



Universidad Autónoma de Chihuahua

Coordinación General de Tecnologías de Información

Departamento de Sistemas de Información

ESTANDARES DE GESTION DE BASES DE DATOS

BASES DE DATOS.

- 1 El nombre de la base de datos debe estar compuesto de solo sustantivos, en minúsculas con letra capital. En caso de ser compuesta por dos o más palabras, cada palabra inicia con letra capital. No se debe hacer uso de abreviaturas, y en caso de la utilización de siglas debe de ser validado por el jefe del Departamento de Sistemas de Información en el formato de Solicitud para utilización de Siglas en el nombre de una Base de Datos **FOR 7.3 DSI 07.**
- 2 El nombre de bases de datos de Internet debe comenzar con la letra "e" (por ejemplo eAcademicoServicioSocial, eFinancieroCajaUnica, etc.).
- 3 El lugar físico de la bases de datos debe de estar en una carpeta llamada "Databases" creada en el servidor donde residirá la base de datos, esta carpeta debe de ser generada por el administrador de los servidores. Dentro de ésta se debe escribir el nombre de la base de datos.

Ejemplo: La base de datos "FinancieroCajaUnica" se debe encontrar en
raiz\Databases\FinancieroCajaUnica\

- 4 El nombre físico y el nombre lógico de los dispositivos de bases de datos debe ser igual al nombre de la base de datos más una terminación que indique a que dispositivo se refiere. Para el registro de transacciones se debe usar la terminación "Log", para el archivo de datos la terminación debe ser "Dat" y para el archivo de Respaldo debe ser únicamente el nombre de la base de datos "Dump". Por ejemplo, los nombres para los archivos de la base de datos "FinancieroCajaUnica" debe ser:
 - o Archivo de Datos.- FinancieroCajaUnicaDat.
 - o Registro de Transacciones.- FinancieroCajaUnicaLog.
 - o Archivo de Respaldo.- FinancieroCajaUnicaDump.
- 5 Al crear una base de datos en SQL Server se debe tener presente la siguiente configuración:
El modelo de recuperación de la base de datos: "Completo" "Simple"
El nivel de compatibilidad de la base de datos: "Nivel 80 (para las bases de datos migradas de SQL Server 2000 a SQL Server 2005) ó Nivel 90 (para las bases de datos nativas de SQL Server 2005) de compatibilidad de la base de datos".
- 6 La documentación de la BD debe contener :
 - o Diccionario de datos que incluya la descripción de la tabla y campos que la conforman.

COPIA
NO CONTROLADA



Universidad Autónoma de Chihuahua

Coordinación General de Tecnologías de Información

Departamento de Sistemas de Información

- o Diagrama de relaciones entre tablas donde se visualicen las tablas que conforman la base de datos y sus relaciones. Las relaciones de tablas con otras Bases de Datos se debe indicar a través de un cuadro de texto para evidenciar su existencia.
 - o El nombre del Diagrama debe de contener la palabra "Diagrama" seguida del nombre del modulo y submódulo correspondiente.
- 7 Los usuarios de la base de datos debe tener las siguientes características:
- o Comenzar con "USR_" seguido del nombre del módulo y Submódulo del que se está accediendo, con la primera letra en capital; o en caso de ser compuesto de dos o más palabras, cada una de ellas debe iniciar en letra capital y no se deben incluir siglas ni abreviaturas, Ejemplo: USR_FinancieroBienesPatrimoniales.
 - o Los nombres de usuarios directos o desarrolladores deben de comenzar con el prefijo "USR_", seguido de la primera letra del nombre en mayúscula y a continuación el primer apellido iniciando con letra capital. En caso de que existan dos usuarios con nombre y apellido paterno igual, agregar el segundo apellido iniciando la primera letra en mayúscula.

TABLAS

1. Se debe evitar duplicar información, es decir, los catálogos solo deben de existir en la base de datos donde recae la responsabilidad de su administración (caso de los programas educativos (SAGU Académico), unidades académicas (SAGU Financiero-contabilidad), etc.) Y si se requiere ampliar la información de un catálogo esta información debe estar contenida en una tabla anexa en la Base de Datos que la necesita, con la autorización del propietario del Catálogo, utilizando el formato de Solicitud para la Creación de Anexo de Catalogo **FOR 7.3 DSI 06**.

Se debe agregar al nombre de la nueva tabla la palabra "Anexos" al final. Por ejemplo: la tabla catcFacultades tiene como anexo la tabla catcFacultadesAnexos.

2. Los nombres de las tablas debe tener los siguientes requisitos:
 - o Prefijo en minúsculas y el nombre en capital de cada palabra.
 - o Deberá tener un sentido plural.
 - o No usar abreviaturas, ni siglas
 - o Conservar un sentido nemotécnico y considerando que solo pueden existir tablas para catálogos, movimientos y de sistema.
 - Bajo la siguiente estructura

tiponivelNombre

Donde:

tipo = indica si la tabla es de Catálogo, Movimiento o Sistema, de acuerdo a la siguiente definición:

COPIA
NO CONTROLADA



Universidad Autónoma de Chihuahua

Coordinación General de Tecnologías de Información

Departamento de Sistemas de Información

Catalogo: Tabla que se utiliza para ser referenciada por otras.

Sistema: Tabla donde se controla el funcionamiento del sistema y generalmente no son visualizadas por el usuario

Movimiento: tabla de trabajo, donde se almacena la información de las reglas de negocio, generalmente guarda relaciones con varios catálogos e información del proceso.

nivel = indica si la tabla es cabecera o detalle, de acuerdo a la siguiente definición:

Cabecera: Tabla con registros que marca inicios de las relaciones.

Detalle: Tabla que guarda una relación de uno a uno o muchos con su cabecera.

Nombre = Nombre de la tabla

Ejemplo:

Catálogos cabecera
catcUsuarios

Catálogos detalle
catdUsuarios

Movimientos cabecera
movcTransacciones

Movimientos detalle
movdTransacciones

Sistema cabecera
siscBitacoras

Sistema detalle
sisdBitacoras

3. Los nombres de los campos deben tener las siguientes características:

- o El primer campo a agregar debe de ser el campo de identificación de registro (identity).
- o Debe ser nemotécnico.
- o Palabras empleadas con letra capital.
- o Estar compuesto de caracteres alfabéticos, sin abreviaturas y todo en español.
- o Uso del prefijo "Id" para claves primarias, si se requiere el uso de una clave secundaria, ésta se debe nombrar igual que la primaria sin el prefijo id.
- o Las claves foráneas deben tener el mismo nombre que la clave primaria de la tabla propietaria.
- o No se deberán definir los campos como nulos

COPIA
NO CONTROLADA



Universidad Autónoma de Chihuahua **Coordinación General de Tecnologías de Información** **Departamento de Sistemas de Información**

4. La tabla siscBitacoras debe contener la siguiente estructura:

Descripción de la Tabla	Campos	Tipo	Longitud	Descripción del campo
Tabla que contiene todos los movimientos realizados por todos los usuarios del Modulo.	IdBitacora	int	4	Identificador único de la bitácora
	Quien	varchar	20	Usuario que realizó la acción
	Cuando	smalldatetime	4	Fecha en que se realiza la acción
	Donde	varchar	50	Campo que registra la dirección IP y en el caso de plataforma tradicional el nombre de la maquina donde se realizó el movimiento
	Que	varchar	800	Sentencia SQL en caso de altas, bajas y cambios, como en caso de consultas una descripción significativa de la misma, indicando sus filtros, siempre y cuando sean opciones de consultas y reportes del sistema.
	NombrePantalla	varchar	60	Nombre de pantalla dentro del sistema Promep donde se realizó la acción
	TipoDeRegistro	varchar	40	tipo de registro

5. Los usuarios externos al DSI deberán tener solo permiso de “select” autorizado por el jefe de departamento.

6. Al requerir información para alimentar sistemas de información externos al departamento, se deberá realizar solo a través de procedimientos almacenados previa autorización del jefe del departamento. Estos procedimientos deben estar encriptados, considerando que en la construcción del SP debe de establecerse la restricción que no corresponde al solicitante

INSTRUCCIONES DE SQL

Las instrucciones de SQL deben de conservar la siguiente estructura:

```

Select  Campo1, Campo2, .....
From    Tabla1 alias1
Inner   Join    Tabla2 alias2
          On      alias1.campoN = alias2.campoM
          And ...
          Or...
Inner   Join    (Select campo1, campo2, ...
          From ...
          Inner Join ...
          :
          ) as alias3
          On      alias2.campoN = alias3.campoN
  
```

**COPIA
NO CONTROLADA**



Universidad Autónoma de Chihuahua

Coordinación General de Tecnologías de Información

Departamento de Sistemas de Información

Union

NOTA: Después de cada instrucción SQL debe aparecer un tabulador.

Considerándose que:

1. Toda palabra reservada debe tener letra capital.
2. Entre los campos de la Sentencia de Selección SQL (Select), después de la coma (,) se debe dejar un espacio sencillo.
3. Después de cualquiera de las palabras reservadas: Select, From, Where, Group By, Order By y Union debe existir un tab para que quede visualmente alineado.
4. Las relaciones entre tablas se debe de realizar mediante la sentencia Join On
5. La sentencia Where debe ser utilizada sólo para condiciones de selección (parámetros o valores), no para relaciones.
6. En la sección de origen de la sentencia SQL (From), debe aparecer la tabla principal de la consulta.
7. El orden de asignación de alias deberá ser en orden alfabético ascendente con una sola literal con minúsculas, en caso de que las letras del alfabeto se terminen se procederá a combinar 2 letras bajo la misma estructura (aa, ab, ac.....az, ba, bb,.....)
8. Cuando haya consultas anidadas se seguirá con la misma lógica para la subconsulta, con su indentación correspondiente.
9. La asignación del valor resultante de un campo calculado debe de asignarse con la palabra reservada AS.
10. Para eficientizar las consultas no se permite utilizar subconsultas en los campos de selección de otra consulta, éstas ligas deberán realizarse en el Join de la consulta.

COPIA
NO CONTROLADA



Universidad Autónoma de Chihuahua

Coordinación General de Tecnologías de Información

Departamento de Sistemas de Información

```
Select  Campo1, Campo2, .....  
        Campon, Campon+1,  
        (Select  Campo1  
From    tabla1 alias1  
Where   condicion1 = x  
        And    condicion2 = y  
        Or     condicionn = z  
        And    Condicion in ()  
        ) as variable1  
From    tabla1 alias1
```

PROCEDIMIENTOS ALMACENADOS y TRIGGERS

1. El nombre del procedimiento debe comenzar con "sp_", seguido del nombre de la tabla o proceso al que se hace referencia.
2. El nombre del trigger debe comenzar con "tr_", seguido del nombre de la tabla y acción en donde se desencadena.
3. La estructura de opciones del procedimiento almacenado, deben ser de la siguiente manera:
 1. Alta: la acción principal es la instrucción INSERT
 2. Baja: la acción principal es la instrucción DELETE.
 3. Modificación: la acción principal es la instrucción UPDATE, para todos los campos, excepto la clave primaria.
 4. Consulta general por la clave primaria.

Estas opciones se emplean para los procesos generales, si se requiere de otra alta, baja, modificación o consulta, se puede agregar en otra opción.

En el caso de que el procedimiento no esté elaborado para una administración de tablas, la opción debe de comenzar en la 5.

4. El tipo de dato para la opción debe ser de tipo smallint y el nombre para la variable debe ser @opcion, y no se deben contemplar mas de 50 opciones en un procedimiento almacenado.
5. En el procedimiento almacenado se deben utilizar las instrucciones BEGIN TRAN y COMMIT TRAN obligatoriamente con las sentencias INSERT, UPDATE y DELETE. De tal forma que se debe de emplear la sentencia XACT_ABORT ON para el manejo del BEGIN TRAN Y COMMIT TRAN

COPIA
NO CONTROLADA



Universidad Autónoma de Chihuahua **Coordinación General de Tecnologías de Información** **Departamento de Sistemas de Información**

Para el caso de transacciones distribuidas, se debe incluir en cada una de las opciones que utilizan las sentencias INSERT, UPDATE y DELETE, el manejo de errores, Ejemplo:

SQL Server 2000	SQL Server 2005
<pre> If @Opcion = n Begin Begin Tran Insert Into Servr Values If @@Error <> 0 Begin Update..... Set..... If @@Error =0 Begin Commit End Else Begin Rollback End End Else Begin Rollback End </pre>	<pre> If @Opcion = n Begin Begin Tran Begin Try --Codigo End Try Begin Catch Select Error_Message() If @@TranCount > 0 Begin RollBack End End Catch If @@TranCount > 0 Begin Commit Tran End </pre>

6. El procedimiento debe de contar con los siguientes comentarios identados, en la parte superior, antes del nombre:

- Objetivo del procedimiento.
- Nombre del autor.
- Fecha de creación.
- Submódulo(s). (con el objetivo de identificar en que submódulos se usa el procedimiento).
- Historia de cambios, con el siguiente formato:

No. Solicitud	Fecha	Nombre	Descripción	Versión del estandar
---------------	-------	--------	-------------	----------------------

En donde:

No. Solicitud	Se refiere a la solicitud de servicio.
Fecha	Fecha de modificación en formato dd -mmm- aaaa.
Nombre	Persona que modificó.
Descripción	Cambios realizados.
Versión de estándar	Número de la revisión del DOC 7.3 DSI 01 aplicado

**COPIA
NO CONTROLADA**



Universidad Autónoma de Chihuahua

Coordinación General de Tecnologías de Información

Departamento de Sistemas de Información

7. En la parte superior de cada opción del procedimiento almacenado, deberá existir un comentario con una breve descripción de lo que hace esa opción.
8. El Trigger debe de contar con los siguientes comentarios identados, en la parte superior:
 - a. Objetivo del procedimiento.
 - b. Nombre del autor.
 - c. Fecha de creación.
 - d. Historia de cambios, con el siguiente formato:

No. Solicitud	Fecha	Nombre	Descripción	Versión del estándar
---------------	-------	--------	-------------	----------------------

En donde:

No. Solicitud Se refiere a la solicitud de servicio.

Fecha Fecha de modificación en formato dd -mmm- aaaa.

Nombre Persona que modificó.

Descripción Cambios realizados.

Versión de estándar Número de la revisión del DOC 7.3 DSI 01 aplicado

9. Se debe mantener la indentación en las líneas del procedimiento que se describe:

```
IF @Opcion      = 1
    Begin
        Begin Tran

            Commit Tran

        End
    Else
        Begin

        End
```

INDICES

Se debe contar al menos con un índice por tabla, en cual debe de incluir la clave primaria.

Para el nombre del índice, se debe iniciar con el prefijo, que no indica el tipo de índice:

PK = Llave primaria

FK = Llave Foránea

IX = Para cualquier otro tipo de índice, es decir, índice único, índice compuesto, índice con duplicados etc...

Seguido del nombre de la columna o las columnas sobre las que se aplica el índice, todo lo anterior debe estar separado con un guión bajo.

COPIA
NO CONTROLADA



Universidad Autónoma de Chihuahua

Coordinación General de Tecnologías de Información

Departamento de Sistemas de Información

Ejemplo:

PK_catcCajas_NumCaja

VISTAS

1. Los nombres de las vistas debe tener las siguientes características:

- o Comenzar con la letra "V" seguida del nombre con sentido nemotécnico que referencia la consulta que se está almacenando, en minúsculas con la primera letra capital y en plural.
- o Si el nombre esta compuesto por varias palabras, estas deben de estar en minúsculas con la primera letra capital.
- o Utilizar solo caracteres alfabéticos, sin abreviaturas, sin siglas y todo en español.

Ejemplo: VHorariosEmpleados

XML

Si son varias las operaciones en la Base de Datos dentro de la misma transacción, para asegurar el comprimiso de los datos se utilizará XML

1. Al utilizar XML en una aplicación y este documento sea enviado a la Base de Datos, este deberá ser recibido en una variable llamada @Documento de tipo ntext.
2. La variable apuntador que manejará el archivo deberá tener por nombre @DocumentoXML de tipo entero.
3. En la bandera de la sentencia OPENXML deberá utilizarse el valor 1 que significa que el nombre de los atributos serán las columnas de la tabla virtual y los renglones cada uno de los tags del documento.
4. Para preparar el documento XML se utilizará el procedimiento almacenado de sistema sp_xml_preparedocument y por fuerza se deberá emplear el procedimiento almacenado de sistema sp_xml_removedocument para liberar la memoria una vez que se haya terminado de utilizar dicho documento.

Para efectos del ejemplo la variable @Documento será asignada

```
Declare @Documento ntext
```

```
Set = '<ALUMNOS><DATOS><Alumno matricula = 168552/><Alumno = 168708/></DATOS></ALUMNOS>'
```

```
Declare @DocumentoXML int
```

```
Exec sp_xml_preparedocument @DocumentoXML OutPut, @Documento
```

```
Select Matricula
```

```
From OpenXML (@DocumentoXML, '/ALUMNOS/DATOS/Alumno',1)
```

```
With (Matricula int)
```

```
Exec sp_xml_removedocument
```

COPIA
NO CONTROLADA