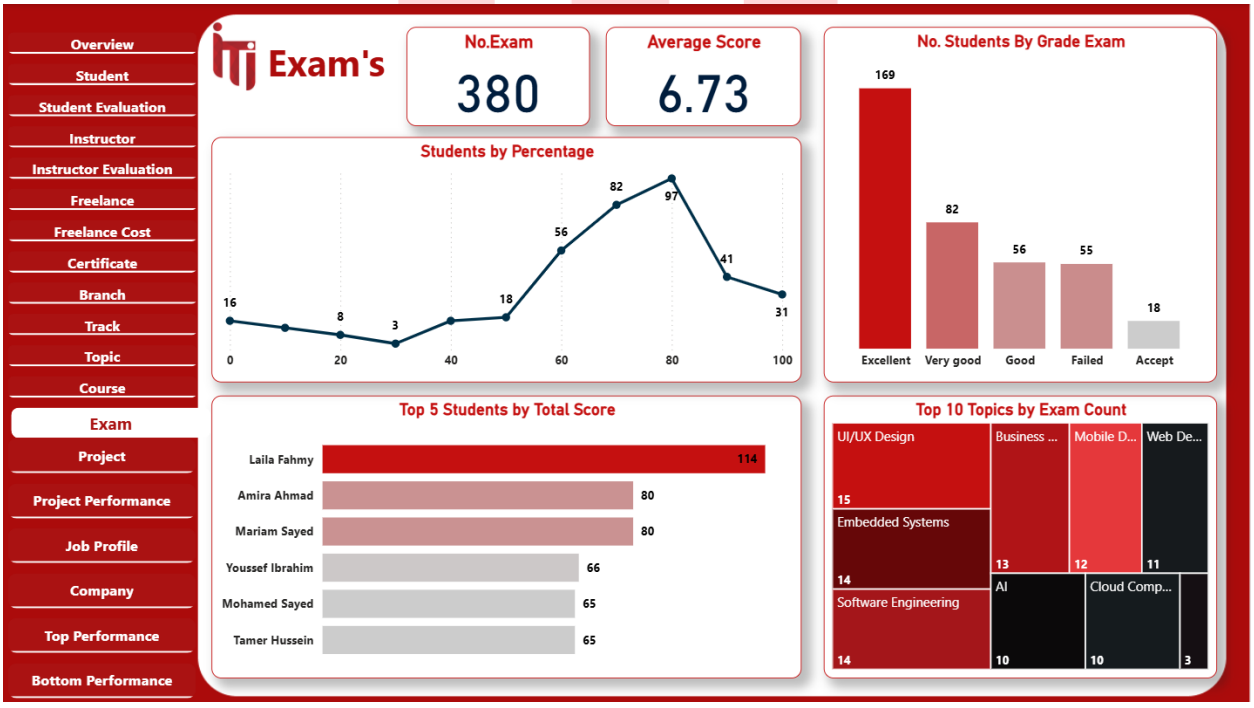

Overview
Student
Student Evaluation
Instructor
Instructor Evaluation
Freelance
Freelance Cost
Certificate
Branch
Track
Topic
Course
Exam
Project
Project Performance
Job Profile
Company
Top Performance
Bottom Performance
Branches
No. Courses Per Branch
72
No. Tracks Per Branch
59
#Branch
22
No. Topics By Branch
22
22
22
21
21
20
17
14
8
Embe... Syste...
Soft... Engin...
UI/UX Design
Busin... devel...
Web Devel...
Mobile Devel...
Cloud Com...
Cyber Secur...
AI
No. Students By Top 5 Branch
90
80
70
50
50
Smart Village Branch
New Capital Branch
Cairo University Branch
Menofia Branch
Sohag Branch
Average of score by Top 5 Branch
Port Said Branch
8.10
Al Arish Branch
7.91
Aswan Branch
7.88
Sohag Branch
7.55
Damietta Branch
7.45

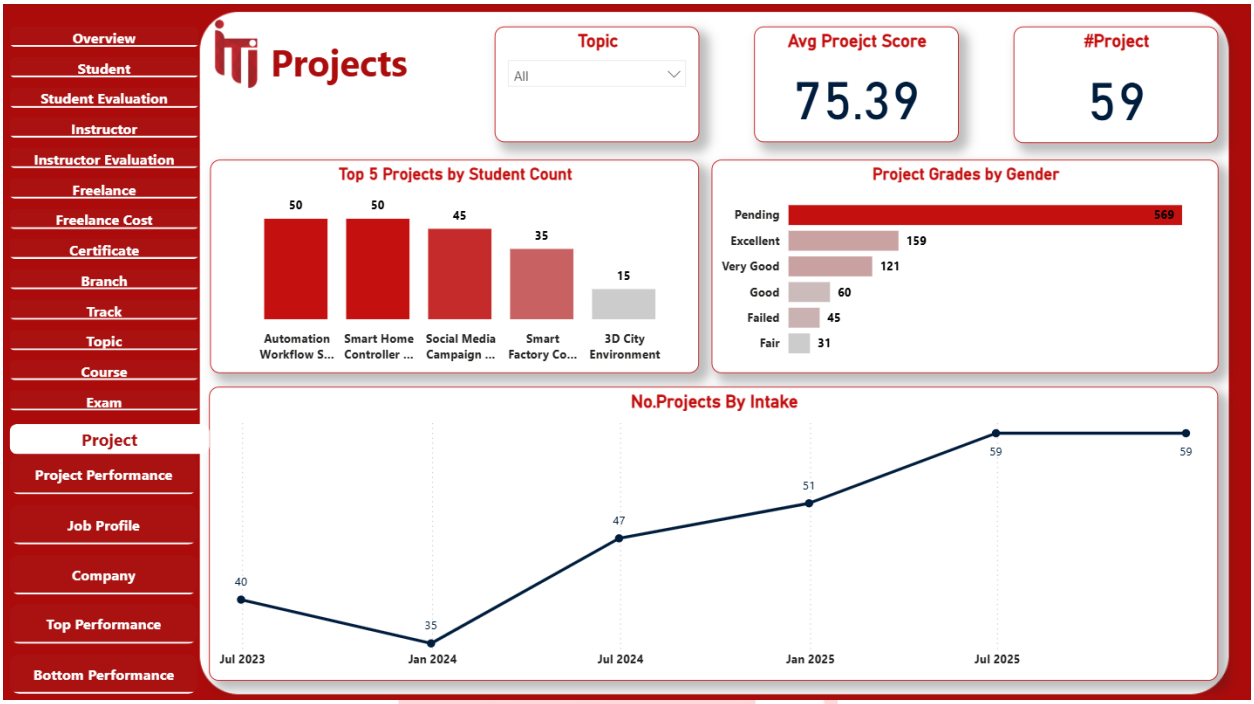


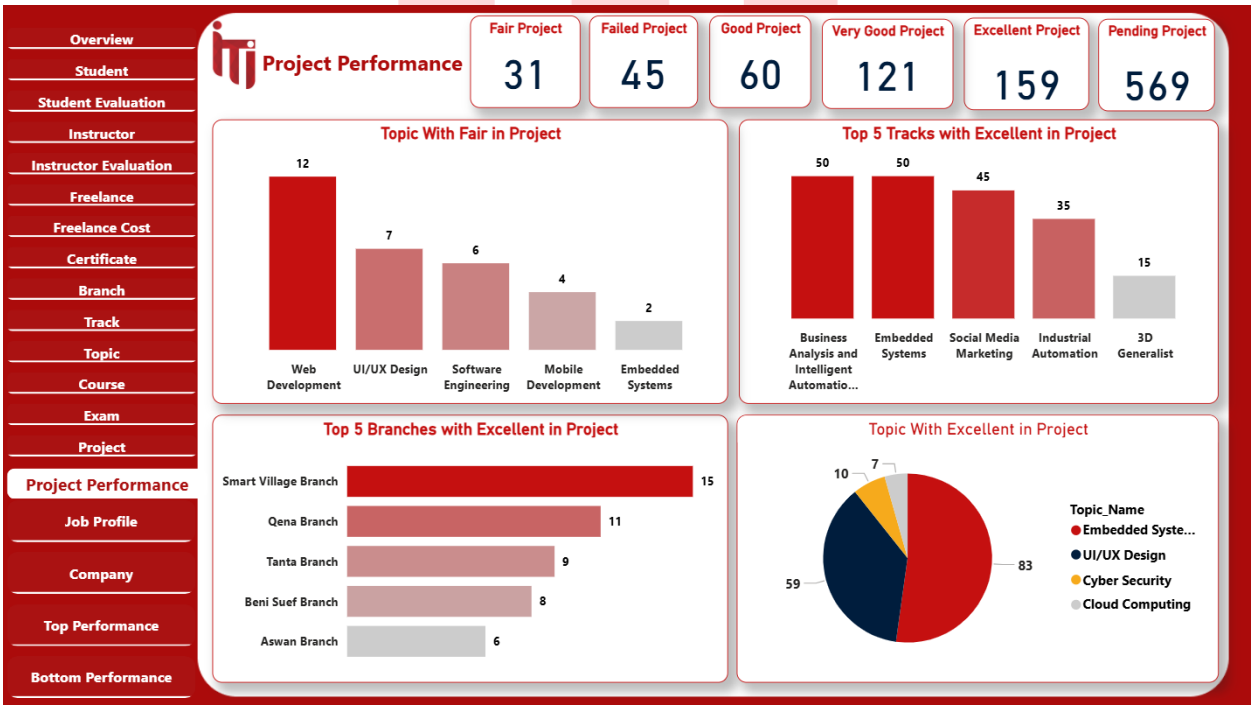


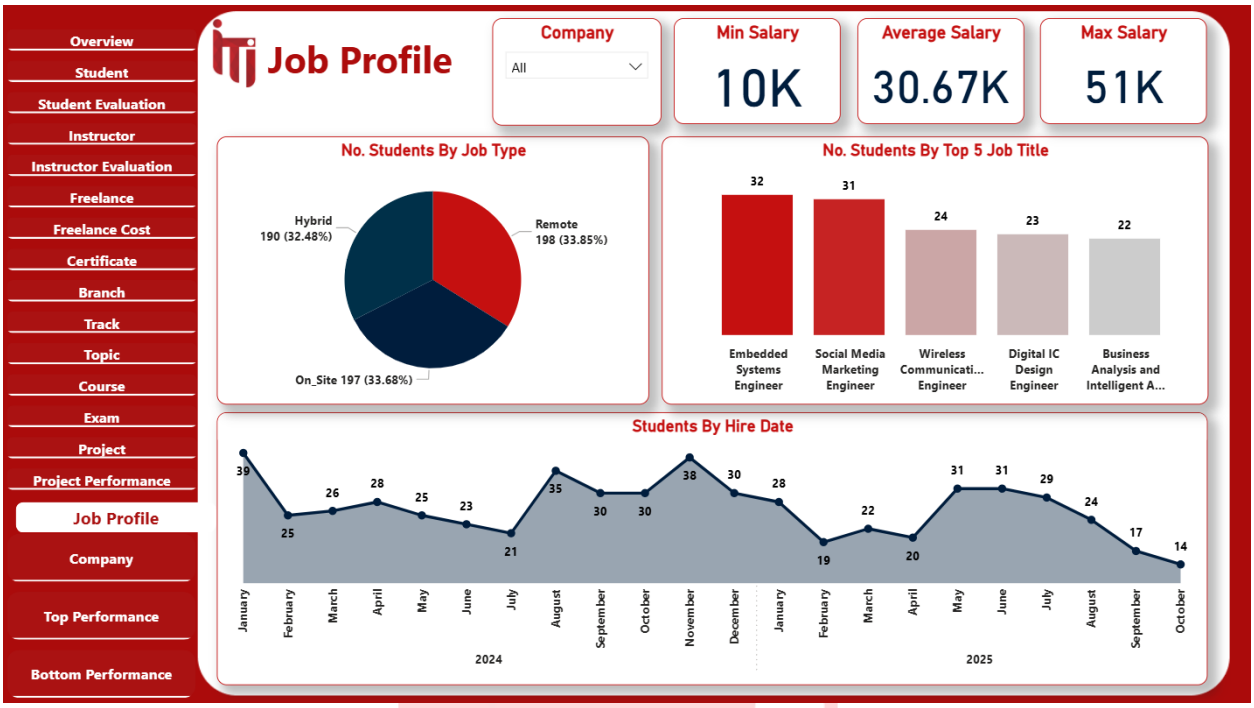
Topic Name	Manager Name	Salary
AI	Sara Mahmoud	50000
Business development	Ahmed Ali	26500
Cloud Computing	Khaled Youssef	37000
Cyber Security	Laila Farouk	45000
Embedded Systems	Hany Adel	24800
Mobile Development	Tamer Nabil	33000
Software Engineering	Mona Hassan	38000
UI/UX Design	Dina Samir	18900
Web Development	Rawan Khaled	15000

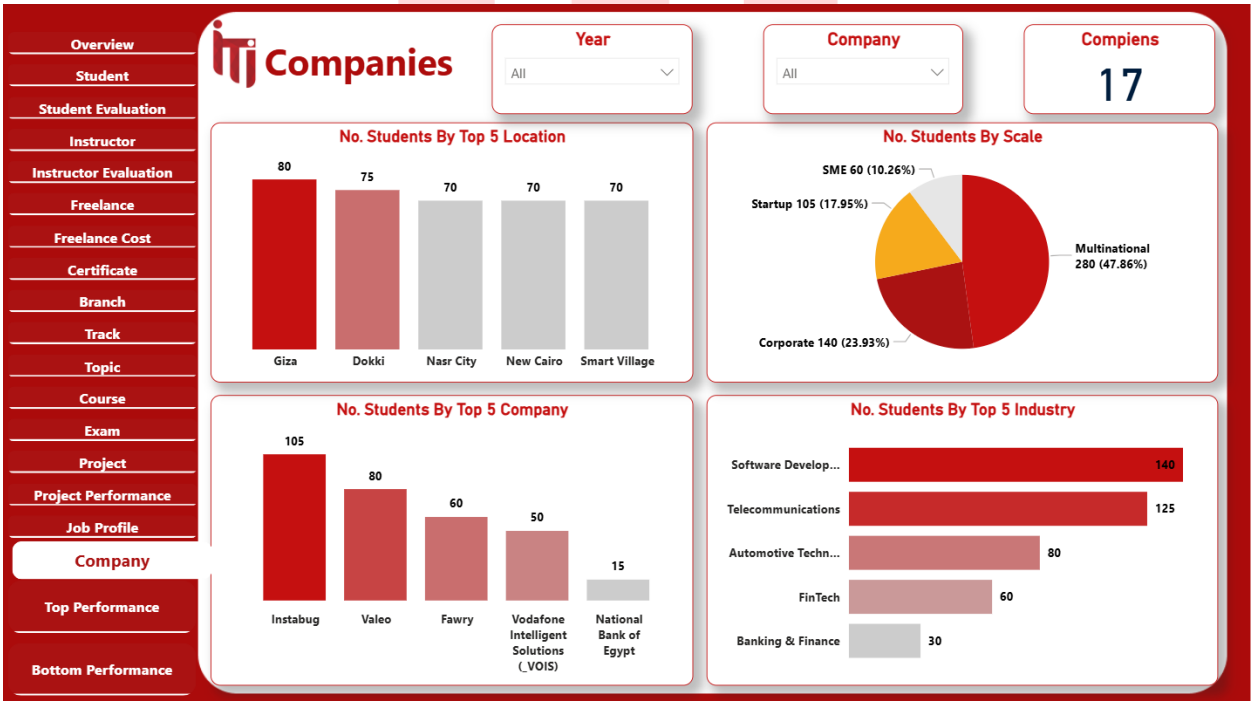

Overview
Student
Student Evaluation
Instructor
Instructor Evaluation
Freelance
Freelance Cost
Certificate
Branch
Track
Topic
Course
Exam
Project
Project Performance
Job Profile
Company
Top Performance
Bottom Performance
Courses
Min Hours
6
Average Hours
23.81
Max Hours
60
No. Courses
72
No. Instructors By Course
Cloud Infrastruct... 3
Advanced Compu... 2
AI 2
Angular Fundame... 2
Automotive Bus ... 2
AWS Cloud 2
Cloud 2
CSS Preprocessor 2
Cyber Threat Hun... 2
Data Exploration,... 2
No. Tracks By Top 5 Course
50 49 49 47 46 46
TypeScript
AI
Modern Responsive Web Desi...
Advanced JavaScript
Responsive Web Fundame...
Source Control
Total Hours By Courses
60 60 42 42 36 36 36 36 36 36 36 36 36 36 36 36 36 36 30 30 30 30 30
Android A...
Database
CSS Preproc...
HTML5 & CS...
Adobe Phot...
Advanced C...
Automotive ...
Data Explor...
Developing ...
Embedded C...
Embedded S...
Metasploit E...
Numerical O...
Operational ...
Power BI
Responsive ...
Software En...
Vue.js funda...
Design Deve...
Introduction...
Malware An...
Python Fund...
Reactive pro...


Overview
Student
Student Evaluation
Instructor
Instructor Evaluation
Freelance
Freelance Cost
Certificate
Branch
Track
Topic
Course
Exam
Project
Project Performance
Job Profile
Company
Top Performance
Bottom Performance
Exam's
No.Exam
380
Average Score
6.73
No. Students By Grade Exam
169 82 56 55 18
Excellent
Very good
Good
Failed
Accept
Students by Percentage
16 8 3 18 56 82 97 41 31
0 20 40 60 80 100
Top 5 Students by Total Score
Laila Fahmy 114
Amira Ahmad 80
Mariam Sayed 80
Youssef Ibrahim 66
Mohamed Sayed 65
Tamer Hussein 65
Top 10 Topics by Exam Count
UI/UX Design 15
Business ... 13
Mobile D... 12
Web De... 11
Embedded Systems 14
AI 10
Cloud Comp... 10
Software Engineering 14
3


Overview
Student
Student Evaluation
Instructor
Instructor Evaluation
Freelance
Freelance Cost
Certificate
Branch
Track
Topic
Course
Exam
Project
Project Performance
Job Profile
Company
Top Performance
Bottom Performance
Projects
Topic
All
Avg Proejct Score
75.39
#Project
59
Top 5 Projects by Student Count
50
50
45
35
15
Automation Workflow S...
Smart Home Controller ...
Social Media Campaign ...
Smart Factory Co...
3D City Environment
Project Grades by Gender
Pending 569
Excellent 159
Very Good 121
Good 60
Failed 45
Fair 31
No.Projects By Intake
40
35
47
51
59
59
Jul 2023
Jan 2024
Jul 2024
Jan 2025
Jul 2025


Overview
Student
Student Evaluation
Instructor
Instructor Evaluation
Freelance
Freelance Cost
Certificate
Branch
Track
Topic
Course
Exam
Project
Project Performance
Job Profile
Company
Top Performance
Bottom Performance
Project Performance
Fair Project
31
Failed Project
45
Good Project
60
Very Good Project
121
Excellent Project
159
Pending Project
569
Topic With Fair in Project
12
7
6
4
2
Web Development
UI/UX Design
Software Engineering
Mobile Development
Embedded Systems
Top 5 Tracks with Excellent in Project
50
50
45
35
15
Business Analysis and Intelligent Automatio...
Embedded Systems
Social Media Marketing
Industrial Automation
3D Generalist
Top 5 Branches with Excellent in Project
Smart Village Branch 15
Qena Branch 11
Tanta Branch 9
Beni Suef Branch 8
Aswan Branch 6
Topic With Excellent in Project
10
7
59
83
Topic_Name
Embedded Syste...
UI/UX Design
Cyber Security
Cloud Computing


Overview
Student
Student Evaluation
Instructor
Instructor Evaluation
Freelance
Freelance Cost
Certificate
Branch
Track
Topic
Course
Exam
Project
Project Performance
Job Profile
Company
Top Performance
Bottom Performance
Job Profile
Company
All
Min Salary
10K
Average Salary
30.67K
Max Salary
51K
No. Students By Job Type
Hybrid
190 (32.48%)
Remote
198 (33.85%)
On_Site 197 (33.68%)
No. Students By Top 5 Job Title
32
31
24
23
22
Embedded Systems Engineer
Social Media Marketing Engineer
Wireless Communicati... Engineer
Digital IC Design Engineer
Business Analysis and Intelligent A...
Students By Hire Date
39
25
26
28
25
23
21
35
30
30
38
30
28
19
22
20
31
31
29
24
17
14
January
February
March
April
May
June
July
August
September
October
November
December
January
February
March
April
May
June
July
August
September
October
2024
2025

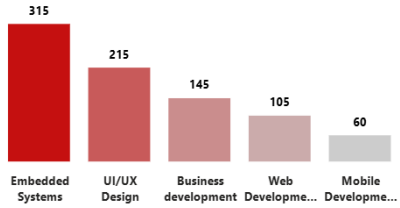

Overview
Student
Student Evaluation
Instructor
Instructor Evaluation
Freelance
Freelance Cost
Certificate
Branch
Track
Topic
Course
Exam
Project
Project Performance
Job Profile
Company
Top Performance
Bottom Performance
Companies
Year
All
Company
All
Compiens
17
No. Students By Top 5 Location
80
75
70
70
70
Giza
Dokki
Nasr City
New Cairo
Smart Village
No. Students By Scale
SME 60 (10.26%)
Startup 105 (17.95%)
Multinational
280 (47.86%)
Corporate 140 (23.93%)
No. Students By Top 5 Company
105
80
60
50
15
Instabug
Valeo
Fawry
Vodafone Intelligent Solutions (_VOIS)
National Bank of Egypt
No. Students By Top 5 Industry
Software Develop...
140
Telecommunications
125
Automotive Techn...
80
FinTech
60
Banking & Finance
30

Overview
Student
Student Evaluation
Instructor
Instructor Evaluation
Freelance
Freelance Cost
Certificate
Branch
Track
Topic
Course
Exam
Project
Project Performance
Job Profile
Company
Top Performance
Bottom Performance

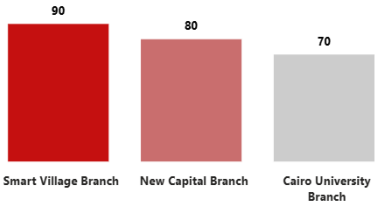
Top Performance

Max Salary of Topic Manager	Max Salary of Job Profile	Max Salary of Instructor
50K	51K	36K

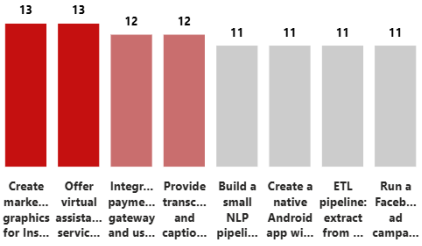
No. Students By Top 5 Topic



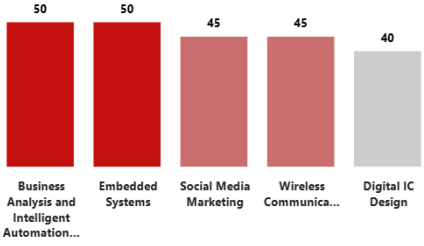
No. Students By Top 3 Branch



No. Students By Top 8 Freelance



No. Students By Top 5 Track



Overview
Student
Student Evaluation
Instructor
Instructor Evaluation
Freelance
Freelance Cost
Certificate
Branch
Track
Topic
Course
Exam
Project
Project Performance
Job Profile
Company
Top Performance
Bottom Performance

Top Performance

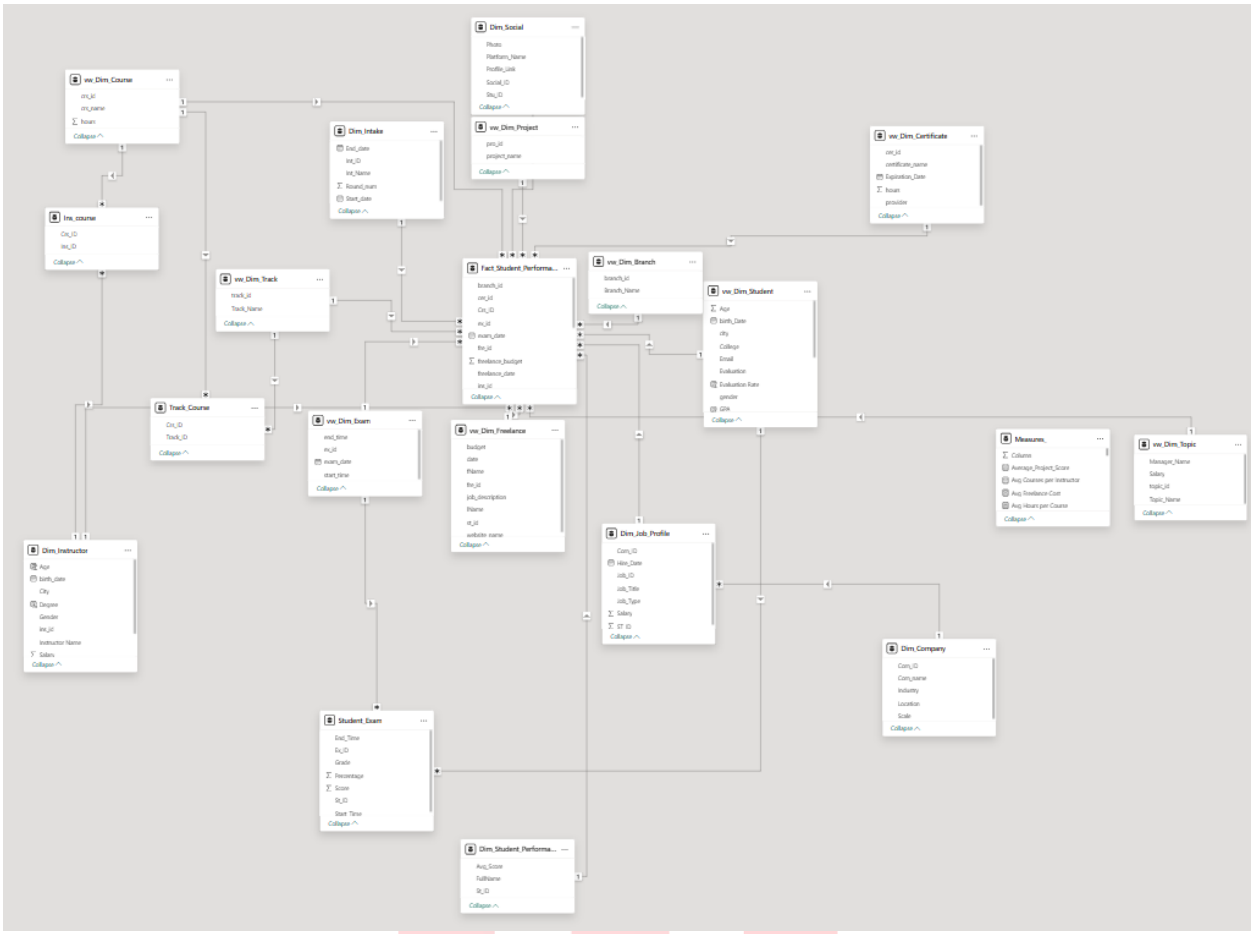
Max Salary of Topic Manager 50K
Max Salary of Job Profile 51K
Max Salary of Instructor 36K

No. Students By Top 5 Topic
315 215 145 105 60
Embedded Systems; UI/UX Design; Business development; Web Developme...; Mobile Developme...

No. Students By Top 3 Branch
90 80 70
Smart Village Branch; New Capital Branch; Cairo University Branch

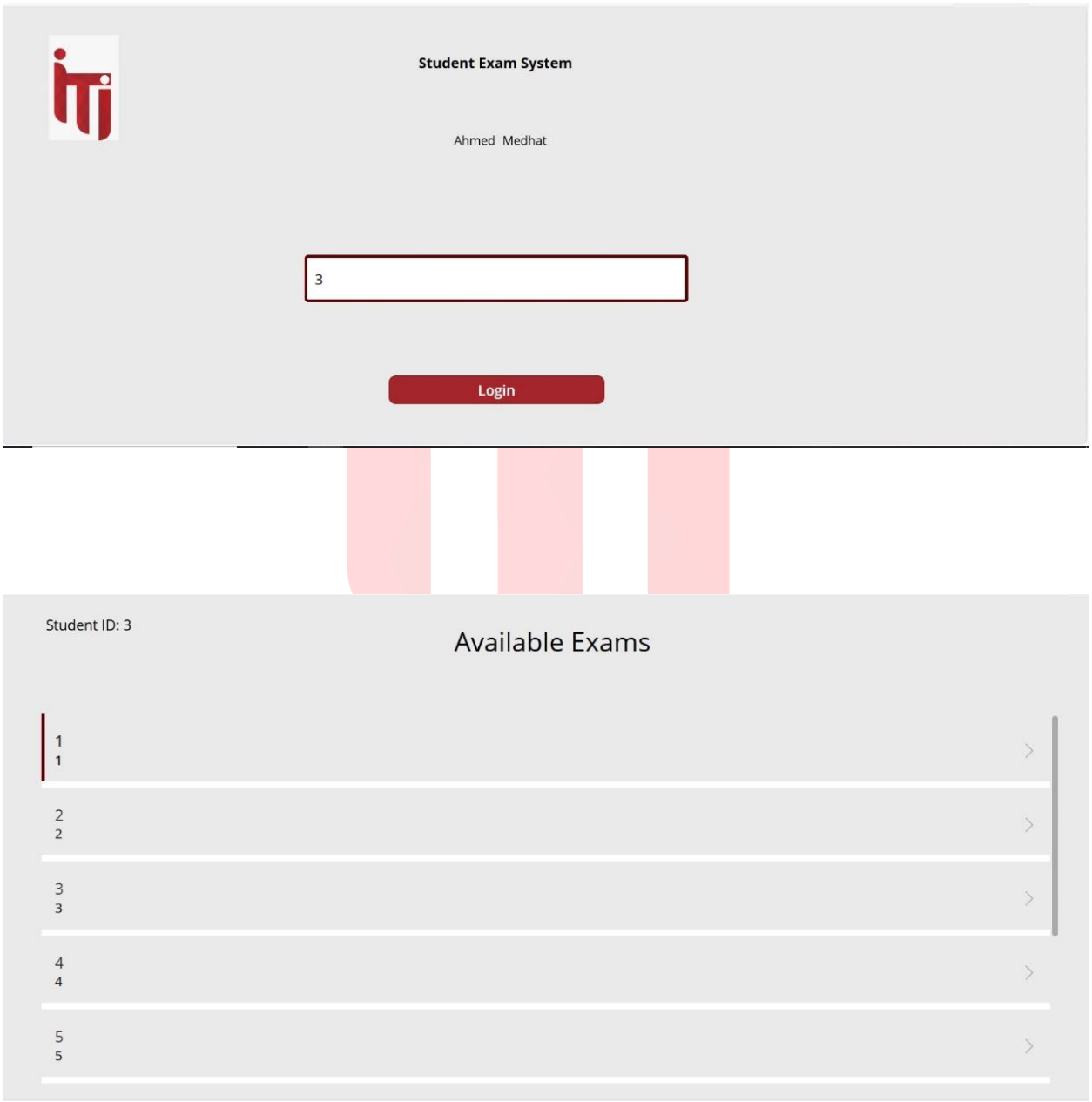
No. Students By Top 8 Freelance
13 13 12 12 11 11 11 11
Create marke... graphics for Ins...; Offer virtual assista... servic...; Integr... payme... gateway and us...; Provide transc... and captio...; Build a small NLP pipeli...; Create a native Android app wi...; ETL pipeline: extract from ...; Run a Faceb... ad campa...

No. Students By Top 5 Track
50 50 45 45 40
Business Analysis and Intelligent Automation...; Embedded Systems; Social Media Marketing; Wireless Communica...; Digital IC Design


Dim_Social
vw_Dim_Course
vw_Dim_Project
vw_Dim_Certificate
Dim_Intake
Ins_course
vw_Dim_Track
Fact_Student_Performa...
vw_Dim_Branch
vw_Dim_Student
Track_Course
vw_Dim_Exam
vw_Dim_Freelance
Measures_
vw_Dim_Topic
Dim_Job_Profile
Dim_Instructor
Dim_Company
Student_Exam
Dim_Student_Performa...

Web App

The Power Apps Examination System enables students to take exams digitally and submit answers through a simple interface. It connects to SQL Server for secure, real-time data handling. Power Apps was chosen for its fast development, mobile support, and Power BI integration.



Submit Exam

Which of the following creates a new table?

- (●) CREATE TABLE
- ( ) ADD TABLE

Which SQL statement is used to rename a table?

- ( ) ALTER RENAME
- (●) RENAME TABLE

Score: 3 / 10 30% ✗ Try Again!

true

CREATE TABLE

false

RENAME TABLE

Score: 3 / 10 30% ✗ Try Again!

false

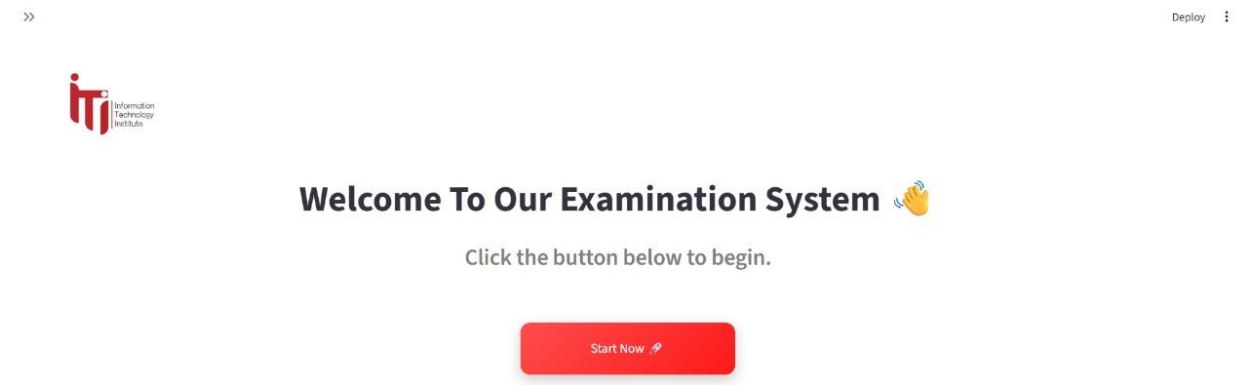
HAVING

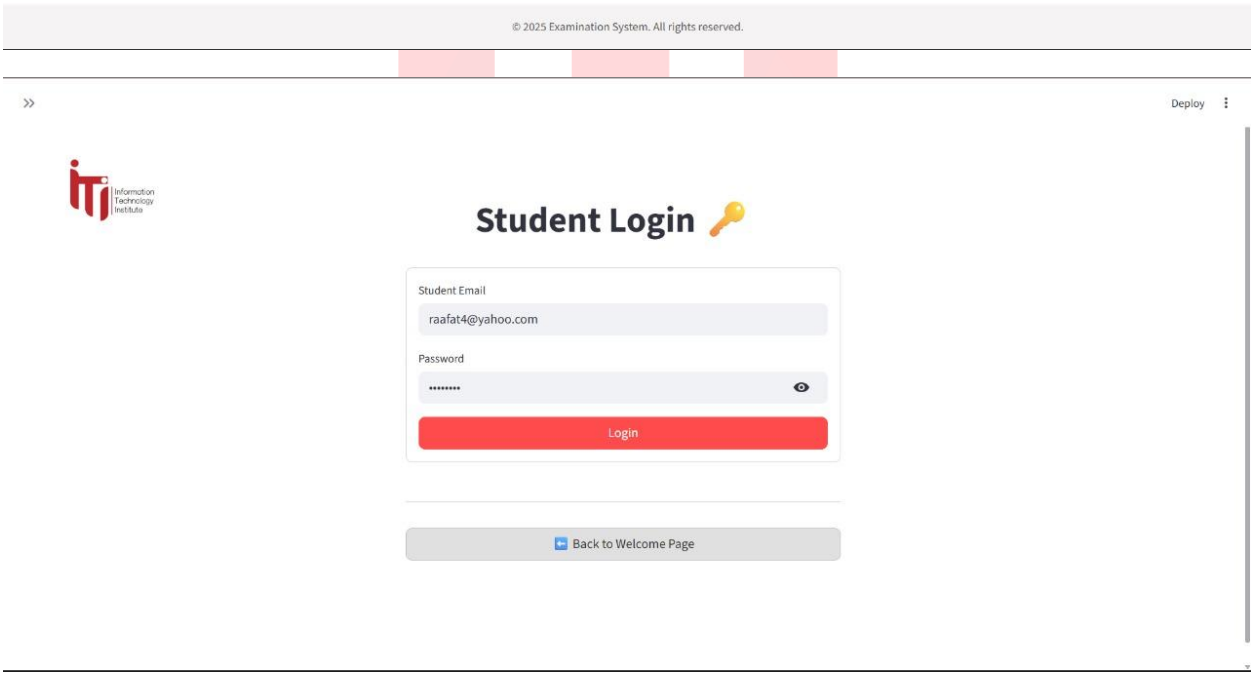
true

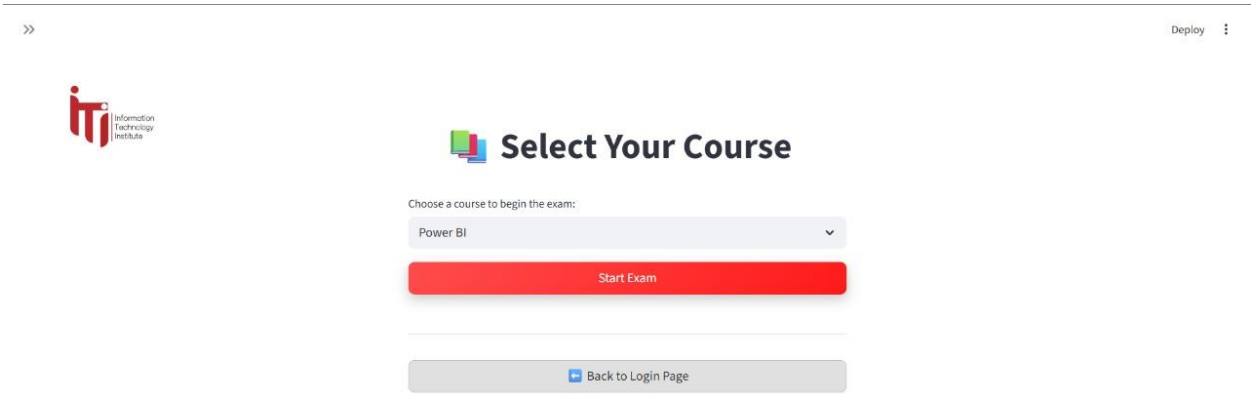
DELETE

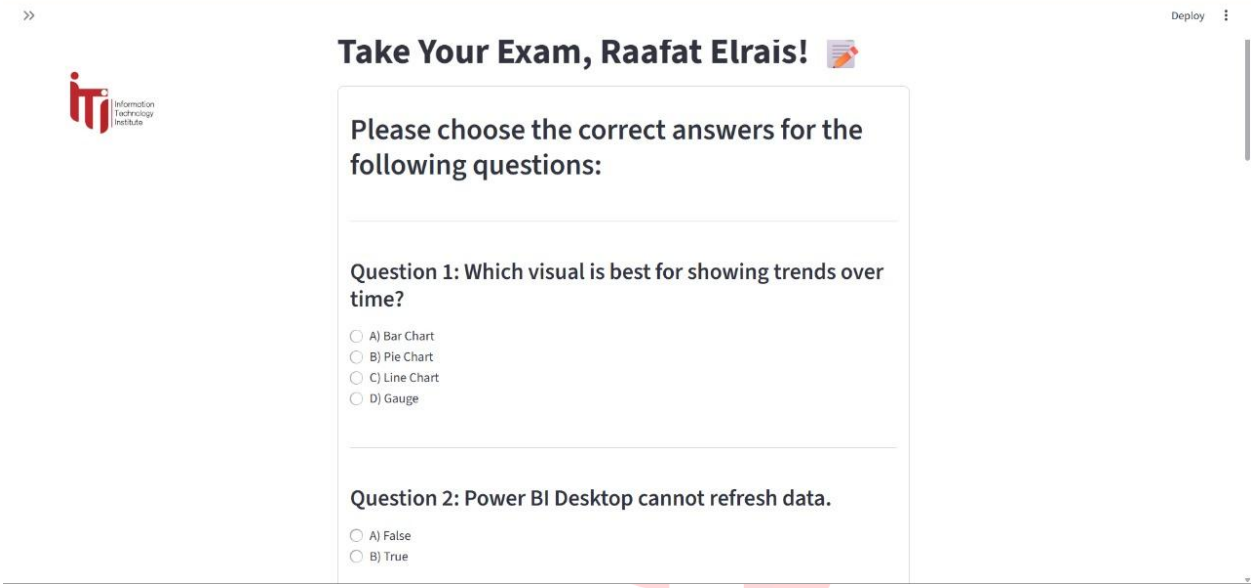
Streamlit Web App

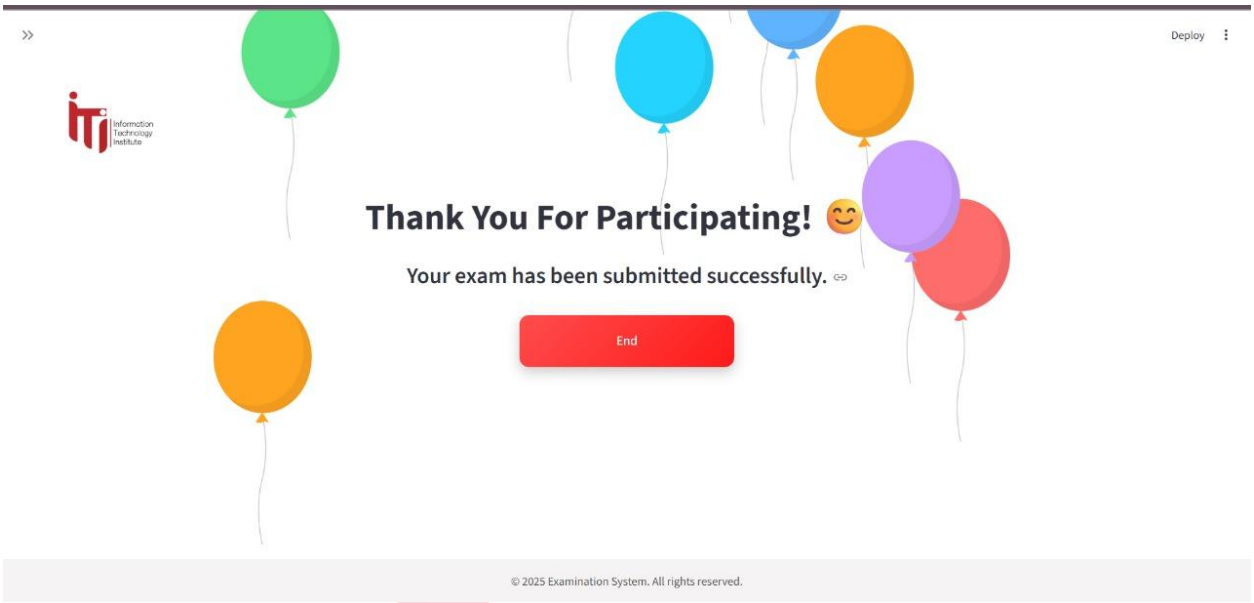
We developed a multi-page Streamlit web application to serve as a user-friendly interface for our SQL-based examination system. The application features a custom, professional UI and seamlessly navigates students through the entire exam process. It robustly connects to our database to execute stored procedures, automating everything from dynamic question generation to the instant correction and display of the final score.






Deploy
Select Your Course
Choose a course to begin the exam:
Power BI
Start Exam
Back to Login Page


Deploy
Take Your Exam, Raafat Elrais!
Please choose the correct answers for the following questions:
Question 1: Which visual is best for showing trends over time?
A) Bar Chart
B) Pie Chart
C) Line Chart
D) Gauge
Question 2: Power BI Desktop cannot refresh data.
A) False
B) True



© 2025 Examination System. All rights reserved.

Obstacles & Problems

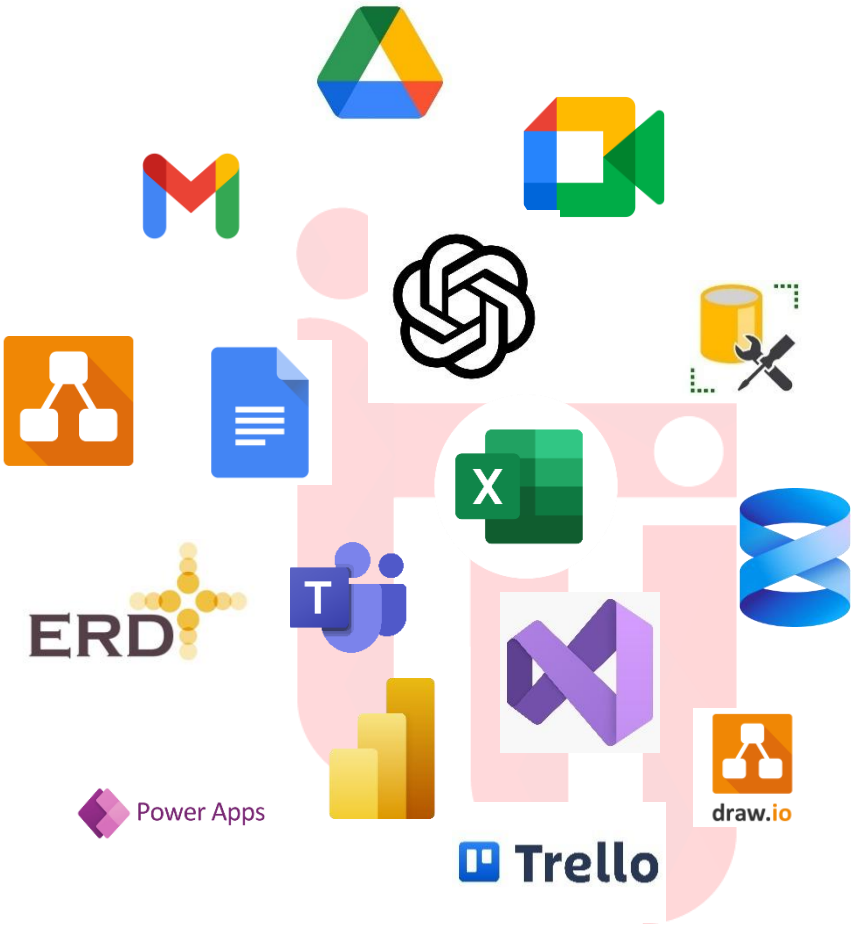
Throughout the development of our examination system, we encountered several technical, logical, and analytical challenges that required careful troubleshooting and collaboration to resolve.

- **Foreign Key Constraint Errors:**
  One of the main issues arose when attempting to delete records from certain tables, as the database prevented deletion due to foreign key constraints. To resolve this, we modified some relationships to include **ON DELETE CASCADE** to allow automatic removal of related records, while in other cases we used **ON DELETE SET NULL** to maintain data integrity without losing important information.

- **Data Duplication and Uniformity:**
  Another challenge was identified during the data analysis phase, where we discovered that the number of students per track and branch was identical across all records. This inconsistency affected the accuracy of Power BI visualizations and analytical results. We resolved this issue by revising and regenerating the dataset, ensuring more realistic and varied data distribution for accurate reporting and insights.

- **Finding a Suitable Data Model:**
  One of the most challenging tasks was determining the most appropriate data model for our business requirements. We had to analyze whether a **Star Schema, Snowflake, or Galaxy** model would best represent our entities and relationships. After several trials and adjustments, we finalized a structure that effectively supported both the database operations and Power BI analysis.

Tools Used



Our Team

Ahmed Medhat
BI Developer

Yousif Sabal
BI Developer

Raafat Elrays
BI Developer

Youssef Ashraf
BI Developer

Omar Mohamed
BI Developer