# Glossary

This is a reference list of terms related to Jethro products. Additional information is available from a number of resources, including the Reference Guide Glossary.

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z

## A

**Adaptive Cache**
As users work with their favorite BI tools, the sequence of SQL queries that these tools generate and send to Jethro has some predictable patterns. For example, most users start from a dashboard (that sends a predictable list of queries), than typically start adding filter conditions and aggregate conditions one at a time.
Adaptive cache is a unique JethroData feature, which leverages those patterns. It is a cache of re-usable, intermediate bitmaps and query results that is automatically maintained and used by the **JethroServer** engine on its local storage. The cache is also **incremental** – after new rows are loaded, the next time a cached query is executed it will in most cases automatically combine previously cached results, and computation will only be performed over the newly loaded rows.
For further details, see section *Managing Adaptive Cache* under chapter Administering Jethro"Administering Jethro" on page .

**Adaptive Index Cache**
The adaptive index cache is part of the adaptive cache that holds intermediate bitmap index entries that were computed on the fly during query execution.
For further details, see section *Managing Adaptive Cache* under chapter Administering Jethro"Administering Jethro" on page .

**Adaptive Query Cache**
The adaptive query cache is part of the adaptive cacheA cache of re-usable, intermediate bitmaps and query results that is automatically maintained and used by the JethroServer engine on its local storage.that holds **intermediate query result sets** (for SELECT statements only).
A query is considered for inclusion in the adaptive query cache if its running completed without errors and took more than **adaptive.cache.query.min.response.time**, which defaults to one second. Also, the cached result set must be lower than **adaptive.cache.query.max.results** rows, which defaults to 100,000 rows.

**Append-only**
Append-only data stores allow only adding data into a table; there is no possibility to remove or change data. Data deletion is only enabled by dropping an object.
Jethro format is append-only, and supports partitioning. As a result, dropping a partition is the only way to delete data.

## B

**BI**
Business intelligence (BI) is a technology-driven process for analyzing data and presenting actionable information to help corporate executives, business managers and other end-users make more informed business decisions. BI refers to a wide variety of tools, strategies, applications, data products, and methodologies that enable companies and organizations to spot, collect, analyze, present, and disseminate business information that arrives from internal systems and external sources. Using BI tools allow companies to develop and run queries against the data, and create reports, dashboards, and data visualizations to make the analytical results available to corporate decision makers as well as operational workers.

BI tools allow understanding the current situation and identifying patterns, based on both historical information and new data gathered from source systems as it is generated. This information can assist in accelerating and improving decision making; optimizing internal business processes; increasing operational efficiency; driving new revenues; and gaining competitive advantages over business rivals. In addition, BI systems can help companies predict future market trends and spot possible business issues that need to be prevented or addressed.
Using BI tools allow making both basic operating decisions, such as product positioning or pricing, and strategic business decisions such as priorities and goals.
While initially BI tools were primarily used by data analysts and other IT professionals who ran analyses and produced reports with query results for business users, the emergence of self-service BI and data discovery tools such as Tableau and QlikSense/QlikView made these tools available to business executives and non-IT company workers.

**Big Data**
Big data is a term that describes data sets - be it structured, semistructured or unstructured data - that are too large or complex to be processed by traditional database and software techniques. The complexity of handling big data is often a result of its main characteristics, known as the 3vs: The extreme volume of data, the wide variety of data types, and the velocity at which the data must be processed.

While the term in itself does not denote any specific volume of data, it is often being used for describing terabytes, petabytes, and even exabytes of data captured over time.
Big data is mostly used by organizations and companies, as it assists in improving operations and making faster, more intelligent decisions. When this data is captured, formatted, manipulated, stored, and analyzed, it can assist in gaining useful insight to increase revenues, get or retain customers, and improve operations.

## C

**Columnar Database**
A columnar database, also known as a column-oriented database, is a database management system (DBMS) that stores data in columns rather than in rows as relational DBMSs. Storing data in columns rather than rows allows the database to more precisely access the data required for answering a query, instead of scanning and discarding unwanted data in rows. As a result, query performance is often increased, particularly in very large data sets.
One of the main benefits of a columnar database is that data can be highly compressed, which allows for a very rapid execution of columnar operations such as MIN, MAX, SUM, COUNT, and AVG. Another benefit is that because a column-based DBMS is self-indexing, it uses less disk space than a relational database management system (RDBMS) that contains the same data. However, the loading process can take time depending on the size of data that is involved.

## D

**Database**
A database is a collection of information that is organized to be easily accessed, managed, and updated by a collection of programs known as database management system (DBMS) . Computer databases typically contain aggregations of data records or files, such as sales transactions, product catalogs and inventories, and customer profiles.
The DBMS, which are sometimes loosely referred to as "databases", use standards such as SQL, JDBC, and ODBC to access applications, thereby allowing a single application (for example, Tableau) to work with multiple DBMS.
Modern DBMS are largely divided into two main types:

- Relational databases (Oracle, SQL Server, DB2 and so on)
- No-SQL (big data) databases (Cassandra, Hadoop and so on)

No-SQL databases, on the other hand, are used for handling rapid growth of unstructured data and scaling them out easily. NoSQL is especially useful when an enterprise needs to access and analyze massive amounts of unstructured data or data that's stored remotely on multiple virtual servers in the cloud, and are therefore used by companies that have such massive amounts of data, such as LinkedIn and Twitter.

**DataNode**
HDFS has a master/slave architecture, with a single master server called NameNode that manages the file system namespace and regulates access to files by clients, and multiple DataNodes, usually one per cluster, with data replicated across them.
DataNodes store the actual data in HDFS - namely, a series of named blocks - and serve read and write requests from the file system's clients by allowing client code to read these blocks or to write new block data. Upon startup, each DataNode announces communicates with the DataName to announce itself and the list of block for which it is responsible, and maintains constant communication with the DataName as long as the DataNode is running (up). Each DataNode also communicates with client code and other DataNodes from time to time.
The replication of data between DataNodes mean that when a DataNode is down, the availability of the data or the cluster is not affected; NameNode ensures that the blocks managed by the DataNode that is down is replicated to other DataNodes.
**Data Types**
A data type, in computer science and computer programming, is a classification that specifies which type of value a variable has, what is the meaning of the data, the way values of that type can be stored, and what type of mathematical, relational or logical operations can be applied to it without causing an error. A string, for example, is a data type that is used to classify text, a float is used for classifying numbers with a decimal point (3.14, for example), and an integer is a data type used to classify whole numbers (5, 15 and so on).
The data type defines which operations can safely be performed to create, transform and use the variable in another computation; for example, a float can be multiplied by an integer (1.5 x 5), but not by a string (1.5 x Dutch). In addition, data types are used for defining the length of information strings to be stored. For example, in MySQL the TEXT data object type can store up to 65,535 characters, and can therefore perhaps hold the text of a single article but is not suitable for storing an entire book.

## H

**HDFS**
The Hadoop Distributed File System (HDFS) is a distributed file system and a framework for the analysis and transformation of very large data sets. HDFS is designed to reliably store very large files across low-cost machines in a large cluster, and to stream those data sets at high bandwidth to user application. By distributing storage and computation across many low-cost servers, the resource can grow with demand while remaining economical at every size; the scaling of computation capacity, storage capacity, and I/O bandwidth is caried out by simply adding commodity servers.
HDFS stores metadata on a dedicated server, called the NameNode. Application data are stored on other servers called DataNodes. All servers are fully connected and communicate with each other using TCP-based protocols.

## I

**Index**
A database index is a data structure that improves the speed of data retrieval operations on a database table (in SQL, SELECT queries and WHERE clauses) by providing a pointer to data in the table. This pointer, known as index key, is used for fast retrieval of data, and can be based either on the primary key (the unique identifier of a row, such as ID number) or on any other, non-unique data such as first name or department name.
An index is a small copy of a database table sorted by key values, without which query languages such as SQL may have to scan the entire table from top to bottom to select relevant rows.
While indexes speed up retrieval operations, they slow input operations such as UPDATE and INSERT, because the index must be updated upon any update of the underlying table.

## J

**JDBC Driver**
JDBC driver is a software component that allows a Java application to connect to databases that support SQL.
JDBC (Java Database Connectivity application programming interface or API) requires drivers to each database, to enable carrying out the following operations:

1. Establishing a connection with a data source
2. Sending queries and update statements to the data source
3. Processing the results

**JethroLoader**
The Jethro's loader utility **JethroLoader** allows fast and efficient data loading from text files into a Jethro table. It parses its input into rows, and loads them into an existing Jethro table.
The default behavior of **JethroLoader** is **APPEND**- inserting the input file data into the existing table. You can also optionally limit the load to just a specific set of partitions – the rest of the records will be skipped. Alternatively, you can specify **OVERWRITE** to replace either an entire table or just a specific set partitions. This allows updating or deleting rows from a table, in cases where dropping a partition is not fine-grained enough.

**JethroMaint**
The JethroMaint service is in charge of running maintenance tasks in the background. Its main responsibilities are:

- **Optimizing indexes** - Performing a background merge of column indexes after data loads.
- **Deleting unneeded files** - After an index was optimized, the files of its old version can be deleted. However, there could be queries already in progress that access the older version of the index, so the older versions are deleted after a delay. Dropped and truncated tables are also physically deleted in the background after the same delay.

If the JethroMaint service was not running for a long time (for example, it was manually stopped or it ran into an unexpected issue), it could affect query performance, as each column may have non-optimized index with many small index files.

**JethroServer** automatically caches some of the JethroData model files (columns and indexes). Caching can accelerate read time for frequently accessed files, as well as avoiding frequent accesses to the NameNode and DataNodes.
The local cache is populated by the **JethroServer** process in the background. It is used by all Jethro processes, including **JethroLoader** and **JethroMaint**.

**Join Index**
Join index is an index on one table, based on the values of a column in another table (dimension) and on a specific join criteria. Typically, it is an index on a large fact table based on the values of a dimension attribute. Join index accelerates queries by eliminating both the fetch of the join key from the fact table and the join implementation (hash join or IN - merging indexes).Join indexes are relevant when you have a relatively large dimension (few K values or more), and the attribute (the column in the dimension) is low cardinality, so that each value in the attribute represents many join key values.

## K

**Kerberos**
Kerberos is a protocol for authenticating service requests between trusted hosts across an untrusted network, such as the Internet. Kerberos is built in to all major operating systems, including Microsoft Windows (where it is an integral component of the Windows Active Directory service), Apple OS X, FreeBSD and Linux.
Kerberos uses three components: a client, a server, and a Distribution Center (KDC), which acts as Kerberos' trusted third-party authentication service. The KDC provides an authentication service and a ticket granting service. Using KDC "tickets" allows nodes to prove their identity to one another in a secure manner.
By using shared secret cryptography, Kerberos authentication prevents packets traveling across the network from being read or changed, and protects messages from eavesdropping and replay attacks.

## L

**Loader Description File**
The loader description file describes the structure of the input file and the way it will be processed. It has three sections:**Table-level section** – describes the format of the input file and some processing options**Column-level section** – a mapping between the fields in the file and columns in the table*(Optional)* **Record description section** – special clause for handling variable format files – files with different format per line (discussed in section Input Files with Variable Format under chapter *Loading Data* in the Reference Guide)

## N

**NameNode**
HDFS has a master/slave architecture, with a single master server called NameNode that manages the file system namespace and regulates access to files by clients, and multiple DataNodes.
NameMode is the centerpiece of HDFS, because it controls access to the system by storing the metadata of HDFS – the directory tree of all files in the file system - and tracking the files across the cluster. Files in HDFS are composed of blocks, which are managed by the various DataNodes (no data is stored in the NameNode), and the NameNode stores the list of blocks, as well as the location for any given file in HDFS, which enables it to construct the file from blocks and to assign (replicate) blocks managed by a DataNode to other DataNodes once the specific DataNode is not running (down).
NameNode is a single point of failure to HDFS, which is is not currently a High Availability system; when the NameNode goes down, the file system goes offline.

O

**ODBC Driver**

Open Database Connectivity (ODBC) is an interface standard that allows any application supporting ODBC to accessing data and communicate with ODBC-compatible database systems, regardless of the operating system (OS), database system (DS) or programming language.
Database compatibility with ODBC is achieved by inserting a middle layer, called ODBC database driver, which translates standard ODBC commands into commands understood by the database's proprietary system. This can work only if both the application and the DBMS are ODBC-compliant - namely, the application must be capable of issuing ODBC commands and the DBMS must be capable of responding to them.

P

**Partition**

A partition is a division of a logical database or its constituent elements into distinct independent parts. Partitioning allows the subdivision of tables, indexes, and index-organized tables into smaller pieces, thereby enabling these database objects to be managed and accessed at a finer level of granularity. Partitioning a database improves performance and simplifies maintenance. Splitting a large table into smaller, individual tables signicantly shortens the amount of time required for running queries, because queries that access only a fraction of the data have much less data to scan. Maintenance tasks, such as rebuilding indexes or backing up a table, can run more quickly.
From a database administrator's point of view, a partitioned object has multiple pieces that can be managed either collectively or individually. This gives administrators considerable flexibility in managing partitioned objects. However, from the perspective of the application, a partitioned table is identical to a non-partitioned table; no modifications are required when using SQL queries and DML statements to access a partitioned table.

R

**Relational Database**

A relational database is a set of related tables, which contain data fitted into predefined categories; for examples, items in a department store or employees in an organization. The objects in the relational databases are strictly structured: all data in the table is stored as rows and columns, and each column has a data type. Relational tables follow certain integrity rules that ensure that the data they hold stay accurate and is always accessible. Relational databases use SQL to store and retrieve data in a structured way.
Relational databases are suitable for handling systems that require complicated querying, database transactions, and routine analysis of data; for example, cell phone companies. They are also the databases of choice for applications involved in many database transactions (for example, superstores), where it is vital that those transactions are processed reliably. Relational databases are characterized by a set of properties that guarantee database transactions are processed reliably (ACID), which is highly valuable for such applications.
However, relational databases are less suitable for handling rapid growth of unstructured data , a task that is usually performed by no-SQL databases.

**Rolling Window Operation**

Rolling window operation refers to the archiving and removal (purging) of old data, when new data is added to the data warehouse. This practice is commonly used when deciding to keep only a few years of data, thereby preventing the data in the data warehouse from growing indefinitely.

S

**Schema**

A database schema is the organization or structure for a database, which is described in a formal language supported by the database management system (DBMS). A schema defines attributes of the database, such as tables, columns, and properties. The term "schema" refers to the organization of data as a blueprint of how the database is constructed, and is being is used in discussing both relational databases, where the schem defines that the database is divided into database tables, and object-oriented databases. Schema sometimes refers to a visualization of a structure and sometimes to a formal text-oriented description.
Using schemas allows grouping objects into separate namespace. When security rules are applied to a schema, these rules are inherited by all objects in the schema. Once you set up access permissions for a schema, those permissions are automatically applied as new objects are added to the schema.

**SQL**

SQL (Structured Query Language) is a standardized programming language used in accessing, editing, or updating, information stored in a database. Initially developed by IBM to run on microcomputers and mainframes, SQL is being supported by PC database systems because it supports distributed databases, namely: databases that are spread out over several computer systems, thereby enabling several users on a local-area network (LAN) to access the same database simultaneously. SQL is now Capable of running on practically every computer, from mainframes to handheld computers; however, it is not a complete programming language capable of creating usable application programs, and must be embedded in another programs or employed through computer languages that can include SQL commands.
SQL became the de facto standard programming language for relational databases after they emerged in the late 1970s and early 1980s. Relational systems, which are also known as SQL databases, are composed of a set of tables that contain data in rows and columns. Each column in a table corresponds to a category of data - for example, customer name or address - while each row contains a data value for the intersecting column.
WHile SQL has a standard version, known as ANSI SQL, most major vendors also have proprietary versions that are incorporated and built on ANSI SQL, for example: SQL*Plus, used by Oracle, and Transact-SQL (T-SQL), used by Microsoft.

**System DSN**

Data Source Name (DSN) is a data structure that contains information about a database, and allows connecting to the database through an ODBC driver. DSN is often used by Active Server Pages (ASP) and Visual Basic programs when querying a database to retrieve information.
When setting up an ODBC Data Source you use either a User DSN - namely, a DSN that is specific to the user's profile and stored only in the user's registry - or a System DSN. A system DSN allows all users logging on to that workstation to have access to that Data Source.

T

**Table**

A table is a collection of related data held in a structured format within a database. It is composed of columns and rows.
Each row in a relational database is uniquely  identified by a primary key, which in most cases is a single column such as employeeID. As a result, it is possible to select every single row by just knowing its primarey key. The primary key may affect the number of rows a table can hold; while no practical limit is enforced, if your table holds the countries that made up the former Yugoslavia, and the primary key is the country's name, then the table is limited to severn rows becuase there are only seven former Yugoslavia countries and you cannot have duplicates in a primary key. A table can also contain zero rows, in which case it is said to be empty.
Columns are defined to hold a specific type of data, such as dates, numeric, or textual data. The most baisc definition of a column is based on the column's name and data type. The name is used in SQL statements when selecting and ordering data, and the data type is used for validating information stored. For example, if column holds the date of admission of UN member states, the column's name is DateOfAdmission, the SQL command used for ordering the colunm is ORDER BY DateOfAdmission, and if you try to add a string into this column the validation process will reject the newly entered text.
The data in a table does not have to be physically stored in the database. Views also function as relational tables, even though the database stores only the view's definition, while the view's data is calculated at query time.

**Tableau**

Tableau is a BI tool made by the Seattle-based Tableau Software, which queries relational databases, OLAP cubes, cloud databases, and spreadsheets, and then generates several graph types.
Tableau has a mapping functionality, and is able to plot latitude and longitude coordinates.
Jethro can be used with Tableau by connecting through the JethroODBC driver. Using Tableau in combination with Jethro highly accelerates the loading time of the graphs generated by Tableau.

V

**View**

A view is a virtual table, based on the result of an SQL query. While a view contains rows and columns just like an ordinary base table, it does not form part of a physical schema; removing (dropping) a view has no effect on the view's underlying data.
Views are very useful for presenting only the requested data; for example, hiding the complexity a query that joins several tables by coding the query's logic into a view, or displaying only the non-confidential rows/columns of table by saving as a view the results of a query that selects theses columns.