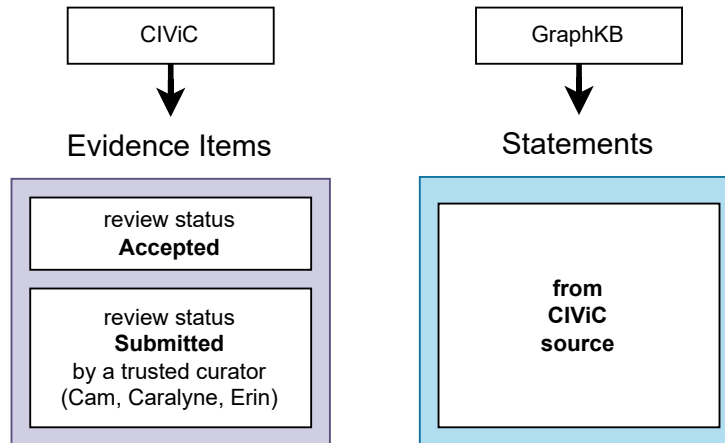


A

Fetching data

A first query fetch all CIViC Evidence Items with an "Accepted" review status. A "--trustedCurators" flag can be used to also fetch "Submitted" Evidence Items from one or more given curator user id, and as many more queries will be made. Since it's a GraphQL API, for each query, all needed join data are fetched at the same time.

GraphKB statements originating from CIViC are also fetched. They will later be used for content comparison since we favor updating records over deleting and re-creating them.



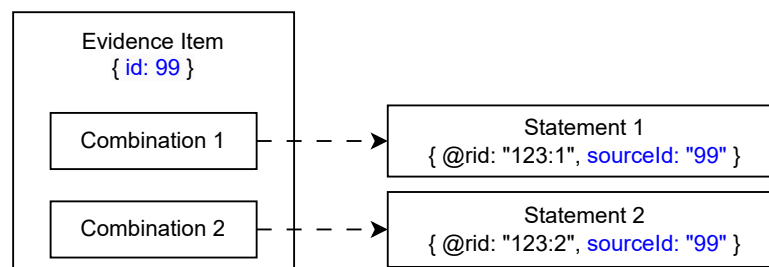
Orphan statements

Statements with a sourceId that doesn't match any Evidence Item id get deleted. These are considered deprecated; it can happen if an Evidence Item gets removed from CIViC or if its review status gets downgraded. This behavior can be overridden by using the "--noUpdate" flag, which can prove itself useful when omitting some or all trusted curators.

B

Splitting individual Evidence Item into combinations

A single CIViC Evidence Item cannot always be translated into a unique GraphKB Statement because of the way both systems handle conditions. In GraphKB, each statement condition is linked to the others with an implicit "AND" operator; the 'OR' operator is not supported. One then needs to create as many statements than there is sets of conditions intended to be linked by "OR". In the context of the GraphKB CIViC loader, each of these sets is called a "combination".



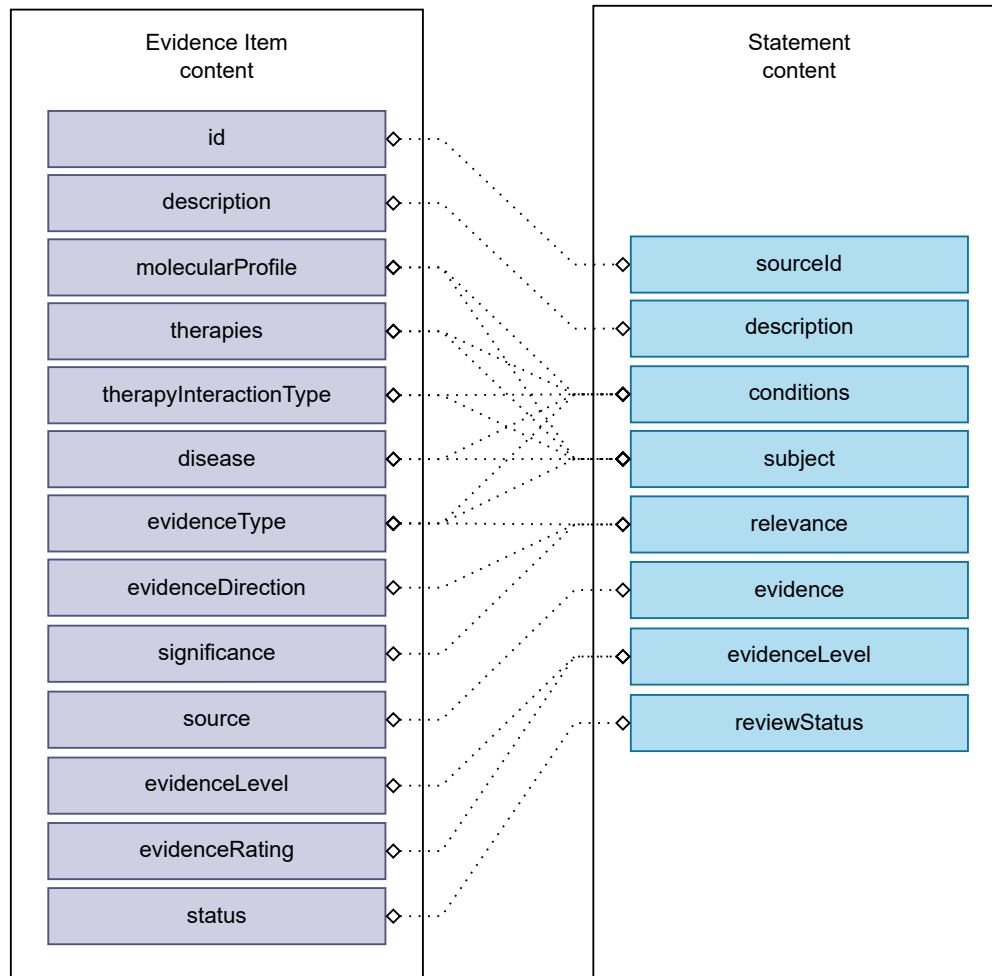
Evidence Item content that can lead to multiple combinations:

- Molecular Profile with explicit OR operator
- Variant with implicit OR in the name
- List of therapies used as substitutes (implicit OR)

C

Processing combinations into Statement contents

In order to create a new GraphKB Statement out of a CIViC Evidence Item combination, each piece of statement content needs to be generated from one or more piece of relevant content in CIViC. Below is a schema that shows how pieces of content are related to each others.



Content common to all combinations gets generated first:
(actually happens before splitting into combinations).

- **sourceId**; from **id**
- **description**; from **description**
- **relevance**; from **evidenceType**, **evidenceDirection** & **significance**
- **evidence**; from **source**
- **evidenceLevel**; from **evidenceLevel** & **evidenceRating**
- **reviewStatus**; from **status**

Content specific to each combination get generated after splitting:

- **conditions** both generated from **molecularProfile**, **therapies**,
- **subject** **therapyInteractionType**, **disease**, & **evidenceType**.

Processing content from CIViC in order to generate suitable content for GraphKB is in some cases a complex operation that leads to the majority of errors encountered. If a piece of content cannot be processed, then that specific combination is discarded. Even if some combinations get discarded though, the following matching step will still go on so partial success can be achieved.

D

Matching combinations to already existing Statements

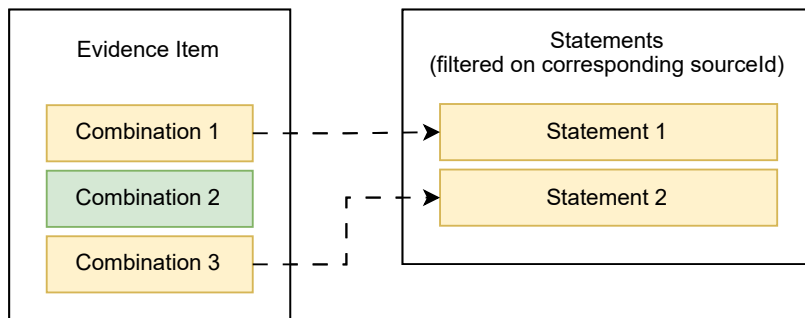
Since we want to avoid deleting and re-creating all CIViC-related GraphKB statements everytime the loader is ran, each combination content has to be matched to its corresponding Statement. That can only be done with successfully processed combinations. Each match gets then directed toward the appropriate CRUD operation.

Matching criteria:

1. First on both conditions and subject
2. Then on subject only
3. Finally at random (until unmatched elements only on one side, or none)

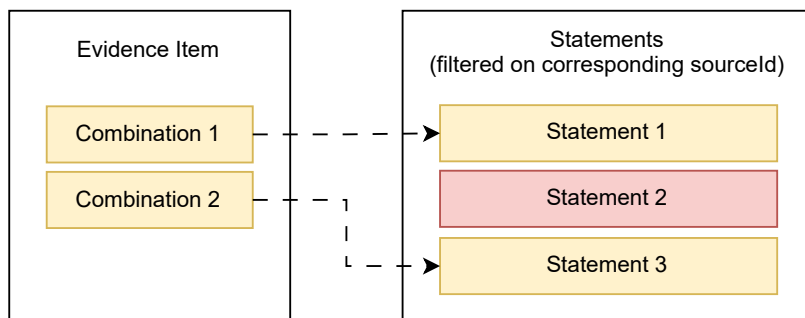
CREATE

Example where a new GraphKB Statement needs to be created based on a novel non-matching combination



DELETE

Example where a deprecated GraphKB Statement needs to be deleted since it can't be matched to any combination anymore



** In cases where some combination(s) were discarded due to content processing errors, unmatched Statement(s) won't be deleted and instead will be logged for further investigation. Among other things, a processing error can occur when a third party resources (e.g. NCBI's Entrez) get irresponsive, which often can be fixed by re-running the loader again.

UPDATE

In previous examples, two GraphKB Statements also needed to be evaluated for update. Will be done by comparing their entire content for an exact match.

