SQL Project Summary: Enrollment Data Management

The project is centred around creating and managing a student enrollment database in MySQL. This includes managing student records, courses, instructors, and exam scores through SQL tables and queries. The project also incorporates triggers for automating ID generation and leverages table joins to combine data from multiple tables for analysis. We named our database "**enrollment_data**".

Key Components

DDL and DML

1. Table Creation:

- students: Stores personal data like student_id, first_name, last_name, gender, etc.

- courses: Contains information on courses, their names, and associated instructors.

- instructors: Holds details about instructors, including their ID and names.

- entrance_exam: Tracks student performance in various courses.

We created tables for each of the entities involved in the enrollment system (students, courses, instructors, and entrance_exams). The primary key for each table was carefully defined to ensure data uniqueness. Additionally, foreign keys were used to link related tables, such as the 'instructor_id' in the 'students' table, which was linked to the 'instructors' table. The inbuilt GUI in MySQL workbench was effective for this.

2. Triggers:

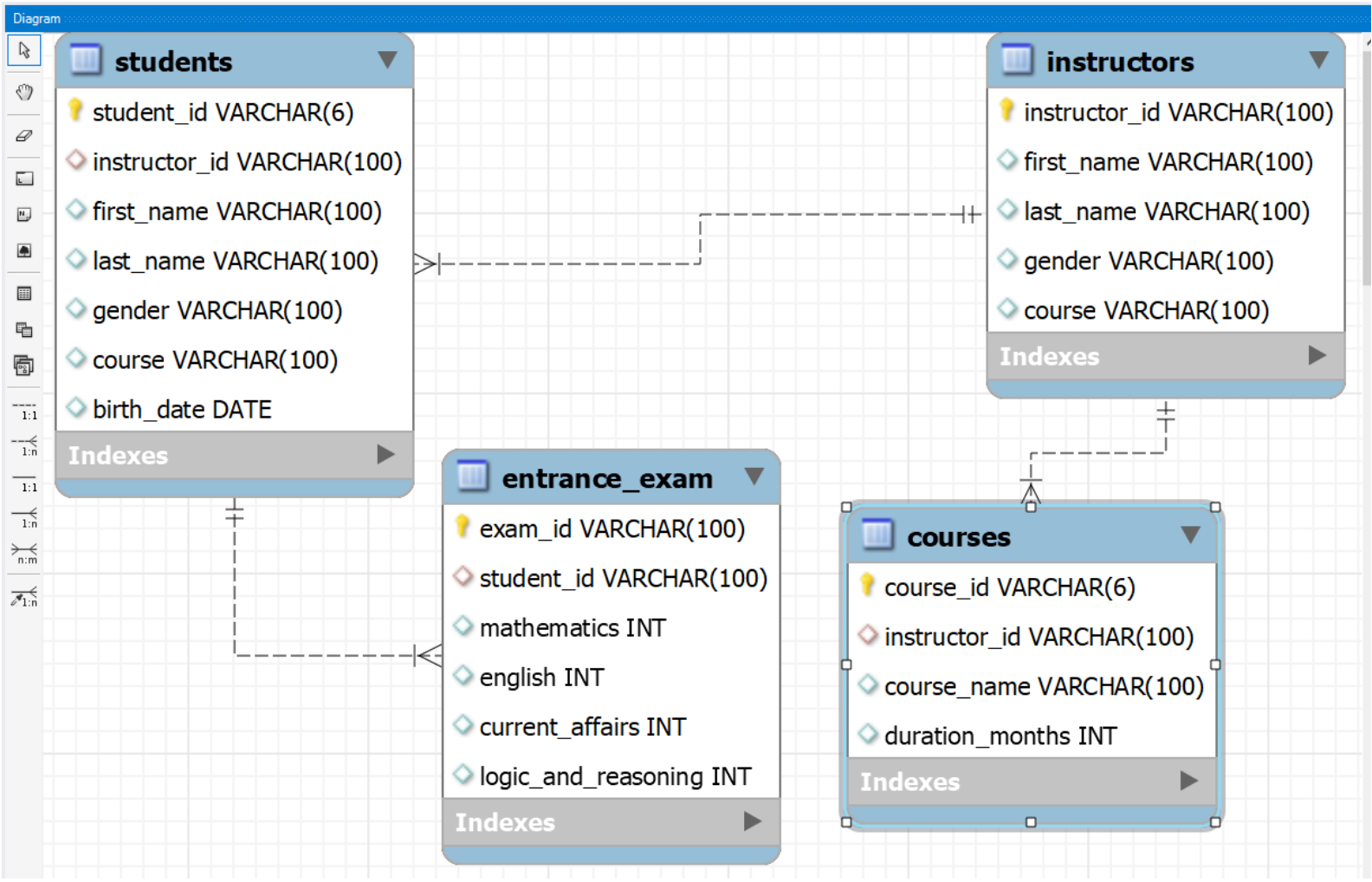
- A trigger is used to automatically generate unique student IDs upon inserting new student records.

3. Data Management; Performing Data Updates and Data Integrity Checks

- Functions to insert, update, and retrieve data are included, as well as SQL queries to join tables such as students and courses to display data in a meaningful way.

Ensuring the data is updated correctly was a critical part of the project. By using JOINs, we could dynamically update the 'instructor_id' in the 'students' table based on the courses they were enrolled in. This involved first linking the 'students' table to the 'courses' table, and then linking the 'courses' table to the 'instructors' table, ensuring that all updates were done accurately.

Below is the ERD for our Students' Enrollment Database ("**enrollment_data**"):



4. Retrieving Data Using SQL Queries: Multiple SQL queries were written to retrieve specific information from the database. For example, one of the queries retrieved all students who were assigned to a particular instructor, while another query fetched students' exam scores from the 'entrance_exam' table. We also performed various joins between tables to ensure that the retrieved data was complete and accurate. The queries used included basic SELECT statements, JOINs, and WHERE conditions. Kindly find the different queries below and their corresponding output:

A. Retrieving Data from the Tables:

-- Query 1: *Retrieving all students' names and their courses*

```
SELECT
        first_name,
        last_name,
        course
FROM
        students;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

first_name	last_name	course
James	Dauda	Data Analysis and Visualisation
Ajibike	Somolu	Software Engineering
Michael	Bintu	Data Science
Emily	Ogbonna	Data Analysis and Visualisation
Dotun	Solomon	Software Engineering
Sarah	Ajibade	Animation
David	Usoro	Product Management
Sophia	Nkemjika	Animation
James	Oladele	Data Analysis and Visualisation
Isabella	Hong	Data Analysis and Visualisation
William	Apkaeno	Cloud Computing
Iyanuoluwa	Sogbesan	Product Management
Emmanuel	Osas	Software Engineering
Ava	Okoro	Product Management
Alexander	Godonu	Game Development
Mary	Okorie	Software Engineering
Lucas	Robeson	AI/ML
Charity	Nnaemeka	Cyber Security
Benjamin	Johnson	Data Science
Amaka	Lewis	Cloud Computing
Henry	Udoh	DevOps
Ene	Young	Data Analysis and Visualisation
Matthew	Akingbade	DevOps
Evelyn	Basil	UI/UX Design
Jackson	Abuma	UI/UX Design
Alice	Saka	DevOps
Sebastian	Chukwue...	UI/UX Design
Evelyn	Okolo	AI/ML
Chinemerem	Chimaobi	Cyber Security
Chidubem	Chimaobi	Data Science

-- Query 2: *Retrieving all course names and their corresponding instructor IDs*

```
SELECT
        course_name,
        instructor_id
FROM
        courses;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

course_name	instructor_id
Data Analysis and Visualisation	INST01
Software Engineering	INST02
Data Science	INST03
Cyber Security	INST04
UI/UX Design	INST05
Product Management	INST06
Cloud Computing	INST07
DevOps	INST08
AI/ML	INST09
Game Development	INST10
Animation	INST11

-- **Query 3:** *Retrieving enrollment exam scores for mathematics, english, and logic_and_reasoning for all students*

```
SELECT
	student_id,
	mathematics,
	english,
	logic_and_reasoning
FROM
	entrance_exam;
```

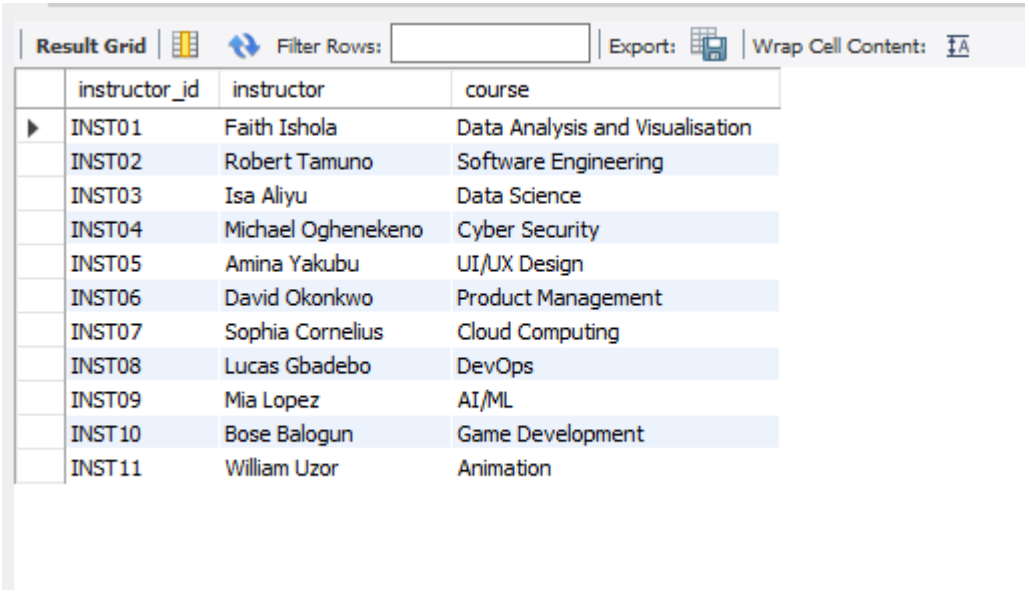
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

student_id	mathematics	english	logic_and_reasoning
STU01	93	83	97
STU02	75	93	77
STU03	65	78	94
STU04	57	76	78
STU05	84	92	75
STU06	66	76	79
STU07	96	82	69
STU08	57	91	75
STU09	93	72	79
STU10	63	80	74
STU11	61	98	89
STU12	71	90	58
STU13	60	96	83
STU14	80	98	64
STU15	68	85	62
STU16	77	94	70
STU17	72	78	79
STU18	95	83	99
STU19	79	61	78
STU20	98	62	65
STU21	62	89	87
STU22	96	93	82
STU23	63	62	88
STU24	77	78	89
STU25	94	76	89
STU26	89	91	57
STU27	59	80	93
STU28	63	69	80
STU29	67	86	86
STU30	84	91	93

entrance_exam 9

-- **Query 4:** *Retrieving all instructors' IDs, names, and their courses*

```
SELECT
	instructor_id,
	CONCAT(first_name, ' ', last_name) as instructor,
	course
FROM
	Instructors;
```

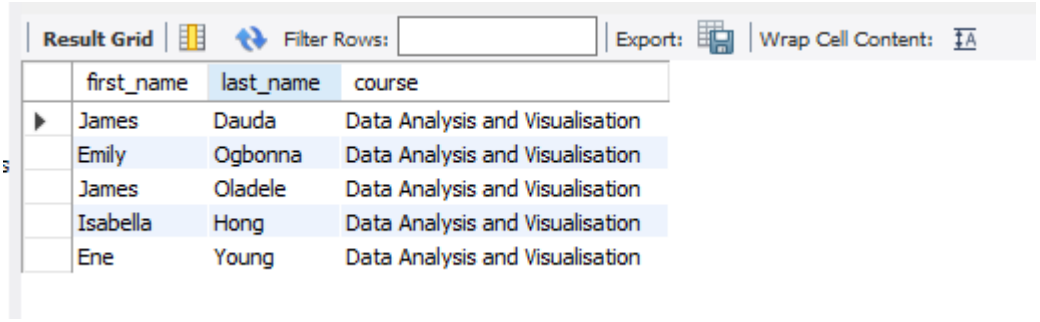


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

instructor_id	instructor	course
INST01	Faith Ishola	Data Analysis and Visualisation
INST02	Robert Tamuno	Software Engineering
INST03	Isa Aliyu	Data Science
INST04	Michael Oghenekeno	Cyber Security
INST05	Amina Yakubu	UI/UX Design
INST06	David Okonkwo	Product Management
INST07	Sophia Cornelius	Cloud Computing
INST08	Lucas Gbadebo	DevOps
INST09	Mia Lopez	AI/ML
INST10	Bose Balogun	Game Development
INST11	William Uzor	Animation

-- Query 5: *Retrieving the list of students enrolled in the Data Analysis and Visualisation course*

```
SELECT
        first_name,
        last_name,
        course
FROM
        students
WHERE
        course = 'Data Analysis and Visualisation';
```



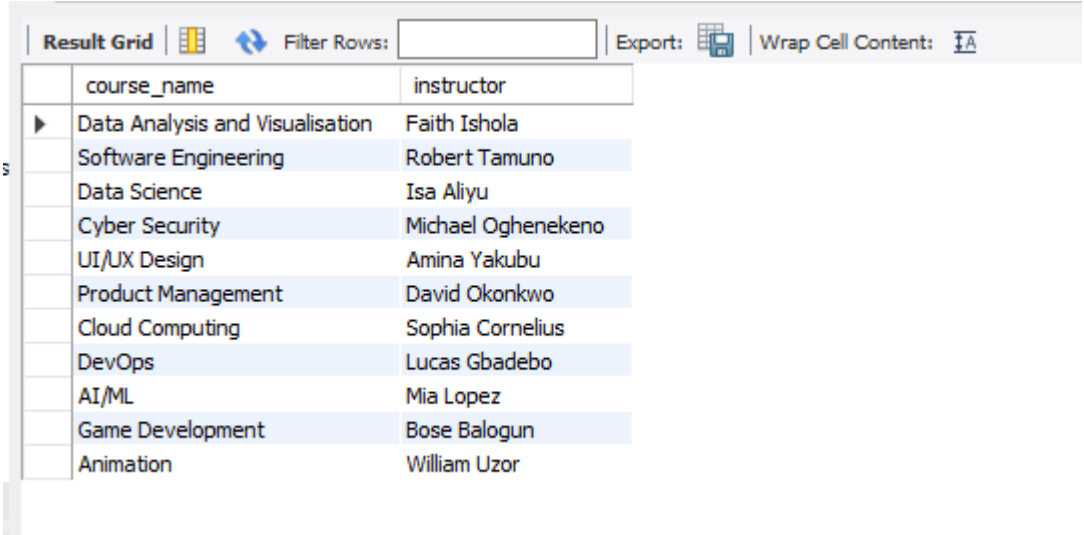
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

first_name	last_name	course
James	Dauda	Data Analysis and Visualisation
Emily	Ogbonna	Data Analysis and Visualisation
James	Oladele	Data Analysis and Visualisation
Isabella	Hong	Data Analysis and Visualisation
Ene	Young	Data Analysis and Visualisation

B. Performing Simple Joins Between Two Tables:

-- Query 1: *Joining the courses and instructors tables to display the course names and the names of their instructors*

```
SELECT
        c.course_name,
        CONCAT(i.first_name, ' ', i.last_name) as instructor
FROM
        courses AS c
JOIN
        instructors AS i
ON
        c.instructor_id = i.instructor_id;
```

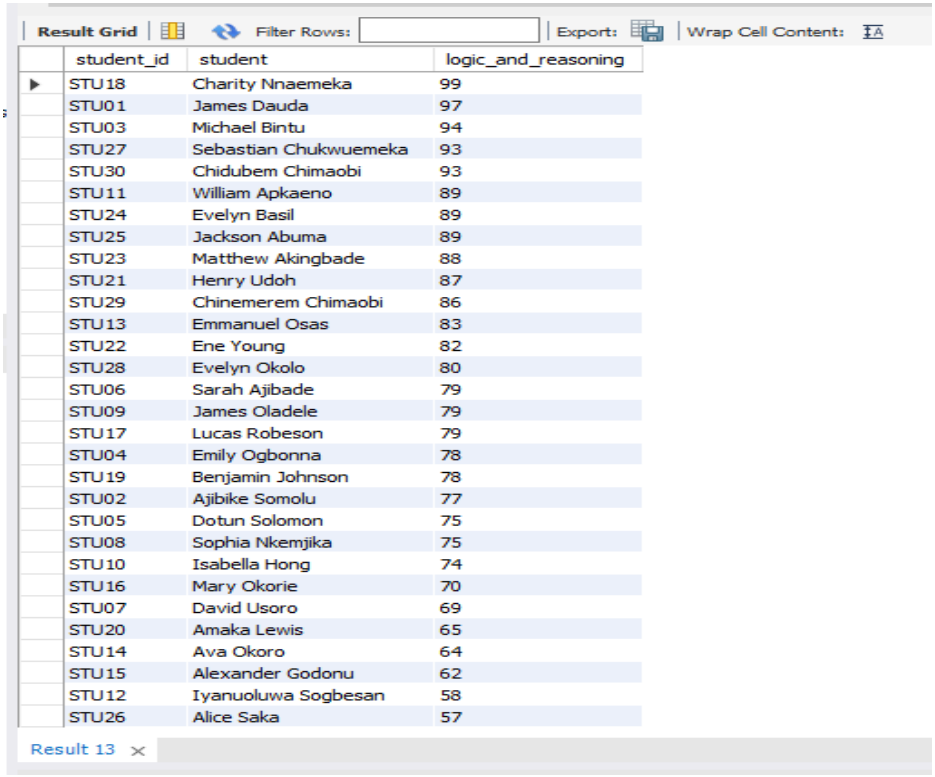


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

course_name	instructor
Data Analysis and Visualisation	Faith Ishola
Software Engineering	Robert Tamuno
Data Science	Isa Aliyu
Cyber Security	Michael Oghenekeno
UI/UX Design	Amina Yakubu
Product Management	David Okonkwo
Cloud Computing	Sophia Cornelius
DevOps	Lucas Gbadebo
AI/ML	Mia Lopez
Game Development	Bose Balogun
Animation	William Uzor

-- Query 2: *Joining the students and entrance_exam tables to display each student's ID, their names and their scores for logic and reasoning starting from highest to lowest*

```
SELECT
        s.student_id,
        CONCAT(s.first_name, ' ', s.last_name) AS student,
        e.logic_and_reasoning
FROM
        students AS s
JOIN
        entrance_exam AS e
ON
        s.student_id = e.student_id
ORDER BY
        e.logic_and_reasoning DESC;
```



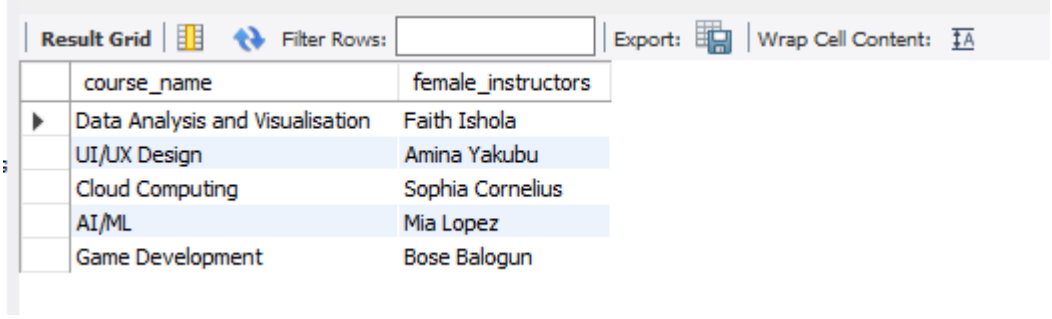
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

student_id	student	logic_and_reasoning
STU18	Charity Nnaemeka	99
STU01	James Dauda	97
STU03	Michael Bintu	94
STU27	Sebastian Chukwuemeka	93
STU30	Chidubem Chimaobi	93
STU11	William Apkaeno	89
STU24	Evelyn Basil	89
STU25	Jackson Abuma	89
STU23	Matthew Akingbade	88
STU21	Henry Udoh	87
STU29	Chinemerem Chimaobi	86
STU13	Emmanuel Osas	83
STU22	Ene Young	82
STU28	Evelyn Okolo	80
STU06	Sarah Ajibade	79
STU09	James Oladele	79
STU17	Lucas Robeson	79
STU04	Emily Ogbonna	78
STU19	Benjamin Johnson	78
STU02	Ajibike Somolu	77
STU05	Dotun Solomon	75
STU08	Sophia Nkemjika	75
STU10	Isabella Hong	74
STU16	Mary Okorie	70
STU07	David Usoro	69
STU20	Amaka Lewis	65
STU14	Ava Okoro	64
STU15	Alexander Godonu	62
STU12	Iyanuoluwa Sogbesan	58
STU26	Alice Saka	57

Result 13

-- Query 3: *Joining the courses and instructors tables to list all courses that are taught by a female instructor*

```
SELECT
	c.course_name, CONCAT(i.first_name, ' ', i.last_name) AS female_instructors
FROM
	courses AS c
JOIN
	instructors AS i
ON
	c.instructor_id = i.instructor_id
WHERE
	i.gender = 'Female';
```



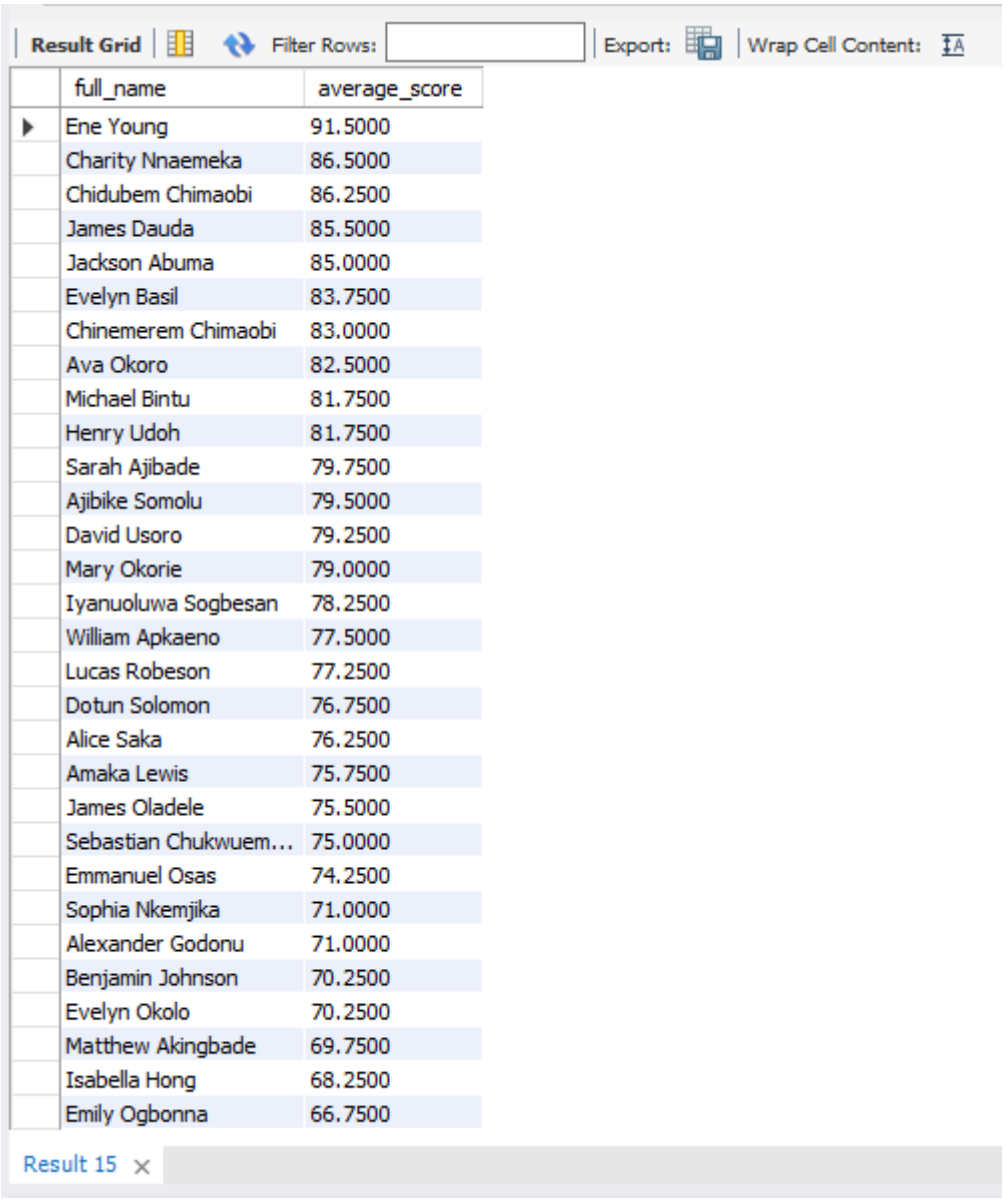
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

course_name	female_instructors
Data Analysis and Visualisation	Faith Ishola
UI/UX Design	Amina Yakubu
Cloud Computing	Sophia Cornelius
AI/ML	Mia Lopez
Game Development	Bose Balogun

-- Query 4: *Joining the students and entrance_exam tables to display the full name of each student and their average entrance exam score from highest to lowest.*

```
SELECT
	CONCAT(s.first_name, ' ', s.last_name) AS full_name,
	(e.mathematics + e.english + e.current_affairs + e.logic_and_reasoning)/4 AS
average_score
FROM
	students AS s
```

```
JOIN
        entrance_exam AS e
ON
        s.student_id = e.student_id
ORDER BY
        average_score DESC;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

full_name	average_score
Ene Young	91.5000
Charity Nnaemeka	86.5000
Chidubem Chimaobi	86.2500
James Dauda	85.5000
Jackson Abuma	85.0000
Evelyn Basil	83.7500
Chinemerem Chimaobi	83.0000
Ava Okoro	82.5000
Michael Bintu	81.7500
Henry Udoh	81.7500
Sarah Ajibade	79.7500
Ajibike Somolu	79.5000
David Usoro	79.2500
Mary Okorie	79.0000
Iyanuoluwa Sogbesan	78.2500
William Apkaeno	77.5000
Lucas Robeson	77.2500
Dotun Solomon	76.7500
Alice Saka	76.2500
Amaka Lewis	75.7500
James Oladele	75.5000
Sebastian Chukwuem...	75.0000
Emmanuel Osas	74.2500
Sophia Nkemjika	71.0000
Alexander Godonu	71.0000
Benjamin Johnson	70.2500
Evelyn Okolo	70.2500
Matthew Akingbade	69.7500
Isabella Hong	68.2500
Emily Ogbonna	66.7500

Result 15

-- Query 5: *Joining the instructors and students tables to display each instructor's name and the names of the students they are teaching.*

```
SELECT
        CONCAT(i.first_name, ' ', i.last_name) AS instructors,
        CONCAT(s.first_name, ' ', s.last_name) AS students
FROM
        instructors AS i
JOIN
        students AS s
ON
        i.instructor_id = s.instructor_id
```

```
ORDER BY
        instructors;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

instructors	students
Amina Yakubu	Evelyn Basil
Amina Yakubu	Jackson Abuma
Amina Yakubu	Sebastian Chukwuemeka
Bose Balogun	Alexander Godonu
David Okonkwo	David Usoro
David Okonkwo	Iyanuoluwa Sogbesan
David Okonkwo	Ava Okoro
Faith Ishola	James Dauda
Faith Ishola	Emily Ogbonna
Faith Ishola	James Oladele
Faith Ishola	Isabella Hong
Faith Ishola	Ene Young
Isa Aliyu	Michael Bintu
Isa Aliyu	Benjamin Johnson
Isa Aliyu	Chidubem Chimaobi
Lucas Gbadebo	Henry Udoh
Lucas Gbadebo	Matthew Akingbade
Lucas Gbadebo	Alice Saka
Mia Lopez	Lucas Robeson
Mia Lopez	Evelyn Okolo
Michael Oghen...	Charity Nnaemeka
Michael Oghen...	Chinemerem Chimaobi
Robert Tamuno	Ajibike Somolu
Robert Tamuno	Dotun Solomon
Robert Tamuno	Emmanuel Osas
Robert Tamuno	Mary Okorie
Sophia Cornelius	William Apkaeno
Sophia Cornelius	Amaka Lewis
William Uzor	Sarah Ajibade
William Uzor	Sophia Nkemjika

Result 16

In summary, effective database management was key to the success of this project. By carefully designing the database schema, defining relationships, and using SQL queries to manage and retrieve data, we were able to achieve the goals of the enrollment database system.