IST 659 REPORT

The "E-Learning Platform" project was developed by **Aditya Marathe, Gautam Balgi,** and **Rohan Gopalan.** It offers a comprehensive overview of their e-learning application called E-Learn. Here is a detailed report on the contents:

Introduction

Accessibility

E-learning eliminates geographical barriers, making education accessible to anyone with internet access. In contrast to conventional learning settings that necessitate physical attendance, e-learning allows students from isolated areas, underrepresented regions, or those with mobility issues to obtain high-quality educational materials. This transformation in education empowers learners to engage with top-tier instructors and institutions, regardless of their geographic location. Additionally, online platforms frequently incorporate multilingual resources and accessibility features, such as subtitles, text-to-speech options, or customizable formats, thereby promoting inclusivity for various audiences, including those with disabilities.

Flexibility

One of the most significant benefits of e-learning is its ability to conform to individual schedules and preferences. Students can move at their own speed, revisiting challenging subjects or speeding through content they already understand as necessary. This tailored approach enables individuals to create a learning experience that fits their objectives and lifestyle, making it especially advantageous for working professionals, parents, and non-traditional learners. Additionally, e-learning platforms provide a wide range of courses, allowing students to choose topics that correspond with their interests and career goals. The capacity to seamlessly incorporate education into everyday life promotes continuous learning and skill development without interfering with other responsibilities.

Key Components

Users

The platform accommodates various user roles, each having unique functions and responsibilities. Students engage with the platform to access their courses, submit assignments, and monitor their learning progress. Instructors are essential for selecting and uploading course materials, designing assignments, and assessing student performance. Administrators oversee the platform's overall operation, ensuring everything runs smoothly, managing user registrations, and maintaining the database. By addressing these roles, the system promotes collaboration and efficiency within the e-learning environment.

Courses

Courses are central to the platform, serving as organized collections of learning materials that provide educational content. Each course is designed to offer a comprehensive learning experience, often including videos, reading materials, and interactive elements such as discussion forums or live sessions. The platform allows for easy navigation and categorization of courses based on topics, difficulty levels, or certifications, ensuring that learners can easily find and enroll in subjects that align with their academic or professional goals.

Scores

Monitoring and assessing student performance is essential for encouraging improvement, with scores playing a key role in this process. The platform keeps a systematic record of grades from assignments, quizzes, and exams, providing detailed feedback to students. These performance metrics are accessible to students for self-assessment, as well as to instructors and administrators for tracking progress and pinpointing areas where students may require additional support.

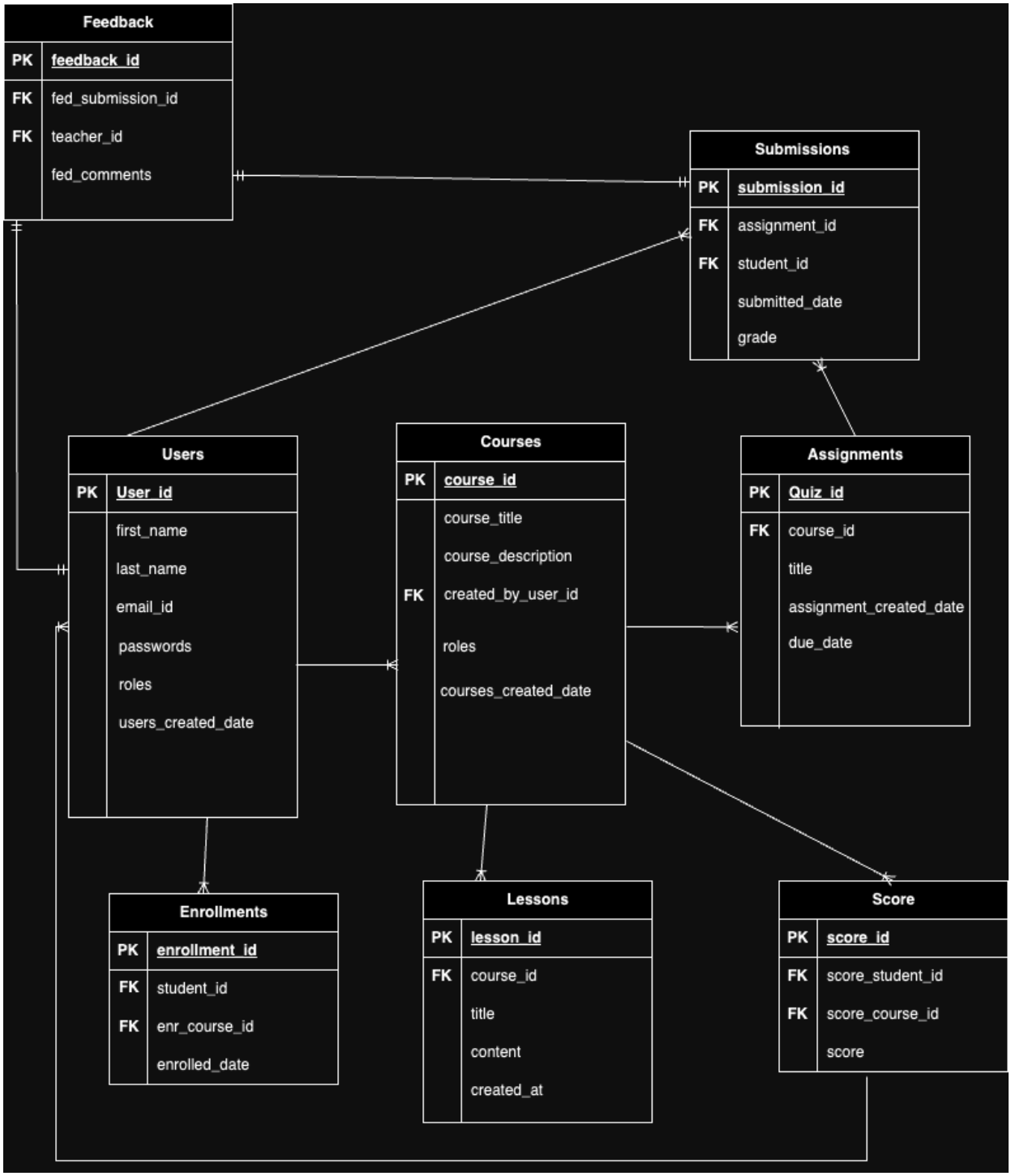
Assignments

Assignments are a fundamental aspect of evaluating students, including quizzes, examinations, and project work. These activities aim to strengthen learning, evaluate understanding, and promote critical thinking. The platform accommodates various formats, ranging from multiple-choice questions to essays and code submissions, giving instructors flexibility in their assessment approaches. Automated grading for objective questions and feedback mechanisms for subjective evaluations simplify the assessment process, ensuring it is both efficient and clear.

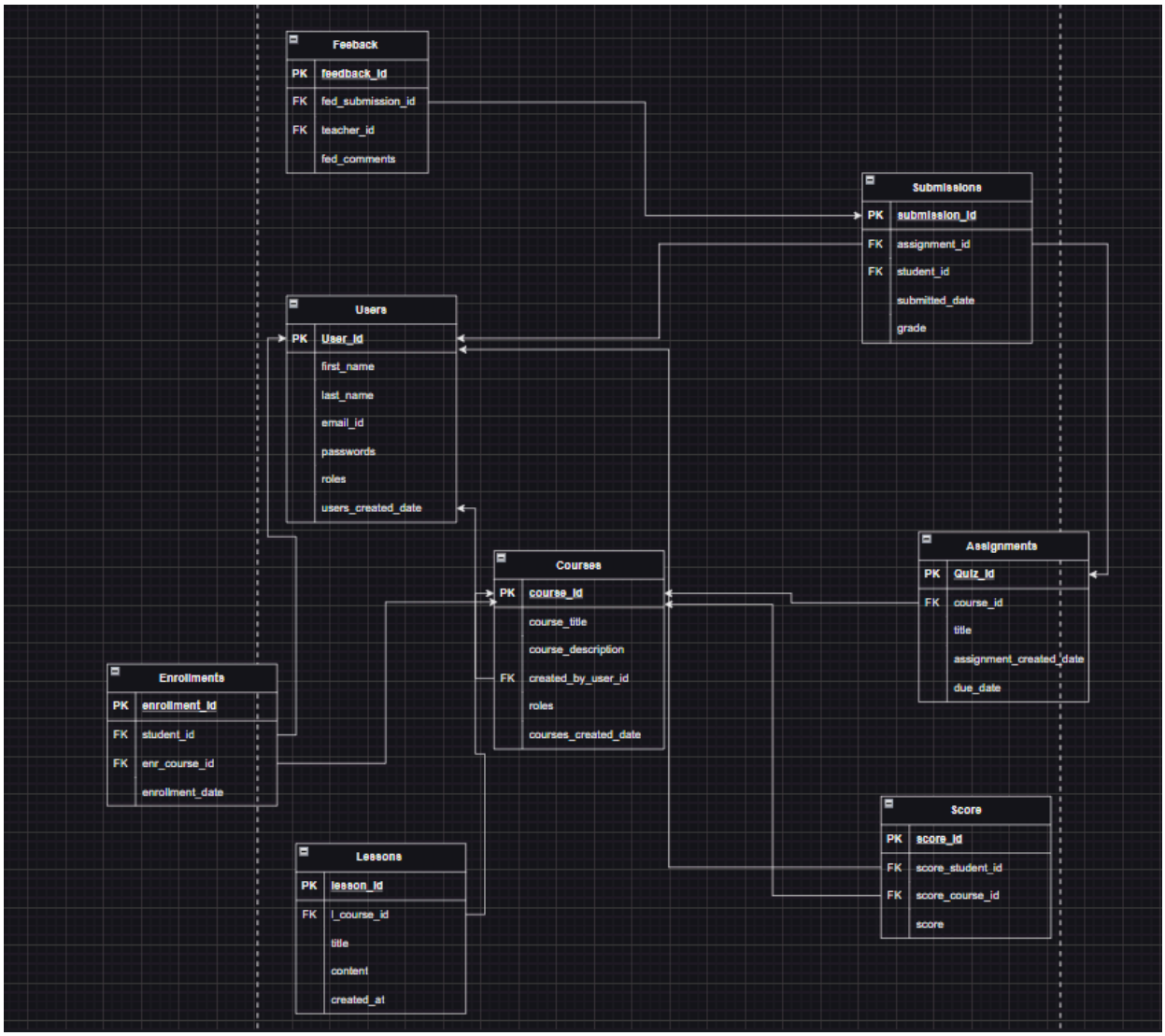
Data Analysis: Entity, Attribute, and Relationship Documentation

Entity	Attributes	Related Entities	Relationship Type
Users	user_id, first_name, last_name, email, password, role, created_at	Courses, Enrollments, Feedback	One-to-Many
Courses	course_id, title, description, created_by, created_at	Users, Enrollments, Lessons, Assignments	One-to-Many
Enrollments	enrollment_id, student_id, course_id, enrolled_at	Users, Courses	Many-to-One
lessons	lesson_id, course_id, title, content, created_at	Courses	Many-to-One
Assignments	assignment_id, course_id, title, description, due_date	Courses, Submissions	One-to-Many
Submissions	submission_id, assignment_id, student_id, submitted_at, grade	Assignments, Feedback	One-to-Many
Feedback	feedback_id, submission_id, teacher_id, comments	Submissions, Users	One-to-One
score	score_id, student_id, course_id, score	Users, Courses	Many-to-One

Conceptual Data Model Diagram:



Logical Model Diagram:



Detailed diagram for each screen used in the application.

We used three approaches to build our application

- Client-based application using python

- Power-Apps(Dashboards)/Power-BI(Dashboards)

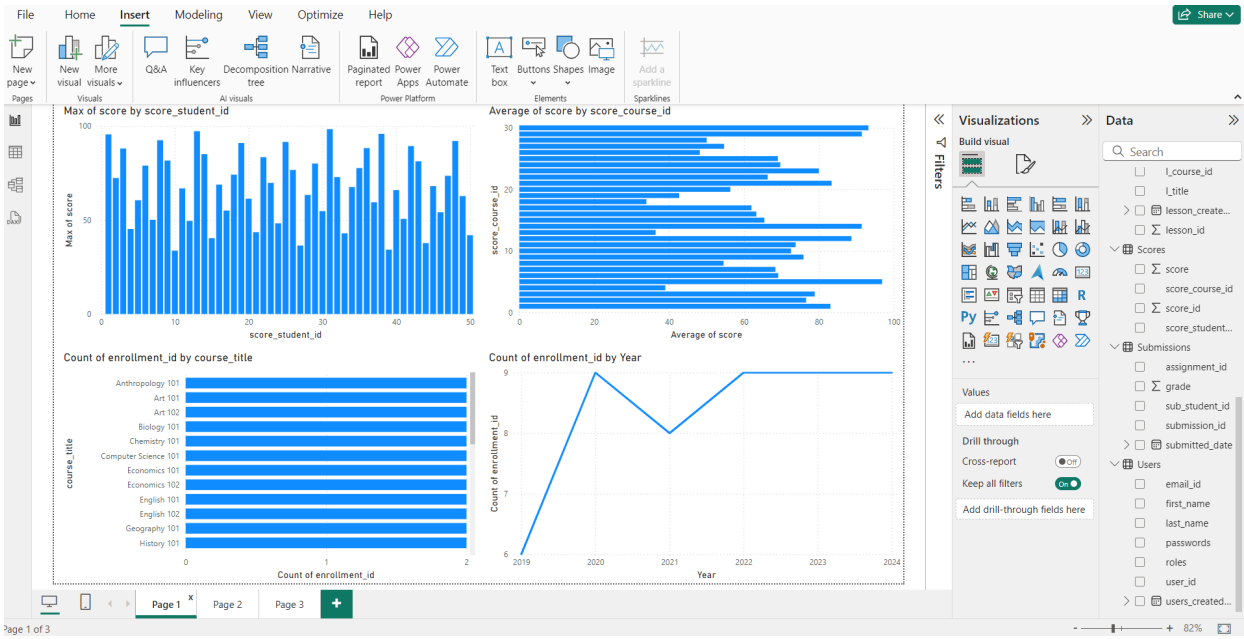
Python Application diagram

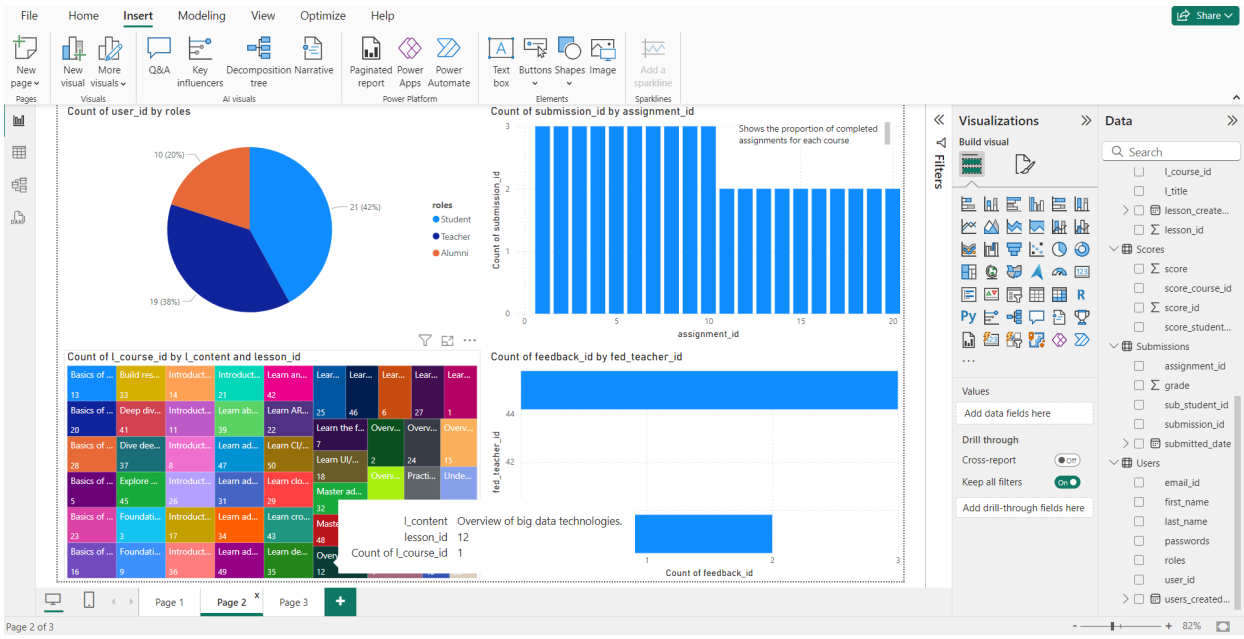
```
> rohangopalan3 > final1.py > connect_to_database
    try:
        server = 'rg316.database.windows.net'
        database = 'finalproject1'
        username = 'sqladmin'
        password = '12345678!A'
        connection_string = (
            f'DRIVER={{ODBC Driver 17 for SQL Server}};SERVER={server};DATABASE={database};UID={username};'
            f'PWD={password};Encrypt=yes;TrustServerCertificate=no;Connection Timeout=30;'
        )
        conn = pyodbc.connect(connection_string)
        return conn
    except Exception as e:
        return str(e)

# Function to display results
def show_results(stdscr, query, description):
    conn = connect_to_database()
    if isinstance(conn, str):
        stdscr.addstr(0, 0, f"Connection error: {conn}", curses.color_pair(2))
        stdscr.refresh()
        stdscr.getch()
```

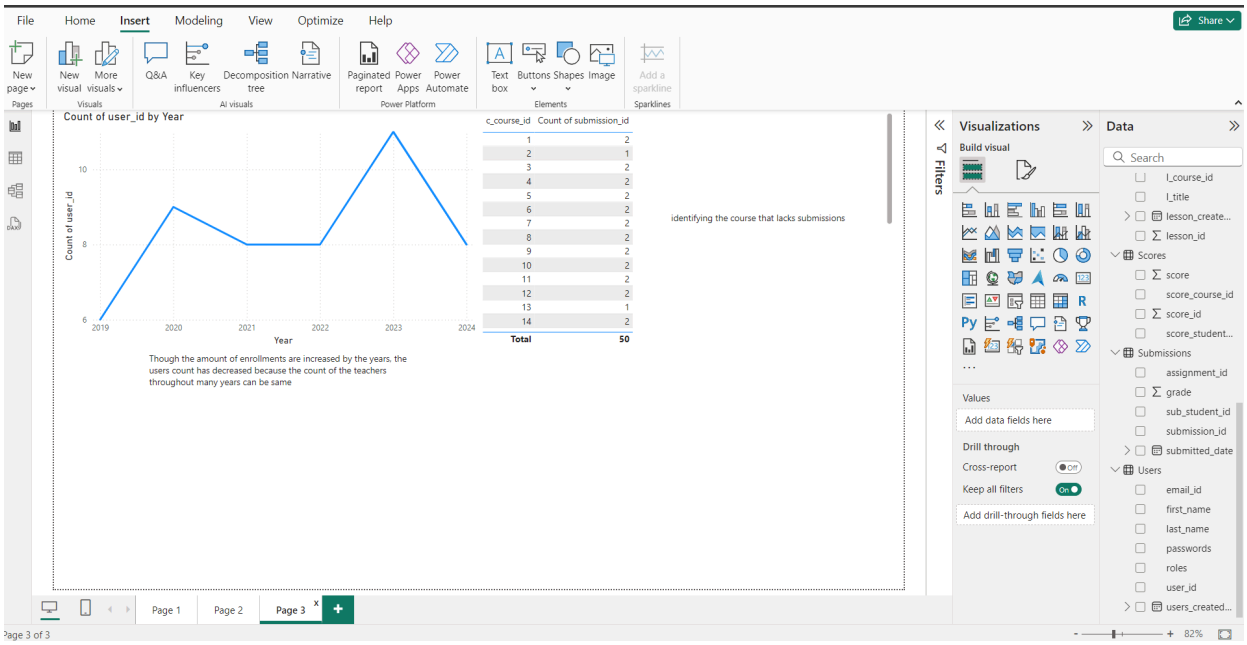
```
EMS 32   OUTPUT   DEBUG CONSOLE   TERMINAL   SEARCH ERROR
L: COMMIT GRAPH DETAILS        TERMINAL
Commit   Working Changes
ch details for commits and
ashes are shown as you
vigate:
lines in the text editor
commits in the Commit
Graph, Visual File History,
or Commits view
stashes in the Stashes

Welcome to the Azure SQL Terminal UI for Course Management
Please choose an option from the menu below:

1. View all courses that are currently open for enrollment.
2. Enroll in a course by selecting from the available courses list.
3. View your progress in courses, including scores and completion status.
4. Search for courses using a specific keyword to find relevant options.
Q. Quit the application and exit the terminal UI.
```

POWER BI:

These are the visualizations (Power BI) which we calculated for the data:


File
Home
Insert
Modeling
View
Optimize
Help
Share
Max of score by score_student_id
Max of score
score_student_id
Average of score by score_course_id
score_course_id
Average of score
Count of enrollment_id by course_title
Anthropology 101
Art 101
Art 102
Biology 101
Chemistry 101
Computer Science 101
Economics 101
Economics 102
English 101
English 102
Geography 101
History 101
course_title
Count of enrollment_id
Count of enrollment_id by Year
Count of enrollment_id
Year
Visualizations
Build visual
Values
Add data fields here
Drill through
Cross-report
Keep all filters
Add drill-through fields here
Data
Search
l_course_id
l_title
lesson_create...
lesson_id
Scores
score
score_course_id
score_id
score_student...
Submissions
assignment_id
grade
sub_student_id
submission_id
submitted_date
Users
email_id
first_name
last_name
passwords
roles
user_id
users_created...
Page 1
Page 2
Page 3
Page 1 of 3
82%


File
Home
Insert
Modeling
View
Optimize
Help
Share
Count of user_id by roles
10 (20%)
21 (42%)
19 (38%)
roles
Student
Teacher
Alumni
Count of submission_id by assignment_id
Shows the proportion of completed assignments for each course
Count of submission_id
assignment_id
Count of l_course_id by l_content and lesson_id
l_content Overview of big data technologies.
lesson_id 12
Count of l_course_id 1
Count of feedback_id by fed_teacher_id
fed_teacher_id
Count of feedback_id
Visualizations
Build visual
Values
Add data fields here
Drill through
Cross-report
Keep all filters
Add drill-through fields here
Data
Search
l_course_id
l_title
lesson_create...
lesson_id
Scores
score
score_course_id
score_id
score_student...
Submissions
assignment_id
grade
sub_student_id
submission_id
submitted_date
Users
email_id
first_name
last_name
passwords
roles
user_id
users_created...
Page 1
Page 2
Page 3
Page 2 of 3
82%



POWER APPS:

Following are the power apps screenshots:

Welcome to Our e-learning Platform

e learn

Start

COURSES
USERS
SCORES
LESSONS
CALENDAR

List of Courses

Search items

Introduction to Biology
Biology 101

Basic English
English 101

Introduction to Programming
Computer Science 101

World History
History 101

Art 101

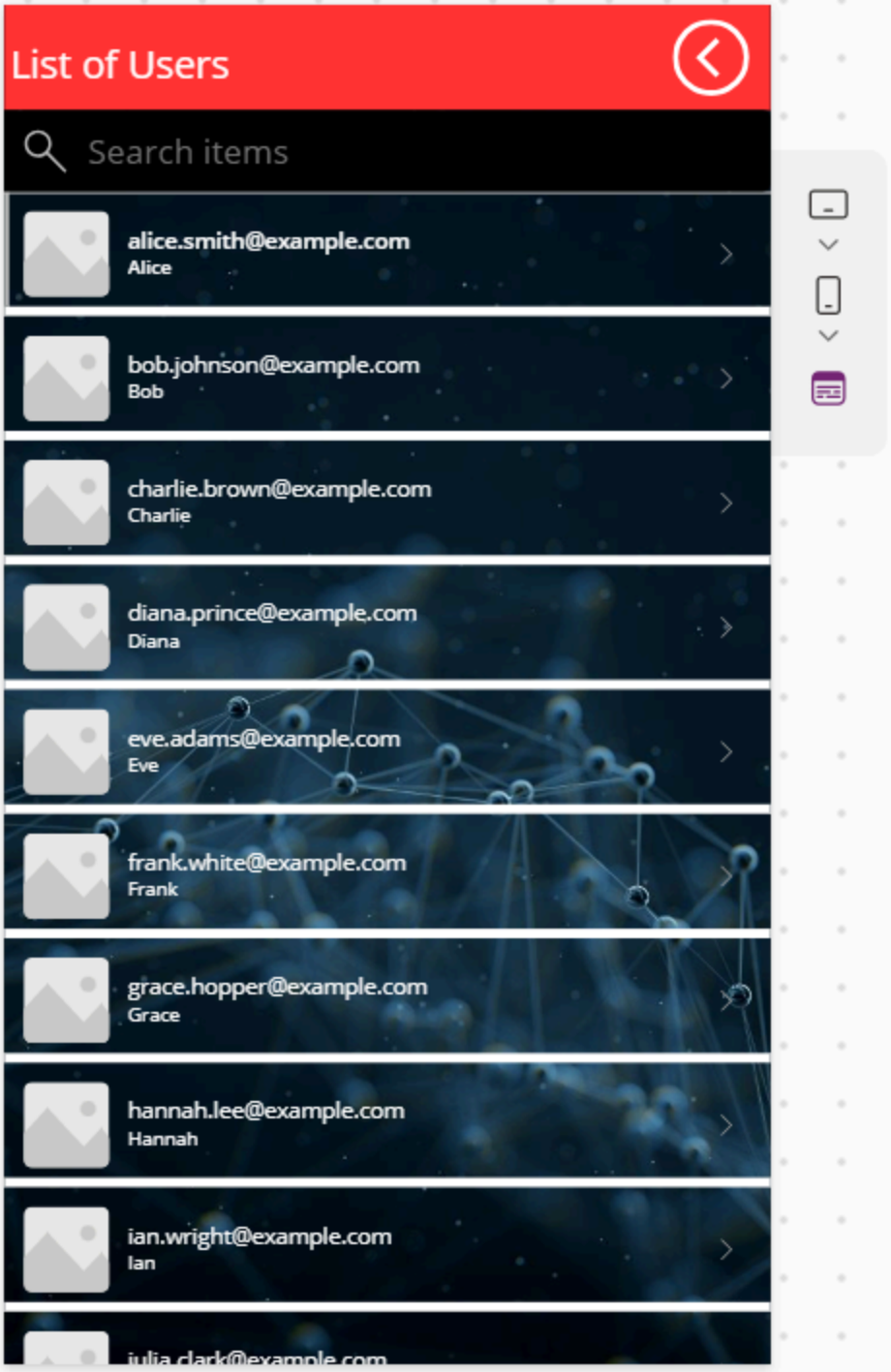
Introduction to Philosophy
Philosophy 101

Introduction to Economics
Economics 101

Fundamentals of Music
Music 101

Basics of Statistics
Statistics 101

Introduction to Environmental Science


List of Users
Search items
alice.smith@example.com
Alice
bob.johnson@example.com
Bob
charlie.brown@example.com
Charlie
diana.prince@example.com
Diana
eve.adams@example.com
Eve
frank.white@example.com
Frank
grace.hopper@example.com
Grace
hannah.lee@example.com
Hannah
ian.wright@example.com
Ian

SCORES

score_id	score	score_co...	score_stu...
1	95.5	5	1
2	72.3	10	2
3	88	15	3
4	45.2	20	4
5	60.5	25	5
6	78.9	3	6
7	50.1	6	7
8	92.4	9	8
9	81.6	12	9
10	33.7	18	10
11	66.8	22	11
12	49.5	28	12
13	97.2	30	13
14	85	1	14
15	40.3	4	15
16	68.8	7	16
17	55	8	17
18	74.1	11	18
19	90.9	14	19
20	61.3	17	20
21	43.5	19	21
22	83.4	21	22
23	69.7	24	23
24	48.2	26	24
25	91.5	29	25
26	76.6	2	26
27	36.4	13	27

List of Lessons

Search items

Web Development Basics

Understand data structures.
Data Structures Overview

Basics of machine learning.
Machine Learning Intro

Learn networking basics.
Networking Foundations

Learn the functions of OS.
Operating Systems Basics

Introduction to cybersecurity.
Cybersecurity Essentials

Foundations of software engineering.
Software Engineering Concepts

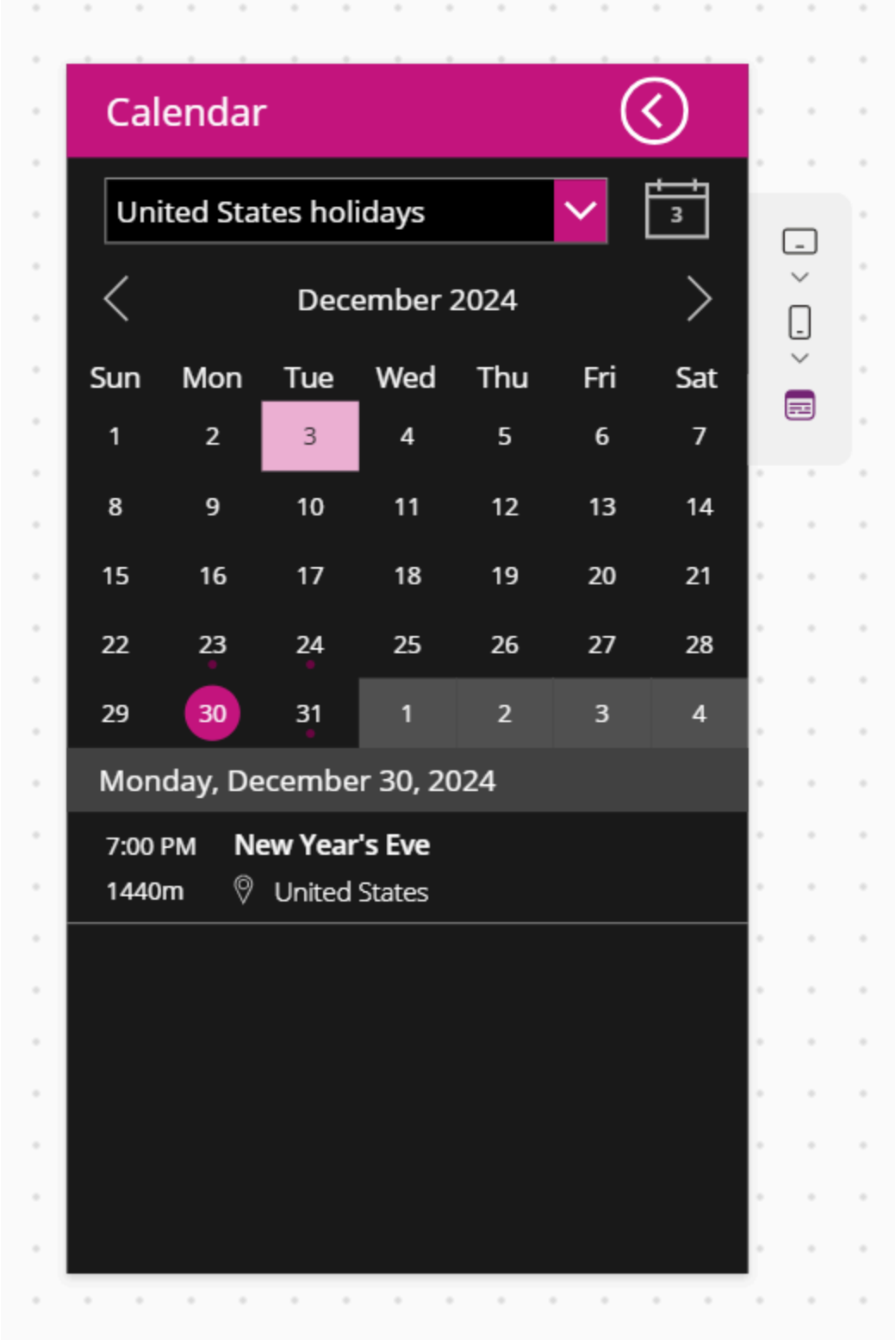
Understand AI concepts.
Artificial Intelligence Basics

Introduction to cloud services.
Cloud Computing Overview

Overview of big data technologies.
Big Data Introduction

Basics of app development.
Mobile App Development

Introduction to blockchain.


Calendar
United States holidays
3
December 2024
Sun Mon Tue Wed Thu Fri Sat
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31 1 2 3 4
Monday, December 30, 2024
7:00 PM
New Year's Eve
1440m
United States

SQL Up/Down Script to Implement the Internal Model with Initial Data

Up Script (Create Tables and Add Initial Data):

```sql
CREATE TABLE Users (
    user_id int PRIMARY KEY ,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    email_id VARCHAR(100) unique NOT NULL,
    passwords VARCHAR(120) NOT NULL,
    roles VARCHAR(20) NOT NULL,
    users_created_date DATETIME
);

CREATE TABLE Courses (
    c_course_id int PRIMARY KEY,
    course_title VARCHAR(100) NOT NULL,
    course_description TEXT NOT NULL,
    created_by int NOT NULL,
    courses_created_date DATETIME
);

CREATE TABLE Enrollments (
    enrollment_id int PRIMARY KEY,
    student_id int NOT NULL,
    enr_course_id int NOT NULL,
    enrolled_date DATETIME DEFAULT GETDATE(),
    CONSTRAINT FK_Student FOREIGN KEY (student_id) REFERENCES Users(user_id),
    CONSTRAINT FK_Course FOREIGN KEY (enr_course_id) REFERENCES Courses(c_course_id)
);

CREATE TABLE Assignments (
    quiz_id int PRIMARY KEY,
    asg_course_id int NOT NULL,
    title VARCHAR(100) NOT NULL,
    assignment_created_date DATETIME,
    due_date DATETIME,
    FOREIGN KEY (asg_course_id) REFERENCES Courses(c_course_id)
);

CREATE TABLE Submissions (
    submission_id int PRIMARY KEY,
    assignment_id int NOT NULL,
    sub_student_id int NOT NULL,
    submitted_date DATETIME,
    grade FLOAT,
    FOREIGN KEY (assignment_id) REFERENCES Assignments(quiz_id),
    FOREIGN KEY (sub_student_id) REFERENCES Users(user_id)
```

```sql
CREATE TABLE Feedback (
    feedback_id int PRIMARY KEY,
    fed_submission_id int NOT NULL,
    fed_teacher_id int NOT NULL,
    fed_comments VARCHAR(100),
    FOREIGN KEY (fed_submission_id) REFERENCES Submissions(submission_id),
    FOREIGN KEY (fed_teacher_id) REFERENCES Users(user_id)
);

CREATE TABLE Scores (
    score_id int PRIMARY KEY,
    score_student_id int NOT NULL,
    score_course_id int NOT NULL,
    score FLOAT NOT NULL,
    FOREIGN KEY (score_student_id) REFERENCES Users(user_id),
    FOREIGN KEY (score_course_id) REFERENCES Courses(c_course_id)
);

create table Lessons (
    lesson_id int PRIMARY KEY,
    l_course_id int not null,
    l_title VARCHAR(250) not null,
    l_content Varchar(300) not null,
    lesson_created_date DATETIME,
    FOREIGN KEY(l_course_id) REFERENCES Courses(c_course_id)
);

------ Inserting the values inside the tables
---- USERS table

INSERT INTO Users (user_id, first_name, last_name, email_id, passwords, roles, users_created_date)
 VALUES
(1, 'Alice', 'Smith', 'alice.smith@example.com', 'P@ssw0rd1!', 'Student', '2019-01-15'),
(2, 'Bob', 'Johnson', 'bob.johnson@example.com', 'B0bJ@123', 'Teacher', '2019-05-10'),
(3, 'Charlie', 'Brown', 'charlie.brown@example.com', 'Ch@r1ie!$', 'Student', '2020-02-18'),
(4, 'Diana', 'Prince', 'diana.prince@example.com', 'D!@naP#$', 'Teacher', '2020-06-25'),
(5, 'Eve', 'Adams', 'eve.adams@example.com', 'EveP@$$!', 'Student', '2020-10-12'),
(6, 'Frank', 'White', 'frank.white@example.com', 'Fr@nkWh1te', 'Student', '2021-03-15'),
(7, 'Grace', 'Hopper', 'grace.hopper@example.com', 'Gr@ceHopp!', 'Teacher', '2021-07-20'),
(8, 'Hannah', 'Lee', 'hannah.lee@example.com', 'H@nnahLee$', 'Student', '2021-12-05'),
(9, 'Ian', 'Wright', 'ian.wright@example.com', 'IanWr!ghT2', 'Student', '2022-04-10'),
(10, 'Julia', 'Clark', 'julia.clark@example.com', 'Ju!i@C!4rk', 'Teacher', '2022-08-22'),
(11, 'Kevin', 'Baker', 'kevin.baker@example.com', 'K3v!nB@ke$', 'Student', '2023-01-14'),
(12, 'Lily', 'Green', 'lily.green@example.com', 'L!lYGr3eN!', 'Student', '2023-03-25'),
```

```sql
INSERT INTO Courses (c_course_id, course_title, course_description, created_by, courses_created_date)
VALUES
(1, 'Math 101', 'Basic Mathematics', 2, '2019-02-15 09:00:00'),
(2, 'Physics 101', 'Introduction to Physics', 4, '2019-07-20 10:30:00'),
(3, 'Chemistry 101', 'Introduction to Chemistry', 2, '2020-03-10 11:00:00'),
(4, 'Biology 101', 'Introduction to Biology', 4, '2020-08-18 14:15:00'),
(5, 'English 101', 'Basic English', 7, '2021-01-25 08:45:00'),
(6, 'Computer Science 101', 'Introduction to Programming', 7, '2021-05-15 13:30:00'),
(7, 'History 101', 'World History', 2, '2021-09-12 15:00:00'),
(8, 'Art 101', 'Introduction to Art', 4, '2022-02-28 10:00:00'),
(9, 'Philosophy 101', 'Introduction to Philosophy', 7, '2022-06-19 11:45:00'),
(10, 'Economics 101', 'Introduction to Economics', 4, '2022-11-05 14:30:00'),
(11, 'Music 101', 'Fundamentals of Music', 7, '2023-01-10 09:20:00'),
(12, 'Statistics 101', 'Basics of Statistics', 2, '2023-03-15 10:10:00'),
(13, 'Environmental Science 101', 'Introduction to Environmental Science', 4, '2023-06-25 11:50:00'),
(14, 'Psychology 101', 'Basics of Human Psychology', 7, '2023-09-18 14:00:00'),
(15, 'Sociology 101', 'Introduction to Sociology', 2, '2023-12-08 08:30:00'),
(16, 'Political Science 101', 'Foundations of Political Science', 4, '2024-02-20 09:45:00'),
(17, 'Anthropology 101', 'Introduction to Anthropology', 7, '2024-05-10 11:15:00'),
(18, 'Literature 101', 'Basics of Literary Studies', 7, '2024-07-22 12:30:00'),
(19, 'Astronomy 101', 'Introduction to Astronomy', 2, '2024-09-15 15:45:00'),
(20, 'Geography 101', 'World Geography Basics', 4, '2024-11-30 10:15:00'),
(21, 'Physics 102', 'Advanced Physics Concepts', 4, '2019-05-10 13:20:00'),
(22, 'Math 102', 'Calculus and Linear Algebra', 2, '2020-04-15 08:10:00'),
(23, 'Chemistry 102', 'Organic Chemistry', 2, '2020-11-20 09:40:00'),
(24, 'Biology 102', 'Genetics and Evolution', 4, '2021-06-05 14:50:00'),
(25, 'English 102', 'Advanced Composition and Rhetoric', 7, '2022-03-12 10:25:00'),
(26, 'Computer Science 102', 'Data Structures and Algorithms', 7, '2023-04-25 11:45:00'),
(27, 'History 102', 'Medieval History', 2, '2023-07-15 14:10:00'),
(28, 'Art 102', 'Advanced Painting Techniques', 4, '2023-10-05 09:30:00'),
(29, 'Philosophy 102', 'Ethics and Moral Philosophy', 7, '2024-01-18 15:20:00'),
(30, 'Economics 102', 'Macroeconomic Theory', 4, '2024-06-20 14:40:00');
```

```sql
INSERT INTO Enrollments (enrollment_id, student_id, enr_course_id, enrolled_date)
VALUES
(1, 1, 1, '2019-03-15 09:00:00'),
(2, 2, 2, '2019-08-10 10:30:00'),
(3, 3, 3, '2020-01-20 11:00:00'),
(4, 4, 4, '2020-05-25 14:15:00'),
(5, 5, 5, '2020-09-10 08:45:00'),
(6, 6, 6, '2021-02-18 10:00:00'),
(7, 7, 7, '2021-07-30 13:30:00'),
(8, 8, 8, '2021-11-15 15:00:00'),
(9, 9, 9, '2022-03-05 09:20:00'),
(10, 10, 10, '2022-08-22 11:45:00'),
(11, 11, 11, '2022-11-10 14:30:00'),
(12, 12, 12, '2023-01-15 09:00:00'),
(13, 13, 13, '2023-04-28 10:10:00'),
(14, 14, 14, '2023-06-18 12:15:00'),
(15, 15, 15, '2023-09-10 14:00:00'),
(16, 16, 16, '2023-11-05 08:45:00'),
(17, 17, 17, '2024-02-12 10:15:00'),
(18, 18, 18, '2024-05-02 13:00:00'),
(19, 19, 19, '2024-07-15 14:30:00'),
(20, 20, 20, '2024-09-10 15:45:00'),
(21, 21, 1, '2019-05-01 09:00:00'),
(22, 22, 2, '2019-10-20 10:30:00'),
(23, 23, 3, '2020-03-10 11:00:00'),
(24, 24, 4, '2020-07-25 14:15:00'),
(25, 25, 5, '2020-11-11 08:45:00'),
(26, 26, 6, '2021-04-08 10:10:00'),
(27, 27, 7, '2021-09-18 13:30:00'),
(28, 28, 8, '2021-12-20 15:00:00'),
(29, 29, 9, '2022-05-05 09:20:00'),
(30, 30, 10, '2022-10-15 11:45:00'),
(31, 1, 11, '2022-12-10 14:30:00'),
(32, 2, 12, '2023-02-15 08:45:00'),
(33, 3, 13, '2023-04-25 10:15:00'),
(34, 4, 14, '2023-08-12 12:00:00'),
(35, 5, 15, '2023-12-05 14:30:00'),
(36, 6, 16, '2024-01-18 09:00:00'),
(37, 7, 17, '2024-04-05 10:30:00'),
(38, 8, 18, '2024-06-20 13:45:00'),
(39, 9, 19, '2024-08-01 15:00:00'),
(40, 10, 20, '2024-11-10 16:15:00'),
(41, 11, 1, '2019-07-15 09:30:00'),
(42, 12, 2, '2019-12-10 11:00:00'),
```

```sql
INSERT INTO Assignments (quiz_id, asg_course_id, title, assignment_created_date, due_date)
VALUES
(1, 1, 'Algebra Basics', '2019-03-10 09:00:00', '2019-03-17 23:59:00'),
(2, 2, 'Newtonian Mechanics', '2019-05-20 10:00:00', '2019-05-30 23:59:00'),
(3, 3, 'Chemical Reactions', '2019-07-15 11:00:00', '2019-07-22 23:59:00'),
(4, 4, 'Cell Biology', '2019-09-05 14:00:00', '2019-09-19 23:59:00'),
(5, 5, 'Essay: My Favorite Book', '2019-11-01 08:30:00', '2019-11-06 23:59:00'),
(6, 6, 'Introduction to Loops in Programming', '2020-01-15 10:00:00', '2020-01-25 23:59:00'),
(7, 7, 'World War I Analysis', '2020-03-10 12:00:00', '2020-03-24 23:59:00'),
(8, 8, 'Create Your Own Artwork', '2020-05-12 14:00:00', '2020-05-22 23:59:00'),
(9, 9, 'Philosophy: Ethical Dilemmas', '2020-07-01 15:00:00', '2020-07-08 23:59:00'),
(10, 10, 'Economic Trends Analysis', '2020-08-15 09:30:00', '2020-08-22 23:59:00'),
(11, 11, 'Compose a Short Melody', '2020-11-10 11:00:00', '2020-11-17 23:59:00'),
(12, 12, 'Statistical Data Analysis', '2021-02-12 13:30:00', '2021-02-22 23:59:00'),
(13, 13, 'Environmental Issues Report', '2021-04-05 10:00:00', '2021-04-12 23:59:00'),
(14, 14, 'Psychological Case Study', '2021-06-20 09:00:00', '2021-07-04 23:59:00'),
(15, 15, 'Sociological Observation', '2021-09-05 11:00:00', '2021-09-10 23:59:00'),
(16, 16, 'Political Systems Research', '2021-11-12 12:30:00', '2021-11-22 23:59:00'),
(17, 17, 'Cultural Anthropology Study', '2022-02-15 14:00:00', '2022-03-01 23:59:00'),
(18, 18, 'Literary Analysis of a Poem', '2022-04-20 15:30:00', '2022-04-27 23:59:00'),
(19, 19, 'Astronomy: Observing Stars', '2022-07-05 08:30:00', '2022-07-15 23:59:00'),
(20, 20, 'Mapping Geographic Data', '2022-09-15 10:00:00', '2022-09-22 23:59:00'),
(21, 21, 'Advanced Physics Lab Report', '2022-12-10 13:00:00', '2022-12-17 23:59:00'),
(22, 22, 'Calculus Problem Set', '2023-02-15 14:30:00', '2023-02-25 23:59:00'),
(23, 23, 'Organic Chemistry Reactions', '2023-05-05 09:30:00', '2023-05-12 23:59:00'),
(24, 24, 'Genetics Research Paper', '2023-07-01 10:00:00', '2023-07-15 23:59:00'),
(25, 25, 'Critical Analysis of a Novel', '2023-10-10 11:30:00', '2023-10-17 23:59:00'),
(26, 26, 'Algorithm Optimization Task', '2024-01-20 12:00:00', '2024-01-30 23:59:00'),
(27, 27, 'Medieval History Presentation', '2024-03-05 13:00:00', '2024-03-19 23:59:00'),
(28, 28, 'Art Portfolio Submission', '2024-05-10 15:00:00', '2024-05-20 23:59:00'),
(29, 29, 'Essay: Moral Philosophy', '2024-07-05 08:00:00', '2024-07-12 23:59:00'),
(30, 30, 'Macroeconomic Policy Analysis', '2024-08-15 10:30:00', '2024-08-25 23:59:00'),
(31, 1, 'Advanced Algebra Concepts', '2019-06-10 09:00:00', '2019-06-17 23:59:00'),
(32, 2, 'Energy Conservation Lab', '2019-08-05 10:30:00', '2019-08-15 23:59:00'),
(33, 3, 'Chemical Bonding Analysis', '2019-10-01 14:00:00', '2019-10-08 23:59:00'),
(34, 4, 'Human Anatomy Overview', '2019-12-10 15:30:00', '2019-12-24 23:59:00'),
(35, 5, 'English Literature Review', '2020-03-10 09:30:00', '2020-03-17 23:59:00'),
(36, 6, 'Building a Small Application', '2020-06-01 10:00:00', '2020-06-11 23:59:00'),
(37, 7, 'Analyzing Renaissance History', '2020-08-20 12:00:00', '2020-09-03 23:59:00'),
(38, 8, 'Design a Creative Poster', '2021-01-05 13:30:00', '2021-01-15 23:59:00'),
(39, 9, 'Philosophical Thought Exercise', '2021-03-15 15:00:00', '2021-03-22 23:59:00'),
(40, 10, 'Economy and Market Prediction', '2021-05-20 11:00:00', '2021-05-30 23:59:00'),
(41, 21, 'Quantum Physics Fundamentals', '2024-07-01 09:30:00', '2024-07-08 23:59:00'),
(42, 22, 'Advanced Calculus Exercises', '2024-07-05 10:00:00', '2024-07-12 23:59:00'),
```

```sql
INSERT INTO Feedback (feedback_id, fed_submission_id, fed_teacher_id, fed_comments)
VALUES
(1, 1, 2, 'Great work! Keep it up.'),
(2, 2, 2, 'Excellent understanding of concepts.'),
(3, 3, 4, 'Good effort, but improve your accuracy.'),
(4, 4, 4, 'Outstanding analysis!'),
(5, 5, 7, 'Needs improvement in writing skills.'),
(6, 6, 7, 'Well-organized work.'),
(7, 7, 15, 'Good, but focus on formatting.'),
(8, 8, 15, 'Interesting perspective!'),
(9, 9, 19, 'Fantastic work with detailed explanations.'),
(10, 10, 19, 'Solid performance!'),
(11, 11, 22, 'Try to address all points in the rubric.'),
(12, 12, 22, 'Clear and concise answers.'),
(13, 13, 25, 'Good, but needs better examples.'),
(14, 14, 25, 'Exceptional creativity!'),
(15, 15, 29, 'Well-researched submission.'),
(16, 16, 29, 'Satisfactory work, but refine your style.'),
(17, 17, 39, 'Keep working on your grammar.'),
(18, 18, 39, 'Great improvement compared to the last assignment.'),
(19, 19, 45, 'Excellent critical thinking displayed.'),
(20, 20, 45, 'You have exceeded expectations!'),
(21, 21, 2, 'Good improvement from the previous assignment.'),
(22, 22, 4, 'Focus more on analysis in your answers.'),
(23, 23, 7, 'Solid effort, but try to expand on key points.'),
(24, 24, 15, 'Great job, but watch for minor errors.'),
(25, 25, 19, 'Exceptional clarity and focus.'),
(26, 26, 22, 'Add more examples to strengthen your points.'),
(27, 27, 25, 'Good attempt, but your formatting needs work.'),
(28, 28, 29, 'Excellent structure and presentation.'),
(29, 29, 39, 'Creative approach, well done!'),
(30, 30, 45, 'Outstanding performance in this submission.'),
(31, 31, 2, 'Thorough analysis, well done!'),
(32, 32, 4, 'Good attention to detail, but expand further.'),
(33, 33, 7, 'Well-researched, but grammar needs work.'),
(34, 34, 15, 'Excellent organization and clarity.'),
(35, 35, 19, 'Your examples were highly effective.'),
(36, 36, 22, 'Solid work, but add more references.'),
(37, 37, 25, 'Great effort, but keep refining your arguments.'),
(38, 38, 29, 'Good job, but ensure consistent formatting.'),
(39, 39, 39, 'Excellent use of evidence to support points.'),
(40, 40, 45, 'Creative and original approach, keep it up!'),
(41, 41, 2, 'Strong effort, but improve consistency.'),
(42, 42, 4, 'Great use of visuals, keep it up!'),
(43, 43, 7, 'Good analysis, but watch formatting.'),
```

Ln

```sql
INSERT INTO Scores (score_id, score_student_id, score_course_id, score)
VALUES
(1, 1, 5, 87.5),
(2, 2, 12, 91.0),
(3, 3, 8, 78.5),
(4, 4, 15, 85.0),
(5, 5, 3, 88.0),
(6, 6, 10, 92.5),
(7, 7, 20, 80.0),
(8, 8, 6, 89.5),
(9, 9, 11, 76.5),
(10, 10, 18, 94.0),
(11, 11, 2, 81.0),
(12, 12, 14, 86.5),
(13, 13, 7, 90.0),
(14, 14, 19, 83.0),
(15, 15, 4, 88.5),
(16, 16, 9, 85.0),
(17, 17, 13, 92.0),
(18, 18, 1, 89.0),
(19, 19, 16, 95.0),
(20, 20, 17, 84.0),
(21, 1, 11, 88.0),
(22, 2, 3, 90.5),
(23, 3, 7, 85.0),
(24, 4, 18, 87.5),
(25, 5, 2, 92.0),
(26, 6, 15, 79.0),
(27, 7, 9, 91.5),
(28, 8, 4, 86.0),
(29, 9, 14, 88.5),
(30, 10, 20, 94.0),
(31, 11, 6, 89.5),
(32, 12, 10, 84.0),
(33, 13, 14, 93.5),
(34, 14, 19, 90.0),
(35, 15, 7, 87.0),
(36, 16, 12, 92.5),
(37, 17, 18, 86.5),
(38, 18, 3, 80.0),
(39, 19, 9, 91.0),
(40, 20, 4, 95.0),
(41, 1, 13, 88.0),
(42, 2, 5, 91.5),
```

Down Script (Remove Data and Tables):

```sql
DELETE FROM Feedback;
DELETE FROM Submissions;
DELETE FROM Scores;
DELETE FROM Assignments;
DELETE FROM Enrollments;
DELETE FROM Courses;
DELETE FROM Users;
```

SQL Up/Down Script of Data Logic for the External Data Model

Data Queries

Retrieve Top 5 Students with the Highest Scores:

```sql
SELECT TOP 5
    S.score_id AS ScoreID,
    CONCAT(U.first_name, ' ', U.last_name) AS StudentName,
    C.course_title AS CourseTitle,
    S.score AS Score
FROM
    Scores S
JOIN
    Users U ON S.score_student_id = U.user_id
JOIN
    Courses C ON S.score_course_id = C.c_course_id
ORDER BY
    S.score DESC;
```

Results | Messages

	ScoreID	StudentName	CourseTitle	Score
1	19	Steve Martin	Political Science 101	95
2	40	Tina Wilson	Biology 101	95
3	50	Julia Clark	Computer Science 101	94
4	30	Julia Clark	Geography 101	94
5	10	Julia Clark	Literature 101	94

Update Student Role After Graduation:

```sql
--Update Student Role After Graduation:
UPDATE Users
SET roles = 'Alumni'
WHERE roles = 'Student' AND user_id IN (
    SELECT DISTINCT student_id
    FROM Enrollments
    WHERE enrolled_date < DATEADD(YEAR, -4, GETDATE())
);

SELECT * FROM Users;
```

Results | Messages

	user_id	first_name	last_name	email_id	passwords	roles	users_created_date
1	1	Alice	Smith	alice.smith@exanple.com	P@ssw0rd1!	Alunni	2019-01-15 00:00:00.000
2	2	Bob	Johnson	bob.johnson@exanple.com	B0bJ@123	Teacher	2019-05-10 00:00:00.000
3	3	Charlie	Brown	charlie.brown@example.com	Ch@rlie!$	Alunni	2020-02-18 00:00:00.000
4	4	Diana	Prince	diana.prince@example.com	D!@naP#$	Teacher	2020-06-25 00:00:00.000
5	5	Eve	Adams	eve.adams@example.com	EveP@$$!	Alunni	2020-10-12 00:00:00.000
6	6	Frank	White	frank.white@exanple.com	Fr@nkWh1te	Student	2021-03-15 00:00:00.000

Find Courses Without Enrollments(left join)

```sql
--Find Courses Without Enrollments
SELECT
    C.c_course_id AS CourseID,
    C.course_title AS CourseTitle,
    E.enrollment_id AS EnrollmentID
FROM
    Courses C
LEFT JOIN
    Enrollments E ON C.c_course_id = E.enr_course_id
WHERE
    E.enrollment_id IS NULL;
```

Results | Messages

	CourseID	CourseTitle	EnrollmentID
1	21	Physics 102	NULL
2	22	Math 102	NULL
3	23	Chemistry 102	NULL
4	24	Biology 102	NULL
5	25	English 102	NULL
6	26	Computer Science 102	NULL

⊗ 14 ⚠ 0

Retrieve Students Enrolled in Courses with Course Details:

```sql
SELECT
    U.user_id AS StudentID,
    CONCAT(U.first_name, ' ', U.last_name) AS StudentName,

    C.course_title AS CourseTitle
FROM
    Enrollments E
JOIN
    Users U ON E.student_id = U.user_id
JOIN
    Courses C ON E.enr_course_id = C.c_course_id
WHERE
    U.roles = 'Student';
```

Results | Messages

	StudentID	StudentName	CourseTitle
1	6	Frank White	Computer Science 101
2	8	Hannah Lee	Art 101
3	9	Ian Wright	Philosophy 101
4	16	Paul Anderson	Political Science 101
5	17	Quincy Jones	Anthropology 101
6	18	Rachel Taylor	Literature 101
7	26	Zane Morris	Computer Science 101
8	27	Aiden Garcia	History 101
9	28	Bella Roberts	Art 101
10	6	Frank White	Political Science 101
11	8	Hannah Lee	Literature 101
12	9	Ian Wright	Astronomy 101
13	16	Paul Anderson	Computer Science 101
14	17	Quincy Jones	History 101
15	18	Rachel Taylor	Art 101

Retrieve Scores with Student and Course Details for Low Scores

```sql
    -- Retrieve Scores with Student and Course Details for Low Scores
SELECT
    U.user_id AS StudentID,
    CONCAT(U.first_name, ' ', U.last_name) AS Students_Name,
    C.c_course_id AS CourseID,
    C.course_title AS CourseTitle,
    S.score AS Score
FROM
    Scores S
INNER JOIN
    Users U ON S.score_student_id = U.user_id
INNER JOIN
    Courses C ON S.score_course_id = C.c_course_id
WHERE
    S.score < 50
ORDER BY
    S.score ASC;
```

Results Messages

	StudentID	Students_Name	CourseID	CourseTitle	Score
1	30	Daisy Mitchell	20	Geography 101	15
2	23	Wendy King	5	English 101	18
3	28	Bella Roberts	16	Political Science 101	24
4	22	Victor Hill	3	Chemistry 101	25.5
5	25	Yara Lopez	9	Philosophy 101	28
6	27	Aiden Garcia	14	Psychology 101	29.5
7	21	Una Walker	1	Math 101	32
8	26	Zane Morris	12	Statistics 101	35
9	29	Caleb Nelson	18	Literature 101	40
10	24	Xavier Cruz	7	History 101	42.5

Procedure to Retrieve Courses by Teacher(avoid repetative queries)

```sql
-- Procedure to Retrieve Courses by Teacher(avoid repetative queries)
    CREATE PROCEDURE GetCoursesByTeacher (@teacher_id INT)
AS
BEGIN
    SELECT
        C.c_course_id AS CourseID,
        C.course_title AS CourseTitle,
        C.course_description AS CourseDescription,
        C.courses_created_date AS CreatedDate
    FROM
        Courses C
    WHERE
        C.created_by = @teacher_id;
END;

EXEC GetCoursesByTeacher @teacher_id = 2;
```

Results | Messages

	CourseID	CourseTitle	CourseDescription	CreatedDate
1	1	Math 101	Basic Mathematics	2019-02-15 00:00:00.000
2	3	Chemistry 101	Introduction to Chemistry	2020-03-10 00:00:00.000
3	7	History 101	World History	2021-09-12 00:00:00.000
4	12	Statistics 101	Basics of Statistics	2023-03-15 00:00:00.000
5	15	Sociology 101	Introduction to Sociology	2023-12-08 00:00:00.000
6	19	Astronomy 101	Introduction to Astronomy	2024-09-15 00:00:00.000

Procedure to Retrieve Assignments for a Specific Course

```sql
CREATE PROCEDURE GetAssignmentsByCourse (@course_id INT)
AS
BEGIN
    SELECT
        A.quiz_id AS AssignmentID,
        A.title AS AssignmentTitle,
        A.assignment_created_date AS CreatedDate,
        A.due_date AS DueDate
    FROM
        Assignments A
    WHERE
        A.asg_course_id = @course_id;
END;

EXEC GetAssignmentsByCourse @course_id = 1;
```

--Index for Faster Retrieval of Enrollments by Student:

```
--Index for Faster Retrieval of Enrollments by Student:
CREATE INDEX IDX_Enrollments_Student
ON Enrollments (student_id);

--
CREATE INDEX IDX_Assignments_Course
ON Assignments (asg_course_id);
```

Down Script (Revert Logic and Indexes):

```
DROP PROCEDURE GetCoursesByTeacher
DROP PROCEDURE GetAssignmentsByCourse

DROP INDEX IDX_Enrollments_Student
DROP INDEX IDX_Assignments_Course
```