

# Bases de Datos

---

## CONVENCIONES



## Control del Documento

<b>Código del Documento:</b>	UP_NOR_BD_v2.0
<b>Estado del Documento:</b>	Final
<b>Creador:</b>	Diego G. Mancini
<b>Fecha de Creación:</b>	15/02/2011
<b>Fecha de Finalización:</b>	24/02/2011

LISTA DE APROBACION			
Apellido y Nombre	Puesto	Área	Fecha Aprobación
Armando Cáceres	Director de Desarrollo de Sistemas	UPSTI	01/03/2011
Lucas Vicente	Coordinador Ejecutivo	UPSTI	01/03/2011

HISTORICO DE VERSIONES		
Versión	Fecha Aprobación	Cambios / Razones de cambio / Comentarios
1.0	15/12/2005	Inicial
2.0	03/03/2011	Incorporación de estándares de documentación y almacenamiento de información.

## Tabla de contenido

<b>OBJETIVO</b> .....	4
<b>1 DOCUMENTACIÓN OBLIGATORIA</b> .....	4
1.1 Diagrama de Entidad de Relaciones.....	4
1.2 Volumen de Datos.....	4
1.3 Documentación detallada .....	4
<b>2 ACCESIBILIDAD</b> .....	5
2.1 Introducción .....	5
2.2 Objetos de la base de datos .....	5
2.3 Información almacenada en las tablas de la base de datos .....	5
<b>3 NOMINACIÓN DE TABLAS, VISTAS, OBJETOS</b> .....	5
3.1 Introducción .....	5
3.2 Reglas Generales.....	5
3.3 Nominación de las Tablas.....	5
3.4 Clave primaria.....	6
3.5 Claves Foráneas (Campos de relación).....	7
3.6 Otros Campos .....	7
3.7 Ordenamiento .....	7
3.8 Nominación de Vistas.....	7
3.8.1 Nominación extendida.....	8
3.8.1.1 "E", por Extendida, Tablas Implícitas .....	8
3.8.1.2 "O" Outer Join.....	8
3.8.1.3 Campos en las vistas .....	8
3.9 Consideraciones Especiales .....	9
3.9.1 Campos de la Pre_Form_07w_13.....	10
3.9.2 Tablas con nombres de campo no normalizados.....	10

## OBJETIVO

El presente tiene como objetivo establecer una guía detallada sobre la normativa vigente para el almacenamiento, seguridad, nomenclatura de objetos y accesibilidad a la información almacenada en la base de datos.

## 1 DOCUMENTACIÓN OBLIGATORIA

Se debe presentar documentación detallada sobre el modelo de datos, volumen de datos, procedimientos almacenados (stores procedures, packages, functions, triggers).

### 1.1 Diagrama de Entidad de Relaciones

Los diagramas deben representarse con los modelos IDEF1X, IE (James Martin) o IE (Crow's Foot).

### 1.2 Volumen de Datos

Se debe presentar un documento con el espacio estimado que ocupará cada tabla en la base de datos.

Esquema (Owner)	Nombre de la Tabla	Espacio apurado por registro	Estimado de Registros	Espacio Total (MB/GB)

### 1.3 Documentación detallada

Se debe presentar un documento especificando los objetos de la base de datos:

- Tablas con sus restricciones

NOMBRE DE TABLA					
DESCRIPCIÓN					
Tipo	Nombre de Campo	Tipo de Datos	Restricción	Valor Predeterminado	Descripción

DISPARADORES		
Nombre	Eventos	Descripción

INDICES		
Nombre	Campos	Tipo

- Procedimientos almacenados con la descripción general del comportamiento

PROCEDIMIENTOS		
Tipo	Nombre	Descripción

- Vistas con la descripción general del contenido

VISTAS	
Nombre	Descripción

## 2 ACCESIBILIDAD

---

### 2.1 Introducción

La accesibilidad comprende tanto a objetos de la base de datos como a la información almacenada en la misma.

### 2.2 Objetos de la base de datos

La codificación PL/SQL (Procedures, Functions, Triggers, Packages) y el contenido de las vistas deberá ser de código abierto, no deberán estar encriptados.

### 2.3 Información almacenada en las tablas de la base de datos

La información almacenada en las distintas tablas de la base de datos deberá poder ser explotada por cualquier producto y/o vía SQL sin ningún tipo de aplicativo intermediario, a tal efecto:

- No se debe almacenar información codificada o encriptada
- No se debe almacenar información con formato propietario

## 3 NOMINACIÓN DE TABLAS, VISTAS, OBJETOS

---

### 3.1 Introducción

Es importante comprender perfectamente la normativa de nominación de objetos utilizada en el desarrollo para comprender la distribución funcional de los objetos en la base de datos. Mediante el conocimiento del nombre cualquier función, módulo u objeto de la base de datos, es posible establecer su ubicación en el esquema general del sistema, su función y su interdependencia con otros objetos de la base.

### 3.2 Reglas Generales

Todos los objetos deben ser accedidos de forma independiente de su dueño (owner), a tal efecto se debe crear un sinónimo público para cada objeto de la base de datos.

### 3.3 Nominación de las Tablas

Todas las tablas del sistema se nominan de acuerdo al siguiente esquema:

{sst}\_{subg}\_{nn}\_{tpo}\_{descripción}

Utilizaremos como ejemplo la tabla: PRE\_EJEC\_20\_CAB\_COMPROMETIDO conformándose cada grupo como sigue:

**{sst}**, Sistema, (caracteres 1 a 3): determinan el sistema al que pertenece la tabla, en nuestro ejemplo “PRE” (Presupuesto)

**{subg}**, Subgrupo (caracteres 5 a 8): determinan el subgrupo funcional dentro el sistema al cual pertenece la tabla, en nuestro ejemplo “EJEC” (Ejecución Presupuestaria)

**{nn}**, Número secuencial, (caracteres 10 a 11): es un número secuencial dentro de cada sistema y subgrupo, en nuestro ejemplo “01”. De esta manera, la combinación de **{sst}\_{subg}\_{nn}** determinan un nombre único para la tabla y su sinónimo público. Todas las referencias a la tabla se realizan mediante su sinónimo público, en nuestro ejemplo: “PRE\_EJEC\_01”

**{tpo}**, Tipo (caracteres 13 a 15): determinan el tipo de tabla en cuanto a su importancia y relaciones con otras tablas. El Tipo puede ser:

- **TBL**: Tabla de códigos, independiente. Contiene información básica del sistema y es referenciada de una o más tablas específicas. Ejemplo: PRE\_FORM\_01\_TBL\_MEDIDAS (Tabla de Unidades de Medida)
- **CAB**: Tabla de cabecera, independiente. Contiene generalmente la cabecera de un documento o transacción. Existirá una o varias tablas de detalle (tipo DET) que la referencien. Ejemplo: PRE\_EJEC\_20\_CAB\_COMPROMETIDO (Tabla de Documentos de Compromiso)
- **DET**: Tabla de detalle, dependiente de una tabla tipo CAB. Contiene generalmente los detalles de un documento o transacción. Existirá una tabla de cabecera (tipo CAB) referenciada desde esta tabla. Ejemplo: PRE\_EJEC\_21\_DET\_COMPROMISOS (Tabla de Detalles de los Documentos de Compromiso)
- **MOV**: Tabla de movimientos, dependiente de más de una tabla tipo CAB o DET. Contiene generalmente los detalles que resultan de la interrelación de más de un documento o transacción. Existirá en esta tabla una referencia a más de una tabla, de la cuales depende. Ejemplo: PRE\_ASIE\_03\_MOV\_BANCOS (Relación entre los bancos y las cuentas Contables)
- **ARB**: Tabla de estructura jerárquica (árbol). Contiene una estructura en forma de árbol, por lo que contendrá una referencia así misma (campo RELA\_PADRE) Ejemplo: PRE\_FORM\_02\_ARB\_INSTITUCIONES (Tabla del Clasificador Institucional)

**{descripción}**, Descripción (resto de los caracteres): Descripción libre aclaratoria de la función de la tabla.

### 3.4 Clave primaria

Todas las tablas del sistema se identifican por una clave primaria normada, la cual es generada automáticamente en el momento de inserción del registro. Dada esta normalización no se permite el establecimiento de claves primarias por metadato (más de un campo o campos e información del usuario). Al ser la clave un único campo numérico se maximiza el tiempo de respuesta en las operaciones de indexado y resolución de vistas en la Base de Datos.

El campo de clave primaria se compone del prefijo “ID” seguido del sinónimo público de la tabla. Para nuestro ejemplo, la clave primaria de la tabla PRE\_EJEC\_20\_CAB\_COMPROMETIDO es el campo ID\_PREEJEC20.

### 3.5 Claves Foráneas (Campos de relación)

Al estar todas las tablas del sistema identificadas por una clave primaria normada, los campos de relación entre tablas responde al mismo formato y son por lo tanto únicos también. (para relacionar una tabla con cualquier otra es necesario sólo un campo de relación)

Todo campo de relación se compone del prefijo “RELA” seguido del sinónimo público de la tabla referenciada, permitiéndose así un rápido reconocimiento de la estructura relacionada de cualquier tabla, y también así facilitando la creación de vistas y consultas.

Por ejemplo la tabla PRE\_EJEC\_21\_DET\_COMPROMETIDO tendrá un campo relacionándola con su tabla cabecera (PRE\_EJEC\_20\_CAB\_COMPROMETIDO). El campo de relación se denominará RELA\_PREEJEC20.

### 3.6 Otros Campos

Todo campo que no sea *clave primaria* o *de relación* se debe componer con el nombre del sinónimo público como prefijo, seguido por el underscore y el nombre del campo.

Por ejemplo la tabla PRE\_EJEC\_21\_DET\_COMPROMETIDO tendrá el campo fecha de alta El campo de se denominará PREEJEC21\_FAPL.

### 3.7 Ordenamiento

Por convención se colocarán al principio de la tabla todos los campos de relación (prefijo RELA), luego se colocará el campo de identificación (clave primaria, prefijo ID) y luego los campos que representen los atributos específicos de la tabla. Este ordenamiento facilita la identificación de la estructura relacionada de cualquier tabla.

### 3.8 Nominación de Vistas

Las vistas se nominarán de acuerdo a su tabla troncal. La tabla troncal de una vista es aquella que tiene concordancia **uno a uno** de registros con la vista. El ID de la tabla troncal **no se repite** en la vista. Por cada registro de la tabla troncal hay un solo registro en la vista.

Luego del nombre de la tabla troncal se agrega la letra “W” (por vieW) y un underscore ‘\_’ y los números correspondientes a las demás tablas involucradas, si existe una tabla con un prefijo diferente a la que la precede en la lista de números, se colocarán las iniciales del prefijo antes del número.

Por ejemplo:

PRE\_FORM\_17W\_15\_12\_SU02\_15

Es una vista con la tabla PRE\_FORM\_17 como tabla troncal, con joins a las tablas PRE\_FORM\_15 y PRE\_FORM\_12, y además joins a las tablas SOC\_USUA\_02 y SOC\_USUA\_15.

El orden en el que se listan las tablas joineadas a la tabla troncal con el mismo prefijo deberá adecuarse en lo posible al camino de obtención de los datos. En el caso presentado: “PRE\_FORM\_17W\_15\_12\_SU02\_15” se entiende que desde la PRE\_FORM\_17 se accede a la PRE\_FORM\_15, y desde la PRE\_FORM\_15 se accede a la PRE\_FORM\_12. Las tablas accesorias de otro esquema (en este caso SU02\_15) se colocan al final.

La identificación de la tabla troncal de la vista es muy importante. Por ejemplo, supongamos que la tabla PRE\_EJEC\_10, es cabecera de la tabla PRE\_EJEC\_12. (relación uno a muchos) La tabla PRE\_EJEC\_12 contiene un campo PREEJEC12\_IMPORTE.

Si existiese una vista llamada: **PRE\_EJEC\_10W\_12**, se entiende que la tabla troncal será la PRE\_EJEC\_10, y que además contiene datos de la PRE\_EJEC\_12.

Siendo la PRE\_EJEC\_10 la troncal, existirá en la vista un registro por cada registro de la PRE\_EJEC\_10 (cabecera) y entonces, los campos correspondientes a la PRE\_EJEC\_12 (detalle), estarán sumados, o agrupados de alguna manera.

Es de esperar en esta vista de encontrarse con un campo SUM\_PREEJEC12\_IMPORTE que correspondería a la suma de los importes del detalle.

En cambio, en una vista llamada **PRE\_EJEC\_12W\_10**, se entiende que la tabla troncal es la PRE\_EJEC\_12, y que además incorpora datos de la PRE\_EJEC\_10 (cabecera). Se entiende entonces, que existe un registro en la vista por cada registro en la PRE\_EJEC\_12, y por lo tanto los datos correspondientes a la cabecera (PRE\_EJEC\_10) se repetirán en todos los registros del detalle.

No se espera ningún campo agrupado en esta vista.

### **3.8.1 Nominación extendida**

#### **3.8.1.1 “E”, por Extendida, Tablas Implícitas**

Cuando las tablas participantes son demasiadas o no es necesario especificarlas por no ser relevantes para la vista, se puede agrega la letra “E” al final de la tabla principal del grupo. Se entenderá “E” por “Extendido”.

Por Ejemplo:

Nombre de la vista: EJE\_REND\_01\_SU02E

Se entiende: la tabla EJE\_REND\_01 como troncal, con datos extendidos de la SOC\_USUA\_02. Conociendo el modelo está implícito que los datos extendidos de la SU02 (Puesto de un usuario del sistema) son las tablas SU01, SU15, SU08, PF11 y PF12. (Usuario, Entidad, Documento, U.Orgánica y Cargo)

#### **3.8.1.2 “O” Outer Join**

Cuando una de las tablas participantes este joineada a la tabla troncal mediante un outer join, se agregará la letra “O” DELANTE de la tabla joineada. Es necesario para diferenciar vistas con y sin outer joins.

Por Ejemplo:

PRE\_EJEC\_21W\_22 <- Todos los registros de la 21 con relación en la 22

PRE\_EJEC\_21W\_O22 <- Todos los registros de la 21 tengan o no 22

#### **3.8.1.3 Campos en las vistas**

##### **Resultantes de Operaciones de agrupación (aggregate functions)**

Los campos que resulten de operaciones de agrupación, se nominarán agregándole la operación como prefijo

Por ejemplo

Sum (PREEJEC12\_IMPORTE) as SUM\_PREEJEC12\_IMPORTE

Count (ID\_SYSFUNCC03) as COUNT\_ID\_SYSFUNCC03

## Resultantes de Decodificaciones (DECODE)

Si una tabla tiene un campo `_COD` o `_TIPO`, codificado con valores 0..n, se creará una vista con el nombre de la tabla + "W" en la cual se incluye un campo con el decode para la descripción del campo codificado. Al campo calculado con decode se le agrega el sufijo `_TEXT`.

Por Ejemplo, la tabla `EJE_ABAS_01` tiene un campo `EJEABAS01_TIPO`, codificado como: 0=Proveedor, 1=Responsable de Fondo. Se creará una vista `EJE_ABAS_01W`, de la siguiente forma:

```
CREATE VIEW EJE_ABAS_01W AS
Select
  RELA_xxxx...
  ,ID_EJEABAS01
  ,EJEABAS01_DESCRI
  ,EJEABAS01_TIPO
  ,Decode(EJEABAS01_TIPO
    ,0,'Proveedor',
    ,1,'Responsable Fondo'
    ,'Tipo '||EJEABAS01_TIPO) as EJEABAS01_TIPO_TEXT
From EJE_ABAS_01
```

A partir de aquí se utilizará esta vista cada vez que se necesite tener la descripción del campo codificado. De esta manera, el decode para el campo se encuentra en un solo lugar.

## Campos RELA existentes en más de una tabla

Es posible que se repita un campo RELA entre las tablas que componen la vista. Se nominarán agregando como prefijo el nombre abreviado de la tabla ***para las tablas no-troncales***.

Por Ejemplo:

```
Create View PRE_EJEC_10W_11_12_02 as
Select
  PRE_EJEC_10.RELA_SOCUSUA02 (rela en la tabla troncal, sin cambios)
  ,PRE_EJEC_11.RELA_SOCUSUA02 as PE11_RELAS_SOCUSUA02 (tabla
secundaria)
  ,PRE_EJEC_02.RELA_SOCUSUA02 as PE02_RELAS_SOCUSUA02 (otra tabla
secundaria)
```

## 3.9 Consideraciones Especiales

### 3.9.1 Campos de la Pre\_Form\_07w\_13

Siendo muy probable que una vista medianamente compleja contenga dos o tres veces esta tabla, se nominarán los campos utilizando una abreviatura del metatipo del clasificador involucrado, por ejemplo:

PF07\_PRG\_CODIGO -> PREFORM07\_CODIGO de un clasificador de programas

PF07\_PRG\_DESCRI -> PREFORM13\_DESCRI de un clasificador de programas

PF07\_OG\_DESCRI -> PREFORM13\_DESCRI de un Objeto del Gasto

### 3.9.2 Tablas con nombres de campo no normalizados

#### Tabla PRE\_FORM\_15 (Conceptos del Gasto)

Esta tabla contiene campos con nombres iguales a los de la tabla PRE\_FORM\_07, para evitar confusiones, usar la vista **PRE\_FORM\_15N**, que posee los campos normalizados.

#### Tablas PRE\_FORM\_01 y PRE\_FORM\_02 (Períodos e Instituciones)

Estas tablas contienen campos con nombres no normalizados, usar las vistas **PRE\_FORM\_01N** y **PRE\_FORM\_02N** que poseen los campos normalizados.

#### Tabla PRE\_FORM\_03 (Mov Período/Institución)

Esta tabla contiene campos con nombres no normalizados, usar las vistas **PRE\_FORM\_03N**, **PRE\_FORM\_03W\_02N** y **PRE\_FORM\_03W\_02\_01** que poseen los campos normalizados.

#### Otras Tablas Normalizadas por una Vista "N"

PRE\_FORM\_07N, PRE\_FORM\_08N, PRE\_FORM\_07W\_13, PRE\_FORM\_07WN\_05\_18, PRE\_FORM\_11N (Orgánicas), PRE\_FORM\_12N (Cargos), PRE\_FORM\_16N, SOC\_USUA\_01N (Usuarios), SOC\_USUA\_02N (Identidades de usuarios), SOC\_USUA\_15N (Entidades), PRE\_FORM\_05N, PRE\_FORM\_06N, PRE\_BIEN\_02N

#### Regla General

En caso de hallar una tabla con campos no normalizados, utilizar la vista "N" correspondiente con los campos normalizados. Usar el nombre de la tabla seguido de la letra "N" cuando se realiza una vista que sólo normalizan los nombres de los campos de una tabla.