

# **EXAMINATION SYSTEM**



# **Our TEAM**



**RAAFAT ELRAIS**



**OMAR MOHAMMED**



**YOUSIF SABAL**



**YOUSIF ASHRAF**



**AHMED MEDHAT**

# Table of Contents

---

- Problem Statement
- Objectives
- ER-Diagram & Schema
- Database implementation
- Stored Procedures
- Streamlit website
- SSRS Reports
- Power BI Dashboards
- Power Apps Exam Screen
- Problems and Solutions
- Used Tools
- Future Plan

# What is **THE PROBLEM?**

---

**Student exam processes are traditionally:**

- **No centralized tracking**
- **Difficult performance measurement**
- **No real-time insights for instructors**
- **No automatic scoring**
- **Difficult to analyze topic weaknesses**

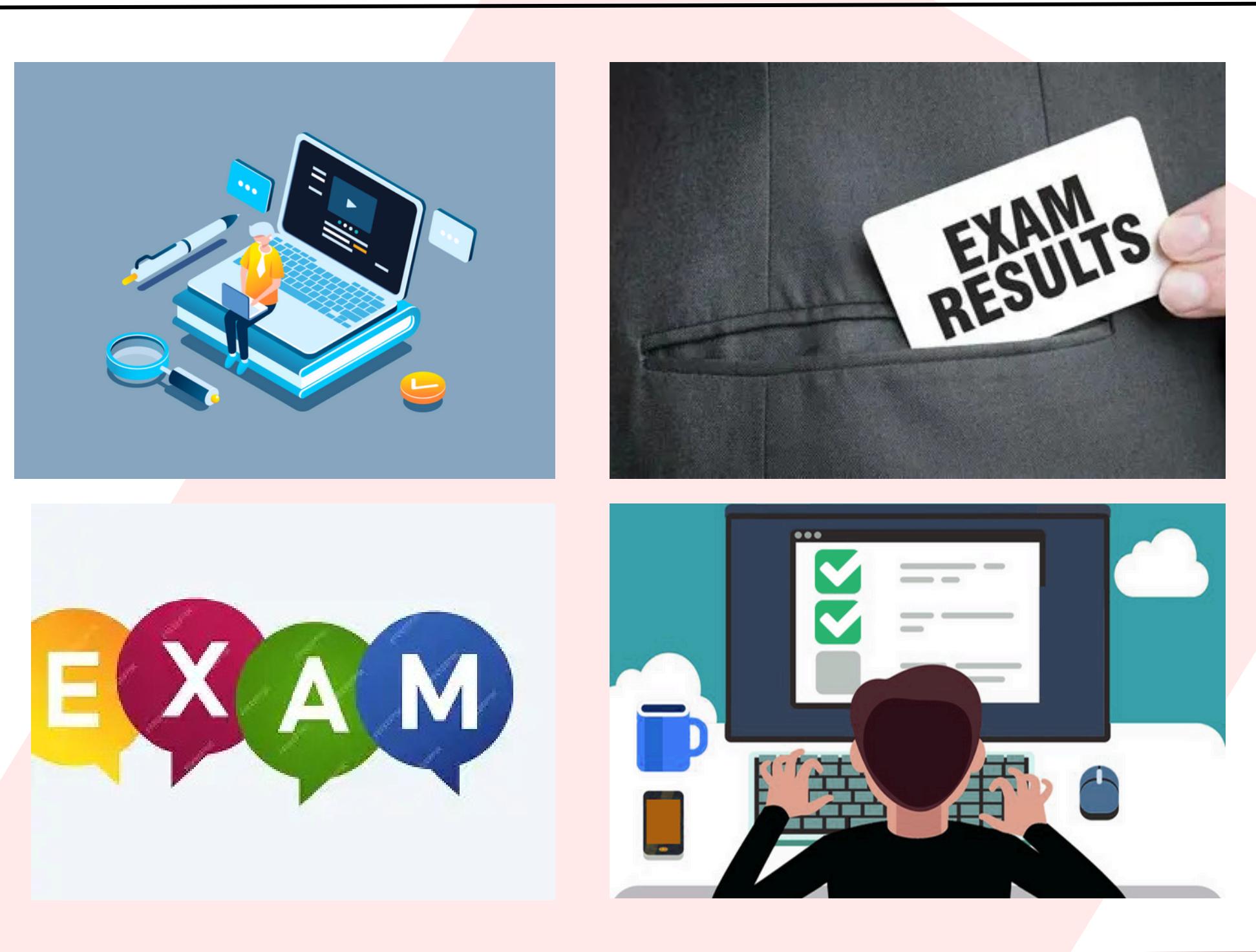
**Academic institutions Suffer from:**

- **Human error in grading**
- **Delayed results**
- **Limited analytics**

# OBJECTIVE

---

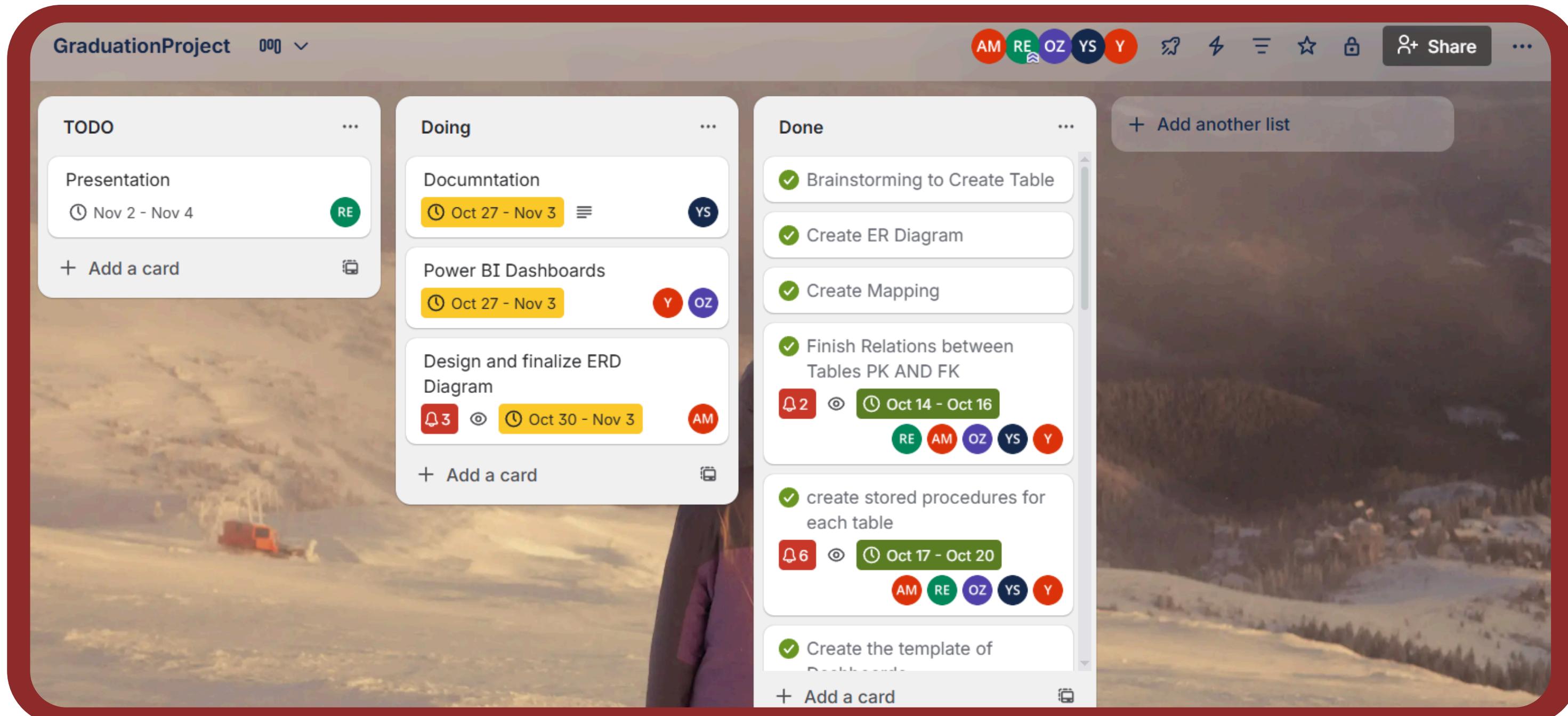
- Digitize the entire student examination process
- Automate grading
- Store answers securely
- Track performance over time



# HOW DO WE MANAGE WORK AMONG US

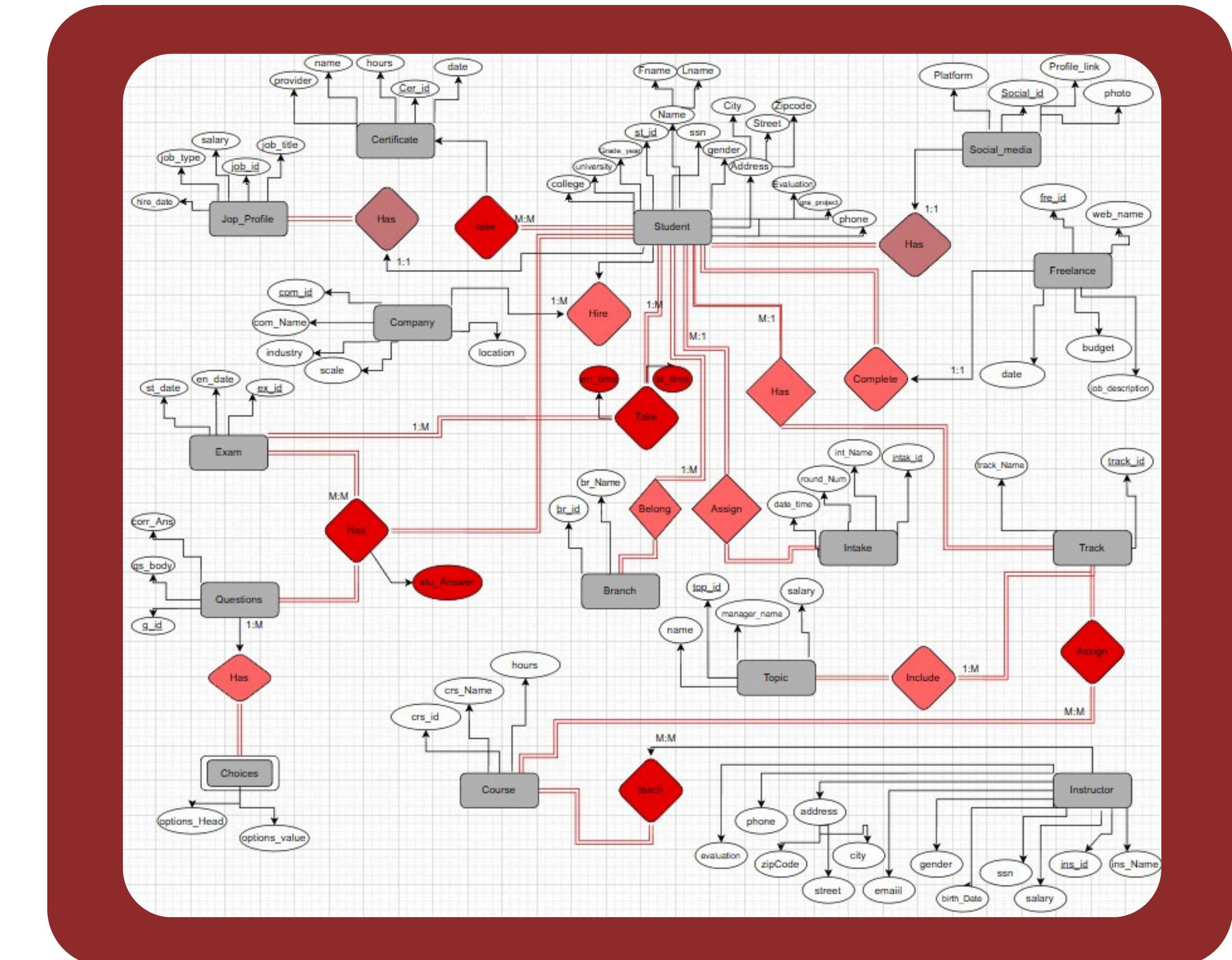
## Methodology:

- Agile (2 Days - Sprint)
- Trello / Jira board



# ER-DIAGRAM

- We designed a relational database with carefully structured tables to ensure data integrity.
- Relationships follow a 1-to-Many model in most areas, such as students to exam answers, and courses to questions.
- Normalization minimized redundancy, while lookup tables improved consistency.



# MAPPING

- 15 Main tables
- 7 M:M TABLES

- **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)

# DATABASE IMPLEMENTATION

1



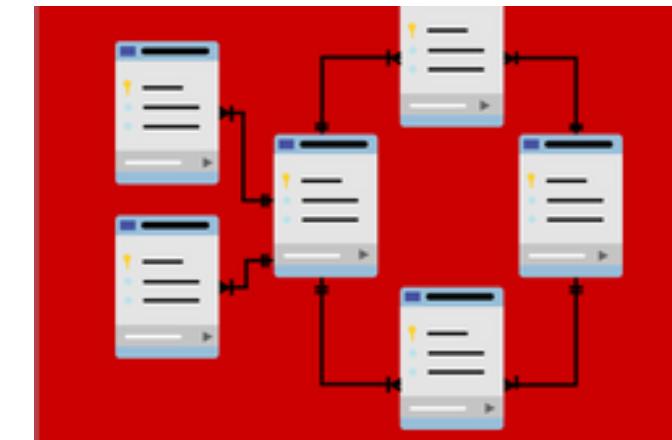
Technology  
Microsoft SQL Server

2



Data Types  
Describe the data types used  
for different fields (e.g., INT,  
VARCHAR, DATE).

3



Schema Diagram  
Outline the schema design,  
including tables, relationships,  
and constraints.

4



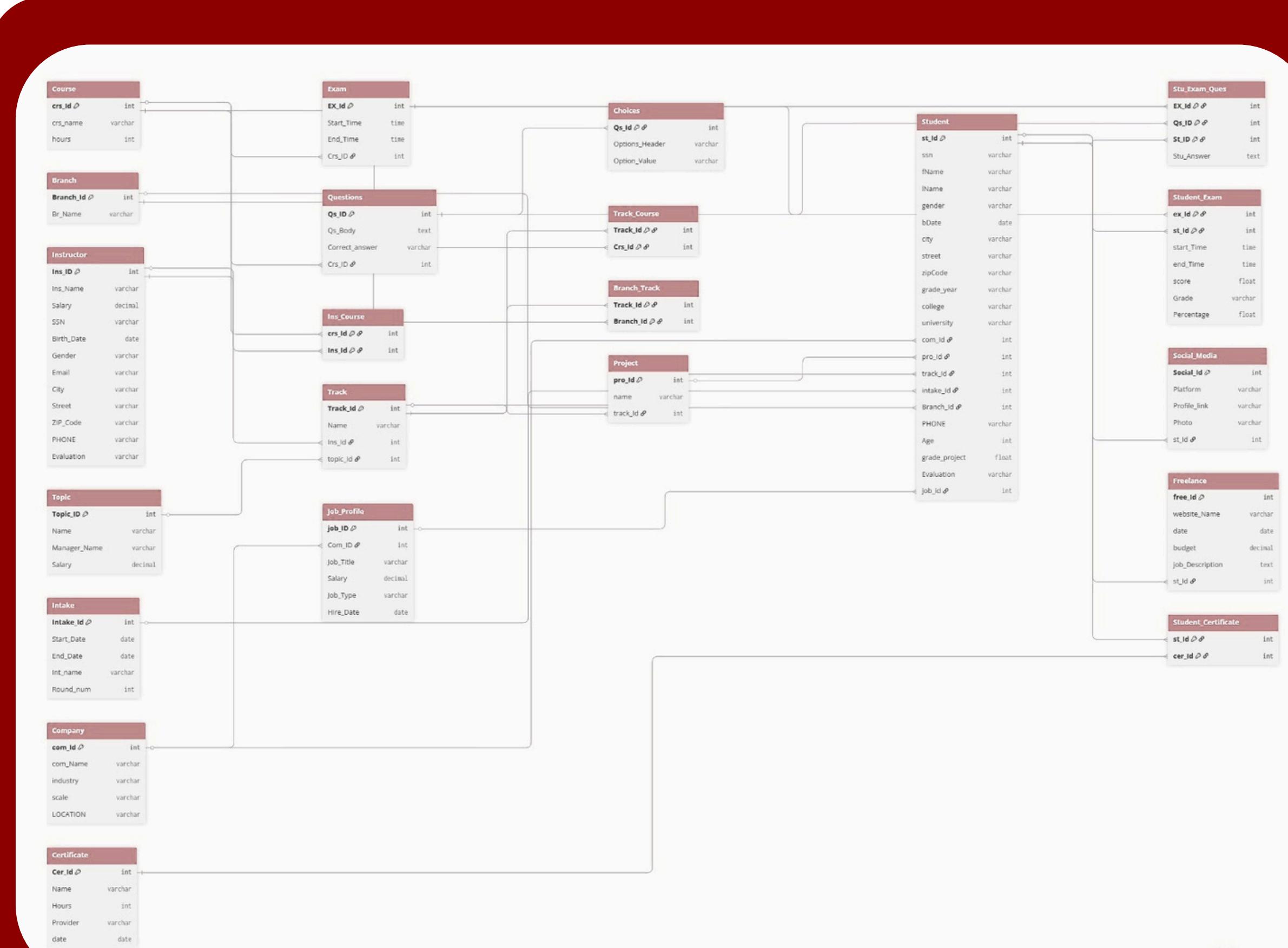
Security  
Encrypted Database Objects  
E.g: SP, Views

5



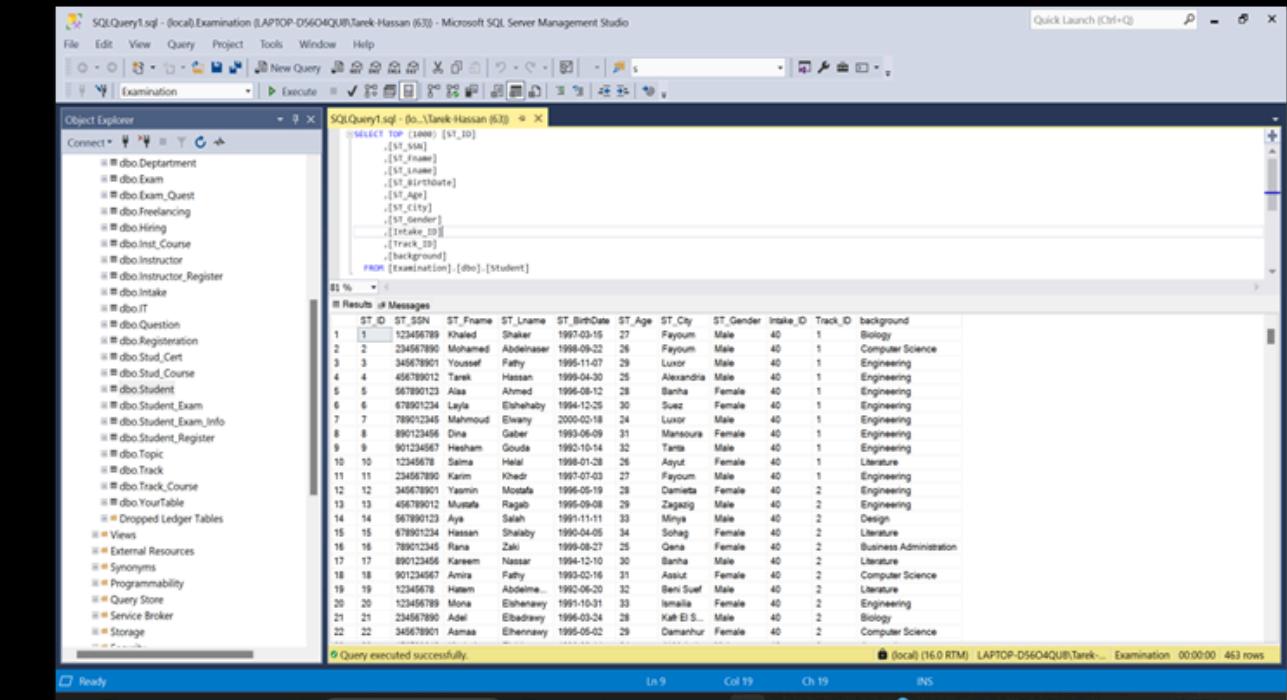
Backup & Recovery  
Daily Backup

# DATABASE DIAGRAM



# DATA FILL

- AI-Generated  
**(Claudera- Gemini - Mockaro - GPT)**



A screenshot of Microsoft SQL Server Management Studio (SSMS) showing a query results grid. The query is:

```
SELECT TOP (100) [ST_ID]
      ,[ST_SSN]
      ,[ST_Fname]
      ,[ST_Lname]
      ,[ST_BirthDate]
      ,[ST_Age]
      ,[ST_City]
      ,[ST_Gender]
      ,[Intake_ID]
      ,[Track_ID]
      ,[background]
  FROM [Examination].[dbo].[student]
```

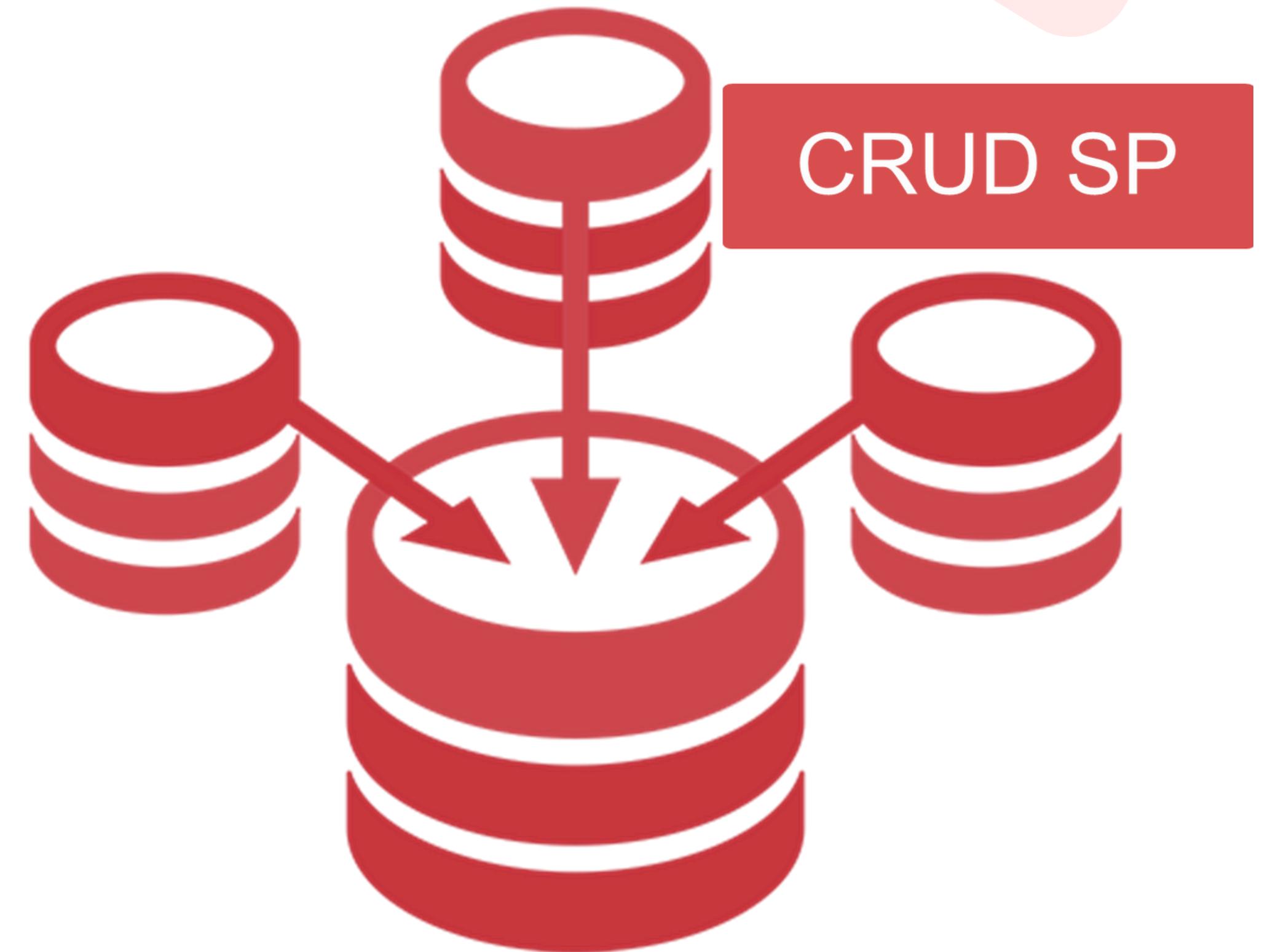
The results grid displays 463 rows of student data. The columns include ST\_ID, ST\_SSN, ST\_Fname, ST\_Lname, ST\_BirthDate, ST\_Age, ST\_City, ST\_Gender, Intake\_ID, Track\_ID, and background. The background column shows values 1, 2, or 3, corresponding to different departments: Biology, Computer Science, Engineering, Literature, Design, and Business Administration.

ST_ID	ST_SSN	ST_Fname	ST_Lname	ST_BirthDate	ST_Age	ST_City	ST_Gender	Intake_ID	Track_ID	background
1	123456789	Khaled	Shaker	1997-03-15	27	Fayoum	Male	40	1	Biology
2	234567890	Mohamed	Abdelnaser	1998-09-22	26	Fayoum	Male	40	1	Computer Science
3	345678901	Yousef	Fathy	1999-11-07	29	Luxor	Male	40	1	Engineering
4	456789012	Ali	Sayed	1998-05-10	31	Alexandria	Male	40	1	Engineering
5	567890123	Alaa	Ahmed	1996-08-12	28	Banha	Female	40	1	Engineering
6	678901234	Layla	Elshehaby	1994-12-25	30	Suez	Female	40	1	Engineering
7	789012345	Mahmoud	Elwayy	2000-02-18	24	Luxor	Male	40	1	Engineering
8	890123456	Dina	Gaber	1993-06-09	31	Mansoura	Female	40	1	Engineering
9	901234567	Hesham	Gouda	1992-10-14	30	Tanta	Male	40	1	Engineering
10	1012345678	Saima	Heal	1998-01-28	22	Aqayd	Female	40	1	Literature
11	1123456789	Khadi	Elshazly	1997-07-10	27	Fayoum	Male	40	1	Engineering
12	12345678901	Yassmin	Mahmoud	1996-05-19	29	Damietta	Female	40	2	Engineering
13	13456789012	Moustafa	Ragib	1995-09-08	29	Zagazig	Male	40	2	Engineering
14	14567890123	Aya	Salah	1993-11-11	33	Minya	Male	40	2	Design
15	15678901234	Hassan	Shalaby	1990-04-05	34	Sohag	Female	40	2	Literature
16	16789012345	Rana	Zaki	1999-08-27	25	Qena	Female	40	2	Business Administration
17	17890123456	Kareem	Nassef	1994-12-10	32	Banha	Male	40	2	Literature
18	18901234567	Amira	Fathy	1993-02-16	32	Assut	Female	40	2	Computer Science
19	19123456789	Hailem	Abdelma	1992-07-01	32	Beni Suef	Male	40	2	Literature
20	20234567890	El-Sherif	El-Sherif	1991-10-31	32	Beni Suef	Female	40	2	Engineering
21	21234567890	Adel	Elbadawy	1996-03-24	28	Kal El S.	Male	40	2	Business
22	22345678901	Asmaa	Ehennawy	1995-05-02	29	Damietta	Female	40	2	Computer Science

# CRUD STORED PROCEDURES

---

**They simplify and secure database operations by encapsulating the logic for Create, Read, Update, and Delete (CRUD) actions for Each Table.**



# CRUD STORED PROCEDURES

---

## Examples

### SP For Selection

```
ALTER procedure [dbo].[SELECT_br_tr]
@Branch_ID INT= NULL,
@Track_ID INT = NULL
AS
BEGIN
SET NOCOUNT ON
SELECT
Branch_ID,
Track_ID
FROM Branch_Track
WHERE
(@Branch_ID IS NULL OR Branch_ID = @Branch_ID)
AND (@Track_ID IS NULL OR Track_ID = @Track_ID)
END
```

### SP For Insertion

```
CREATE PROCEDURE InsertStudent
@SSN NVARCHAR(14),
@Fname NVARCHAR(100),
@Lname NVARCHAR(100),
@Gender NVARCHAR(10),
@Birth_Date DATE,
@City NVARCHAR(50),
@Street NVARCHAR(100),
@ZipCode NVARCHAR(10),
@Phone VARCHAR(11),
@Grade_Year VARCHAR(10),
@Com_Id INT,
@Pro_ID INT,
@Track_ID INT,
@Intake_ID INT,
@Social_ID INT,
@Branch_ID INT,
@College VARCHAR(100),
@University VARCHAR(100),
@Grade_Project VARCHAR(30),
@Evaluation NVARCHAR(20)
AS
BEGIN
if Exists(select 1 from Student
where SSN=@SSN)
begin
print('This Student already exists');
return;
end;
begin try
IF NOT EXISTS (SELECT 1 FROM Student WHERE SSN = @SSN)
BEGIN
INSERT INTO Student
([SSN],[Fname],[Lname],[Gender],[Birth_Date],[City],[Street],
[ZipCode],[Phone],[Grade_Year],[Com_ID],[Pro_ID],[Track_ID],
[Intake_ID],[Social_ID],[Branch_ID],
[College],[University],[Grade_Project],[Evaluation])
VALUES
(@SSN,@Fname,@Lname,@Gender,@Birth_Date,@City,@Street,
@ZipCode,@Phone,@Grade_Year,@Com_Id,@Pro_ID,@Track_ID,
[Intake_ID],@Social_ID,@Branch_ID,
[College],@University,@Grade_Project,@Evaluation)
END
END TRY
BEGIN CATCH
PRINT 'An error occurred while inserting student information';
END CATCH
END
```

# CRUD STORED PROCEDURES

## SP For Deletion

```
ALTER PROCEDURE [dbo].[Delete_Student]
    @St_Id INT
AS
BEGIN
    BEGIN TRY
        DELETE FROM Student
        WHERE St_Id = @St_Id;
        PRINT ('Student deleted successfully')
    END TRY
    BEGIN CATCH
        PRINT ('Cannot delete Student')
    END CATCH
END;
```

## SP For Update

```
@Pro_ID INT,
@Track_ID INT,
@Intake_ID INT,
@Social_ID INT,
@Branch_ID INT,
@College NVARCHAR(100),
@University NVARCHAR(100),
@Grade_Project VARCHAR(30),
@Evaluation NVARCHAR(20)
AS
BEGIN
    UPDATE Student
    SET
        [Fname] = @Fname,
        [Lname] = @Lname,
        [Gender] = @Gender,
        [Birth_Date] = @Birth_Date,
        [City] = @City.
```

# EXAM STORED PROCEDURES

---

## 1 Add New Exam

```

ALTER 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;

```

## 2 Register New Student

```

ALTER proc [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;

```

# EXAM STORED PROCEDURES

---

## 3 Generate Random Questions

```

@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;

```

## 4 Submit Answers and Calculate Scores

```

]ALTER 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;

```

# EXAM SP CHALLENGES

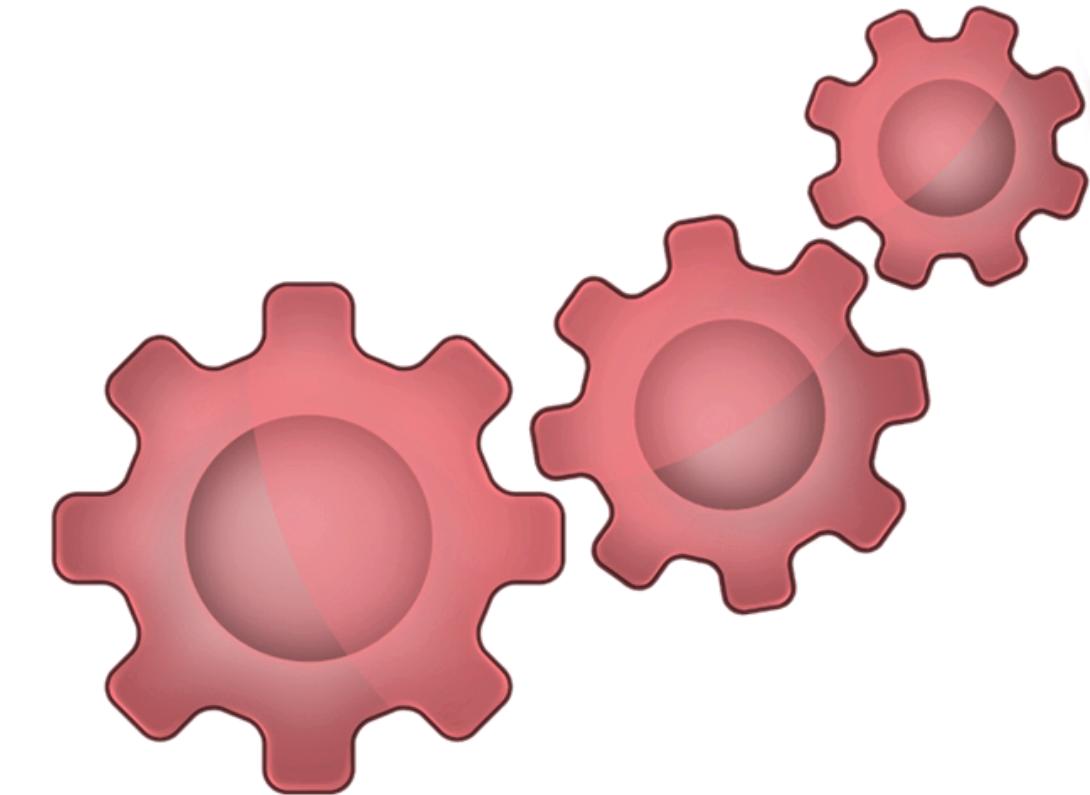
---



**Handling Random  
Question Generation**



**Mapping Answers  
to the Correct  
Question Order**



**Preventing Multiple  
Exam Submissions**

# WEB APPLICATION

---

**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 .**



# WEB APPLICATION

The screenshot shows the 'Student Login' page. At the top, there is a logo for 'Information Technology Institute'. Below it, the heading 'Student Login' is displayed with a key icon. There are two input fields: 'Student Email' containing 'raafat4@yahoo.com' and 'Password' with a masked value. A red 'Login' button is positioned below these fields. At the bottom of the page, there is a link 'Back to Welcome Page'.

## Take Your Exam, Raafat Elrais! 📝

The screenshot shows the 'Take Your Exam' interface. It features a question and answer section. The question is: 'Question 1: Which visual is best for showing trends over time?'. The options are: A) Bar Chart, B) Pie Chart, C) Line Chart, and D) Gauge. Below the question, there is a red 'Start Exam' button. To the right, the user's score is displayed as 'Your Score: 70 / 100' and the grade as 'Grade: C (Good)'. At the bottom, there is a red 'End and Return to Welcome Page' button.

Thank You, Raafat Elrais! 😊

# The Process



1

**Exam Initialization:** Upon course selection, the system calls the **Add\_Exam** and **Register\_Student** procedures to create a unique, secure exam session linking the specific student to a new exam record.

2

**Dynamic Question Generation:** The **Generate\_And\_Show\_Exam** procedure is then executed, which dynamically selects 10 random questions from the database relevant to the chosen course, ensuring a unique and fair test for every student.

3

**Interactive Answering :** The 10 questions and their choices are displayed in a clean web interface. After the student makes their selections, all answers are securely packaged and sent to the database upon submission.

4

**Automated Scoring & Results:** Finally, the **SubmitExamAnswers** procedure is called. It instantly corrects the submitted answers, calculates the final score, saves it to the Student\_Exam table, and the result is immediately retrieved and displayed to the student.

## The Process

---

# **SQL SERVER REPORT SERVICES (SSRS)**

**Using SSRS in Visual Studio, we created interactive reports that transform raw database data into clear, actionable insights for students, instructors, and administrators.**



# SSRS

- Report that returns the students information according to Department No parameter. (**GetStudentTopic**)

Power BI Report Server Home > Reports\_ > Students Info

Topic Id 2

1 of 2 ? 100% |

ITI Information Technology Institute

## Students Info

### Software Engineering

Name	Gender	City	Age	College	University	Grade Year
Mohamed Sayed	M	Suez	21	Faculty of Engineering	Tanta University	2023
Youssef Ibrahim	M	Mansoura	24	Faculty of Media & Mass Comm	Luxor University	2022
Amira Hussein	F	Mansoura	21	Faculty of Computer & Information Technology	Beni-Suef University	2024
Zainab Younis	F	Alexandria	23	Faculty of Business	Aswan University	2023
Mariam Sayed	F	Assiut	26	Faculty of Computer & Information Technology	Al-Azhar University	2020

# SSRS

•Report that takes course ID and returns its topics.

Power BI Report Server Home > Reports\_ > Courses By Topic

crs id 14

|< 1 of 1 >| 100%

Topic Name
AI
Business development
Embedded Systems

## Courses By Topic

### Big Data

Topic Name
AI
Business development
Embedded Systems

## (GetTopicbyCourses) Procedure

•Report that takes exam number and returns the Questions in it and choices [freeform report]

Power BI Report Server Home > Reports\_ > Exam's Questions

Exam ID 15

|< 1 of 1 >| 100% |

ITI 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()

## (GetQuestions) Procedure

# SSRS

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

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

# (GetStudentGrade) Procedure

# **(GetInstructorInfo) Procedure**

# SSRS

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

The screenshot shows a Power BI Report Server interface. At the top, there is a navigation bar with a yellow bar icon, followed by the text "Power BI Report Server" and "Home > Reports\_ > Student's Answers". Below the navigation bar, there are two input fields: "Exam ID" with the value "1" and "Student ID" with the value "2". Underneath these fields is a set of navigation controls including arrows for page navigation, a magnifying glass for search, and a refresh symbol. To the right of these controls is a dropdown menu set to "100%".



## Student's Answers

### (GetStudentAnswers) Procedure

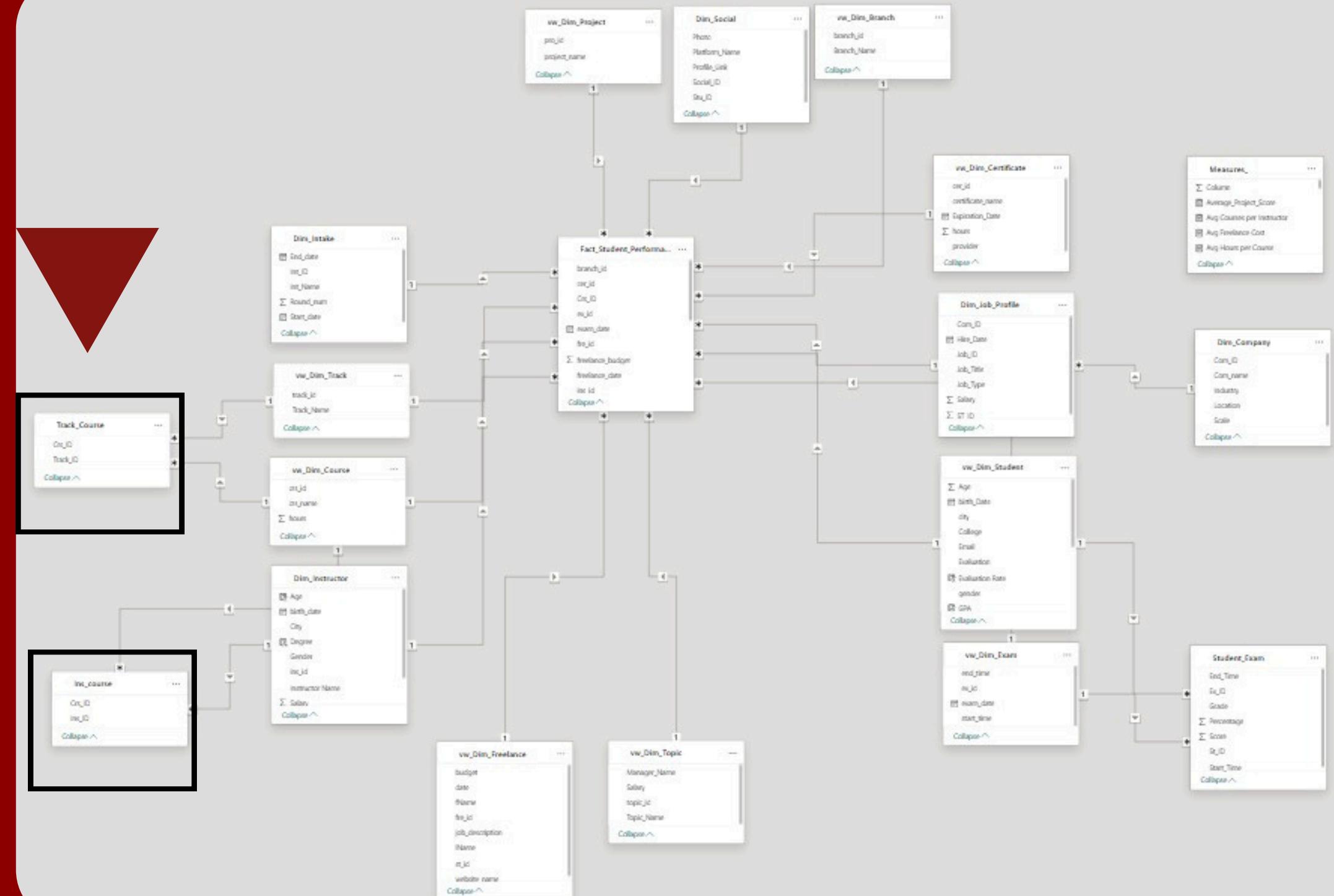
Exam ID: 1 | Student ID: 2

Qs Body	Student Answer
Which clause is used to filter records in SQL?	A
Which constraint ensures a column cannot have NULL values?	A
Which clause groups rows that have the same values?	A
What is the main key in a table called?	A
Which SQL function gives the current date?	A
Which command changes data in an existing table?	B

# Data Model

- THE SYSTEM USES A **STAR SCHEMA** WITH A CENTRAL **FACT\_STUDENT** **A** TABLE STORING KEY STUDENT METRICS. IT CONNECTS TO MULTIPLE DIMENSION TABLES, SUCH AS STUDENT, INSTRUCTOR, COURSE, EXAM, AND TRACK, AND PROVIDES CONTEXT FOR FILTERING AND ANALYSIS.
- A **BRIDGE TABLE** IS INCLUDED TO PROPERLY HANDLE **MANY-TO-MANY** RELATIONSHIPS AND ENSURE ACCURATE REPORTING. A DEDICATED MEASURES TABLE ORGANIZES DAX CALCULATIONS FOR BETTER PERFORMANCE AND MAINTENANCE.
- THIS DESIGN DELIVERS **FAST QUERIES**, **FLEXIBLE FILTERING**, AND A **SCALABLE FOUNDATION** FOR BUSINESS INTELLIGENCE.

# Star Schema



# **POWER BI DASHBOARD**

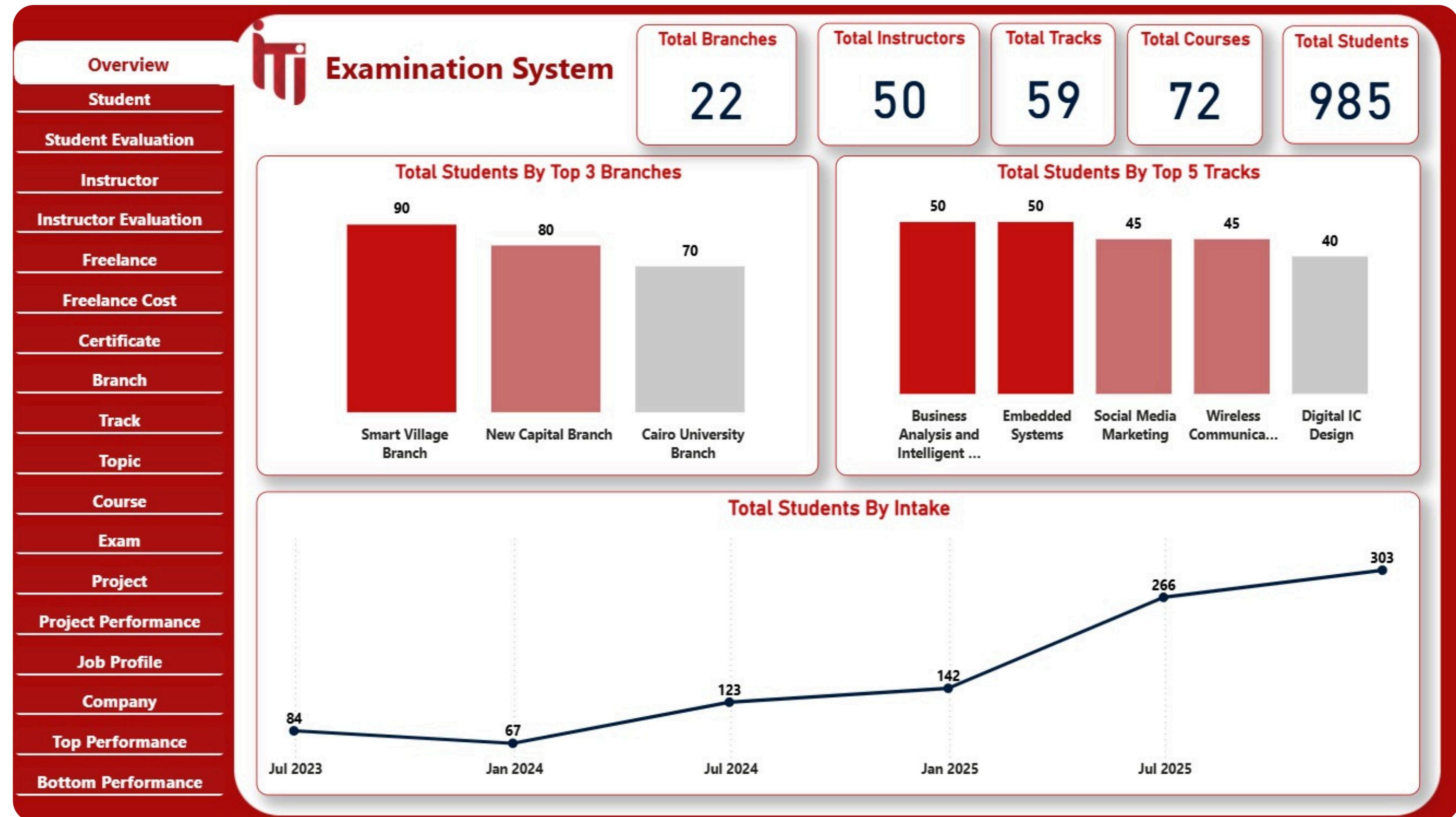


Power BI

# POWER BI DASHBOARD

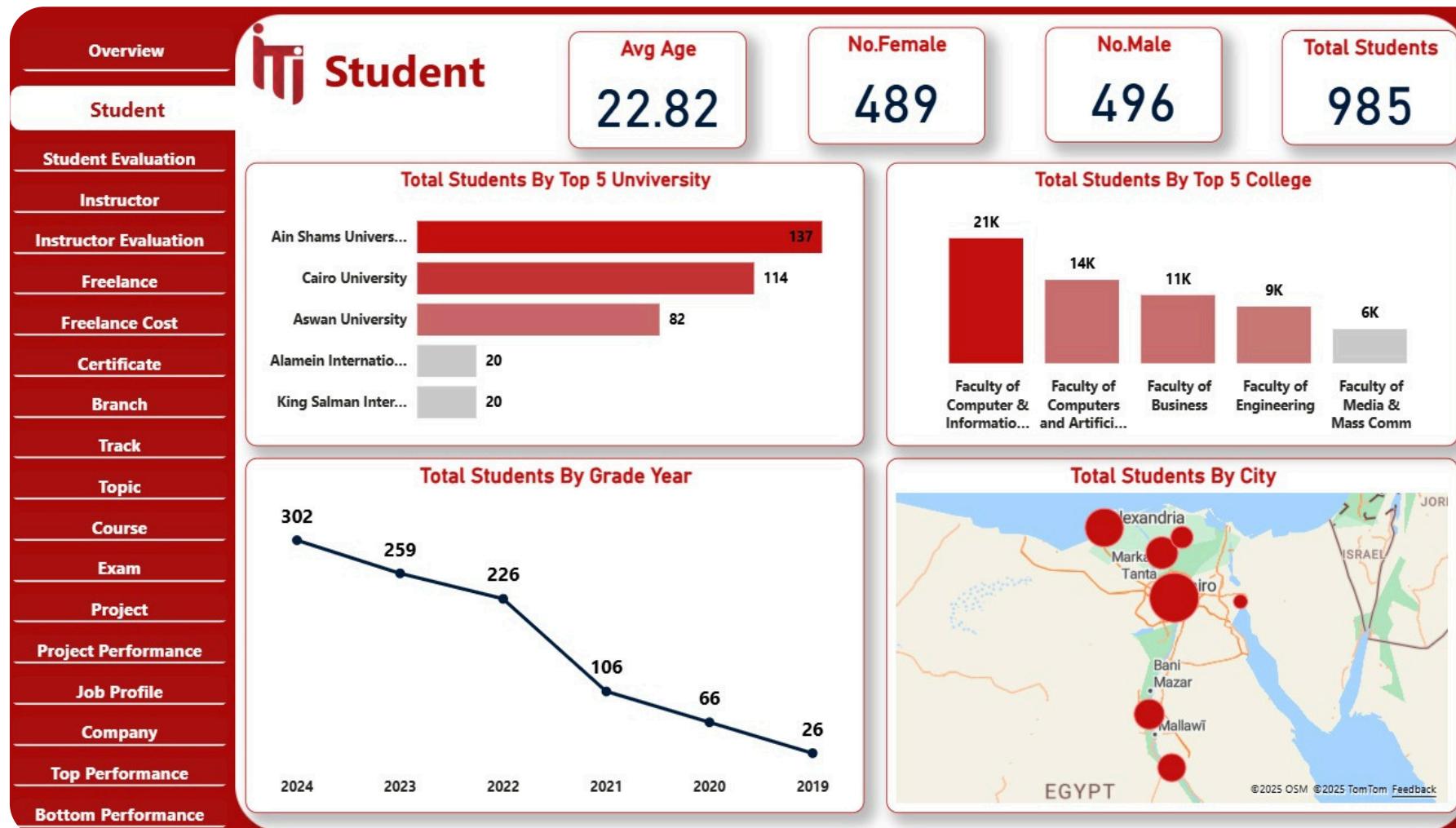
1

## OVERVIEW

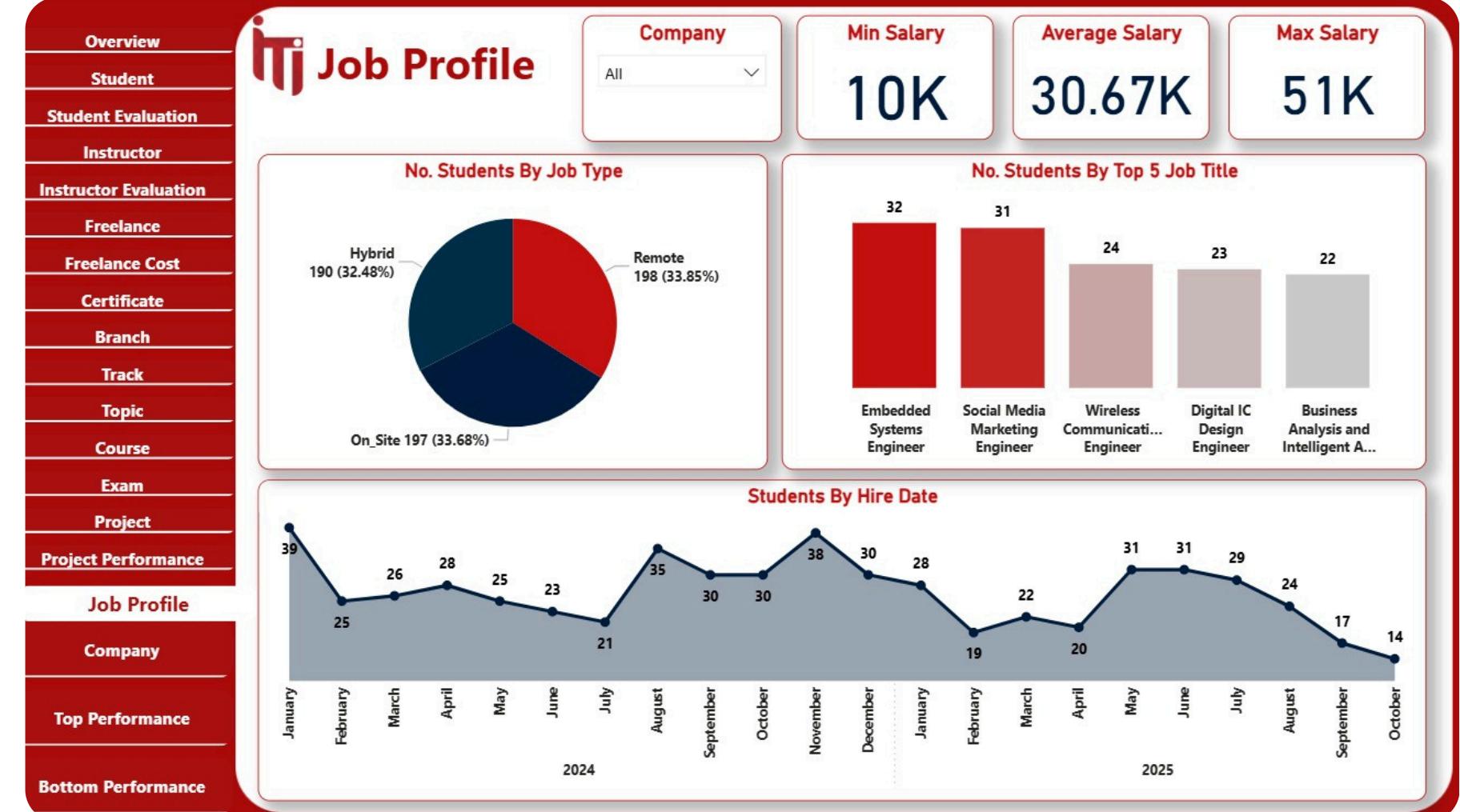


# POWER BI DASHBOARD

## 2 STUDENT

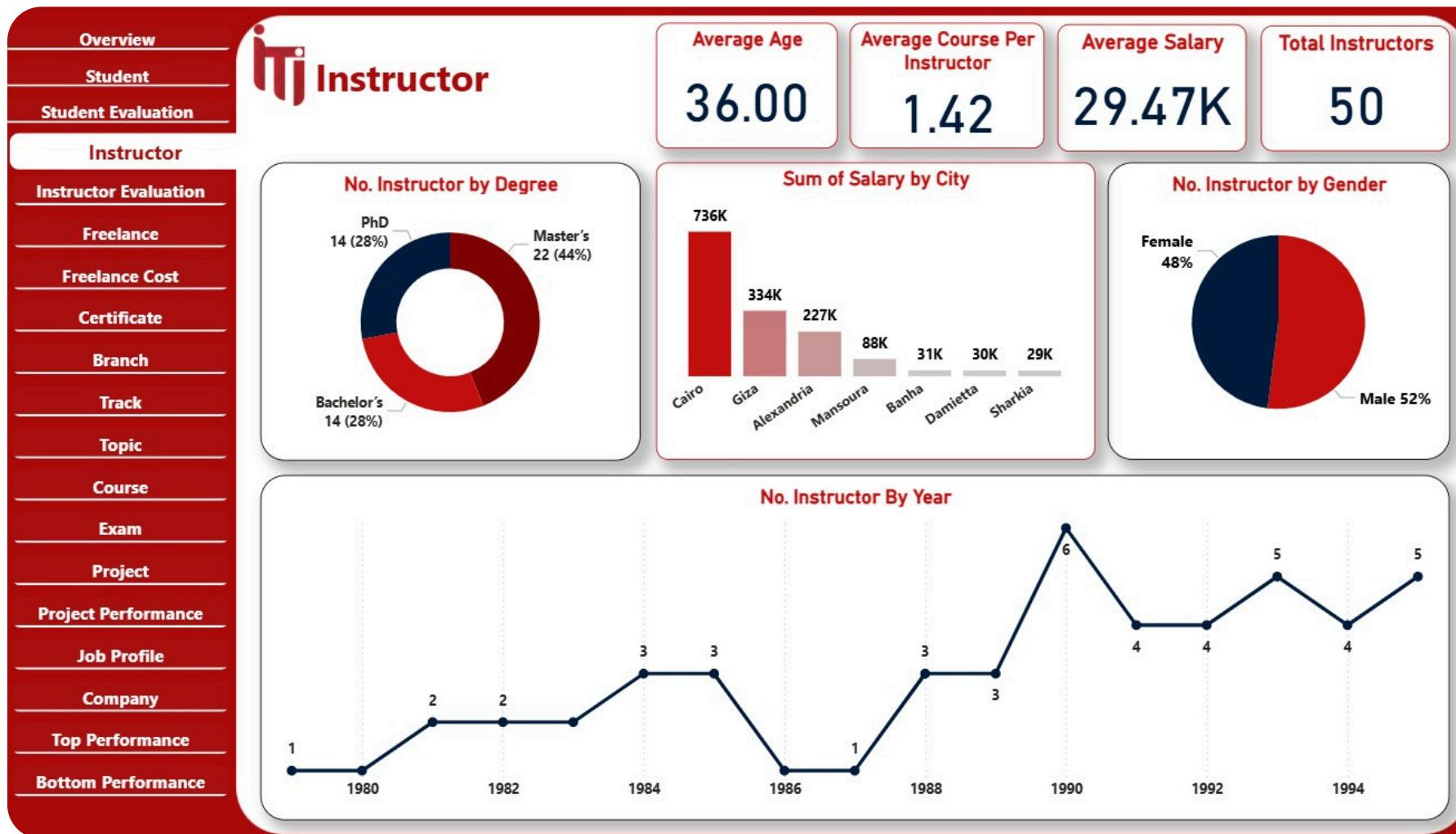


## 3 JOB PROFILE

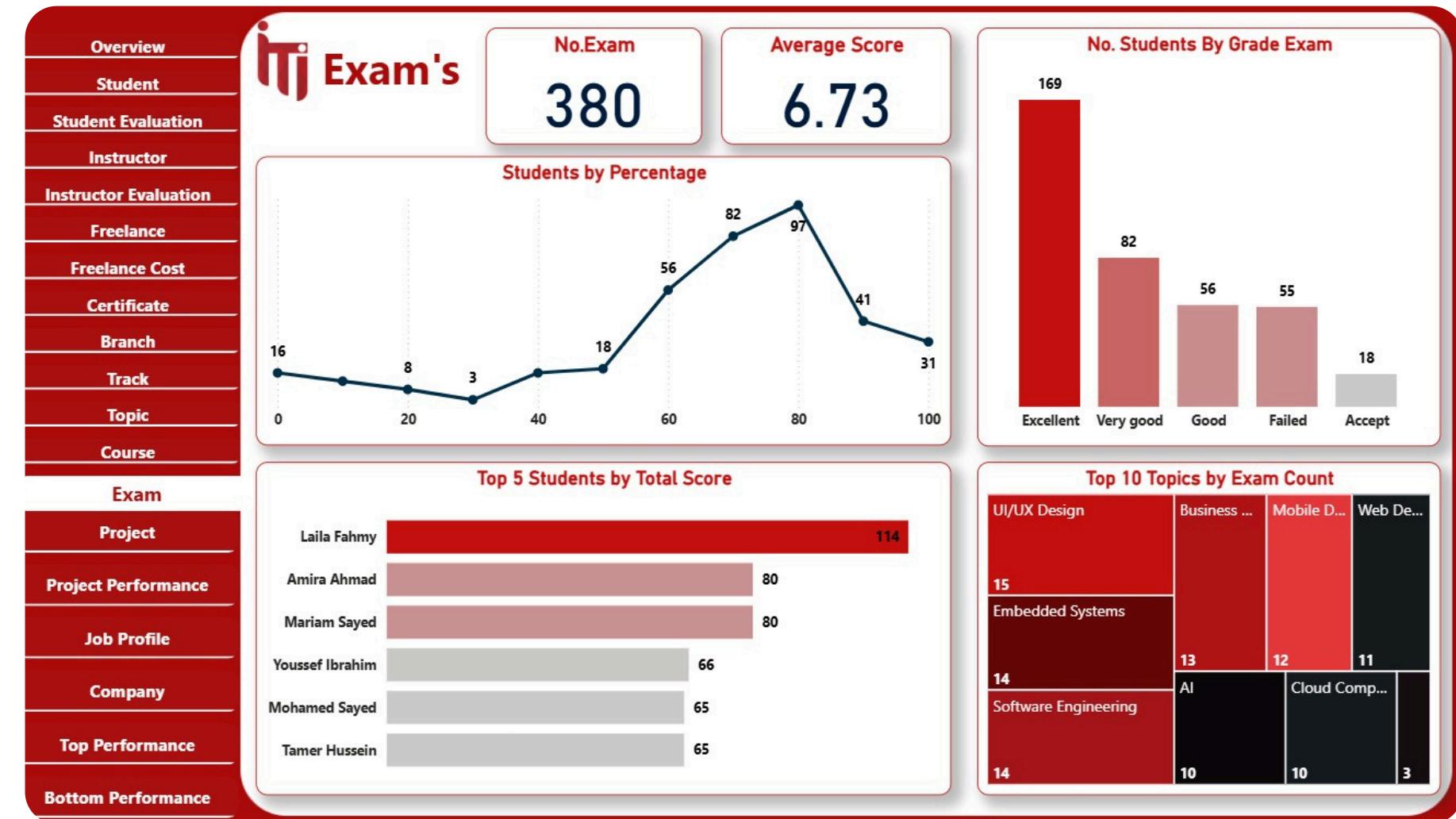


# POWER BI DASHBOARD

## 4 INSTRUCTOR



## 5 EXAM



# POWER APP

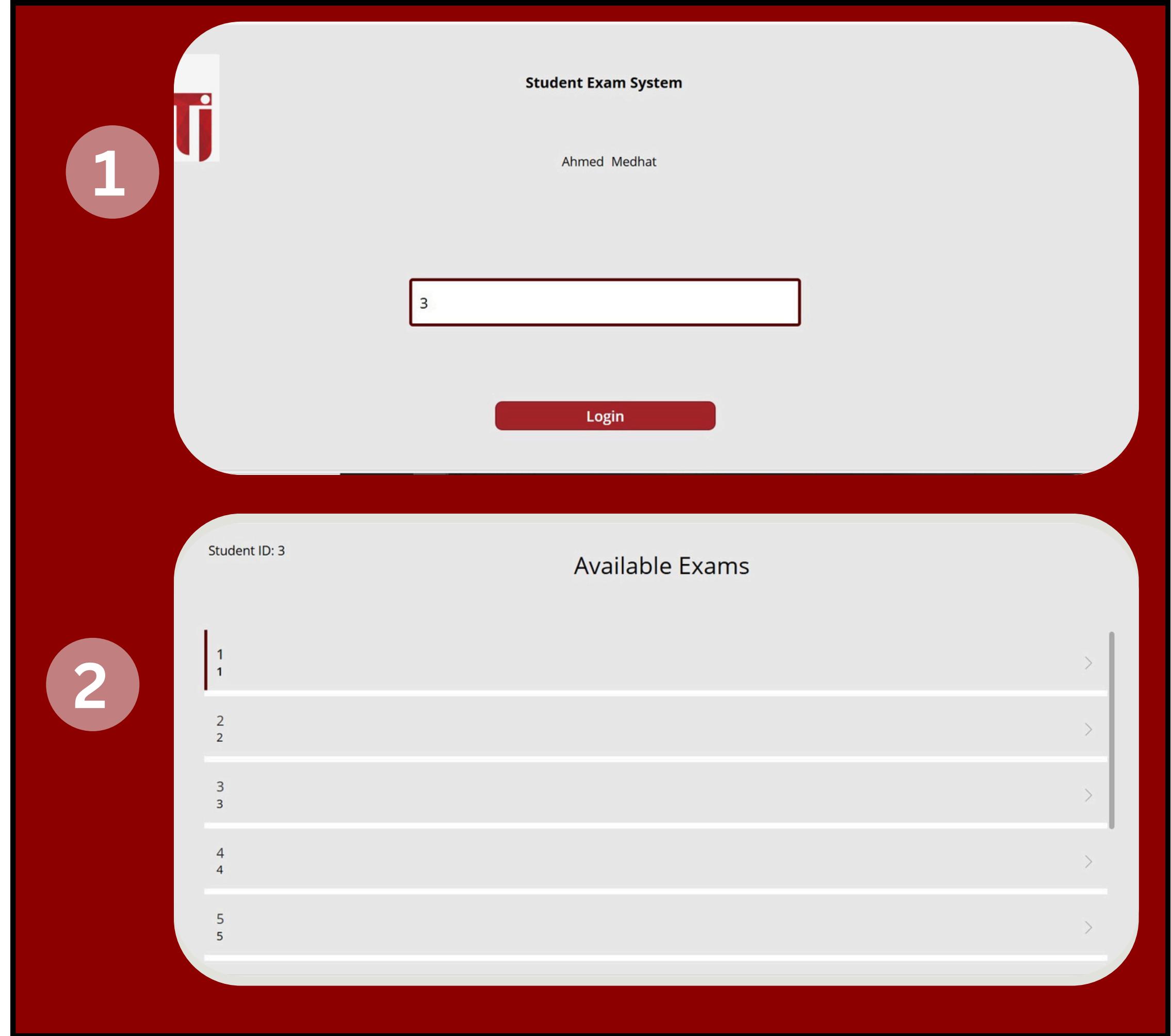
---

**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.**



Power Apps

# The System Flow



Submit Exam

3

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

4

Score: 3 / 10

30%

✗ Try Again!

true

**CREATE TABLE**

false

**RENAME TABLE**

## The System Flow

# Problems and Solutions

---

## Problems

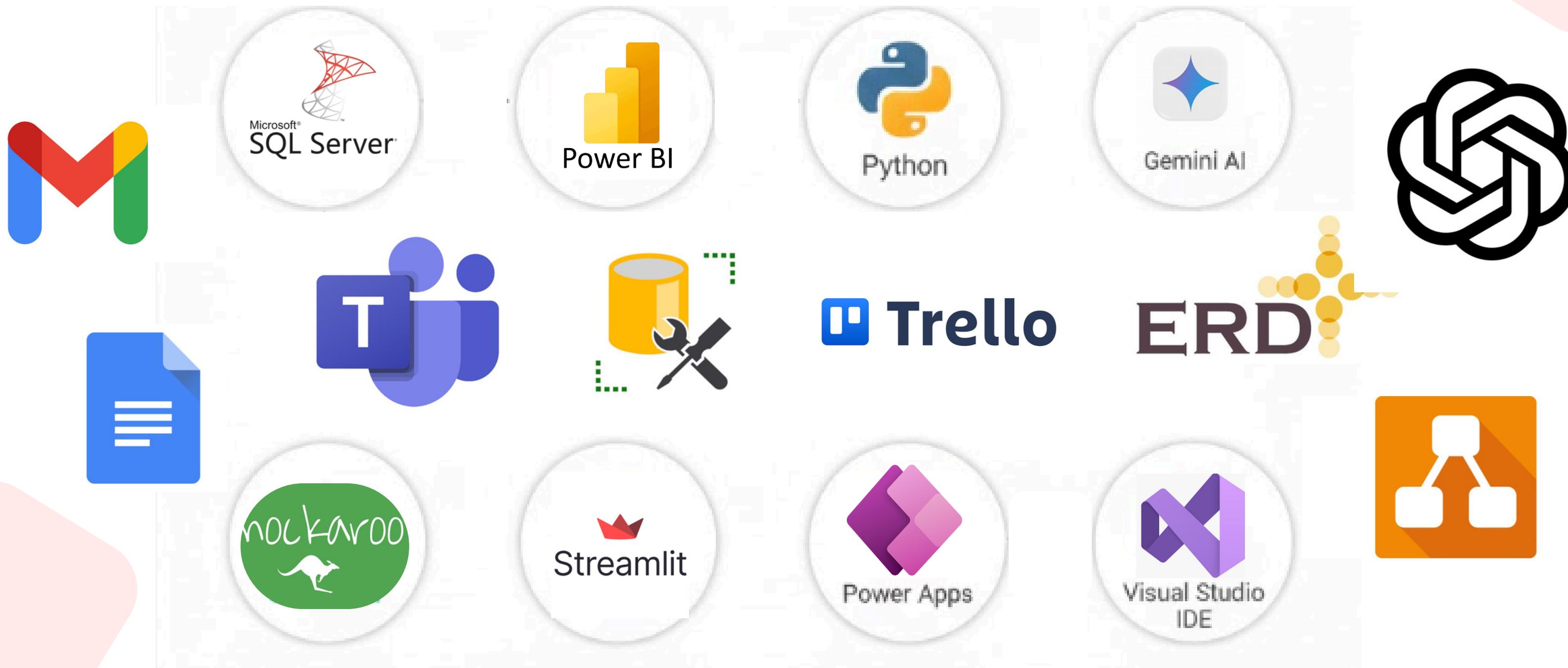
- Unbalanced data impacting result accuracy.
- Foreign key deletion constraint issue.
- Difficulty choosing optimal data model

## Solutions

- Applied asymmetric distribution for accuracy.
- Applied cascade and null strategies.
- Implemented efficient star schema design

# Used Tools

---



# Future Plan

---

- Instructor Question Configuration
- Smart Question Bank Expansion
- Adaptive Exam Generation
- Mobile Examination App



# **QUESTIONS ??**

# **THANK YOU**