**POLITICAS DE DISEÑO DE BASE DE DATOS**

**Objetivo.-** Con este se busca estandarizar aspectos relevantes de los objetos como el uso de nombres, documentación, seguridad, y rendimiento.

**Estructura de tablas**

Los campos de una tabla deben tener la siguiente implementación:

1. Llave artificial
2. Llave Natural(Índice Único)
3. Todos los demás campos en orden natural

Se recomienda que las imágenes, binarios y textos largos deben incluirse en una tabla separada

**Nombres de base de datos**

* Deben representar el propósito de la misma
* Se deben usar la nomenclatura Pascal
* No deben ser necesariamente iguales al nombre de la aplicación

Ejemplo SIGMA, SIGEP, SIE, etc.

**Nombres de las tablas**

* Los nombres siempre serán sustantivos en singular(plural).
* Deben empezar con un acrónimo que permita agrupar de alguna manera lógica o funcional las tablas que estén asociadas, seguido del underscore.
* El alias de la tabla debe ser único y debe tener 3 a 10 letras como máximo.

Ejemplos Emp\_Empleados, Emp\_Proyectos

**Columnas**

* En algunos casos se recomienda usar acronimos para definir el tipo de dato del campo, ejemplo:

vc\_NombreEmpleado (varchar),b\_Cancelado (Boolean)

* Preferentemente las columas asociadas a las llaves foráneas no deberán tener ningún sufijo ni prefijo y llevaran el mismo nombre de la llave primaria de la tabla origen.
* El nombre de la descripción de la tabal deberá tener la siguiente nomenclatura:

desc\_<nombre de tabla> Ej. desc\_banco

* En algunos casos se usan abreviaturas para el resto de campos que agrupen ciertos campos , ejemplo:

Ejemplos: dir\_ …., ctrl\_...., resp\_...

**Nombres de llaves primarias**

* Se recomienda que la llave primaria sea un llave artificial que genere un numero secuencial y único.
* Se debe usar como acrónimo «PK» para identificar que se trata de una llave primaria

Ejemplo : pk\_entidades

**Constraints**

* Foreign Key.- Se debe usar el acrónimo ‘FK‘ seguido por el carácter ‘\_’ alisas hasta ,’\_’ y alias desde.

Estructura: fk\_<alias\_hasta>\_<alias\_desde>

Ejemplo: fk\_da\_ent

* Unique Key.- Se debe usar el acrónimo ‘UK’ seguido de ‘\_’ y un alias

Estructura: uk\_<alias>

Ejemplo: uk\_ents

**Indices**

* Los índices se denominaran de la siguiente manera:

I\_<alias>\_<descripción>

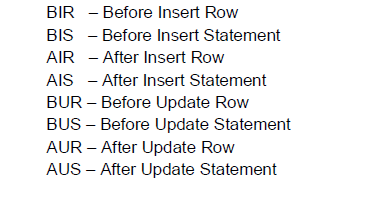
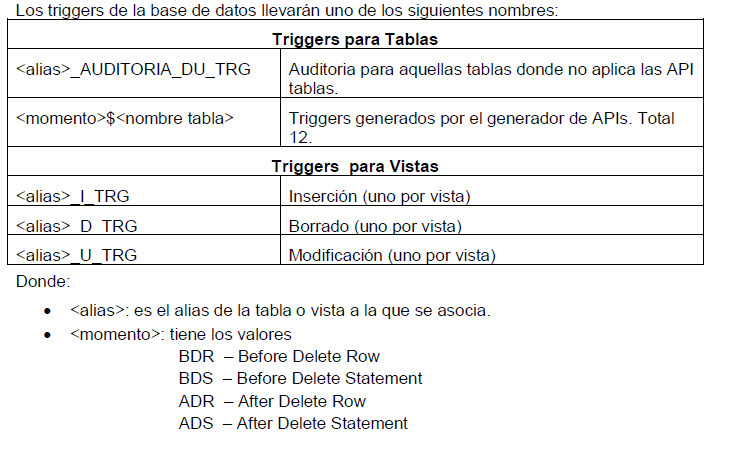
Donde el alias es el alias de la tabla y la descripción es un comentario memotécnico de la utilidad del mismo.

Ejemplo: I\_Ent\_FechaIngreso

* Los índices relacionados a llaves foráneas deben conservar la denominación de la constraint precedida del prefijo I

Ejemplo: I\_Fk\_CtaBen\_Ban

**Triggers**



## 

## Modelo conceptual

El diagrama a continuación representa una realidad hipotética cuya información se quiere almacenar en base de datos..

|  |  |
| --- | --- |
|  | **En esta realidad existen compañías que tienen productos. Los productos son vendidos a clientes, mediante ordenes de compra.**  **Una misma orden de compra puede utilizarse para comprar varios productos de una compañía**  **El precio de los productos puede variar con el tiempo, pero se debe almacenar con el precio que fue vendido en cada ocasión.**  **Restricción no estructural: Una compañía solo emite ordenes de compra de los aquellos productos que posee.** |

## MAL Ejemplo

El siguiente es un mal diseño de esquema relacional para la realidad anteriormente planteada.

|  |  |
| --- | --- |
| mal ejemplo | 1. **Mezcla idiomas** 2. **Tablas en singular** 3. **Un mismo campo recibe distintos nombres** 4. **No define FK** 5. **No define PK** 6. **Las entidades Orders y OrderItems residen en una misma tabla, campos redundantes (no es 3NF)** 7. **Abreviaciones** 8. **Pascal case no respetado** |

## Ejemplo sugerido

El siguiente es el ejemplo sugerido de diseño para la realidad anteriormente planteada:

|  |  |
| --- | --- |
|  | **Observaciones:**   * **Como criterio se prefijaron con el nombre de la tabla aquellas columnas cuyos nombres se encuentran repetidos en varias tablas, para evitar problemas de ambigüedad en las consultas** * **En la tabla Orders se define una clave de un solo campo para evitar incluir el campo CompanyId en la clave de OrderItems** |

## Relaciones 1:N

Para la definición de tablas que correspondan a una relación 1:N como puede ser la siguiente:

|  |  |
| --- | --- |
|  | **Una compañía emplea 1..N empleados** |

Se sugiere utilizar una estructura de tablas como la siguiente:

|  |  |
| --- | --- |
|  |  |

## Relaciones N:M

Para la definición de tablas que correspondan a relaciones N:M com puede ser la siguiente:

|  |  |
| --- | --- |
|  | **Un doctor atiende N pacientes, y un paciente es atendido por M doctores.** |

Se sugiere utilizar una estructura de tablas como la siguiente:

|  |  |
| --- | --- |
|  |  |