

## **Private or Public Synonyms for Database Objects – or no Synonyms at all ?**

*A decision support paper visualizing the pro's and contras in tabular form and evaluating the arguments.*



<http://www.IT-Checklists.com>

Copyright © 2005 Mercury Consulting  
Published in December 2005  
Last updated in July 2006

Oracle® and Oracle-based trademarks and logos are trademarks or registered trademarks of Oracle Corporation, Inc. in the United States and other countries.

## Conventions

The following typographical conventions are used in this book:

Constant width	Used for source code
<b>Constant width bold</b>	user input
[ ]	optional elements
{ }	mandatory choice
	separates choices

The author appreciates your feedback sent to [office@Mercury-Consulting-Ltd.com](mailto:office@Mercury-Consulting-Ltd.com)

# Table of Contents

Summary..... 4

Audience..... 5

Tabular Overview..... 6

Evaluation..... 7

How Mercury Consulting Ltd. can help you ..... 12

## Summary

The decision for private or public synonyms or against synonyms at all has impact on

- application (code) design
- user-administration
- how many database-instances will be required for development and testing (Budget- / Cost – Impact)

There is no “best” decision, the decision depends on the individual circumstances.

It is very likely that two of the 5 approaches are selected for one application,

- one for the technical user account used by application batch-processes and application middle ware,
- an other one for Users connecting via Client-Server.

There might be one decision for the Development- and Test-Environments and another decision for the Production / Live Environment.

When **selecting a standard software package** the customer should

- assess if the approach used by the standard software package is compatible with his/her standards and/or requirements (e.g. if customer uses a 3<sup>rd</sup> party reporting tool)
- vendor's support for / acceptance of customizations (e.g. Does the vendor allow the customer to create additional private or public synonyms or will vendor drop support?)

When designing and **developing individual software** the customer should from the very first beginning make and document the decision. In case of

- Outsourced bespoke development it is recommended to document this in the requirement definition phase in the section “non-functional requirements”.
- In house development this also needs to be documented in the requirements document if it's not already in the companies design- and development standards or if the requirements for this new project are different from the existing design- and development standards.

## **Audience**

- Operations Manager
- Testing Manager
- Environment Manager
- Application Support Staff
- System-DBA, Application-DBA, Development-DBA
- Quality Assurance Manager

# Tabular Overview

Note: You might decide to use two of those approaches within one application,

- one for the technical user account used by application batch-processes and application middle ware,
- an other one for Users connecting via Client-Server.

Characters in square brackets e.g. [D2] or [C] point to a more detailed evaluation in the next section.

Approach of accessing database objects →  Aspect ↓	Alternatives to Synonyms			Synonyms	
	(A) Login to schema of object owner	(B) Code uses schemaname.objectname	[C] alter session set current_schema = <username_of_objectowner>	(D) Private synonyms	(E) Public Synonyms
(1) Creating a new database user...	(+) no action required	(+) no action required	(+) no action required	(-) creating new synonyms is required [D1]	(+) no action required
(2) There exist in different schemas database objects (tables, views) with same name	(+) no action required	(+) no action required	(+) no action required	(+) The schema owner creating the synonym must decide to which object the private synonym will point. [D2]	(-) The public synonym will point to only one object [E2]
(3) Shared Pool	(+) only one version of SQL-statement is stored.	(+) only one version of SQL-statement is stored.	(+) only one version of SQL-statement is stored.	(-) The same SQL-Statement issued by different users will – although having same SQL_HASH_VALUE – result in different “Versions” to be stored in Shared Pool because each private Synonym is a different Object – detected during Soft-Parse. [D3]	(+) only one version of SQL-statement is stored.  (-) But there is some overhead because of “negative dependency checking”
(4) Performance – low impact, only in huge OLTP-systems	(+) As <user=owner> no access-privileges need to be evaluated during softparse	(+/-)	Impact is dependent on number of users and numbers of soft-parses. Note that soft-parses are different from hard parses! Soft-parses take place even when using bind-variables but can be reduced to a minimum by optimizing cursor handling (keeping cursors open).		
(5) Security	(--) logged in as schema_owner implies that ALL object privileges are present, even “drop table” - VERY dangerous when using (graphic) database tools in read/write mode – one click on wrong button and the table is gone....	(+) In all cases only the privileges granted to the user logged in are available. Note: “Public Synonym” does not allow access to the database object by public! Synonyms just “re-direct” the user e.g. SELECT * from CUSTOMER is redirected to SELECT * from APPL_OWNER.CUSTOMER and if the user logged in does not have sufficient access rights (in this case SELECT) via database roles or direct granted privileges he/she will get an error message.			

## Evaluation

The indicators in square brackets refer to the related cell in the tabular overview, e.g. [D1] refers to column D of row 1.

### ***[B] Application Code prefixes the objects with schema-name of object (table, view, procedure, package) – owner.***

This approach is good from a security point of view, as the application account will get only the required privileges, but most likely not e.g. "drop table" privileges.

This approach is also good from an performance- and shared pool perspective, as it avoids the name resolution during hard- and soft parsing.

Note: If the database account name owning the application objects is not configurable during installation but a fixed string, you can have only one environment per Oracle database. That means you cannot share one Oracle database for several test- and development environments.

### ***[C] alter session set current\_schema = <schema-name of object-owner>***

Requires "alter session" privilege

- 3<sup>rd</sup> Party Software (e.g. a reporting-tool) might not support / utilize this feature.

This command is very useful for application support team during troubleshooting on databases where no synonyms exist (because 3-tier application uses technical database account and manages user accounts on application level).

Using this command members of application support can still convenient work using their personal database account (without "drop table" privileges, best with only SELECT privileges) without the need to prefix the table-/ view-owner in each select and avoids the risks associated with logging in as the owner of the database objects.

## ***[D1]– After creating a new database user the creation of new private synonyms is required***

Creating a private synonym, e.g. "create synonym CUSTOMER for APPL\_OWNER.CUSTOMER" stores this information into the data dictionary and has following impact:

- Size of data dictionary increases – 10,000 users x 1,000 private synonyms (for 1,000 tables and views owned by the application) = 10 Mio rows in the data dictionary. The tables of the data dictionary are stored in the library cache, a part of the shared pool.
- It is a DDL-transaction – and therefore requires the ST-enqueue (Space Transaction Enqueue), and there is only a single ST-enqueue available per database instance.
- Creating 1,000 synonyms might take a few minutes.

The magnitude of the impact is different between a Development- or Test-Environment with e.g. 10 users and a Production / Live-Environment with > 1,000 users.

## ***[D2], [E2] In different database schemas there are database objects (tables, views) with same name***

Main reasons for this situation:

### **(a) Several test-environments, e.g. Unit-test, system-test, integration-test, user-acceptance test (UAT) are installed in separate schemas in the same database instance.**

(This is a popular approach to reduce the number of database instances when the IT cost center or service center (no matter in-house or outsourced) have charges per database instance.)

There exists standard software which allows the administrator to choose during installation process if he/she wants to use public synonyms or private synonyms. In case of selecting "private synonyms" those are created when creating a new database user; but at that point it must be decided which test-environment (= database schema) this user will use. The documented background for this feature is to allow running several Development- and Test-Environments in the same database and to use public synonyms for the Production / Live Environment with many users.

### **(b) Several applications are installed in separate schemas of same database instance, and each application has a table "CUSTOMERS", "ACCOUNTS", "ORDERS".**

Example:

```
CRM_OWNER.CUSTOMERS
```

```
BILLING_OWNER.CUSTOMERS
```

Private and public synonym can point to only one object in one schema.

It must be carefully evaluated if private synonyms are usable or cause confusion, e.g. if a user wants to create a report accessing tables in both schemas (e.g. Customer-complaints from CRM-system and Invoice-Problems from Billing-System). A private synonym pointing to BILLING\_OWNER.CUSTOMER might be worse than having no synonym at all and forcing user to prefix the table in both cases.

Note: Even if CRM- and Billing application get their own Oracle database, this issue might raise again in the Reporting-database containing data of both applications.

Workarounds:

```
create [public] synonym BILLING_CUSTOMER for BILLING_OWNER.CUSTOMER;
```

```
create [public] synonym CRM_CUSTOMER for CRM_OWNER.CUSTOMER;
```

## **[D3] , [E3] Shared Pool**

The negative influence of **private synonyms** takes only effect in case of high number of users. It is difficult to quantify the impact, a measurement during a full-scale load-test or stress-test is recommended.

**Public Synonyms** are significant better, but there is still some shared pool overhead for ensuring that no object with the same name exists in the users schema ("negative dependency checking").

## **[A] Security**

Logging in to the database as owner of the database objects is very dangerous, as it is not possible to revoke privileges from the object owner, e.g. it's not possible to "revoke drop table" or "insert", "update", "delete" from the table-owner.

Although new releases of database utility software allow configuration to "read only mode", this is unfortunately not always used, resulting in following risks of accidental mistakes:

- Hitting (by mistake) the "drop" button might result in loss of a database table
- Putting the cursor into the result-grid of a SELECT might already lock this record for update, causing locking-problems and blocking business transactions.
- Reviewing code of a stored procedure and hitting the "save" button

## **[C5-E5]**

Sometimes members of application support try to justify logging in using the master-account (database account of table-owner) with the argument that's not acceptable to prefix all table- and view-names during investigations... The options

[C] alter session set current\_schema = <schema-name of table\_owner>;

[D] private synonyms

[E] public synonyms

are alternatives and should be considered as the login using the table-owner's account is really dangerous.

## [A5]

A quite common approach is that the database account of the schema owning the database tables is created as type "OS Authenticated" / "identified externally" - usually known as "OPS\$"-account (*although the prefix OPS\$ isn't required, it's just the default value for the OS\_AUTHENTICATION\_PREFIX*). This allows that the application middle tier or batch jobs can start automatically without the need to store a password and avoids direct client/server login.

Unfortunately this requires that the application middle tier runs on the same server as the database, as the REMOTE\_OS\_AUTHENTICATION must remain to FALSE and never should be set to TRUE.

## How Mercury Consulting Ltd. can help you

Mercury Consulting Limited (MCL) is a professional consultancy providing experience, support and training in IS/IT operations for companies during high-time-pressure startup phase and following consolidation phase, especially in the Telecom-market.

### **Our Products**

you can purchase online and immediately download at our eBook-Shop



Our Checklists and Templates will help you to ensure that good or best practice is not only known but consistently applied!

### **Database Independent Products**

<b>Product</b>	<b>Benefit</b>
150 Non functional Requirements.	Requirements Template with 150 non functional requirements for selecting or developing robust, reliable, stable and manageable applications to meet the Service Level Agreements (SLA's). For external RFP's (Request for Proposal) and for internal development.
Checklist for Data Migration	65 important questions to identify and address or exclude typical migration pitfalls in an early phase of the project, thus ensuring the confidence for keeping the time plan.
Template: Systems- and Operations Handbook	Template to establish that documentation auditors like to see!
Interface Checklist	Those questions which you need to ask before starting the development! Requirements, Checklist and Template for Planning, Defining and Documenting Application Interfaces.
Checklist for Production Release and System Handover	Checklist for small and medium projects focusing on non-functional aspects for operations team. Simple but effective!
Business Requirements for Archiving and Purging	Template with Business Requirements for Archiving & Restore & Purging. Not removing old customer data can cause conflicts with privacy laws. <b>Business must act</b> and clearly specify what to purge and what to archive!
Application Health Check: Stability Assessment	Using this template to check your systems - and DOCUMENT the findings might show you even more potential issues beyond invalid objects.

## Database Specific Products

<b>Product</b>	<b>Benefit</b>
Template for Database Operations Level Agreement (OLA) / SLA	<p>If a Service Level Manager needs to offer to Business a Service Level Agreement (SLA) for an End-to-End IS/IT-Service, he sign this SLA only after he arranged within IS/IT for each system or component used to provide this service an <b>Operations Level Agreement (OLA)</b> with the providing team or department.</p> <p>This document provides a template for such an Operations Level Agreement (OLA) for Oracle Databases containing the agreed values and for QA-purpose also the measures implemented to reduce the likelihood of violations of those agreed values.</p> <p>It does not only deal with availability, but contains also comprehensive service catalogue of advanced DBA services and a template for the Service Level Reporting (SLR).</p>
Database Health Check - Part 1: Stability Assessment	<p>Stability Assessment of your Oracle Database.</p> <p>Most Application- and Database-Crashes can be avoided when detecting early indicators and reacting to them.</p> <p><b>Be Proactive - Check Now!</b></p>
DBA and Application Support: Job Description and Self Assessment	<p>Checklist to ensure that all 60 DBA-duties are assigned:</p> <p>System DBA, Development DBA and Application DBA versus Application Support</p> <p>If you have just a job role "DBA", but not a dedicated job role "Application DBA" those 19 duties must be explicitly assigned to either "Application Support" or to the "DBA" - otherwise they might not be executed!</p> <p><b>Detect unassigned tasks before an auditor reports them !</b></p> <p>This product addresses disputes between System DBA and Application Support (or, if existing, dedicated Application DBA) regarding the responsibility for the application's database objects.</p>

## Our free Whitepapers

<http://www.mercury-consulting-ltd.com/wp/whitepapers.html>

<b>Whitepaper</b>	<b>Benefit and Description</b>
The Importance of Application Level Monitoring.	Keeping your applications free from invalid objects is an important task, but does not guarantee error free operations. This free whitepaper explains the difference between "Application Level Monitoring" to "Database Monitoring" and "System/Server/OS"-Monitoring.
Important Facts about Redolog- and Archivelog in Oracle Databases for Change Managers and Application Support Staff	Understanding the topic "archivelog volume" can avoid unexpected troubles when applying changes or conducting upgrades. This awareness paper explains why Change Managers must ask questions about the archivelog volume created during changes and upgrades and why application support staff – and not only the DBA - must understand this topic.
The Danger of Invalid Database Objects	An awareness paper for Operations Managers and Application Support describing the problems and potential risks caused by invalid objects in an Oracle Database.
Private or Public Synonyms – or no Synonyms at all ?	A decision support paper visualizing the pro's and contras on a single page in tabular form and evaluating the arguments.
Daily, weekly or monthly Partitions ?	This document evaluates the different factors impacting the decision if daily, weekly or monthly partitions should be used.

## Our Services

<http://www.Mercury-Consulting-Ltd.com/services.htm>

## About the Author

The Author of this white paper is an



with 12 years experience as DBA and 6 years experience in Telecommunications Companies.