## Introduction

### About this Document

This document is intended for professionals working with relational databases at Stanford University. It describes the naming conventions - standards and guidelines - to be used for naming database objects. For each object to be named, it is indicated if the naming convention is a guideline. If there is no specific mention it is a standard by default.

It covers naming conventions for Sybase as well as Oracle database objects. Wherever possible, the guidelines are kept consistent for both DBMSs.

In the case of an **application software package**, the convention should be adhered to as much as possible. However, it is not warranted to customize the application software package solely to meet these naming conventions.

### Change Log

**Version Date Change description**

| 1.0 | 12/29/95 | First draft version |
| --- | --- | --- |
| 2.0 | 3/7/96 | Reviewed with all DBAs; Total revision; no side bars added. |
| 2.1 | 4/12/96 | Minor changes; added convention for Oracle archive files |
| 2.2 | 4/25/96 | removed identifying suffixes (constants) |
| 2.3 | 8/21/96 | Changes to name of Oracle archive files| |

**Note:**  
Changes introduced with the last revision are marked with sidebars.

### General Guidelines for naming

Database object names should follow the general naming conventions listed below. Wherever possible, naming conventions are kept consistent between Oracle and Sybase.

* + Oracle database object names are limited to a maximum of 30 characters with the exception of database names which are 8 characters.
  + Sybase database object names are limited to a maximum of 30 characters.
  + Object names follow all rules for the DBMS being used.
  + Object names contain only the letters [a-z] [0-9] and underscore �\_�.
  + Object names must start with a letter.
  + Object names will follow the rules for standard abbreviations as documented in the "Naming standards for Data Models" document (available on the Data Administration Web page).
  + Object names are all lowercase except where explicitly indicated.
  + Names cannot be Oracle or Sybase reserved words.

### Conventions

In this document, naming conventions are presented as templates composed of one or more of the following:

* + underscores: the elements of a name are separated by underscores to improve readability.
  + substitutions: represented by italicized characters, indicating that the element is composed of characters provided by the user.
  + constants: represented by non-italicized characters.
  + brackets: indicate that the enclosed character is included only under specific conditions as defined.

For example, the naming for Sybase triggers is presented as:  
***tablename*\_[d][i][u]*\_* trg**

* + The italicized characters "***tablename***" indicate that appropriate text will be substituted.
  + The underscores demarcate the name�s three elements.
  + The brackets around the characters "**d**", "**i**" and "**u** " indicate that these characters will be included or omitted based upon some criteria.
  + The suffix "**trg**" is a constant; all trigger names will end with this string.

# Sybase Naming Conventions

### SQL Servers

SQL Servers should be named: ***name\_vvv*** where

| ***name*** | is a meaningful name describing the application type(s) running on the server, | |
| --- | --- | --- |
| ***vvv*** | represents the version of the server. | |
|  | **dev** for development | **ppd** for pre-production |
| **prd** for production | **cnb** for crash and burn |
| **Examples** | txn\_dev | adhoc\_prd |

**Note:**In the past, Stanford adhered to a single server per node convention. With more new applications in the database environment, there may be multiple servers per node and it is not a good practice to include the node (host) name in the server name.

### Databases

A database is a collection of tables holding related information.Databases should be named: ***aaa[\_v]\_ttt*** where

| ***aaa*** | is a meaningful three character abbreviation for the applicationusing this database.  It should be the three character name assigned for ECS/Control-M | | |
| --- | --- | --- | --- |
| ***v*** | is a one character, alphanumeric version indicator (optional) | | |
| ***ttt*** | is the three character indication of the environment, and use of this database | | |
|  | **dev** for development | **ppd** for pre-production | |
| **prd** for production | **cnb** for crash and burn | |
| **Examples:** | ood\_5\_dev | ams\_1\_rpt | ehs\_trn |

**Keep in mind:** Database names are lowercase.**Note:** This naming convention is consistent with naming convention used for Oracle database.

### Tables

Since it is difficult to predict the many different types of databases and determine a suitable naming standard for tables at Stanford University, the following should be used as a guideline for naming tables. Additional conventions can be adopted by each application team as long as they do not conflict with any of the defined naming standards.Table names (also know as internal names) are the same as the entity names in the logical data model upon which they are based, with the following exceptions:

* + Use underscores for spaces
  + Use appropriate standard terms

For more detail on standard terms and internal entity naming conventions, please refer to the "Naming Standards for Data Models" document on the Data Administration Web page.Here are some additional considerations:

* + A work table is a table where content is discarded upon successful completion of a program, script or job and left intact after failure. It should be suffixed with **\_w** and should contain the name of the associated table.
  + A report table is the same as a work table except the content is left intact regardless of success or failure of the program, script or job. It should be suffixed with **\_r** and should also contain the name of the associated table.
  + A history table is a a table is used to archive records from active tables. It should be suffixed with **\_h** and should also contain the name of the associated table.
  + If a table is in tempdb but has to persist beyond the session, Work Tables, Report Tables and History Tables should contain the process id if multiple instances are present at any given time.
  + Do not use reserved words as the table name (Example for a reserved word is **user**).
  + For Sybase, work tables should be created in **tempdb** where they will be deleted when the server is shutdown.

### Views

Since it is difficult to anticipate the many different types of databases and determine a suitable naming standard for views at Stanford University, the following should be used as a guideline for naming views. Additional conventions can be adopted by each application team as long as they do not conflict with any of the defined naming standards.View names (also know as external names) are the same as the entity names in the logical data model upon which they are based, with the following exceptions:

* + Use underscores for spaces
  + Use appropriate standard terms

For more detail on standard terms and internal entity naming conventions, please refer to the "Naming Standards for Data Models" document on the Data Administration Web page.Here are some additional considerations:

* + All terms (separate by underscores) start with an uppercase letter
  + Add suffix **\_v** to designate this as a view.

### Columns

It is difficult to anticipate the many different types of databases and, therefore, define a suitable naming standard for columns at Stanford University. Therefore, the following should be used as a guideline for naming columns. Additional conventions can be adopted by each application team as long as they do not conflict with any of the defined naming standards.Column names are the same as the attribute names in the logical data model upon which they are based, with the following exceptions:

* + Use underscores for spaces
  + Use appropriate standard terms

For more detail on standard terms and attribute naming conventions, please refer to the "Naming Standards for Data Models" document on the Data Administration Web page.In addition, there are a few additional considerations as noted below:

* + Foreign key column names maintain the same names as in the parent table, that means same as the primary key column names.
  + Multiple instances of a column name in the same table get an appropriate prefix added to the column name. The prefix is based on the relationship given in the logical data model. For example, there are two 'part identifier' columns in the 'part' table. Their column names are 'part\_id\_*parent'* and 'part\_id\_*child'*.
  + When abbreviating a word in a column name, always abbreviate it the same for all columns the database.
  + Do not use reserved words as the column name (Example for a reserved word is **user**).

### Indexes

An index is a physical construct that contains a list of values for a column or columns and points to the locations of rows containing the values.Indexes should be named: ***tablename\_*[u][c] *\_* idx*\_n*** where

| ***tablename*** | is the name of the table the index is defined for |
| --- | --- |
| **u** | is a constant indicating a *unique index* (otherwise omitted) |
| **c** | is a constant indicating a *clustered index* (otherwise omitted) |
| **idx** | is a constant indicating that the object is an index |
| ***n*** | is a sequential identifier where **1** indicates the index on the primary key, and **2 and higher** are used for all other indexes |

**Examples:** fin\_roll\_summary\_uc\_idx\_1 (a unique, clustered index on the primary key of the *fin\_roll\_summary* table)

### User-defined data types

A user-defined data type is a custom defined data type which is different or enhanced from the system supplied data types. The power of user-defined data types is to enforce domain restrictions by binding rules and defaults to the user-defined data types.User-defined data types should be named as: ***ss\_name*\_typ** where

| ***ss*** | is a code representing the Sybase system data type (see below) upon which the user-defined data type is built |
| --- | --- |
| ***name*** | is a descriptive and meaningful name |
| **typ** | is a constant indicating the object is a user-defined data type |

**Examples:** ch\_carrier\_id\_typ, in\_branch\_number\_typ, sm\_rebate\_typ

| **CODE** | **Sybase system type** |
| --- | --- |
| in | *int* |
| si | *smallint* |
| ti | *tinyint* |
| nm | *numeric* |
| dc | *decimal* |
| fl | *float* |
| re | *real* |
| dp | *double precision* |
| mo | *money* |
| sm | *small money* |
| ch | *char* |
| vc | *varchar* |
| bi | *binary* |
| vb | *varbinary* |
| bt | *bit* |
| dt | *datetime* |
| sd | *smalldatetime* |
| tx | *text* |
| im | *image* |

### Physical Data Types

In some DBMSs, you can specify a particular data type more than one way. Whenever possible use the ANSI Standard definitions for the data type. However, this standard does not restrict the use of non ANSI standard data types.The following table shows the mapping of ANSI standard data types to the Sybase implementation.

| **ANSI Standard** | **Sybase Type** | **Physical Format** |
| --- | --- | --- |
| CHAR(n) | CHAR(n) | max 255 characters |
| CHAR VARYING(n) | VARCHAR(n ) | max 255 characters |
| CHAR VARYING(n) | LONG VARCHAR | max 2gb characters |
| NUMERIC(p,s) | NUMERIC(p,s) | precision = 1 thru 38  scale = 0 thru 38 |
| NUMERIC(p,s) | DECIMAL (p,s) | precision = 1 thru 38  scale = 0 thru 38 |
| FLOAT(p) | FLOAT(p) | 4 or 8 bytes |
| REAL | REAL | max 4 bytes |
| DOUBLE PRECISION | DOUBLE PRECISION | max 8 bytes |
| n/a | MONEY | max 8 bytes |
| n/a | SMALLMONEY | max 4 bytes |
| n/a | BINARY(n) | max 255 bytes |
| n/a | VARBINARY(n) | max 255 bytes |
| n/a | BIT | 8 per byte |
| INTEGER | INTEGER | 4 byte integer |
| SMALLINT | SMALLINT | 2 byte integer |
| SMALLINT | TINYINT | 1 byte integer |
| n/a | DATETIME | 8 bytes |
| n/a | SMALLDATETIME | 4 bytes |
| n/a | TEXT(n) | max 2gb characters |
| n/a | IMAGE | max 2gb binary datA |

### Rules

A rule specifies the domain of acceptable values for a particular column.Rules should be named as: ***name*\_rul** where

| ***name*** | is a descriptive and meaningful name |
| --- | --- |
| **rul** | is a constant indicating that the object is a rule |

**Examples:** branch\_number\_rul, employee\_number\_rul, carrier\_id\_rul

### Defaults

A default specifies a value to insert in a column (or in all columns of a user-defined datatype) if no value is explicitly supplied at insert time.Defaults should be named as: ***name*\_def** where

| ***name*** | is a descriptive and meaningful name |
| --- | --- |
| **def** | is a constant indicating that the object is a default |

**Examples:** country\_def, commission\_percentage\_d

### Constraints

A constraint is a restriction to enforce integrity when data is manipulated in a table.Constraints should be named as: ***tablename\_*[c]*\_ type*** where

| ***tablename*** | is a the tablename of the table to which this constraint belongs | |
| --- | --- | --- |
| **c** | represents a constant included if the constraint is implemented as a *clustered index* (omitted if nonclustered; not applicable for check constraints) | |
| ***type*** | indicates the type of constraint | |
| Examples include: | **chk** for check constraints |
| **unq**for unique constraints |
| **pky** for primary key constraints |

**Examples:** carrier\_pky, employee\_c\_unq, meal\_chk

### Triggers

A trigger is a special form of a stored procedure that goes into effect when a table is manipulated.Triggers should be named as: ***tablename*\_[d][i][u]*\_* trg** where

| ***tablename*** | is a name of the affected table |
| --- | --- |
| **d** | indicates trigger is defined for *delete* (otherwise omitted) |
| **i** | indicates trigger is defined for *insert* (otherwise omitted) |
| **u** | indicates trigger is defined for *update* (otherwise omitted) |
| **trg** | is a constant indicating that the object is a trigger |

**Examples:** air\_segment\_i\_trg, group\_client\_d\_trg, employee\_i\_trg

### Stored Procedures

A stored procedure is a collection of SQL statements that are stored in a database and can be executed by name.Stored procedures should be named as: ***name\_action*\_proc** where

| ***name*** | gives the name(s) of the objects upon which the procedure is carried out |
| --- | --- |
| ***action*** | indicates the main action or business rule implemented by the stored procedure |
| **proc** | is a constant indicating that the object is a stored procedure |

**Examples:** carrier\_get\_proc, group\_client\_add\_proc, employee\_transfer\_proc**Note:** For Sybase, system-wide stored procedures developed by Stanford University must be prefixed **sp\_su\_**. The suffix \_proc is not present for system-wide stored procedures.

### SQL Server logins and Sybase Users

In general, SQL Server login and Sybase User names should map to the user's respective Unix account name. The naming convention for Unix logins is ***f[m]llll*** where ***f*** is the first character of the user's firstname, ***m*** is the first character of the user's middlename and is optional, and ***llll*** is the user's lastname.When adding a login with sp\_addlogin, always specify the parameter *fullname* as"lastname, firstname". The default database should be a different database than the master database.When adding a user with sp\_adduser, always check if the user already has a SQL Server login (by checking for lastname, firstname), then add the user with the same user name. This process will ensure that the user has only one username across multiple SQL servers.Database Owners will be established with a user name of ***name\_*dbo** where ***name*** is the name of the database or application and **dbo** is a constant.

### Database Devices (Logical)

A database device is any piece of disk or file in the file system that you use to store databases and database objects. The term "device" can refer to raw partition or a file in the file system.Database devices should be named as: ***devicetype\_n*** where

| ***devicetype*** | represents the database device type | |
| --- | --- | --- |
| Examples include: | **log** for transaction log device  **data** for data device |
| ***n*** | is a sequential identifier beginning with 1 | |

**Examples:** log\_ 1, data\_5**Keep in mind:**

* + All Sybase devices at Stanford University should be raw partitions except tempdb. Tempdb should always be setup as a file system for performance reasons.
  + Physical device names are different than database device names and follow a different naming convention which is not documented here.

### Segments

A segment is a named subset of database devices available to a particular database.It is difficult to anticipate the many different types of databases and, therefore, define a suitable naming standard for segments at Stanford University. The following should be used as a guideline for naming segments.For small databases (approximately 200 mb or less), the indices and tables are most likely stored on the default segment.For larger databases, data and indices will be separated based on the analysis done by the DBA and the application team. Segments which hold multiple indexes should be called **index*\_n***. Segments which hold multiple tables should be called **data*\_n***.If there is a need to isolate a table or index into its own segment, it should be called ***objectname\_n***.**Note: *n*** is a sequential identifier beginning with 1.**Examples:** index\_1, data\_23, person\_1

# Oracle Naming Conventions

### SIDs and Databases

SIDs and databases should be named: ***AAA[n]\_TTT*** where

| ***AAA*** | is a meaningful three character abbreviation for the application using this database. It should be the three character name assigned for ECS/Control-M |
| --- | --- |
| ***n*** | is a one character alphanumeric version indicator (optional) |
| ***TTT*** | is a meaningful indication of the version, environment and use of the database. Examples are: **DEV** for development, **PPD** for pre-production, **PRD** for production, **RPT** for reporting, **TRN** for training and **CNB** for crash and burn. |

**Examples:** OOD1\_DEV, AMS2\_RPT**Note:**

* + The SID and database names can be a maximum of 8 characters.
  + SID names have to be unique across the network.
  + The corresponding database name must always match the SID name.
  + SID and database names are always uppercase.

### Tables

[See Sybase naming conventions](http://www.lifecycle-toolkit.com/tools/NmngCnv/dbstnds/namsho.htm#sytables)

### Views

[See Sybase naming conventions](http://www.lifecycle-toolkit.com/tools/NmngCnv/dbstnds/namsho.htm#syviews)

### Columns

[See Sybase naming conventions](http://www.lifecycle-toolkit.com/tools/NmngCnv/dbstnds/namsho.htm#sycolumn)

### Indexes

[See Sybase naming conventions](http://www.lifecycle-toolkit.com/tools/NmngCnv/dbstnds/namsho.htm#syindexes)

### Physical Data Types

In some DBMSs, you can specify a particular data type more than one way. Whenever possible use the ANSI Standard definitions for the data type. However, this standard does not restrict the use of non ANSI standard data types.The following table shows the mapping of ANSI standard data types to the Oracle implementation.

| **ANSI Standard** | **Oracle Type** | **Physical Format** |
| --- | --- | --- |
| CHAR(n) | CHAR(n) | max 255 characters |
| CHAR VARYING(n) | VARCHAR2(n) | max 2000 characters |
| n/a | LONG | max 2gb characters |
| NUMERIC(p,s) | NUMBER(p,s) | precision = 1 thru 38  scale = -84 thru 127 |
| DECIMAL(p,s) | NUMBER (p,s) | precision = 1 thru 38  scale = -84 thru 127 |
| FLOAT(p) | NUMBER | precision = 38 |
| REAL | NUMBER | precision = 38 |
| DOUBLE PRECISION | NUMBER | precision = 38 |
| INTEGER | NUMBER(38) | precision = 38 |
| SMALLINT | NUMBER(38) | precision = 38 |
| n/a | DATE | 8 bytes |
| n/a | RAW | max 2000 binary data |
| n/a | LONG RAW | max 2gb binary data |
| n/a | ROWID |  |

### Constraints

A constraint is a rule that restricts the values for one or more columns in a table.Constraint names should be named: ***tablename\_type\_desc*** where

| ***tablename*** | is the name of the table to which the constraint applies | |
| --- | --- | --- |
| ***type*** | identifies the type of constraint and can only be: | |
| **p** for primary key | **f** for foreign key |
| **c** for check constraint |
| ***desc*** | adds additional information on the columns named in the constraint or their usage | |

**Example:**emp\_p\_empno Constraint on primary key column empno for table emp**Note:**Constraints should not be used to enforce uniqueness. Instead, a unique index should be created to accomplish the result of the unique constraint.

### Triggers

A trigger is a stored block of SQL statements that is associated with a table. ORACLE automatically executes a trigger when a specified SQL statement is issued against the table.Trigger should be named: ***tablename \_x\_y \_*trg** where

| ***tablename*** | is the name of the table to which the trigger is applied |
| --- | --- |
| ***x*** | identifies the time of the trigger and can only be:  **b** for before  **a** for after |
| ***y*** | identifies the type of trigger and can only be:  **d** for delete  **i** for insert and  **u** for update |
| **trg** | is a constant |

**Examples:** employee\_b\_di\_trg, department\_a\_diu\_trg

### Stored Procedures

[See Sybase naming conventions](http://www.lifecycle-toolkit.com/tools/NmngCnv/dbstnds/namsho.htm#syprocedures)

### Packages

A package is an encapsulated collection of related procedures, functions, and other program objects stored together in the database.Packages should be named: ***name\_*pkg** where

| ***name*** | is a meaningful name for the stored procedures it contains |
| --- | --- |
| **pkg** | is a constant |

**Examples:** employee\_maint\_pkg, contract\_update\_pkg

### Roles

A role is a set of privileges that can be granted to users or to other roles.Roles should be named: ***name******\_*rol** where

| ***name*** | is a descriptive name |  |
| --- | --- | --- |
| **rol** | is a constant |  |

**Examples:** security\_admin\_rol

### Profiles

Profiles are used to control the use of system resources and are associated with one or many users.Profiles should be named: ***name******\_*prf** where

| ***name*** | is a descriptive name |
| --- | --- |
| **prf** | is a constant |

**Examples:** adhoc\_rpt\_prf

### Tablespaces

A tablespace is an allocation of space in the database that contains database objects.Tablespaces should be named: ***nametype*** where

| ***name*** | is a meaningful name for the usage of the tablespace |
| --- | --- |
| ***type*** | is the type of tablespace |
| Examples are: **d** for data and **x** for index. |

**Examples:** gldetaild, glsummaryx**Note:**

* + The tablespace that stores the rollback segment should be named **rbs**.
  + The tablespace that stores the temporary segments should be named **temp**.

### Rollback Segments

A rollback segment is an object that is used by ORACLE to store data necessary to reverse, or undo, changes made by transactions.Rollback segments should be named: **rbs*nn*** where

| **rbs** | identifies this object as a rollback segment |
| --- | --- |
| ***nn*** | is a unique number from 01 to 99 (include leading zero for sorting ) |

**Example:** rbs05

### Data Files

Data files are physical files belonging to a tablespace.Data files should be named:  
(dir name)/***tablespacename***\_***SIDNAME\_nn***.**dbf** where

| ***tablespacename*** | is the name of the associated tablespace |
| --- | --- |
| ***SIDNAME*** | is the name of the SID/ database |
| ***nn*** | is a unique number from 01 to 99 (include leading zero for sorting) |
| **dbf** | is a constant |

**Example**: ...gldetaild\_FIN1\_DEV\_01.dbf

### Control Files

Control files should be named:  
(dir name)/***SIDNAME***\_***n***.**ctl** where

| ***SIDNAME*** | is the name of the SID/database |
| --- | --- |
| ***n*** | identifies a number of 1 or higher |
| **ctl** | is a constant |

**Example**: ...FIN1\_DEV\_1.ctl

### Redo log Files

Redo log files should be named:(dir name)/**redo**\_***groupnn****\_****membernn***.**log** where

| **redo** | is a constant |
| --- | --- |
| ***groupnn*** | is the number of the redo log group (betw 01 & nn) |
| ***membernn*** | is the number of the redo log members (betw 01 & maxlogfiles) |
| **log** | is a constant |

**Example**: ...log\_01\_02.rdo

### Parameter Files

Parameter files should be named:(dir name)/**init*SIDNAME.*ora** where

| **init** | is a constant |
| --- | --- |
| ***SIDNAME*** | is the name of the SID/database |
| **ora** | is a constant |

**Example**: ...initOOD1\_DEV.ora

### Archive Files

Archive files should be named:<dir name>/**oracle\_*SIDNAME\_n\_m.*arc** where

| **oracle** | is a constant |
| --- | --- |
| ***SIDNAME*** | is the name of the SID/database |
| ***n*** | is the thread number (between 1 & n) |
| ***m*** | is the sequence number (between 1 & n) |
| **arc** | is a constant |

**Example**: ...oracle\_OOD1\_DEV\_1\_2.arc

### Service Names

Service names only apply to SQL\*Net V2.0. A service name is the logical name that identifies a SID on a particular host to which a connection can be made. They should be named the same as the SID names.

### Database Links

Database links can be created as private or public. At Stanford University, private links will not be used unless there are application-specific requirements to do so. Only public database links are used. They should be named the same as the SID names.

### Snapshot Names

A snapshot is a table that contains the results of a query of one or more tables or views, often located on a remote database.Snapshot names should be named: ***SIDNAME\_tablename*** where

| ***SIDNAME*** | is the same as the SID or database name of the source system |
| --- | --- |
| ***tablename*** | is the name of the table from which a snapshot is created |

**Examples:** OOD1\_DEV\_donor, AMS2\_RPT\_student

# Summary of Naming Conventions

# Sybase

| **Object Type** |  | **Object Naming convention** |  | **Examples** |
| --- | --- | --- | --- | --- |
| SQL Servers |  | *name\_vvv* |  | adhoc\_prd, txn\_dev |
| Databases |  | *aaa[\_v]\_ttt* |  | dems\_3\_dev, ehs\_trn |
| Tables |  | *eeee\_eeee\_eeee* |  | sales\_detail |
| Views |  | *vvvv\_vvvv\_vvvv\_*v |  | sales\_by\_product\_v |
| Columns |  | *aaaa\_aaaa\_aaaa* |  | employee\_number |
| Indexes |  | *tablename\_*[u][c]\_idx\_*n* |  | fin\_roll\_summary\_uc\_idx\_1 |
| User-defined data types |  | *ss\_name* |  | in\_branch\_number |
| Rules |  | *name* |  | branch\_number |
| Defaults |  | *name* |  | commission\_percentage |
| Constraints |  | *tablename\_*[c]*\_type* |  | carrier\_pky, employee\_c\_unq |
| Triggers |  | *tablename\_*[d][i][u] |  | air\_segment\_i |
| Procedures |  | *name\_action* |  | carrier\_get |
| Server logins and Users |  | *f[m]lllll* |  | mhart, mtpuff |
| Database device (logical) |  | *devicetype\_n* |  | log\_1, data\_5 |
| Segments |  | *index\_n, data\_n, objectname\_n* |  | index\_1, data\_23, person\_1 |
|  |  |  |  |  |

# ORACLE

| **Object Type** |  | **Object Naming convention** |  | **Examples** |
| --- | --- | --- | --- | --- |
| SIDs and Databases |  | *AAA[n]\_VVV* |  | OOD1\_DEV, AMS\_RPT |
| Tables |  | *eeee\_eeee\_eeee* |  | sales\_detail |
| Views |  | *vvvv\_vvvv\_vvvv\_*v |  | sales\_by\_product\_v |
| Columns |  | *aaaa\_aaaa\_aaaa* |  | employee\_number |
| Indexes |  | *tablename\_*[u][c]\_idx\_*n* |  | fin\_roll\_summary\_uc\_idx\_1 |
| Constraints |  | *tablename\_type\_desc* |  | emp\_p\_empno |
| Triggers |  | *tablename\_x\_y* |  | employee\_b\_di |
| Stored Procedures |  | *name\_action* |  | carrier\_get |
| Packages |  | *name* |  | employee\_maint |
| Roles |  | *name* |  | security\_admin |
| Profiles |  | *name* |  | adhoc\_rpt |
| Tablespaces |  | *nametype* |  | gldetaild, glsummaryx |
| Rollback Segments |  | rbs*nn* |  | rbs05 |
| Data Files |  | *<dirname>tblspacename\_SIDNAME\_nn*.dbf |  | ...gldetaild\_FIN1\_DEV\_01.dbf |
| Control Files |  | *<dirname>SIDNAME\_nn*.ctl |  | ...FIN1DEV\_01.ctl |
| Redo Log Files |  | *<dirname>redo\_groupnn\_membernn*.log |  | ...redo\_01\_02.log |
| Parameter Files |  | *<dirname>initSIDNAME*.ora |  | ...initFIN1DEV.ora |
| Archive Files |  | *<dirname>/oracle\_SIDNAME\_n\_m*.arc |  | ...OOD1\_DEV\_1\_2.arc |
| Service Names |  | *SIDNAME* |  | OOD1\_DEV, AMS\_RPT |
| Database Links |  | *SIDNAME* |  | OOD1\_DEV, AMS\_RPT |
| Snapshot Names |  | *SIDNAME\_tablename* |  | OOD1\_DEV\_donor |