



MANUAL DE ESTANDARES DE MODELAMIENTO DE DATOS v1.0

**Intendencia Nacional de Sistemas de Información
2005**

INDICE

1. OBJETIVO	4
2. ALCANCE	4
3. DESCRIPCIÓN GENERAL	4
4. RESPONSABILIDADES.....	4
5. MODELAMIENTO CONCEPTUAL DE DATOS	5
5.1 DE LA ENTIDAD.....	5
5.1.1 <i>Nombre de la Entidad</i>	5
5.1.2 <i>Reglas de la Entidad</i>	5
5.1.3 <i>Atributos de la Entidad</i>	5
5.1.4 <i>Relaciones y Claves Foráneas de la Entidad</i>	5
5.2 DE LOS ATRIBUTOS.....	6
5.2.1 <i>Nombre de los Atributos</i>	6
5.2.2 <i>Clave Primaria–Atributos</i>	6
5.2.3 <i>Sintaxis</i>	6
5.2.4 <i>Reglas de los Atributos</i>	6
5.2.4.1 <i>Nombre</i>	6
5.2.4.2 <i>Atributo Propio</i>	7
5.2.4.3 <i>Atributo Derivado</i>	7
5.2.4.4 <i>Atributos – Clave Primaria</i>	7
5.2.5 <i>Nomenclatura del Código del Atributo</i>	7
5.3 DE LAS RELACIONES	8
5.3.1 <i>Cardinalidad</i>	8
5.3.2 <i>Clasificación de las Relaciones</i>	9
5.3.2.1 <i>Relación Identificadora</i>	9
5.3.2.2 <i>Relación No-Identificadora</i>	9
5.3.2.3 <i>Relación Mandatoria No-Identificadora</i>	9
5.3.2.4 <i>Relación Opcional No-Identificadora</i>	9
5.3.2.5 <i>Relaciones Recursivas</i>	10
5.3.3 <i>Nombre de la Relación</i>	10
5.3.4 <i>Reglas de las Relaciones</i>	11
5.3.4.1 <i>Composición</i>	11
5.3.4.2 <i>Relación Identificadora/relación No Identificadora</i>	11
5.3.4.3 <i>Relación Total/Relación Parcial</i>	11
5.3.4.4 <i>Entidad Independiente/ Entidad dependiente</i>	11
6. MODELAMIENTO FISICO DE DATOS	13
6.1 CRITERIOS GENERALES	13
6.2 CRITERIOS ESPECIFICOS	14
6.3 ONLINE Y BASES DE DATOS	16
6.3.1 <i>Nomenclatura Online, Instancia ó Data Server</i>	16
6.3.2 <i>Nomenclatura de Base de Datos</i>	16
6.4 OBJETOS DE LA BASE DE DATOS.....	16
6.4.1 <i>Objetos Esquema de la Base de Datos</i>	16
6.4.1.1 <i>Tabla - Nomenclatura</i>	16

6.4.1.2 Tabla Temporal - Nomenclatura.....	17
6.4.1.3 Columnas – Nomenclatura	17
6.4.1.4 Índices – Nomenclatura	18
6.4.1.5 Clave Primaria (Primary Key).....	19
6.4.1.6 Clave Foránea (Foreign Key)	20
6.4.1.7 Check - Nomenclatura	21
6.4.1.8 Vista - Nomenclatura.....	21
6.4.1.9 Sinónimo - Nomenclatura.....	22
6.4.1.10 Trigger - Nomenclatura	22
6.4.1.11 Procedimiento Almacenado – Nomenclatura (Stored Procedure).....	22
6.4.1.12 Función - Nomenclatura	23
6.4.1.13 Package - Nomenclatura.....	23
6.4.1.14 Secuencia - Nomenclatura	23
6.4.1.15 Resumen de Prefijos de Objetos Esquema	23
6.4.2 <i>Objetos no Esquema de la Base de Datos</i>	23
6.4.2.1 DbSPACE ó Tablespace - Nomenclatura	24
6.4.2.2 Chunks ó Datafiles - Nomenclatura	24
Esta sección del manual se encuentra en proceso de definición por la División de Soporte Técnico.....	24
6.4.2.3 Usuarios Propietarios - Nomenclatura.....	24
6.4.2.4 Roles - Nomenclatura	24
6.4.2.5 Profile - Nomenclatura	25
7. TERMINOLOGÍA	25
ANEXO I: ROLES Y USUARIOS INTRANET, INTERNET, EXTRANET	30

1. OBJETIVO

Uniformizar criterios para la creación de objetos en la base de datos y el correcto uso en el desarrollo de sistemas de información facilitando su lectura e implementación y la eficiente Administración de los Modelos de Datos.

2. ALCANCE

Todos los objetos a ser creados en la Base de Datos por el personal de la Gerencia de Desarrollo de Sistemas Aduaneros y la Gerencia de Desarrollo de Sistemas Tributarios y de Gestión.

3. DESCRIPCIÓN GENERAL

El presente manual contiene las reglas y estándares a ser utilizados en el análisis y diseño de los Modelos de Datos durante las fases de Modelamiento de Requerimientos y Modelamiento de Tecnología de la MDSI (Metodología de Desarrollo de Sistemas de Información).

El modelamiento de Datos, comprende el modelamiento Conceptual y Físico de Datos.

El Modelamiento Conceptual de Datos, hace referencia al documento de estándares internacionales: IEEE Std 1320.2-1998 Standard for Conceptual Modeling Language Syntax and Semantics for IDEF1X₉₇

El Modelamiento Físico de Datos contempla la creación de los objetos esquema y no-esquema de las bases de datos (para manejadores de base de datos: Oracle, Informix, Mysql).

4. RESPONSABILIDADES

Es responsabilidad del personal de la Gerencia de Desarrollo de Sistemas Aduaneros y de la Gerencia de Desarrollo de Sistemas Tributarios y de Gestión utilizar los estándares definidos en el presente manual al realizar el Análisis y Diseño en las fases de Modelamiento de Requerimientos y de Tecnología.

Es responsabilidad de la Oficina de Ingeniería de Procesos y Software verificar el cumplimiento del uso de los estándares establecidos en el presente manual en la fase de Modelamiento de Tecnología, así como tener actualizado el presente documento.

Es responsabilidad de la División de Control de Calidad recibir los pases a producción con la aprobación de la Oficina de Ingeniería de Procesos y Software, con relación a lo establecido en el presente manual.

Es responsabilidad de la División de Soporte Técnico realizar el soporte al desarrollo de los sistemas de información sobre la base de los estándares establecidos en el presente manual.

Es responsabilidad de la División de Operaciones recibir los pases a producción previa aprobación de los estándares establecidos por la Oficina de Ingeniería de Procesos y Software, así como de la División de Control de Calidad, los cuales serán referenciados en los pases a producción.

5. MODELAMIENTO CONCEPTUAL DE DATOS

El Modelo conceptual de datos representa la estructura lógica de la base de datos, que es independiente del software y de la estructura de almacenamiento de datos.

5.1 DE LA ENTIDAD

5.1.1 Nombre de la Entidad

- a) El nombre de la entidad es un sustantivo que describe lo que la entidad representa.
- b) El nombre debe ser definido en singular.
- c) Se permiten las abreviaturas y siglas.
- d) El nombre de la entidad debe ser consistente y debe reflejar el significado de lo que representa.
- e) Gráficamente una entidad es representada por un rectángulo.

5.1.2 Reglas de la Entidad

- a) Una entidad debe tener un nombre único.
- b) El mismo significado también debe aplicarse al mismo nombre de la entidad.
- c) El mismo significado no puede ser aplicado a diferentes nombres de entidades, a excepción de que los nombres sean alias o sinónimos de la entidad.
- d) La entidad no debe tener el mismo nombre que los atributos.
- e) No debe tenerse dos entidades, cuyos nombres sean sinónimos. Dos nombres son sinónimos si cada uno de ellos es directa o indirectamente un alias de otro o si existe un tercer nombre que es un alias de los dos nombres.
- f) Cada entidad debe tener múltiples ocurrencias.

5.1.3 Atributos de la Entidad

- a) Una entidad puede tener uno o más atributos, cuyos valores únicos identifican cada instancia de la entidad.
- b) Una entidad debe tener uno o más atributos que son propios de la entidad o derivados de otra entidad a través de las relaciones.

5.1.4 Relaciones y Claves Foráneas de la Entidad

- a) Una entidad puede tener una o muchas relaciones con otras entidades en un modelo de datos.
- b) Si una clave foránea es contenida completamente en la clave primaria de una entidad, luego la entidad debe ser dependiente.

- c) Si algunos atributos de la clave foránea ó que no sean parte de la clave foránea son usados como parte de la clave primaria de una entidad entonces la entidad será independiente.

5.2 DE LOS ATRIBUTOS

5.2.1 Nombre de los Atributos

- a) Cada atributo es identificado y tiene un nombre único.
- b) El nombre es expresado como un sustantivo que describe las características del atributo.
- c) Son permitidas las abreviaturas y las siglas.
- d) El nombre del atributo debe ser significativo, consistente y debe estar representado en el modelo.
- e) En el Modelo de Datos, un atributo debe ser designado como opcional cuando el mapeo es parcial (algunas instancias no tienen valor para el atributo)
- f) Un atributo opcional tiene al menos un valor para una instancia de la entidad.
- g) Un atributo no designado como opcional es por defecto **mandatorio**.
- h) Un atributo mandatorio tiene exactamente un valor para cada instancia o registro de la entidad.
- i) Un atributo de una entidad que no es clave foránea se dice que es propia de la entidad.

5.2.2 Clave Primaria–Atributos

- a) Una entidad debe tener un atributo o grupo de atributos que han sido elegidos como identificador único de una entidad.
- b) Este atributo o atributos forma la clave primaria de la entidad.

5.2.3 Sintaxis

- a) Los atributos deben ser mostrados como listados de nombres en la entidad.
- b) Los atributos que no son parte de la clave primaria pueden no tener un valor (no es obligatorio) Por ello en los modelos transaccionales los atributos tendrán un valor por defecto, como un espacio en blanco, 0 y en los casos que sea necesario indicar valores nulos estos deben ser fundamentados.
- c) Los atributos que especifican la clave primaria deben estar posicionados en la parte superior de la lista de atributos y estar marcadas y/o separadas de los demás atributos.

5.2.4 Reglas de los Atributos

5.2.4.1 Nombre

- a) Cada atributo debe tener un nombre único. El mismo significado no debe aplicarse a diferentes nombres a menos que los nombres del atributo sean alias o sinónimos de estos.
- b) Un atributo debe ser nombrado por su nombre de atributo o uno de su alias.

- c) En un modelo de datos, si un atributo es un atributo propio en una entidad y es un atributo derivado en otra entidad, entonces debe tener el mismo nombre en ambos.
- d) Ningún modelo puede contener dos nombres de atributos en los que los nombres son alias o sinónimos. Dos nombres son sinónimos si cada uno es directa o indirectamente un alias del otro, ó si hay un tercer nombre para que ambos denominen su alias.

5.2.4.2 Atributo Propio

- a) Una entidad puede tener la cantidad de atributos necesarios.
- b) En el modelo entidad relación, cada atributo debe ser propio de una sola entidad.

5.2.4.3 Atributo Derivado

Los atributos derivados, son atributos cuyos valores se pueden determinar ó calcular de otros datos en el modelo.

- a) Una entidad puede tener uno o muchos atributos derivados.
- b) Un atributo derivado formará parte de la clave primaria de una entidad padre relacionada ó de una entidad genérica.

5.2.4.4 Atributos – Clave Primaria

Cada instancia de una entidad debe tener un valor para cada atributo que es parte de su clave primaria.

5.2.5 Nomenclatura del Código del Atributo

Los atributos tienen como parte del nombre del código del atributo, los prefijos que representan el tipo de atributo.

Nombre conceptual	Descripción	Prefijo del nombre del Código del atributo
Año	Se almacena el año en cuatro dígitos.	ann_
Cantidad	Cantidad de elementos de un conjunto.	cnt_
Clave	Claves artificiales creadas para estructuras de datos específicas como; búsqueda de cadenas en RUC, o claves artificiales en Data Warehouse.	clv_
Código	Cadena de caracteres alfanuméricos. También se considera los códigos que indican el tipo de dato, ejemplo: cod_tipacceso, cod_tipdocidedec, etc.	cod_
Descripción	Descripción de códigos asociados. Por ejemplo: Descripción de ubigeo.	des_
Observación	Observación	obs_
Nombre	Nombre de persona, razón social, nombre de archivo	nom_

Nombre conceptual	Descripción	Prefijo del nombre del Código del atributo
Fecha	Fecha	fec_
Indicador	Cadena de caracteres alfanuméricos que indica un conjunto discreto y finito de valores con significado propio.	ind_
Monto	Expresa una cantidad de unidades monetarias.	mt_
Mes	Almacena los datos de tipo mes (MM)	mes_
Número	Cadena de caracteres numéricos.	num_
Periodo	Almacena datos de tipo periodo (AAAAMM)	per_
Porcentaje	Porcentaje	por_
Semana	Semana.	sem_
Expresión	Expresión (fórmulas y concatenación de códigos)	exp_
Hora	Indica la Hora (Formato : HHMISS)	hor_
Dirección	Dirección de calles, ip, url, postal.	dir_
Imagen	Imágenes almacenadas en los campos tipo byte y blob	img_
Archivo	Se almacena Archivos de información en los tipos de datos text y blob.	arc_
Mensaje	Se almacena los mensajes de texto.	msj_
Audio	Audio.	aud_
Valor	Valor del campo	val_
Video	Video	vid_

5.3 DE LAS RELACIONES

Al usar el término relación denotará relaciones específicas a menos que se indique lo contrario.

5.3.1 Cardinalidad

El número de instancias esperado en cada extremo de la relación es llamado *cardinalidad*.

Una relación debe especificar su cardinalidad. Desde la perspectiva de la entidad padre se pueden ver las siguientes cardinalidades:

- Cada instancia de la entidad padre debe tener asociada al menos una instancia de la entidad hijo.
- Cada instancia de la entidad padre puede tener asociada cero ó una instancia asociada de la entidad hijo.
- Cada instancia de la entidad padre esta asociada con un número exacto de instancias de la entidad hijo.
- Cada instancia de la entidad padre esta asociada con cero o más instancias de la entidad hijo (si no se especifica la cardinalidad desde la perspectiva de la entidad padre, esta se da por defecto)

Una relación no específica debe indicar la misma cardinalidad en ambas direcciones de la relación (muchos a muchos)

5.3.2 Clasificación de las Relaciones

Una relación es designada como identificadora, si los atributos de la clave foránea están contenidas en la clave primaria de la entidad hijo. Caso contrario la relación es designada como no-identificadora

La cardinalidad desde el punto de vista de la entidad hijo puede ser:

5.3.2.1 Relación Identificadora

La instancia de la entidad hijo esta identificada por la asociación con la entidad padre. Cada instancia de la entidad hijo debe estar asociada con exactamente una y solo una instancia de la entidad padre. La existencia del hijo en este tipo de entidad depende del padre: el hijo solo existe si existe el padre.

El hijo en una relación Identificadora es siempre dependiente del padre, por ejemplo una instancia de la entidad hijo debe existir solo si esta relacionada a una instancia de la entidad padre. Una relación identificadora es siempre mandatoria desde la perspectiva de la instancia hijo.

5.3.2.2 Relación No-Identificadora

Cada instancia de la entidad hijo puede ser unívocamente identificada sin conocer la instancia asociada de la entidad padre. Por ejemplo, una relación dependiente entre las entidades Comprador y Orden de Compra, las órdenes de compra pueden ser únicamente identificadas por un número de orden de compra sin identificar la compra asociada.

- a) Ambos padre y entidades hijos deben identificar entidades independientes en una relación No-Identificadora a menos que una o ambas sean entidades hijos en alguna otra relación que sea una relación Identificadora.
- b) Una relación No-Identificadora debe ser designada como obligatoria u opcional desde la perspectiva de la instancia de la entidad hijo.

5.3.2.3 Relación Mandatoria No-Identificadora

En una relación No-Identificadora obligatoria, cada instancia de la entidad hijo esta relacionada exactamente con una instancia de la entidad padre.

5.3.2.4 Relación Opcional No-Identificadora

- a) En una relación opcional No-Identificadora, cada instancia de la entidad hijo está relacionada a cero o a una instancia de la entidad padre.

- b) Una relación opcional No-Identificadora debe representar una dependencia condicional. Una instancia de la entidad hijo en la cual cada atributo de la clave foránea para la relación tiene un valor, debe tener una instancia padre asociada en la cual los atributos de la clave primaria del padre son equivalentes en valor a los atributos de la clave foránea del hijo.

5.3.2.5 Relaciones Recursivas

- a) Una entidad puede participar en una relación en la cual es tanto padre como hijo. Tales relaciones son llamadas *recursivas*.
- b) Las relaciones recursivas son permitidas. Sin embargo, la recursividad debe incluir al menos una relación No-Identificadora.

5.3.3 Nombre de la Relación

- a) Una relación debe tener un nombre, el cual debe expresarse como un verbo o una frase verbal. Ejemplo: tiene, pertenece a, es asignado, etc.
- b) El nombre de cada relación entre dos entidades debe ser único, pero un nombre de la relación requiere no ser único en el modelo.
- c) El nombre para una relación es usualmente expresado en la dirección padre-hijo tal como una sentencia puede ser formada por la combinación del nombre de la entidad padre, el nombre de la relación (verbo), la cardinalidad y el nombre de la entidad hijo.
- d) Si una relación no especifica no tiene noción de los roles del padre o del hijo, la orden del verbo debe depender de la posición relativa de las entidades, como se muestra a continuación:
 - 1) El primero debe expresar la relación desde la entidad de la izquierda a la entidad de la derecha (si las entidades se encuentran en forma horizontal) o de la entidad superior a la entidad que se encuentra en la parte inferior (si se encuentran en forma vertical).
 - 2) La segunda porción del nombre la relación debe expresar la relación desde la otra dirección, que es de la entidad de la derecha a la entidad de la izquierda o de la entidad inferior a la entidad superior, nuevamente dependiendo de la orientación.
 - 3) La orientación de arriba abajo debe tomar precedente de la orientación de izquierda a derecha, y si las entidades se forman de derecha superior hacia abajo a la izquierda, el primer verbo describe la relación de la perspectiva de la primera entidad.
- e) Para una relación no especifica, la relación debe ser nombrada de tal manera que las sentencias puedan ser formadas por la combinación de los nombres de las entidades con las frases.
Por ejemplo, las sentencias “*un proyecto tiene cero, uno o más empleados*” y “*Un empleado tiene asignado cero, uno o muchos proyectos*” pueden ser derivados desde una relación no especifica nombrada de la siguiente manera: “tiene” y “es asignado” entre las entidades *proyecto* y *empleado*. (La secuencia asume la entidad *proyecto* que aparece sobre o a la izquierda de la entidad *empleado*).

5.3.4 Reglas de las Relaciones

5.3.4.1 Composición

- a) Una relación siempre se establece entre dos entidades.
- b) Una entidad debe ser asociada con una o muchas entidades, como hijo o como padre.
- c) Una instancia de una entidad padre debe ser asociada con cero, una ó más instancias de la entidad hijo dependiendo de la cardinalidad especificada en el modelo.
- d) En una relación no específica, una instancia de una de las entidades debe ser asociada con cero, una ó más instancias de la otra entidad, dependiendo de la cardinalidad especificada.

5.3.4.2 Relación Identificadora/relación No Identificadora

- a) Una relación debe ser clasificada como una de las siguientes:
 - 1) Una relación Identificadora, ó
 - 2) Una relación mandatoria No-Identificadora(total) ó
 - 3) Una relación opcional No-Identificadora(parcial)
- b) Solo la relación no Identificadora y una relación no específica pueden ser recursivas; es decir pueden relacionar una instancia de una entidad a otra instancia de la misma entidad.
- c) Una relación Identificadora no debe ser recursiva.

5.3.4.3 Relación Total/Relación Parcial

- a) En una relación Identificadora, una instancia de la entidad hijo debe ser asociada con exactamente una instancia de la entidad padre.
- b) En una relación Mandatoria no Identificadora (total), una instancia de la entidad hijo debe ser asociada con exactamente una instancia de la entidad padre.
- c) Solo una relación No-Identificadora puede ser parcial, es decir opcional desde la perspectiva del hijo.
- d) En una relación parcial, una instancia de la entidad hijo debe ser asociada con cero o una instancia de la entidad padre.

5.3.4.4 Entidad Independiente/ Entidad dependiente

- a) Si una clave foránea es usada totalmente en la clave primaria de una entidad, luego la entidad debe ser clasificada como dependiente.
- b) Si solo una porción de una clave foránea o atributo de una no clave foránea es usada para una clave primaria de una entidad, luego la entidad debe ser clasificada como independiente.
- c) La entidad hijo en una relación Identificadora debe ser siempre una entidad dependiente.
- d) La entidad hijo en una relación no Identificadora debe ser una entidad independiente a menos que sea una entidad hijo en alguna relación Identificadora.

- e) La entidad padre en una relación Identificadora debe ser una entidad independiente a menos que esta sea también la entidad hijo en alguna otra relación Identificadora.
- f) Una entidad de categoría no debe ser una entidad hijo en una relación Identificadora a menos que la clave primaria, que es parte de la relación sea completamente contenida en la clave primaria de la entidad de categoría.

6. MODELAMIENTO FISICO DE DATOS

El Modelo Físico de Datos se realiza a partir del modelo conceptual y/o lógico de datos normalizado o del modelo de clases en el caso de diseño orientado a objetos.

La implementación del modelo se realiza a través de un motor de base de datos específico.

6.1 CRITERIOS GENERALES

1. En el Modelamiento Físico de Datos, encontramos los objetos esquema de la base de datos y los objetos no esquema de la base de Datos.
2. El nombre de los objetos esquema y no esquema de la base de datos se rigen por los estándares indicados en su correspondiente clasificación.
3. Los nombres de los objetos no deben contener caracteres especiales como : “,’/,#,),(,%,&,\$,=,?,¿,¡,|,.,;.
4. Los nombres de los objetos serán creados en mayúsculas para el caso del manejador de base de datos Oracle y en minúscula para el caso de la base de datos en Informix.
5. No se pueden utilizar como nombres, palabras reservadas de los manejadores de base de datos. (comandos de los manejadores de base de datos).
6. En base al modelo de datos se genera el script de creación y/o modificación de los objetos de la base de datos a ser ejecutados en los ambientes de desarrollo, pruebas y de producción.
7. Cada aplicación ó sistema dispone de un(os) tablespace(s) ó dbspace(s) para los objetos que ella contemple, a fin de lograr una mayor performance del motor de la base de datos.
8. Considerar la creación de índices en un tablespace ó dbspace creado para el almacenamiento de estos objetos de base de datos.
9. El nombre de los objetos esquema y no esquema de la base de datos se rigen por los estándares indicados en su correspondiente clasificación.
10. No deben existir dos tablas con el mismo nombre en el Modelo de Datos Institucional.
11. Las tablas deben tener clave primaria obligatoria, a excepción de las tablas log ó de auditoria u otros casos que según evaluación se consideren como excepción.
12. En casos en que el manejador de base de datos tenga como objeto de base de datos una clave primaria esta podría presentarse de dos formas:
 - a) Que la clave primaria este conformada por el constraint y el índice único en un solo objeto de base de datos. Se aplica cuando el manejador de base de datos es Oracle. Por ejemplo: el único objeto será pk_t7091 clave primaria de la tabla t7091contenv.
 - b) Que la clave primaria solo este conformada por el constraint. En estos casos el analista deberá indicar el constraint de la clave primaria y el índice único en el modelo de datos como dos objetos diferentes. Se aplica cuando el manejador de base de datos es Informix. Ejemplo: Se creará primero el unique index indicando el dbspace de índices: in01t7090 y luego se creará el constraint pk_t7090 objetos de la tabla t7090enlace.
13. Las tablas deben tener como máximo 4 índices por cada tabla para mantener mejores tiempos de consulta de los aplicativos.

14. Mantener actualizado el diccionario de datos de los modelos de datos. Indicando las descripciones de cada atributo o columna y al indicar el nombre de la tabla.
15. Indicar en el diccionario de datos el rango de valores que pueda contener la columna ó el código de parámetro referenciado.
16. Evitar tener campos nulos, se indicará en los modelos el valor por defecto. (Existen excepciones como las tablas de carga del datawarehouse).
17. Los defaults de una columna será la misma en las diferentes tablas del Modelo de Datos.
18. Modelos normalizados hasta la tercera forma normal para los Modelos de los Sistemas Transaccionales, y la desnormalización en el caso de los Modelos de Datawarehouse (OLAP).
19. Los índices deben ser optimizados antes de pasar a producción.
20. No se permite realizar delete sobre las tablas, por ello se recomienda utilizar campos de eliminación lógica, y un número correlativo de la tabla.
21. Al ser eliminado un objeto su numeración podrá ser asignada a un nuevo objeto.

6.2 CRITERIOS ESPECIFICOS

1. El nombre de la tabla y de las columnas no deberá tener caracteres especiales: “,’,/,#,),(,%,&,\$,=,?,¿,¡,|,.,,:. En forma excepcional, se podrá utilizar uno ó varios underline dentro del nombre de la tabla pero que estén separados cada 3 caracteres como mínimo.
2. El nombre de las columnas utilizará un underline obligatorio (_) para dividir el prefijo de la columna según estándar con el resto del nombre del atributo.
3. En forma excepcional, se podrá utilizar uno ó varios underline dentro del nombre de la columna pero que estén separados cada 3 caracteres como mínimo. Para el caso de los sistemas aduaneros, el ambiente de la base de datos contiene un objeto no esquema que es el usuario propietario, el cual para cada aplicativo, será el 'dueño o propietario' de todos los objetos creados en dicho tablespaces, y el único autorizado a alterar la estructura de las tablas y demás objetos. No aplica a los sistemas tributarios, pues se utiliza solo el usuario informix.
4. El valor de las columnas de la clave primaria no deben ser modificadas en el tiempo.
5. Para los sistemas Aduaneros, se crearán sinónimos públicos. Los sinónimos privados no serán creados salvo por excepción y debidamente fundamentado.
6. Las columnas que tengan el mismo significado deberán tener el mismo nombre en los diferentes modelos de datos que utilicen dichos campos.
7. La longitud de las columnas que tengan el mismo nombre deben tener la misma longitud en los modelos de datos.
8. Los valores que son considerados para las validaciones de los checks, deberán estar contenidos en las tablas de parámetros de tributos internos y en las tablas generales de los sistemas aduaneros solo en el caso que el manejador de base de datos lo permita. Caso contrario indicara los valores en la base de datos a través de las validaciones de los checks.
9. Las claves foráneas ó foreign keys deben ser creados en la base de datos para mantener la integridad referencial, a excepción de aquellos no considerados por el analista de sistemas previo análisis, ni los modelos de datawarehouse en los cuales se aplica la desnormalización.
10. En los procedimientos no considerar variables que no se utilizarán en el programa.

11. Se indican nombres de columnas que serán utilizadas en los diferentes modelos de datos y que tengan el mismo significado:

cod_personal	: código de personal ó número de registro.
fec_modif.	: fecha de modificación.
cod_usumodif	: código de usuario de modificación
fec_inivig	: Fecha de inicio de vigencia.
fec_finvig	: Fecha Final de vigencia.
fec_inicio	: Fecha de inicio
fec_fin	: Fecha final
cod_aduana	: Código de Aduana
cod_depen	: Código de Dependencia
num_ruc	: Número de Ruc
ind_del	: Flag de eliminación lógica

12. No se crearán claves primarias que agrupan casi todas las columnas de la tabla. Estos casos serán evaluados.
13. Si se crea una tabla que tiene integridad referencial con otra tabla de estructura antigua, la creación de los campos de la nueva tabla tendrá que ser de acuerdo al estándar definido en el presente manual. La integridad de la información se realizará a través de otros constraints ó a través de las validaciones de los programas.
14. Las tablas a crearse deberán tener obligatoriamente los campos **cod_usumodif** (código de usuario de modificación) y **fec_modif.** (fecha de modificación), los cuales no deberán contener valores nulos o en blanco. Las excepciones serán sustentadas y evaluadas.
15. Deberá indicarse una breve descripción de lo que se realiza en los procedimientos, paquetes y funciones en forma comentada en la cabecera de estos objetos.
16. Las tablas históricas no deben tener valores por defecto en las columnas, a excepción de las columnas que indiquen fecha de modificación y otras columnas de control de cambios.
17. En los casos de las columnas que son parte de las dependencias no se consideran valores por defecto, ya que tienen valores definidos por la dependencia.
18. Los valores por defecto para el caso del **date(oracle)** y **datetime(informix)**:
- En Oracle: **default to_date('01/01/0001','DD/MM/YYYY')**
 default to_date('31/12/9999','DD/MM/YYYY') - Para fecha final.
 default sysdate
- En Informix: **default datetime (0001-01-01 00:00:00) year to second**
 default datetime (9999-12-31 00:00:00) year to second
 default current year to second
- Se exceptúa su uso sólo en los sistemas administrativos en 4gl (Mantenimiento).

6.3 ONLINE Y BASES DE DATOS

6.3.1 Nomenclatura Online, Instancia ó Data Server

Formato:	<sigla del equipo>_<nombre del onconfig>
Longitud Máxima:	Total : 15 posiciones <sigla del equipo> : máx. tamaño 4 <nombre del onconfig> : máx. tamaño 10
Descripción:	El online se aplica para el manejador de base de datos Informix
Ejemplo:	s0_bancos s1b_prico scc1_coa s4_bancos s4_prico

6.3.2 Nomenclatura de Base de Datos

Formato:	bd<nombre de la base de datos>
Longitud máxima:	8 posiciones. No deben tener caracteres especiales.
Descripción:	El nombre asignado deberá explicar la información que se almacena.
Ejemplo:	bdfisca: Base de datos con la información de fiscalización.

6.4 OBJETOS DE LA BASE DE DATOS

6.4.1 Objetos Esquema de la Base de Datos

Un esquema es una colección de estructuras lógicas de la data u **objetos esquema**. Este conjunto de objetos pertenece a un usuario propietario. Los objetos esquema son creados y manipulados por el SQL e incluye los siguiente tipos de objetos:

6.4.1.1 Tabla - Nomenclatura

Formato:	<z>xxxx<nombre tabla>
Longitud Máxima:	18 posiciones
Descripción:	z : Tipo de Tabla t : Tabla Fija(También considera tabla temporal fija) f : Fact ó Tabla de Hechos d : Dimensión l : Log p : Tabla de Procesamiento xxxx : Número Correlativo de Tabla <nombre tabla> : En forma excepcional, se podrá utilizar uno ó varios underline dentro del nombre de la tabla pero que estén separados cada 3 caracteres como mínimo No se utilizará underline después del número correlativo de Tabla. No se aceptan caracteres especiales, indicados en el numeral 6.2 punto 1.
Ejemplo:	t1492correl : Tabla de correlativos

6.4.1.2 Tabla Temporal - Nomenclatura

Formato:	tmp<nombre de tabla>
Longitud Máxima:	18 posiciones
Descripción:	tmp : No se asigna numeración , son tablas temporales que son creadas y eliminadas por el programa. Se asignará el dbspace o tablespace temporal de la bd.
Ejemplo:	tmpcalculo : Tabla temporal de cálculo

6.4.1.3 Columnas – Nomenclatura

Formato:	xxx <nombre de columna>
Longitud Máxima:	18 posiciones
Descripción:	xxx : Indicador de tipo de columna <nombre de columna> En forma excepcional, se podrá utilizar uno ó varios underline dentro del nombre de la columna pero que estén separados cada 3 caracteres como mínimo

Indicador de Tipo de Columna (Prefijo)	Tipo de Datos Manejador de b.d. Informix	Tipo de Datos Manejador de b.d. Oracle
Ann_ : año	Smallint	Number
Cnt_ : cantidad	Smallint, Decimal,Integer	Number
Clv_ : clave para algoritmo de búsqueda	Char	Char
Cod_ : código	Char	Char
Des_ : descripción	Varchar, Text	Varchar2, Long
Obs_ : observación	Varchar, Text	Varchar2, Long
Nom_ : nombre de persona, razón social, nombre de archivo	Varchar	Varchar2
Fec_ : fecha	Date, Datetime	Date
hor_ : hora	Datetime (formato hora)	Date (formato hora)
Ind_ : indicador	Char(1)	Char(1)
Mto_ : monto	Decimal, money	Number
Mes_ : mes	Char(2)	Char(2)
Num_ : número	Smallint,Integer,Char,decimal	Number, Char
Per_ : periodo (año, mes)	Char(6)	Char(6)
Por_ : porcentaje	Decimal	Number
Sem_ : semana	Char(1)	Char(1)
Exp_ : expresión (fórmulas y concatenación de códigos)	Varchar	Varchar2
Dir_ : dirección de calles, ip, url, postal	Varchar	Varchar2
img_ : Imagen	Byte	Blob
Arc_ : Archivo	Text, byte	Long, Blob
msj_ : mensaje	Varchar	varchar2
Aud_ : audio	Byte	blob
val_ : valor	Varchar,integer,smallint,decimal	varchar2 ó number
Vid_ : video	Byte	Blob

Ejemplo :

cnt_bultos	Number(15,3)
nom_decide	Varchar2(50)
fec_inivig	Datetime ó Date

Excepciones :

- Las columnas de tipo código que tienen datos variables deberán ser definidos como varchar (en informix) ó varchar2 (en oracle).
- La columna cod_ de tipo código será considerado varchar (informix) ó varchar2 (oracle) en los casos de migración de datos. Sin embargo, este tipo de datos será otorgado provisionalmente hasta que la data se actualice a nivel de toda la tabla, cambiando al final el tipo de datos a char.
- La columna cod_ también será considerado como smaliint para uso exclusivo de datawarehouse, puesto que este tipo de datos faciita el procesamiento de grandes cantidades de registros.
- La columna de tipo número: num podrá ser de tipo varchar (en informix) ó varchar2 (en oracle) solo en los casos que existan dependencias con dichas columnas(creados antes de la aplicación del estándar vigente), indicando en el requerimiento de base de datos el nombre de la tabla y columna con la cual existe la dependencia.
- El campo año con prefijo: ann_ también podrá ser utilizado con el tipo de dato char y varchar (en informix) ó varchar2(en oracle) en los casos que exista dependencia con tablas creadas antes de la aplicación del estándar vigente.
- El prefijo exp, será utilizado para la concatenación de códigos y en el caso que se fundamente su almacenamiento también será utilizado para entidades externas.
- Se utilizará el prefijo ape_ : apellido para casos específicos. Por ejemplo cuando en una misma tabla indicamos diferentes nombres para diferentes agentes del modelo.
- Las columnas con prefijo indicador tendrán constraints como son los checks de validación.
- Los tipos de estados son considerados como códigos, por ejemplo se tendrá tipo de documento como cod_tipdoc.
- Para los tipos de columna con prefijo ann, se podrá utilizar el tipo de dato char en el caso de mantenimiento de Sistemas desarrollados en lenguajes diferente al java..
- En el caso de tablas de auditoría ó tablas log se consideran campos que registran valores y por el cual se considerará el prefijo val para estos casos.

6.4.1.4 Índices – Nomenclatura**Índice Único de la Clave primaria (manejador de b.d. Informix)**

Formato:	Inpkzxxxx
Longitud Máxima:	10 posiciones
Descripción:	in : prefijo del Índice pk : de la clave primaria yy : Nro correlativo de índice por tabla zxxxx : Prefijo y Nro de tabla
Ejemplo:	inpkt1492 : Índice único de clave primaria – tabla t1492

Índices

Formato:	inyyzxxxx_<nombre de 1era columna que conforma el índice>
Longitud Máxima:	9 posiciones
Descripción:	in : prefijo del Índice yy : Nro correlativo de índice por tabla (01,02,etc) xxxxx : Prefijo y Nro de tabla _<nombre de 1era columna que conforma el índice> : opcional
Ejemplo:	in01t1492 : Índice número 1 de la tabla t1492 in02f2456 : Índice número 2 de la tabla f2456

6.4.1.5 Clave Primaria (Primary Key)

6.4.1.5.1 Nomenclatura

Formato:	pk_zxxxx
Longitud Máxima:	8 posiciones
Descripción:	pk : prefijo de la clave primaria xxxxx : Prefijo y Nro de tabla
Ejemplo:	pk_t1492

6.4.1.5.2 Sintaxis

- a) En el caso de creación de nuevas tablas, las columnas que componen la clave primaria de una tabla debe estar posicionada en la parte superior de la lista de columnas de la tabla.

Excepción : Para el caso de modificación de la clave primaria de la tabla, se acepta que las columnas sean agregadas al final de la tabla.

6.4.1.5.3 Reglas

Composición :

- Toda tabla debe tener una clave primaria a excepción de las mencionadas en el numeral 6.1.
- Además de una clave primaria, una tabla puede tener una o más claves alternas específicas.
- Una clave (primaria o alterna) consiste de una columna ó de una combinación de columnas.
- Cada registro de la tabla debe tener un valor para cada columna de la clave primaria.
- Una clave (primaria o alterna) debe contener solo las columnas que contribuyen al identificador único de la entidad.
- Si la clave primaria se compone de uno ó mas columnas, el valor de cada columna no clave será dependiente de la clave primaria.
- Las columnas que conforman la clave primaria y las claves alternas de una tabla deben ser propietarias de la entidad o derivarse a través de la relación.

- h) Cada columna que no es parte de la clave (primaria o alterna) debe ser funcionalmente dependiente de la clave primaria y la clave alterna..
- i) Se deberá evaluar el uso de campos varchar como parte de una clave primaria ó de una clave alterna en base de datos Informix.

Clave Alterna :

- a) Una columna especificada como parte de una clave alterna no necesariamente puede tener un valor.

Clave Primaria en la Generalización (OOM) :

- a) Una clase debe tener la misma clave primaria que su entidad genérica.

6.4.1.6 Clave Foránea (Foreign Key)

6.4.1.6.1 Nomenclatura

Formato:	fknn_ <zxxxx ó Nro de tabla dependiente>_<zyyyy
Longitud Máxima:	16 posiciones
Descripción:	fk : prefijo del Foreign Key nn : Nro correlativo de Foreign key por tabla (si es el caso) zxxxx : Prefijo y Nro de tabla dependiente zyyyy : Prefijo y Nro de tabla que contiene la clave Primaria
Ejemplo:	fk01_t1432_t1628

6.4.1.6.2 Reglas

Relaciones :

Una columna de clave foránea puede ser parcialmente ó completamente la clave primaria, ó una clave alterna, ó como una columna no clave de una tabla. Si todas las columnas de la clave primaria de una tabla padre son migrados como parte de la clave primaria de la tabla hijo, entonces la relación con las columnas migrados es una relación Identificadora. Si alguna de las columnas no migrados no son parte de la clave primaria de la entidad hijo, luego la relación es una relación no Identificadora.

Por ejemplo si las tareas son únicamente numeradas en un proyecto, luego la columna migrada id_proyecto podrá ser combinado con un atributo propio id_tarea, para especificar la clave primaria de tarea. La tabla proyecto tendrá una relación Identificadora con la tabla tarea. De otro lado, la columna id_tarea será siempre única, aún a través de los proyectos, luego la columna migrada id_proyecto deberá ser una columna no clave de la tabla tarea. En este caso, la tabla proyecto podrá tener una relación no Identificadora con la tabla tarea.

Relaciones Múltiples

En algunos casos, una tabla hijo puede tener múltiples relaciones con la tabla padre. La clave primaria de la tabla padre aparecerá como una columna de la clave foránea en la tabla hijo para cada relación. Para un registro de la tabla hijo, los valores de las columnas migradas debe ser diferente para cada relación, dos diferentes registros de la tabla padre deben ser referenciadas.

Sintaxis

- Una clave foránea debe ser representado por la posición de los nombres de cada columna de la clave foránea en el rectángulo como se encuentra representada la tabla.
- Cada nombre de la columna de la clave foránea debe consistir del nombre de la columna seguido con las letras FK en paréntesis.
- Si cualquier columna de la clave foránea no pertenece a la clave primaria de la tabla hijo, la columna debe estar posicionado debajo de las columnas de la clave primaria, y la tabla debe ser clasificado como un identificador independiente con respecto a la relación.

Reglas

- Cada columna de la clave primaria de una tabla padre en una relación debe ser una columna de la clave foránea (migrada) en la relación con la tabla hijo.
- Cada columna de la clave primaria de una entidad genérica en una estructura de generalización debe ser un atributo de la clave foránea (heredado) en la relación con la entidad de categoría.
- Cada columna de la clave primaria de una entidad genérica en una generalización debe ser parte de la clave primaria de la entidad de categoría.

6.4.1.7 Check - Nomenclatura

Formato:	cknn_zxxxx
Longitud Máxima:	10 posiciones
Descripción:	ck : prefijo de los constraints tipo Check nn : Nro correlativo de Check por tabla (si es el caso) xxxxx : Prefijo y Número de tabla
Ejemplo:	ck01_t1432

6.4.1.8 Vista - Nomenclatura

Formato:	vw<prefijo y número de tabla ó nombre de vista>
Longitud Máxima:	18 posiciones
Descripción:	vw : Prefijo de Vistas
Ejemplo:	vwt1498 vwpersonalproy

6.4.1.9 Sinónimo - Nomenclatura

Los sinónimos son a menudo utilizados por seguridad y conveniencia, puesto que pueden hacer lo siguiente:

- Enmascarar el nombre y propietario de un objeto.
- Proveer un nombre y total transparencia de los objetos remotos utilizados en base de datos distribuidas.
- Simplifica el uso de las sentencias sql a los usuarios de base de datos.

Los sinónimos son muy útiles en bases de datos distribuidas y no distribuidas porque permite esconder la identidad de los objetos, incluyendo su locación en el sistema. Esto es ventajoso porque si el objeto es renombrado ó movido, únicamente el sinónimo necesitaría ser redefinido.

Por ello, las aplicaciones basadas en sinónimos continúan funcionando sin modificarse, puesto que son únicamente los sinónimos los que varían.

El sinónimo puede ser igual al nombre de la tabla ó corresponder a la parte nemotécnica de la tabla (no se considera zxxxx) ó ser igual al nombre de la vista, en este último casose elimina el prefijo VW.

Ejemplo: t1498documento, personalproy, docum (tabla: txxxxdocum)

Excepción: En el caso que algunas tablas sean cambiadas de propietarios ó sistemas, se mantiene el sinónimo creado originalmente ó inicialmente.

6.4.1.10 Trigger - Nomenclatura

Formato:	trnn<acción>_ZXXXX
Longitud Máxima:	13 posiciones
Descripción:	tr : Prefijo del objeto Trigger nn : Nro Correlativo por tabla acción : i Insert u Update d Delete Máxima longitud de Acción : 3 posiciones zxxxx : Tipo de tabla y Nro correlativo de tabla
Ejemplo:	tr01iu_t1452 : Trigger de insert y update en la tabla T1452 tr02i_t2112 : Trigger de insert en la tabla T2112

6.4.1.11 Procedimiento Almacenado – Nomenclatura (Stored Procedure)

Formato:	sp<Nombre procedimiento> No se aceptan caracteres especiales, indicados en el numeral 6.1 punto 3.
Longitud Máxima:	18 posiciones
Descripción:	sp : Prefijo de Stored Procedures
Ejemplo:	spcalculodiario : Stored Procedure que realiza el cálculo diario.

6.4.1.12 Función - Nomenclatura

Formato:	fn<nombre de la función> No se aceptan caracteres especiales, indicados en el numeral 6.1 punto 3.
Longitud Máxima:	18 posiciones
Descripción:	fn : Función
Ejemplo:	fndiasutiles : Función de calculo de día útil

6.4.1.13 Package - Nomenclatura

Formato:	pa<nombre del paquete> No se aceptan caracteres especiales, indicados en el numeral 6.1 punto 3.
Longitud Máxima:	18 posiciones
Descripción:	pa :Paquete
Ejemplo:	Pavalida

6.4.1.14 Secuencia - Nomenclatura

Formato:	se<nombre de la secuencia> No se aceptan caracteres especiales, indicados en el numeral 6.1 punto 3.
Longitud Máxima:	18 posiciones
Descripción:	se :Secuencia
Ejemplo:	Sedinumero

6.4.1.15 Resumen de Prefijos de Objetos Esquema

Tipo de Objeto	Prefijo
Tabla	t : Tabla Fija y Temporal fija f : Fact ó Tabla de Hechos d : Dimensión l : Log p :Tabla de procesamiento
Tabla Temporal	tmp
Indice	in
Clave Primaria (Primay Key)	pk
Clave Foránea (Foreign Key)	fk
Check	ck
Vista	vw
Trigger	tr
Procedimiento Almacenado (Stored Procedure)	sp
Función	fn
Package	pa
Secuencia	se
Usuario Propietario	us
Rol	rl

6.4.2 Objetos no Esquema de la Base de Datos

Son los demás tipos de Objetos que también son almacenados en la base de datos pero no están contenidos en el esquema.

6.4.2.1 Dbspace ó Tablespace - Nomenclatura

Formato:	ts<nombre de onconfig ó código del sistema>_xxx
Longitud Máxima:	Total : 17
Descripción:	Donde: <nombre de onconfig ó código del Sistema> tam. máx. 10 z: Indica que tipo de datos se guardará en dbspace i : Indice ó d: Data xx: Número Correlativo (opcional)
Ejemplo:	tsprico_d01 tsfi_d

Excepción : Para los casos en que todo un tablespace ó dbspace es asignada a una tabla, el nombre de la tabla será usado en lugar del nombre del onconfig de la nomenclatura de la tabla.

6.4.2.2 Chunks ó Datafiles - Nomenclatura

Esta sección del manual se encuentra en proceso de definición por la División de Soporte Técnico.

6.4.2.3 Usuarios Propietarios - Nomenclatura

Se aplican estos usuarios propietarios en las bases de datos con desarrollo de sistemas Aduaneros en manejador de base de datos Oracle.

Formato:	usamss00
Longitud Máxima:	8 posiciones
Descripción:	ss : Indica el Subsistema.
Ejemplo:	usamdi00 Usuario propietario de importaciones

6.4.2.4 Roles - Nomenclatura

Los Roles se aplican al desarrollo de los sistemas en el manejador de base de datos Oracle:

Formato:	rlss
Longitud Máxima:	4 posiciones
Descripción:	rl : Prefijo de Rol ss : Subsistema
Ejemplo:	rldi Rol de Importaciones

6.4.2.5 Profile - Nomenclatura

Formato:	pr<nombre del profile> No se aceptan caracteres especiales, indicados en el numeral 6.1 punto 3.
Longitud Máxima:	18 posiciones
Descripción:	pr : Profile
Ejemplo:	prrecauda : Profile del área de recaudación

7. TERMINOLOGÍA

Atributo : Representa las características ó propiedades que sirven para calificar, identificar, clasificar, cuantificar o expresar el estado de una entidad.

Atributo Propio : Son atributos que nacen con la entidad y que no son derivados o heredados de otra entidad.

Atributo Derivado: En adición a un atributo propio de la entidad, un atributo debe ser presentado en una entidad para su derivación a través de una relación o a través de una generalización. Por ejemplo, si cada empleado es asignado a un departamento, luego el atributo *cod_dpto* debe ser un atributo de la tabla *empleado*, el cual ha migrado de la entidad *departamento* a la entidad *empleado* a través de la relación entre ellas.

La entidad *departamento* debe ser la propietaria del atributo *cod_dpto*. Y solo los atributos de la clave primaria debe ser derivados a través de la relación. El atributo *nombre_dpto*, por ejemplo no es un atributo derivado de *empleado*, en tanto no es parte de la clave primaria de la entidad *departamento*.

Atributos – Herencia : En un modelo entidad relación (ER), cada atributo es propio de una entidad, pero un atributo puede ser heredado a otra entidad por herencia. Por ejemplo: el atributo código del empleado de la tabla empleado (entidad genérica) es heredado por la entidad salario_empleado (entidad que hereda).

Base de Datos : Es una colección de información contenida en tablas relacionadas entre sí.

Clave Alterna : Cuando exista mas de una clave candidata, una es designada como la clave primaria y otra clave candidata es designada como una clave alterna. Si existe solo una clave candidata, entonces esta será la clave primaria.

Clave Candidata : Una clave candidata de una entidad es un atributo ó grupo de atributos que podrían ser elegidos como clave primaria.

Una clave candidata para ser aceptada como clave primaria debe identificar unívocamente cada instancia de la entidad, no puede ser nula o tener alguna parte nula y todos los atributos no-clave deben depender completamente de ella.

Una clave candidata no debe ser escogida como parte de la clave primaria.

Clave Foránea ó Foreign Key : Es la columna o grupo de columnas en una tabla llamada hija que contiene valores que concuerdan con la clave primaria de otra tabla llamada padre.

Claves foráneas en la Generalización : La clave primaria para cada entidad de categoría hereda de la clave primaria de la entidad genérica. Por ejemplo, si *salario-empleado* es una entidad de categoría de la entidad genérica *empleado* y el atributo *id_empleado* es la clave primaria de la entidad *empleado*, el atributo *id_empleado* podrá ser la clave primaria de *salario-empleado*.

Clave Primaria o Primary Key : Es una columna o grupo de columnas que han sido elegidos como un identificador único de una tabla. Los valores de estas columnas son diferentes en cada fila. Es decir cada fila es única. Representa una restricción única de los valores de las propiedades de la entidad.

Check : es un constraint de integridad de una columna o set de columnas que requieren que una condición específica sea verdadera o falsa para cada fila de la tabla.

Chunks : Una colección contigua de espacio en disco asignada a un dbspace de una instancia.

Datafiles : Los datafiles son los ficheros físicos en los que se almacenan los objetos que forman parte de un tablespace. Un datafile pertenece solamente a un tablespace y a una instancia de base de datos. Un tablespace puede estar formado por uno o varios datafiles. Cuando se crea un datafile, se debe indicar su nombre, su ubicación ó directorio, el tamaño que va a tener y el tablespace al que va a pertenecer.

Dbspace : Una colección lógica de chunks que forman un pool de espacio en disco, la cual es usada para almacenar objetos de las bases de datos.

Entidad : Es la representación de un conjunto de cosas reales y abstractas como personas, objetos, lugares, eventos, ideas, combinación de cosas, etc, que son reconocidas como del mismo tipo y que tienen información relevante en el modelo de datos. Tienen atributos y características comunes y pueden relacionarse entre ellas.

Entidad de Categoría : Pertenecer a una colección de entidades caracterizadas por satisfacer un cierto predicado común.

Entidad Independiente : Una Entidad es Independiente si cada instancia de la entidad puede ser únicamente identificada sin determinar su relación con otra entidad.

Entidad Dependiente : Una entidad es dependiente, si la única identificación de una instancia de la entidad depende de su relación con otra entidad.

Expresada en términos de la clave foránea, una entidad será dependiente si cada clave foránea es completamente contenida en la clave primaria de otra entidad. De otro lado sería independiente.

Las entidades categorías son siempre dependientes-identificadas.

Función : Son sentencias ó comandos en SQL que pueden complementar el manejo de los datos en las consultas. Se utilizan dentro de las expresiones y actúan con los valores de las columnas, variables o constantes

Identificador Unico : Una entidad debe tener un atributo (o combinación de atributos) cuyos valores únicos identifican cada instancia de la entidad.

Índices : Datos estructurados que están asociadas a una o más columnas en una tabla, en que los valores de la columna se ordenan para mejorar el rendimiento de algunas consultas

Instancia : Una instancia de una entidad con sus correspondientes valores de atributos representa un objeto concreto del mundo real. Por lo tanto, podemos decir que una entidad describe un conjunto de objetos del mundo real llamados instancias.

No pueden existir dos entidades con los mismos atributos.

Online, instancia ó Data Server : Es el programa que administra el contenido de la base de datos y su almacenamiento en los discos, es decir, como las tablas, filas y columnas están almacenadas físicamente en el computador. Asimismo, el online también interpreta y ejecuta comandos SQL.

Un online esta conformado por los Server procesos, la shared memory y el espacio en disco.

Package : Este objeto se usa en Oracle para almacenar código PL/SQL en la Base de Datos, el cual agrupa procedimientos almacenados y funciones

Procedimiento Almacenado (Stored Procedure) : Es una rutina definida por el usuario que es escrita en SPL (Storage Procedure Lenguaje).

Es una extensión del SQL y son almacenados en el system, catálogo en formato ejecutable.

Una SPL puede ejecutar rutinas escritas en C u otros lenguajes externos.

Profiles : Se refiere a las limitaciones de recursos de la base de datos asignadas a los usuarios a través de estos perfiles, los cuales previenen el consumo excesivo de recursos del sistema de base de datos.

Relaciones : Las relaciones son usadas para representar asociaciones entre entidades.

Relación específica : Es una asociación entre dos entidades en la cual cada instancia de una entidad (referida a la entidad padre) es asociada con cero, uno o más instancias de la segunda entidad (referida a la entidad hijo). Por lo tanto, cada instancia de la entidad hijo es asociado con cero o una instancia de la entidad padre.

Las relaciones padre-hijo deben ser consideradas como *Relaciones específicas* porque ellas especifican precisamente como las instancias de una entidad se relaciona con las instancias de una segunda entidad.

Relación no específica : Debe ser usada para representar una asociación de muchos a muchos entre dos entidades. Una relación no específica (relaciones de muchos a muchos) es una asociación entre dos entidades en la cual cada instancia de la primera entidad es asociada con cero, uno o muchas instancias de la segunda entidad, y cada instancia de la segunda entidad esta asociada con cero, una o más instancias de la primera entidad.

En el desarrollo inicial de un modelo, es a menudo útil identificar relaciones no específicas entre entidades. Esta relación no específica debe ser después reemplazado

con relaciones específicas en el desarrollo del modelo al introducir una tercera entidad tal como se generan en los Modelos Físicos. Una entidad introducida para resolver una relación no específica es algunas veces llamada una entidad asociada.

Roles : Son grupos de privilegios que se otorga a los usuarios o a otros roles. Los roles facilita la administración de los usuarios de la base de datos y también de los privilegios de los objetos esquema.

Secuencia : Objeto de datos del manejador de base de datos Oracle que genera automáticamente números únicos. Normalmente se utiliza para crear un valor de clave primaria, Sustituye al código de aplicación y acelera la eficacia del acceso a los valores de secuencia al almacenarse en memoria caché.

Sinónimo : Es un alias de una tabla, vista, secuencia, procedimiento, función ó package. La utilidad de los sinónimos es la posibilidad de independizar las aplicaciones de los nombres físicos de las tablas que se manejan.

Tabla : Unidad básica de almacenamiento; Esta formada por filas (registros) y columnas (campos), identificadas por una o más columnas.

Tablespace : Es una colección lógica de extents que contiene una tabla específica, un índice o un fragmento.

Trigger : Es un objeto que reside en la base de datos y se encarga de especificar cuando una particular acción (INSERT, DELETE, UPDATE, EXECUTE PROCEDURE or EXECUTE FUNCTION) ocurre sobre una tabla particular. También se les denomina reglas activas por que básicamente, un trigger consiste en un evento trigger (conjunto de condiciones) y la resultante es una acción de trigger.

Usuarios Propietarios : Son los dueños de los objetos que se crean en una determinada instancia. Estos usuarios otorgan los privilegios de los objetos creados por él.

Vista : Es la representación de los datos de una o más tablas utilizando el lenguaje estructurado de consulta (SQL).

ANEXO

ANEXO I: ROLES Y USUARIOS INTRANET, INTERNET, EXTRANET**A. ROLES DE AGRUPACIONES DE SISTEMAS WEB ADUANEROS**

Los Roles se aplican al desarrollo de Sistemas por Agrupaciones de Sistemas Web en el manejador de base de datos Oracle. (Roles que contienen roles de los sistemas)

Formato:	RLAGRU
Longitud Máxima:	6 posiciones
Descripción:	RL : Prefijo de Rol AGRU : Agrupación de Subsistemas web
Ejemplo:	RLPRCT Rol de Prevención del Contrabando contiene RLPC, RLIA

DEFINICIÓN DE ROLES DE AGRUPACIONES DE SISTEMAS WEB ADUANEROS

Roles de Agrupaciones	Sistemas	Roles de los Sistemas
RLPRCT - Prevención del Contrabando	Cta. Cte. De Importación Zofratatna	RLPC,RLIA,RLDI
RLTEAD - Técnica Aduanera	Control de Contenedores y Carga Suelta Mercancía Prohibida y Restringida Módulo de Firmas Despacho Urgente o Anticipado Registro y Consulta de Guías Postales Rayos X Firmas Autorizadas para mercancías restringidas Notas Explicativas del Arancel de Aduanas Índice Nacional de Criterios de Clasificación. Módulo de Traslado PECO-Amazonía. Control de Precintos Registro de Manifiesto Terrestre Provisional	RLTG,RLDI,RLMC
RLFIAD - Fiscalización Aduanera	Consulta de Empresas no habidas Registro de Empresas No habidas Registro y Consulta de Fichas Electrónicas Alertas Seguimiento de Auditorías de Fiscalización Mercancías sensibles al Fraude Dossier del Operador de Comercio Exterior Importadores Courier Cuenta Corriente Courier Perfil Comercial	RLFI,RLDP
RLREAD - Recaudación Aduanera	Garantías Sanciones Movimiento de L/C Cálculo de Interés de Devolución Cobranzas Coactivas	RLFI, RLRI, RLRS
RLROAD - Regímenes y Operaciones Aduaneras	Regímenes Definitivos Regímenes Suspensivos Regímenes Temporales Perfeccionamiento Activo Operaciones Aduaneras Teledespaço Operadores	RLDI,RLDE, RLTI,RLPA, RLST,RLTE, RLTG,RLIV

Roles de Agrupaciones	Sistemas	Roles de los Sistemas
	Informes de Verificación Dossier de Mercancías Consulta de Documento de Identidad	
RLINCE - Indicadores de Comercio Exterior	Indicadores de Comercio Exterior Indicadores de Despacho Aduanero Percepciones a la Importación	RLTG,RLDI
RLADRE-Administración Remate Aduanas	Administración de Remates de Aduanas	RLCA
RLCONS – Consulta de Todos los Aplicativos	Todos los Sistemas	RLDIC,RLREC,RLDEC,RLMCC,RLTGC,etc.

B. USUARIOS PARA INTRANET, INTERNET, EXTRANET

Estos usuarios serán creados para el desarrollo de los sistemas tributarios y sistemas aduaneros en web.

Usuario de Escritura

Formato:	USXXAGRU
Longitud Máxima:	8 posiciones
Descripción:	US : Prefijo de Usuario XX : Indica el tipo de red Indica el tipo de red IA Indica si es red Intranet IT Indica si es red Internet ET Indica si es red Extranet AGRU : Indica los cuatro sufijos del Rol por agrupaciones de los sistemas web.
Ejemplo:	USIAREAD Usuario Intranet de Recaudación Aduanera USIARETR Usuario Intranet de Recaudación Tributaria

Usuario de Lectura

El usuario de lectura será **USXXCONS**, el cual será utilizado en todas las bases de datos con manejador informix y oracle.

Formato:	USXXCONS
Longitud Máxima:	8 posiciones
Descripción:	US : Prefijo de Usuario XX :Indica que puede ser Intranet(IA) ó Internet(IT) ó Extranet(ET) CONS: Indica que es consulta de la base de datos.
Usuario en todas las b.d.	USXXCONS Usuario de Consulta de la base de datos

DEFINICIÓN DE USUARIOS PARA INTRANET, INTERNET Y EXTRANET DE SISTEMAS ADUANEROS

Usuarios		Agrupaciones de Sistemas WEB
USIAPRCT	Usuario Intranet	PRCT - Prevención del Contrabando
USITPRCT	Usuario Internet	
USETPRCT	Usuario Extranet	
USIATEAD	Usuario Intranet	TEAD - Técnica Aduanera
USITTEAD	Usuario Internet	
USETTEAD	Usuario Extranet	
USIAFIAD	Usuario Intranet	FIAD - Fiscalización Aduanera
USITFIAD	Usuario Internet	
USETFIAD	Usuario Extranet	
USIAREAD	Usuario Intranet	READ – Recaudación Aduanera
USITREAD	Usuario Internet	
USETREAD	Usuario Extranet	
USIAROAD	Usuario Intranet	ROAD - Regímenes y Operaciones Aduaneras
USITROAD	Usuario Internet	
USETROAD	Usuario Extranet	
USIAINCE	Usuario Intranet	INCE - Indicadores de Comercio Exterior
USITINCE	Usuario Internet	
USETINCE	Usuario Extranet	
USIAADRE	Usuario Intranet	ADRE - Administración de Remate de Aduanas
USITADRE	Usuario Internet	
USETADRE	Usuario Extranet	