FOM Hochschule für Oekonomie & Management

university location Bonn

Bachelor Thesis

in the study course Wirtschaftsinformatik

to obtain the degree of

Bachelor of Science (B.Sc.)

on the subject

Development of a Query Language for Full-Text Search in Relational Databases

by

Sebastian Bunge

Advisor:	Prof. Dr. Peter Steininger
Matriculation Number:	539441
Submission:	August 11, 2022

Contents

List of Figures

List of Tables

List of Abbreviations

HTML	Hypertext Markup Language
MS	Microsoft
PDF	Portable Document Format
SQL	Structured Query Language
XML	Extensible Markup Language

List of Symbols

1 Abstract

Abstract

2 Full-Text Search

Commercial database management has long focused on structured data and the industry requirements have matched those of structured storage applications quite well. The problem is that only a small part of the data stored is completely structured, while most of it is completely unstructured or only semi-structured, in the form of documents, emails, web pages, etc. (cf. Hamilton, Nayak 2001, p. 7)

2.1 MS SQL Server Search Architecture

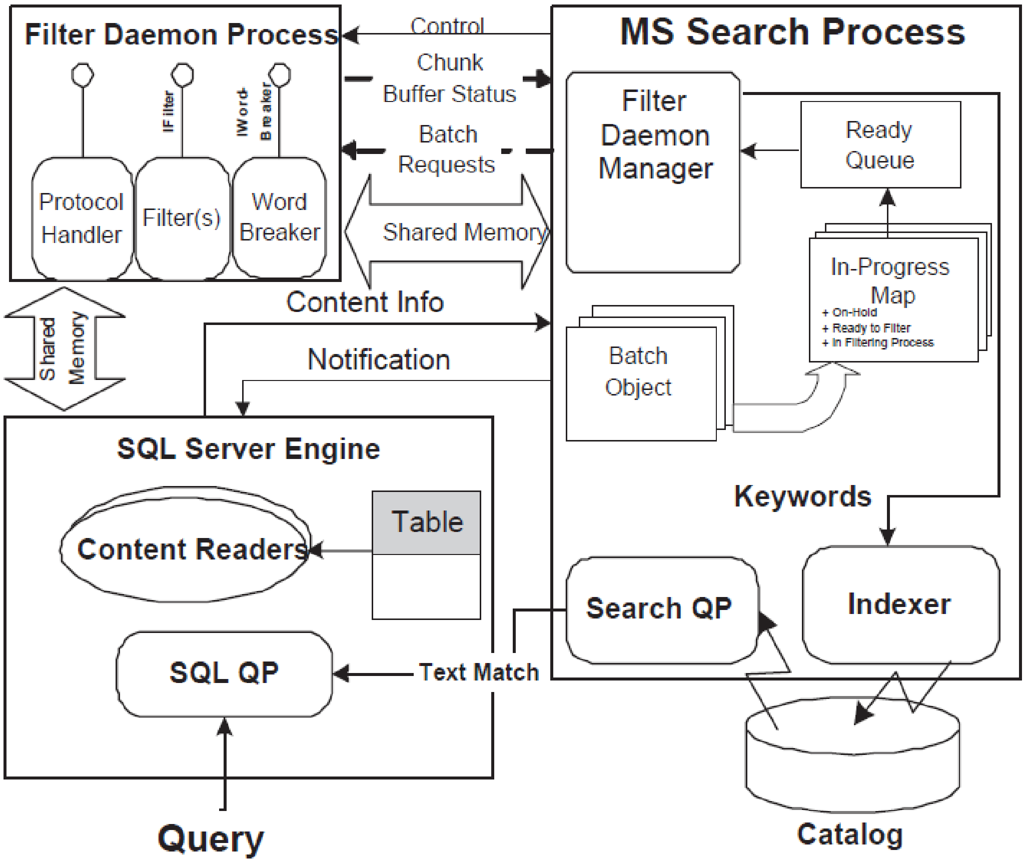
Structured Query Language (SQL) Server uses the same access method and infrastructure for full-text search as other Microsoft (MS) products and the Index Service for file systems. This decision enables standardized semantics for full-text search of data in relational databases, web-hosted data, and data stored in the file system and mail systems. On SQL servers, not only simple strings can be indexed, but also data structures, such as Hypertext Markup Language (HTML) and Extensible Markup Language (XML), and even complex documents, such as Portable Document Format (PDF), Word, PowerPoint, Excel and other custom document formats. (cf. Hamilton, Nayak 2001, p. 7)

The architecture can be divided into five modules, which interact with each other to perform a full-text search. (See Figure 1)

The **content reader** scans indexed data stored in SQL Server tables to assemble data and its associated metadata packets. These packets are then injected into the main search engine, which triggers the search engine filter daemon to consume the data.

Depending on the content, the **filter daemon** calls different filters, which parse the content and output so-called chunks of the processed text. A chunk is a related section with relevant information about this section like the language-id of the text. These chunks are output separately for any properties, which can be elements like the title, an author or other content-specific elements.

Figure 1: Architecture of MS SQL Server Full-Text Search



Source: Hamilton, Nayak 2001, p. 8

Word breakers split the chunks into keywords and additionally provide alternative keywords and the corresponding position in the text. Word breakers can recognize human languages and on MS SQL Server several word breakers for different languages are installed by default. The generated keywords and metadata are passed on to the MS Search process, which processes the data with an indexer.

The **indexer** generates an inverted keyword list with a batch containing all keywords of one or more items. These indexes are compressed to use memory efficiently, this may lead to high costs for updates of these indexes. Therefore a stack of indexes is maintained. New documents first create their small indexes, which are regularly merged into a larger index, which in turn is merged into the base index. This stack can be deeper than three, but the concept remains and allows a strongly compressed index without driving the update costs too high. If a keyword is searched, all indexes are accessed, so the depth should still be kept reasonable.

A **query processor** manages the insertion and merge operations and collects statistics on distribution and frequency for ranking purposes and query execution. (cf. Hamilton, Nayak 2001, pp. 8-9)

2.2 MS SQL Server Full-Text Query Features

"The full text indexes supported by SQL Server are created using the familiar `CREATE INDEX` `SQL DDL` statement. These indexes are fully supported by SQL Server standard utilities, such as backup and restore, and other administrative operations, such as database attach/detach work unchanged in the presence of full text search indexes. [...] Indexes are created and maintained online using one of three options: 1) Full Crawl scans the full table and builds or rebuilds a complete full text index on the indexed columns of the table. This operation proceeds online with utility progress reporting. 2) Incremental Crawl uses a timestamp column on the indexed table to track changes to the indexed content since the last re-index. 3) Change Tracking is used to maintain near real time currency between the full text index and the underlying text data. The SQL Server Query Processor directly tracks changes to the indexed data and this data is applied in near real time to the full text index." (Hamilton, Nayak 2001, p. 9)

3 My Language

My Language

4 Summary

Summary

Appendix

Appendix 1: Appendix

Appendix

Bibliography

HAMILTON, James R.; NAYAK, Tapas K.: Microsoft SQL server full-text search. In: *IEEE Data Eng. Bull.* 24 (2001) Nr. 4. Publisher: Citeseer, pp. 7–10

Declaration in lieu of oath

I hereby declare that I produced the submitted paper with no assistance from any other party and without the use of any unauthorized aids and, in particular, that I have marked as quotations all passages which are reproduced verbatim or near-verbatim from publications. Also, I declare that the submitted print version of this thesis is identical with its digital version. Further, I declare that this thesis has never been submitted before to any examination board in either its present form or in any other similar version. I herewith agree that this thesis may be published. I herewith consent that this thesis may be uploaded to the server of external contractors for the purpose of submitting it to the contractors' plagiarism detection systems. Uploading this thesis for the purpose of submitting it to plagiarism detection systems is not a form of publication.

Bonn, 11.8.2022	S. B
(Location, Date)	(handwritten signature)