The ModERN Database: White Paper

Úna Faller, Valentina Fedchenko, Dario Maria Nicolosi, Glenn Roe

Introduction

This document seeks to report on the state of the database of the ERC project ModERN as of August 2024. It will outline the structure of the database and provide an explanation of each table and its columns. The aim of this paper is also to provide documentation for current and incoming team members of the characteristics of the current database and to facilitate any future changes or modifications to its structure.

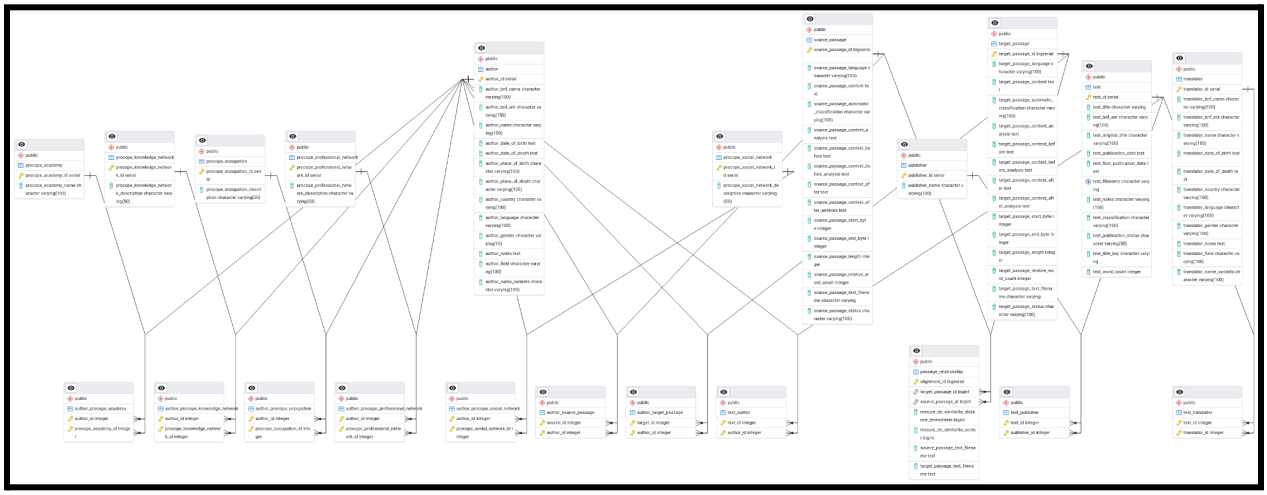

Fig.1 : Entity Relationship Diagram (ERD) generated by pgAdmin4

Database Structure

The decision to use postgres[1] as the database management system was taken in November 2023 after consultation with the team and an evaluation of the project's needs.

The database currently consists of 22 tables, of which 10 are 'bridge tables', designed to manage the many-to-many relationships between the data entities. Each table will be discussed in the following section.

Some repeated features across the tables should be addressed in order to better understand the overall structure of the database. As is common in relational databases, the uniqueness of each row added to the database is guaranteed by an auto-increment primary key, called 'tablename_id' for each table. This is an integer that is also used as a foreign key to connect tables. Tables with similar functions, such as `author` and `translator` have the same structure, as do `source_passage` and `target_passage`. This creates a more homogeneous database and facilitates data insertion. There are 5 tables that share the same column structure along with their relevant bridge tables.

[1] https://www.postgresql.org/

1. Main tables

a. author

This table contains 14 columns relating to the metadata of the authors in our corpus. The design of the database allows for the majority of these columns to remain empty, to reflect the nature of historical research, where some of this information may not be readily available, particularly in the case of anonymous authors. This table allows for open data to be implemented in the database, as we have added external identifiers, namely unique identifiers from the National Library of France. The column 'author_gender' will allow for future study into the composition of the corpus and will facilitate research into the role women played in contributing to Enlightenment ideas and knowledge.

There is a series of bridge tables that are relevant to this table; these will be discussed in their own section.

b. text

This table contains 12 columns and contains metadata relating to texts in our corpus. This table shares some similarities with `author,` such as empty columns for data that has not yet been collected. We have also opted to collect external identifiers from the National Library of France. One important aspect of this table is the columns 'text_publication_date' and 'text_first_publication_date'. This refers to the fact that often a text may be used as a later edition, while the original date of publication may be much further in the past. This distinction was necessary, as we were originally confronted with the fact that a textual reuse may have otherwise been improbably found. For example, a Classical text in an 18th century edition being identified with a text published before the publication of the later edition would look like the Classical author used text that was published centuries afterwards. Hence the use of 'text_first_publication_date' removes this ambiguity.

Another important column from this table is ' text_word_count'. Collecting the data for this figure allows us to compare the word count of source and target passages to facilitate initial data sorting.

c. publisher

This table holds the data of the publishers of texts in the database. As this information is not homogenous for texts of this period, the column 'publisher_name' does not have a NOT NULL constraint. This table is connected to `author` via a bridge table that will be discussed in a separate section.

d. translator

This table mirrors the structure of `author` with 12 columns. Information on a text's translator(s) includes an identifier from the National Library of France. This table plays an important role in the database and in managing the coherency of the source and target passages.

For example, a translated text may have sections added in by the translator, such as appendices, from where a source or target passage may be drawn and then would then need to be addressed.

e. source_passage

This table contains the data that relates to the identified string of characters in a text that has a corresponding target in another text. It currently contains 14 columns. Unlike other tables, most of the columns in this table have a NOT NULL constraint, as there is certain data that a source passage must have associated with it, for example, the start and end byte in the related parent text file. This table is joined to the table `text` via 'source_passage_text_filename', as this information is unique to each text in the database. This means a row in `source_passage` inherits the data from the corresponding row with the same filename, which in turn is connected to `author` via the bridge table `text_author` (discussed in more detail below).

Other important columns in this table include 'source_passage_context_before' and 'source_passage_context_after'. These columns record the textual content that comes before and after the extracted source passage string, which allows us to identify any passages that may not be useful to the project and are to be used in the interface currently under construction.

f. target_passage

This table has the same columns and datatypes as `source_passage` and records the data of strings identified as having appeared in a similar form in another text published prior to the target passage's parent text publication date.

g. passage_relationship

This table manages the many-to-many relationships between source and target passages. It draws from both tables separately to allow for cases where a source can have more than one target using a foreign key constraint with 'source_passage_id' and 'target_passage_id'. The uniqueness of a particular relationship is maintained by the primary key 'alignment_id'. Two other columns that are important in this table include 'mesure_de_similarite_vector' and 'mesure_de_similarite_distance_levenshtein'. This data allows us to compare the strings of passages that to a human, are identical, but have been classified by the machine as being separate strings often due to poor optical character recognition, or misplaced commas and other punctuation signs.

2. Procope classification tables

These tables merit their own section, as there are 5 tables that have the same purpose and structure. It was decided to enhance the author data with external data, namely the Metadata Schema Procope[2], an attempt to propose a standardised classification for Englightenment authors. These tables thus contain 2 columns, an ID that serves as the primary key, and the

[2] As proposed in M. T. Comsa, M. Conroy, D. Edelstein, C. Summers, E. and C. Willan, "The French Enlightenment Network", in The Journal of Modern History, September 2016, Vol. 88, No. 3, p. 495-534

description of the classification. For example, there are currently 15 potential professional networks in the table `procope_professional_network`, with this data being housed in the column 'procope_professional_network_description'. These tables are connected to the `author` table via bridge tables, as an author can belong to more than one network and a network can be applied to more than one author. Currently the procope classification tables do not contain data as of August 2024, and this thus remains a task to be completed.

3. Bridge tables

There are 10 bridge tables in the database to manage many-to-many relationships between entities. Eight of these are related to `author` and two to `text`.

a. Author bridge tables

As discussed above, the five procope tables relate to external classifications of Enlightenment authors. The 'author_id' is used as a foreign key in each of the five bridge tables along with the corresponding id from the procope table. This creates a composite primary key that maintains the relationship between the data.

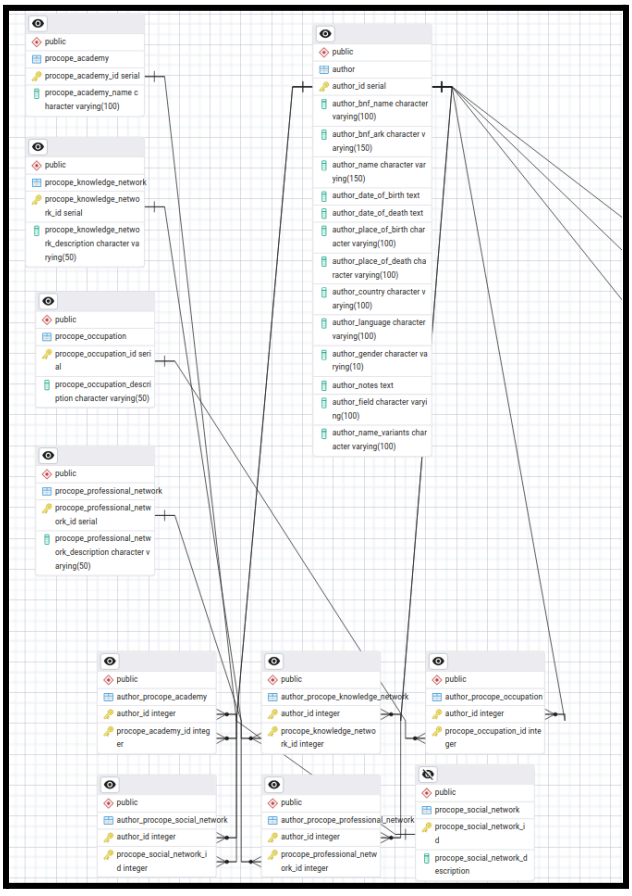

Fig.2 ERD for bridge tables for procope tables and `author`

The remaining author-related bridge tables relate to source_passage, target_passage and text. Firstly, the many-to-many relationship between `author` and `text` is managed via `text_author`. This table has 2 columns, 'author_id' and 'text_id' which are foreign keys drawn from the respective parent table. This structure allows for a text to have many authors, and an author to have written many texts, which we anticipate will be the case for many data entries.

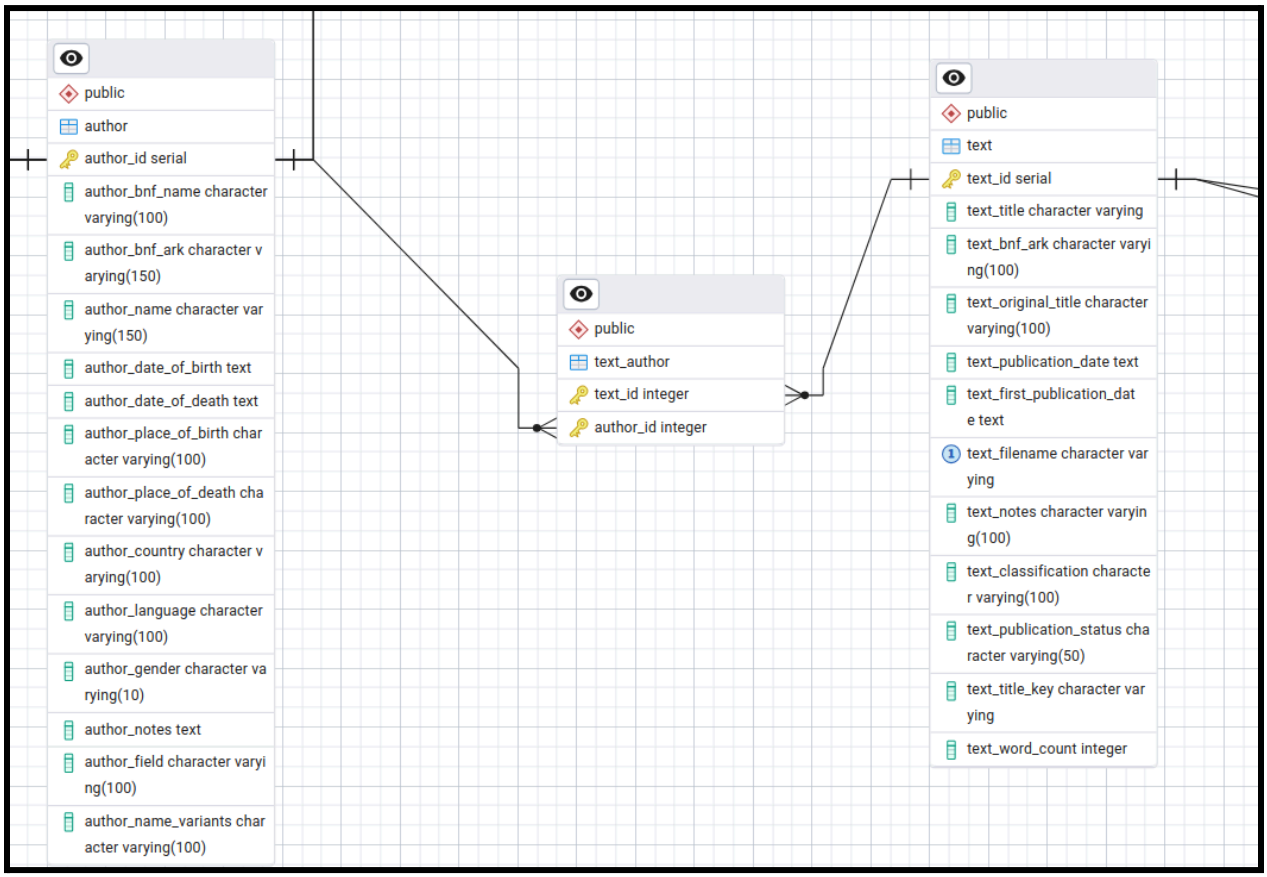

Fig.3 ERD for bridge table showing the relationship between `text`, `author` and `text_author`

Secondly, it was also decided to add two further bridge tables that connect `author` with `source_passage` and `target_passage`. While it is assumed that for the majority of cases the author of the text will be the default author of the passage under consideration, there will potentially be cases where this is not the case, such as the case of translators (as discussed above in section 1.d.

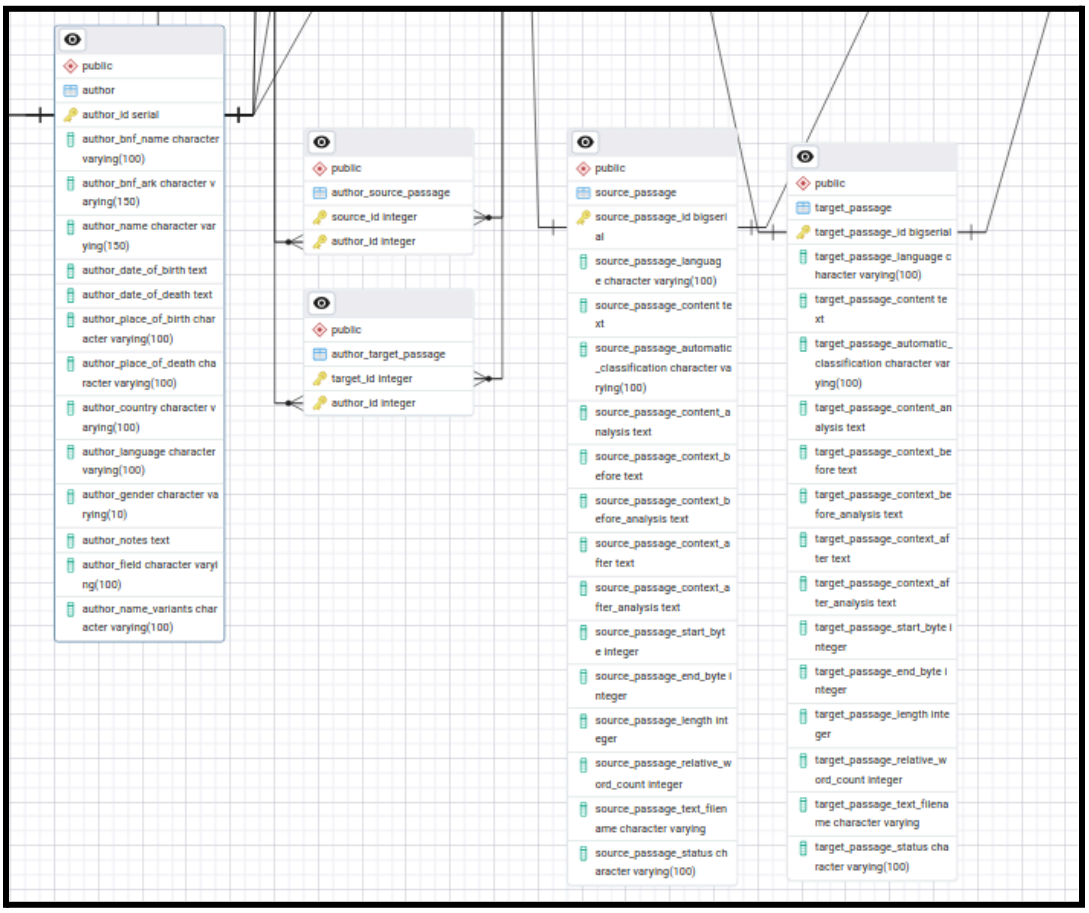

Fig.4 ERD for bridge tables involving `author`, `source_passage` and `target_passage`

b. Text bridge tables

These two tables concern the information relating to the translator and publisher of a text. They share a similar structure to other bridge tables in this database and they maintain potential many-to-many relationships, as a text in our database may have more than one translator, and often a translator is responsible for multiple texts, for example. The two columns are filled from `text` and `translator/publisher`.

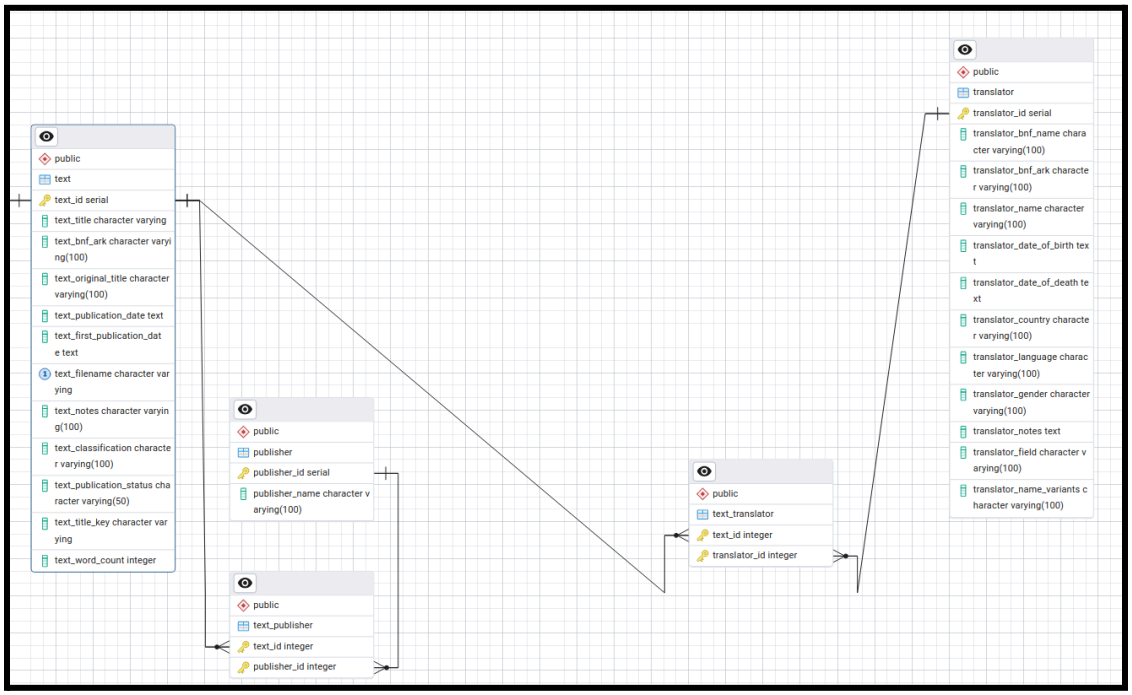

Fig.4 Bridge tables related to `publisher`, `translator` and `text`

Future directions

Some tables are empty as of August 2024; namely the procope tables. This data is to be cleaned before insertion. There are also issues that remain to be discussed and implemented. For one, a potential improvement would be assigning generated ids to 'source_passage' and 'target_passage *before* insertion into the database, as this would facilitate the filling of the table `passage_relationship` because the data will have already been rendered unique.

The need for the table `publisher` could also be discussed, it offers less insight as `translator` and the data is far from homogenous and available across the board. The rationale behind the bridge tables `author_source_passage` and `author_target_passage` also remains to be clarified and tested. Finally, the datatypes for data such as year of publication, birth death etc have been entered as TEXT and not DATE. This may cause problems in the future and may need to be rectified and data cleaning processes may need to be carried out.