



## The Quick Response Database By QD Technology

A Smart Relational Database Management Solution that  
Enables Easy and Fast Ad Hoc Queries

*Contact:*

Aaron Shapiro

Founder/CEO

973.747.7500 *mobile*

973-943-4137 x117 *office*

973 473 2032 *fax*

668 Passaic Avenue

Clifton, NJ 07012 U.S.A.

[ashapiro@qdtechnology.com](mailto:ashapiro@qdtechnology.com)

# Table of Contents

Executive Summary .....	2
Introduction .....	3
The Quick Response Database .....	4
Increased Data Accessibility .....	4
Accelerated Query Speed .....	4
<i>Example:</i> .....	5
Additional Benefits .....	6
Data Security .....	7
<i>Protection against unauthorized access:</i> .....	7
<i>Deterrence against misuse and abuse of data:</i> .....	7
Applying QD .....	8
Summary .....	8
Appendix 1: QD Architecture Overview .....	8
Appendix 2: Data Compression Techniques .....	10

## Executive Summary

QD Technology's™ Quick Response Database™ (QD™) is a breakthrough relational database management solution. QD is a quick, smart, RDBMS that delivers a secure, compressed, portable, up-to-date, read-only copy of a production database to the desktop or laptop computers of users such as database analysts.

While delivering high-speed query performance, QD also brings large volumes of enterprise data to a new level of accessibility, enabling enterprise database users to achieve greater efficiency and productivity than were previously possible. Because QD gives users a local copy of their database separate from the active production copy, queries can be made without disrupting production IT operations. QD allows analysts to develop, test, and execute complex iterative queries or to send ad hoc queries to the database whenever necessary. As a result, QD delivers a powerful yet cost-effective solution to today's needs for data access, querying, and reporting.

QD Technology is also well aware of the security concerns that surround the placing of critical databases on laptops. For this reason, QD includes the elective application of military quality encryption to any database, as well as password-protected access to the database, and the ability to lock down or destroy the database after a particular date, a number of unsuccessful access attempts, or after a preset interval during which the client database has not authenticated itself to the server (heart beat policy with poison dart). If database destruction is actualized, the database is removed using DoD style deletion.

QD supports all standard SQL92 query syntax as well as common query tools that your analysts currently use. It can read and compress data from all of today's industry leading databases including Oracle, Sybase, DB2, Microsoft SQL Server, and MySQL across all of the major operating system platforms along with flat files and system log files. Current releases of mission-critical applications such as Business Objects, Cognos, Hyperion, SAS, and Crystal Reports can all run against the QD database.

QD's additional functionality and performance can simplify the corporate database infrastructure by removing the need for data marts and cubes and reducing the day-to-day network load. QD is simple to implement, manage, and maintain as it eliminates the need for additional database administrators.

In this White Paper, we introduce QD and discuss how its key features deliver significant benefits to enterprises that wish to realize increased secure data accessibility or improved query speed.

# Introduction

In 2007, a Gartner survey reported that 1400 CIO's identified business intelligence (BI) as their #1 priority, and Gartner analysts predicted that more than \$20 billion will be spent annually on BI services by Y2010. Yet, while the demand for BI has been evolving for nearly a decade, presently fewer than 10% of today's corporate data users have access to the data they need, and the majority of users report that it takes more than three to five days to obtain a report from IT.<sup>1</sup>

Business intelligence – the ability to gather and analyze data for the purpose of improved business decision making – is growing more important --albeit more challenging-- with each passing year. Online storage requirements are growing by 100% or more per year<sup>2</sup> while widespread use of data mining and analytics is obstructed because corporate data is often widely fragmented and dispersed across many different locations and databases.

Compounding the problem is the bottleneck of database performance. In many IT organizations, key databases are run in only a single location and on a single set of database servers. Running multiple copies of the same database can be extraordinarily difficult, if not impossible. As a result, day-to-day production which requires adding and modifying transactions consumes all available computational resources leaving no time or resources for queries and analysis. Corporate policies often restrict significant database queries to evenings and weekends to help alleviate resource conflicts between writing and reading queries. Days-long queries are unacceptable, and analysts are given few, if any, chances to build complex queries incrementally through trial and error. In many enterprises, the opportunity for an analyst to come up with a new question and obtain a prompt answer is almost non-existent.

Furthermore, traditional database architecture requires that analysts who wish to run database queries be connected to the corporate network. As a result, road warriors such as remote salespeople or traveling executives without reliable network connections are completely cut off from the database. Salespeople who can query the database just before or during a sales call will be much better able to address their customers' concerns and answer their questions in a timely manner.

A new approach to database management and analysis is required for enterprises to realize the greatest possible value from their massive collections of critical data.

---

1) Database Trends and Applications, Volume 22, Number 1, March 2008

2) Peter Lyman and Hal R. Varian, "How Much Information 2003?", School of Information Management & Systems, University of California, Berkeley, 2003. <http://www2.sims.berkeley.edu/research/projects/how-much-info-2003/>.

## The Quick Response Database

The Quick Response Database from QD Technology is a smart relational database management system (RDBMS) that solves many of these problems. With no impact to production service, QD makes it simple to run SQL queries against a local read-only copy of a complete database, selected tables, rows, or columns from within a database, a batch of delimited flat files, or a collection of disparate databases. QD allows these queries to run to completion, often many times faster than could be achieved by their production RDBMS counterpart.

QD achieves these remarkable results through a combination of features designed to support two key product objectives: data accessibility and query speed. A review of these two objectives is followed by a description of QD's other additional benefits and its provisions for data security.

### ***Increased Data Accessibility***

- QD uses *smart compression* (data-dependent compression techniques) to reduce the size of a user-specified production data set to roughly one-third to one-tenth of its original size, thereby enabling the data to easily fit on a user's desktop or laptop. QD then delivers a complete, read-only copy of the compressed data onto the user's PC making it available for dedicated analysis 24/7 with no further network connectivity required.
- Additionally, QD can combine tables from multiple databases into a single QDB (a physical Quick Response Database) which can then be queried as a single database.

Databases can be combined even if they originate in different vendors' RDBMS's or on different computing platforms. Entire databases can be combined, or specific tables can be selected for inclusion in the target QDB.

- Utilizing its SQL pre-processing feature, QD can also serve as a basic ETL solution for creating database tables containing only the columns, rows, or transforms needed from the original data source.

### ***Accelerated Query Speed***

Traditional data warehouses and transactional production databases are limited in the levels of data-analysis-and-query-only performance they can deliver. This is attributed to the complex IT environments they are required to support and by the inherent added complexity required to support database write transactions e.g. record locking, transaction logging, index and table space management, etc. These database capabilities that are "must haves" from a database updating perspective actually become liabilities from a query-only perspective as they serve to impose unnecessary overhead and delay query performance.

QD, conversely, is oriented specifically for read-only applications. By design, it accelerates query performance through a combination of techniques:

- QD's *smart compression* technology enables the retrieval of far more data per disk read request than would otherwise be possible, thereby accelerating data queries and searches.

First, smart compression allows only those columns that are necessary for determining the query results to be retrieved from the database. This prevents memory from being wasted on columns that do not contribute to the determination of the solution set.

Second, it looks at each data column individually in order to select the algorithm that will yield the highest compression ratio for that column. The greater the compression factor, the more data can be read per disk access. The result: faster searching and faster query performance.

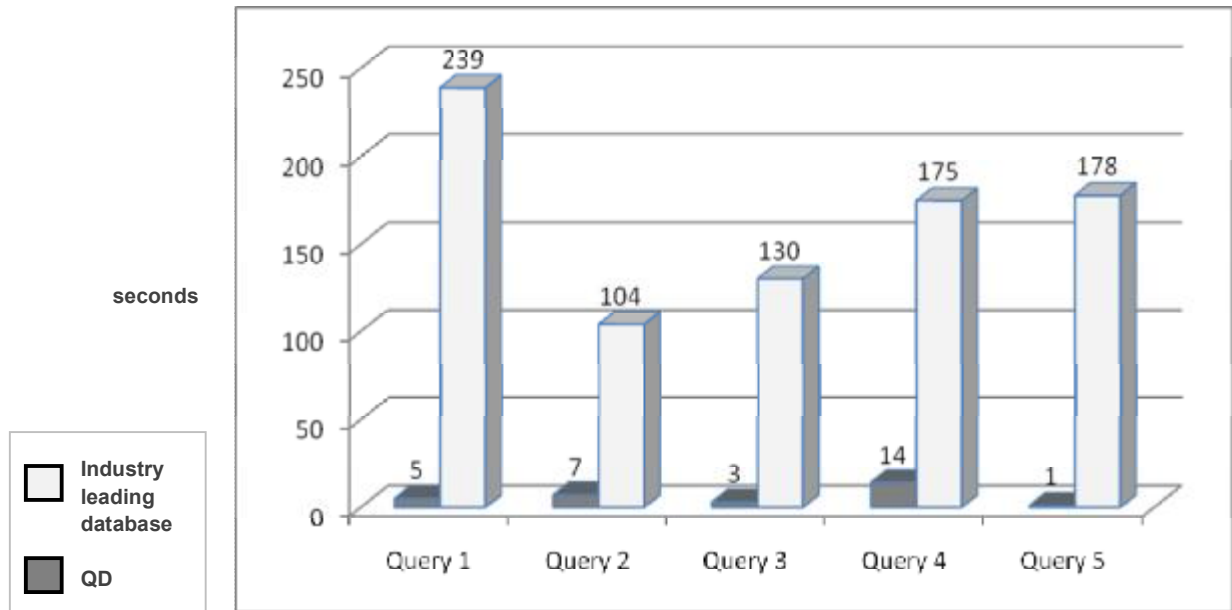
- QD's optimized query engine operates on the data *without decompressing interim results* in order to evaluate columns against the SQL WHERE clause criteria. Only the final result set is decompressed.
- QD eliminates the overhead of logging, locking, and other processes required by a writeable database to maintain consistency and manage multiple users.
- By offloading query processing from the production platform, QD eliminates contention for network, server, and database resources.
- QD provides more complete indexing of the database.

***Example:***

Analytical ad hoc querying is generally performed upon rich data sets (many columns per row) often based upon small subsets of criteria. This is precisely the kind of data/analysis model against which QD databases perform best.

Below is one such example of the kind of query performance customers have achieved using QD.

**Example:** Query speed against a database of over 20 million rows of demographics data.



#### U.S. Demographics Database

- Query 1: Average number of adopted children per thousand households
- Query 2: Percentage of housing units that are solar-heated, ranked by geographic region
- Query 3: Percentage of mortgage-paying households that have a monthly mortgage payment of \$3000 or more
- Query 4: The 10 highest paying occupations
- Query 5: Ratio of single-parent mothers to single-parent fathers ranked by state

### **Additional Benefits**

- By executing all queries against a local copy of a production data set, QD also reduces contention for server resources. In some cases, this has resulted in an improvement in the performance of the master database on the original database server.
- Where a large data warehouse is partitioned across a second tier of data-mart servers, QD can potentially reduce the amount of required second-tier hardware, database software, and associated support resources. By effectively transferring the data-mart server function right onto the user's PC, QD can provide a simplified and cost-effective alternative to an existing complex architecture.
- Due to the product's simple installation process and its operational ease of use in creating and distributing databases, QD requires no specialized DBA skills. Virtually any technical user who is competent in connecting to and running queries against databases is capable of installing and using QD.
- In the event of a corporate IT disaster, workers who utilize QD as their data source will still be able to function. In an extreme emergency, selected production database tables can be rebuilt from their compressed QDB.

## ***Data Security***

The QD Technology engineering team has always been aware that placing data on a laptop entails special security risks. From the product's inception, QD was engineered with technical safeguards at multiple levels to provide protection against unauthorized access as well as to deter against the misuse and abuse of data by users who *are* authorized.

### ***Protection against unauthorized access:***

- At the user level, access to a QDB is controlled by requiring user authentication consisting of a valid user name and password. A QD administrator can further configure a QDB for even more effective security when unauthorized usage is suspected; – for example, configuration can be specified so that upon a configurable number of consecutive invalid login attempt, the QDB is wiped (i.e. obliterated not just removed from the directory listing) from the disk.
- At the hacker level, QD also offers two layers of protection against direct examination of the physical disk sectors that store a QDB. First, as a natural consequence of smart compression, QDBs will be inherently unintelligible to all but the most sophisticated and determined of hackers. Second, for situations where even the most remote security threat must be mitigated, a QD administrator can further configure a QDB to utilize AES 256 bit encryption – the same algorithm approved by the U.S. Government for encryption of top-secret information.
- QD offers a patent-pending database security feature on the client database to mitigate any possibility of a database falling into the wrong hands The heart-beat with poison dart feature enables the administrator to set an interval within which the client database must successfully authenticate with the server application. Once the interval has expired and the authentication has failed, the database is automatically removed from the client machine, or locked down for access (depending on the configuration setting for that database/user)

### ***Deterrence against misuse and abuse of data:***

- To deter against misuse and abuse of data by authorized users (or to support traceability of such suspected activity back to its source) QD provides an audit feature that captures and records critical events executed against a QDB. Audit information includes all actions executed against a QDB, including the actual SQL queries executed, and the respective user id's of the individuals who executed them. All audit data is encrypted and automatically transmitted on a scheduled basis to the QD Server and loaded into a QDB for administrative reporting access.
- The administrator can control the period of time a database can 'live' on a client machine by creating an expiration date for the QDB. When a user attempts to read the database beyond the expiration period, the database is either rendered inaccessible, or wiped of the disk. A database wipe is comprised of a DoD deletion algorithm which mitigates the possibility of retrieving deleted data from a disk.

## Applying QD

The value of any new technology comes from applying it to real life situations where it has the chance to improve upon methodologies employed in the past. Can the new technology make systems, people, and processes more efficient, and ultimately improve the “bottom line”?

QD can expedite old methods of accessing and querying data, and it can increase the efficiency of analysts. For examples of how organizations can leverage the capabilities of QD, please see the Business Examples available on our web site at [www.qdtechnology.com](http://www.qdtechnology.com).

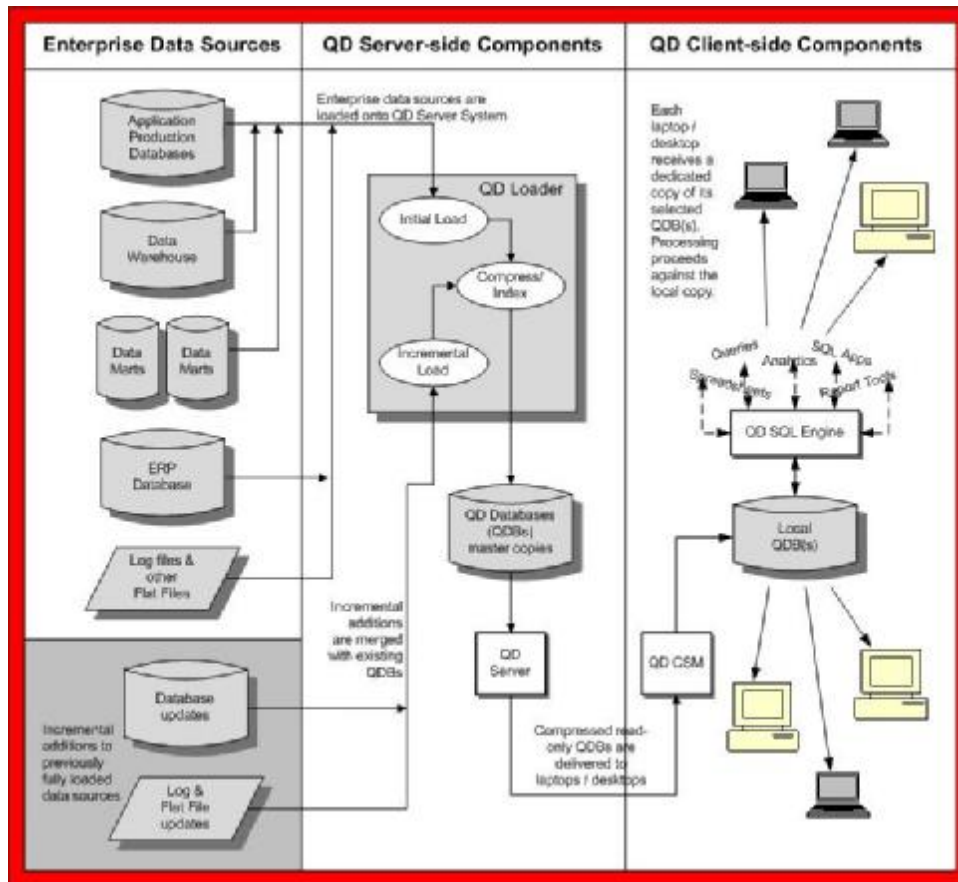
## Summary

QD Technology’s Quick Response Database is a unique technology that combines secure database distribution with compression to deliver high-speed read-only database access. It will allow your users to develop and perform complex queries more quickly and easily than ever before.

The applications of QD are broad; QD can be used in almost any enterprise that has a large and busy transaction-oriented database such as applications in financial services, healthcare, military, public sector, data providers and marketing organizations.

## Appendix 1: QD Architecture Overview

An overview of the QD architecture is illustrated in the diagram below, and a description of its key components follow. Note that not all of the components listed are explicitly shown in the diagram.



The QD Administrator is a GUI application that runs on the QD Server System. It is used to define users, roles, and databases for authentication and authorization within QD.

The QD Loader automates the decision process for loading, compressing, and indexing the data. It reads the entire source database and examines each column in every table and schema to decide which indexing and compression methods to use for each part of the database. The Loader supports both full and incremental loading of the specified data sources.

The QD Server is a continuously running TCP/IP server application on the QD Server System whose purpose is to handle client requests for QDB downloads. The Server establishes the authority for the user to access databases on the QD Server System machine. Once it has authorized a user, the Server enables replication of new databases and updates to existing databases on the user's Client System.

The QD Client System Manager (CSM) is a GUI application that runs on the QD Client System and supports the definition and replication of the CSM user's local QDB. It connects via a TCP/IP secure SSL link to a constantly running QD Server application on the QD Server System.

The QD SQL Engine determines the most efficient execution plan for accessing a QDB's data in order to answer a specified SQL query. It then performs the optimized execution

plan and provides the desired query result set.

The **QD Notifier** alerts the QD Client System user of available QDBs residing on the QD Server System.

## Appendix 2: Data Compression Techniques

As mentioned earlier in this white paper, QD implements *smart compression* to determine the best method of compression for each column. Methods of compression include, but are not limited to:

- LZO:

Lempel-Ziv-Oberhumer (LZO) is a popular data-block compression algorithm that includes the ability to adjust the balance between compression quality and compression speed.

- RLE:

Run-length encoding (RLE) is a technique that stores sequences of consecutive data elements as a single data value and count.

- Date Flooring:

A *number floor* is a minimal value for data stored in a column. This floor is subtracted from all of the data in the column and stored separately. During the decompression process, the floor is added back into the data.

- Tokenization:

*Tokenization* is the process of demarcating and possibly classifying sections of a string of input characters. Data may be compressed by substituting for tokens if the data lends itself to this form of data compression.