ScriptsDB - Proceso: Estándares y Prefijos

Esta sección presenta a continuación las normas de denominación de variables y objetos estándar declaradas dentro de las bases de datos relacionales (Oracle PLSQL) de la Compañía. El estándar ha sido generado teniendo presente el estándar Oracle el cual puede ser consultados desde Nombres y convenciones Oracle

Notas de Versión

Versión	Elabora	Revisa	Aprueba	Fecha Publicación	Observaciones
1.0	carlos.torres, abdimar.estrada	daberson.henao	daberson.henao	05/10/2020	Versión Inicial

Tablas y Entidades

Las siguientes reglas se aplican a la denominación de las Tabla y Entidades:

- No debe superar una longitud máxima de 30 caracteres.
- Debe construirse mediante la separación de palabras con guiones bajos “_”.
- Los nombres de tabla deben ser construidos a partir de palabras completas cuando sea posible. Por ejemplo, la tabla “SERVICE_ORDER” es mejor que “SRV_ORD”.
- Los nombres de tablas serán singular, no en plural.
- No utilice palabras reservadas de Oracle para un nombre de tabla.
- Las tablas con particiones deben empezar por PT_
- Las tablas external deben empezar por EXT_

Ejemplo:

```
* APPLICATIONS
* APPLICATION_FUNCTIONS
* APPLICATION_FUNCTION_ROLES
```

Definición de Nombres

A continuación se establecen los criterios para la definición de nombres de tablas:

- Tablas de Maestros: MAE_
- Tablas de Detalles: DET_

Tablas Alias / Alias para Consultas, Vistas

Todas las tablas deben tener un nombre de alias único (4 caracteres en lo posible). Por lo general constan de una abreviatura del nombre de la tabla.

Ejemplo:

```
  * APPLICATIONS = APPL (4)
  * APPLICATION_FUNCTIONS = APFU (2:2)
  * APPLICATION_FUNCTION_ROLES = APFR (2:1:1)
```

Vistas

Las siguientes reglas se aplican a la denominación de las vistas:

- No debe superar una longitud máxima de 30 caracteres incluyendo la palabra clave.
- Debe construirse mediante la separación de palabras con guiones bajos “_”.
- Debe empezar por V_ .
- Las vistas Materializadas deben empezar VM_
- Los nombres deben estar en singular, no en plural.
- No debe utilizar palabras reservadas de Oracle para el nombre de la vista.

Ejemplo:

```
  * V_TERCEROS
  * VM_TERCEROS
```

Atributos y Columnas

Las siguientes reglas se aplican a la denominación de las columnas:

- No debe superar una longitud máxima de 30 caracteres.
- Las palabras en el nombre deben estar separados por un guion bajo.
- Deberán evitarse las abreviaturas, prefiriendo palabras completas, siempre que sea posible.
- Los nombres de columna y componentes de nombre serán en singular, no en plural.
- Los nombres de columna que se refieren a una bandera booleana o estado tendrán un nombre que empezará por FLG_ . Se informa que una bandera debe ser utilizada cuando un valor sólo puede ser uno de dos estados y debe estar definida como NOT NULL.
- Los nombres de columna que se refieren a un código deben empezar con COD_
- Los nombres de columna que se refieren a un LOB deben empezar con LOB_
- Los nombres de columna que se refieren a un CLOB deben empezar con CLOB_
- Los nombres de columna que se refieren a un BLOB deben empezar con BLOB_
- Las columnas CLOB/BLOB deben inicializarse con la función EMPTY_CLOB/EMPTY_BLOB según el tipo
- No utilice tipos binarios para almacenamiento de recursos o archivo evalúe el uso de BLOB o puede utilizar CLOB y almacenar el recurso en Base64 de esta forma optimiza el espació y se permite la lectura del recurso sin herramientas de terceros o visores especiales
- No utilizar palabras reservadas de Oracle en el nombre de columna.
- No utilizar el tipo de dato CHAR como opción utilice VARCHAR2.
- Las columnas VARCHAR2 deben estar definidas en base binaria (1,2,4,8,16,32,64,128,256…).

- Las columnas DATE deben empezar con la palabra FECHA_.

Ejemplo:

```
* FLG_ACTIVO NUMBER
* FECHA_REGISTRO DATE
* DESCRIPCION  VARCHAR2(256)
* CLOB_MEMO CLOB DEFAULT EMPTY_CLOB()
```

Llaves o Claves

Primarias

- El nombre debe estar en singular.
- Puede estar compuesta por el alias de la tabla.
- Cuando la llave primaria este compuesta de un solo campo el nombre debe ser el nombre de la tabla o su alias y debe empezar por PK_ Ej: PK_TERCERO
- Cuando la llave primaria este compuesta por más de un campo el nombre debe ser el nombre de la tabla o su alias y debe empezar por PKC_ Ej: PKC_TERCERO
- No debe superar una longitud máxima de 30 caracteres.

Foráneas

- El nombre debe estar en singular.
- Puede estar compuesta por el alias de la tabla referenciada.
- Cuando la llave foránea está compuesta por un solo campo, debe empezar por FK_[nombre de la tabla][identificador]. El identificador será el número de referencia que corresponde a la tabla asociada empezando por 01, 02, 03, 04…NM EJ: FK_TERCERO01.
- Cuando la llave foránea está compuesta por varios campos, debe empezar por FKC_[nombre de la tabla][identificador]. El identificador será el número de referencia que corresponde a la tabla asociada empezando por 01, 02, 03, 04…NM EJ: FKC_TERCERO01.
- No debe superar una longitud máxima de 30 caracteres.

Únicas

- El nombre debe estar en singular.
- Puede estar compuesta por el alias de la tabla.
- Cuando la llave única está compuesta por un solo campo, debe empezar por UK_[nombre de la tabla][identificador]. El identificador será el número de referencia que corresponde a la siguiente llave única en la tabla empezando por 01, 02, 03, 04…NM EJ: UK_TERCERO01.
- Cuando la llave única está compuesta por varios campos, debe empezar por UKC_[nombre de la tabla][identificador]. El identificador será el número de referencia que corresponde a la siguiente llave única en la tabla empezando por 01, 02, 03, 04…NM EJ: UKC_TERCERO01.
- No debe superar una longitud máxima de 30 caracteres.

Índices

Índice de Clave Primaria

- El nombre debe estar en singular.
- Puede estar compuesto por el alias de la tabla.
- Debe empezar por IDX_
- En lo posible utilizar el mismo nombre utilizado para la clave EJ: PK_TERCERO → IDX_TERCERO.
- En lo posible almacenar los índices en un tablespace diferente al que se utilizó para la definición de la tabla.
- No debe superar una longitud máxima de 30 caracteres.

Índice de Clave Externa

- El nombre debe estar en singular.
- Puede estar compuesto por el alias de la tabla.
- Si el índice no es un índice de mapa de bits el nombre estará formado por el nombre de la restricción de clave externa y debe empezar por IDX_
- Si el índice es un índice de mapa de bits debe empezar por BIDX_
- En lo posible almacenar los índices en un tablespace diferente al que se utilizó para la definición de la tabla.
- No debe superar una longitud máxima de 30 caracteres.

Índice de base Función

- El nombre debe estar en singular.
- Puede estar compuesto por el alias de la tabla.
- Debe empezar por FBI_
- En lo posible almacenar los índices en un tablespace diferente al que se utilizó para la definición de la tabla.
- No debe superar una longitud máxima de 30 caracteres.

Índice de Clave Única

- El nombre debe estar en singular.
- Puede estar compuesto por el alias de la tabla.
- Si el índice no es un índice de mapa de bits, ó basado en función debe empezar por IDX_.
- Si el índice es un índice de mapa de bits debe empezar por BIDX_
- índices únicos no se crearán a menos que exista una restricción única correspondiente.
- En lo posible almacenar los índices en un tablespace diferente al que se utilizó para la definición de la tabla.
- En lo posible utilizar el mismo nombre utilizado para la clave única EJ: UK_TERCERO01 → IDX_TERCERO01.
- No debe superar una longitud máxima de 30 caracteres.

Otros Índices

- El nombre debe estar en singular.
- Puede estar compuesto por el alias de la tabla.
- Si el índice no es un índice de mapa de bits, ó basado en función debe empezar por IDX_.
- Si el índice es un índice de mapa de bits debe empezar por BIDX_
- En lo posible almacenar los índices en un tablespace diferente al que se utilizó para la definición de la tabla.
- No debe superar una longitud máxima de 30 caracteres.

Secuencias

- El nombre debe empezar por SEQ_.
- El nombre no puede ser superior a 30 caracteres de longitud.
- No utilice palabras reservadas de Oracle para componentes de nombre de secuencia.
- Cuando una secuencia es utilizado por una tabla se debe utilizar el alias de la tabla ej: SEQ_TERCEROS.
- Cuando una secuencia es utilizada por varias tablas se deben usar los alias de las tablas que la utilizan ej: SEQ_TER_PLAC.

Triggers

- El nombre debe empezar por TRG_.
- El nombre no puede superar los 30 caracteres de longitud.
- El nombre debe seguir el siguiente convenio de denominación: TRG_ (activación: Before= B, After: A)_(operación: Insert= I, Update= U, Delete=D)_tabla Ej: TRG_BI_TERCERO

Check Constraints

- El nombre debe empezar CHK_
- El nombre no puede superar los 30 caracteres de longitud.

Roles

- Deben empezar por ROL_
- El nombre no puede superar los 30 caracteres de longitud.

DBLinks

- El nombre debe empezar por DBL_
- El nombre debe seguir el siguiente convenio de denominación: DBL_ (Alias Base de Datos Destino)_(Alias Usuario Base de Datos). Ej: DBL_CALIDAD_PRESUP.

Directorios

- El nombre debe empezar DIR_

Bases de Datos

Las reglas para la creacion de bases de datos serán definidas por el equipo de implantación sin embargo se deben considerar las siguientes recomendaciones:

- El nombre debe estar en mayusculas.
- El nombre debe intuir el nombre del cliente (utilizar alias).
- De ser posible asociar al nombre un identificador numerico. EJ DESARROLLO00

PL/SQL

Paquetes

- Debe empezar por PKC_, ó PKG_
- Los nombres de paquetes no pueden superar los 30 caracteres de longitud.
- No utilice palabras reservadas de Oracle para componentes de nombre del paquete.

Funciones

- Debe empezar por FN_
- Los nombres de funciones no pueden exceder los 30 caracteres de longitud.
- No utilice palabras reservadas de Oracle para componentes de nombre de función.

Procedimientos Almacenados

- Debe empezar por SP_
- los nombres de procedimiento no pueden superar los 30 caracteres de longitud.
- No utilice palabras reservadas de Oracle para componentes de nombre procedimiento.

Variables

Alcance

- Global: g
- Local: l
- Constante: c

Tipos de Datos

- Number: n_
- Varchar2: v_
- Date: dt_
- Tipos: tpy_
- Cursor: cur_
- Exception: exc_

Parametros

- Entrada: ai
- Salida: ao
- Entrada/Salida: aio

Extenciones de Archivos

Se permite el tipo generico sql para todos los archivos ó según el tipo de objeto se permiten los siguientes tipos:

- Especificacion de Paquete: .pks
- Cuerpo de Paquete: .pkb
- Consultas: .sql
- Triggers: .trg
- Funciones: .fnc
- Procedimientos: .prc
- Vistas: .v
- Vistas Materializadas: .vm
- Especificación deTipo: .tps
- Cuerpo de Tipo: .tpb

Recomendaciones:

La siguientes recomendaciones no imponen reglas de utilización sin embargo trate en lo posible de aplicarlas ya que el hacerlo optimiza el código generado:

- Agrupe funciones y procedimientos en paquetes.
- Evite la función raise_application_error utilice en su lugar Raise y defina excepciones personalizadas utilizando el inicializador pragma exception_init.
- Evite los operadoradores SQL% ???? en su lugar utilice capturadores begin exception.
- Evite el uso del Open al abrir cursores en su lugar utilice el bucle for.
- Todas las consultaas select into deben estar incluidas dentro de un bloque begin exception y al menos debe validar las excepciones NOT_DATA_FOUND y TOO_MANY_ROWS
- Evite el uso de las funciones de formatos en consultas pesadas
- TO_CHAR: si valida fechas utilice mejor el operador EXTRACT
- TRUNC en variables numéricas trate de utilizar el operador matematico que desea aplicar.
- Utilice los campos de las tablas de manera explícita. Evite el uso de SELECT * o INSERT INTO

VALUES sin especificación de columnas.
- Siempre coloque el nombre del esquema propietario de los objetos para el uso de los mismos (Tablas, Vistas, Secuencias, Funciones, Procedimientos, Paquetes, Triggers, etc). Por ejemplo TESORE01.maestro_terceros.
- Utilice la función NVL, NLV2, DECODE o CASE, en los SELECT cuando no se soporten valores nulos.
- No utilizar Caracteres especiales.
- No utilizar Carácter 'ñ'.
- No utilizar Vocales con tilde, ni siquiera en los comentarios.
- Siempre valide las operaciones de división para evitar la división entre cero.
- Utilizar %TYPE y %ROWTYPE para la definición de variables.
- Borrar código y variables que no se usen. Si conocemos una parte de un código o variable que no se va a usar más debemos eliminarla, asegurando previamente que no tiene más presencia en el código. Evitar el código en comentarios.
- Evitar las conversiones implícitas de tipos de datos. Se deben utilizar las funciones predefinidas para la conversión (TO_DATE, TO_CHAR, TO_NUMBER).
- Luego de cualquier instrucción de manipulación de Datos – DML (INSERT, UPDATE, DELETE), se debe realizar un COMMIT;.
- Al crear registros en las tablas de uso general para todos los clientes, se debe primero generar un INSERT con los valores genéricos y luego un UPDATE para valores específicos para un cliente dado.
- Tener especial cuidado en las modificaciones posteriores de una instrucción. Por ejemplo, un CREATE TABLE debe tener modificaciones posteriores con un ALTER TABLE. Una definición de Funciones, procedimientos y paquetes siempre deben manejarse con CREATE OR REPLACE.
- Las instrucciones DDL no llevan COMMIT.
- Las instrucciones ejecutables deben terminar siempre con un “;”.
- Para la ejecución de bloques ejecutables continuos (CREATE OR REPLACE) se debe colocar el “/”.
- Verificar que las consultas que se componen de subconsultas que deben retornar solo una fila, no retornen más de una fila.
- No se deben mezclar instrucciones DML con instrucciones DDL.
- No se deben utilizar comandos DROP, a menos que sean para actualizaciones.
- Toda instrucción DELETE debe tener un WHERE.
- Manejar la empresa siempre en los querys, si no necesita la empresa poner la empresa por defecto ‘9999999999’ (en los esquemas multiempresa).
- Los comentarios de linea se definen con –
- Los comentarios de bloque se definen con /* */

←Volver atrás

From:
https://wiki.adacsc.co/ - **Wiki**

Permanent link:
https://wiki.adacsc.co/doku.php?id=ada:howto:sicoferp:factory:sicoferpscriptsdb:process:standardandpref&rev=1607003772

Last update: **2020/12/03 13:56**