

# DBXplorer: Enabling Keyword Search over Relational Databases

Sanjay Agrawal  
Microsoft Research  
sagrawal@microsoft.com

Surajit Chaudhuri  
Microsoft Research  
surajitc@microsoft.com

Gautam Das  
Microsoft Research  
gautamd@microsoft.com

## 1. Motivation

Internet search engines have popularized keyword based search. While relational database systems offer powerful structured query languages such as SQL, there is no support for keyword search over databases. The simplicity of keyword search as a querying paradigm offers compelling values for data exploration. Specifically, keyword search does not require a priori knowledge of the schema. The above is significant as much information in a corporation is increasingly being available at its intranet. However, it is unrealistic to expect users who would browse and query such information to have detailed knowledge of the schema of available databases. Therefore, just as keyword search and classification hierarchies complement each other for document search, keyword search over databases can be effective.

## 2. Overview of DBXplorer

We present DBXplorer, a system that enables keyword search over SQL databases. Technical details of our system may be found in [4]. Given a set of query keywords, our system returns all rows (either from single tables or by joining tables connected by foreign-key joins) such that each row contains all keywords.

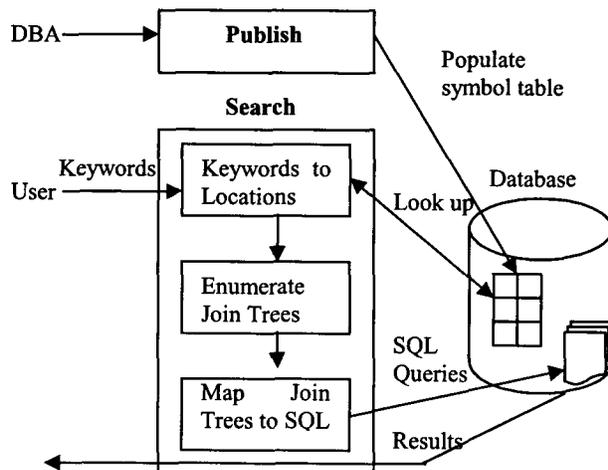


Figure 1. DBXplorer Architecture

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ACM Sigmod '2002, June 4-6, Madison, Wisconsin, USA.

Copyright 2002 ACM 1-58113-497-5/02/06...\$5.00.

Our system has two core components – *publish* and *search*. The *publish* component is a preprocessing step that prepares the database for keyword search by building a *symbol table* that helps map a keyword to a database structure (tables, columns and/or rows). We build a compact symbol table that leverages the physical design of the database to enable fast lookup.

Figure 1 shows the various steps of the search component. Given a set of query keywords, the system first uses the symbol table to determine the locations within the database where the keywords occur. Next, the system uses this location information to enumerate *join trees*. A join tree is a sub-graph of the database schema graph where the tables together contain all the query keywords and the leaf tables match at least one of the user specified keywords. The join trees are mapped to SQL queries that are executed against the database to retrieve results. Thus, a keyword search is mapped to one or more SQL queries. Finally, our system also allows navigation capabilities through the database as a part of the search experience.

The core components of DBXplorer - *publish* and *search* - are packaged as two separate COM (Component Object Model [3]) objects. This packaging allows us to build different applications, e.g. an ASP/web server or C++/VB application, that can invoke *publish* and *search* functionalities.

## 3. What will be demonstrated?

We have built a database search and browsing application on top of *publish* and *search* COM components using Internet Explorer, Active Server Pages (ASP) [2], Microsoft SQL Server, and Internet Information Server [1]. We will show (a) how to *publish* a relational databases using the *publish* interface, (b) how to search over multiple databases and retrieve records within a database that spans all keywords, (d) how to navigate from an answer record to related information within the database, and (e) how to expose similarity searches that exploit domain information in the databases.

## 4. References

[1] Microsoft Internet Information Server.

<http://www.microsoft.com/>.

[2] Active Server Pages. <http://www.microsoft.com/asp/>.

[3] Component Object Model.

<http://www.microsoft.com/com/>.

[4] Agrawal, S., Chaudhuri S., and Das G. DBXplorer: A System for Keyword-Based Search Over Relational Databases. IEEE ICDE, 2002, 5-16.