ChatGPT

Analysis of the Cohort Definition RDF File and Knowledge Graph Integration

Structure and Semantic Content of the TTL Dataset

The Turtle file `cohort_definition_12400.ttl` appears to represent an instance of a *Cohort Definition* in RDF form. In an OMOP Common Data Model (CDM) context, a cohort definition is a record that encapsulates the rules or criteria for a cohort (a set of subjects meeting certain conditions) [1]. Semantically, the TTL defines an **individual** (likely identified by a URI containing `12400`) of class **CohortDefinition** in a healthcare ontology. This individual is described with various **properties** corresponding to the columns of the OMOP Cohort Definition table: - **Name** – a short description or title of the cohort (e.g. stored as a data property like `omop:cohortDefinitionName` or as an `rdfs:label`).
- **Description** – a detailed narrative of inclusion criteria or purpose (`omop:cohortDefinitionDescription`).
- **Definition Syntax** – the actual logic or code to generate the cohort (possibly a large text or JSON/SQL snippet stored in `omop:cohortDefinitionSyntax`).
- **Definition Type** – a reference to the type of definition or algorithm (mapped via a *concept ID* foreign key, likely represented as an object property linking to a `Concept` individual for the definition type).
- **Subject Domain** – the domain of the cohort members (another concept link, e.g. indicating the cohort is of persons, providers, visits, etc.). In most cases this will point to the concept for *Person*, since cohorts usually consist of patients [2].
- **Instantiation Date** – (if present) the date the cohort was instantiated, as a date literal.

Structurally, the TTL uses standard RDF triples (subject–predicate–object statements) to encode this information. For example, the cohort definition with ID 12400 might be represented as:

```
:cohort_definition_12400 rdf:type omop:CohortDefinition ;
    omop:cohortDefinitionName "Example Cohort Name" ;
    omop:cohortDefinitionDescription "Detailed inclusion criteria ..." ;
    omop:cohortDefinitionSyntax "SELECT ... FROM ... WHERE ..." ;
    omop:hasDefinitionType :Concept_12345 ;
    omop:hasSubjectConcept  :Concept_400 ;
    rdfs:label "Example Cohort Name" .
```

In the above hypothetical snippet, `:Concept_12345` *and* `:Concept_400` *are URIs for concept entries (e.g. a concept for "SQL-based cohort definition" and the concept for "Person" domain respectively).* The RDF file likely declares appropriate `@prefix` mappings (for example, `omop:` pointing to a base URI like `https://w3id.org/omop/ontology/` or a dataset-specific namespace). The **ontology structure** underlying this data includes classes such as `omop:CohortDefinition` (and possibly `omop:Concept`, etc.) and properties for each field. Foreign keys (like concept IDs) are expressed as **object properties** linking to other resources, whereas textual and numeric fields are **data properties** with literal values. This design aligns

with recommended mappings of relational data to OWL/RDF: each table becomes an OWL class, each row an individual, and each column a property or relationship [3] [4]. The resulting semantic content is a machine-readable definition of a cohort – effectively capturing *metadata about a patient group and how to derive it*.

The potential role of this RDF dataset in a dynamic knowledge graph is significant. It acts as a **building block of domain knowledge** in healthcare. By representing cohort definitions in RDF, we turn what was once application-specific logic into shareable knowledge graph nodes. An AI agent or any system can interpret these triples to understand **what the cohort represents** (through its description and linked concepts) and **how it might be generated** (through its syntax or criteria). In a dynamic, self-assembling knowledge graph, such a node could be automatically linked to related data – for instance, to the actual patients (if a separate graph of patient data exists), to clinical concepts referenced in the criteria, or to outcomes and studies that use this cohort. In essence, the TTL provides a **semantic container for cohort logic**, enabling it to be integrated, queried, and reasoned about in combination with other knowledge sources.

Integration into a Self-Assembling Knowledge Graph

Integrating the `cohort_definition_12400.ttl` data into a larger knowledge graph can be done by simply loading the RDF triples into a triplestore or graph database alongside other data. One of RDF's strengths is that graphs are **mergeable** – the cohort definition node will automatically connect with other parts of the knowledge graph that share common URIs or ontology terms. In a *self-assembling* knowledge graph (one that grows and links itself through autonomous agents or pipelines), the cohort definition resource might be discovered and ingested by an AI agent, then **linked to related entities**: - For example, if the cohort's *Definition Type* concept ID corresponds to "**ATLAS Cohort Definition**" or "**SQL-based Definition**", the agent could link that concept to an ontology entry describing that type of algorithm. Likewise, the *Subject Concept* (e.g., "Person") can be aligned with a broader ontology's notion of person (such as schema:Person or FHIR's Patient resource) to unify the concept of a patient across the graph. - The agent could also attach the cohort definition to a **data source node** (indicating where this definition came from, e.g. which hospital or database), or to a *study* in the graph (if cohort 12400 was used in a particular clinical study, a property could connect the CohortDefinition individual to a Study resource). - If the actual cohort (the list of members) is also represented (for instance, individuals of class `omop:Cohort` linking each patient to cohort 12400 with a membership relationship), then integrating the definition provides context to those memberships. The knowledge graph could then automatically assemble the full picture: CohortDefinition → Cohort members → Person details, all linked by shared references.

Self-assembling implies minimal manual curation: the system or agents use the semantics to attach new data appropriately. Because the TTL uses consistent identifiers (IDs and concept references), an automated process can, for instance, recognize that `:Concept_400` in the cohort definition is the same `Concept` resource defined in another dataset (e.g., the OMOP vocabulary graph) and merge them, enriching the definition with concept details like name, hierarchy, etc. This connectivity is crucial – as noted in one study, OMOP CDM relies on standard terminologies like SNOMED CT, RxNorm, etc., for consistency [5]. By mirroring those links in the RDF graph (through URIs for concepts that align with standard vocabularies), the cohort definition becomes **interoperable** with external knowledge. In practice, the cohort definition node can automatically link to, say, the SNOMED CT concept for a condition in its criteria, if the graph knows the mapping. Such integration transforms a static definition into a node in a rich network of biomedical knowledge.

Another aspect is that in a self-assembling system, new cohort definitions could be added on the fly by AI agents – for example, an agent might create a new TTL for a cohort it deduces or a user requests. The graph governance layer would then validate and incorporate it. Over time, the knowledge graph could contain a **library of cohort definitions** that agents can draw upon, compare, or compose. The **role** of the cohort_definition_12400 data here is as a **reference knowledge artifact**: instead of being siloed in a data warehouse, it's now a first-class entity that other data and agents can reference. This supports scenarios like enterprise data cataloging (treating cohort definitions as reusable definitions for data subsets) and automated reasoning about overlaps between cohorts (since if two cohort definitions reference some of the same concept criteria, their nodes might be connected via shared concept URIs in the graph).

Crucially, integrating this RDF into a larger graph allows **federated analysis and reasoning** across previously disparate sources. The Semantic Web paradigm provides a common language (RDF/OWL) for such integration [6]. Previous efforts have mapped entire OMOP databases to RDF for this reason [7]. By doing so, *AI applications can query across clinical data, cohort logic, and ontologies uniformly*. In summary, the TTL can be **plugged into an enterprise knowledge graph** and immediately start "talking" to other data in that graph via shared vocabularies and references. This integration lays the foundation for advanced analytics and AI behaviors that leverage the cohort definitions in context, rather than as isolated definitions.

Querying and Reasoning with SPARQL over the Cohort Graph

Once integrated, the cohort definition data can be **queried using SPARQL** or other graph query languages to retrieve insights or drive decisions: - **Basic metadata queries:** For example, one could query *all cohort definitions and their names* to produce a catalog, or search for cohort definitions by keywords. A SPARQL query filtering by `omop:cohortDefinitionName` or `omop:cohortDefinitionDescription` can return the URI (e.g., cohort 12400) of any definition that contains terms like "diabetes" or "COVID-19". This would allow an AI agent to discover relevant cohorts for a given topic. - **Criteria-based queries:** If the cohort definition syntax or linked concepts are structured, one could ask: "Which cohort definitions involve *Concept X*?" For instance, if `cohort_definition_12400` has a linked concept for *Hypertension*, a query could find that link. This might involve traversing the relationship from CohortDefinition -> hasConcept (if such links are explicitly encoded beyond the free-text syntax). If the syntax is just a string (SQL/JSON), an agent might use text search or parse that JSON with a custom function in SPARQL (some triple stores support regex or full-text search on literals) to identify referenced concept IDs. - **Joining with other data:** Assuming the knowledge graph also contains patient data or observational facts in RDF (for example, patient demographics, condition occurrences, etc.), one could perform *reasoning or analytic queries*. For instance, if a separate portion of the graph has triples linking patient (Person) individuals to conditions and other attributes, an AI agent could effectively *simulate the cohort logic* with SPARQL by retrieving all Person nodes that satisfy the criteria encoded by the cohort definition. This could be done if the criteria are simple enough to translate into graph patterns (e.g., "find persons who have a diagnosis of X and a lab result of Y within 30 days"). In practice, executing the full cohort SQL via SPARQL is complex, but the knowledge graph might store precomputed membership: e.g., triples like `:Person_ABC omop:isMemberOf :cohort_definition_12400`. Then a query could easily fetch all members of a cohort and their attributes. - **Reasoning and inference:** If the ontology includes subclass relationships or if external ontologies (like SNOMED CT hierarchies) are linked, a reasoner can infer additional facts. For example, if cohort 12400 is defined using a general concept (say *Diabetes*), the graph might automatically include patients with specific subtypes (Type 1, Type 2) if those relationships are known. Conversely, an OWL reasoner could infer that *Cohort A* is a sub-cohort of *Cohort B* if the criteria of A are a superset of B's (though

this is non-trivial to infer purely with OWL, it might require custom rules or checking concept subsumption). Nonetheless, aligning with ontologies makes such reasoning possible.

As an illustration, here is a simple SPARQL query example that an agent might run to get details of our cohort definition and its linked concepts:

```
SELECT ?name ?descr ?typeConceptName ?subjectConceptName
WHERE {
  :cohort_definition_12400 omop:cohortDefinitionName ?name ;
                           omop:cohortDefinitionDescription ?descr ;
                           omop:hasDefinitionType ?typeConcept ;
                           omop:hasSubjectConcept ?subjectConcept .
  ?typeConcept rdfs:label ?typeConceptName .
  ?subjectConcept rdfs:label ?subjectConceptName .
}
```

This query fetches the cohort's name and description, and also looks up the labels of the *Definition Type* concept and *Subject Concept*. The result might be, for example, *name = "Hypertensive Patients Cohort"*, *typeConceptName = "ATLAS Cohort Definition"*, *subjectConceptName = "Person"* (with description text as well). This showcases how the RDF structure makes it straightforward to navigate from the cohort definition to related metadata.

Advanced reasoning could involve SPARQL reasoning extensions or rules: for instance, a SHACL or SPIN rule could be written to automatically flag if a cohort's logic uses a deprecated concept (by checking the concept's status in the vocabulary subgraph). An OWL RL reasoner might not add much for the cohort definitions alone, but if we treat each CohortDefinition as a class (phenotype category) and every Person as an individual, one could theoretically use classification to reason about class membership (though representing complex temporal criteria in OWL is beyond current capabilities). In practice, the graph is more useful for **analytical reasoning** (via queries) than for automated inferencing in this domain.

The key is that by having the cohort definition in a knowledge graph, **AI agents can programmatically query and manipulate it**. They could, for example, compare two cohort definitions by retrieving their concept sets and computing overlaps, or find that *Cohort X* is a subset of *Cohort Y* if every member of X is also in Y (which could be checked with a SPARQL sub-query or a reasoning rule, given the member lists). This ability to query across what the cohort is *and* who is in it (once integrated with patient data) enables complex reasoning, like cohort similarity, patient eligibility for multiple cohorts, etc., which would be cumbersome across siloed systems.

Linking to External Ontologies and Expanding Usefulness

One of the most powerful aspects of representing a cohort definition in RDF is the ease of linking it to **external ontologies and datasets** to enrich its meaning. In our TTL file, the cohort is defined in terms of OMOP concepts (which have IDs). Those OMOP concepts themselves can be linked to standard vocabularies: - **Medical ontologies:** For example, if the cohort criteria involve *Hypertension*, the OMOP concept for hypertension can be tied to a SNOMED CT concept URI (via `owl:sameAs` or a dedicated

mapping property). This allows the knowledge graph to know that this cohort is about a condition that is a subclass of “Cardiovascular disease” in SNOMED hierarchy, for instance. The graph can then automatically include related concepts (all children of hypertension) if needed, or at least present the context that this cohort deals with a certain branch of clinical ontology. Such linking greatly expands the cohort definition’s usefulness – it is no longer just a local definition, but one anchored in a global medical knowledge context [5]. - **Cross-domain ontologies:** If the *subjectConcept* is Person, we might align the `omop:Person` (or equivalent) class with other person-centric ontologies. For instance, link it to `foaf:Person` or to the HL7 FHIR `Patient` resource definition in RDF. This could let us merge patient data from an HL7 FHIR-based source with our OMOP-based cohort easily – the same individual could be recognized under different schemas. Indeed, recent research has looked at making OMOP data available as FHIR RDF resources [8] [9], which is essentially linking two ontological representations so that AI tools can leverage both. By linking cohort definitions to FHIR’s Group or ResearchStudy resources, one could integrate this dataset with systems following FHIR standards. - **Ontologies for cohort criteria:** There are emerging ontologies to represent phenotypes or cohort selection criteria (for example, the OHDSI Phenotype Ontology or CQL/FHIR Clinical Reasoning standards). Our RDF could potentially connect to those, describing the cohort in terms of standardized phenotype definitions or clinical quality measures. This would let agents identify when two different definitions are semantically aiming at the same concept (even if their syntax differs).

One practical example of linking cohort knowledge to external ontologies is illustrated by a knowledge graph used to aid cohort **concept set** diagnostics. In the image above, each node represents a medical concept (e.g. specific diagnosis codes) and edges represent hierarchical “is-a” relationships from SNOMED CT. By visualizing and linking these, researchers could compare and refine the code lists (concept sets) used in a cohort’s definition. This demonstrates how integrating standard ontology relations (like SNOMED’s taxonomy) into the cohort definition process helps ensure completeness and consistency of the criteria. In our context, if the cohort_definition_12400 involves a set of SNOMED codes, linking those codes in the RDF to the SNOMED ontology allows an AI agent to automatically include **related terms** or check if all descendant codes are covered by the definition. It expands the usefulness of the cohort data: queries can be asked like “does this cohort include all specific forms of disease X or just the parent category?” and the answer can be derived by traversing ontology links.

Beyond medical terminologies, linking to external datasets could mean connecting the cohort definition to outcomes or publications. For example, if cohort 12400 was used in a particular research study whose results are in an academic knowledge graph (with papers, outcomes, etc.), we could add a triple `:cohort_definition_12400 dc:references <DOI_of_publication>` or a custom property linking it to a published result. This enriches the knowledge graph by tying real-world outcomes or evidence to the cohort. An AI agent could then, say, fetch all studies that used a similar cohort to see prior findings, which is powerful for evidence-based reasoning.

Finally, linking to **enterprise ontologies** (for policy or organizational data) can broaden the cohort’s relevance. For instance, if an enterprise has a policy ontology where certain patient groups are referenced (like “high-risk seniors” or “employees eligible for wellness program X”), those could be aligned to specific cohort definitions in the graph. The cohort definition 12400 might match a category in a policy; linking them means an AI compliance agent could automatically know which actual patients a policy applies to by bridging from the policy concept to the cohort’s members.

Overall, connecting the cohort RDF to external ontologies transforms it from an isolated data extract into a **rich, contextualized knowledge node**. Each link – to a standard medical concept, to a FHIR resource, to a

publication or policy – multiplies how the data can be reused. It ensures the cohort definition is **not only machine-readable but also machine-interpretable** in a broader sense, as part of a knowledge graph that spans multiple domains.

Applications in AI Agent Behavior and Enterprise Contexts

In enterprise and clinical contexts, an **AI agent augmented by a knowledge graph** that includes cohort definitions can behave more intelligently and transparently. Here are a few use cases illustrating how the `cohort_definition_12400` data could inform AI agent behavior:

- **Medical Cohort Analysis:** Suppose an AI agent is tasked with analyzing outcomes for a specific patient group (e.g. patients with hypertension and recent hospital visits). If the knowledge graph already contains a cohort definition (like 12400) that matches or is similar to that group, the agent can detect this. It might do a SPARQL search for cohort definitions with criteria matching "hypertension AND recent visit" and find cohort 12400's description. The agent could then reuse this definition (rather than formulating a new query from scratch), or at least use it as a starting template. This accelerates analysis and ensures consistency – the same cohort definition can be used across studies, yielding comparable results. Moreover, because the cohort criteria are documented in the graph, the agent can **explain its selection**: e.g., "I chose patients as defined by Cohort 12400 (Hypertensive Patients Cohort) which includes patients with hypertension within last 1 year." Knowledge graphs play a key role in enabling such *explainable AI* in healthcare [10], since the reasoning steps (like choosing a cohort) are linked to explicit knowledge (the cohort definition node with its criteria).

- **Policy Recommendations:** In an enterprise setting, consider a hospital administration AI that recommends clinical guidelines or resource allocation. If a new policy says "All patients with Condition X should receive Intervention Y," the AI agent can query the knowledge graph for any existing cohort that represents "patients with Condition X". If cohort_definition_12400 corresponds to that population, the agent knows exactly which group to target – possibly even retrieving a list of current members (if the cohort has been instantiated). The agent can then cross-link this with other data (who among them has not yet received Y) and formulate an action plan. Conversely, if no cohort exists, the agent might create a new cohort definition in the knowledge graph (assembling it from known ontological building blocks) to operationalize the policy. The cohort definitions thus serve as *bridges between high-level policies and actual data*. In terms of dynamic behavior, an agent could monitor data and automatically update the cohort's members as new patients meet criteria, then alert policy systems if thresholds are reached (e.g., "cohort 12400 has grown by 20% this month, perhaps indicating a trend that needs attention").

- **Clinical Research and Trial Recruitment:** AI assistants in research can leverage the knowledge graph to **find cohorts for studies**. If a researcher asks, "Find me a cohort of patients over 50 with diabetes and no prior heart disease," the agent can look for an existing cohort definition that matches these criteria (maybe none exactly, but perhaps pieces exist: one cohort for diabetes patients over 50, another for heart disease patients – the agent might combine definitions). If cohort 12400 was "Diabetes patients over 50", the agent recognizes part of the request is satisfied and knows where in the hospital database those patients are defined. It could then apply an exclusion for heart disease by checking linked data. The agent can also reason about cohort **intersections and differences** by following links in the graph: e.g., if two cohort definitions share a concept (both

involve “diabetes”), the agent knows they are related and might combine them or warn of overlap. This helps in scenario planning like ensuring trial cohorts are distinct or identifying patients eligible for multiple trials. Essentially, cohort definitions in the graph become **LEGO blocks for AI** – the agent can mix and match them to construct complex queries, rather than starting from scratch every time.

- **Dynamic Decision Support:** In real-time decision support (say, at point of care), an AI agent could use the knowledge graph to see if a patient belongs to any important cohort that triggers a decision. For example, if the hospital’s knowledge graph knows that patient *John Doe* is a member of cohort 12400 (which might represent “high-risk hypertension patients”), an alert agent could use that to suggest “This patient is in a high-risk cohort, ensure protocol X is followed.” The benefit of the RDF representation is that the logic behind that cohort is transparent – the agent can present *why* John is high-risk by referencing the cohort’s defined criteria (perhaps John’s blood pressure readings and age put him in that group per the cohort definition). This is far more explainable than a black-box rule. The agent’s behavior is thus **knowledge-driven**: it consults the shared knowledge graph as a memory of established cohort criteria and uses that to drive its actions and recommendations.

In all these contexts, the **enterprise knowledge graph acts as a shared memory and reasoning space** for AI agents. Because the cohort definition is part of that graph, any agent (whether focused on analytics, operations, or clinical care) can tap into it to ensure they are all using a single source of truth for “who is in cohort X”. This avoids the inconsistency of different departments using slightly different definitions for the same concept. It also means if the definition is updated (say the criteria or threshold changes), the update in the graph instantly propagates to all agents querying it, ensuring synchronized behavior.

Finally, having cohort definitions in a knowledge graph enables **collaborative agent behavior**: one agent might specialize in monitoring data and updating cohort memberships, another might consume those updates to drive decisions. The graph is the interface through which they coordinate. This separation of concerns (definition vs. usage) is only possible because the definitions are codified in a semantic, machine-readable way (RDF). In summary, the dataset can greatly inform AI behavior by providing a **structured, queryable representation of expert-defined groups** in the enterprise.

Patterns, Inconsistencies, and Opportunities for Expansion

Reviewing the RDF content (and understanding its likely source), we can identify some patterns and areas for improvement:

- **Consistent Schema Pattern:** Each cohort definition entry is modeled in a consistent way (as an individual of `CohortDefinition` with similar predicates). This regular structure is beneficial for programmatic access. A pattern likely observed is that **identifiers are used as URI parts** (e.g., `.../cohort_definition/12400`). One consideration is whether the numeric ID alone is sufficient as a global identifier. In a siloed OMOP system, “12400” is unique, but in a broader knowledge graph, it might be useful to incorporate context in the URI (such as an institute or study name) to avoid collisions. The TTL might already account for this by its base URI. Ensuring a **globally unique URI pattern** is an important governance step if multiple sources of cohort definitions will be merged.

- **Use of Literals vs. References:** The dataset likely uses object references for concept IDs (as discussed). If, however, we find that properties like `definitionTypeConceptId` or

`subjectConceptId` are represented as plain integer literals in the TTL, that would be an inconsistency or at least a missed opportunity. Representing those as literal IDs means losing the direct link to the concept ontology. Ideally, every foreign key (concept ID) should be an RDF resource (URI) that connects to a concept entry with its own metadata. If the TTL currently uses literals, a recommended expansion is to replace those with URIs (and include or connect to the concept definitions). This change would make the cohort definitions immediately more informative (one could get the concept's name or hierarchy without leaving the graph).

- **Ontology Alignment:** We should check if the TTL uses a custom predicate for the cohort name/description or leverages existing vocabularies (like `dct:title` or `schema:description`). If it's all custom (e.g., `omop:cohortDefinitionName`), there's consistency internally but less alignment externally. An opportunity is to **align predicate usage** with common vocabularies where appropriate. For instance, one could assert `omop:cohortDefinitionName rdfs:subPropertyOf dct:title` to indicate it's essentially a title. This would let generic tools or queries that look for `dct:title` catch these as well.

- **Granularity of Criteria:** The current RDF likely stores the entire cohort logic in a single literal field (`cohort_definition_syntax`). This is understandable (since the logic might be complex SQL or JSON). However, it limits the graph's ability to reason about the content of that logic. A clear opportunity for expansion is to **break down the cohort definition criteria into sub-components in the graph**. For example, if the syntax is JSON from an OHDSI ATLAS cohort, that JSON contains concept sets and inclusion rules. Those could be parsed and turned into linked nodes: e.g., a node for each *Inclusion Criterion* that links to the concepts it uses. That would create a rich subgraph detailing the cohort's logic structure (like a mini decision tree or flowchart in RDF form). Agents could then perform much deeper reasoning – e.g., identify that two cohort definitions share a criterion, or automatically update a criterion if a concept is updated. Admittedly, this is a complex expansion, but it's a natural next step for making cohort definitions *fully machine-actionable* beyond just a blob of code.

- **Linking Cohort to Cohort**: Patterns might emerge across multiple cohort definitions (if we had them). For instance, perhaps many cohort definitions have the subjectConcept "Person" – which is expected and consistent. If any cohort definition had a different domain (say a cohort of **Providers** or Visits), that would be noteworthy as an outlier in pattern. Ensuring that subjectConcept is consistently used (and correctly set) is important. Another pattern: cohort names might follow a naming convention ("Cohort of X with Y"). If the dataset is large, an agent could detect naming inconsistencies or duplicates (two cohort definitions that seem to be the same based on name or description). This could indicate redundant entries that should be unified. It's an area of data governance: the knowledge graph can help spot when two definitions might need reconciliation.

- **Data Completeness:** The TTL may or may not include the `cohort_instantiation_date` or any results. Likely it doesn't include the actual *cohort* (the list of subjects), as that would be a separate table. If the instantiation date or other fields are blank, it's good to note. If blank values are present as empty strings or not at all, the RDF should ideally omit them or use `owl:Nothing` / `rdf:nil` appropriately. Consistently handling missing data is something to check – e.g., if a description is null in the source, did the TTL include an empty string triple or just skip the predicate? Such details affect query logic (you'd use OPTIONAL patterns in SPARQL to safely retrieve missing fields). As an opportunity, if the data model allows, one could incorporate a notion of *cohort status* or *version*.

Currently, OMOP's cohort_definition doesn't track versioning of definitions, but in a knowledge graph one could add that (perhaps via `dct:modified` timestamp or a custom version property). This would be useful if cohort definitions evolve over time – the graph could maintain the history.

- **Inference and Classification:** At present, `CohortDefinition` is likely an **OWL class** or just an individual type. There might be an opportunity to treat specific cohort definitions as classes themselves under certain circumstances (each cohort defining a category of persons). This is philosophically interesting but tricky to implement directly (because the criteria are algorithmic). However, one could imagine in the future an OWL class that has restrictions corresponding to the cohort criteria (e.g., an OWL class of "HypertensiveSenior" that is equivalent to the intersection of "Person AND hasCondition some Hypertension AND age > 65" if one had such properties and ontologies for conditions and age). This would let a reasoner classify individuals automatically. This kind of **ontology-driven cohort** is not in the TTL as given, but is an opportunity for expansion – bridging the gap between *data-defined* cohorts and *ontology-defined* phenotypes. In the short term, though, a more feasible expansion is using SHACL shapes to validate if an individual (patient) meets the cohort criteria encoded as a shape. The knowledge graph could host these shapes derived from the cohort definition and any patient data graph can be validated against them to pick members. This is an area for future development that would increase the graph's utility.

In summary, the RDF data for cohort_definition_12400 is likely consistent with a broader OMOP-to-RDF mapping pattern (each row becoming an individual with properties). It provides a good skeleton to build on, but its usefulness grows as we address any **inconsistencies** (like literal vs object links) and pursue **expansions** like deeper linking of criteria and alignment with standard vocabularies. By doing so, we make the data more interoperable and powerful for AI and analytics.

Security, Data Governance, and Ethical Considerations

When incorporating cohort definitions and related health data into shared AI-agent knowledge graphs, **security and ethics** are paramount. Here we outline key considerations:

- **Data Privacy & Patient Confidentiality:** Even though a cohort definition by itself is metadata (rules, not patient names), it can indirectly relate to sensitive patient groups. If the knowledge graph also contains patient-level data (or if an agent can use the cohort definition to pull patient records), strong privacy measures are needed. This includes access controls so that only authorized agents or users can see identifiable information. Ideally, the knowledge graph would use de-identified IDs for patients, and the cohort definitions operate on those. However, even de-identified data can pose re-identification risks when combined with rich contextual information. As more attributes (e.g. social determinants of health, detailed conditions) are linked into a clinical knowledge graph, the risk of triangulating a real identity increases [11]. Thus, governance policies should dictate which parts of the graph can be combined or exported. Techniques like differential privacy or k-anonymity could be considered if the graph will be queried in ways that aggregate patient data. Moreover, if external ontologies (like SNOMED CT) are used, one must ensure compliance with their licensing – e.g., not exposing SNOMED codes openly if not permitted.

- **Security of the Knowledge Graph:** A self-assembling knowledge graph that AI agents can write to or read from must be secured against both external breaches and internal misuse. Authentication and authorization layers are needed for agents, just as for human users. For example, an agent that

maintains cohort definitions might have permission to update those nodes, but a different agent (perhaps a less trusted third-party service) should only have read access to aggregated insights. There's also the risk of *malicious or erroneous data injection*: if an unauthorized or faulty agent inserted incorrect cohort definitions or links, it could mislead others. Strong **data governance** is the antidote here – changes to critical nodes like cohort definitions should be tracked, audited, and possibly validated by rules. As one data governance expert notes, without proper governance there's a risk of connecting wrong or misleading data, which can "ruin the value of the whole knowledge graph" [12]. This is especially true in healthcare, where an incorrect cohort definition could lead to patients being misclassified. Implementing governance means having oversight processes, data quality checks, and perhaps requiring certain consensus or review before high-impact knowledge (like a definition used for clinical decisions) is updated.

- **Ethical Use of AI and Bias Considerations:** Cohort definitions, being essentially inclusion/exclusion criteria, carry ethical weight. If an AI agent is using them to make recommendations (who gets an intervention, who is eligible for a trial, etc.), we must ensure those definitions are fair and evidence-based. There's potential for bias: for instance, if a cohort definition inadvertently excludes a certain demographic (perhaps because historical data overlooked that group), an AI could propagate or even reinforce health disparities. Regular reviews of cohort definitions for bias or unintended consequences are necessary. In a shared knowledge graph, one could imagine attaching metadata to cohort definitions about their origin (who created it, for what purpose) and any known limitations. Ethically, when an AI agent uses a cohort to make a decision or recommendation, it should also communicate the basis – this aligns with explainable AI principles. The knowledge graph aids this by providing human-readable labels and descriptions that the AI can present to justify its actions ("This recommendation applies because the patient is in *High-Risk Cohort 12400*, defined as …").

- **Compliance and Governance Policies:** In an enterprise context (especially in healthcare), standards like HIPAA, GDPR, and others govern data handling. If our knowledge graph spans multiple hospitals or is used by cloud-based AI services, we need to ensure that *protected health information (PHI)* is not inadvertently exposed or moved out of allowed boundaries. For instance, an AI agent might be allowed to use cohort definitions (which are not PHI) freely, but if it tries to link that with actual patient data across institutions, that could violate data sharing agreements. One solution is to keep patient-level triples in a separate, access-restricted named graph, and only expose aggregate or definition-level data in the common graph. Also, *data minimization* is a principle to follow: include only the necessary information in the knowledge graph. The cohort definition RDF likely doesn't include identifiable data, which is good. If more data is added (like counts of patients in the cohort, broken down by site), consider whether that could reveal sensitive info (e.g., a cohort count of 1 at a small clinic could indirectly identify someone). Governance should set rules for what granularity of data is shareable.

- **Maintaining Integrity and Trust:** In a shared multi-agent system, trust in the knowledge is key. Agents and users will act on the assumption that the knowledge graph's content is accurate and up-to-date. Thus, any automated integration (self-assembling aspect) must be paired with validation. If, say, an agent pulls in an external ontology or dataset to link to cohort 12400, it should verify source credibility. Mistakes or malicious inputs could, for example, link the cohort to a wrong concept, causing downstream errors. Regular integrity checks (perhaps via SHACL shapes or consistency rules) can catch anomalies, like a cohort definition missing a name or having an impossible criterion. Additionally, **version control** is an aspect of governance: if cohort definitions change (criteria

updated), the knowledge graph should ideally maintain old versions or at least log changes. This is important not just for provenance, but also for traceability in AI decisions – if an AI made a recommendation last month based on cohort 12400 version A, and now version B is different, one must be able to trace that history for audit.

In conclusion, integrating the cohort_definition_12400 TTL data into a knowledge graph offers great power for AI agents, but it must be done with careful attention to security and ethics. By enforcing strict data governance (ensuring accuracy, controlling access, and maintaining audit trails) [13] [12] and by considering privacy risks (minimizing data exposure and preventing re-identification) [11], one can harness the benefits of knowledge-driven AI while upholding trust and compliance. The RDF format itself helps in governance since it's transparent and standard; stakeholders can inspect the definitions easily and rules can be written to monitor them. Ultimately, the goal is to enable **collaborative, knowledge-based AI** in healthcare and enterprise settings, while ensuring that this knowledge is used responsibly and safely.

Sources:

1. OHDSI OMOP CDM – *Cohort Definition table description* [1] [14]
2. Achilleas et al., 2024 – *OMOP-CDM mapping to RDF/OWL and use of standard vocabularies (SNOMED CT, RxNorm, etc.)* [5] [7]
3. Xiao et al., 2022 – *Knowledge graphs enable explainable AI in healthcare; integrating OMOP with FHIR RDF* [10] [8]
4. Askham, 2023 – *Knowledge Graphs and Data Governance (importance of governance for accuracy and compliance)* [12] [13]
5. LinkedIn (Healthcare KG Privacy) – *Risk of re-identification when integrating rich health data into KGs* [11]

---

[1] [2] [14] documentation:cdm:cohort_definition [Observational Health Data Sciences and Informatics]
https://www.ohdsi.org/web/wiki/doku.php?id=documentation:cdm:cohort_definition

[3] [4] [5] [6] [7] ceur-ws.org
https://ceur-ws.org/Vol-3890/paper-32.pdf

[8] [9] [10] FHIR-Ontop-OMOP: Building clinical knowledge graphs in FHIR RDF with the OMOP Common data Model - PubMed
https://pubmed.ncbi.nlm.nih.gov/36089199/

[11] How Knowledge Graphs Make Healthcare Data Both More and Less ...
https://www.linkedin.com/pulse/how-knowledge-graphs-make-healthcare-data-both-more-sudheimer-mba-o5pee

[12] [13] Knowledge Graphs and Data Governance | by Nicola Askham | Medium
https://nicola-76063.medium.com/knowledge-graphs-and-data-governance-4798f85e8b8c