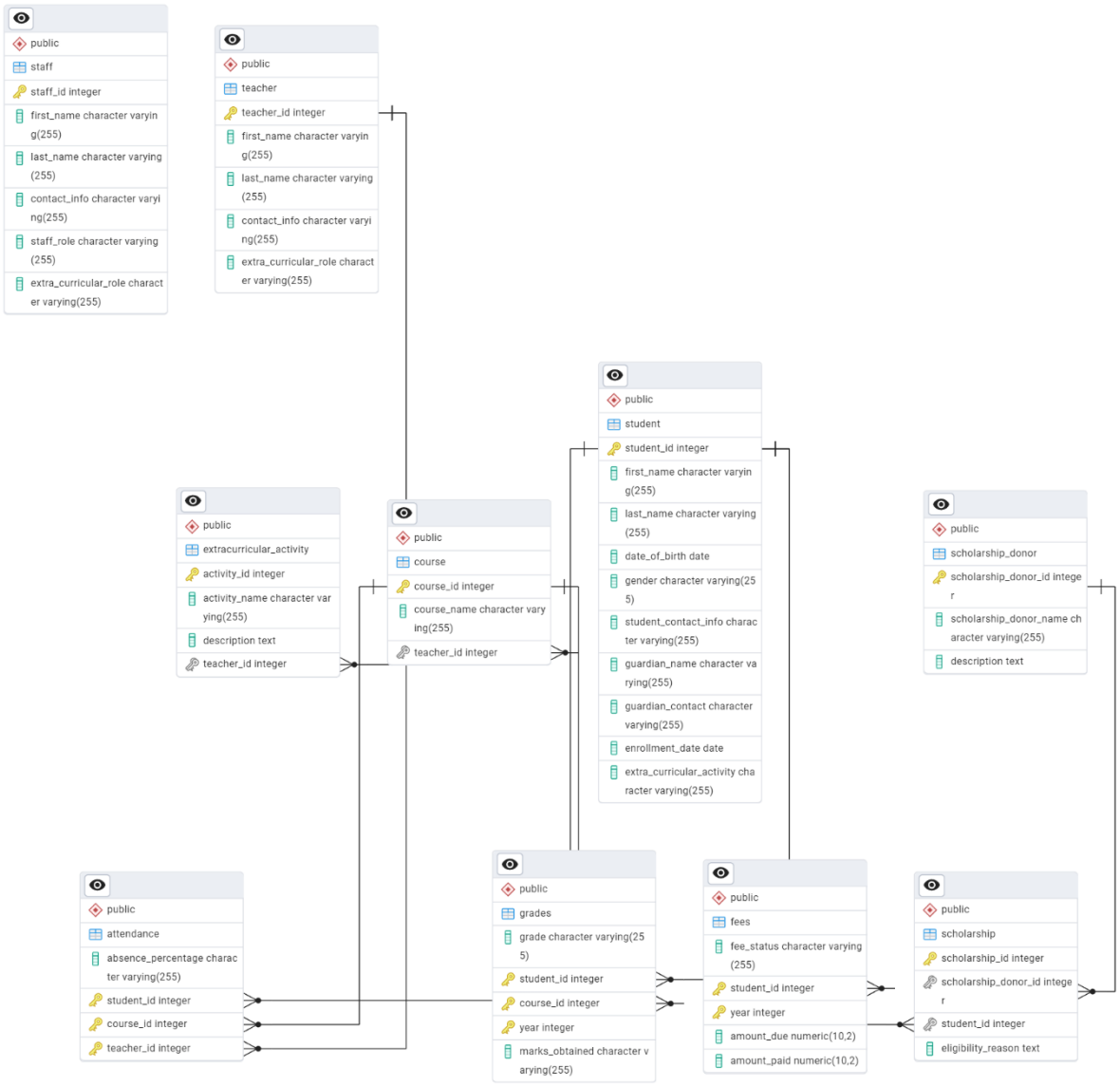
SCHOOL MANAGEMENT SYSTEM

The problem we aim to tackle is provision of quality education. We will design and implement a relational database system to work towards this SDG. This database system will make management and tracking of records in school much easier thus allowing smooth administration to take place. To function effectively, schools need to manage a lot of data which can be tiresome and easily conjure problems. This system prevents just that. Apart from the traditional data management of grades, attendance, fees etc, schools might also feel the need to have surplus personal operations for the benefit of the school and students, this also accommodates for that. (easy expansion as well). An example would be the need to share a student's details to various organizations for scholarships chances based on certain criteria.

Mainly students and teachers will interact with the system from different ends. The teachers depending on their position will have the power to make changes in the system. The students' data will be safely held, and they will all be able to gauge their respective personal progress, all while the teachers do the same to be able to plan on strategies and know where they stand as a school academically and if they are improving.

ERD

This below is the most accurately possible ERD we could make that represents our database.



Normalised tables

CREATE TABLE Staff(

staff_id int PRIMARY KEY,

first_name VARCHAR(255) NOT NULL,

last_name VARCHAR(255) NOT NULL,

contact_info VARCHAR(255) NOT NULL,

staff_role VARCHAR(255) NOT NULL,

extra_curricular_role VARCHAR(255) DEFAULT NULL

```
);

CREATE TABLE Student(
Student_id int PRIMARY KEY,
first_name VARCHAR(255) NOT NULL,
last_name VARCHAR(255) NOT NULL,
date_of_birth DATE,
gender VARCHAR(255) NOT NULL,
student_contact_info  VARCHAR(255) NOT NULL,
guardian_name VARCHAR(255) NOT NULL,
guardian_contact VARCHAR(255) NOT NULL,
enrollment_date DATE,
extra_curricular_activity VARCHAR(255) DEFAULT NULL
);

CREATE TABLE Teacher(
teacher_id int PRIMARY KEY,
first_name VARCHAR(255) NOT NULL,
last_name VARCHAR(255) NOT NULL,
contact_info VARCHAR(255) NOT NULL,
extra_curricular_role VARCHAR(255) DEFAULT NULL
);

CREATE TABLE Course(
course_id int PRIMARY KEY,
course_name VARCHAR(255) NOT NULL,
teacher_id INT,
FOREIGN KEY(teacher_id) REFERENCES Teacher(teacher_id)
);
```

```
CREATE TABLE Grades(
grade VARCHAR(255) NOT NULL, --tracked yearly
student_id INT,
course_id INT,
year INT,
marks_obtained VARCHAR(255) NOT NULL,
PRIMARY KEY(student_id,course_id,year),
FOREIGN KEY(student_id) REFERENCES Student(student_id),
FOREIGN KEY(course_id) REFERENCES Course(course_id)
);

CREATE TABLE Attendance(
absence_percentage VARCHAR(255) NOT NULL, -- tracked yearly
student_id INT,
course_id INT,
teacher_id INT,
PRIMARY KEY(student_id,course_id,teacher_id),
FOREIGN KEY(student_id) REFERENCES Student(student_id),
FOREIGN KEY(course_id) REFERENCES Course(course_id),
FOREIGN KEY(teacher_id) REFERENCES Teacher(teacher_id)
);

CREATE TABLE Fees(
fee_status  VARCHAR(255) NOT NULL,     -- tracked yearly
student_id INT,
year INT,
amount_due DECIMAL(10,2) DEFAULT NULL,
amount_paid DECIMAL(10,2) NOT NULL,
```

```
PRIMARY KEY(student_id,year),
FOREIGN KEY(student_id) REFERENCES Student(student_id)
);

CREATE TABLE Extracurricular_Activity (
activity_id INT PRIMARY KEY,
activity_name VARCHAR(255),
description TEXT,
teacher_id INT, --the patron in charge
FOREIGN KEY (teacher_id) REFERENCES Teacher(teacher_id)
);

CREATE TABLE Scholarship_Donor (
scholarship_donor_id INT PRIMARY KEY,
scholarship_donor_name VARCHAR(255),
description TEXT
);

CREATE TABLE Scholarship (
scholarship_id INT PRIMARY KEY,
scholarship_donor_id INT,
student_id INT,
eligibility_reason TEXT,
FOREIGN KEY (scholarship_donor_id) REFERENCES
Scholarship_Donor(scholarship_donor_id),
FOREIGN KEY (student_id) REFERENCES Student(student_id)
);
```

Amazing navigation queries

Some with examples

```
-- ========================
-- NAVIGATION QUERY SECTION
-- ========================

-- 1. Students with their grades and course names
SELECT
    s.student_id,
    s.first_name,
    s.last_name,
    c.course_name,
    g.grade,
    g.marks_obtained
FROM
    Student s
JOIN Grades g ON s.student_id = g.student_id
JOIN Course c ON g.course_id = c.course_id;
```

	student_id integer	first_name character varying (255)	last_name character varying (255)	course_name character varying (255)	grade character varying (255)	marks_obtained character varying (255)
1	103	Jerry	Niva	INTERNATIONAL RELATIONS	B	449
2	101	Mark	Gitau	INTERNATIONAL RELATIONS	A	499
3	102	Harriet	Hurruey	BBIT	D	399

```
-- 2. Teachers with the courses they teach
SELECT
    t.teacher_id,
    t.first_name,
    t.last_name,
```

```
    c.course_name
FROM
    Teacher t
JOIN Course c ON t.teacher_id = c.teacher_id;
```

	teacher_id integer	first_name character varying (255)	last_name character varying (255)	course_name character varying (255)
1	47	Donald	Trump	INTERNATIONAL RELATIONS
2	45	Patrick	Kamau	BBIT

```
-- 3. Students with their fee status and balance
SELECT
    s.student_id,
    s.first_name,
    s.last_name,
    f.fee_status,
    f.amount_due,
    f.amount_paid
FROM
    Student s
JOIN Fees f ON s.student_id = f.student_id;
```

	student_id integer	first_name character varying (255)	last_name character varying (255)	fee_status character varying (255)	amount_due numeric (10,2)	amount_paid numeric (10,2)
1	101	Mark	Gitau	CLEARED	[null]	5000.00
2	102	Harriet	Hurruey	NOT CLEARED	1000.00	4000.00
3	103	Jerry	Niva	CLEARED	[null]	5000.00

```
-- 4. Attendance details per course and teacher
SELECT
    s.student_id,
    s.first_name,
    s.last_name,
    c.course_name,
```

```
    t.first_name AS teacher_first_name,
    t.last_name AS teacher_last_name,
    a.absence_percentage
FROM
    Attendance a
JOIN Student s ON a.student_id = s.student_id
JOIN Course c ON a.course_id = c.course_id
JOIN Teacher t ON a.teacher_id = t.teacher_id;
```

	student_id integer	first_name character varying (255)	last_name character varying (255)	course_name character varying (255)	teacher_first_name character varying (255)	teacher_last_name character varying (255)	absence_percentage character varying (255)
1	101	Mark	Gitau	INTERNATIONAL RELATIONS	Patrick	Kamau	10
2	102	Harriet	Hurruey	BBIT	Donald	Trump	31
3	103	Jerry	Niva	INTERNATIONAL RELATIONS	Patrick	Kamau	29

Successfully run. Total query runtime: 89 msec. 3 ro

```
-- 5. Which students are taught by which teachers
SELECT
    DISTINCT s.student_id,
    s.first_name,
    s.last_name,
    t.first_name AS teacher_first_name,
    t.last_name AS teacher_last_name,
    c.course_name
FROM
    Student s
JOIN Grades g ON s.student_id = g.student_id
JOIN Course c ON g.course_id = c.course_id
JOIN Teacher t ON c.teacher_id = t.teacher_id;
```

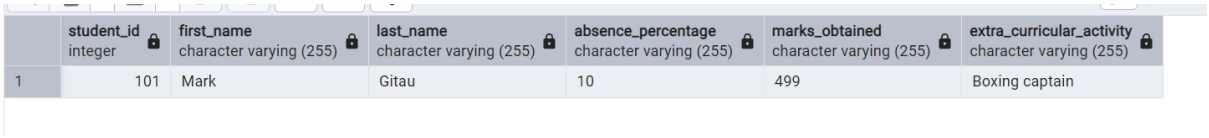
	student_id integer	first_name character varying (255)	last_name character varying (255)	teacher_first_name character varying (255)	teacher_last_name character varying (255)	course_name character varying (255)
1	103	Jerry	Niva	Donald	Trump	INTERNATIONAL RELATIONS
2	101	Mark	Gitau	Donald	Trump	INTERNATIONAL RELATIONS
3	102	Harriet	Hurruey	Patrick	Kamau	BBIT

```
-- 6. Students enrolled in each course with teacher info
```

```
SELECT
    c.course_name,
    t.first_name AS teacher_first_name,
    t.last_name AS teacher_last_name,
    s.student_id,
    s.first_name,
    s.last_name
FROM
    Course c
JOIN Teacher t ON c.teacher_id = t.teacher_id
JOIN Grades g ON c.course_id = g.course_id
JOIN Student s ON g.student_id = s.student_id;

-- 7. Students viable to receive scholarship
SELECT
    s.student_id,
    s.first_name,
    s.last_name,
    a.absence_percentage,
    g.marks_obtained,
    s.extra_curricular_activity
FROM
    Student s
JOIN Attendance a ON s.student_id = a.student_id
JOIN Grades g ON s.student_id = g.student_id
JOIN Course c ON g.course_id = c.course_id
JOIN Teacher t ON c.teacher_id = t.teacher_id
WHERE
    CAST(a.absence_percentage AS DECIMAL) < 30
```

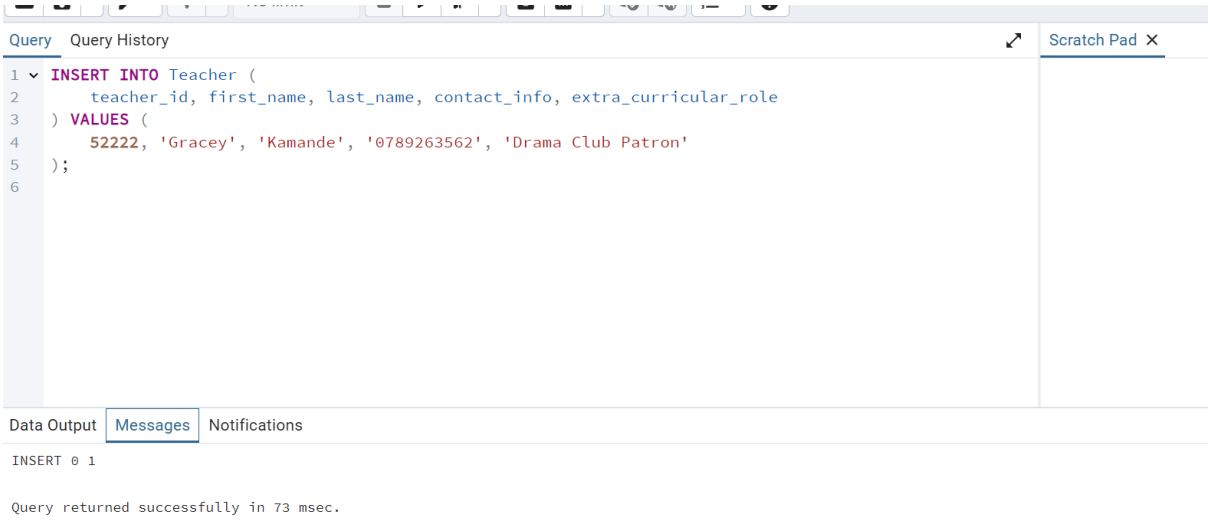
```
    AND CAST(g.marks_obtained AS DECIMAL) > 450
    AND s.extra_curricular_activity IS NOT NULL;  -- Ensure student is involved in an extracurricular activity
```



	student_id integer	first_name character varying (255)	last_name character varying (255)	absence_percentage character varying (255)	marks_obtained character varying (255)	extra_curricular_activity character varying (255)
1	101	Mark	Gitau	10	499	Boxing captain

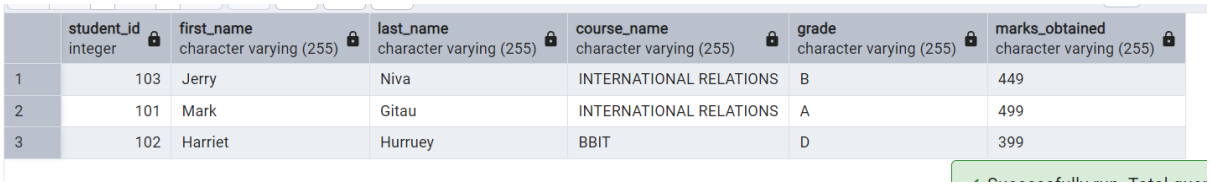
```
-- 8. View all students
SELECT * FROM Student;


-- 9. To insert a new record
INSERT INTO Teacher (
    teacher_id, first_name, last_name, contact_info, extra_curricular_role
) VALUES (
    52222, 'Gracey', 'Kamande', '0789263562', 'Drama Club Patron'
);
```




```
-- 10. To show students progress
SELECT
    s.student_id,
```

```
    s.first_name,
    s.last_name,
    c.course_name,
    g.grade,
    g.marks_obtained
FROM
    Grades g
JOIN Student s ON g.student_id = s.student_id
JOIN Course c ON g.course_id = c.course_id;
```



	student_id integer	first_name character varying (255)	last_name character varying (255)	course_name character varying (255)	grade character varying (255)	marks_obtained character varying (255)
1	103	Jerry	Niva	INTERNATIONAL RELATIONS	B	449
2	101	Mark	Gitau	INTERNATIONAL RELATIONS	A	499
3	102	Harriet	Hurruey	BBIT	D	399

The screenshots are only but a few of the results obtained from the flawless queries that we felt may be of use to a school. The results brought forth by the queries will be of use to teachers are they monitor their students and to students as they monitor their personal progress. This will result in much smoother operations and retrieval of much required data. It will also be less tiring and much easier to do. (no funds spent on training).