Assignment Based on ATT&CK-KG

Level 1 — Easy (query warm-ups)

1. List all labels in the graph and their counts.
   Use: kg_labels(), kg_node_counts_by_label()
   Deliverable: top 10 labels by node count.

2. List all relationship types and their counts.
   Use: kg_relationship_types(), kg_relationship_counts()
   Deliverable: top 10 rel types by frequency.

3. What stix_type buckets exist under :Attack, and how many nodes in each?
   Use: kg_attack_stix_distribution()
   Deliverable: a small table of stix_type → count.

4. Show 10 sample :Attack nodes (any types).
   Use: kg_sample_nodes("Attack", limit=10)
   Deliverable: mini table with name, stix_type, id (or keys).

5. Print 20 technique names alphabetically.
   Use: list_techniques(limit=20)
   Deliverable: 20 technique names.

6. Print 20 intrusion-set (group) names alphabetically.
   Use: list_groups(limit=20)

7. Print 20 software names (with kind = tool/malware).
   Use: list_software(limit=20)

8. Print all ATT&CK tactics (x-mitre-tactic) shortnames.
   Use: list_tactics()

9. Print 20 mitigation names.
   Use: list_mitigations(limit=20)

10. What key ATT&CK links exist in this dataset (uses/mitigates/subtechnique-of/in_tactic)?
    Use: kg_reltype_property_counts(), kg_check_link_presence()

Level 2 — Multistep (analytics & joins)

1. **Software overlap for a technique name** (e.g., “malicious file”).
   Use: diagnose_Q1(term="malicious file") and/or list_groups_using_attack_pattern(...), list_software_used_by_those_groups(...)
   Deliverable: software used by multiple groups that also use the matched technique(s); show software, kind, group_count, groups (sample).

2. **Top-3 techniques for a group by software implementations + mitigations** (e.g., “APT28”).
   Use: fixed version of Q2_top_3_tech_mitigations_for_group("APT28") Deliverable: technique, software_count, mitigations (list, can be empty).

3. **Most-used sub-techniques under the “execution” tactic and who uses them.**
   Hint: filter techniques in the execution tactic via IN_TACTIC, then count groups on ATTACK_REL {rel_type:'uses'}.
   Deliverable: sub-technique name, number of groups, top 5 groups.

4. **Shared tactics between two very different software families** (e.g., “rar” and “PsExec”).
   Use: software_tactics("rar"), software_tactics("psexec"), then Q4_shared_tactics_between_software("rar","psexec")
   Deliverable: intersecting tactics + example techniques from each side.

5. **Unmitigated software risks.**
   Goal: find software → techniques that (a) have no mitigates from any course-of-action and (b) are used by at least one group.
   Hint: NOT ((:Attack {stix_type:'course-of-action'})-[:ATTACK_REL {rel_type:'mitigates'}]->(tech)) and existence of (grp:intrusion-set)-[:ATTACK_REL {rel_type:'uses'}]->(tech) and (soft:tool|malware)-[:ATTACK_REL {rel_type:'uses'}]->(tech).
   Deliverable: software, technique, example_group.

6. **Technique → Tactic coverage for a given group** (e.g., APT29).
   Goal: for techniques used by the group, list the unique tactics they belong to.
   Deliverable: tactic, techniques_count, example_techniques.

Level 3 — Graph-algorithm flavored

1. **Centrality of techniques** (influence/pivot points).
   Intent: PageRank on a bipartite-ish subgraph (groups ↔ techniques ↔ software) where edges are ATTACK_REL {rel_type:'uses'}.
   GDS path: project a graph of intrusion-set, attack-pattern, tool, malware with uses edges; run PageRank; return top 20 attack-pattern.
   Fallback: degree-like proxy — rank techniques by number of distinct (groups + software) attached.
   Deliverable: top 20 techniques with score.

2. **Communities ("attack kits") of groups+software+techniques.**
   Intent: Louvain (if GDS available).
   Fallback: your "connected components over uses" approach (GA3_communities_uses_fast(...)) that seeds from groups and expands uses up to N hops; dedupe components; summarize each community.
   Deliverable: top 10 communities by size; show counts per type and sample members.

3. **Link prediction (what techniques might a group adopt next?).**
   Use: the fixed GA4_link_prediction_for_group("APT1") where the degree subquery is corrected (no AS st on a pattern).
   Approach: Adamic-Adar-like weight from group's known software to candidate techniques not yet linked; optional "other groups" signal.
   Deliverable: top 15 predicted techniques with scores.