

# SQLOne

The Zero SQL approach to search Databases

**By**

**Radha Vembu  
Technical Architect**

## Abstract

Data - the backbone of all enterprise applications is often locked up in one or more databases from different database vendors. Looking for particular information across multiple databases can be quite tedious, as users have to be familiar with different query constructs supported by these different database vendors. Users are also expected to know the exact location of the data to be extracted and more detailed information such as database name, table name, and column name that contains the data. The problem gets complex when the tables in the database are normalized, as this will yield only partial result, when the query is made only on the table that contains the search word. In such cases, knowledge about how the different tables are related becomes a necessity so that the different tables can be joined to fetch a meaningful and valuable result.

The practical and workable approach to solve this problem is by dumping the data to text files and using the text-based search engine to fetch the relevant results. This crude alternative approach is not suited for normalized tables, as it cannot exploit any of the special characteristics of the database such as the primary key and foreign key relationship information of the tables. As text-based search engines fail in the primary objective, it is safe to conclude that text-based search engine to fetch data from database is a very poor alternative to fetching the data directly using SQL queries.

The only way to solve this problem is by providing a database specific search engine for any database - SQLOne.

## Introduction

SQLOne offers a zero SQL approach to fetching data across databases by providing a single search interface based on keywords. It not only circumvents the need for the knowledge of SQL, knowledge of the different databases, its tables and how data is stored in them but also generates the queries that can be used for fetching the data. This with the capability to work with multiple databases, each of a different type and the intelligence it has about the tables by analyzing the common keys between different tables adds more power to SQLOne's database search.

SQLOne exploits the special characteristics of the database to fetch intelligent results.

## SQLOne Advantages

### *1. Centralized and Complete View of Data:*

Based on the size of the enterprise different departments may use one or more applications with each accessing the same or multiple databases, where each database could be of a different type. An integrated view of the data will be of a great value in understanding the different aspects of the data you are looking for. Say for example, the HR department maintains all the personal details about an employee in a database and the finance department maintains all the salary details of the employee in another database. By issuing a search with the employee name, you can get all the information about the employee fetched from these different databases.

### *2. View Data with its Relationship:*

Relational principles, such as normalization force applications to store information about a particular subject in multiple tables. While searching for information related to that subject, even if all the search words occur in a single table, fetching the corresponding row(s) from that table alone might give only partial information about the subject. SQLOne fetches not only the row(s) from the table containing the search words, but also provides meaningful information by fetching a cluster of information related to the search result from other related tables also.

### *3. Search words spanning across multiple tables:*

There could be scenarios where search words need not be present in the same table but split across multiple tables, which might contain related information. This information might be linked through a common key (e.g: explicit or implicit primary-foreign key relation), which is smartly identified by SQLOne to fetch the relevant rows across these tables.

### *4. Ranking based on Relevance:*

SQLOne helps users access the searched data quickly; SQLOne ranks the search results to present the most relevant results on the top of the search results page.

### *5. Query generation based on search words:*

Generally the SQL query to fetch a particular data, which has been searched for, might be very useful to the user. SQLOne creates all possible queries to fetch the data containing the search words. The users can make use of these queries make modifications, if needed and execute them in the database.

### *6. Synonym for search words*

Data available in the database can be mapped to more meaningful strings so that searching for any one of them can fetch all the rows containing the mapped values. Real world (or Application) representation of data need not be the same as it is stored in the database. For Example, Male or Female information presented in the application might be represented as M or F in the database. Even users who search for this data would naturally like to search them using the terminologies as it is presented in an application. SQLOne provides mechanism to configure this mapping so that end users could search using the key words they are familiar with and the search engine will smartly provide the relevant results.

### *7. SQL function support using keywords*

SQLOne helps you to fetch results for simple SQL functions from database. For Example, you can directly type `sale_amount: >1000` in the search box to get results where the `sale_amount` values greater than 1000 will be listed.

### *8. Search Enabling Unmodeled database:*

In the unmodeled case, the Primary Key and Foreign Key relationships are not well defined for the tables. In such cases, SQLOne can find possible relationships between the tables by analyzing the data and makes use of such relationship during search to give a meaningful and complete search result to the user.

### *9. Incremental Indexing:*

For huge databases where there are frequent updates to the tables, it is very costly to re-index the whole DB every time. To avoid this, SQLOne offers support for Incremental Indexing that once enabled, will index only the changes happened to the database since the last indexing.

### *10. Preventing unwanted tables \ columns in Search:*

For applications that have tables, which are not directly useful for the users, like the tables, which are needed for the internal working of the application, SQLOne offers the ability to select the tables, columns that have to be indexed so that only the selected tables are displayed during search.

## The Nature of data in an Enterprise

Most enterprises have legacy applications with data available in the database and often; the Meta information available about them is either unavailable or outdated. In such cases, where no one has knowledge about the database, SQLOne can intelligently find out the Meta information based on the analysis of data itself with out the need for any user input. To facilitate this, the Crawler engine of the SQLOne works with two scenarios.

- Modeled Database
- Unmodeled Database

### Modeled Database

The modeled database is one for which the Primary Key and Foreign Key information is well defined. During search, SQLOne makes use of this primary key and foreign key information to fetch the complete information about the search words from multiple tables. In the modeled case, since it is well defined, the JDBC API is used to fetch them and populate the necessary index tables.

### Unmodeled Database

The Unmodeled database is one for which the Primary Key and Foreign Key information is NOT well defined. During search, SQLOne makes use of the primary key and foreign key information to fetch the complete information about the search words from multiple tables. Though it is not defined in the unmodeled case, there are cases where the same data is duplicated in different tables (without defining them explicitly as a foreign key), which implies that the information in these tables is about the same entity. For such unmodeled case, SQLOne uses the data to analyze and find the relationships and populate the necessary index tables.

All the information about the tables found by the Crawler engine are stored in the index tables and made use of during search process to fetch meaningful cluster of results from different tables.

The important components of SQLOne are:

- 1 Crawler
- 2 Search Engine
- 3 Database Adapter
- 4 Client

*Crawler:*

This component takes care of indexing the given database. The important index information stored about the data is

- The location of the occurrence of a word
- The count of the occurrences of a particular word
- The type of the column in which a particular word occurs
- The rank of the tables calculated based on the importance of the table
- The relationship between different tables

Search based on partial keywords is also supported if it is enabled during the crawling process.

The other important configurations that can be made to the crawler are

- Specify the delimiters to be used during indexing so that the value of a cell is tokenized based on the configured delimiters and indexed
- Specify the list of tables\columns that should be indexed so that the others can be ignored during indexing
- Specify the columns based on which the results of a table should be sorted while displaying the search results
- Schedule the indexing at frequent intervals to make sure that search stays up-to-date.

All these configurations made by the user and the index information created by SQL is stored in the default database of SQLOne.

#### *Search Engine:*

This component takes care of fetching the right information from the database, given the search words, based on the index information. If multiple search words are provided, the search algorithm will fetch the data from relevant tables if the search words are present in

- Same cell of a table
- Across different cells of a table in the same row
- Across cells of different tables, provided the rows of the different tables are related

For each of the above cases, the search result will not only contain the rows that contained the search words, but all the rows from other tables that are related to the given table. This enables the user to see a meaningful data from the results in cases where information about a single entity is split across multiple tables because of normalization. Retrieval of data from all the related tables is done using the relationship between the keys of different tables that is indexed. Here, all the tables are fetched using a single query joining all the tables involved. The query used is also given as part of the search result. The output SQL query is specific to the SQL dialect of the type of database (like Oracle, MySQL...) that contains the data.

SQLOne has two important components to calculate the rank of the results. The first component is the static component, which is purely dependent on the table properties and this component is calculated during the indexing process. The second component is the dynamic component, which is dependent on the importance of the table with respect to the given key words, and the nature of the search done (like exact match, multiple words with match all, multiple words with 'match any' etc.). This dynamic component is calculated during the search process and the final rank of the table is calculated as a function of both the static component and the dynamic component.

Apart from search, SQLOne console also supports simple queries from the search interface. We can use some simple keywords and get the results of the SQL queries.

#### *Database Adapter*

This component is responsible for the database neutrality of SQLOne. The database adapter provides a common interface for doing the Create, Retrieve, Update and Delete (CRUD) operations. This layer internally handles the discrepancies between different databases, such as the difference in SQL syntax, data types supported and features provided by different databases. The Crawler and the Search Engine of SQLOne interfaces with the database adapter layer to perform all the database operations.

The important components of the database adapter layer are the SQL Generator component that generates database specific SQL Statements at run time and the database adapter component that executes the generated SQL statements and outputs the result. Separate implementation of the SQL Generator and Database Adapter will be provided for each of the databases supported to address the database inconsistencies. Currently, the default implementation for Oracle, MS SQL Server, and MySQL are provided. This could further be extended to support other databases easily.

### *Client*

The Client component of SQLOne provides a simple administrative console and the search interface. The Admin console provides the GUI for making all the indexing configurations for a database and the Scheduling the indexing process. The search interface apart from providing a default view for showing the search results also provides a customizer using which the results can be shown as per the applications need.

### *Flow of Sequence*

The list of databases that needs to be search-enabled is given as input to the Crawler with the configuration parameters specific to each of the data sources. The crawler will take the input database, scan through the tables and the data available in each of the given DB and creates the index information. This index information is again stored in the database. The index tables are created in the default database bundled with SQLOne.

Given a set of search words to the search engine, it first looks up the index database and sees whether there are any matching data found. If so, from the index it fetches the information about the tables (& its related tables) in which the search word occurs. From this it constructs the appropriate SQL queries that would fetch the required rows in which the search word occurs.

Ranking is done for the results and the different tables in the result are sorted based on the importance. Then the query for each of these tables is then executed on the actual database and the result is returned. Both the crawler and the Search Engine go through the Database Adapter layer for doing any database operations.

The search results are displayed as tables with column value pairs in the default UI. This can be customized as per the application using the client customizer. This allows changing the layout of the columns shown in the result, the order in which they are shown, the way information from other tables is shown.

### *Features offered by AdventNet SQLOne*

- Given the name of a host, SQLOne can automatically discover the databases available and list them, so that the database to be indexed can be selected by the user.
- The crawler can be easily configured from an administrative console to suit the specific needs of the application
- New databases can be submitted on the fly to SQLOne so that it can be indexed and once complete, data can be fetched from the database-using search.
- Single interface supporting different databases like Oracle, MS SQL, and MySQL
- SQLOne can handle more than one database, so any number of databases can be indexed and searched

- For unmodeled database where the primary key and foreign key information of a table are not defined, SQLOne can analyze the data and find the possible relationships between the tables.
- Provides functionality to map any value in the database to an equivalent meaningful value. This would be helpful especially in cases where the column values are of single char or int or Boolean data types.
- SQLOne auto-generates SQL queries (involving any level of joins required) that could fetch the data row(s) containing the search word. Users can run of the crawler.
- Most relevant output in the first page of the search results by sorting them based on importance with respect to the given keywords.
- Intelligence to pull meaningful data even if all the search words are not present in a single table but spread across multiple tables which are related through their relationship keys
- Ability to fetch data even based on the partial keywords given as search input.
- Search based on single word, multiple words with AND\OR options.
- Options to prevent indexing sensitive data using configuration options like Tables To Ignore, Columns To Ignore and Data type To Ignore
- Support for incremental indexing which will index only the changes made after the previous run of the crawler.
- Support for scheduling the crawler at a specified time as well as after specified intervals of time
- Support to filter out most common words (Stop Words) from indexing.
- Support for Delimiters based on which the value of a cell will be tokenized and indexed. Default tokenizers are comma, colon, white space, at (@) symbol, open bracket, close bracket, curly open and close brackets.
- View Customizer to help users customize search results using the knowledge of the application
- Offers ease of integration of SQLOne with enterprise applications with simple and easy client level customization and advanced application code level integration with SQLOne API.

## Advanced Search

Apart from Search, one can also get results of specific queries from the search interface by using some basic keywords. Some of the SQL functions supported from the search interface are

- Fetching all the rows of a table given the name of the table as input
- Fetching the given columns of a table given the column name alone as input to search
- Select specific columns of a table and view the results
- Search the given words in the specified tables
- Search the given words in the specified columns of the table
- Search based on some simple criteria with logical, relational and other sql operators.
- Support for SQL functions like MIN, MAX, AVG, COUNT.

*Example:*

This section will provide a simple example to explain the search capabilities of SQLOne.

Consider two applications, one for the HR (Human Resource department) to maintain information about the employees and other for the accounts department to maintain the salary information of its employees.

Let us assume that the HR application data is stored in a MS SQL database and the accounts application makes use of the Oracle database. Let the tables defined in the *HR database* be

*Employee*

EMPLOYEEID  
EMPLOYEEENAME  
AGE  
QUALIFICATION  
MARITALSTATUS

*ContactInfo*

EMPLOYEEID  
DOORNUM  
STREETNAME  
CITY  
ZIPCODE  
PHONENUM

*SpouseInfo*

EMPLOYEEID  
SPOUSENAME  
AGE  
QUALIFICATION  
ISEMLOYED

*JobProfile*

EMPLOYEEID  
TEAMNAME  
DURATION  
ROLE  
RATING

Let the *Accounts database* have the following tables

Employee  
EMPLOYEEENAME  
ACCOUNTID  
CATEGORY

*SalaryInfo*

ACCOUNTID  
BASIC  
HRA

*CarAllowance*

ACCOUNTID  
CARALLOWANCE  
FUELCOST  
MAINTENANCE  
DRIVERALLOWANCE

*MedicalAllowance*

ACCOUNTID  
PERSONAL  
DEPENDENTS

If for insurance purposes, the HR wants to see the spouse information and the details of income of the employee, then we need to fetch the data explicitly from the MS SQL Server and Oracle databases either through the respective application interfaces or using specific MS SQL queries and Oracle queries. If SQL queries are to be used the user has to know the details of the name of the table, column that contains the given information and the list of tables that he is interested in and how these tables are related to the original table. With SQLOne, the user can get to see the data from all the databases from a single interface by just giving the name of the employee as input. It is achieved as follows:

First index the HR and the Accounts database using the Crawler engine of SQLOne. Once crawling is complete, search with EMPLOYEEID as input. The search result will pull data from both HR and Accounts databases. Moreover, it will also fetch the data for the given employee from tables like ContactInfo, SpouseInfo, JobProfile and SalaryInfo, CarAllowance, MedicalAllowance.

For a search with word 'John' (assuming John is an employee) the results will contain the following information:

- Data from all the databases containing the search words
- Data from table that contains the search words
- Data from all the other tables that are related to the table containing the given search words.

## HR application database

### *Employee*

EMPLOYEEID - 10  
EMPLOYEEID - John  
AGE - 25  
QUALIFICATION - B.E  
MARITALSTATUS – Married

### *ContactInfo (fetched using EMPLOYEEID)*

EMPLOYEEID - 10  
DOORNUM - 19  
STREETNAME - II CROSS  
CITY - CHENNAI  
ZIPCODE - 600 042  
PHONENUM - 2244 6688

### *SpouseInfo (fetched using EMPLOYEEID)*

EMPLOYEEID - 10  
SPOUSENAME - Rose  
AGE - 25  
QUALIFICATION - B.E  
ISEMLOYED – true

### *JobProfile (fetched using EMPLOYEEID)*

EMPLOYEEID - 10  
TEAMNAME - SUPPORT  
DURATION - 0.5  
ROLE - Support Engineer  
RATING - Very Good  
EMPLOYEEID - 10  
TEAMNAME - MARKETING  
DURATION - 1.5

*Accounts application database*

Employee

EMPLOYEE\_NAME - John

ACCOUNTID - 35

CATEGORY - Member Technical Staff

*SalaryInfo (fetched using ACCOUNTID)*

ACCOUNTID - 35

BASIC - 15000

HRA - 7500

*CarAllowance (fetched using ACCOUNTID)*

ACCOUNTID - 35

CARALLOWANCE - 3000

FUELCOST - 500

MAINTENANCE - 500

DRIVERALLOWANCE - 2000

*MedicalAllowance (fetched using ACCOUNTID)*

ACCOUNTID - 35

PERSONAL - 1000

DEPENDENTS - 500

With the above result HR personnel could navigate through the search results across the HR and Accounts databases and get the relevant information of the employee 'John'

## **Summary**

SQLOne with its unique methodology of understanding the relationship between the primary and foreign keys gives a multi dimensional view of the data to the users. With its capability to handle multiple databases from different vendors, it gives a common interface to data from different applications, which cannot be got from any of the existing application. In way SQLOne could provide an enterprise wide search interface for applications (databases) without the need for application interface themselves.

Though many existing products have the capability to construct query based on the inputs given by the user through nice GUIs, search words based query generation is something has not been attempted so far. SQLOne is a unique product that offers true value to enterprises by offering an effective database search and empowering enterprises to get the information needed at the time of need.