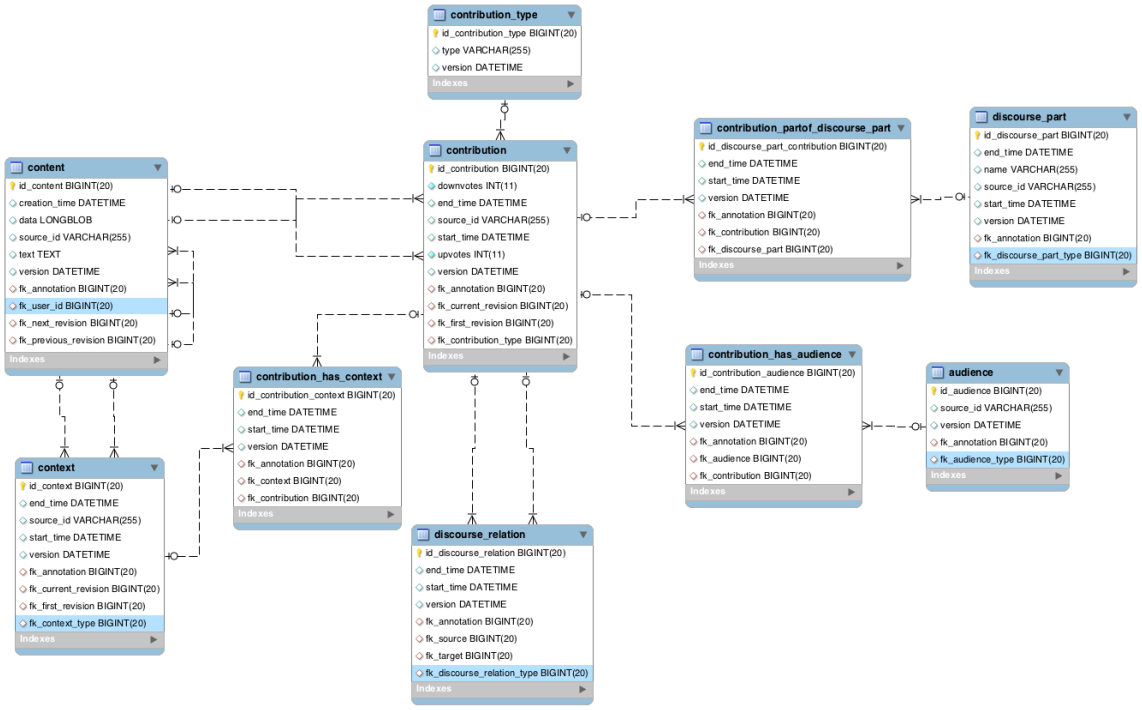
Informal description of the DiscourseDB data model

Chris Bogart, version 1, Oct 14, 2015

Contributions

A *contribution* is a textual snippet found in some online space. A *contribution* may consist of one or more *content* items – these are revisions that may be made over time, so the *contribution* points to the first and last in a linked list of revisions. Each *content* item may be by a different *user* (not shown here), reflecting the possibility that users might edit each others' *contributions*. The *content* table is what physically holds the text.



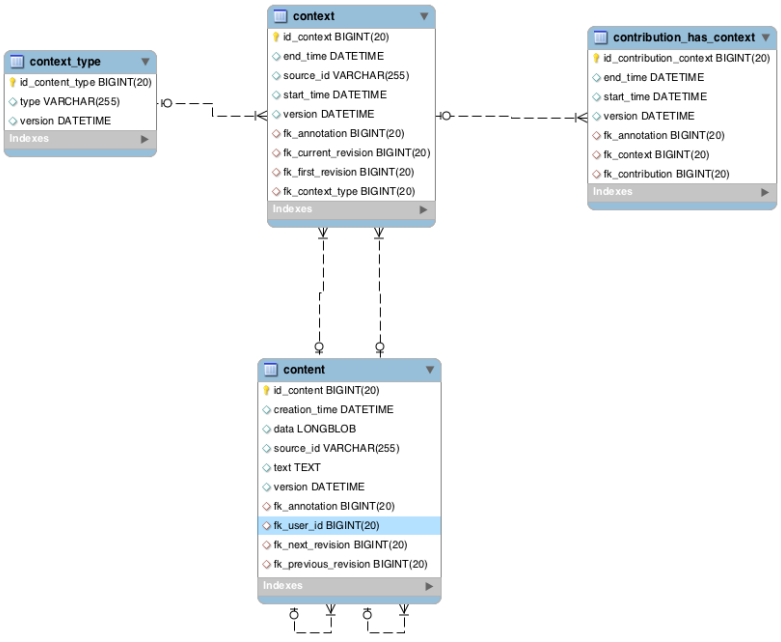
A *discourse_part* is a somewhat coherent *conversation* made of *contributions* (and *contexts*; see below) by people who are able to see and respond to each other's *contributions*. This could be, for example, a forum or subforum, a thread of comments at the bottom of a blog post, a log of an IRC channel, or the code review comments on a Github pull request. Each *discourse_part* is labeled with a *discourse_part_type* indicating what kind of infrastructure the conversation is happening within: e.g. FORUM or TWITTER.

Contributions can belong to multiple *discourse_parts* (via the table *contribution_partof_discourse_part*). Maybe, for example, some text is cross-posted to two mailing lists, or maybe we would like to consider the commit comment a developer makes to some code in Github to belong to a *discourse_part* about changes to that code, but also part of an "issue" discussion that prompted the code change.

A *contribution* also has a *contribution_type* that indicates its role within the *discourse_part*, like "THREAD_STARTER" or "POST". *Contributions* can be associated with each other via a *discourse_relation*: for example "REPLY" or "DESCENDENT". The difference between a *contribution_type* and *discourse_relation* is that a *contribution_type* pertains to a single *contribution*, and a *discourse_relation* ties together two *contributions*.

Each *contribution* may have one or more *audience* entries (linked by the table *contribution_has_audience*): these describe who can read the *contribution,* or perhaps who is likely to read it. This could be a specific list of people, for a comment restricted to a group of known usernames. It could also be a general category like "public"; for example a tweet can be read by everyone. It could capture likely or intended audiences as well: for example a tweet reply is likely to be seen by the original tweet's author, and also by the two authors' followers, but is also public – the existence of these three groups can all be encoded as *audience* entries.

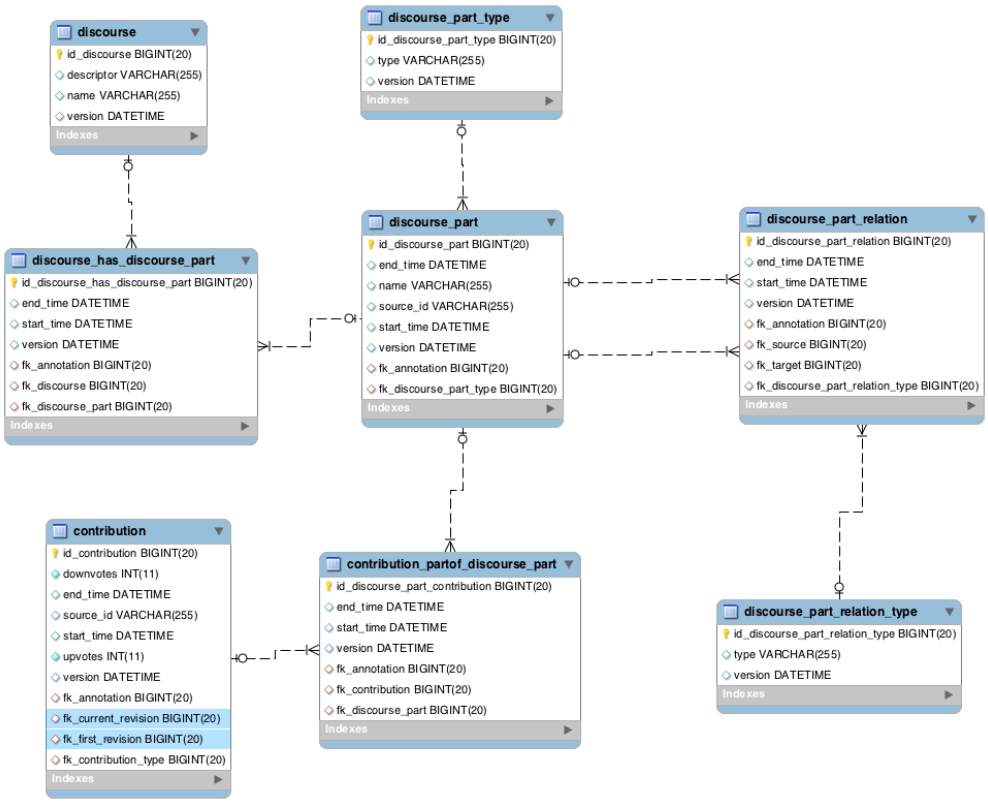
Context



Context is whatever the *contributions'* text is referring to. For example if the *contributions* are comments on a Wikipedia talk page, then *context* might be the wiki

page itself. *Contexts* are associated with particular *contributions* via the *contribution_has_context* table. *Context* has not been used yet in any application, so it's not clear yet exactly how it will be used. Like *contributions*, the actual text of *context* items is held in another table, *content*, in order to capture revisions.

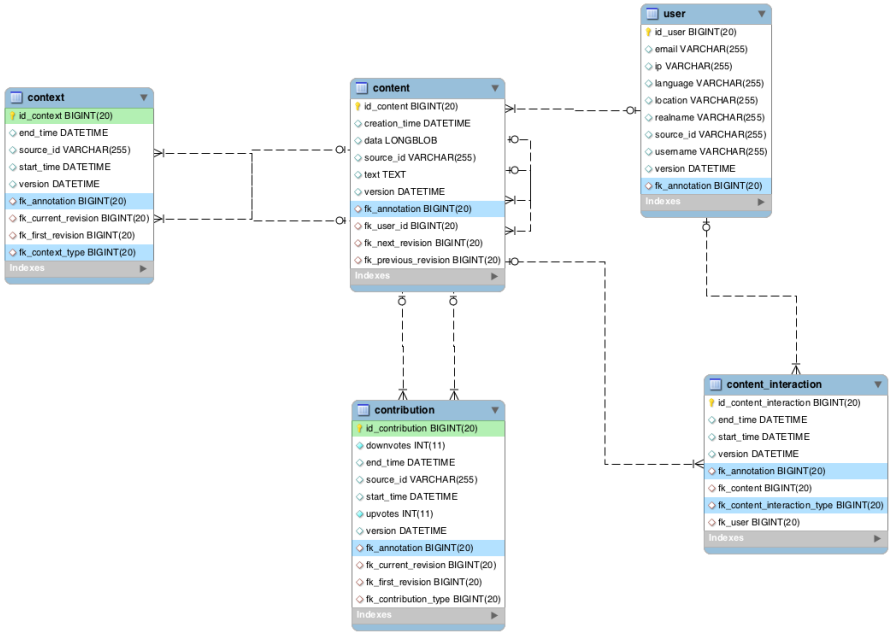
Discourse Part



Discourse_parts contain *contributions* and *contexts,* and can have a variety of relationships with each other; for example one might be a subset of the other, like a forum and a subforum; or might relate to each other in some way, e.g. a comment thread on a blog post about a Github issue. *Discourse_part_relation* captures these relationships, and *discourse_part_relation_type* allows for different kinds of relationships between them.

Discourse_parts can also belong to different discourses, linked by the *discourse_has_discourse_part* table. If Github projects were modeled as distinct *discourses*, the discussion surrounding a pull request might be linked to both the source and destination *discourses* for the request, since it could be considered part of both projects.

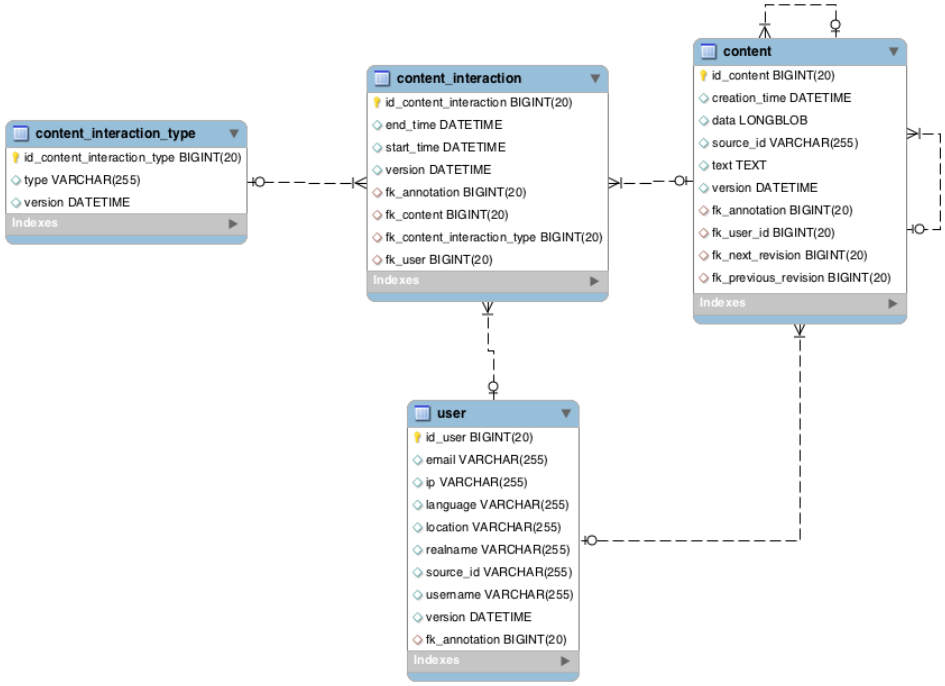
Content



Content records contain the actual snippets of text that constitute different revisions of a *contribution*. *Content* is authored by a *user*, and users can also interact with them in other ways (see *content_interaction* below).

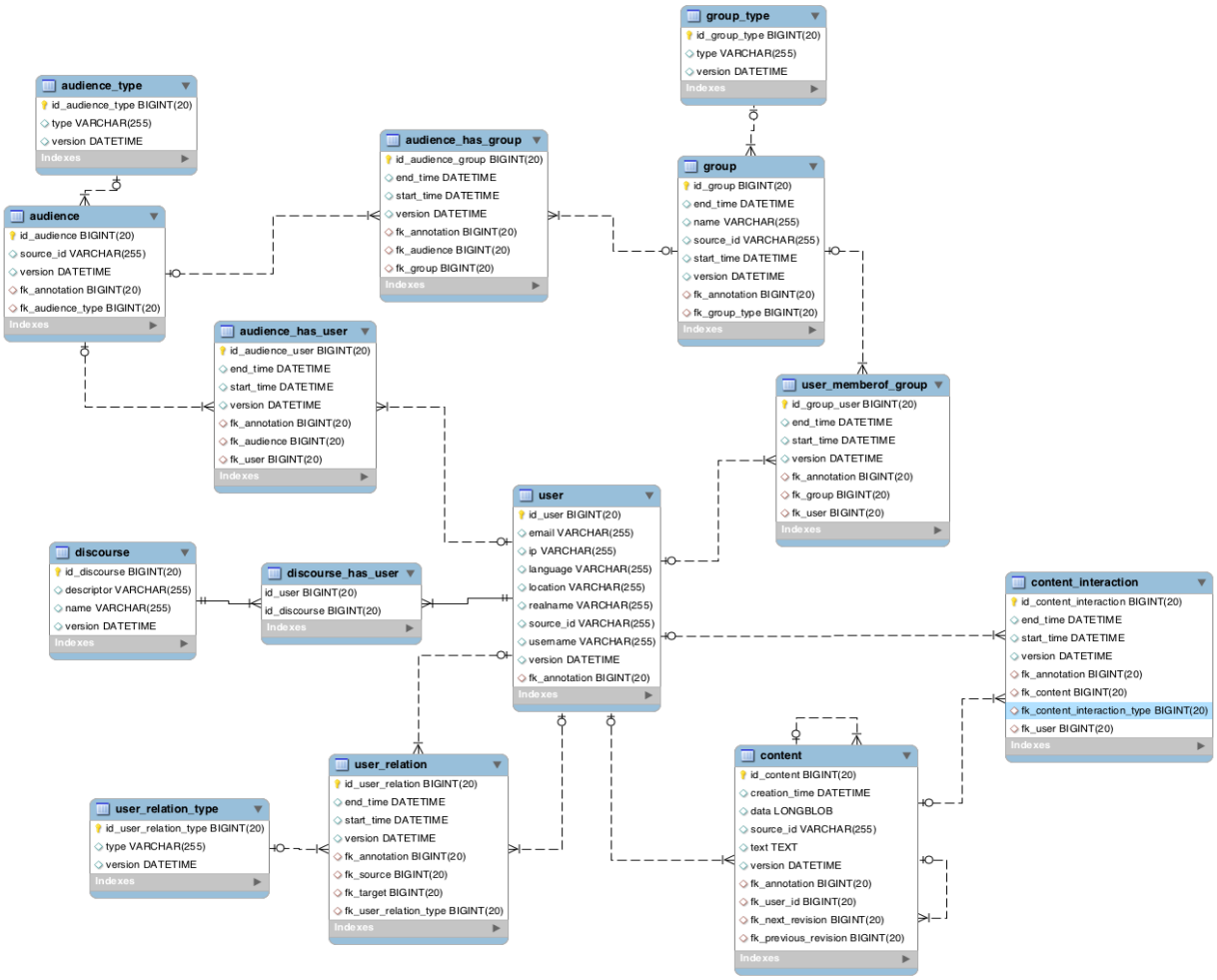
Note that the *contribution* and *context* tables both have a similar relationship with *content*: *content* holds the individual revisions of the data, while *contribution* and *context* are the bit of conversation or artifact, respectively, that persists and might be edited.

Content Interaction



Content_interactions are the ways users can interact with content: i.e. by reading it, editing it, deleting it, voting for it, etc. *content_interaction_type* distinguishes the different kind of action people can take.

User, Audience, Group

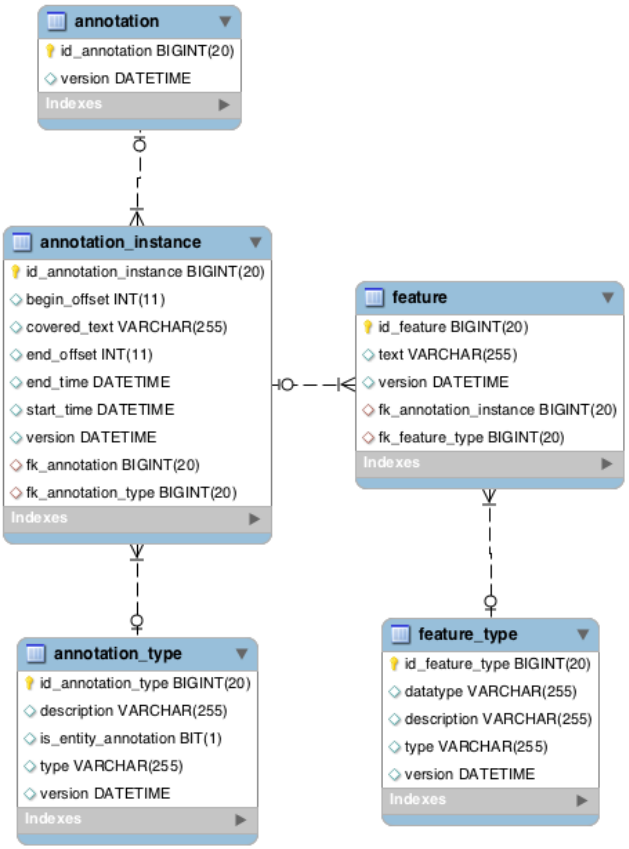


Users may exist across multiple *discourses*: for example the same *user* might take more than one MOOC class, each represented by a different *discourse*.

Users may form *groups*: these are formal categories like student, teacher, admin, or in-class team.

Audience was mentioned above; *audiences* may contain *users*, but also might just be named placeholders like “PUBLIC” representing categories of people or intended kind of audience. *Audiences* may include *groups*: for example a thread might be visible only to students within a single team, as well as administrators.

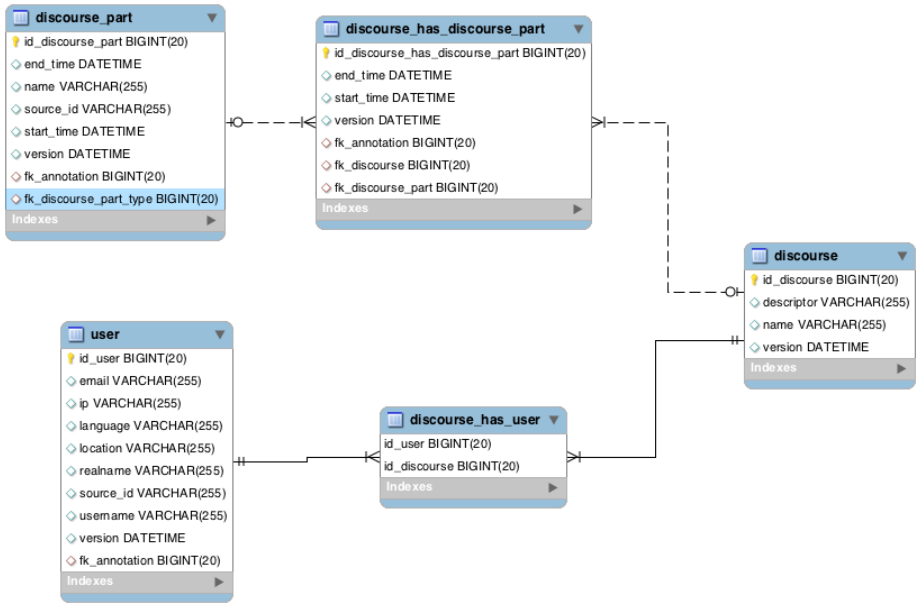
Annotations and Features



Annotations attach to almost every entity in the database; they're a general purpose way of tagging and labeling entities. Most tables have an annotation_id referring to the *annotation* table; that's tied to a set of *annotation_instance* entries, which can be of various *annotation_types*, and have various *features* associated with them. The the *annotation_instance* table has start and end time and offset values allowing regions within text to be labeled (if this is the case, then the *annotation_type*'s is_entity_annotation field must be set to TRUE).

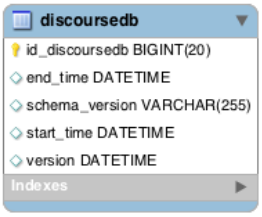
For example: a part of speech tagger might tie each *content* row to an *annotation,* then create an *annotation_instance* entry for each word in the *content*'s body, pointing to *annotation_type* with value WORD. Each of the *annotation_instances* could have start and end offset values showing where the word starts and ends in the *content* body field. A *feature* row could label each *annotation_instance* with the name of the part of speech (e.g. NOUN, PREPOSITION, etc), with its fk_feature_type field pointing to a row in *feature_type* saying "POS".

Discourse



These tables were mentioned above, but are included for completeness. There is a many-to-many relationship between *discourses* and *discourse_parts*, and between *discourses* and *users*. *Discourse_has_user* should associate a set of users with each discourse that at least includes all the users mentioned in the *groups, audiences, contributions, content,* and *contexts* associated with the *discourse.*

DiscourseDB



The *DiscourseDB* table is basic bookkeeping information about the database as a whole. *Annotation_instance* refers to the schema_version