



ActiveKnowledge™

An Innovative Performance, Access Control,
and Scalability Solution for the Enterprise

White Paper

Call: +972-3-7522685
Toll free: 866-300-402
Email: info@active-base.com
Visit us: www.active-base.com

Executive Summary.....	3
Database Growth Trends.....	4
Existing Strategies for Performance Improvement.....	5
ActiveKnowledge™.....	5
ActiveKnowledge Results.....	6
How does ActiveKnowledge accomplish these results?	6
Topology	7
Functional Layers.....	7
Application Connection Switching Layer	7
Optimisation Rules Layer	8
ActiveKnowledge SQL Expert Layer	8
Configuration.....	8
ActiveKnowledge - A Highly Available Solution	9
Examples from Current Production Implementations.....	9
Performance Improvement in a Business Objects System	9
Performance Improvement in a Chordiant CRM System	10
Access Control Enhancements from a Banking System	10

Executive Summary

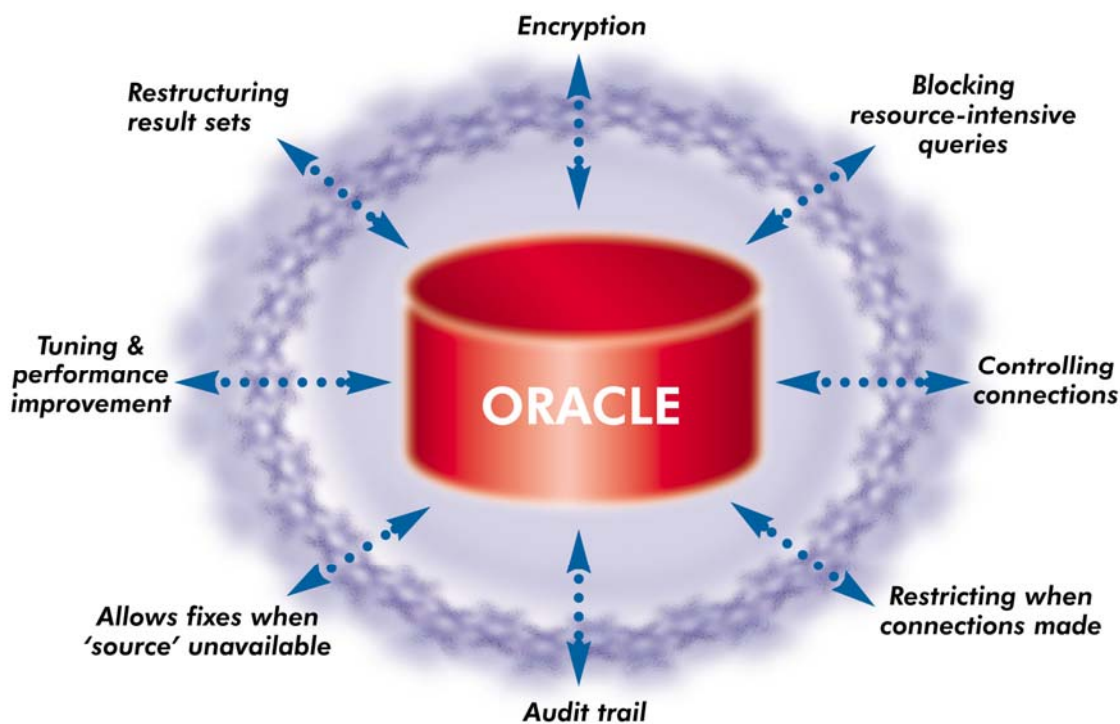
Data is your one of your most valuable assets.

You have invested very considerable sums of money in creating your data warehouse. You spend countless hours of processing time loading data into it, cleansing it, backing it up, and generally supporting and managing it.

But what do you get back from it. Is your data warehouse a frustrating animal that soaks up all the data being thrown at it, but rarely gives it back when you need it?

Part of the reason for this is the un-stoppable demand for the valuable information held within the warehouse that can drive your business forward. The volume of queries and reports continues to spiral ever upwards. The complexity of those reports increases relentlessly, as the problems that the warehouse is being used to solve become ever more complex.

Your data warehouse truly is a key resource, and must be protected, and ActiveKnowledge can provide that protection.



ActiveKnowledge can fulfil the functions of a 'SQL firewall', silently intercepting SQL on the way to the database, analysing the SQL, and then taking appropriate action. It provides benefits in two main areas.

Firstly it has a role in data access. It can be used to prevent access from unauthorised people or applications. It can then control the data to be returned to the originator, by either blocking individual queries whilst returning suitable error messages, or by amending the queries to change the selection clauses so as to only return authorised data. This is all carried out invisibly to the originator of the query – he simply sees what he is permitted to see, unless you desire to inform him otherwise. Where particularly sensitive information is queried, then it may be encrypted, or scrambled prior to display to the originator. An audit trail of this activity is maintained at the same time.

The second main area of benefit in protection of a data warehouse is in terms of performance and capacity. Many query tools can now generate exceedingly complex SQL queries, but unfortunately much of that SQL is very inefficient. It is a challenge however to rectify that problem, as much of it is generated on-the-fly. If it is amendable then it is a mammoth task to take all the queries and reports, and manually tune them to their optimum. ActiveKnowledge again provides a solution. It has a powerful SQL hint generator to allow the rapid determination of the most efficient SQL alternative to the constructed query, and can then apply that performance enhancement in real time as the SQL travels through ActiveKnowledge to the database. It does not require any changes to either the source SQL or the target database. It also permits you to manually tune the SQL using your own expertise if that is desirable. If a query should not be allowed at a particular time, because sufficient resources are not available, then that can also be managed through ActiveKnowledge by temporarily blocking, or delaying the query.

ActiveKnowledge allows you to make a step change in the performance of your data warehouse. It allows you to control who has access to what data, and for you to know who has been accessing what. It is simply installed, and in days can improve the performance by tens of percentage points. It can even do this when you cannot access the source of those queries. It permits considerable cost savings to be made by improving the overall efficiency of the warehouse, and potentially frees up resource that was being strangled by inappropriate and purely tuned reports. It allows you to restrict what data can be seen by whom, and when.

Read more to find out how ActiveKnowledge can achieve this for you, and how it allows you to take control of **your** data warehouse now.

Database Growth Trends

Over the past few years, organisations have made enormous investments in applications designed to store every piece of information in a database. They have deployed various applications managing every aspect of business operations, creating multi terabyte-sized data warehouses, and spent considerable sums of money on multiprocessing servers powerful enough to process the resulting massive volumes.

Even the most powerful database servers cannot always keep pace with the demands imposed by scores of financial analysts, marketing researchers, strategic planners and customer relationship managers.

As a result, many organisations are literally choking on the terabytes of data and processing loads that drive resource usage beyond acceptable limits, and exceed system capacities. System performance slows to a crawl, with a corresponding increase in the number of problems being reported to IT departments, and indeed complaints being made as a result.

This problem can be exacerbated by the ability of modern report generators and similar products to generate and issue complex ad-hoc queries. Often the originators of these queries are completely unaware of the complexities, nor the affect they are having on the system performance as a whole. Similarly many of the IT departments are similarly constrained by insufficient manpower and resources to be able to thoroughly investigate the situation, and restore performance levels.

Confronted with this challenge, IT managers are looking for solutions that improve application performance and availability from a simple and seamless 'out-of-the-box' approach; one that imposes minimal additional work load on the existing IT workforce, and can be immediately installed with no hidden modifications or maintenance costs is most likely to gain approval..

They seek solutions that have a profound impact on performance, stability and availability, and that increase efficiencies of past infrastructure investments - an immediate return on investment with minimal risks involved.

Existing Strategies for Performance Improvement

Database administrators enjoy a few alternatives when it comes to improving processing performance.

The hardware upgrade ('big hammer approach') strategy of upgrading the database server hardware is the ultimate solution when budget is not an issue. This can vary between expanding existing hardware by the addition of further CPUs, memory, and other components, and the associated software license costs that are often dependent on the hardware components, up to implementing complete replacement servers. This approach will therefore usually demand large capital allocations. As can be readily understood the planning, analysis, installation and testing phases to ensure this approach is successful are often lengthy, with the resultant application pain continuing, and not without risk.

As a consequence organisations try to use other performance improvement solutions in order to benefit from their existing infrastructure for a longer period of time.

The Alternate Database approach (replication) is an approach that copies parts of the database to a replica. These replicated databases are used by Business Intelligence tools that do not necessarily require real-time and absolutely up-to-date information. Replication mechanisms usually require purchasing additional hardware and software, and the management and overhead of data synchronisation between the different replicas. In addition, replication requires the management of access to the different databases by different users running different applications. This solution is expensive, complex and requires on-going maintenance and tuning.

It can however be very effective.

Tuning the Database approach (Performance Management Tools). Using database and application tuning tools is a solution that many enterprises deploy. In this approach use is made of monitoring tools which can help to pinpoint and make changes to database and application parameters, such as the size of general storage areas and local pools, and recommend making changes to the database indexing strategy. This can provide timely but usually limited performance relief. On the other hand other application monitoring tools can identify which sets of application requests cause most of the overhead on the database server and the longest application response times. They can even suggest corrections for these statements in a way that the query can be executed more efficiently whilst preserving the SQL results. Even then, however the support teams may be unable to correct the problematic requests for various reasons; the application may be a closed package, the development staff to amend and retest are not available, or maybe because the requests are reports and ad-hoc queries created and run by an end user in business intelligence environments, over which there is no control.

ActiveKnowledge™

ActiveKnowledge is a simply and easily deployed software package which speeds problematic reports and ad-hoc queries on their way to the database, tuning them and applying performance optimisation policies in real-time.

Installed transparently between applications and databases, it boosts performance by identifying in real-time problematic SQL requests on their way to the database, analysing and correcting them automatically – by applying predetermined and tested rules - without making any changes to the original applications or databases.

Requests that are thought to place too heavy a load on the system may be blocked before reaching the database, with the ability to return customised notification to the originating user. Blocking these statements is one way to prevent performance degradation.

Other optimisation rules may include the addition of database hints to control the Oracle database optimiser, or the complete re-writing of a statement using a known and tested better alternative.

ActiveKnowledge can also be used to enhance the access control to, and security of an application. By intercepting queries and reports on the way to the database, checks can be made to ensure that the originator of the query is authorised to view the data being requested, or to remove data from the query that the originator is not authorised to see. Similarly sensitive information could be, for example, scrambled to prevent leakage of that information.

This can all be achieved whilst providing detailed usage patterns and audit-trails of user access and requests.

ActiveKnowledge can typically be installed and configured in less than an hour, without any modifications imposed on clients, applications or databases. The installation into existing database infrastructure is simple, quick, and easy, and does not require any installation downtime or application customisation.

ActiveKnowledge Results

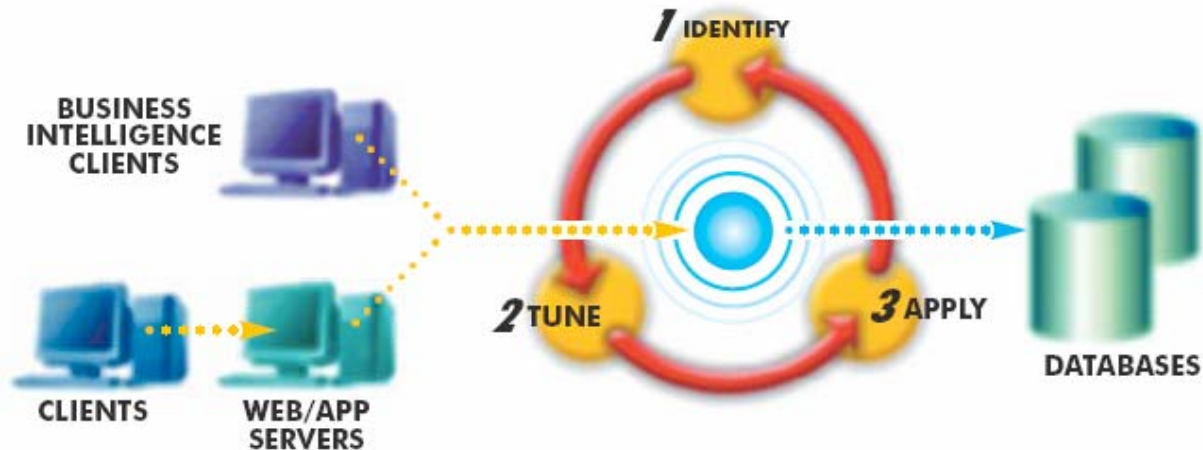
- Improved response times for queries, reports and requests by up to 80%, using an "out-of-the-box" approach. This profound impact translates directly into significant improvements in business performance, with no added demands on end users.
- Increased database capacity achieved, allowing the support of additional data and query activity from existing hardware and software resources.
- Improved visibility of user, application and database object usage patterns using detailed audit trail and application activity, reducing licensing costs and improving security.
- Performance gains can easily be achieved by Database Administrators and other non-expert users including report writers, developers and super users.
- Reduced expertise is required for improving performance thereby speeding the process of performance improvement.

How does ActiveKnowledge accomplish these results?

ActiveKnowledge operation automates and streamlines the following actions:

1. **Identify** – ActiveKnowledge provides facilities to identify long running queries, problematic requests and reports. It can provide a full audit trail on all SQL statements and database execution plan operations used. It also complements the use of other performance monitoring tools, in the identification and gathering of problematic SQL statements. It allows for the input of those previously identified statements, or merely those reported by the end-users as being of concern, into the second area of the product, for more detailed analysis.
2. **Tune** - ActiveKnowledge SQL expert provides the facilities to tune problematic statements by evaluating the addition of database 'Hints', based on the underlying Oracle execution plan and cost. It provides the facilities to flexibly benchmark the previously identified alternative statements, comparing resource utilisation as it goes. Finally the most desirable variant can be simply converted into an optimisation policy, and applied to future SQL queries. Alternatively skilled IT staff can provide alternative SQL statements themselves, using their knowledge and experience of the database and applications to further refine and tune the SQL queries and reports, compounding the benefits achieved.
3. **Apply** - ActiveKnowledge automatically intercepts previously identified SQL statements in real-time and can flexibly apply the predetermined optimisation or security policies, before forwarding the statements onwards to the database or rejecting them altogether. This allows for the re-tuning of poor performing

queries on-the-fly, without intrusively 'recoding' the applications or manually rewriting ad-hoc queries and business reports.



ActiveKnowledge can be installed either on the database server or on a separate physical server, whichever is most desirable. Each ActiveKnowledge server can manage multiple different applications and databases, and similarly each ActiveKnowledge server can be managed by multiple ActiveKnowledge Management Consoles, allowing for a flexible and scalable implementation.

All Oracle database versions from 7.3 or higher are fully supported, and the ActiveKnowledge server can be located on all major platforms including Windows, HP-UX, Solaris, and Linux. Porting to new hardware variants is a quick and seamless process; versions for additional operating systems are being evaluated and tested.

Topology

ActiveKnowledge is built on Database Network Router TM technology, transparently installed between the application and the database. It acts as a router - all communications between applications and databases are routed through it, as it works on the database network protocol layer. ActiveKnowledge is completely transparent - clients and application servers see ActiveKnowledge as their database server; database servers see ActiveKnowledge as any other application / client.

A Remote Management Console enables the easy configuration and management of the ActiveKnowledge server, and the definition of optimisation and security rules and policies. The Management Console is not used in the actual application of policies to the incoming SQL; merely in the up-front definition of those policies.

Functional Layers

ActiveKnowledge is built of three independent functional layers.

Application Connection Switching Layer

All clients or applications connect to the databases using a single ActiveKnowledge listener port (called ActiveKnowledge Port), which defaults to port number 1526. Using client connection information,

ActiveKnowledge identifies the database service name or SID and routes the connection accordingly to the database server. The Switching layer enables administrators to create different routing types and optimisation policies customised for each application program name. Possible routing types include 'apply optimisation rules', 'connect directly' - bypassing ActiveKnowledge and 'refuse all connections'.

Optimisation Rules Layer

This enables the ActiveKnowledge administrator to define and manage optimisation rules and management policies, which are automatically applied to statements on their way to the database. Possible rules include applying database 'Hints', statement rewrites and blocking unwanted statements.

Each optimisation rule consists of a two step process:

1. **Statement identification.** Each rule requires a means by which the incoming SQL can be compared and then matched against, to determine which rule or rules should be applied. SQL identification methods include statement text match, syntax match (includes SQL parsing), execution plan match (full plan match, partial or single execution step match), matching by means of the invocation of a PL/SQL function, partition range matching, Oracle cost matching, and time-of-day comparison.
2. **Action applied.** When a rule is considered to have been matched, then an action can be defined in the rule to be applied to that SQL statement. These actions can include adding database hints, rewriting, blocking (with a suitable returned message), and invoking a PL/SQL function, to carry out more complex operations, or just audit logging.

The optimisation rules may be the result of using the ActiveKnowledge SQL Expert, or through manual creation which can involve applying tuning suggestions from other 3rd party tool-sets.

Many rules can be defined for any given database / client combination. They are held in a folder structure, which is searched in a top-down manner, seeking out rules that match the incoming statement. In addition folders themselves have a matching criteria, (which is identical to that of the rules themselves) to decide whether the contents of the folder are examined or skipped. Once a rule or folder has been matched, and the action applied, then there is control over whether to stop scanning the rule tree or to continue to see whether further rules may apply. For example, if a security rule is applied, to see whether the client is authorised to view the data, then it may be appropriate to continue searching for a rule that would improve the performance of the query as well.

ActiveKnowledge SQL Expert Layer

The ActiveKnowledge SQL Expert tunes SQL statements by analysing all possible hints that generate a unique execution plan in the database. These possible hints are presented as a list of optimisation alternatives, sorted according to hint type and Oracle cost. ActiveKnowledge SQL Expert allows for the benchmarking of the optimisation alternatives to verify each alternative's execution time and resource usage. By simply selecting an alternative and clicking on the 'Fix SQL' button, an optimisation rule is created in the Optimisation Rules Layer, to be automatically applied to incoming statements – concluding the performance enhancement cycle.

Configuration

1. Clients or Applications are routed to use the ActiveKnowledge server host name and ActiveKnowledge port as their primary database address, with the database server as a fall-back using the standard Oracle client's failover capability (to eliminate single point of failure).
2. Each database to be monitored by ActiveKnowledge is defined via the Management Console.

3. Each client or application (defined by a number of characteristics, such as program name, host name etc) is also defined to ActiveKnowledge within the Management Console.
4. A routing relationship is defined, again in the Management Console, linking any database with any client or application. It is in this routing relationship, that one or more rules or policies are defined, which define the matching characteristics to be compared against incoming SQL statements, and the actions that are applied to those statements, when a match has occurred.
5. In real-time, client connections are then routed through the ActiveKnowledge server to the database servers.
6. Each statement is checked as it passes through the ActiveKnowledge server in real-time for comparison against one or more of the optimisation rules. When such a match occurs, the action defined in the rule is applied and the statement is updated or optimised and forwarded for execution to the database, or blocked. Similarly the result set from the query is passed back to the client from the database, via the ActiveKnowledge server.

ActiveKnowledge - A Highly Available Solution

ActiveKnowledge is fully compatible with high availability solutions, and need not be seen as a potential source of failure.

Even if the ActiveKnowledge server has been completely removed from the network through failure or otherwise, the standard Oracle client automatic fail-safe mechanism is available whereby all clients are immediately routed to the original database server, using the failover definitions in the client configuration.

In addition, for those sites requiring an even higher level of resilience, then the ActiveKnowledge server can be installed on a cluster service, with a hot failover capability.

ActiveKnowledge should not be seen as a bottleneck in any proposed solution; the propagation delay added by ActiveKnowledge is normally less than one millisecond.

Examples from Current Production Implementations

This section describes briefly some scenarios and examples where ActiveKnowledge has recently been deployed.

Performance Improvement in a Business Objects System

Orange TM (Israel) have recently finalised a purchase of ActiveKnowledge for a 14 terabyte data warehouse, running on a 52cpu Solaris server, supporting in excess of 1000 users, principally using Business Objects as the query tool. Here are the highlights:

- Top 30 reports examined and tuned using the ActiveKnowledge SQL Expert. From that tuning exercise, 7 partial explain plan policies were defined.
- These affect over 100 separate queries and reports, which are run many hundreds of times each day.
- The average report response time has seen a reduction of 35%, from 5.3 to 3.4 minutes.
- The most important reports had runtimes reduced by a factor of 5 times.
- A reduction in database server utilisation of around 10%, allowing the release of 4 cpus with corresponding software licenses
- This was achieved in less than three weeks.

Performance Improvement in a Chordiant CRM System

A UK client had been experiencing considerable performance problems whilst using a Chordiant CRM package. Query performance was poor, with some reports taking many hours, and the database server was completely 'maxed out'. The system is managed by Kognitio Ltd, who provides a fully managed service.

- Three ActiveKnowledge rules were produced by the System Integrators, Kognitio Ltd, which because of their detailed knowledge of the application could be based on partial database execution plans, and hence were very generic.
- As a consequence many tens of queries and reports were affected, with substantial improvements to the system.
- The most impressive improvement to an individual report is an 80% reduction in run-time.
- Over 50% of queries are affected by the defined policies, with their runtimes having reduced by over 80%
- Server CPU utilisation has dropped by approaching 70%
- A quote from Ben Cooper, the Operations Director of Kognitio Ltd:

"Kognitio have been using ActiveKnowledge to enhance performance of a leading marketing automation tool - in an extensive evaluation using production volumes we have seen performance enhancements in excess of 50%. As we continue to tune ActiveKnowledge further we anticipate that the run times for most queries will at least be halved, reducing cpu contention on the server whilst also avoiding expensive hardware upgrades for our clients."

Access Control Enhancements from a Banking System

In this situation, performance improvement was not an issue – the bank was quite happy with the performance that they were already achieving. However, the ability to intercept SQL queries on the way to the database, and manipulate them, gave them additional ways of improving security and access control, without recourse to the traditional Oracle security products.

- ActiveKnowledge is used to add additional 'where' clauses to the SQL queries, thereby restricting the amount of data that is visible. MS Active Directory is queried to obtain the user's department, and this is added in the where clause being used, or added as a where clause.
- ActiveKnowledge is used to restrict column lists that may be queried. Again MS Active Directory is queried to ascertain if the user is allowed to query certain columns. Depending on the outcome, the action of the ActiveKnowledge rule is to rewrite the query either permitting or removing the particular column.

In other cases, Credit Card numbers are scrambled before presentation to certain classes of users, again via the use of an ActiveKnowledge rule.

As can be seen ActiveKnowledge does provide significant real benefits in many different circumstances. Experience those benefits yourself, and call for more information.



ApplicationPerformance.com

