Best Practices for Improving SQL Performance

Version 2024.1
2024-07-02

InterSystems Corporation 1 Memorial Drive Cambridge MA 02142 www.intersystems.com

Best Practices for Improving SQL Performance
InterSystems IRIS Data Platform Version 2024.1 2024-07-02

For Support questions about any InterSystems products, contact:

InterSystems Worldwide Response Center (WRC)
Tel: +1-617-621-0700
Tel: +44 (0) 844 854 2917
Email: support@InterSystems.com

Table of Contents

Best Practices for Improving SQL Performance

While InterSystems SQL provides a number of mechanisms that will automatically perform certain operations to improve SQL performance, such as Adaptive Mode, there are a number of steps you can take yourself to ensure best performance. This page serves as a guide through some of these options.

1 Define Indexes

Indexes are a crucial part of optimizing the efficiency of your SQL queries. Adding an index on one or more fields can significantly speed up the performance of queries that use those fields for filtering, grouping, and JOIN operation by offering a faster access path, as opposed to reading the entire master map. To read more about what fields should have an index on them, refer to What to Index.

InterSystems SQL supports multiple different kinds of indexes that are each specialized for certain situations. To read about the different types of indexes, refer to the following sections of Define and Build Indexes:

- Standard Indexes
- Bitmap Indexes
- Bitslice Indexes
- Columnar Indexes

The addition of an extent index to your tables can also greatly improve the efficiency of your queries. This kind of index helps determine the existence of IDs in your table. The most efficient variant of this index is the bitmap extent index, which can be defined if a table has a bitmap-compatible IDKEY (as is the default). The CREATE TABLE DDL Statement automatically defines a bitmap extent index on the table it creates. For information about adding an extent index to a table defined through a persistent class in ObjectScript, see Define SQL Optimized Tables Through Persistent Classes.

2 Decide on a Storage Layout

Tables can make use of either columnar or row-wise storage or employ a combination of both strategies. Each of these strategies can be highly effective when utilized in the proper setting. A columnar storage layout is recommended on data that requires frequent filtering and aggregating operations to perform analytical queries on large amounts of data. A row-wise storage layout is recommended on tables where you want to select small sets of rows at a time and on tables that will experience frequent inserts, updates, and deletes of data. For more information about storage layouts, see Choose an SQL Table Storage Layout.

3 Leverage Table Statistics

Table statistics, like ExtentSize, Selectivity, and BlockCount, describe the distribution of data within the table. The SQL Optimizer is able to make use of these insights to properly determine which query plan will run fastest.

The TUNE TABLE utility collects these statistics that are essential to the performance of SQL queries and should be run when the table has been populated with a representative quantity of real data.

For more information about how this command works, see Table Statistics for Query Optimizer. You should also run TUNE TABLE when the distribution of values in the columns changes significantly after adding a sizeable amount of data.

4 Configuration Optimization

By default, the **Memory and Startup Settings** default to **Automatically** configured, and the **Maximum Per-Process Memory** defaults to –1, which denotes unlimited use. When configuring a production system, you should verify if any of the other Memory and Startup Settings should be tuned based on this guide. For further details, refer to Memory and Startup Settings in the Configuring InterSystems IRIS page.

5 Examine INFORMATION_SCHEMA

The INFORMATION_SCHEMA schema packages information about the schemas, tables, indexes, views, triggers, Integrated ML models, and SQL Statements that exist on an instance of InterSystems IRIS®. You can efficiently view crucial information about your SQL configuration, including distribution of data through a table and performance metrics, by issuing SELECT statements against the various tables in the schema.

In particular, you may consider querying the INFORMATION_SCHEMA.STATEMENTS table to view SQL Statements or the INFORMATION_SCHEMA.COLUMN_HISTOGRAMS table to view the distribution of data in columns of a table.

Information about these tables, including what columns you will find in them, can be found in the class reference for INFORMATION.SCHEMA.

6 Troubleshoot Query Performance

Monitoring performance statistics and runtime statistics of queries that are run on the system can provide insights into what further optimizations you might want to make. It is best practice to periodically monitor statistics for newer queries in order to determine if they have sub-optimal performance and where their efficiency can be increased.

InterSystems provides multiple tools for such monitoring. To monitor queries as they are being executed, use the SQL Activity page in the Management Portal, found by navigating to **System Operation** > **SQL Activity**. To view performance data of historical queries, you should examine the SQL Runtime Statistics. The Statement Index offers an easy interface to browse all of this information and review SQL statements that represent a high load on the system, based on the combination of times executed and execution time. When you have identified such statements, you may consider adding additional indexes to speed them up.

If you notice some undesired behavior (for example, unexpectedly high values for TimeSpent or GlobalRefs or both), you should look at the query execution plan that the system has generated to understand how it is executing the query.

For a more concerted analysis effort of such statements, you may take advantage of the utilities in the SQL Performance Analysis Toolkit.

If you cannot determine the cause of inadequate performance, contact InterSystems Worldwide Response Center (WRC) by using the Generate Report tool to submit a query performance report.