



Examination System

CairoBranch ITP 2024

By:

Mohamed Soliman

Technologies Used:

- SQL Server
- PowerApps
- PowerBI
- SSIS
- SSRS

DataBase

Entity-Relationship Diagram (ERD) Using [Darw.io](#) :

Steps:

Initial Planning:

- Defined the scope of the ERD.
- Identified the entities, attributes, and relationships to be included.

Accessing draw.io:

- Accessed the draw.io website.

Creating a New Diagram:

- Choose appropriate shapes for entities, attributes, relationships from the shapes provided by draw.io.

Adding Entities:

Defining Attributes:

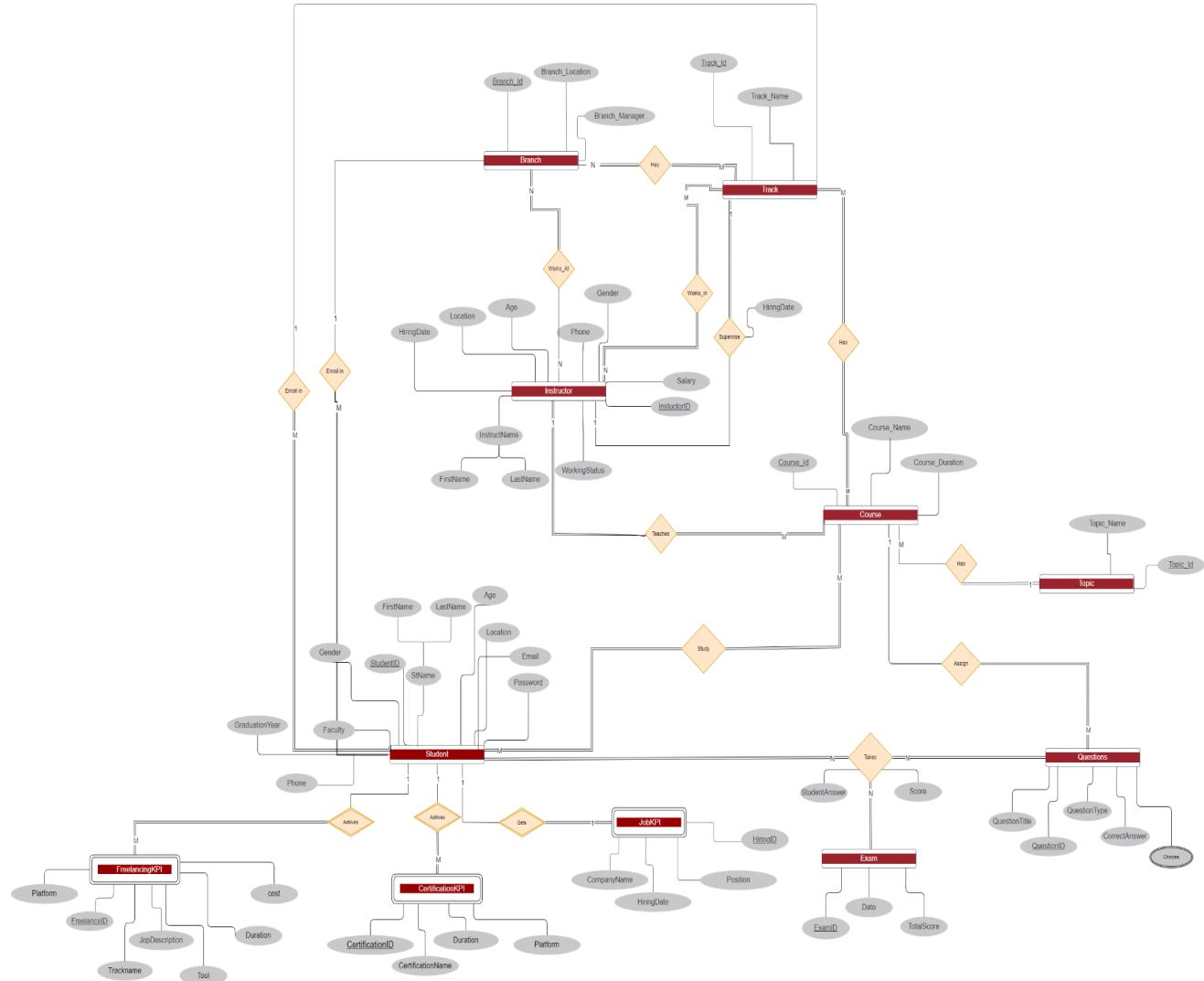
Establishing Relationships:

- Used lines to establish relationships between entities.

Refining the Diagram:

- Organize entities and relationships neatly on the canvas.

Results:



Mapping:

Steps:

Reviewed the ERD:

Identified Entities:

- Review each entity in the ERD and identify them for translation into tables in the database schema.

Translated Entities into Tables:

- Created a table for each entity identified in the ERD.

Identified Attributes:

- For each entity, identify the attributes (fields) represented in the ERD.

Defined Columns:

- Created columns within each table to represent the attributes identified in the ERD.

Primary Keys:

- Identified the primary key attribute(s) for each entity in the ERD.

Translated Relationships:

- Review the relationships between entities in the ERD.
- Translate these relationships into foreign key constraints in the database schema.

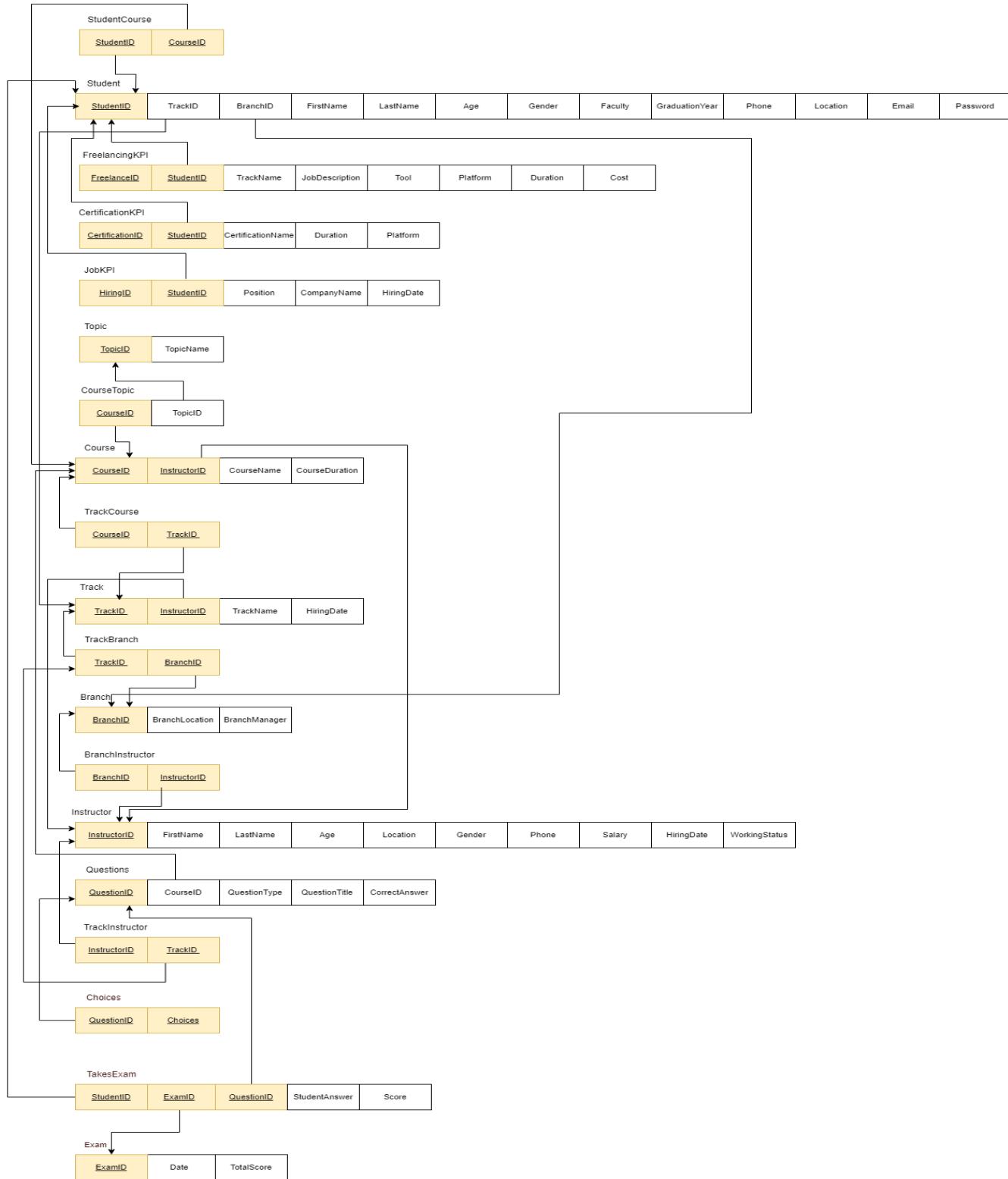
Foreign Keys:

- Add foreign key columns to child tables, referencing the primary key(s) of the related parent table(s).

Normalization:

- Applied normalization techniques to ensure that the database schema is structured efficiently and minimizes redundancy.

Results:



Creating DataBase on SQL SERVER:

Steps:

Wrote CREATE TABLE Statements:

- CREATE TABLE statement for each entity identified in the ERD.
- Specify the table names and list the attributes (columns) along with their data types and constraints.

Defined Columns:

- For each attribute in the entity, specified the column names, data type, and any constraints such as NOT NULL, UNIQUE, or DEFAULT values.

Primary Key Constraints:

- Included the primary key constraint using the PRIMARY KEY keyword after defining the column(s).
- If the primary key is composed of multiple columns, use the PRIMARY KEY constraint followed by the column names enclosed in parentheses.

Foreign Key Constraints:

- Identify relationships between entities that require foreign key constraints.
- Add foreign key columns to child tables referencing the primary key(s) of the related parent table(s).
- Define foreign key constraints using the FOREIGN KEY keyword followed by the column name and the REFERENCES keyword specifying the parent table and column.

Execute SQL Statements:

- Execute the SQL statements.

Results:

Examples:

```

CREATE TABLE Student(
    StudentID INT PRIMARY KEY NOT NULL,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Age INT CHECK (Age BETWEEN 22 AND 35),
    Gender VARCHAR(50),
    Phone int,
    [Location] VARCHAR(50),
    Faculty VARCHAR(50),
    GraduationYear INT,
    Email VARCHAR(50) NOT NULL CHECK (EMAIL LIKE '%_@%.%'),
    [Password] VARCHAR(50) NOT NULL);
ALTER TABLE Student
ADD TrackID INT,
CONSTRAINT Fk_TrackIDStudent FOREIGN KEY (TrackID) REFERENCES Track(TrackID);

```

```

CREATE TABLE Course(
    CourseID INT PRIMARY KEY NOT NULL,
    InstructorID INT,
    CourseName VARCHAR(255),
    CourseDuration INT,

CONSTRAINT FK_Course_Instructor FOREIGN KEY(InstructorID) REFERENCES Instructor(InstructorID)
);

```

```

CREATE TABLE StudentCourse(
    StudentID INT NOT NULL,
    CourseID INT NOT NULL,

CONSTRAINT Fk_StudentID_StudentCourse FOREIGN KEY (StudentID) REFERENCES Student(StudentID),
CONSTRAINT FK_CourseID_StudentCourse FOREIGN KEY (CourseID) REFERENCES Course(CourseID),
CONSTRAINT PK_StudentCourse PRIMARY KEY( StudentID,CourseID)
);

```

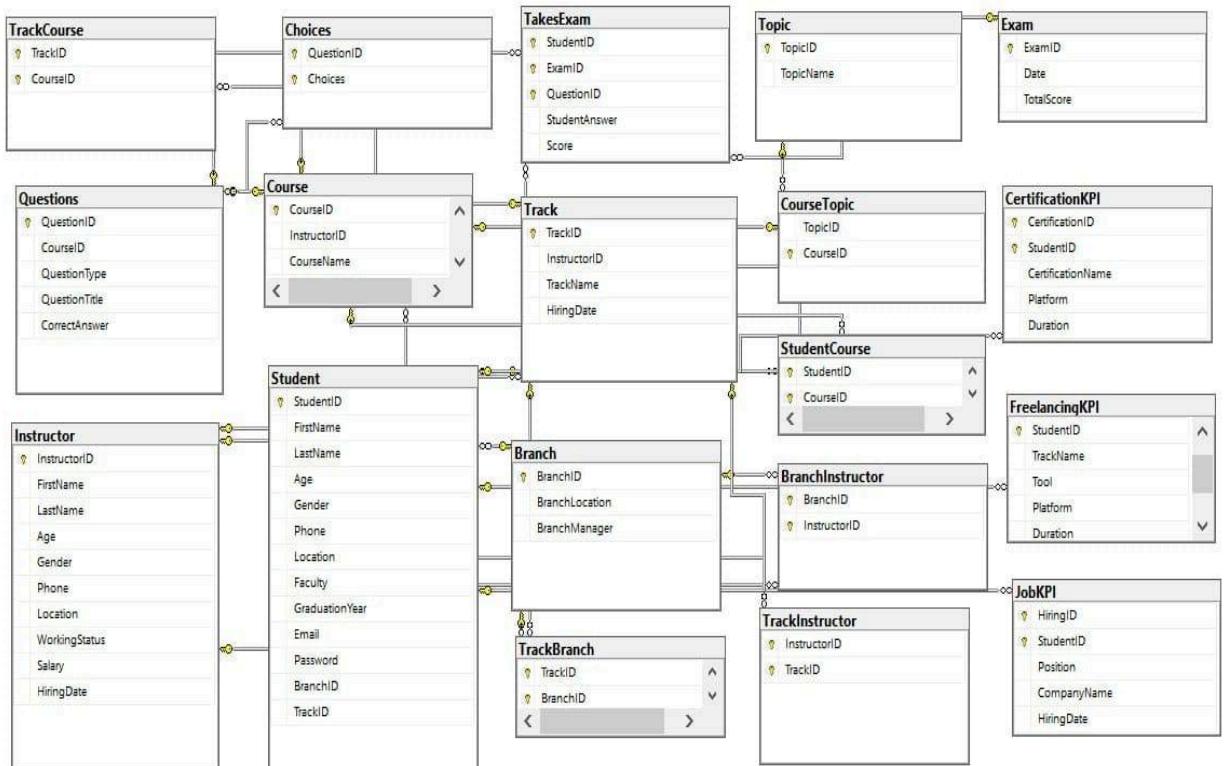
```

CREATE TABLE TakesExam(
    StudentID INT NOT NULL,
    ExamID INT NOT NULL,
    QuestionID INT NOT NULL,
    StudentAnswer VARCHAR(255) NOT NULL ,
    Score INT NOT NULL,

    CONSTRAINT PK_TakesExam PRIMARY KEY (StudentID,QuestionID,ExamID),
    CONSTRAINT FK_StudentID_TakesExam FOREIGN KEY (StudentID) REFERENCES Student(StudentID),
    CONSTRAINT FK_QuestionID_TakesExam FOREIGN KEY (QuestionID) REFERENCES Questions(QuestionID),
    CONSTRAINT FK_ExamID_TakesExam FOREIGN KEY (ExamID) REFERENCES Exam(ExamID),
    CONSTRAINT CHK_Score CHECK (Score IN (0, 1))
);

```

DataBase Diagram:



Stored Procedures:

- Programmability
 - Stored Procedures
 - + System Stored Procedures
 - + dbo.AdminLoginProdcedure
 - + dbo.Answer_and_Correction
 - + dbo.AssignCourseToTrack
 - + dbo.AssignInstructorToTrack
 - + dbo.CourseScore
 - + dbo.CreateExamv02
 - + dbo.CreateExamv03
 - + dbo.DeleteBranch
 - + dbo.DeleteCertificationKPI
 - + dbo.DeleteCourse
 - + dbo.DeleteFreelancingKPI
 - + dbo.DeleteInstructorByName
 - + dbo.DeleteJobKPI
 - + dbo.DeleteQuestion
 - + dbo.DeleteStudent
 - + dbo.DeleteTopic
 - + dbo.GenerateExam
 - + dbo.GetCourseID
 - + dbo.GetGrades
 - + dbo.GetQuestoins
 - + dbo.InserNewtTrack
 - + dbo.InsertBranch
 - + dbo.InsertCertificationKPI
 - + dbo.InsertCourse
 - + dbo.InsertFreelancingKPI
 - + dbo.InsertInstructor
 - + dbo.InsertJobKPI
 - + dbo.InsertQuestion
 - + dbo.InsertStudent
 - + dbo.InsertTopic
 - + dbo.SelectBranch
 - + dbo.SelectBranches
 - + dbo.SelectCertificationKPIByStudent
 - + dbo.SelectFreelancingKPI
 - + dbo.SelectInstructor
 - + dbo.SelectJobKPI
 - + dbo.SelectQuestforCourseld
 - + dbo.SelectStudent
 - + dbo.SelectStudentCourses
 - + dbo.SelectStudents
 - + dbo.StudentLogin
 - + dhn.STUDFNTSCORF

Examples:

```
Create PROCEDURE [dbo].[InsertStudent]
    @firstname VARCHAR(50), @lastname VARCHAR(50), @age int, @gender VARCHAR(6), @phone NCHAR(10),
    @location VARCHAR(255), @faculty VARCHAR(50), @graduationyear INT, @Email VARCHAR(50), @Password
VARCHAR(50),
    @branchname VARCHAR(50),@trackname VARCHAR(50)
AS
BEGIN
    SET NOCOUNT ON;
    DECLARE @trackid INT
    SELECT @trackid = TrackID FROM Track WHERE TrackName = @trackname

    DECLARE @BranchID INT
    SELECT @BranchID = BranchID FROM Branch WHERE BranchLocation = @branchname;

    INSERT INTO Student (FirstName, LastName, Age, Gender, Phone, [Location], Faculty,
GraduationYear, Email, [Password],BranchID, TrackID)
    VALUES (@firstname, @lastname, @age, @gender, @phone, @location, @faculty,
@graduationyear,@email, @Password,@BranchID ,@trackid)

    Declare @StudentID INT
    Set @StudentID = SCOPE_IDENTITY()

    INSERT INTO StudentCourse (StudentID, CourseID)
    SELECT @StudentID, CourseID
    FROM TrackCourse
    WHERE TrackID = @TrackID;
END;
```

```
CREATE PROCEDURE InsertBranch
    @BranchLocation Varchar(50),
    @BranchManager  Varchar(50) = NULL
with encryption
AS
BEGIN
    -- Check location already exists
    IF NOT EXISTS (SELECT BranchLocation FROM Branch WHERE BranchLocation = @BranchLocation)
    BEGIN
        -- Insert into branch table
        INSERT INTO Branch (BranchLocation, BranchManager)
        VALUES (@BranchLocation, @BranchManager);
    END
    ELSE
    BEGIN
        RAISERROR('Branch already exists.', 16, 1);
    END;
```

Data Transformation Process(SSIS)

Defined Data Fields:

- For each entity in database schema, defined the corresponding data field
- Specify the data types, formats, and any other constraints for each field.

Generate Data:

- After defining the data schema and fields, instruct Mockaroo to generate dummy data based specifications.
- Adjusted the number of rows to generate as needed.

Download Data:

- After data generation is complete, downloaded the generated data in Csv.

Examples:

Field Name	Type	Options
BranchID	Number	min: 1 max: 14 decimals: 0 blank: 0 % Σ X
BranchLocation	Custom List	Smart Village, New Capital, Cairo University, Alexandria, Assiut, Aswan, Beni Suef, Fayoum, Ismaili Σ X
BranchManager	Custom List	Ahmed Khalil, Amina Abbas, Youssef Hamdi, Layla Farid, Kareem Salah, Mariam Nasser, Omar Ma Σ X
+ ADD ANOTHER FIELD GENERATE FIELDS USING AI...		
# Rows:	14	Format: Excel ▾

Field Name	Type	Options
TrackID	Number	min: 1 max: 7 decimals: 0 blank: 0 % Σ X
InstructorID	Number	min: 1 max: 210 decimals: 0 blank: 0 % Σ X
TrackName	Custom List	Web Development, Power BI, Full stack, DevOps, Mobile Applications, Cyber Security Associate, In Σ X
HiringDate	Datetime	01/01/2020 <input type="button" value="to"/> 01/01/2024 <input type="button" value="format: dd/mm/yyyy"/> blank: 0 % Σ X
+ ADD ANOTHER FIELD GENERATE FIELDS USING AI...		
# Rows:	7	Format: Excel ▾

Field Name	Type	Options
InstructorID	Number	min: 10 max: 210 decimals: 0 blank: 0 % Σ X
FirstName	Custom List	Ahmed, Omar, Youssef, Mahmoud, Khalid, Ali, Hadi, Tariq, Ziad, Samir, Faisal, Saif, Rami, Karim, Ni... random blank: 0 % Σ X
LastName	Custom List	Abdel Aziz, Abdel Hakim, Abdel Nasser, Abdel Rahman, Abou El Fadl, Ali, Amin, Anwar, Ashraf, Aw... random blank: 0 % Σ X
Age	Number	min: 23 max: 50 decimals: 0 blank: 0 % Σ X
Location	Custom List	Alexandria, Aswan, Asyut, Beheira, Beni Suef, Cairo, Dakahlia, Damietta, Falyum, Gharbia, ... weighted blank: 0 % Σ X
Gender	Custom List	Female, Male random blank: 0 % Σ X
Phone	Phone	format: ##### blank: 0 % Σ X
Salary	Number	min: 10000 max: 30000 decimals: 0 blank: 0 % Σ X
HiringDate	Datetime	01/01/2010 to 01/01/2024 format: dd/mm/yyyy blank: 0 % Σ X
WorkingStatus	Custom List	Onsite, Remote, Hybrid weighted blank: 0 % Σ X

+ ADD ANOTHER FIELD GENERATE FIELDS USING AI...

ETL Using SSIS

Created a New SSIS Project:

Added Data Flow Task:

- New Data Flow Task to represent the data transformation process.

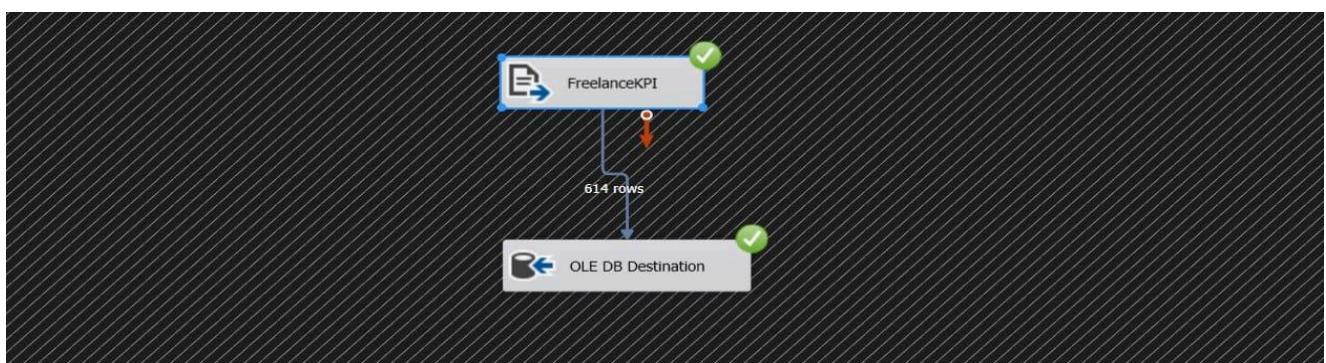
Configure Data Flow Components:

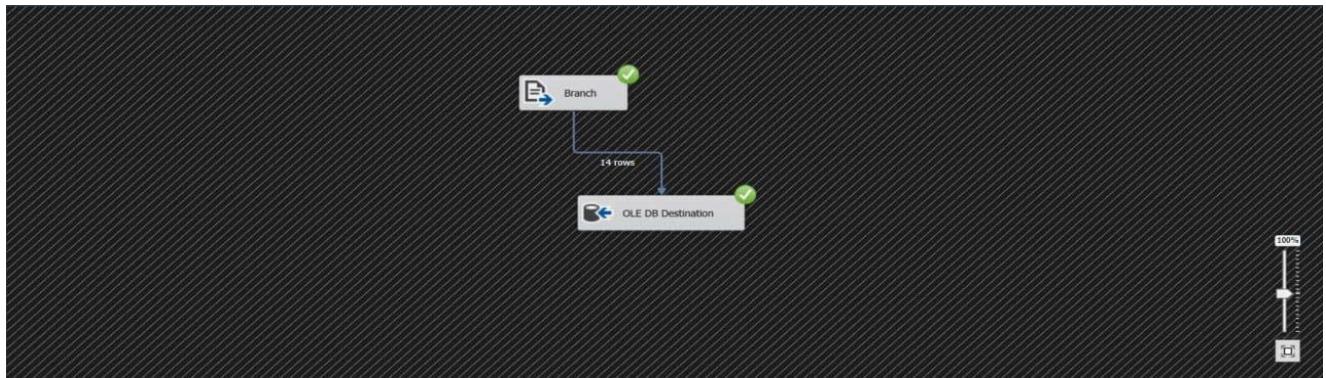
- Within the Data Flow Task, configure various components such as Source, Transformation, and Destination.
Used data source as CSV .
- Specify the destination where transformed data will be loaded (SQL Server database)

Execute SSIS Package:

- Executed the SSIS package to perform the data transformation process.

Examples:





Reports Using SSRS:

Instructor ID

1 100% Find | Next

Instructor's Courses and Student Count

coursename	Count Students
Critical Thinking and Problem-Solving	983
Presentation Skills	983

student ID

1 100% Find | Next

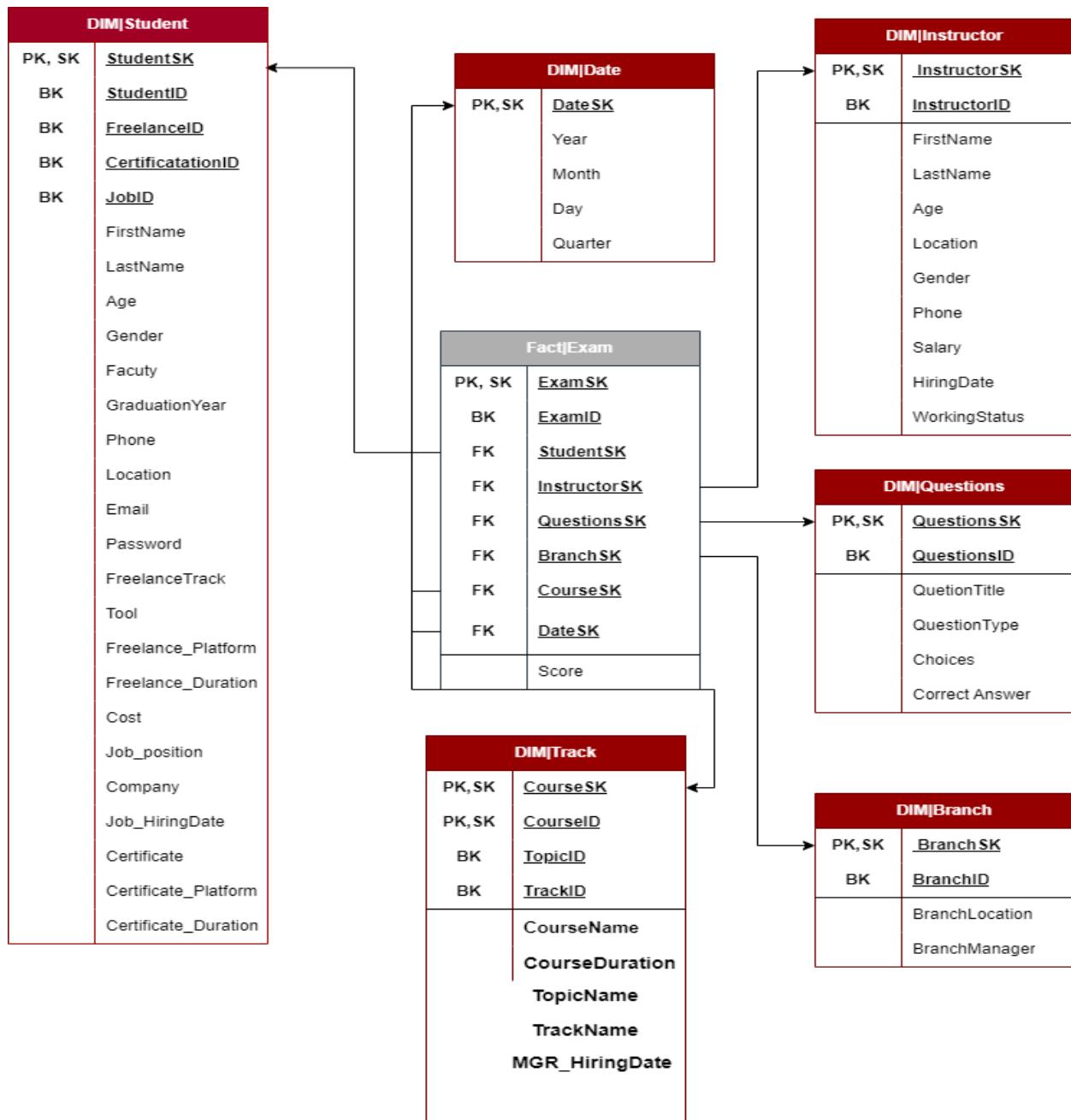
The Grades of the Student in Each Course.

Student ID	Course Name	TOTALSCORE
360	Database Design and Management	70 %
360	Front-End Frameworks	50 %

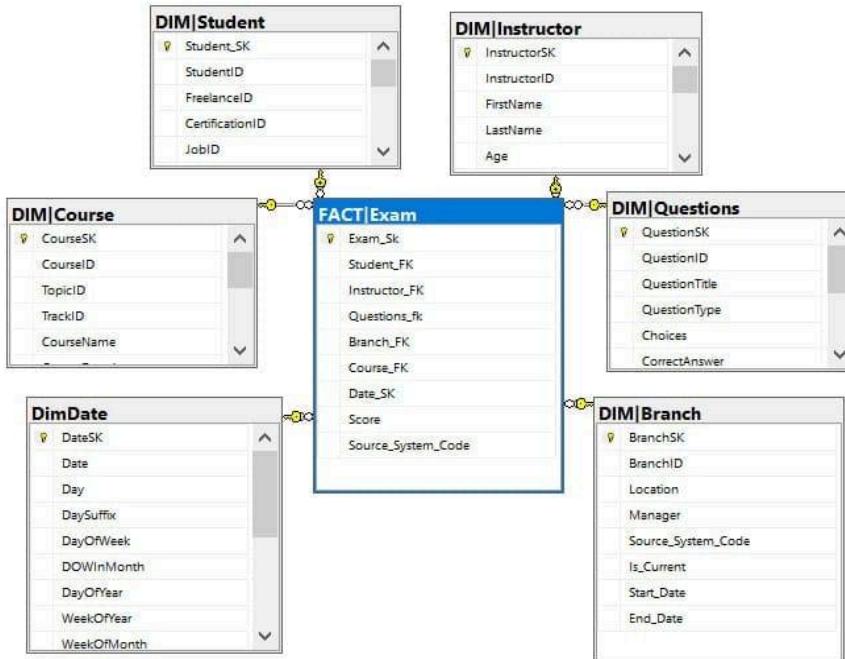
Data Warehouse:

Data Modeling

- Designed the dimensional model for the Data Warehouse, including fact tables, dimension tables, and their relationships.
- Identified key dimensions and measures to support reporting and analysis.



Create DWH Schema on SQL Server

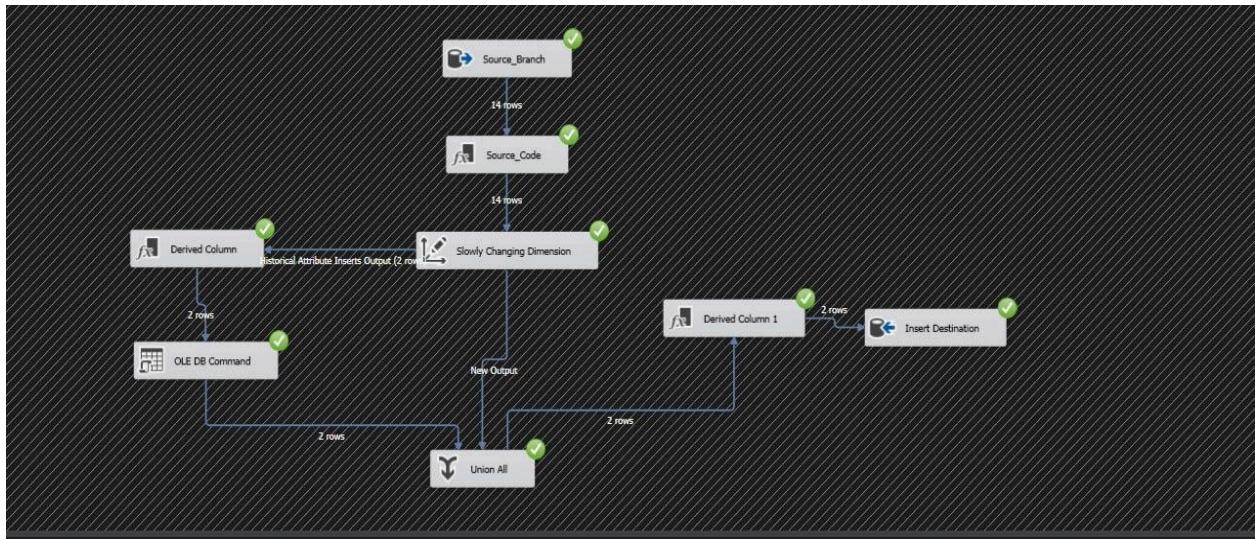


ETL Using SSIS:

- Identify the data source as OLTP DataBase from SQL SERVER to extract data for the Data Warehouse.
- Use SSIS Data Flow Tasks to extract data from OLTP database.
- Use Data Flow Tasks to transform the extracted data according to the dimensional model.
- Apply various transformations such as data cleansing, data type conversion, aggregation, and surrogate key generation.

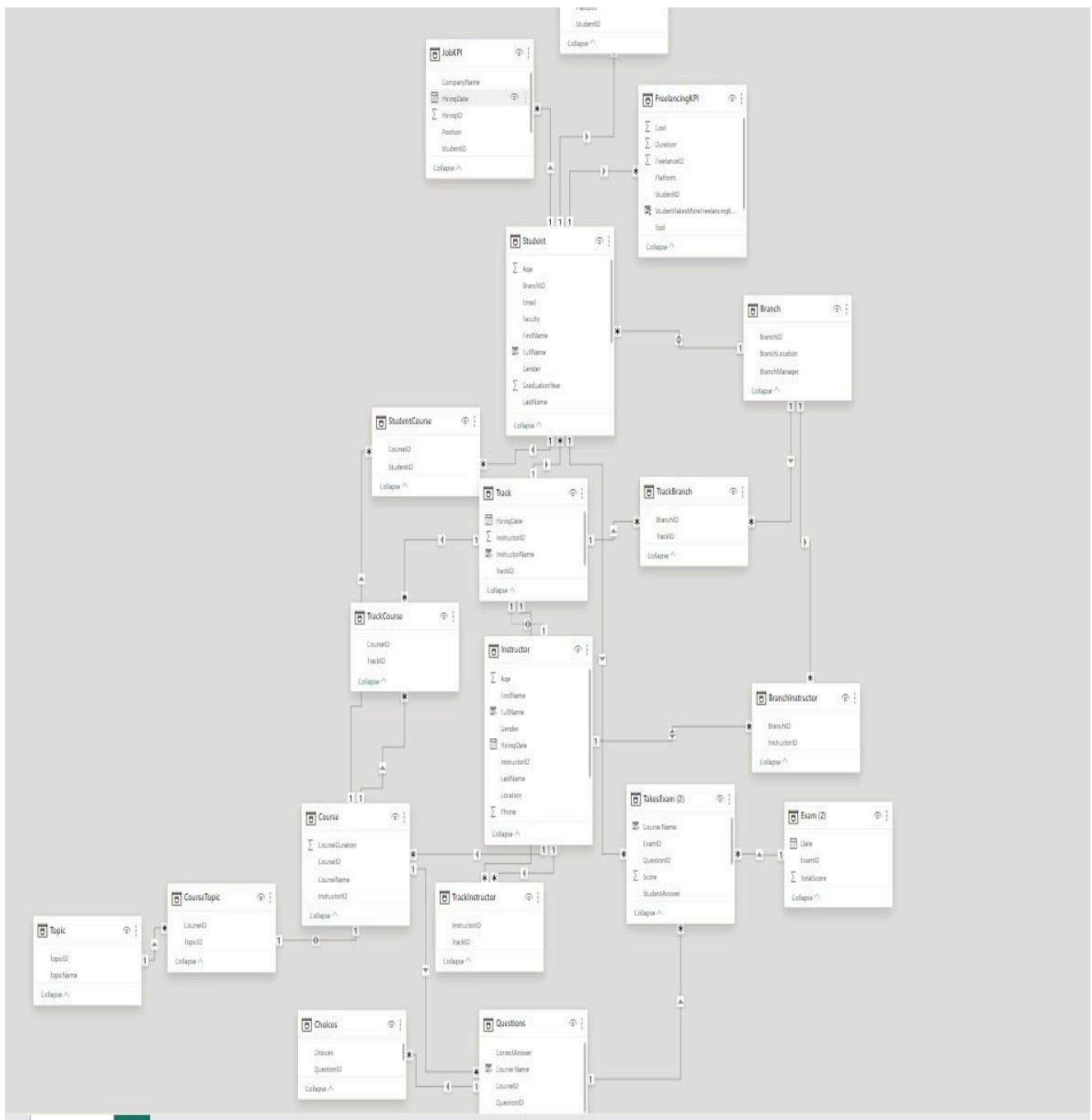
Examples:





PowerBI Dashboards:

StarSchema On PowerBI



Dashboard Examples:

The dashboard displays various metrics and visualizations related to instructors:

- Instructor Report**: The main title.
- Filter Panel (Left):**
 - Gender:** All
 - FullName:** All
 - WorkingStatus:** All
- Key Metrics (Top Left):**
 - 25 Instructor
 - 37 Average of Age
 - 16 Location
 - 21.9K Average of Salary
- Instructors by WorkingStatus (Bar Chart):**

WorkingStatus	Count
Onsite	10
Hybrid	8
Remote	4
- Instructors by Gender (Card):**

Gender	Count
Females	13
Males	12
Supervisors	7
- Supervisor and Track List (Table):**

SupervisorName	TrackName
Sirine Sheriff	DevOps
Fatima Youssef	Cyber Security
Nadia Wael	Full stack
Mariam Gamal	Interior Design
Tarik Bader	Mobile Applications
Ibrahim Mohsen	Power BI
Samira Saleh	Web Development
- Average of Salary by Gender (Donut Chart):**

Gender	Average Salary	Percentage
Female	20.67K	(47.39%)
Male	22.95K	(52.61%)
- Instructors by Location (Map):** A map of Egypt showing the distribution of instructors across various locations. Legend: Alexandria (Blue), Asyut (Orange), Beheira (Yellow), Cairo (Purple), Qalyoub (Green), Gharbia (Red), Imaida (Teal). Major cities like Cairo, Alexandria, and Giza are labeled.

Overview

Students 983

Topics 9

Instructors 9

Certificates 258

Faculties 11

Courses 12

Tracks 7

Freelances 614

Graduation Year All

Faculty All

Students Location All

Instructors working status Hybrid

Branches All

Students by Gender

Gender	Count	Percentage
Male	482	49.03%
Female	501	50.97%

Instructors By Gender

Gender	Count	Percentage
Female	2	22.22%
Male	7	77.78%

Instructors and Students by Branches

Branch	Instructors	Students
Fayoum	67	7
Minya	48	7
New Capital	49	7
Ismailia	136	6
Mansoura	90	6
Sohag	69	6
Beni Suef	50	5
Menofia	124	5
Giza	34	5
Alexandria	30	4
Assuit	4	85
Cairo	114	4
Smart Village	4	87

Students by Track

Track	Count	Percentage
Full stack	300	30.52%
Web Devle...	150	15.26%
Interior D...	128	13.02%
Mobile ...	72	7.32%
Power BI	72	7.32%
DevOps	50	5.09%
Cyber Sec...	183	18.62%

