**Assignment: Translating Medical Policy into Computable Criteria, policy engine that can handle any new policy text.**

* **Background**
  1. Medical policies are written in natural language. Case managers, utilization management (UM) teams, and payers often need to translate policy criteria into computable rules for decision support, claims adjudication, or prior authorization.
  2. In this assignment, you will analyze a bariatric surgery policy and translate it into structured SQL-like filters and/or a knowledge graph representation.
  3. This exercise mimics real-world workflows in utilization management where clinicians and analysts convert policies into data-driven criteria.
  4. <https://www.anthem.com/medpolicies/abc/active/gl_pw_d085821.html>
* **Policy Summary (Natural Language)**

1. A bariatric surgery is considered medically necessary if the patient meets all the following:
   * Age: Patient must be 18 years or older.
   * BMI:
     + BMI ≥ 40, OR
     + BMI ≥ 35 with at least one comorbid condition (e.g., diabetes, hypertension, sleep apnea).
2. Procedure Code: Surgery must be one of the following CPT codes:
   * 43845 (Biliopancreatic bypass with duodenal switch)
   * 43644 (Laparoscopic Roux-en-Y)
   * 43846 (Open Roux-en-Y)
   * 43770 (Adjustable gastric banding)
   * 43775 (Sleeve gastrectomy)
   * 43842 (Vertical banded gastroplasty)
3. Diagnosis Codes (examples):
   * E66.01 (Morbid obesity)
   * Z68.41–Z68.45 (BMI documentation codes)
   * E11.9 (Type 2 diabetes mellitus)
   * I10 (Hypertension)
   * G47.33 (Obstructive sleep apnea)
4. Documentation Requirements:
   * Evidence of participation in a weight loss program (6–12 months).
   * Pre-operative medical and mental health evaluation.
   * Patient received pre-operative education.
5. Date of Service: Must fall within the coverage period.

* **Tasks**

1. Task A: Extract Data Elements
   * From the policy, list the structured data fields needed in a claims or patient record.
   * Example: age, BMI, procedure\_code, diagnosis\_code, date\_of\_service, weight\_loss\_program\_flag.
2. Build a Knowledge Graph Representation
   * Represent the policy criteria as nodes and relationships.
   * Enrich the graph by automatically mapping terms → codes using external sources (like Google APIs or clinical knowledge bases).
     + Build a mapping layer, the knowledge graph will not just be a picture of rules — it will be grounded in real-world codes that can be applied to patient/claims data.
   * Nodes (things): Patients, Procedures (CPT), Diagnoses (ICD-10), Policies, Criteria (age, BMI, documentation).
   * Edges (relationships): Patient -> has\_procedure -> CPT:43845  
     Policy -> requires -> BMI >= 40  
     Policy -> requires -> Comorbidity
   * This allows:
   * Policy reasoning: Query all policies a patient qualifies for.
   * Explainability: Show why a patient met/didn’t meet criteria (graph paths).
   * Reuse: Common conditions (e.g., hypertension, obesity) are shared nodes, not repeated in every SQL query.
   * Maintenance: Update a diagnosis code or threshold once → it propagates to all policies that reference it.
   * Example triples (in **subject → predicate → object** format):
     + Patient → has\_procedure → CPT:43845
     + Patient → has\_condition → ICD10:E66.01
     + Patient → meets\_criteria → Bariatric\_Surgery\_Policy
     + Bariatric\_Surgery\_Policy → requires → Age >= 18
     + Bariatric\_Surgery\_Policy → requires → Weight Loss Program Documentation
     + Take a **policy text block**.
     + Identify key **clinical terms**.
     + Use **Google API** (or scraping ICD10data, CMS, AMA CPT) to get the **codes + descriptions**.
     + Store these as **nodes/edges in a KG** (Neo4j or NetworkX).
     + Show how a **patient’s record (with codes)** connects to a **policy rule** and passes/fails eligibility.
3. Evaluation Criteria
   * Completeness: Did you capture all criteria (age, BMI, procedure, diagnosis, documentation, coverage)?
   * Accuracy: Do your SQL/DataFrame filters align with policy intent?
   * Clarity: Is your knowledge graph structured and readable?
   * Creativity: Did you add extra insights (e.g., reusable functions, validation checks)?

**Notes**

* A Patient connected to possible procedures (CPT codes) and conditions (ICD-10 codes).
* A Policy node requiring specific eligibility criteria (age, BMI thresholds, documentation, coverage period).
* Relationships like has\_procedure, has\_condition, and requires.