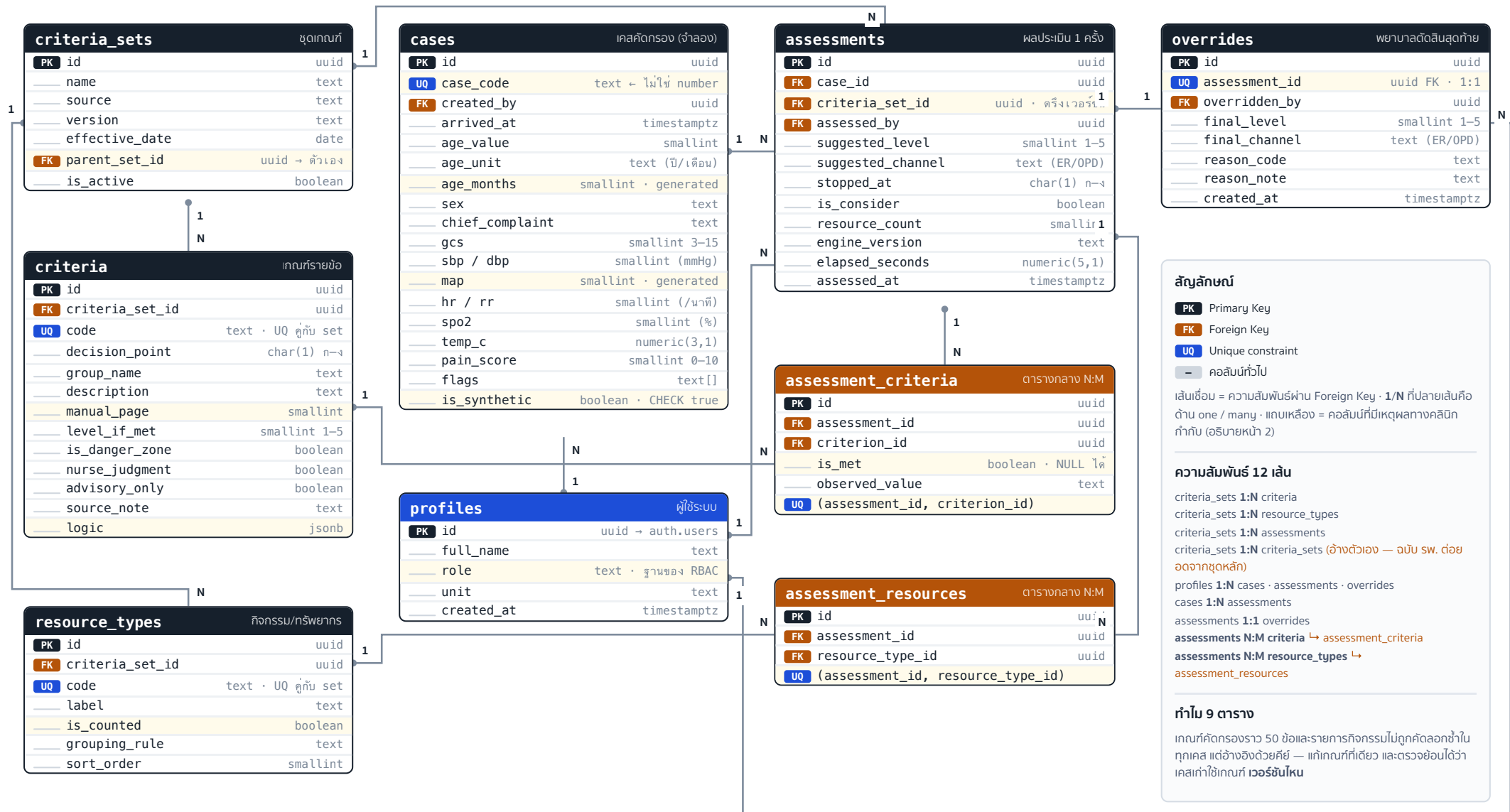




ระบบทำอะไร — และทำไม schema ต้องหน้าตาแบบนี้

- พยาบาลคัดกรองหน้าห้องฉุกเฉินต้องจัดระดับความเร่งด่วน 1–5 ภายในไม่กี่นาที โดยเทียบเกณฑ์คู่มือ **MOPH ED Triage (กรมการแพทย์ 2561)** ราว 50 ข้อจากความจำ และระดับที่ให้กำหนดช่องทาง ER หรือ OPD โดยอัตโนมัติ เคสทำทั้งที่รอยต่อระดับ 2/3 จึงมักถูก “ดันขึ้น” เพื่อความปลอดภัย ระบบเสนอระดับ + ช่องทาง พร้อม**แสดงว่าเข้าเกณฑ์ข้อไหนเป็นรายข้อ** โดยพยาบาลตัดสินใจท้ายเสมอ — ข้อกำหนดนี้บังคับรูปร่างของ schema 5 ข้อ:
- เกณฑ์ต้องเป็นข้อมูล ไม่ใช่โค้ด → แยก `criteria_sets` / `criteria` / `resource_types` ออกมา และเก็บเงื่อนไขของแต่ละข้อเป็น `logic jsonb` — โปรแกรมอ่านเกณฑ์มาทำงาน ไม่ได้ฝังเกณฑ์ไว้ในตัวเอง
 - ต้องตรวจสอบได้ว่าข้อเสนอมาจากเกณฑ์ข้อใด → เก็บผลการเทียบ “รายข้อ” ของทุกการประเมิน = `assessment_criteria` (N:M)
 - การนับทรัพยากรมีกติกาของมันเอง (คู่มือหน้า 13–14 แยก “กิจกรรมที่นับ / ไม่นับ”) → `resource_types.is_counted` + `assessment_resources` (N:M) นับที่กว่าเคสนับเลือกกิจกรรมใดไว้จริง
 - ต้องแยก “สิ่งที่ระบบเสนอ” จาก “สิ่งที่พยาบาลตัดสินใจจริง” → `assessments` แยกจาก `overrides` เพื่อวัด over-/under-triage
 - โรงพยาบาลปรับเกณฑ์เองได้ (คู่มือหน้า 12 เขียนไว้เอง) → `criteria_sets.parent_set_id` อ้างกลับไปยังชุดที่ด้อยออกมา

9 Entities และเหตุผลที่ต้องมี

ตาราง	เก็บอะไร / ทำไมต้องแยก
profiles	ผู้ใช้ + role (nurse / doctor / head_nurse / admin) ผูกกับ <code>auth.users</code> ของ Supabase — ฐานของ RBAC และเป็นตัวแปรใน RLS policy
criteria_sets	ชุดเกณฑ์ 1 เวอร์ชัน · parent_set_id อ้างตัวเอง ทำให้ “ฉบับโรงพยาบาล” ชี้กลับไปที่ฉบับคู่มือที่ด้อยออกมาได้
criteria	เกณฑ์รายข้อ ~50 ข้อ + จุดตัดสินใจ n-ง + เลขหน้าคู่มือ + <code>logic jsonb</code> · ถ้าไม่แยก ข้อความเกณฑ์จะถูกคัดลอกซ้ำในทุกเคส
resource_types	รายการกิจกรรมของแต่ละชุดเกณฑ์ พร้อม <code>is_counted</code> และ <code>grouping_rule</code> (เช่น “CXR + CT = 2 กิจกรรม”)
cases	เคส 1 ราย: อาการนำ สัญญาณชีพ GCS pain score — สถานการณ์จำลองล้วน ไม่มีชื่อ / HN / เลขบัตร
assessments	การประเมิน 1 ครั้ง + stopped_at (หยุดที่จุดไหน) — 1 เคสประเมินซ้ำได้ จึงเป็น 1:N ไม่ใช่ 1:1
assessment_criteria	N:M assessments × criteria + ผลเทียบรายข้อ = หลักฐานว่า “ข้อเสนอนี้มาจากเกณฑ์ข้อไหน”
assessment_resources	N:M assessments × resource_types = กิจกรรมที่คาดว่าจะใช้ในเคสนั้น (ใช้คำนวณ <code>resource_count</code>)
overrides	ระดับ/ช่องทางที่พยาบาลตัดสินใจ + เหตุผล — 1:1 กับ assessments บังคับด้วย UNIQUE

ความสัมพันธ์และคีย์

ความสัมพันธ์	แบบ	คีย์ที่ใช้เชื่อม
criteria_sets → criteria	1:N	criteria.criteria_set_id
criteria_sets → resource_types	1:N	resource_types.criteria_set_id
criteria_sets → assessments	1:N	assessments.criteria_set_id
criteria_sets → criteria_sets	1:N	parent_set_id — FK อ้างตารางตัวเอง
profiles → cases	1:N	cases.created_by → profiles.id
profiles → assessments	1:N	assessments.assessed_by
profiles → overrides	1:N	overrides.overrideen_by
cases → assessments	1:N	assessments.case_id
assessments → overrides	1:1	overrides.assessment_id + UNIQUE
assessments ↔ criteria	N:M	ผ่าน assessment_criteria (UNIQUE คู่)
assessments ↔ resource_types	N:M	ผ่าน assessment_resources (UNIQUE คู่)

Primary Key ใช้ `uuid` ทุกตาราง (`gen_random_uuid()`) แทนเลขไล่ลำดับ เพราะกันการเดารหัสเคสจาก URL และรวมข้อมูลหลายโรงพยาบาลได้โดยไม่ชนกัน · **Foreign Key** **ซ้ำได้** จึงอยู่ฝั่ง N ของทุกความสัมพันธ์เสมอ · ตารางกลางทั้งสองใส่ **UNIQUE คู่คีย์** เพื่อกันบันทึกซ้ำของคู่เดิม

เหตุผลของชนิดข้อมูล (Data Type)

ฟิลด์	ชนิด	ทำไม
case_code	text	รหัสประจำตัว (nominal scale) ไม่ใช่ปริมาณ — เก็บเป็นตัวเลขจะเสียเลข 0 นำหน้า (หลักเดียวกับ HN)
age_value + age_unit	smallint + text	การบอกอายุเป็นเดือน ผู้ใหญ่เป็นปี — เก็บค่าและหน่วยตามที่รอกองจริง
age_months	smallint generated	ตาราง danger zone แบ่ง 4 ช่วงอายุเป็นเดือน — ให้อ่านข้อมูลคำนวณเอง จะไม่มีทางไม่ตรงกับ 2 คอลัมน์ข้างบน
map	smallint generated	(SBP + 2×DBP) + 3 — คู่มือใช้ MAP < 60 เป็นเกณฑ์ shock ที่จุด n
gcs, sbp, dbp, hr, rr, spo2, pain_score, level	smallint	จำนวนเต็มช่วงแคบ + ใส่ CHECK ของเขตสรีรวิทยา (GCS 3–15, SpO ₂ 50–100, pain 0–10)
temp_c	numeric(3,1)	ทศนิยม 1 ตำแหน่ง — ใช้ numeric ไม่ใช่ float กันค่าเพี้ยนจากการปัดเลขฐานสอง
arrived_at, assessed_at / effective_date	timestampz / date	ต้องเรียงลำดับ/คำนวณได้ + ผูก time zone — ห้ามเป็น text · วันที่เกณฑ์เริ่มใช้ไม่ต้องการเวลา จึงเป็น date
is_met	boolean NULL ได้	สามสถานะ : true = เข้าเกณฑ์ · false = ไม่เข้า · NULL = คัดสินไม่ได้เพราะยังไม่ได้วัด — ถ้าบังคับ NOT NULL “ยังไม่ได้วัด” จะถูกสืบเป็น “ไม่เข้าเกณฑ์” เขียน ๆ อันตรายทางคลินิก
logic	jsonb	เงื่อนไขของแต่ละข้อ (เช่น อายุ > 8 ปี และ ชีพจร > 100) โครงสร้างไม่เหมือนกัน จึงไม่เหมาะกับคอลัมน์ตายตัว
flags	text[]	ชุดธงที่พยาบาลตี๊ก จำนวนไม่แน่นอน ค้นด้วย GIN index

ความปลอดภัย — RLS + RBAC (เตรียมสำหรับ Part 2)

เปิด `alter table ... enable row level security` ทุกตาราง ไม่มีข้อยกเว้น แล้วกำหนด policy ตาม `profiles.role`

Role	อ่านเคส	ประเมิน / override	แก้ไขเกณฑ์
nurse (พยาบาลคัดกรอง)	เฉพาะเคสที่ตนสร้าง	✓	×
doctor	เฉพาะเคสในหน่วยตน	ดูอย่างเดียว	×
head_nurse (audit)	ทุกเคสในหน่วย	×	×
admin	×	×	✓ + จัดการสิทธิ์

RBAC นอกว่่านบทบาททำอะไรได้ · RLS นอกว่่าเห็นแถวไหนได้ — ใช้คู่กัน กฎอยู่ที่ฐานข้อมูลที่เดียว ทำงานทุก query แม้ frontend มีบั๊ก · ตัวอย่าง policy UU cases:

`create policy "nurse sees own cases" on cases for select using (created_by = auth.uid());`

PDPA: ตาราง `cases` ไม่มีคอลัมน์ระบุตัวบุคคลใด ๆ (ไม่มีชื่อ HN เลขบัตร วันเกิด) เก็บเฉพาะอายุ · `is_synthetic` default true และมี CHECK บังคับ true ตลอดเฟสของคอร์ส เพื่อกันข้อมูลผู้ปัจจัยจริงหลุดเข้ารระบบโดยไม่ตั้งใจ · ชุดเกณฑ์และรายการกิจกรรมเป็นข้อมูลจากคู่มือสาธารณะ: ไม่มีข้อมูลส่วนบุคคล