

# CASE STUDIES IN PERFORMANCE

---

---

## Executive Summary

The best way to compare the performance of database products is in a head-to-head test using a real application, preferably one of your own. This is especially true when evaluating Caché's post-relational technology, because "standard" transaction processing benchmarking methodologies assume the restrictive "row and columns" format of a relational database. They cannot accurately predict the performance of real applications, which often use complex data models.

Because of contractual prohibitions imposed by well-known database vendors, there is very little published data from "real world" performance tests using relational databases such as Oracle and Microsoft SQL Server. Companies who perform such benchmark tests on their own typically find that Caché outperforms relational databases by a factor of five or more, even without changing application code and just using SQL. This performance differential often increases when applications are optimized to take advantage of Caché's post-relational technology.

---

A technical white paper by:

Mary Finn  
Product Marketing Manager  
InterSystems Corporation

# CASE STUDIES IN PERFORMANCE

## Introduction

---

Many companies perform head-to-head comparisons of Caché and other databases. This paper presents the results of performance benchmarks performed by independent analysts and by InterSystems' customers and prospects as they evaluated their database technology options. Because of the contractual restrictions imposed by the large relational database vendors, the names of the companies conducting performance benchmarks have been omitted, as have the names of the relational databases that competed against Caché.

These tests were performed using real applications and real data under real-world conditions. The results are conclusive. Using SQL to query the databases, Caché typically outperforms Oracle, SQL Server, and other relational databases by a factor of five or more.

In most of the studies presented here, SQL was used as the query language because that makes it easier to conduct a head-to-head test of Caché and relational databases. In these cases Caché's superior performance is due to its multidimensional architecture, which eliminates the processing overhead needed to perform "joins" across multiple tables. However, SQL is not the only available method of querying a Caché database. Caché runs even faster – potentially several times faster – when the code is optimized to directly access Caché's multidimensional data structures.

## Case Study #1

---

A power supply company found Caché to be 5 times faster than a well-known relational database in their data warehousing ETL (Extract, Transform, Load) application. They tested both databases using the field order module of the application, processing over 13 million source records. Both databases used the same table definitions and queries. However, the existing field order module relied heavily on database-specific stored procedures, so those procedures were re-implemented using the Caché stored procedure language. (This only required 40 person-hours of work.)

The following table summarizes the benchmark results:

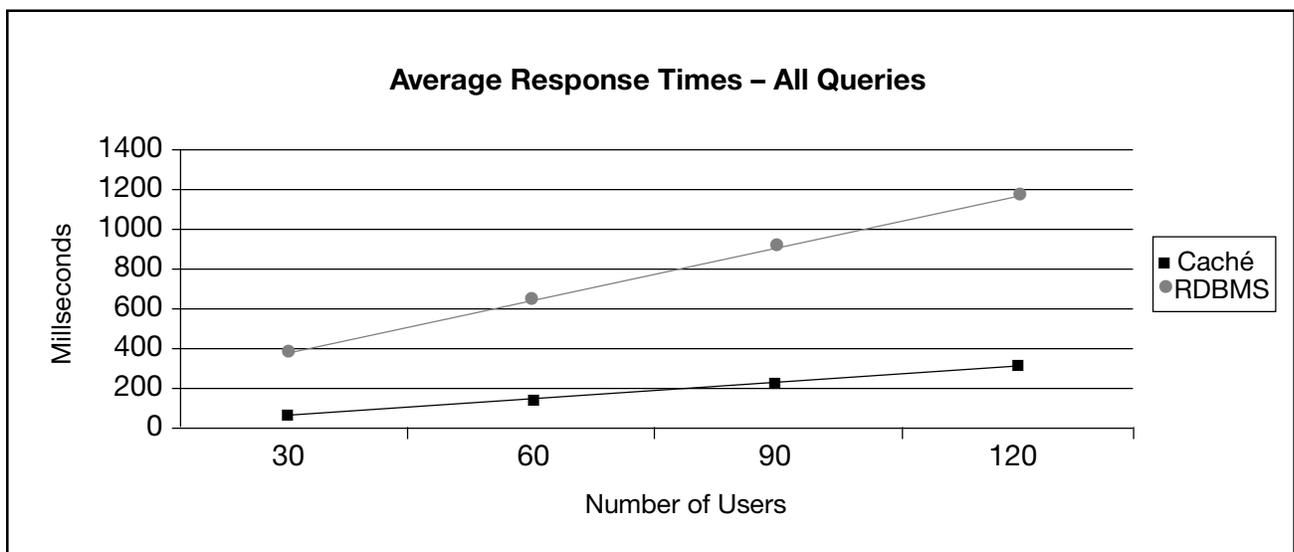
Task	Elapsed Time		Relative Performance
	RDBMS	Caché	Caché : RDBMS
Load data into staging table	138 min.	6 min.	23 : 1
Load second staging table and filter data	245 min.	23 min.	10 : 1
Generate field order dimension table	168 min.	24 min.	7 : 1
Generate field order fact table	890 min.	233 min.	4 : 1
Total time to load and filter data	1441 min.	286 min.	5 : 1

## Case Study #2

A healthcare application provider conducted a test to compare the SQL response times of Caché and a well-known relational database. Using historical patient data (7 tables, over 6.5 million records), a battery of 8 queries was run with simulated loads of 30, 60, 90, and 120 concurrent users.

The following table gives the average response times in milliseconds for both databases. Both were fast, but Caché averaged from 4 to 6 times faster. In addition, a graph of the results shows Caché to be more scalable. For instance, when the load is expanded from 90 to 120 users, Caché's response time increases by only 84 ms, while the relational database slows down by 231 ms – 2.7 times as much.

# of concurrent users	Average Response Time (over all eight queries)		Relative Performance
	RDBMS	Caché	Caché : RDBMS
30	375.125 ms	59.125 ms	6.3 : 1
60	637.25 ms	137.75 ms	4.6 : 1
90	915.625 ms	206.875 ms	4.2 : 1
120	1146.375 ms	290.125 ms	3.9 : 1



### Case Study #3

---

A consultant compared Caché and a well-known relational database, to measure performance in a data warehousing scenario. Using a million-record table of real order-processing data, he measured both validation/load performance and query response times.

Even though the Caché data structures were not optimized for performance, the validation/load of 1,000,000 records was three times faster (2681 seconds vs. 8597 seconds) with Caché than with the RDBMS. Furthermore, the resulting Caché database required only 146 MB of disk space compared to 216 MB for the RDBMS.

The results of the SQL query response tests are given below:

Query	Response Time		Relative Performance
	RDBMS	Caché	Caché : RDBMS
Large aggregation on the whole table with all dimensions	638 sec.	180 sec.	3.5 : 1
Aggregation on all dimensions with restrictions on 2 dimensions	3 sec.	<1 sec.	>3 : 1
Find unused Purpose codes across whole table	8 sec.	3 sec.	4 : 1
Find all records where one dimension is greater than a specified number	50 sec.	16 sec.	3.1 : 1

## Case Study #4

---

A nationwide telephone company, using their on-line phone book application to test performance, discovered that Caché averaged 10 times faster than a well-known relational database.

Loading the relational database from text files took approximately 10 hours and required 60 GB of disk space. This compared to approximately 45 minutes and 2GB of disk space for the Caché database. Because the Caché database was dramatically more efficient in its use of space, it made much better use of cache memory. As a result, the query response time was much faster for Caché than for the RDBMS:

Query	Response Time		Relative Performance
	RDBMS	Caché	Caché : RDBMS
Search by name, street	.33 sec.	.07 sec.	4 : 1
Search by name	.46 sec.	.02 sec.	23 : 1
Search by name, street, zipcode	.76 sec.	.04 sec.	19 : 1
Search by house number	.14 sec.	.02 sec.	7 : 1
Search by, name1, name2, street, zipcode	.01 sec.	.01 sec.	1 : 1
Search by county	.02 sec.	.01 sec.	2 : 1
Average for all queries	.287 sec.	.028 sec.	10 : 1

Indexes were used on both databases to increase query performance, and even though the RDBMS was slower than Caché, it still delivered sub-second response times. However, updates to the RDBMS (and the necessary rebuilding of the indexes) must be done off-line to avoid significant performance problems. The application is actually designed to use two relational databases – one “live” while the other is updated – swapping them out day by day. In contrast, the “live” Caché database could be updated without significant performance degradation. In addition, Caché enabled doing “begins with” searches, which the RDBMS could not do.

## Case Study #5

---

A solution provider to the telecommunications industry was asked to develop a short messaging system (SMS) event tracking application. The application would need to accept high volumes of data from several remote sources, index the records, and write them to a central database on disk. Using a prototype of the application, the solution provider compared the performance of Caché and a well-known relational database.

In an initial test, both databases were tested using the same code. There was no other processing load on the system. The results are shown below:

Query	Average records/second processed		Relative Performance
	RDBMS	Caché	Caché : RDBMS
Load, index, and write data	315	1,700	5 : 1

Based on these results, efforts to implement the application using the RDBMS were abandoned. When the code was optimized for Caché's native multidimensional data structures the application performance increased by a factor of six. Using multiple-stream input, and simulating a processing load on the system, the optimized Caché-based application achieved an average processing speed of 10,600 records per second.

## Conclusion

---

The best true measure of database technology is how well it performs in real-world applications. The case studies presented in this paper document head-to-head performance tests of Caché and well-known relational databases. The tests were conducted by independent analysts and InterSystems' customers and prospects using real applications. Results show that Caché typically responds to SQL queries 5 times faster than relational databases. Additional performance increases can be realized by altering application code to directly manipulate Caché's multidimensional data structures.

## Additional Reports

---

For more comparisons of Caché and relational databases, see:

- “A Summary of Application/Database Performance”, by KLAS Enterprises
- The IDC report: “Breaking the Relational Barrier: User Data Management Triumphs With InterSystems’ Caché”, by Carl Olofson

You can read these reports at [www.InterSystems.com](http://www.InterSystems.com), or request a hard copy by sending an email to [info@InterSystems.com](mailto:info@InterSystems.com).

## Perform Your Own Benchmark

---

InterSystems frequently helps customers and prospects perform benchmark tests like the ones outlined in this paper. If you wish to conduct your own test of Caché’s performance and scalability, InterSystems can help by providing technical assistance and temporary Caché licenses at no charge. For more information, call 1.800.753.2571.

InterSystems Corporation  
World Headquarters  
One Memorial Drive  
Cambridge, MA 02142-1356  
Tel: +1.617.621.0600  
Fax: +1.617.494.1631  
[www.InterSystems.com](http://www.InterSystems.com)

**INTERSYSTEMS**