U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

(PRI - M2)

André Lima - 202008169
Guilherme Almeida - 202006137
Jorge Sousa - 202006140
José Castro - 202006963

Group 75

Introduction - **What is it?** (Recap)

- **Lyric-based** Search Engine for **finding songs**
- Aggregates **various metadata** regarding musical content
- Aims to:
  - Allow **text-based querying** with the aid of context-sensitive **filters**
  - Observe a specific song's **composition** (both **lyrically** and **structurally**)

Introduction - **Why do it?** (Recap)

Lyrical and structural musical data is **vast but mostly unexplored**

Centralized and rich data sources for our data collection

In its final stage, it will be a **fun and interactive way** to interact and **perceive music production and its history**

Domain Model (Recap)

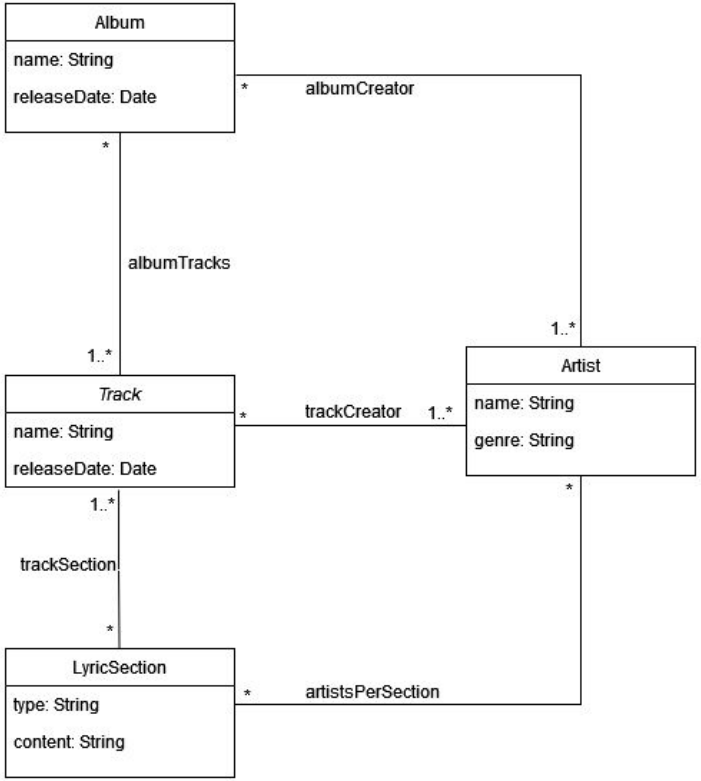

Fig 1: Verse Vault's Domain Model

Group 75

Tooling

Group 75

Search Scenarios

We have **four search scenarios**:

- I want to find a track talking about a **specific subject** inside a **particular structural section**.
- I want to find a song talking about a **given topic**.
- I want to find a song of a **given genre** that talks about a **specific topic**.
- I want to find a track that talks about a **given topic** even if the **spelling is wrong**.

Indexing - **Indexed Fields**

Field Name	Type	Indexed
_id	string	false
name	basic_text_t	true
duration	plong	true
url	string	false
artist	artist_t	true
publishedAt	pdate	true
genres	basic_text_t	true
lyrics.title	lyrics_title_t	true
lyrics.content	text_general	false
lyrics.content_en	content_en_t	true
lyrics.content_es	content_es_t	true
lyrics.content_phonetic_en	content_phonetic_en_t	true
lyrics.content_phonetic_es	content_phonetic_es_t	true
album.name	text_general	true
album.image	string	false
entities.text	basic_text_t	true
entities.start	plong	false
entities.end	plong	false
entities.type	string	false

Table 1: Fields defined in the refined schema

Indexing - Indexed Analyzers

- Tokenizers:
  - Standard - track name, lyrics section content, album name, etc.
  - Letter - artist (e.g. “L.M.F.A.O.” matches with “L M F A O”)
  - Whitespace - lyrics section title (e.g. “Pre-Chorus” -> “PreChorus”)
- Filters:
  - **Lowercase**
  - **Pattern Replace**
  - **ASCII Folding** - useful to convert UTF-8 characters to ASCII
  - **Keyword Repeat** - useful to preserve non-stemmed data
  - **Porter Stem** - english language stemmer
  - **Spanish Light Stem** - spanish language stemmer
  - **Remove Duplicates**
  - **Beider-Morse Phonetic Matching** - phonetic token matching

Information Need 1

Search scenario: I want to find a track talking about something inside a particular structural section.

Information Need: Find songs that talk about playing the guitar in the chorus.

Query:

```
{!parent which="doc_type:track" filters=

$childfq score=max}{!edismax qf="content_en"}

("play guitar"~5)
```

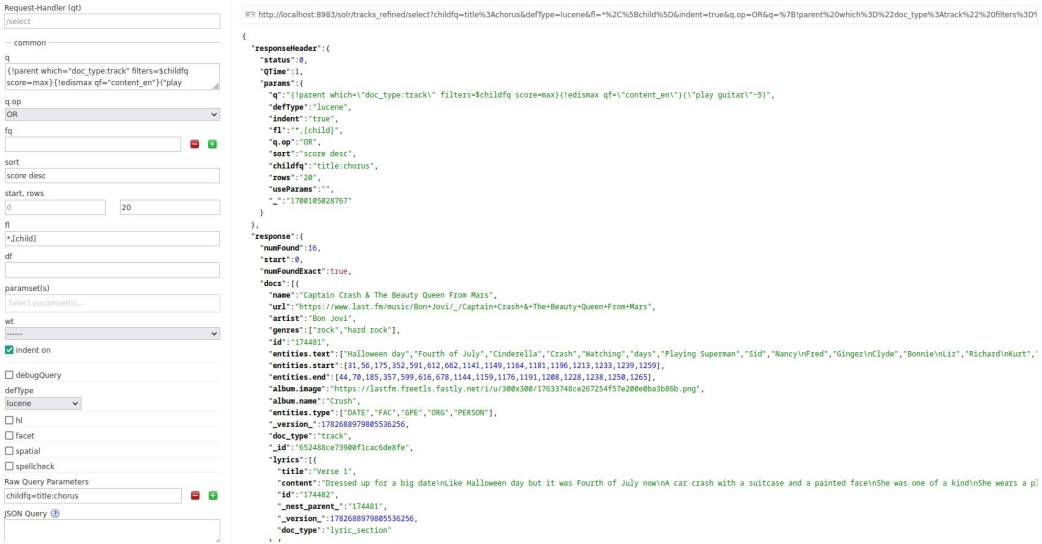

Fig 2: Information Need 1 - Query result (improved query)

Information Need 1 - **Evaluation**

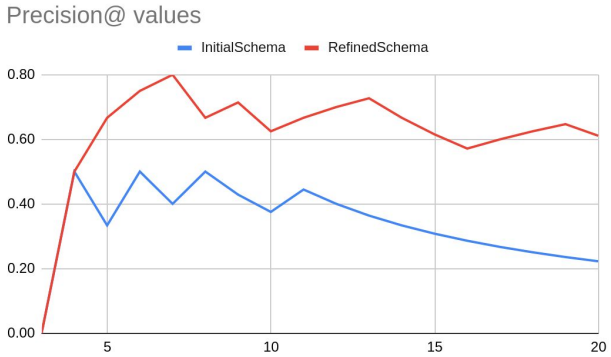

Fig 3: Information Need 1 - Precision@ Values

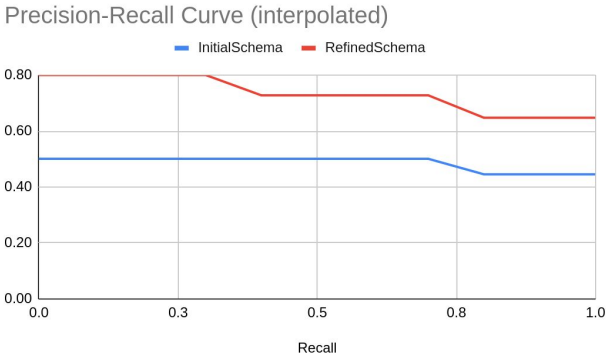

Fig 4: Information Need 1 - P-R Curve

Group 75

Information Need 2

Search scenario: I want to find a song talking about a given topic.

Information Need: Find songs that talk about slavery in Africa.

Query:

```
{!parent which="doc_type:track" score=total}

{!edismax qf="content_en"}(+(slave apartheid^6)

+africa)
```

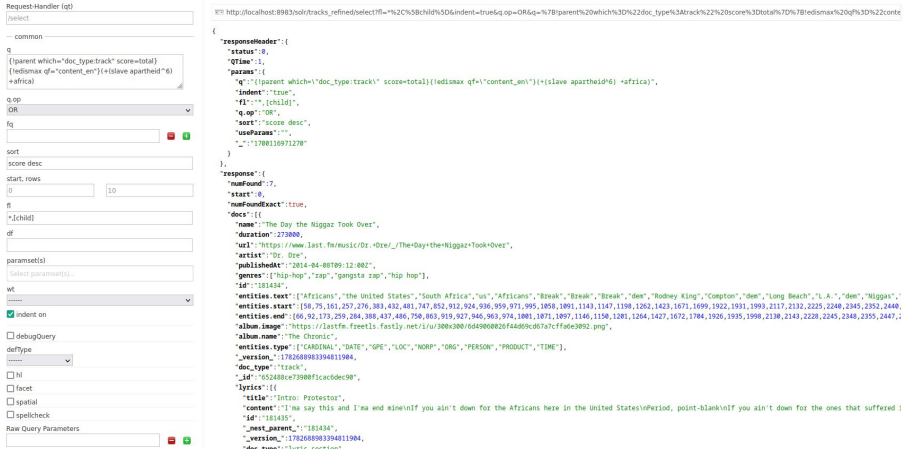

Fig 5: Information Need 2 - Query result (improved query)

Information Need 2 - **Evaluation**

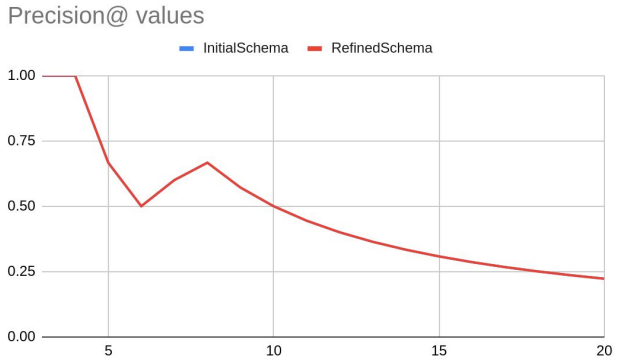

Fig 6: Information Need 2 - Precision@ Values

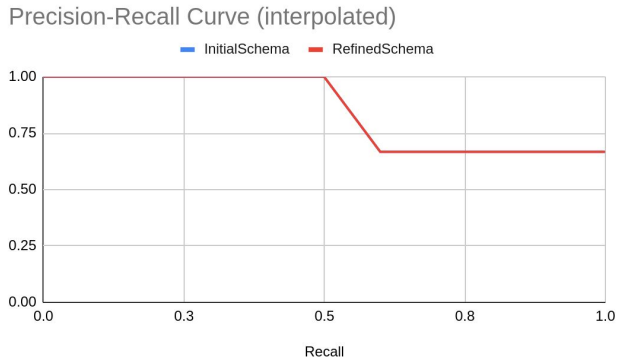

Fig 7: Information Need 2 - P-R Curve

Group 75

Information Need 3

Search scenario: I want to find a song of a given genre that talks about a specific topic.

Information Need: Find rap songs that talk about immigration.

Query:

```
{!parent which="doc_type:track" score=total}

{!edismax qf="content_en"}foreigners,immigration}
```

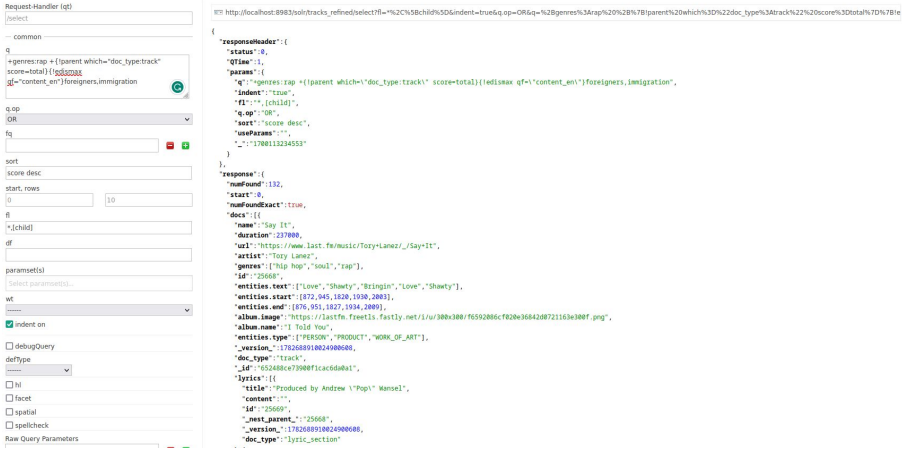

Fig 8: Information Need 3 - Query result (improved query)

Group 75

Information Need 3 - **Evaluation**

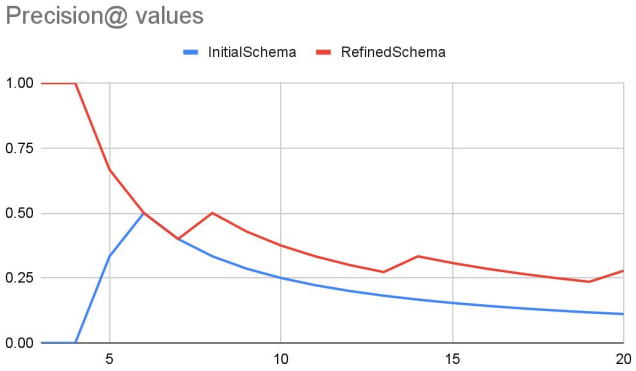

Fig 9: Information Need 3 - Precision@ Values

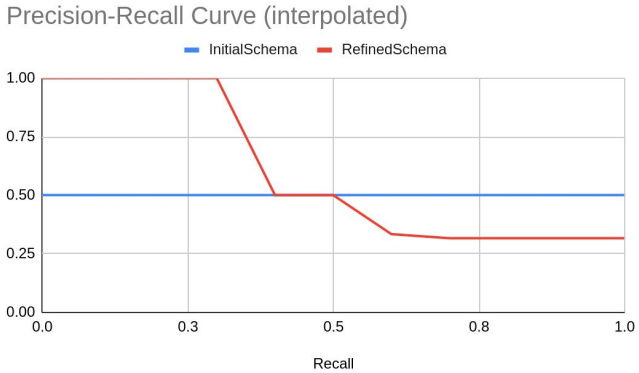

Fig 10: Information Need 3 - P-R Curve

Group 75

Information Need 4

Search scenario: I want to find a track that talks about a given topic even if the spelling is wrong.

Information Need: Find songs about Christmas even if my spelling is wrong.

Query:

```
{!parent which="doc_type:track" filters=$childfq

score=total}

{!edismax bq="+publishedAt:[NOW/DAY-

15YEAR TO NOW/DAY]" qf="content_en content_es"}

(crissmas~2,novidad~1)
```

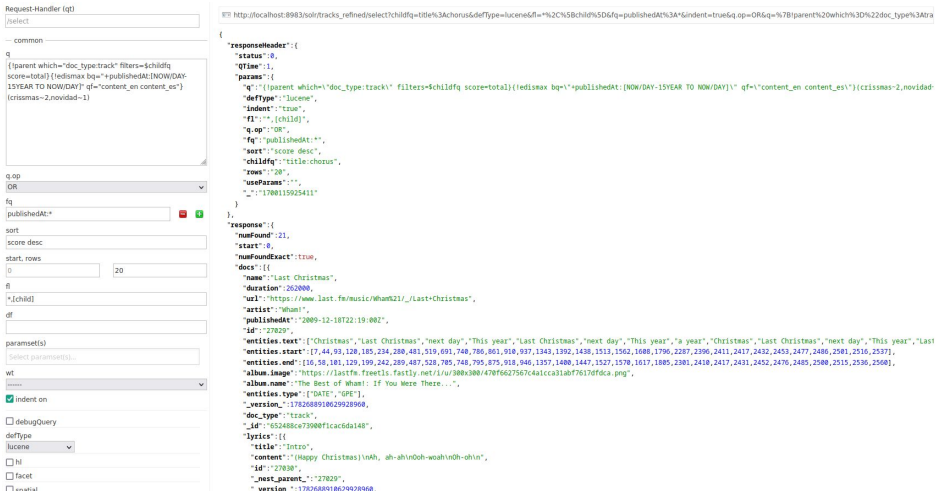

Fig 12: Information Need 4 - Query result (improved query)

Information Need 4 - **Evaluation**

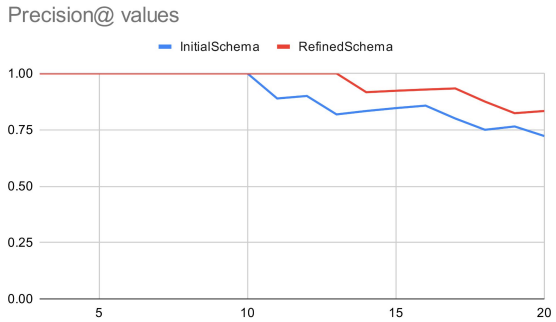

Fig 13: Information Need 4 - Precision@ Values

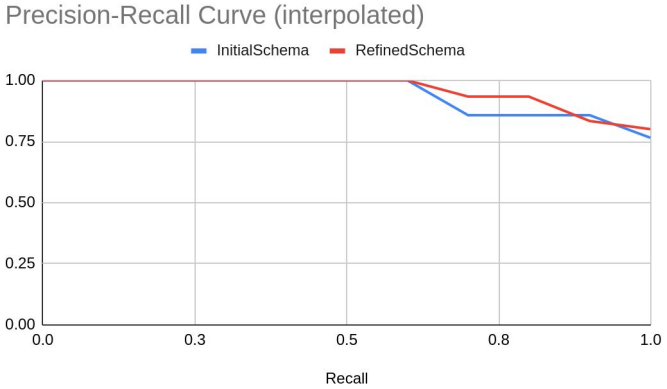

Fig 14: Information Need 4 - P-R Curve

Group 75

Conclusions and **Future Work**

- **Search scenarios** were **covered**
- A **well-defined schema** and **query optimization** are necessary
- Handled **Nested Documents**
- There could be **more Information Needs**
- **Query Optimization** could be **more refined**
- **Develop** a **Frontend** for the end-user
- Include fields from multiple nesting levels for the scoring