Course
- id
- title
- period
- description

Entry
- id
- course_id
- type

entries

incorrect
correct
incorrect
correct

n, n, n, n, n, n,

entries

type=paragraph
type=paragraph
type=paragraph

type=unit

type=section

«Entry»
Unit
- id : FK @entry
- section_id
- full_title
- breadcrumb_title
- next_unit_id
- prev_unit_id

prev
next
0..1 0..1 1
1
1
1

1
n
1

«Entry»
Section
- id : FK @entry
- title
- description
- parent_entries_id

«Entry»
Paragraph
- id : @entry
- number
- title
- text
- paragraph_type

«Entry»
YesNoExercise
- title
- text
- answer : boolean
- blocks : boolean

1 1

«Entry»
MultiAnswerExercise
- title
- text
- blocks : boolean

1 1

answer_candidates

AnswerCandidate
- id
- text
- key : boolean

The key determines
whether this candidate's
text represents a correct
solution.

Note

I want to point out a few things in this design:

1. The entry relation does not have a reference to the course,
   the reason is, that both paragraphs and units must be contained
   in a section. So only sections are connected with a course.
2. Sections can be recursively nested through their
   parent_section_id
3. I have consolidated paragraphs, definitions, examples, theorems,
   etc. into a single Paragrahp relation that is discriminated by
   a paragraph_type field. The tag can be used as a short
   keyword to describe the element. e.g. in "Definition 1 (Alphabet)",
   "Alphabet" would be the tag.
4. The entry relation is total and disjoint.