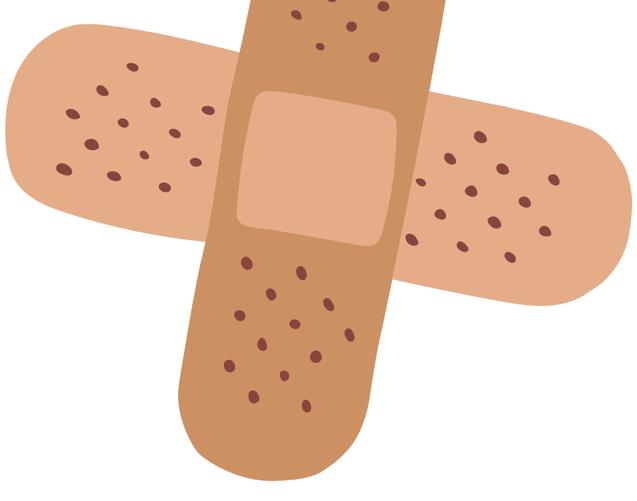


CODE FIRST GIRLS

SQL proejct

ELECTRONIC HEALTH RECORD

by Maria Grabar

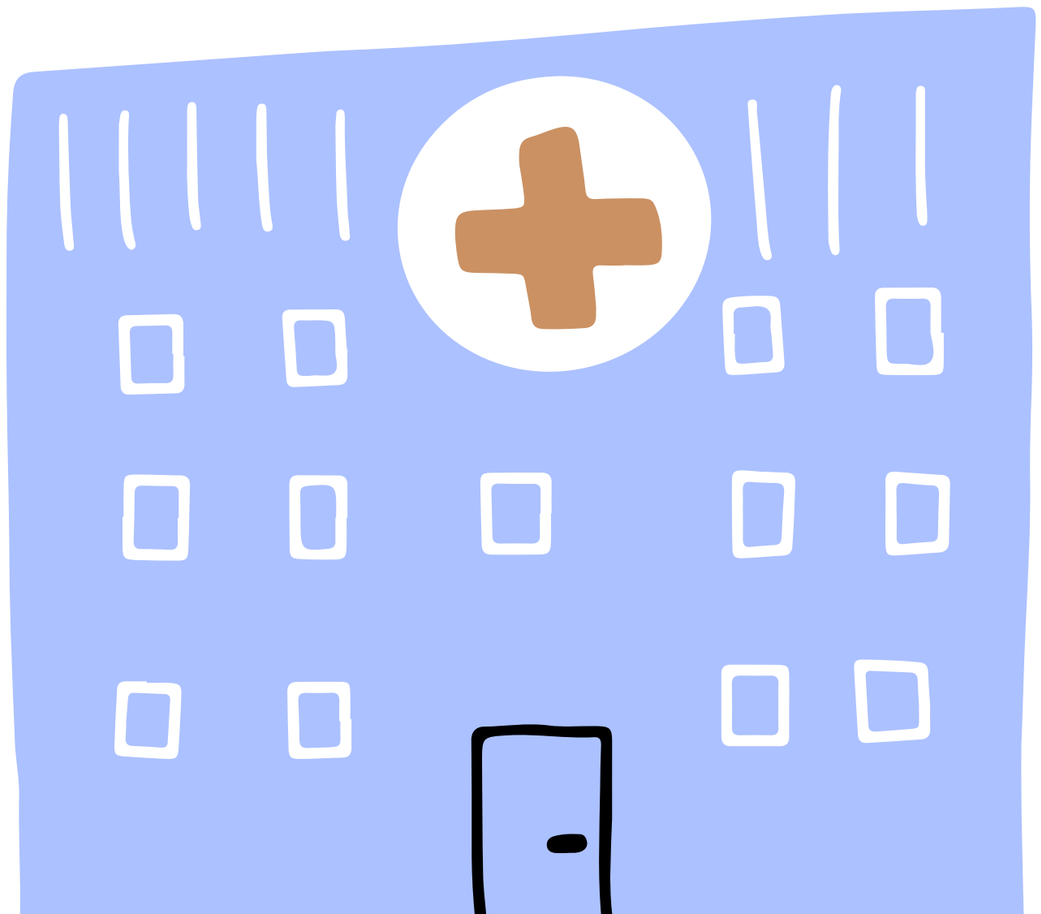
PROJECT OVERVIEW

For the past 2 months, I've been learning how to use SQL to manipulate and analyze databases and generate meaningful insights. As a part of the course, I decided to create an **Electronic Health Record (EHR) database.**

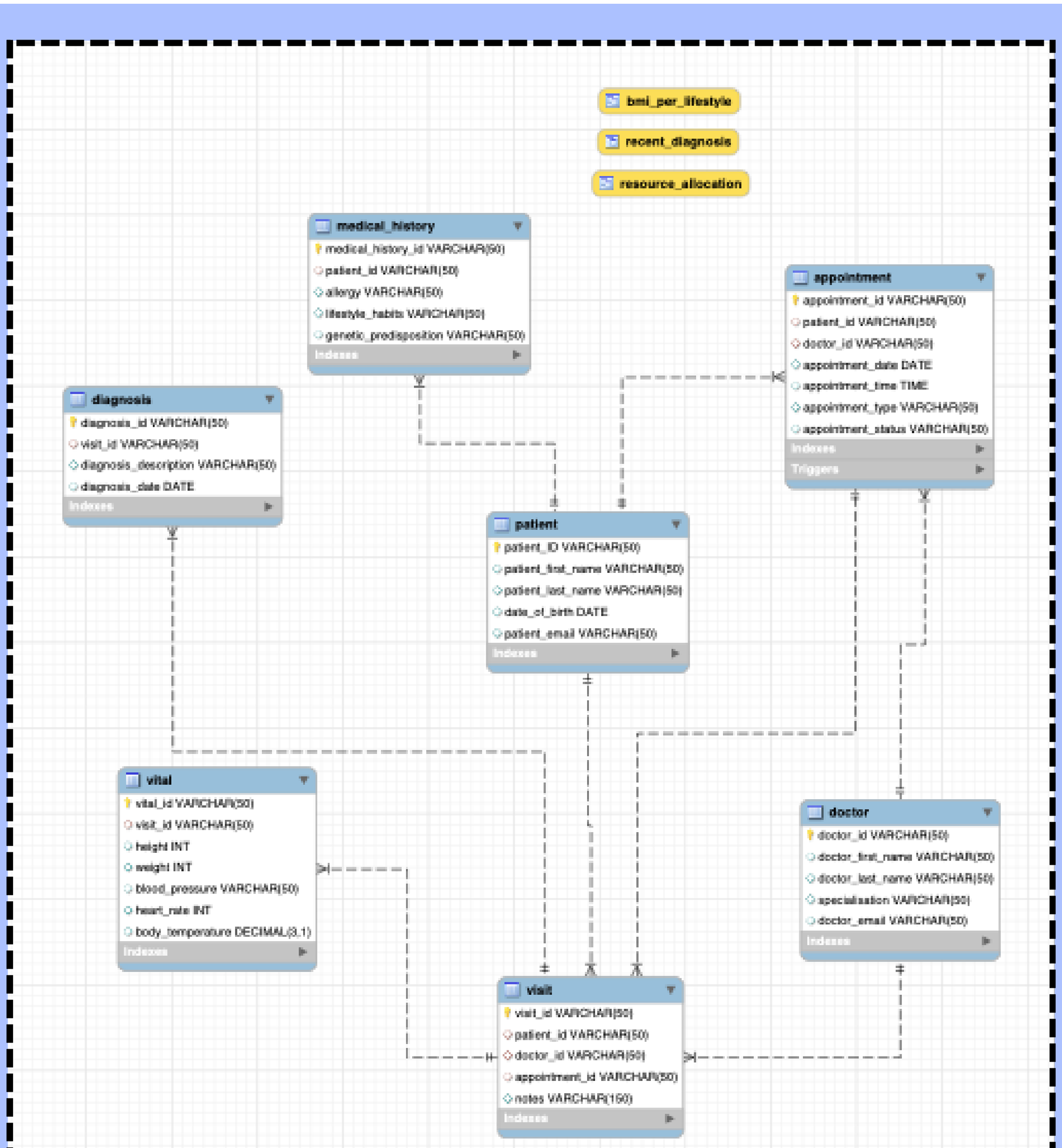
This database serves as a comprehensive system for organizing hospital-related information – patient's and doctor's details, medical history, upcoming appointments, and more.

Having an EHR system in real-life clinics helps doctors make prompt evidence-based decisions and facilitates the transfer of private information between different medical organizations.

This project isn't just tables and queries; it's my way of using tech to contribute to something I care about. It's a journey of learning, problem-solving, and seeing how what we code impacts the real world.



ER DIAGRAM



This ER diagram comprises 7 tables: **patient, medical information, doctor, appointment, visit, vital, and diagnosis**. It also represents the relationship between tables and provides insight into quick views: **resource allocation, recent diagnosis and BMI**.

TABLES

A table serves as the repository for data in a relational database. The illustrations below depict the structure of various tables within my EHR database.

```sql
CREATE TABLE patient (
patient_ID VARCHAR(50) PRIMARY KEY,
patient_first_name VARCHAR(50),
patient_last_name VARCHAR(50),
date_of_birth DATE,
patient_email VARCHAR(50));
```

patient_ID	patient_first_na...	patient_last_na...	date_of_bir...	patient_email
P0001	Dexter	Mitchell	1986-08-15	dexter@email.com
P0002	Stella	Reynolds	1998-04-02	stella@email.com
P0003	Caleb	Turner	2005-09-23	caleb@email.com
P0004	Nora	Harrison	1959-02-11	nora@email.com
P0005	Ethan	Palmer	2018-06-07	ethan@email.com
P0006	Zoey	Foster	2016-10-30	zoey@email.com
P0007	Oliver	Jensen	1987-12-25	oliver@email.com
P0008	Maya	Bennet	1951-03-18	maya@email.com
P0009	Leo	Chambers	2009-01-14	leo@email.com
P0010	Ava	Richardson	2003-07-09	ava@email.com
NULL	NULL	NULL	NULL	NULL

TABLE 1: PATIENT (patient ID, first name, last name, date of birth, email)

```sql
CREATE TABLE medical_history (
medical_history_id VARCHAR(50) PRIMARY KEY,
patient_id VARCHAR(50),
FOREIGN KEY (patient_id) REFERENCES patient(patient_id),
allergy VARCHAR(50),
lifestyle_habits VARCHAR(50),
genetic_predisposition VARCHAR(50));
```

medical_history...	patient_id	allergy	lifestyle_habits	genetic_predisposition
MH0001	P0001	Local anesthetics	High sugar diet	NULL
MH0002	P0002	Dust	Regular exercise	IBS
MH0003	P0003	NULL	Sedentary lifestyle	NULL
MH0004	P0004	Gluten	NULL	Celiac disease
MH0005	P0005	NULL	NULL	NULL
MH0006	P0006	Dust	NULL	Asthma
MH0007	P0007	Pollen	NULL	NULL
P0008	NULL	Sedentary lifestyle	Type 1 diabetes	
P0009	NULL	Intense exercise	Scoliosis	
P0010	Local anesthetics	High protein diet	Type 1 diabetes	
NULL	NULL	NULL	NULL	

TABLE 2: MEDICAL HISTORY (medical history ID, patient ID, allergy, lifestyle_habits, genetic predisposition)

```sql
CREATE TABLE doctor (
doctor_id VARCHAR(50) PRIMARY KEY,
doctor_first_name VARCHAR(50),
doctor_last_name VARCHAR(50),
specialisation VARCHAR(50),
doctor_email VARCHAR(50));
```

doctor_id	doctor_first_na...	doctor_last_na...	specialisation	doctor_email
D0001	Lucas	Parker	General practitioner	lucas@email.com
D0002	Isabella	Thompson	General practitioner	isabella@email.com
D0003	Mason	Carter	Dentist	mason@email.com
D0004	Chloe	Williams	Pediatrician	chloe@email.com
D0005	Noah	Anderson	Pediatrician	noah@email.com
D0006	Lily	Hayes	Dentist	lily@email.com
NULL	NULL	NULL	NULL	NULL

TABLE 3: DOCTOR (doctor ID, first name, last name, specialization, email)

```sql
CREATE TABLE appointment (
appointment_id VARCHAR(50) PRIMARY KEY,
patient_id VARCHAR(50),
FOREIGN KEY (patient_id) REFERENCES patient(patient_id),
doctor_id VARCHAR(50),
FOREIGN KEY (doctor_id) REFERENCES doctor(doctor_id),
appointment_date DATE,
appointment_time TIME,
appointment_type VARCHAR(50),
appointment_status VARCHAR(50));
```

TABLE 4: APPOINTMENT (appointment ID, patient ID, doctor ID, date, time, type, status)

appointment...	patient_id	doctor_id	appointment_d...	appointment_ti...	appointment_type	appointment
A0001	P0001	D0006	2023-11-19	14:00:00	Preoperative assessment	Completed
A0002	P0002	D0002	2023-11-21	11:30:00	Routine checkup	Completed
A0003	P0003	D0001	2023-11-22	14:30:00	Routine checkup	Rescheduled
A0004	P0003	D0001	2023-11-23	14:00:00	Routine checkup	Completed
A0005	P0005	D0004	2023-11-24	17:45:00	Follow-up appointment	Completed
A0006	P0001	D0006	2023-11-25	12:00:00	Surgery	Scheduled
A0007	P0009	D0005	2023-11-24	11:30:00	Specialist consultation	Completed
A0008	P0006	D0004	2023-11-24	11:45:00	Follow-up appointment	Completed
A0009	P0008	D0002	2023-11-28	10:15:00	Routine checkup	Pending
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```sql
CREATE TABLE visit (
visit_id VARCHAR(50) PRIMARY KEY,
patient_id VARCHAR(50),
FOREIGN KEY (patient_id) REFERENCES patient(patient_id),
doctor_id VARCHAR(50),
FOREIGN KEY (doctor_id) REFERENCES doctor(doctor_id),
appointment_id VARCHAR(50),
FOREIGN KEY (appointment_id) REFERENCES appointment(appointment_id),
notes VARCHAR(150));
```

TABLE 5: VISIT (visit ID, patient ID, doctor ID, appointment ID, notes)

visit_id	patient_id	doctor_id	appointment...	notes
V0001	P0001	D0006	A0001	Advanced dental decay in tooth #14
V0002	P0002	D0002	A0002	Advised on maintaining healthy lifestyle
V0003	P0003	D0001	A0004	Common cold confirmed; rest and hydration rec...
V0004	P0005	D0004	A0005	Genetic testing and serological blood tests confi...
V0005	P0009	D0005	A0007	Physical therapy recommended
V0007	P0006	D0004	A0008	Inhaler given
NULL	NULL	NULL	NULL	NULL

```sql
CREATE TABLE vital (
vital_id VARCHAR(50) PRIMARY KEY,
visit_id VARCHAR(50),
FOREIGN KEY (visit_id) REFERENCES visit(visit_id),
height INT,
weight INT,
blood_pressure VARCHAR(50),
heart_rate INT,
body_temperature DECIMAL(3,1));
```

TABLE 6: VITAL (vital ID, visit ID, height, weight, blood pressure, heart rate, body temperature)

vital_id	visit_id	height	weight	blood_pressu...	heart_rate	body_temperatu...
VT0001	V0002	162	54	120/80	77	36.7
VT0002	V0003	173	61	119/81	100	37.0
VT0003	V0005	170	55	NULL	NULL	NULL
VT0004	V0007	115	25	90/70	110	36.5
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```sql
CREATE TABLE diagnosis (
diagnosis_id VARCHAR(50) PRIMARY KEY,
visit_id VARCHAR(50),
FOREIGN KEY (visit_id) REFERENCES visit(visit_id),
diagnosis_description VARCHAR(50),
diagnosis_date DATE);
```

diagnosis_id	visit_id	diagnosis_descripti...	diagnosis_da
DG0001	V0001	Dental decay	2023-11-19
DG0002	V0003	Common cold	2023-11-23
DG0003	V0004	Celiac disease	2023-11-24
DG0004	V0005	Scoliosis	2023-11-24
DG0005	V0007	Asthma	2023-11-24
NULL	NULL	NULL	NULL

TABLE 7: DIAGNOSIS (diagnosis ID, visit ID, description, date)

Using primary and foreign keys, I created relations between different tables to ensure referential integrity within the database.

JOINS & VIEWS

```sql
CREATE VIEW recent_diagnosis AS
SELECT
    p.patient_first_name,
    p.patient_last_name,
    d.diagnosis_description
FROM patient p
JOIN visit v ON p.patient_id = v.patient
JOIN diagnosis d ON v.visit_id = d.visit
ORDER BY p.patient_first_name ASC;
```

patient_first_na...	patient_last_na...	diagnosis_descripti...
Caleb	Turner	Common cold
Dexter	Mitchell	Dental decay
Ethan	Palmer	Celiac disease
Leo	Chambers	Scoliosis
Zoey	Foster	Asthma

```sql
CREATE VIEW resource_allocation AS
SELECT
    d.doctor_first_name,
    d.doctor_last_name,
    a.appointment_date
FROM doctor d
LEFT JOIN appointment a ON d.doctor_id = a.docto
```

doctor_first_na...	doctor_last_na...	appointment_d...
Lucas	Parker	2023-11-22
Lucas	Parker	2023-11-23
Isabella	Thompson	2023-11-21
Isabella	Thompson	2023-11-28
Mason	Carter	NULL
Chloe	Williams	2023-11-24
Chloe	Williams	2023-11-24
Noah	Anderson	2023-11-24
Lily	Hayes	2023-11-19
Lily	Hayes	2023-11-25

These relations allow us to join tables and extract the desired pieces of information.

For example, what diagnosis is assigned to each patient or whether there is a doctor that doesn't have any appointments booked.

STORED PROCEDURE

```sql
DELIMITER //
CREATE PROCEDURE update_appointment_status(
    IN update_appointment_id VARCHAR(50),
    IN update_status VARCHAR(50))
BEGIN
    UPDATE appointment
    SET appointment_status = update_status
    WHERE appointment_id = update_appointment_id;

END //
DELIMITER ;

CALL update_appointment_status('A0011', 'Completed');
```

appointment...	appointment_stat...
A0011	Scheduled

appointment...	appointment_stat...
A0011	Completed

Stored procedures simplify tasks, promote code reusability, and ensure consistency in database operations. This procedure makes updating appointment status easy with just one line of code, providing a straightforward way to manage appointments.

EVENT

patient_id	entry_timestamp
P0013	2023-11-25 16:08:30
P0014	2023-11-25 16:29:39

```sql
CREATE TABLE patient_entry_log (
    patient_id VARCHAR(50),
    entry_timestamp TIMESTAMP);

DELIMITER //
CREATE EVENT log_patient_entry_time
ON SCHEDULE EVERY 5 MINUTE
STARTS NOW()
DO BEGIN
    INSERT INTO patient_entry_log (patient_id, entry_timestamp)
    SELECT patient_id, NOW()
    FROM patient
    WHERE patient_id NOT IN (SELECT patient_id FROM patient_entry_log);
END //
DELIMITER ;

INSERT INTO patient
VALUES
('P0014', 'Anna', 'Smith', '2001-06-06', 'anna@email.com');
```

For clinical audits, keeping a record of changes and updates in the database is useful for maintaining an accurate and comprehensive history of patient-related activities.

This event, scheduled every 5 minutes, automatically logs the date and time when new patient IDs are added to a separate table.

STORED FUNCTION

```sql
DELIMITER //
CREATE FUNCTION calculate_bmi(height FLOAT, weight FLOAT)
RETURNS FLOAT
DETERMINISTIC
BEGIN
    DECLARE bmi FLOAT;
    DECLARE b_height FLOAT;
    DECLARE b_weight FLOAT;
    SET b_height = height;
    SET b_weight = weight;
    SET bmi = b_weight / ((b_height/100) * (b_height/100));
    RETURN bmi;
END //
DELIMITER ;
```

Stored functions promote reusability, and contribute to efficient data processing in the database.

This function simplifies the calculation of a patient's BMI using previously entered weight and height.

patient_first_na...	patient_last_na...	bmi	bl[illegible]_pressu...	heart_rate	allergy	lifestyle_habits	genetic_predisposit...
Stella	Reynolds	20.5761	120/80	77	Dust	Regular exercise	IBS
Caleb	Turner	20.3816	119/81	100	NULL	Sedentary lifestyle	NULL
Leo	Chambers	19.0311	NULL	NULL	NULL	Intense exercise	Scoliosis

TRIGGER

```sql
DELIMITER //
CREATE TRIGGER prevent_double_booking
BEFORE INSERT ON appointment
FOR EACH ROW
BEGIN
    IF EXISTS (
        SELECT 1
        FROM appointment
        WHERE doctor_id = NEW.doctor_id
            AND appointment_date = NEW.appointment_date)
    THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'This timeslot is not available.';
    END IF;
END;
//
DELIMITER ;
```

Triggers enforce data integrity and automate responses to specific events.

This trigger activates whenever someone attempts to book an unavailable slot in the database.

Response

Error Code: 1644. This timeslot is not available.

QUERY WITH A SUBQUERY

```sql
SELECT
    p.patient_id,
    p.patient_first_name,
    p.patient_last_name,
    (
        SELECT
            COUNT(sub_a.appointment_id)
        FROM appointment sub_a
        WHERE sub_a.patient_id = p.patient_id
    ) AS appointment_count
FROM patient p
JOIN appointment a ON p.patient_id = a.patient_id
GROUP BY p.patient_id
HAVING COUNT(a.appointment_id) > 1;
```

patient_id	patient_first_na...	patient_last_na...	appointment_co...
P0001	Dexter	Mitchell	3
P0003	Caleb	Turner	2

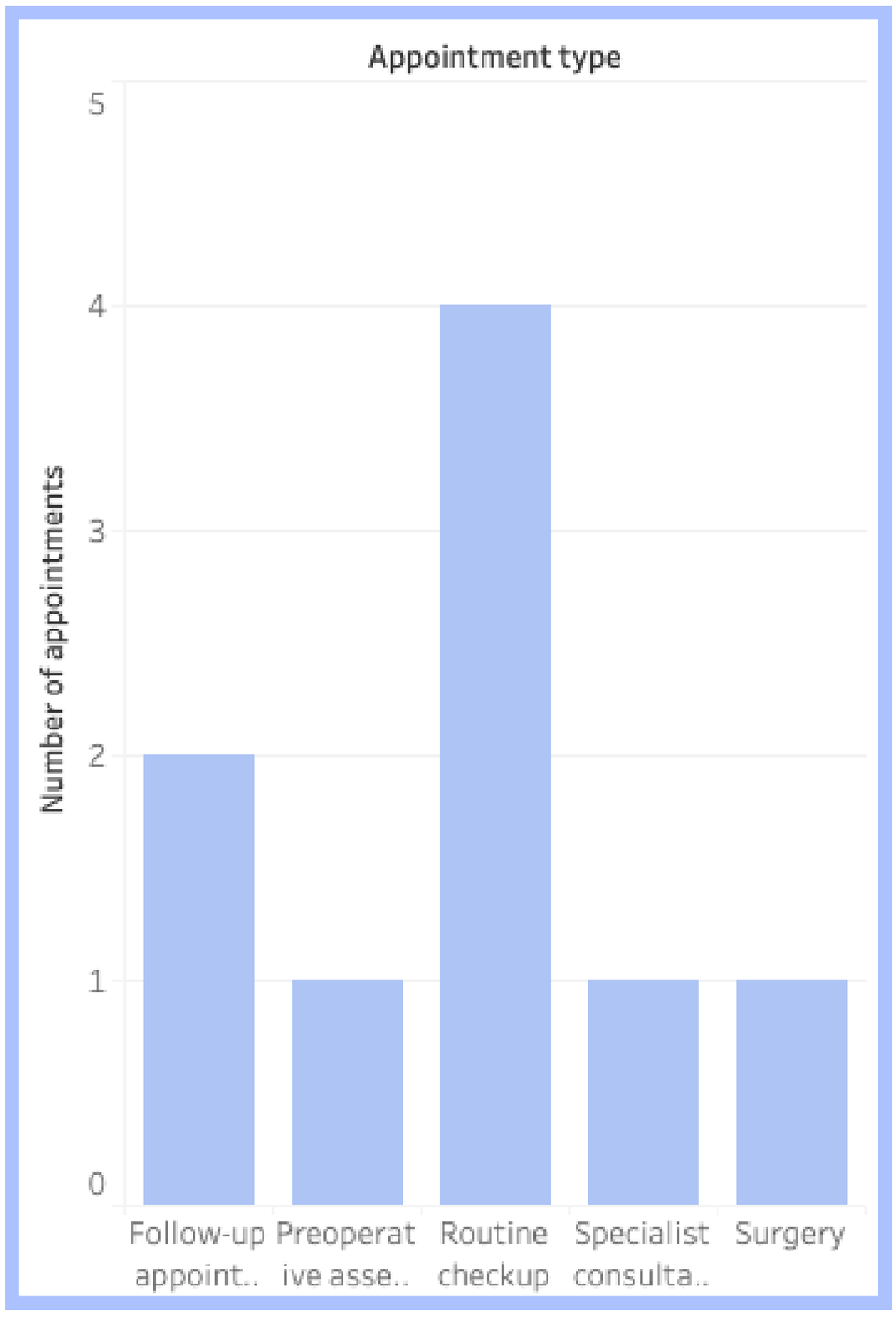
Subqueries enhance the flexibility and precision of data retrieval in complex scenarios.

This query extracts details about patients who have booked more than one appointment.

DATA VIZ

Now you know how this EHR database is organized and how information can be extracted from it. But what about **storytelling**?

Let’s use Tableau to see what was the most popular appointment type this month.



THANK YOU

CONTACT DETAILS

https://www.linkedin.com/in/mariagrabar/
https://github.com/mariagrabar