Entity Descriptions and Design Assumptions

- **User** (user_id)
  Attributes: name, email, time_zone, preferred_reminder_window
  Assumptions: one account per person; email unique for login; time zone used for scheduling windows.

- **NotificationPrefs** (user_id)
  Attributes: email_reminders, missed_dose_summary
  Assumptions: exactly one preference row per user (true 1:1 with User).

- **Drug** (drug_id)
  Attributes: rxnorm_code, name
  Assumptions: rxnorm_code (if it's present) is unique; name is unique in our catalog to prevent duplicates.

- **SideEffect** (drug_id,effect)
  Attributes: description (optional)
  Assumptions: multi valued attribute of a drug; one row per (drug,effect).

- **Prescription** (rx_id)
  Attributes: user_id, drug_id, frequency, qty_on_hand, refills, rx_text
  Assumptions: a prescription isn't uniquely determined by a user–drug pair; the same user can have multiple prescriptions for the same drug.

- **Reminder** (reminder_id)
  Attributes: user_id, rx_id, remind_time, override_frequency (nullable)
  Assumptions: reminders are scheduled events that are tied to a single prescription; overrides are exceptions per reminder.

- **EmergencyContact** (contact_id)
  Attributes: user_id, name, phone, email, trigger
  Assumptions: Each contact is uniquely identified for a user by their email address.

- **Disease** (disease_id)
  Attributes: name, wry_specialty, description
  Assumptions: we store one primary specialty per disease (enough for referrals).

- **DiseaseSymptom** (disease_id,symptom)
  Attributes: description (optional)
  Assumptions: one row per (disease, symptom label).

- **Doctor** (doctor_id)
  Attributes: name, specialty, address, phone
  Assumptions: exactly one specialty per doctor.

Relationships & Cardinalities (and why)

- **User 1 – 1 NotificationPrefs**
  Rationale: one preference set per user; prevents conflicting settings.

- **User 1 – * Prescription** and **Prescription * – 1 Drug**
  Rationale: a person can have many prescriptions; each prescription is for one drug.

- **Prescription 1 – * Reminder** (and **User 1 – * Reminder** for scoping)
  Rationale: multiple reminders per prescription; reminders are owned by the same user for authentication or scheduling.

- **User 1 – * EmergencyContact**
  Rationale: multiple contacts per user; each contact belongs to exactly one user.

- **Drug 1 – * SideEffect**
  Rationale: side effects are stored separately so they can be searched and maintained without duplicating data.

- **Disease 1 – * DiseaseSymptom**
  Rationale: symptoms are normalized into a separate table to make them searchable and support automated matching.

- **Referrals:** select **Doctor** where Doctor.specialty = Disease.primary_specialty.
  Rationale: Match by specialty instead of linking diseases to specific doctors; avoids implying a fixed Disease → Doctor mapping; no physical foreign key is stored

BCNF Justification (table by table)

- **User**: user_id → (name,email,time_zone,window); PK is a superkey.

- **NotificationPrefs**: user_id → (email_reminders,missed_dose_summary); PK = user_id (1:1).

- **Drug**: drug_id → (rxnorm_code,name); rxnorm_code and name are unique to avoid duplicates; no non-key determinant.

- **SideEffect**: key (drug_id,effect); any description depends on the key only.

- **Prescription**: key rx_id; we do not assume (user_id,drug_id) determines the row → no BCNF violation.

- **Reminder**: key reminder_id; override_frequency is independent of rx_id unless set, no duplication of prescription data.

- **EmergencyContact**: key contact_id; UNIQUE(user_id,email) prevents representing the same fact twice; no attribute depends on a non-key determinant.

- **Disease**: disease_id → (name,primary_specialty,description); PK is a superkey.

- **DiseaseSymptom**: key (disease_id,symptom); description depends on that key; BCNF.

- **Doctor**: doctor_id → (name,specialty,address,phone); PK is a superkey.

Relational Schema (logical design, not SQL)

User(user_id:UUID [PK], name:VARCHAR(120), email:VARCHAR(254), time_zone:VARCHAR(64), preferred_reminder_window:INT)

NotificationPrefs(user_id:UUID [PK, FK to User.user_id], email_reminders:BOOLEAN, missed_dose_summary:BOOLEAN)

Drug(drug_id:UUID [PK], rxnorm_code:VARCHAR(40), name:VARCHAR(200))

SideEffect(drug_id:UUID [FK to Drug.drug_id], effect:VARCHAR(200), description:TEXT, PK: (drug_id, effect))

Prescription(rx_id:UUID [PK], user_id:UUID [FK to User.user_id], drug_id:UUID [FK to Drug.drug_id], frequency:VARCHAR(64), qty_on_hand:INT, refills:INT, rx_text:TEXT)

Reminder(reminder_id:UUID [PK], user_id:UUID [FK to User.user_id], rx_id:UUID [FK to Prescription.rx_id], remind_time:TIMESTAMP, override_frequency:VARCHAR(64))

EmergencyContact(contact_id:UUID [PK], user_id:UUID [FK to User.user_id], name:VARCHAR(120), phone:VARCHAR(32), email:VARCHAR(254), trigger:VARCHAR(140))

Disease(disease_id:UUID [PK], name:VARCHAR(200), primary_specialty:VARCHAR(120), description:TEXT)

DiseaseSymptom(disease_id:UUID [FK to Disease.disease_id], symptom:VARCHAR(200), description:TEXT, PK: (disease_id, symptom))

Doctor(doctor_id:UUID [PK], name:VARCHAR(120), specialty:VARCHAR(120), address:VARCHAR(300), phone:VARCHAR(32))

Domain notes: UUIDs may be stored as CHAR(36); timestamps may include time-zone information; VARCHAR lengths are approximate and can be adjusted as needed.