

General SQL Parser

[Download »](#)
[Buy Now »](#)


General SQL Parser is a package that enables you to add powerful SQL functionality to your applications.

General SQL Parser offers a custom SQL engine for many widely-used databases, including a query parser for SQL Server, Oracle, DB2, MySQL, Teradata, Sybase, Informix, Netezza, PostgreSQL, Hadoop Hive and Access databases.

Toolsverse offers permanent **30% discount** on all [Gudu Software products](#). Read about, download and try all versions of the General SQL Parser from either web site but if you [buy it from our e-shop](#) you will get a 30% discount + great support directly from Gudu software.

About the General SQL Parser

Adding a home-grown SQL parser to your applications is a difficult problem. Not only is decoding SQL grammar difficult, but database vendors are constantly releasing new versions of their databases. Maintaining your own SQL parser is time-consuming, error-prone, and costly.

[Gudu Software](#) has developed a SQL parser that simplifies decoding SQL grammar and helps your applications stay current with the latest versions of database programs.

The General SQL Parser is a package that enables you to add powerful SQL functionality to your applications. General SQL Parser offers a custom SQL engine for many widely-used databases, including a query parser for SQL Server, Oracle, DB2, MySQL, Teradata, Sybase, Informix, Netezza, PostgreSQL, Hadoop Hive and Access databases.

Using the General SQL Parser library, you can save hundreds of hours of development time, improving your applications and speeding them to your customers.



Problems the General SQL Parser Solves

The General SQL Parser solves many common and difficult SQL development problems, including the following:

- Checks SQL syntax offline so that you can validate syntax without connecting to a database.
- Formats SQL with more than 100 highly customizable format options.
- Performs an in-depth analysis of SQL scripts, including a detailed SQL parse tree node structure.
- Provides full access to the SQL query parse tree, including fetching, modifying, and rewriting SQL segments.
- Prevents SQL injection attacks.
- Translate SQL between different databases

Using the General SQL Parser

The General SQL Parser is available for major programming languages, including C #, VB.NET, Java , C/C++ , Delphi, and Visual Basic.

Offline SQL Syntax Check

With our vendor-specific offline SQL syntax check, you can validate SQL syntax without connecting to the database server. Syntax errors can be found even before executing SQL on your production database server. This is especially useful if your SQL was dynamically built based on user input. Supported databases include Oracle (PLSQL), SQL Server , DB2 , MySQL , Teradata , Sybase, Informix, Netezza, PostgreSQL, Hadoop Hive and Access databases.

SQL Formatter

You can easily integrate our SQL formatter into your application to generate a color-coded, professional, and intuitive SQL layout that gives your product a professional look and feel. Our highly customizable SQL formatter offers more than 100 format options, including text, HTML, and RTF output.

We know how difficult it is to build a flexible and stable SQL formatter. If you need to support more than one SQL dialect – such as Oracle , SQL Server , DB2 , and MySQL – then this library could be the smartest investment you ever make, saving you hundreds of hours of coding.

Avoid SQL Injection Attacks

Worried about being vulnerable to SQL injection attacks on your ASP.NET or Java applications? Our operational library automatically detects malicious SQL segments .

Database Intelligence

Precisely determining and renaming every table and column in stored SQL statements that include considerable nesting sub-queries is a complex procedure. Using the General SQL Parser can make this tedious procedure hassle-free for you, so that you can easily:

- Extract all tables and field names.
- Parse the values and field names in where or having clauses, and verify that the user has specified certain columns in the where clause.
- Determine which tables are accessed, and how these tables are accessed (read, write, or both).
- Determine which table and column are used in a select list field.
- Rename specific table names and rebuild the SQL statements.
- Perform column impact analysis and build source-to-target column/table mappings.
- Read SQL scripts to determine which tables have Create, Read, Update, Delete, and Insert operations against them.

SQL Query Parse Tree

Parsing SQL is a notoriously difficult task. But the General SQL Parser can produce a SQL query parse tree in XML format, so you can further process SQL scripts .

How the General SQL Parser works

The General SQL Parser consists of complete library packages available for the following development environments:

- .NET
- Java
- C /C++
- COM
- VCL

The General SQL Parser supports a multitude of databases, including SQL Server , Oracle , DB2 , MySQL , Teradata , Sybase, Informix, Netezza, PostgreSQL , Hadoop Hive and Access databases.

Not only do we provide the libraries. We also offer a wide range of free demos that you can copy and adapt to solve your specific problems.

Below is the high-level architecture of the General SQL Parser.

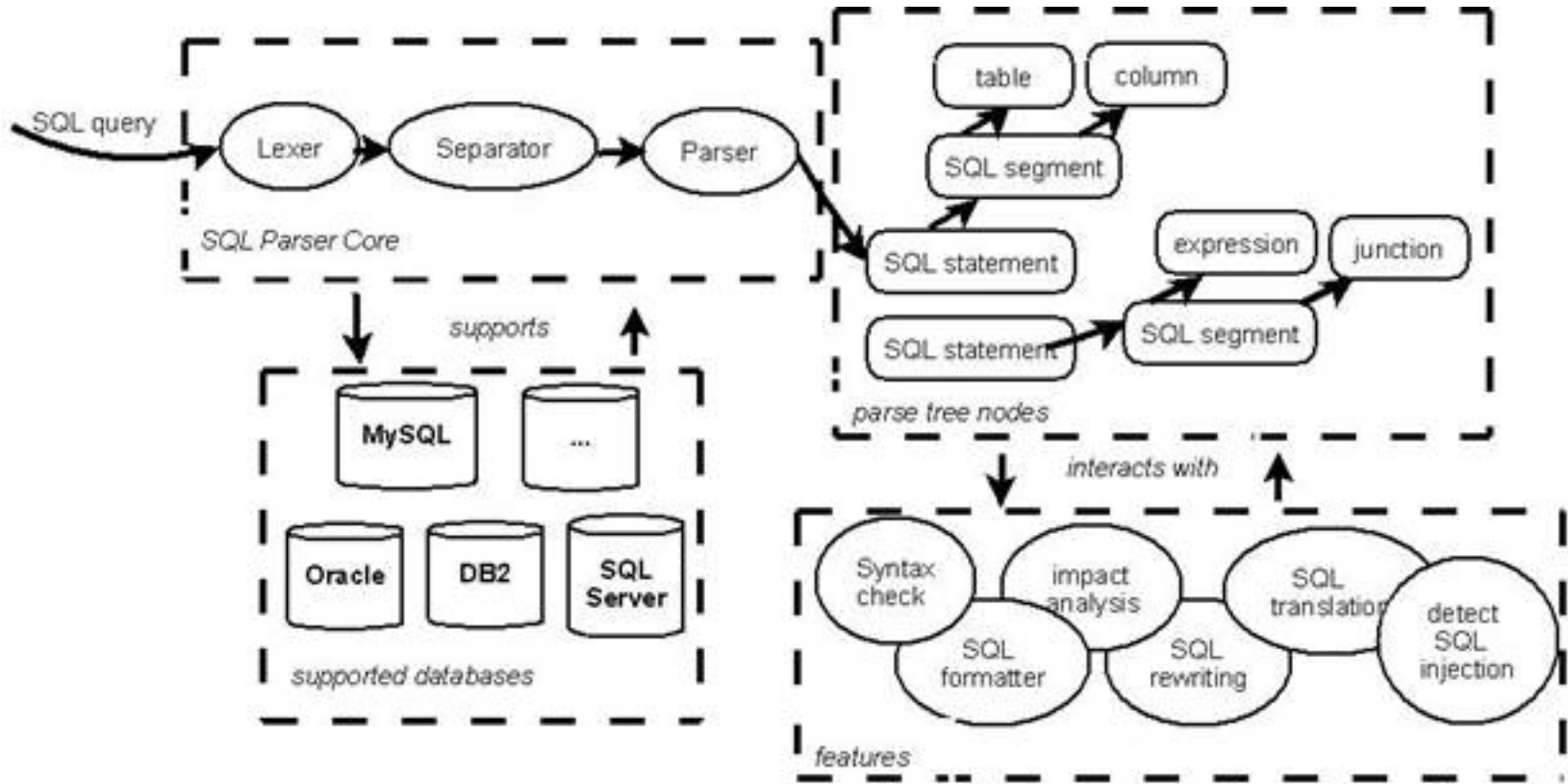


Figure1: Architecture of General SQL Parser.

The General SQL Parser reads the input SQL text of the query. Lexer (lex parser) tokenizes the input SQL into a list of tokens. Separator turns the source tokens into a list of SQL statements. Each separate SQL statement includes a fragment of source tokens, which is the input to the Yacc Parser.

The Yacc Parser reads the source tokens of each SQL statement. If no syntax error is found, the Parser creates a raw parse tree , based on the BNF of different dialects of databases. Otherwise, the syntax error is detected. A syntax error in one SQL statement doesn't affect others.

Parser then translates the raw query parse tree into a formal parse tree. The top-level node of the formal parse tree is a specific type of SQL statement, such as TSelectSqlStatement , TInsertSqlStatement , TDeleteSqlStatement , TUpdateSqlStatement , or TCreateTableSqlStatement , which, in turn, includes other sub-parse-tree nodes.

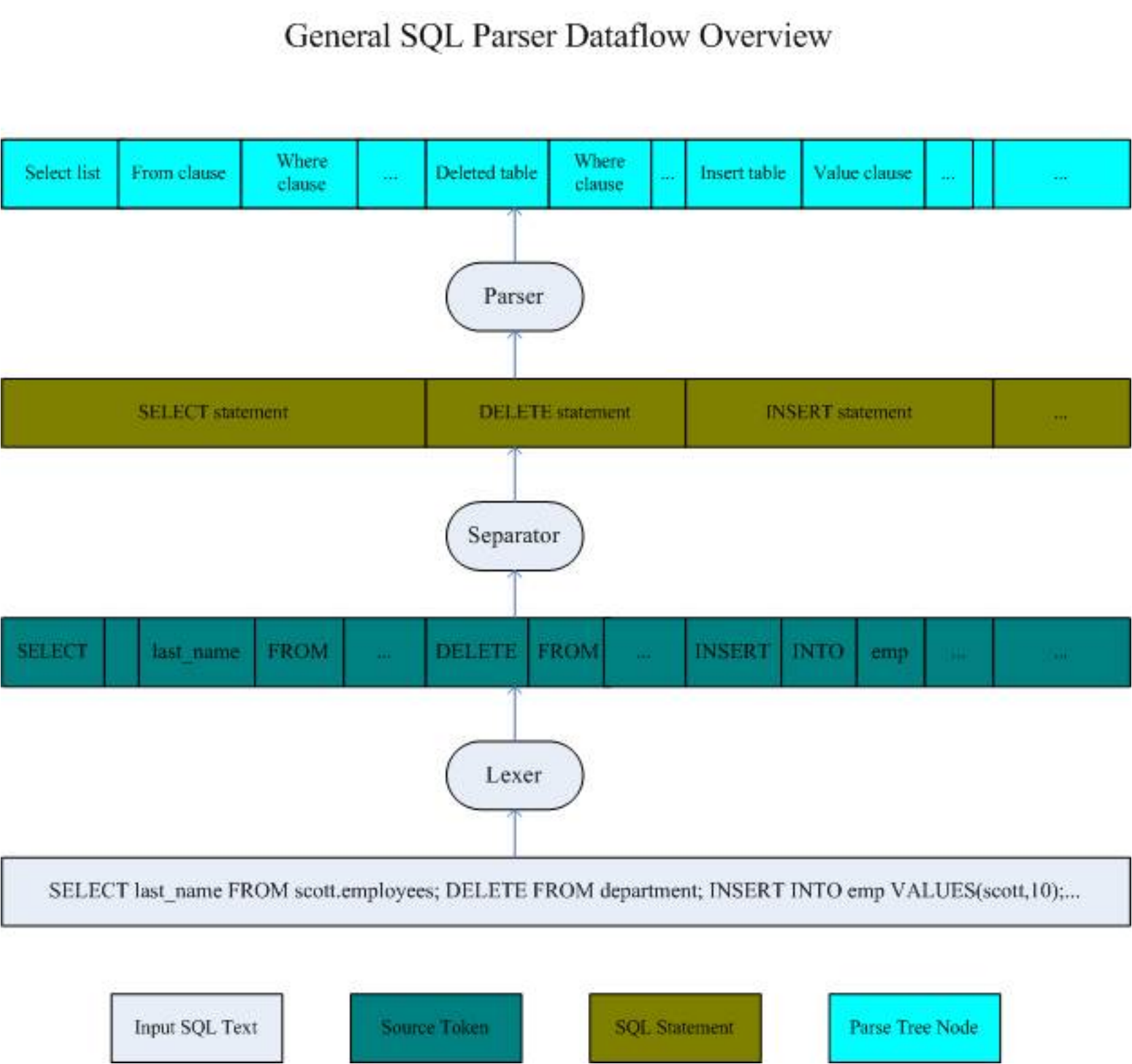


Figure 2: Data Flow in General SQL Parser .

The formal parse tree has a variety of uses, such as the following:

- Obtaining information, such as tables and fields, in a SQL statement is easy. These items are ready for use, along with other database objects in SQL objects.
- Parse tree visitors that descend from TLzVisitorAbs can be used to iterate the parse tree . TLzParseTree2XMLVisitor is a simple visitor that translates the parse tree to XML output. This class is a good reference before you write your own visitor to handle the parse tree.

Many other uses depend only on your requirement and imagination.

Comparing the General SQL Parser to other products

When comparing the General SQL Parser to other products, consider the following points:

- Other products are available only for a single language. The General SQL Parser is available for major programming languages, including C #, VB.NET, Java , C/C++ , Delphi , and Visual Basic .
- Other products are available only for a single database. The General SQL Parser is available for major databases, including SQL Server , Oracle , DB2 , MySQL , Teradata , Sybase, Informix, Netezza, PostgreSQL , Hadoop Hive and Access databases.
- The SQL parsers of database vendors are embedded into the database. There is not API available for use by third-party developers and vendors.
- The General SQL Parser 's powerful SQL parse and analysis capabilities enable it to parse complex SQL statements that other parsers cannot handle.
- Makers of other parsers actually recommend the General SQL Parser :
 - <http://zql.sourceforge.net>
 - <http://jsqlparser.sourceforge.net>

ETL Framework	ETL Framework	ETL Framework	News
General SQL Parser	General SQL Parser	General SQL Parser	Testimonials
Case Studies			Credits
			Contact Us
Services	Support	Clients	
Custom Software	FAQ		
Development	ETL Scenario		
Cloud Solution	Examples		
	Java Examples		
	Video Tutorials		
	Knowledge Base		