

SeSQL data model guide

General

SeSQL data model is described in a 'config.py' file.

It should define three variables :

FIELDS A list (or tuple) of fields.

MASTER_TABLE_NAME The name of the master table to use.

TYPE_MAP Association between Django types and tables.

Concepts

SeSQL data model is composed of *fields*. A field is something on which you can perform queries and order the data, it's similar to a Django field.

Fields are computed from *sources*. A source will fetch the values to compose a field, taking object attributes, concatenating several of them, calling methods, or following relations.

Fields

Each field has a type, a name, a source and may have some type dependant options.

Known types are :

IntField Normal integer field.

StrField Normal string field, up to 255 characters wide, "size" can be specified.

ClassField Handle the class of the object.

DateField Date field, without time.

DateTimeField Date field, with time.

IntArrayField Field storing multi-valued integers.

FullTextField The beast for which !SeSQL was designed, a full-text index, can be made **primary** to be used in rankings.

FIXME : give example and specify options

Sources

Each field requires a 'source'. The source can be one of the following classes :

SimpleField Will just fetch an attribute from the Django object, can specify a condition (Q object) to filter on.

MethodCaller Will call a method from the Django object.

SubField Will walk accross one or several many-to-many mappings, fetching attributes of the related objects.

TextAggregate Concatenate the result of other sources.

Easy writing of sources

Source can also be given in a more friendly way :

- as a normal string for a **SimpleField** (ie, `workflow_state`) ;
- as a normal string terminated with `()` for a **MethodCaller** (ie, `"getFullName()"`) ;
- as a path separated by `.` for a **SubField** (ie, `".authors.firstname"`) ;
- as a list or tuple for a **TextAggregate** (ie (`"firstname", "lastname"`)).

If the source is not specified, it'll be a **SimpleField** of the same name that the index.

Type map

The type map is a list (or tuple) of `(class, table_name)`. All Django objects of this class will be indexed into the given table. All objects of a subclass too, unless the subclass has a specific entry in the mapping.

Example

```
TYPE_MAP = ((models.Photo, "sesql_photo"),
             (models.Comment, "sesql_comment"),
             (models.BaseModel, "sesql_default"))
```