

# Solutions for Business Applications

Performance improvement, QoS stability and Data Leakage Prevention

## Challenges

- Improve end-user response time and ensure acceptable cycle execution time → with **ActiveBase Priority**
- Prevent performance degradation caused by the growth of data and business requirements, without buying more hardware and licenses → with **ActiveBase Priority**
- Prevent performance degradation and database locks due to uncontrolled analyst reports, developers and DBA tools accessing production → with **ActiveBase Performance**
- Prevent leakage of Sensitive and Personal Information from production databases → with **ActiveBase Security**

## Customers Experience

- ◆ A leading Telecom company uses ActiveBase Priority on an ERP application. Several times a day, the database server experiences resource peaks due to the requests of business analysts and application batches. During these peaks, call center users suffer unacceptable response time degradation.  
**ActiveBase Priority** was quickly installed, and rules implemented that automatically identified resource peak times.
  - Before outage was reached, rules allocated 25% less CPU and I/O resources to analysts and batch. This timely resource reduction ensured acceptable performance levels to the call center users.
- ◆ Another large Telecom company customer uses **ActiveBase Priority** on a billing application. An important billing cycle needed to be finished by 8:00 AM, yet due to contention on server resources (consumed by low priority reports and jobs) it is completed after 10:00 AM – unacceptable by the business.  
**ActiveBase Priority** was quickly installed, where rules guarantee that the billing cycle is completed in due time by ensuring that it receives adequate CPU and I/O resources.
- ◆ A Large Bank uses **ActiveBase Security** to control and audit usage of development and DBA tools across all production datacenter, including ERP, CRM, Billing, DW and more. In addition, access to personal



*"Using ActiveBase software made our business reporting ten times faster and more efficient, saving substantial resources and enabling us to expand while still maintaining our existing server."*

Ms. Limor Malay, DW  
Division Manager  
Orange



Operational Dashboard



information in production is automatically masked (e.g., SSN, names and addresses), allowing access to the database objects, yet preventing leakage of personal information from production systems.

- ◆ A Telecom company has users running Business Objects reports on their ERP production. Several times a month, some Business Objects reports are causing massive performance degradation and productivity loss. **ActiveBase Performance** was quickly installed and restored adequate response time using a few rules like:
  - *Block the requests that run full scan on large tables*
  - *Block the reports that do not include a date restriction*
  - *Limit both the number of concurrent reports executed, as well as the amount of parallelism assigned to these requests (limit to less than four parallel query servers).*

## ActiveBase Priority Solution

### Challenge:

The exponential growth of data, on-line users and business requirements cause resource peaks. During these peaks users experience productivity loss as response times deteriorate and SLA is missed.

### The Solution:

**ActiveBase Priority's** resource prioritization is based on user-defined rules that dynamically allocate database server CPU and I/O resources to important business transactions, batches and cycles while reducing the resources used by less important transactions.

As more resources are available to important business transactions, their response time is improved and acceptable SLA is maintained at all times.

A dashboard enables system operators during peak times to drill down to specific performance penalizing requests or OS processes, and to immediately act to resolve resource contention with a single mouse click.

### Rules examples:

- 1) *When CPU is in high load (like > 85%) reduce reporting CPU and I/O consumption by 50%*
- 2) *During day time, the billing cycle should not consume more than 20% of server resources.*

## Benefits

- ✓ Save on costly hardware upgrades and software licensing.
- ✓ Improve business transaction response time.
- ✓ Guarantee QoS and provide predictable service levels to important business transactions, batches and cycles.
- ✓ Automate the prioritization of business transactions through powerful Rules.
- ✓ Control user activities by blocking performance degrading SQL queries and unplanned jobs
- ✓ Powerful Rules prevent Server from reaching resource outage.



# ActiveBase Performance Solution

## Benefits

### Challenge:

Frequently, business applications suffer from performance degradation caused by penalizing SQL requests generated by closed or packaged applications (improper use of requests within application forms and reports), and by unrestricted reporting, development and DBA tools.

### The Solution:

**ActiveBase Performance's** unique SQL\*Net Proxy transparently intercepts long running SQL requests, and user-defined Rules apply blocking actions (including returning a message back to the user) or various SQL optimization techniques, including rewrites or adding Oracle Hints, with **no changes on application source code or databases!!**

### A) Blocking penalizing requests

**ActiveBase Performance** user-defined rules can:

→ Apply usage restrictions on analyst reports, developers and DBA teams, based on time of day, full scan execution, date ranges or other considerations

*For example. a rule can identify and block full scan operation on a specific table (like BILLING\_ACCOUNT table) (see example 4)*

→ Block different DML/DDDL/DCL SQL commands while returning an appropriate message to the user

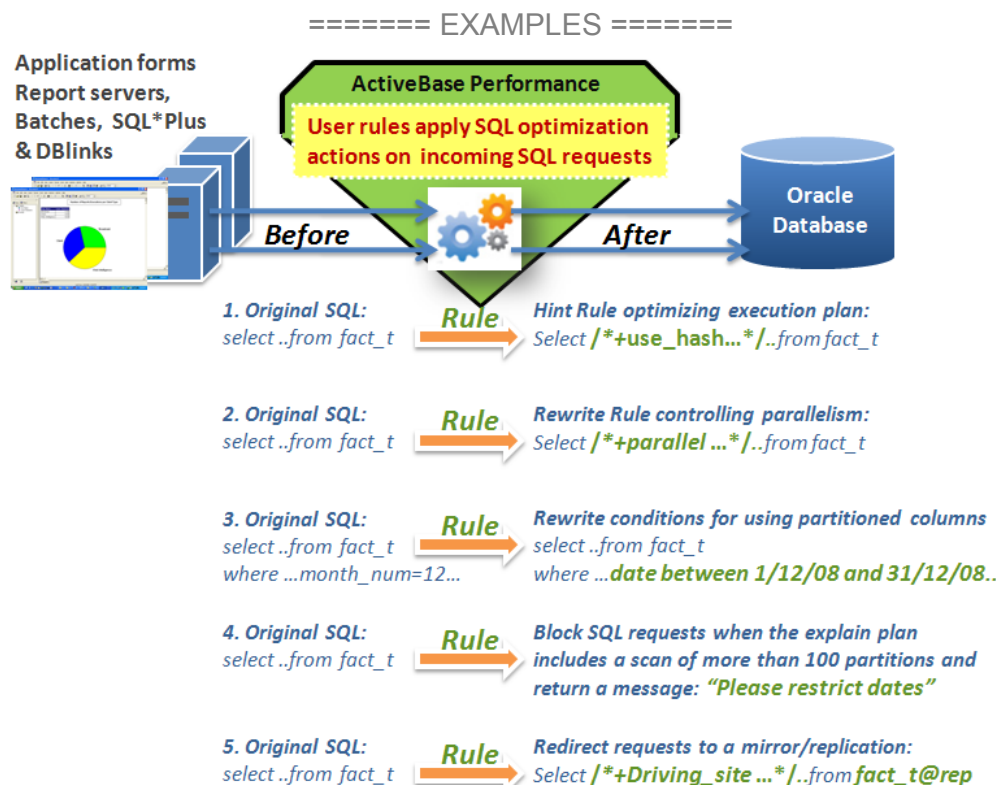
→ Change session settings (such as restrict parallel requests).

### B) Improving response time of long running requests

→ SQL optimization techniques result in up to x100 faster response

### C) Redirecting requests to a replication/mirror (see example 5)

- 👉 **Without changing applications or touching databases:**
- ✓ Prevent 'query-from-hell' from degrading production performance.
- ✓ Improve response time from hours to minutes by optimizing reports and ad-hoc queries.
- ✓ Automatically switch reports from production to a mirror/replication or archive database.



# ActiveBase Security Solution

## Challenge:

- ◆ Over time, new regulations and evolving business usage require to enforce higher level of security on customizations, on interfaces to other applications and on external users accessing the enterprise applications from outside the organization.
- ◆ The increasing compliance and regulations requirements for protecting sensitive and personal information on production CRM, ERP, Billing and Proprietary applications has become a corporate imperative.

In most cases, the time and cost of enhancing the security within the application is too high, and other means should be used.

- ◆ Moreover, existing security built in applications do not apply on developers, QA and DBA tools neither does it protect from reporting tools accessing production data - yet organizations are obliged to prevent data leakage from these tools as well.

Business applications have thousand of database objects where personal information can be exploited.

➔ Restricting access to personal information or restricting 'grants' on objects can be extremely complex and might cause productivity loss while frustrated end-users, developers, QA and DBAs are not able to do work.

## The Solution:

**ActiveBase Security** offers a new approach: customized Rules simply change the SQL statements 'on-the-fly' and apply **hiding**, **masking** or **scrambling** functions to the results returned to unauthorized users.

Rules can have a broad scope so a single rule will apply to all similar SQL requests that retrieve personal information.

For example, a rule that masks Social Security Number will automatically mask results retrieved by all 'Select' requests generated from external development tools BUT NOT from legitimate application requests.

Rules apply across databases and applications, with **NO CHANGES TO THE APPLICATIONS OR THE DATABASE!**

For example, instead of blocking access to the table per\_people\_f (a central table in an Oracle Application ERP suite containing personal information) which is used in many forms throughout the application, **ActiveBase Security** rules will selectively change only those SQL requests coming from reports, development and DBA tools that retrieve personal columns from per\_people\_f table (such as names) in order to mask the results.

# Benefits

- ✓ Prevent leakage of Sensitive and Personal Information from reporting or DBA Tools accessing the production environment.
- ✓ Allow selective access to personal information based on time-of-day, specific screens, users, modules or specific Security Officer grants
- ✓ Enforce application security policies across applications and tools accessing production database.
- ✓ Apply database security, without changing database configurations or data!
- ✓ Secure SQL access to databases even when code source is not available!
- ✓ Traceability: ActiveBase audits, controls and alerts on access to SPI or changing user grants and account information.
- ✓ Audit, control or block all users NOT even passing through AB\*Security

Nom

Qualité M.

Nom BR\*\*\*

Nom de jeune fille BR\*\*\*

Prénom légal PHILIPPE

Prénom usuel PHILIPPE

Multi-situations 0/0

Date d'effet

Multi-situations

*ActiveBase Security rules mask personal information within a packaged HR application screens that users are not allowed seeing*

### Advantages of using ActiveBase's IN-MOTION SECURITY

- ◆ Masking customer, financial, or company confidential data on-the-fly delivers legible data which retains the data's properties, such as its width, type, and format, without the need for customizations in existing applications.
- ◆ **ActiveBase Security** does not risk application or data integrity which might occur when masking primary keys required by the applications or databases, as ActiveBase masks personal information 'in-motion'.
- ◆ It also saves regeneration of new QA and test environments for applying mask functions on data – since **ActiveBase Security** solution is operational immediately upon installation.

Find/Enter Customers

Customer Type Organization

Basic Advanced Text

Customer

Name A%

Party Number

Account Name

Customer Number

Status Active

Address

Address1

Address2

City

Postal Code

Province

Site Number

Contact

Last Name

Search Type Exact (E) Fuzzy

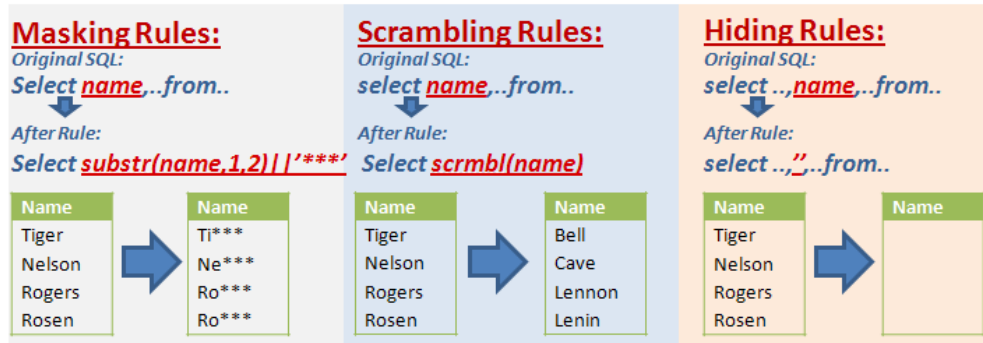
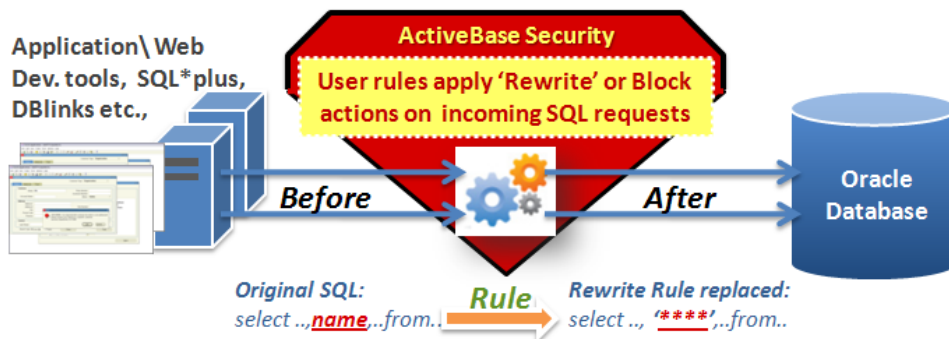
Clear Find

Error

ORA-00900: You requested full customer list which is not authorized!  
Please restrict your request to a specific customer.  
Security Department, int. 8188

OK Details...

*ActiveBase Security rule blocks users from retrieving sensitive business information within application screens while returning a custom message changing standard ORA-900 error text*



*ActiveBase Security Rules are transparently applied without changing applications or touching databases*