-- 02_schema_data_checks.sql
https://docs.google.com/document/d/1pEpGEPdWJP7scJ8JDHhnTzzvlUBhsDiazubFj9Vu0Eo/edit?usp=sharing

```
--    ======================================================================
--    MEDICAL APPOINTMENT & RECORDS SYSTEM - FULL SCHEMA + SAMPLE DATA
--    Run this while connected to database: medicaldb
-- Consists of 3 parts:
--   0. Optional DROP TABLE IF EXISTS
--   1. CREATE TABLE (schema)
--   2. INSERT sample data
--   3. SELECT queries to verify the data

-- ======================================================================
-- 0. OPTIONAL: drop existing tables if you want a clean rebuild
-- ======================================================================


-- DROP TABLE IF EXISTS audit_logs CASCADE;
-- DROP TABLE IF EXISTS billing CASCADE;
-- DROP TABLE IF EXISTS prescriptions CASCADE;
-- DROP TABLE IF EXISTS medical_records CASCADE;
-- DROP TABLE IF EXISTS appointments CASCADE;
-- DROP TABLE IF EXISTS pharmacists CASCADE;
-- DROP TABLE IF EXISTS doctors CASCADE;
-- DROP TABLE IF EXISTS patients CASCADE;
-- DROP TABLE IF EXISTS users CASCADE;

-- ======================
```

-- 1. SCHEMA (TABLES)

```
-- ======================

-- USERS: all login identities
CREATE TABLE users (
    user_id       SERIAL PRIMARY KEY,
    username      VARCHAR(50) UNIQUE NOT NULL,
    password_hash TEXT NOT NULL,
    role          VARCHAR(30) NOT NULL,  -- 'patient', 'doctor', 'pharmacist', 'admin'
    email         VARCHAR(100),
    phone         VARCHAR(30),
    created_at    TIMESTAMP NOT NULL DEFAULT NOW(),
    last_login_at TIMESTAMP
);

-- PATIENTS
CREATE TABLE patients (
    patient_id    SERIAL PRIMARY KEY,
```

```
    user_id       INTEGER UNIQUE REFERENCES users(user_id) ON DELETE CASCADE,
    full_name     VARCHAR(100) NOT NULL,
    dob           DATE,
    contact_info  TEXT,
    last_login_at TIMESTAMP
);

-- DOCTORS
CREATE TABLE doctors (
    doctor_id      SERIAL PRIMARY KEY,
    user_id        INTEGER UNIQUE REFERENCES users(user_id) ON DELETE CASCADE,
    specialty      VARCHAR(100),
    license_number VARCHAR(50),
    contact_info   TEXT,
    last_login_at  TIMESTAMP
);

-- PHARMACISTS
CREATE TABLE pharmacists (
    pharmacist_id SERIAL PRIMARY KEY,
    user_id       INTEGER UNIQUE REFERENCES users(user_id) ON DELETE CASCADE,
    can_dispense  BOOLEAN NOT NULL DEFAULT TRUE
);

-- APPOINTMENTS
CREATE TABLE appointments (
    appointment_id SERIAL PRIMARY KEY,
    patient_id     INTEGER NOT NULL REFERENCES patients(patient_id) ON DELETE
CASCADE,
    doctor_id      INTEGER NOT NULL REFERENCES doctors(doctor_id) ON DELETE
CASCADE,
    start_time     TIMESTAMP NOT NULL,
    end_time       TIMESTAMP,
    status         VARCHAR(20) NOT NULL DEFAULT 'scheduled', --
scheduled/cancelled/completed
    created_by     INTEGER REFERENCES users(user_id),
    created_at     TIMESTAMP NOT NULL DEFAULT NOW(),
    notes          TEXT
);

-- MEDICAL RECORDS
CREATE TABLE medical_records (
    record_id       SERIAL PRIMARY KEY,
    patient_id      INTEGER NOT NULL REFERENCES patients(patient_id) ON DELETE
CASCADE,
    doctor_id       INTEGER REFERENCES doctors(doctor_id) ON DELETE SET NULL,
    visit_date      DATE NOT NULL,
    diagnosis       TEXT,          -- will encrypt later
    treatment_notes TEXT,          -- will encrypt later
    created_at      TIMESTAMP NOT NULL DEFAULT NOW(),
```

```
    modified_at     TIMESTAMP,
    visibility      VARCHAR(20) NOT NULL DEFAULT 'normal'  -- normal/restricted
);

-- PRESCRIPTIONS
CREATE TABLE prescriptions (
    prescription_id SERIAL PRIMARY KEY,
    record_id       INTEGER NOT NULL REFERENCES medical_records(record_id) ON DELETE
CASCADE,
    patient_id      INTEGER NOT NULL REFERENCES patients(patient_id) ON DELETE
CASCADE,
    doctor_id       INTEGER REFERENCES doctors(doctor_id) ON DELETE SET NULL,
    pharmacist_id   INTEGER REFERENCES pharmacists(pharmacist_id) ON DELETE SET
NULL,
    drug_name       VARCHAR(200) NOT NULL,
    dosage          VARCHAR(100) NOT NULL,
    frequency       VARCHAR(100) NOT NULL,
    status          VARCHAR(20) NOT NULL DEFAULT 'issued', -- issued/dispensed
    created_at      TIMESTAMP NOT NULL DEFAULT NOW(),
    dispensed_at    TIMESTAMP
);

-- BILLING
CREATE TABLE billing (
    billing_id      SERIAL PRIMARY KEY,
    patient_id      INTEGER NOT NULL REFERENCES patients(patient_id) ON DELETE
CASCADE,
    appointment_id  INTEGER REFERENCES appointments(appointment_id) ON DELETE SET
NULL,
    amount          NUMERIC(10,2) NOT NULL,
    payment_status  VARCHAR(20) NOT NULL DEFAULT 'unpaid', -- unpaid/paid
    insurance_provider VARCHAR(100),
    insurance_claim_id VARCHAR(100),
    created_at      TIMESTAMP NOT NULL DEFAULT NOW()
);

-- AUDIT LOGS
CREATE TABLE audit_logs (
    log_id       SERIAL PRIMARY KEY,
    user_id      INTEGER REFERENCES users(user_id) ON DELETE SET NULL,
    action       VARCHAR(100) NOT NULL,
    table_name   VARCHAR(50) NOT NULL,
    record_id    INTEGER,
    timestamp    TIMESTAMP NOT NULL DEFAULT NOW(),
    details      TEXT
);

-- ================================================
```

-- 2. SAMPLE DATA

```
-- ================================================

-- 2.1 USERS (17 total: 10 patients, 5 doctors, 1 pharmacist, 1 admin)
INSERT INTO users (username, password_hash, role, email, phone)
VALUES
-- Admin
('admin1',        'hash_admin',   'admin',       'admin@clinic.com',           '+352271000000'),

-- Patients
('patient_john',  'hash_john',    'patient',     'john.miller@example.com',     '+352621000001'),
('patient_mary',  'hash_mary',    'patient',     'mary.johnson@example.com',    '+352621000002'),
('patient_luc',   'hash_luc',     'patient',     'luc.schmit@example.com',      '+352621000101'),
('patient_sophie','hash_sophie',  'patient',     'sophie.klein@example.com',    '+352621000102'),
('patient_marc',  'hash_marc',    'patient',     'marc.hoffmann@example.com',   '+352621000103'),
('patient_claire','hash_claire',  'patient',     'claire.meyer@example.com',    '+352621000104'),
('patient_anna',  'hash_anna_p',  'patient',     'anna.wagner@example.com',
'+352621000105'),
('patient_tom',   'hash_tom',     'patient',     'tom.becker@example.com',      '+352621000106'),
('patient_nina',  'hash_nina',    'patient',     'nina.hansen@example.com',     '+352621000107'),
('patient_paul',  'hash_paul',    'patient',     'paul.frank@example.com',      '+352621000108'),

-- Doctors
('doctor_smith',  'hash_smith',   'doctor',      'smith@clinic.com',           '+352271000201'),
('doctor_brown',  'hash_brown',   'doctor',      'brown@clinic.com',           '+352271000202'),
('doctor_muller', 'hash_muller',  'doctor',      'muller@clinic.com',          '+352271000203'),
('doctor_fischer','hash_fischer', 'doctor',      'fischer@clinic.com',         '+352271000204'),
('doctor_schmit', 'hash_schmit',  'doctor',      'schmit@clinic.com',          '+352271000205'),

-- Pharmacist
('pharm_anna',    'hash_anna',    'pharmacist',  'anna@pharmacy.com',          '+352271000301');

-- 2.2 PATIENTS (10)
INSERT INTO patients (user_id, full_name, dob, contact_info)
VALUES
(
 (SELECT user_id FROM users WHERE username='patient_john'),
 'John Miller',
 '1990-04-12',
 '21 Rue de Hollerich, L-1741 Luxembourg City'
),
(
 (SELECT user_id FROM users WHERE username='patient_mary'),
 'Mary Johnson',
 '1985-09-23',
 '4 Rue du Fort Wallis, Gare, L-2714 Luxembourg City'
),
(
 (SELECT user_id FROM users WHERE username='patient_luc'),
```

```
 'Luc Schmit',
 '1992-03-05',
 '15 Rue de Hollerich, L-1741 Luxembourg City'
),
(
 (SELECT user_id FROM users WHERE username='patient_sophie'),
 'Sophie Klein',
 '1988-11-19',
 '8 Rue du Fort Wallis, L-2714 Luxembourg City'
),
(
 (SELECT user_id FROM users WHERE username='patient_marc'),
 'Marc Hoffmann',
 '1979-06-27',
 '24 Avenue de la Liberté, L-1930 Luxembourg City'
),
(
 (SELECT user_id FROM users WHERE username='patient_claire'),
 'Claire Meyer',
 '1995-02-14',
 '3 Rue des Roses, Belair, L-2412 Luxembourg City'
),
(
 (SELECT user_id FROM users WHERE username='patient_anna'),
 'Anna Wagner',
 '1983-09-03',
 '10 Rue du Brill, L-4041 Esch-sur-Alzette'
),
(
 (SELECT user_id FROM users WHERE username='patient_tom'),
 'Tom Becker',
 '1991-07-30',
 '6 Rue de la Poste, L-4501 Differdange'
),
(
 (SELECT user_id FROM users WHERE username='patient_nina'),
 'Nina Hansen',
 '2000-01-09',
 '12 Grand-Rue, L-9220 Diekirch'
),
(
 (SELECT user_id FROM users WHERE username='patient_paul'),
 'Paul Frank',
 '1986-05-21',
 '5 Rue du Pont, L-5551 Remich'
);

-- 2.3 DOCTORS (5)
INSERT INTO doctors (user_id, specialty, license_number, contact_info)
VALUES
```

```
(
 (SELECT user_id FROM users WHERE username='doctor_smith'),
 'Cardiology',
 'DOC-LUX-1001',
 'Cardio Clinic, 10 Rue de Kirchberg, L-1858 Luxembourg City'
),
(
 (SELECT user_id FROM users WHERE username='doctor_brown'),
 'General Practice',
 'DOC-LUX-1002',
 'Family Practice, 5 Rue de la Gare, L-1611 Luxembourg City'
),
(
 (SELECT user_id FROM users WHERE username='doctor_muller'),
 'Pediatrics',
 'DOC-LUX-3001',
 'Pediatric Clinic, 20 Rue de Kirchberg, L-1858 Luxembourg City'
),
(
 (SELECT user_id FROM users WHERE username='doctor_fischer'),
 'Dermatology',
 'DOC-LUX-3002',
 'Dermatology Center, 7 Rue du Nord, L-2229 Luxembourg City'
),
(
 (SELECT user_id FROM users WHERE username='doctor_schmit'),
 'Neurology',
 'DOC-LUX-3003',
 'Neuro Clinic, 4 Rue du Canal, L-4050 Esch-sur-Alzette'
);

-- 2.4 PHARMACISTS (1)
INSERT INTO pharmacists (user_id, can_dispense)
VALUES
(
 (SELECT user_id FROM users WHERE username='pharm_anna'),
 TRUE
);

-- 2.5 APPOINTMENTS (10 total)
INSERT INTO appointments (patient_id, doctor_id, start_time, end_time, status, created_by,
notes)
VALUES
-- 1) John with Dr. Smith (completed)
(
 (SELECT patient_id FROM patients WHERE full_name='John Miller'),
 (SELECT d.doctor_id FROM doctors d JOIN users u ON d.user_id=u.user_id WHERE
u.username='doctor_smith'),
 '2025-01-12 10:00', '2025-01-12 10:30', 'completed',
 (SELECT user_id FROM users WHERE username='patient_john'),
```

```
 'Routine cardiac check-up'
),

-- 2) Mary with Dr. Brown (scheduled)
(
 (SELECT patient_id FROM patients WHERE full_name='Mary Johnson'),
 (SELECT d.doctor_id FROM doctors d JOIN users u ON d.user_id=u.user_id WHERE
u.username='doctor_brown'),
 '2025-01-15 14:00', NULL, 'scheduled',
 (SELECT user_id FROM users WHERE username='admin1'),
 'First-time GP appointment'
),

-- 3) Luc with Dr. Brown
(
 (SELECT patient_id FROM patients WHERE full_name='Luc Schmit'),
 (SELECT d.doctor_id FROM doctors d JOIN users u ON d.user_id=u.user_id WHERE
u.username='doctor_brown'),
 '2025-01-20 09:00', '2025-01-20 09:20', 'completed',
 (SELECT user_id FROM users WHERE username='patient_luc'),
 'Follow-up on blood pressure'
),

-- 4) Sophie with Dr. Muller
(
 (SELECT patient_id FROM patients WHERE full_name='Sophie Klein'),
 (SELECT d.doctor_id FROM doctors d JOIN users u ON d.user_id=u.user_id WHERE
u.username='doctor_muller'),
 '2025-01-21 11:00', '2025-01-21 11:30', 'completed',
 (SELECT user_id FROM users WHERE username='patient_sophie'),
 'Pediatric consultation for child, but record kept for parent'
),

-- 5) Marc with Dr. Fischer
(
 (SELECT patient_id FROM patients WHERE full_name='Marc Hoffmann'),
 (SELECT d.doctor_id FROM doctors d JOIN users u ON d.user_id=u.user_id WHERE
u.username='doctor_fischer'),
 '2025-01-22 16:00', '2025-01-22 16:20', 'completed',
 (SELECT user_id FROM users WHERE username='admin1'),
 'Skin rash examination'
),

-- 6) Claire with Dr. Schmit
(
 (SELECT patient_id FROM patients WHERE full_name='Claire Meyer'),
 (SELECT d.doctor_id FROM doctors d JOIN users u ON d.user_id=u.user_id WHERE
u.username='doctor_schmit'),
 '2025-01-23 13:30', NULL, 'scheduled',
 (SELECT user_id FROM users WHERE username='patient_claire'),
```

```
 'Neurology consultation scheduled'
),

-- 7) Anna with Dr. Brown
(
 (SELECT patient_id FROM patients WHERE full_name='Anna Wagner'),
 (SELECT d.doctor_id FROM doctors d JOIN users u ON d.user_id=u.user_id WHERE
u.username='doctor_brown'),
 '2025-01-24 10:15', '2025-01-24 10:40', 'completed',
 (SELECT user_id FROM users WHERE username='patient_anna'),
 'Flu-like symptoms'
),

-- 8) Tom with Dr. Smith
(
 (SELECT patient_id FROM patients WHERE full_name='Tom Becker'),
 (SELECT d.doctor_id FROM doctors d JOIN users u ON d.user_id=u.user_id WHERE
u.username='doctor_smith'),
 '2025-01-25 15:00', NULL, 'scheduled',
 (SELECT user_id FROM users WHERE username='admin1'),
 'Check-up before starting sports program'
),

-- 9) Nina with Dr. Muller
(
 (SELECT patient_id FROM patients WHERE full_name='Nina Hansen'),
 (SELECT d.doctor_id FROM doctors d JOIN users u ON d.user_id=u.user_id WHERE
u.username='doctor_muller'),
 '2025-01-26 09:45', '2025-01-26 10:05', 'completed',
 (SELECT user_id FROM users WHERE username='patient_nina'),
 'Routine pediatric examination'
),

-- 10) Paul with Dr. Fischer
(
 (SELECT patient_id FROM patients WHERE full_name='Paul Frank'),
 (SELECT d.doctor_id FROM doctors d JOIN users u ON d.user_id=u.user_id WHERE
u.username='doctor_fischer'),
 '2025-01-27 08:30', NULL, 'scheduled',
 (SELECT user_id FROM users WHERE username='patient_paul'),
 'Follow-up on dermatitis treatment'
);

-- 2.6 MEDICAL RECORDS (one for each completed appointment)
INSERT INTO medical_records (patient_id, doctor_id, visit_date, diagnosis, treatment_notes)
VALUES
-- John
(
 (SELECT patient_id FROM patients WHERE full_name='John Miller'),
```

```
 (SELECT d.doctor_id FROM doctors d JOIN users u ON d.user_id=u.user_id WHERE
u.username='doctor_smith'),
 '2025-01-12',
 'Mild hypertension',
 'Recommended diet changes, exercise, and home blood pressure monitoring.'
),

-- Luc
(
 (SELECT patient_id FROM patients WHERE full_name='Luc Schmit'),
 (SELECT d.doctor_id FROM doctors d JOIN users u ON d.user_id=u.user_id WHERE
u.username='doctor_brown'),
 '2025-01-20',
 'Borderline high blood pressure',
 'Advised periodic checks; no medication yet.'
),

-- Sophie
(
 (SELECT patient_id FROM patients WHERE full_name='Sophie Klein'),
 (SELECT d.doctor_id FROM doctors d JOIN users u ON d.user_id=u.user_id WHERE
u.username='doctor_muller'),
 '2025-01-21',
 'Migraine episodes',
 'Prescribed mild pain relief and recommended sleep hygiene.'
),

-- Marc
(
 (SELECT patient_id FROM patients WHERE full_name='Marc Hoffmann'),
 (SELECT d.doctor_id FROM doctors d JOIN users u ON d.user_id=u.user_id WHERE
u.username='doctor_fischer'),
 '2025-01-22',
 'Contact dermatitis',
 'Topical corticosteroid cream, avoid irritant detergents.'
),

-- Anna
(
 (SELECT patient_id FROM patients WHERE full_name='Anna Wagner'),
 (SELECT d.doctor_id FROM doctors d JOIN users u ON d.user_id=u.user_id WHERE
u.username='doctor_brown'),
 '2025-01-24',
 'Seasonal influenza',
 'Rest, fluids, antipyretic medication as needed.'
),

-- Nina
(
 (SELECT patient_id FROM patients WHERE full_name='Nina Hansen'),
```

```
 (SELECT d.doctor_id FROM doctors d JOIN users u ON d.user_id=u.user_id WHERE
u.username='doctor_muller'),
 '2025-01-26',
 'Healthy child exam',
 'No issues; routine follow-up in one year.'
),

-- Mary (pre-filled for her scheduled visit)
(
 (SELECT patient_id FROM patients WHERE full_name='Mary Johnson'),
 (SELECT d.doctor_id FROM doctors d JOIN users u ON d.user_id=u.user_id WHERE
u.username='doctor_brown'),
 '2025-01-15',
 'Pending evaluation',
 'Appointment scheduled; notes to be updated after visit.'
),

-- Claire
(
 (SELECT patient_id FROM patients WHERE full_name='Claire Meyer'),
 (SELECT d.doctor_id FROM doctors d JOIN users u ON d.user_id=u.user_id WHERE
u.username='doctor_schmit'),
 '2025-01-23',
 'Pending neurology consultation',
 'Symptoms: occasional dizziness; further tests planned.'
),

-- Tom
(
 (SELECT patient_id FROM patients WHERE full_name='Tom Becker'),
 (SELECT d.doctor_id FROM doctors d JOIN users u ON d.user_id=u.user_id WHERE
u.username='doctor_smith'),
 '2025-01-25',
 'Pre-sport assessment',
 'Baseline check; waiting for lab results.'
),

-- Paul
(
 (SELECT patient_id FROM patients WHERE full_name='Paul Frank'),
 (SELECT d.doctor_id FROM doctors d JOIN users u ON d.user_id=u.user_id WHERE
u.username='doctor_fischer'),
 '2025-01-27',
 'Dermatitis follow-up',
 'Improvement noted; continue treatment for two more weeks.'
);

-- 2.7 PRESCRIPTIONS (5 examples)
INSERT INTO prescriptions (record_id, patient_id, doctor_id, pharmacist_id, drug_name,
dosage, frequency, status)
```

```
VALUES
-- John: hypertension
(
 (SELECT record_id FROM medical_records mr JOIN patients p ON mr.patient_id=p.patient_id
  WHERE p.full_name='John Miller' LIMIT 1),
 (SELECT patient_id FROM patients WHERE full_name='John Miller'),
 (SELECT d.doctor_id FROM doctors d JOIN users u ON d.user_id=u.user_id WHERE
u.username='doctor_smith'),
 (SELECT pharmacist_id FROM pharmacists LIMIT 1),
 'Amlodipine',
 '5 mg',
 'Once daily',
 'issued'
),

-- Marc: dermatitis cream
(
 (SELECT record_id FROM medical_records mr JOIN patients p ON mr.patient_id=p.patient_id
  WHERE p.full_name='Marc Hoffmann' LIMIT 1),
 (SELECT patient_id FROM patients WHERE full_name='Marc Hoffmann'),
 (SELECT d.doctor_id FROM doctors d JOIN users u ON d.user_id=u.user_id WHERE
u.username='doctor_fischer'),
 (SELECT pharmacist_id FROM pharmacists LIMIT 1),
 'Hydrocortisone cream',
 'Apply thin layer',
 'Twice daily for 7 days',
 'issued'
),

-- Anna: flu
(
 (SELECT record_id FROM medical_records mr JOIN patients p ON mr.patient_id=p.patient_id
  WHERE p.full_name='Anna Wagner' LIMIT 1),
 (SELECT patient_id FROM patients WHERE full_name='Anna Wagner'),
 (SELECT d.doctor_id FROM doctors d JOIN users u ON d.user_id=u.user_id WHERE
u.username='doctor_brown'),
 (SELECT pharmacist_id FROM pharmacists LIMIT 1),
 'Paracetamol',
 '500 mg',
 'Every 6 hours as needed',
 'dispensed'
),

-- Luc: BP monitoring (no drug yet, but demo prescription)
(
 (SELECT record_id FROM medical_records mr JOIN patients p ON mr.patient_id=p.patient_id
  WHERE p.full_name='Luc Schmit' LIMIT 1),
 (SELECT patient_id FROM patients WHERE full_name='Luc Schmit'),
 (SELECT d.doctor_id FROM doctors d JOIN users u ON d.user_id=u.user_id WHERE
u.username='doctor_brown'),
```

```
 (SELECT pharmacist_id FROM pharmacists LIMIT 1),
 'Home blood pressure monitor',
 'Device',
 'Use daily in the morning',
 'issued'
),

-- Nina: vitamins
(
 (SELECT record_id FROM medical_records mr JOIN patients p ON mr.patient_id=p.patient_id
  WHERE p.full_name='Nina Hansen' LIMIT 1),
 (SELECT patient_id FROM patients WHERE full_name='Nina Hansen'),
 (SELECT d.doctor_id FROM doctors d JOIN users u ON d.user_id=u.user_id WHERE
u.username='doctor_muller'),
 (SELECT pharmacist_id FROM pharmacists LIMIT 1),
 'Vitamin D drops',
 '400 IU',
 'Once daily',
 'issued'
);

-- 2.8 BILLING (one per appointment)
INSERT INTO billing (patient_id, appointment_id, amount, payment_status, insurance_provider,
insurance_claim_id)
VALUES
(
 (SELECT patient_id FROM patients WHERE full_name='John Miller'),
 (SELECT appointment_id FROM appointments a JOIN patients p ON a.patient_id=p.patient_id
  WHERE p.full_name='John Miller' LIMIT 1),
 120.00,
 'paid',
 'Allianz Health',
 'CLM-JN-20250112'
),
(
 (SELECT patient_id FROM patients WHERE full_name='Mary Johnson'),
 (SELECT appointment_id FROM appointments a JOIN patients p ON a.patient_id=p.patient_id
  WHERE p.full_name='Mary Johnson' LIMIT 1),
 80.00,
 'unpaid',
 'AXA Health',
 'CLM-MR-20250115'
),
(
 (SELECT patient_id FROM patients WHERE full_name='Luc Schmit'),
 (SELECT appointment_id FROM appointments a JOIN patients p ON a.patient_id=p.patient_id
  WHERE p.full_name='Luc Schmit' LIMIT 1),
 90.00,
 'paid',
 'CNS',
```

```
 'CLM-LS-20250120'
),
(
 (SELECT patient_id FROM patients WHERE full_name='Sophie Klein'),
 (SELECT appointment_id FROM appointments a JOIN patients p ON a.patient_id=p.patient_id
  WHERE p.full_name='Sophie Klein' LIMIT 1),
 70.00,
 'paid',
 'CNS',
 'CLM-SK-20250121'
),
(
 (SELECT patient_id FROM patients WHERE full_name='Marc Hoffmann'),
 (SELECT appointment_id FROM appointments a JOIN patients p ON a.patient_id=p.patient_id
  WHERE p.full_name='Marc Hoffmann' LIMIT 1),
 95.00,
 'unpaid',
 'Foyer Santé',
 'CLM-MH-20250122'
),
(
 (SELECT patient_id FROM patients WHERE full_name='Claire Meyer'),
 (SELECT appointment_id FROM appointments a JOIN patients p ON a.patient_id=p.patient_id
  WHERE p.full_name='Claire Meyer' LIMIT 1),
 130.00,
 'unpaid',
 'Allianz Health',
 'CLM-CM-20250123'
),
(
 (SELECT patient_id FROM patients WHERE full_name='Anna Wagner'),
 (SELECT appointment_id FROM appointments a JOIN patients p ON a.patient_id=p.patient_id
  WHERE p.full_name='Anna Wagner' LIMIT 1),
 60.00,
 'paid',
 'CNS',
 'CLM-AW-20250124'
),
(
 (SELECT patient_id FROM patients WHERE full_name='Tom Becker'),
 (SELECT appointment_id FROM appointments a JOIN patients p ON a.patient_id=p.patient_id
  WHERE p.full_name='Tom Becker' LIMIT 1),
 110.00,
 'unpaid',
 'Foyer Santé',
 'CLM-TB-20250125'
),
(
 (SELECT patient_id FROM patients WHERE full_name='Nina Hansen'),
 (SELECT appointment_id FROM appointments a JOIN patients p ON a.patient_id=p.patient_id
```

```
 WHERE p.full_name='Nina Hansen' LIMIT 1),
 65.00,
 'paid',
 'CNS',
 'CLM-NH-20250126'
),
(
 (SELECT patient_id FROM patients WHERE full_name='Paul Frank'),
 (SELECT appointment_id FROM appointments a JOIN patients p ON a.patient_id=p.patient_id
  WHERE p.full_name='Paul Frank' LIMIT 1),
 85.00,
 'unpaid',
 'AXA Health',
 'CLM-PF-20250127'
);

-- 2.9 AUDIT LOGS (small demo)
INSERT INTO audit_logs (user_id, action, table_name, record_id, details)
VALUES
(
 (SELECT user_id FROM users WHERE username='admin1'),
 'CREATE_PATIENT',
 'patients',
 (SELECT patient_id FROM patients WHERE full_name='John Miller'),
 'Admin registered patient John Miller.'
),
(
 (SELECT user_id FROM users WHERE username='doctor_smith'),
 'UPDATE_MEDICAL_RECORD',
 'medical_records',
 (SELECT record_id FROM medical_records mr JOIN patients p ON mr.patient_id=p.patient_id
  WHERE p.full_name='John Miller' LIMIT 1),
 'Doctor Smith updated diagnosis and treatment notes.'
),
(
 (SELECT user_id FROM users WHERE username='pharm_anna'),
 'DISPENSE_PRESCRIPTION',
 'prescriptions',
 (SELECT prescription_id FROM prescriptions pr JOIN patients p ON pr.patient_id=p.patient_id
  WHERE p.full_name='Anna Wagner' LIMIT 1),
 'Pharmacist Anna dispensed medication for Anna Wagner.'
);
```

-- + Add the “System Bot” actor:

```
INSERT INTO users (username, password_hash, role, email, phone)
VALUES ('system_bot', 'hash_system', 'system', NULL, NULL);
```

-- + Add 1 example audit log entries generated by the bot

```
INSERT INTO audit_logs (user_id, action, table_name, record_id, details)
VALUES (
  (SELECT user_id FROM users WHERE username='system_bot'),
  'SEND_APPOINTMENT_REMINDER',
  'appointments',
  (SELECT appointment_id FROM appointments ORDER BY start_time LIMIT 1),
  'Reminder sent to patient for upcoming appointment.'
);
```

-- + 1 for billing:

```
INSERT INTO audit_logs (user_id, action, table_name, record_id, details)
VALUES (
  (SELECT user_id FROM users WHERE username='system_bot'),
  'FLAG_OVERDUE_BILL',
  'billing',
  (SELECT billing_id FROM billing WHERE payment_status='unpaid' LIMIT 1),
  'Overdue bill flagged for follow-up notification.'
);
```

-- That gives us concrete evidence that the bot exists and writes logs.

-- ==========

-- 3. Checking (SELECT queries for checking every part of our database)

```
-- They verify:
-- That sample data loaded correctly;
-- Relationships are consistent;
-- Foreign keys point to the right rows;
-- Parts of the DB work before we add Access Control, RLS, Encryption.
-- I group them by table and purpose.
-- ==========

-- 1. Verify total counts (can run all together)

SELECT 'users' AS table, COUNT(*) FROM users
UNION ALL SELECT 'patients', COUNT(*) FROM patients
UNION ALL SELECT 'doctors', COUNT(*) FROM doctors
UNION ALL SELECT 'pharmacists', COUNT(*) FROM pharmacists
UNION ALL SELECT 'appointments', COUNT(*) FROM appointments
UNION ALL SELECT 'medical_records', COUNT(*) FROM medical_records
```

```
UNION ALL SELECT 'prescriptions', COUNT(*) FROM prescriptions
UNION ALL SELECT 'billing', COUNT(*) FROM billing
UNION ALL SELECT 'audit_logs', COUNT(*) FROM audit_logs;

--You should see:
```

table	count
users	18
patients	10
doctors	5
pharmacists	1
appointments	10
medical_records	10
prescriptions	5
billing	10
audit_logs	5

```
-- 2. List all users with roles

SELECT user_id, username, role, email, phone
FROM users
ORDER BY role, username;

-- 3. Show all patients with their linked user account

SELECT p.patient_id, u.username, p.full_name, p.dob, p.contact_info
FROM patients p
JOIN users u ON u.user_id = p.user_id
ORDER BY p.full_name;

-- 4. Show all doctors with specialties

SELECT d.doctor_id, u.username, d.specialty, d.license_number, d.contact_info
FROM doctors d
JOIN users u ON u.user_id = d.user_id
ORDER BY d.specialty, d.doctor_id;

-- 5. See all appointments (joined for readability)

SELECT
    a.appointment_id,
```

```
    p.full_name AS patient,
    d.specialty AS doctor_specialty,
    a.start_time,
    a.end_time,
    a.status,
    a.notes
FROM appointments a
JOIN patients p ON p.patient_id = a.patient_id
JOIN doctors d ON d.doctor_id = a.doctor_id
ORDER BY a.start_time;

-- This is a perfect sanity check:
-- You immediately see if dates, names, doctors match.


-- 6. See appointments by doctor

-- Example: appointments for Dr. Brown:

SELECT
    a.appointment_id,
    p.full_name AS patient,
    a.start_time,
    a.status
FROM appointments a
JOIN patients p ON p.patient_id = a.patient_id
JOIN doctors d ON d.doctor_id = a.doctor_id
WHERE d.doctor_id = (
    SELECT doctor_id FROM doctors d
    JOIN users u ON d.user_id=u.user_id
    WHERE u.username='doctor_brown'
)
ORDER BY a.start_time;

-- 7. See medical records with doctor + patient names

SELECT
    mr.record_id,
    p.full_name AS patient,
    d.specialty AS doctor,
    mr.visit_date,
    mr.diagnosis,
    mr.treatment_notes
FROM medical_records mr
JOIN patients p ON p.patient_id = mr.patient_id
LEFT JOIN doctors d ON d.doctor_id = mr.doctor_id
ORDER BY mr.visit_date;

-- 8. List prescriptions (joined)
```

```
SELECT
    pr.prescription_id,
    pa.full_name AS patient,
    d.specialty AS doctor,
    pr.drug_name,
    pr.dosage,
    pr.frequency,
    pr.status
FROM prescriptions pr
JOIN patients pa ON pa.patient_id = pr.patient_id
LEFT JOIN doctors d ON d.doctor_id = pr.doctor_id
ORDER BY pr.prescription_id;

-- 9. Billing overview

SELECT
    b.billing_id,
    p.full_name AS patient,
    b.amount,
    b.payment_status,
    b.insurance_provider,
    b.insurance_claim_id
FROM billing b
JOIN patients p ON p.patient_id = b.patient_id
ORDER BY b.billing_id;

-- 10. Audit log with usernames

SELECT
    al.log_id,
    u.username AS actor,
    al.action,
    al.table_name,
    al.record_id,
    al.timestamp,
    al.details
FROM audit_logs al
LEFT JOIN users u ON u.user_id = al.user_id
ORDER BY al.timestamp DESC;



-- =====
-- Checks for relationships (very important before Access Control & RLS)

-- 1. Patients with number of appointments:

SELECT p.full_name, COUNT(a.appointment_id) AS appointment_count
FROM patients p
LEFT JOIN appointments a ON a.patient_id = p.patient_id
GROUP BY p.full_name
```

```
ORDER BY appointment_count DESC;

-- 2 Doctors with how many patients they see:

SELECT d.specialty, COUNT(a.appointment_id) AS num_appointments
FROM doctors d
LEFT JOIN appointments a ON a.doctor_id = d.doctor_id
GROUP BY d.specialty;

-- 3 Patients who have prescriptions:

SELECT p.full_name, COUNT(pr.prescription_id) AS prescriptions
FROM patients p
LEFT JOIN prescriptions pr ON pr.patient_id = p.patient_id
GROUP BY p.full_name;
```

-- 3 Audit_logs table. This joins usernames, when the System Bot, admin, doctor,... acted:

```
SELECT
    al.log_id,
    u.username AS actor,
    u.role,
    al.action,
    al.table_name,
    al.record_id,
    al.timestamp,
    al.details
FROM audit_logs al
LEFT JOIN users u ON u.user_id = al.user_id
ORDER BY al.timestamp DESC;
```