CMS
DATABASE ADMINISTRATION
ORACLE STANDARDS

5/16/2011

Contents

1. Overview

Standards are established rules, principles, or measures that are widely used, available, or supplied, and recognized and accepted as having permanent value. These Oracle standards identify steps necessary to implement an Oracle application database at the Centers for Medicare and Medicaid Services (CMS). The standards are intended to compliment the methodology and procedures described in the *Roles and Responsibilities* document and Oracle reference manuals.
These standards must be followed with care and consideration given to database object naming conventions, appropriate database object usage, and required object parameter settings. Any requests for deviation from these standards must be submitted in writing to, reviewed by, and approved by the Central Oracle DBA staff at CMS.

2. Oracle Database Development Life Cycle

2.1 Development Phase

1. The Division of Data Services (DDS) is formally notified of the new project initiative and a Central Data Administrator (DA) and Central Database Administrator (DBA) are assigned. If DDS is going to provide Local Data or Database Administration support, these resources are assigned as well.

2. Requirements are gathered, analyzed, and documented by project or application analysts.

    - To initiate a DDS project, contact the DDS Director. You will be asked to submit a Database Development Form.

    - You should submit an initial project plan to DDS for review and approval of Data Administration and Database Administration tasks and schedule.

3. The Local Data Administrator determines the project data requirements and develops a preliminary logical data model. All models must be developed according to DDMSS Data Administration standards.

4. The logical model is reviewed and approved by the Central DA. It is recommended that the Local and Central DBA be involved in the review as well. For more information, see the DDMSS Data Administration Standards.

5. A preliminary physical model can now be developed by the Local DA/DBA and submitted to the Central DBA for review and conditional approval. At this point the Central DBA will create a new instance or modify an existing instance for the Local DBA to use in developing the database based on the approved model.

6. Local DBAs will be given access to create database objects within their own schemas, and should keep the Central DBA informed of all activity taking place in the database. The Central DBA will provide all Oracle support related to the database software, space allocations, backup and recovery, and database troubleshooting.

    - It is expected that through the normal development process, the data model will change to address issues related to application design, requirement changes, etc. The local DA and DBA can make these changes at their discretion, but should involve the Central DA/DBA in any significant changes to the model by scheduling an Application Architecture Review.

    - The final version of the data model will be reviewed, and must be approved, by the Central DA/DBA staff before the database will be allowed to migrate to the test/validation server for testing and validation (which must be done prior to moving to production).

2.2 Test Validation Phase

1. When database development is complete, the database must be moved to the test/validation server for user testing, migration plan testing, and performance monitoring. However, before moving to the test/validation server, a Pre-Validation Migration Review shall be conducted.

2. Upon successful completion of the Pre-Validation Review, the Central DBAs will migrate the database to the test/validation server. User accounts for the application will be set up to support UAT or other testing. No changes to the database design can be made in this environment. All changes must be made in development and migrated to the test/validation server.

3. The local DBA, application developers, and system owners will manage the actual testing, and document the results and approvals.

2.3 Production Phase

1. Upon successful completion of testing on the test/validation server, the project team will scheduled a Pre-Production Migration Review with the Central DBAs in preparation for the move to production.

2. After approval by the Central DBA, the production move will be scheduled and performed by the DDS Central DBA staff. Local DBAs and developers will not be given access to the production environment.

3. Local DBAs should be available to the Central DBAs during the process to

answer questions and provide assistance, if necessary.

4. Post production support from the local DBA should be available for a four week period following the implementation of the new or modified database in the production environment.

2.4 Maintenance Phase

1. The Central DBAs from DDS are responsible for all maintenance, backup and recovery, monitoring and tuning once the database is in production.

2. The Central DBAs, at their discretion, may make changes to the database to improve performance or stability. This would cover changes that would not require modifications to the applications that use the database. Changes requiring application modifications would be made through the normal application development process. All changes by Central DBA will be documented and maintained in the Erwin data model for that database.

2.5 Retirement of Development and Test Environments

The development and test environments for each project will remain in place for four weeks after production implementation. After that, the databases will be de-allocated from the development and test servers until they are needed for development purposes again. The development environment will be re-established upon reception of a Database Development Form, and the steps outlined above for the Oracle database development life cycle will be followed. The test environment will be recreated as part of the SDLC.

3. Oracle Database Design Standards

3.1 Oracle Design Overview

The Oracle Design Standards define the steps necessary to evolve a logical data model into a physical schema and ultimately a set of physical database structures in Oracle. At CMS, tools have been identified to facilitate this process. CA/Platinum ERwin should be used to translate the approved logical data model to a physical model.

While developing the physical database design, all standards must be followed. CMS Central Oracle DBAs are responsible for creating the instance, database, and tablespaces according to user requirements and available resources.

3.2 Instances

An Oracle instance refers to the System Global Area (SGA) and the database background processes. An instance is started (memory allocated and background processes started) and then a database (datafiles) is mounted by the instance. To start an instance, Oracle must read a parameter file. Parameters must be modified based on database requirements.

3.2.1 Instance Naming Standards

See Standard Naming Conventions.

3.2.2 Object Usage

STANDARD:

- Create the parameter file in the following directory:

  'oracle/admin/dbname/pfile'
- Parameter file name must be: 'initdbname.ora'.
- Database should start up using spfile.
- symbolic link(s) must be created for pfile in /dbs.
- pfile should be in sync with spfile.
- Control files must be suffixed with '.ctl'.
- Instance name and DB NAME must be the same.

3.2.3 Required Parameters (DDL Syntax)

STANDARD: The parameters listed below must be modified in the init.ora parameter file defining an instance. Oracle default settings must not be assumed for any of these parameters.

Parameter	Instructions
DB_NAME	Provide a valid db name. Refer to the Standard Naming Conventions. This name is written to the control file for the database.
INSTANCE_NAME	Specifies the name of the instance. CMS standards require the instance name to be the same as the database name.
DB_DOMAIN	Provide a valid domain name. This should be a valid network domain name. Domain name should be WORLD.

Parameter	Instructions
CONTROL_FILES	Specifies the name of the control files. Two control files must be specified on different mount points. A fully qualified directory path must be specified for the control files. Refer to the Standard Naming Conventions.
DB_BLOCK_SIZE integer	Specifies database block size. Recommended minimum block size is 4096 for OLTP databases. A larger block size is recommended for OLAP, DSS, and mixed databases. This value cannot be changed once the database has been created.
SGA_MAX_SIZE	Specifies the maximum size of the SGA for the lifetime of the instance
SGA_TARGET	Specifies the total size of all SGA components.
SHARED_POOL_SIZE integer	Specifies the size of the shared pool. This is optional if SGA_TARGET is set.
PROCESSES integer	Specifies the number of operating system user processes that can simultaneously connect to an Oracle server.
LOG_BUFFER integer	Specifies the amount of memory that is used when buffering redo log files. In general, larger values for the log buffer reduce I/O, particularly if the transactions are long and numerous.
LOG_CHECKPOINT_ INTERVAL integer	Specifies the frequency of checkpoints in terms of the number of redo log file blocks that are written between consecutive checkpoints.
LOG_CHECKPOINT_ TIMEOUT integer	Specifies maximum time in seconds before another checkpoint occurs. Setting the value to 0 disables time-based checkpoints and is not recommended. Default value for Enterprise Edition is 1800.
BACKGROUND_DUMP_ DEST	Specifies the pathname for a directory where trace files for background processes are written.
USER_DUMP_DEST	Specifies the pathname for a directory where user trace files are written.
AUDIT_FILE_DEST	Specifies the operating system directory into which the audit trail is written for user SYS. It should be under sub-directory of /backup.
LOG_ARCHIVE_DEST_n	Specifies 1 through 5 destination parameters for archive logging. The pathname must be fully qualified for the archive log destination.

Parameter	Instructions
LOG_ARCHIVE_ FORMAT	Specifies the naming format of the archive log files. Recommended format is '%t_%s.dbf' where t = thread number, s = log sequence number.
CORE_DUMP_DEST	Specifies the pathname for a directory where core files are dumped.
SORT_AREA_SIZE integer	Specifies the maximum amount of memory to use for sorts. Increasing size improves the efficiency of large sorts.
COMPATIBLE	Allows you to use new release while guaranteeing backward compatibility.
JOB_QUEUE_ PROCESSES integer	Specifies the number of SNPn background processes per instance. It should be set to 0 in most cases due to ST&E.
UNDO_MANAGEMENT	Specifies which undo space management mode the system should use.

3.3 Databases

A Database in Oracle is an object which is created to logically group other Oracle objects such as datafiles, tablespaces, tables, and indexes within the Oracle Data Dictionary.

3.3.1 Database Naming Standards

See Standard Naming Conventions.

3.3.2 Object Usage

STANDARD:

- Datafiles must be defined in a fully qualified path.
- Database name must follow the Oracle Standard Naming Conventions.
- Redo log files must end with '.log' and datafiles with '.dbf'.

3.3.3 Required Parameters (DDL Syntax)

STANDARD: The parameters listed below must be included in the Oracle data definition language (DDL) when defining an application database. Oracle default

settings must not be assumed for any of these parameters.

Parameter	Instructions
DATABASE dbname	Provide a valid dbname. Refer to the Standard Naming Conventions. This name is written to the control file for the database.
CONTROLFILE REUSE	Normally used for re-creating an existing database. This will reuse an existing control file specified in the init.ora file. This option is not used when creating a new database.
DATAFILE dfname	Specifies one or more files to be used as system datafiles. Refer to the Standard Naming Conventions.
LOGFILE GROUP integer	Specifies one or more files to be used as redo logs. GROUP uniquely identifies a redo log file group. Values 1 to MAXLOGFILES. If this parameter is omitted then Oracle automatically generates its value. A database must have at least 3 redo log groups with 2 members each.
MAXLOGFILES integer	Specifies the maximum number of redo log file groups that can be created for the database.
MAXDATAFILES integer	Specifies the initial sizing of the datafiles section of the control file at CREATE DATABASE or CREATE CONTROLFILE time.
ARCHIVELOG	Specifies that the contents of a redo log group must be archived before the group can be reused. Required if point-in-time is a DB requirement. Default is NOARCHIVELOG for Development and Validation databases and ARCHIVELOG for PRODUCTION databases.
CHARACTER SET	Specifies the character set the database uses to store data. The default value is AL32UTF8.
NATIONAL CHARACTER SET	Specifies the national character set used to store data in columns specifically defined as NCHAR, NCLOB, or NVARCHAR2. The default value is AL16UTF16.

3.4 Tablespaces

A tablespace is a logical structure where tables, indexes, and clusters are stored. A tablespace is an area of storage made up of one or more physical datafiles. One or more tablespaces make up the database.

There are two types of space management within tablespaces: dictionary-managed and locally-managed tablespaces. Locally-managed tablespaces are created by default for application objects.

3.4.1 Tablespace Naming Standards

See Standard Naming Conventions.

3.4.2 Object Usage

STANDARD:

- Assign one or more tables/indexes per tablespace.
- Create tablespaces explicitly using the CREATE TABLESPACE command rather than implicitly by creating a table without specifying a tablespace name.
- Create a least one tablespace as temporary for user sorts.
- Specify suffix '.dbf' for application datafile names, '.log' for log datafile names.
- Set the default storage parameters to account for the typical object size.
- Specify PCTINCREASE 0. Exceptions can only be made with Central DBA approval.
- AUTOEXTEND must be on.
- MAXSIZE must be set.

3.4.3 Required Parameters (DDL Syntax)--Locally Managed

STANDARD: The parameters listed below must be included in the Oracle data definition language (DDL) when defining locally managed tablespaces. Oracle default settings must not be assumed for any of these parameters.

Parameter	Instructions
DATAFILE datafile path and name	Specify the mount point, directory, and name of the datafile.
SIZE size parameter	Specifies the size of the file. Use K or M to specify kilobytes or megabytes.
EXTENT MANAGEMENT LOCAL	Specifies that the tablespace is locally managed. AUTOALLOCATE: specifies that the tablespace is system managed, cannot specify the extent size. UNIFORM: specifies that the tablespace is managed with uniform extents. Specify size in K or M.

3.5 Tables

Oracle tables are the basic storage component in the Oracle database. Information is stored in tables in columns and rows. The data represented in these tables can be accessed using SQL data manipulation language (select, insert, update, and delete). Generally, tables are used in Oracle for applications and end-users to retrieve and manage business data.

3.5.1 Table Naming Conventions

See Standard Naming Conventions.

3.5.2 Object Usage

- Tables must be created in a tablespace that was explicitly created for the corresponding application. User tables must not be created in the system tablespace.
- All rows of a table should be uniquely identified by a column or set of columns to avoid duplicate rows. Every table defined in Oracle shall include an explicitly named primary key, and, if applicable, explicitly named foreign keys.
- All Oracle tables must have storage characteristics--see the section on Storage.

3.5.3 Required Components of the CREATE TABLE Statement

STANDARD: The components listed below must be included in the CREATE TABLE statement.

Parameter	Instructions
Schema	Must identify the application owning the given table. (If this parameter is omitted, the table is created in the user's own schema.)
Table Name	Must be a unique table identifier, following Standard Naming Conventions.
TABLESPACE	Specifies the tablespace name.
STORAGE	Specifies storage clause.

3.6 Columns

All columns must have a corresponding datatype to indicate the format of the data contained in that column. In addition to datatype, other edits can be associated with columns in Oracle to enforce defined business rules. These edits include default values, check constraints, unique constraints, and referential integrity (foreign keys).

3.6.1 Column Naming Standards

See Standard Naming Conventions in the Oracle standards and *Creating Physical Names for Elements and Columns* in the Data Administration standards.

3.6.2 Object Usage

- Columns that represent the same data but are stored on different tables must have the same name, datatype and length specification. An example of this is where tables are related referentially. If on a PROVIDER table, for example, the primary key is defined as PROV_NUM CHAR(08), then dependent tables must include PROV_NUM CHAR(08) as a foreign key column.

- Columns that contain data that is determined to have the same domain must be defined using identical datatype and length specifications. For example, columns LAST_CHG_USER_ID and CASE_ADMN_USER_ID serve different business purposes, however both should be defined as CHAR (08) in DB2. This standard applies to the entire enterprise and should not be enforced solely at an application level.

- All columns containing date information must be defined using the DATE data type. Specify a datatype that most represents the data the column will contain. For numeric data, use one of the supported numeric data types taking into account the minimum and maximum value limits as well as storage requirements for each.

- For character data that may exceed 20 characters in length, consider use of the VARCHAR(2) datatype. This could provide substantial savings in storage requirements. When weighing the benefits of defining a column to be variable in length, consider the average length of data that will be stored in this column. If the average length is less than 80% of the total column width, a variable length column may be appropriate.

- Consider sequence of the column definitions to improve database performance. Use the following as a guideline for sequencing columns on Oracle tables.

    1. Primary Key columns (for reference purposes only)
    2. Frequently read columns
    3. Infrequently read columns
    4. Infrequently updated columns
    5. Variable length columns
    6. Frequently updated columns

- For columns defined as DECIMAL be sure to use an odd number as the precision specification to ensure efficient storage utilization. Precision represents the entire length of the column, so in the definition DECIMAL (9, 2), the precision is 9.

- For columns with whole numbers SMALLINT and NUMBER shall be used.

3.6.3 Datatypes

BLOB is a valid data type that we do not recommend using because it is problematic. It does not allow for searching of the data, and data integration is poor. Blobs do not support CMS's Vision for the integration of the agencies structured and unstructured data to provide an overall enterprise view.

ECM is our standard because it aligns our systems to the CMS Vision to integrate and consolidate our systems. ECM allows us to search BLOB data, it indexes and identifies the unstructured data and it provides a central location for all unstructured data. This is very important because it will allow applications with unstructured the ability to share unstructured data with minimal effort.

IBM's Content Manager is referenced in the CMS Technical Reference Architecture Document posted on the CMS website as the CMS Standard for Unstructured Data and ECM should be used as the CMS standard unstructured data repository unless otherwise directed.

3.7 Indexes

An index is an ordered list of all the values that reside in a column or columns of a table. An index greatly increases the efficiency of a query running against the data within the column, returning the results of the query more quickly. Oracle can use indexes to improve performance when searching for rows with specified index column values and when accessing tables in index column order.

3.7.1 Object Usage

- When rows are initially inserted into a new table, it is generally faster to create the table, insert the rows, and then create the index. If the index is created before inserting the rows, Oracle must update the index for every row inserted.

- If an index is required, it must always be created in a tablespace reserved for indexes only.

- The index entry becomes the concatenation of all data values from each column. You can specify the columns in any order. The order you choose is important to how Oracle uses the index.

- Do not index columns that are frequently modified. UPDATE statements that

modify indexed columns and INSERT and DELETE statements that modify indexed tables take longer than if there were no index. Such SQL statements must modify data in indexes as well as data in tables. They also generate additional undo and redo information.

- Do not index columns that appear only in WHERE clauses with functions or operators. A WHERE clause that uses a function (other than MIN or MAX), or an operator with an indexed column, does not make available the access path that uses the index.

- When choosing whether to index a column, consider whether the performance gain for queries is worth the performance loss for INSERT, UPDATE, and DELETE statements and the use of the space required to store the index.

3.8 Referential Constraints (Foreign Keys)

A referential constraint is a rule that defines the relationship between two tables. It is implemented by creating a foreign key on a dependent table that relates to the primary key (or unique constraint) of a parent table.

3.8.1 Foreign Key Naming Standards

See Standard Naming Conventions.

3.8.2 Object Usage

STANDARD:

- Names for all foreign key constraints must be explicitly defined using data definition language (DDL). Do not allow Oracle to generate default names.

- DDL syntax allows for foreign key constraints to be defined alongside of the corresponding column definition, as a separate clause at the end of the table definition, or in an ALTER TABLE statement. Each of these methods is acceptable at CMS provided all of the required parameters noted below are included in the definition.

- When possible, attempt to limit the number of levels in a referential structure (all tables which have a relationship to one another) to three.

- Define indexes on foreign keys.

3.8.3 Required Parameters (DDL Syntax)

STANDARD: The parameters listed below must be included in the Oracle data

definition language (DDL) when defining a foreign key constraint in an application table. Oracle default settings must not be assumed for any of these parameters.

Parameter	Instructions
CONSTRAINT constraint name	Specify a name for the foreign key constraint based on the Standard Naming Conventions.
FOREIGN KEY (column name...)	Indicate the column(s) that make up the foreign key constraint. These columns must correspond to a primary key or unique constraint in the parent table.
REFERENCES table name	Specify the name of the table to which the foreign key constraint refers. If the foreign key constraint is based on a unique constraint of a parent table, also provide the column names that make up the corresponding unique constraint.
ON DELETE delete rule	Specify the appropriate delete rule that should be applied whenever an attempt is made to delete a corresponding parent row. If you omit this clause, Oracle does not allow you to delete referenced key values in the parent table that have dependent rows in the child table. Valid values include SET NULL and CASCADE. *(**Warning:** Use of the CASCADE delete rule can result in the mass deletion of numerous rows from dependent tables when one row is deleted from the corresponding parent. No response is returned to the deleting application indicating the mass delete occurred. Therefore, strong consideration should be given as to the appropriateness of implementing this rule in the physical design.)*

3.9 Temporary Tables

Within Oracle, you can create a table for use only in your session or whose data lasts for the duration of the transaction you are conducting. These types of tables are known as temporary global tables or temporary tables. Unlike a permanent table, a temporary table does not allocate storage in a permanent tablespace, but uses sort space to store the data. Should that space fill up, additional extents are created within the user's temporary tablespace.

3.9.1 Object Usage

When you create a temporary table, you must specify whether you want the data to persist for the transaction or the entire session. The clauses that control the duration of the rows are:

- ON COMMIT DELETE ROWS—specifies that rows are only visible within the transaction;
- ON COMMIT PRESERVE ROWS—specifies that rows are visible for the entire session.

Indexes built upon temporary tables have the same store as the data that resides within them. Triggers and views can be defined upon temporary tables; however, a view built upon a join between temporary and permanent tables is not allowed. Definitions of temporary tables may be exported and imported.

3.9.2 Syntax

```
create global temporary table temp_emp
    (empno number, name varchar2(20) , salary number)
on commit delete rows;
```

CREATE GLOBAL TEMPORARY TABLE AS SELECT is another method of creating a temporary table.

3.10 Views

A view in Oracle is an alternative representation of data from one or more tables or views. Views do not contain data. Actual data is stored with the underlying base table(s). Views are generally created to solve business requirements related to security or ease of data access. A view may be required to limit the columns or rows of a particular table a class of business users are permitted to see. Views are also created to simplify user access to data by resolving complicated SQL calls in the view definition.

3.10.1 Naming Standards for Views

See Standard Naming Conventions.

3.10.2 Object Usage

STANDARD:

- View definitions must reference base tables only. Do not create views which reference other views.
- Use views sparingly. Create views only when it is determined that direct access to the actual table does not adequately serve a particular business need.

3.10.3 Required Parameters (DDL Syntax)

STANDARD: The parameters listed below must be included in the Oracle data definition language (DDL) when defining a view. Oracle default settings must not be assumed for any of these parameters.

Parameter	Instructions
CREATE VIEW view name	Specifies the view name. OR REPLACE may be used in addition to CREATE VIEW. This will recreate the view without dropping, re-creating, and re-granting object privileges.
OF type_name	Explicitly creates an object view of type_name.
WITH OBJECT IDENTIFIER identifier name	Specifies the attributes of the object type that will be used as a key to identify each row in the object view.
AS select statement	Defines the contents of this view. Do not include existing views as part of the view definition.
WITH CHECK OPTION	Specifies that updates and inserts performed through the view must result in rows that the view can select.
CONSTRAINT constraint name	Used in conjunction with the 'WITH CHECK OPTION'. Assigns the name of the constraint.
WITH READ ONLY	Specifies that no deletes, inserts, or updates can be performed through the view.

3.11 Materialized Views

Materialized views are schema objects that can be used to summarize, pre-compute, replicate, and distribute data. They are suitable in various computing environments such as data warehousing, decision support, and distributed or mobile computing. The tables in the query are called master tables or detail tables. The databases containing the master tables are called the master databases.

3.11.1 Naming Standards for Materialized Views

See Standard Naming Conventions.

3.11.2 Object Usage

STANDARD:

- Either dimensions should be de-normalized (each dimension contained in

one table), or the joins between tables in a normalized or partially normalized dimension should guarantee that each child-side row joins with one and only one parent-side row

- For all dimensions, each child key value must uniquely identify its parent key value.
- Use the VALIDATE_DIMENSION procedure of the DBMS_OLAP package to verify hierarchical integrity in a de-normalized dimension.
- Fact tables and dimension tables should similarly guarantee that each fact table row joins with one and only one dimension table row.

3.11.3 Required Parameters (DDL Syntax)

STANDARD: The parameters listed below must be included in the Oracle data definition language (DDL) when defining a view. Oracle default settings must not be assumed for any of these parameters.

Parameter	Instructions
CREATE MATERIALIZED VIEW View name	Specifies the view name. SNAPSHOT may be used in place of MATERIALIZED VIEW. Limit the view name to 19 characters.
INITIAL size parameter	Specifies in bytes the size of the objects first extent. Use K or M. You cannot specify INITIAL in an ALTER statement.
NEXT size parameter	Specifies in bytes the size of the objects next extent. Use K or M. For rollback segments, use the same INITIAL and NEXT values.
MINEXTENTS	Specifies the total number of extents to allocate when the object is created. If the MINEXTENTS value is greater than 1, then Oracle calculates the size of the subsequent extents based on the values of the INITIAL, NEXT, and PCTINCREASE parameters.
MAXEXTENTS	Specifies the total number of extents, including the first that Oracle can allocate for the object.
PCTINCREASE	Specifies the percent by which the third and subsequent extents will grow over the preceding extent. Set PCTINCREASE to 0. This will prevent exponential growth during extent allocation.
TABLESPACE tablespace name	Specify the tablespace name for the MATERIALIZED VIEW.

Parameter	Instructions
USING INDEX	Specifies parameters for the indexes Oracle creates to maintain the materialized view.

3.12 Synonyms

A synonym is an alias for any table, view, snapshot, sequence, procedure, function, or package. Because a synonym is simply an alias, it requires no storage other than its definition in the data dictionary. Synonyms are used for security and convenience. They mask the name and owner of an object, provide location transparency for remote objects of a distributed database, and simplify SQL statements for database users.

3.12.1 Object Usage

Synonyms may be public or private synonyms. A public synonym is owned by the special user group named PUBLIC and every user in a database can access it. A private synonym is in the schema of a specific user who has control over its availability to others. Private synonyms are preferred.

3.12.2 Syntax

Create a synonym using the SQL command CREATE SYNONYM:

```
CREATE PUBLIC SYNONYM public_emp FOR schema.emp;
```

Drop a synonym that is no longer required using the SQL command DROP SYNONYM. To drop a private synonym, omit the PUBLIC keyword; to drop a public synonym, include the PUBLIC keyword:

```
DROP SYNONYM emp;
```

The following statement drops the public synonym named PUBLIC_EMP:

```
DROP PUBLIC SYNONYM public_emp;
```

When you drop a synonym, its definition is removed from the data dictionary. All objects that reference a dropped synonym remain; however, they become invalid (not usable).

4. Standard Naming Conventions

4.1 Object and Dataset Names

This section discusses the standard naming conventions for Oracle objects and datasets. These conventions were designed to meet the following objectives:

- Guarantee uniqueness of Oracle object names within an Oracle database;
- Provide a uniform naming structure for Oracle objects of similar types;
- Simplify physical database design decisions regarding naming strategies;
- Provide ability to visually group Oracle objects by the application and/or subject area which the objects were designed to support.

4.1.1 Usage

The following naming standards apply to any Oracle object (database, tablespace, table, view, PL/SQL routine, etc.) used to hold or maintain user data, as well as to external user objects (files, directories, script names, etc.) that will be used in conjunction with Oracle as part of standard application development and system operation procedures.

4.1.2 Standard Naming Format

All Oracle objects will be named according to the standard formats listed in the table below. All object names must begin with an alphabetic character.

Oracle Object Type	Required Attributes of Name	Example
Instance	• Up to 8 character name; • Begins with the application identifier; • Ends with d if a development database, t if a validation/test database or p if a production; • database; • Must be unique within server environment.	echimpd echimpt echimpp
Database	• Same as the instance name and follows the instance naming conventions; • Must be unique within the instance.	echimpd echimpt echimpp

Oracle Object Type	Required Attributes of Name	Example
Tablespace	• Up to 25 character name; • Begins with the application identifier and ends with; • **_data** if the tablespace holds data; • **_indx** if the tablespace holds index data; • Must be unique within database.	adm_data, adm_indx
Table	• 30 character name (maximum); • Must be descriptive as it applies to the application; • Cannot contain reserved words or special characters.	PSR_CR_AUDIT_ISS_TYP_DTL
Index	• 30 character name (maximum); • Must be unique within database; • Must be the table name, suffixed by the type of index.	
View	• 30 character name (maximum).	
Synonym	• 30 character (maximum).	Hsc01t
Column	• 30 character (maximum); • Unique within the corresponding table or view; • Should be derived from the • business name identified during the business/data analysis process; • Each word within the column name must be separated by an underscore (_); • Column name may not contain reserved words or special characters; • All abbreviated words must be obtained from or recorded in the list of CMS standard abbreviations. For more information, see *Creating Physical Names* in the Data Administration Standards.	Provider_id
Column Constraints	• Column check constraints must be constructed by an application identifier, followed by the column name, suffixed by _ck.	adm_provider_id_ck

Oracle Object Type	Required Attributes of Name	Example
Primary Key	• 30 character name (maximum); • Primary key name must be the table name suffixed by **_pk**; • Primary key name must be unique within the corresponding table definition.	Provider_pk
Foreign Key	• 30 character name (maximum); • Foreign key name must be the table name suffixed by **_fk**; • Foreign key name must be unique within the corresponding table definition.	Provider_fk01
Procedure, Function, Package	• 30 character name (maximum); • Prefixed by application identifier, followed by a description; • Suffixed by **_proc** (procedure), **_func** (function), or **_pkg**(package).	Hcis_get_cust_id_proc Hcis_get_cust_id_func Hcis_get_cust_id_pkg
Schema	• usually the application identifier.	hcis

4.2 File Names

This section specifies the naming convention for standard library names to be used in Oracle application systems.

4.2.1 File Naming Convention

The file naming convention for database files is the tablespace name, followed by a two digit sequential number, depending upon the number of datafiles assigned to that tablespace, followed by '.dbf', which stands for "database file." The database file should be contained within a parent directory that is the application ID.

Individual scripts, PL/SQL routines, etc., must be uniquely named for the server to avoid multiple copies of the same script residing on the same server.

4.2.2 Sample File Name

A database file which is within the HCIS system and is being used as the second file for the REF tablespace would be named as follows:
/db501/hcis/ref02.dbf

4.3 Utility File Names and Script Names

Oracle tables are often loaded with files utilizing SQL LOADER. These files should uniquely identify the particular table that is being loaded with the given file. The target table should be identifiable by examining the file name.

Utility script names should readily identify the purpose and nature of the script. Utility scripts should also be fully documented by utilizing comment blocks within the script itself.

5. Oracle Standards: Packages

5.1 Overview

Packages encapsulate related procedures, functions, associated cursors and variables together as a unit in the database.

5.2 Naming Standards

See Standard Naming Conventions.

5.3 Object Usage

STANDARD: Packages are used to define related procedures, variables, and cursors and are often implemented to provide advantages in the areas discussed below.

- Encapsulation of related procedures and variables: Packages allow you to encapsulate or group stored procedures, variables, datatypes, and so forth in a single named, stored unit in the database. This strategy provides better organization during the development process. Encapsulation of procedural constructs in a package also makes privilege management easier. Granting the privilege to use a package makes all constructs of the package accessible to the grantee.

- Declaration of public and private procedures, variables, constants, and cursors: The methods of package definition allow you to specify which variables, cursors, and procedures are public and which are private.

    1. Public items are directly accessible to the user of a package.

    2. Private items are hidden from the user of a package.

5.4 Required Parameters (DDL Syntax) - Package Specification

STANDARD: The parameters listed below must be included in the Oracle data definition language (DDL) when defining a Package. Oracle default settings must not be assumed for any of these parameters.

Parameter	Instructions
CREATE/OR REPLACE	Specifies the creation of a new package. 'OR REPLACE' may be used to recreate an existing package.
PACKAGE schema.package_name	Identifies the schema owner and package name.
AS/IS	Identifies the beginning of the package specification.
PL/SQL_package_spec	Specify the package specification, which can contain type definitions, cursor declarations, variable declarations, constant declarations, exception declarations, PL/SQL subprogram specifications, and call specifications (declarations of a C or Java routine expressed in PL/SQL).
END package_name	Closes the package specification.

6. Oracle Standards: Stored Procedures/Functions

6.1 Overview

A procedure or function is a schema object that consists of a set of SQL statements and other PL/SQL constructs, grouped together, stored in the database, and executed as a unit to solve a specific problem or perform a set of related tasks. Procedures and functions permit the caller to provide parameters that can be input only, output only, or input and output values.

6.2 Naming Standards

See Standard Naming Conventions.

6.3 Object Usage

STANDARD:

- Define procedures to complete a single, focused task. Do not define long procedures with several distinct subtasks, because subtasks common to many procedures might be duplicated unnecessarily in the code of several procedures.

- Do not define procedures that duplicate the functionality already provided by other features of Oracle. Do not define procedures to enforce simple data integrity rules that you could easily enforce using declarative integrity constraints.

- Use Packages to encapsulate Stored Procedures.

- Stored Procedures must be commented and must include the following for the original version and each revision:

    1. Author's initials
    2. Date of the change
    3. Reason for the change (for the original procedure, the reason is 'Original')

- All Stored Procedures must contain an 'EXCEPTION' handling section.

- Logging of execution steps for ease of tracking.

- Reserved words must be capitalized, all other words lowercase.

- Each Stored Procedure must have a descriptive name.

6.4 Required Parameters (DDL Syntax) - Stored Procedure Specification

STANDARD: The parameters listed below must be included in the Oracle data definition language (DDL) when defining a Stored Procedure Specification. Oracle default settings must not be assumed for any of these parameters.

Parameter	Instructions
CREATE/REPLACE	Optional keywords used to create a stand-alone procedure.
PROCEDURE procedure-name	Identifies the name of the procedure.
Parameter declaration	List of parameters that are defined to pass information into and send information out of the procedure, back into the calling program.

Parameter	Instructions
IN, OUT, IN OUT	Part of the parameter declaration that defines the behavior of the parameter. 'IN' allows you to pass values into the subprogram; 'OUT' allows you to return values from the subprogram; 'IN OUT' allows you to pass initial values into the subprogram and return updated values. 'IN' is the default if not specified.
datatype	Part of the parameter declaration that defines the type of the parameter (NUMBER, VARCHAR2 etc.).
RETURN datatype	Used for functions, it is the datatype returned by the function—e.g., NUMBER, BOOLEAN.

6.5 Required Parameters (DDL Syntax) - Stored Procedure Body

STANDARD: The parameters listed below must be included in the Oracle data definition language (DDL) when defining a Stored Procedure Body. Oracle default settings must not be assumed for any of these parameters.

Parameter	Instructions
IS/AS	Keywords 'IS' or 'AS' identifying the beginning of the procedure body.
Declaration statements	Declarations of local identifiers for the procedure.
BEGIN	Keyword identifying the beginning of the procedure's executable statements.
EXCEPTION	Keyword identifying the beginning of exception handling.
END procedure name	Keyword identifying the end of the procedure body.

7. Oracle Security Standards

7.1 Overview

CMS has some basic Oracle security requirements for all new applications coming into the CMS environment. Any exceptions to these base policies must be approved by the System Security Group (OIS/SSG) and the Division of Data Management and Support Services (OIS/EDG/DDMSS). Other questions related to Oracle security should be directed to DDMSS.

7.2 Oracle Security Requirements

Requirement	Comments
All Users for the application must be authenticated to the Database.	The application interface is required to capture the user logon and password to authenticate that user to the database when accessing the Oracle database on the users' behalf. The user must have a valid Oracle user ID and password on the database and any database activity must be performed under the User's ID for audit purposes.
Application User Interface (UI) must support Oracle database ID expiration and password change dialog.	Users of the application must be able to change their Oracle database password through the application's UI.
All Database users must be assigned to specific database roles.	Users cannot be assigned individual object privileges.
No application users will be assigned database administrator roles.	Reserved for DBA staff.
Logon level auditing must be implemented on all databases.	See DDMSS for assistance.
Oracle logon ID's will be assigned by the CMS RACF administrator.	
Database Links are not recommended	Must be cleared with DDMSS.

7.3 Database Link Requirements

The EDG central DBA staff does not recommend the use of database links. Any type of a direct connection between two or more database adds complexity in relation to database monitoring, coordinating the up time of the linked databases, coordinating maintenance management for the central DBAs. This includes additional elements that require of the linked database, coordinating the backup of the linked databases, and additional security elements. Since a linked database adds to the workload of the centrals DBAs, operational and maintenance cost will be increased.

There are several alternative to database links, such as Gentran, ETL and FTP to name a few. You may also contact the DBA staff at EDG for some suggestion of

alternative to a database link.

If it is decided to pursue a database link then following criteria must be met:

- The databases that are being linked must be on the same server.
- No more than one link can be established per database.
- Only private links can be created.
- All passwords pertaining to the link must be encrypted.
- The linking databases must be Oracle databases.
- The Central DBA staff must validate the design and technical implementation of the link.
- Written approval from the business owner of all databases that are linked must be received by EDG.