**Estructura de la Base de Datos**

Las tablas/Vistas/Procedimientos almacenados deben tener la siguiente codificación:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Concepto | Base | Módulo | Sección | Elemento | Nombre |
| Caracteres | 1 | 2 | 2 | 1 | 7 |

Por lo tanto, el nombre de un elemento será de 12 caracteres. Las distintas opciones de cada sección:

**Base de Datos**: son las distintas bases de datos que conforman el sistema. Cada una tiene una función específica, las mismas son:

* C: central, en esta BD esta la información central del sistema. Aquí están todas las tablas que estructuran la lógica, sus procesos, idiomas, clasificaciones, interfaces de usuarios, información de los usuarios, etc. En esta base no está la información que cargan los usuarios, excepto todo lo concerniente al ingreso al sistema.
* P: Procesos, en esta BD esta la información de todas las instancias de los procesos, esta es la información real del proceso y la que cargó el usuario.

**Módulo**: estos son en las partes que se puede dividir la base de datos.

* Ge: general, agrupa la información (tablas/vistas/proc alm) general de la base de datos.
* Pr: procesos, agrupa la información (tablas/vistas/proc alm) específica de los procesos.

**Sección**: clasifica los distintos grupos de tablas y elementos dentro del módulo.

* Ge: gestión.
* Veh: vehículos.
* Par: partes.

**Elemento**: clasifica el tipo de elemento de la base de datos.

* T: tabla.
* P: procedimiento almacenado.
* V: vista.

**Nombre**: define al tipo de elemento.

* Dejar claramente el tipo de elemento.

Todas las partes del nombre se deben escribir utilizando Camel. Con esta codificación podemos ordenar los elementos correctamente.

Ejemplos:

CPrGeTProcesos (Central, Proceso, General, Tabla, Procesos)

CVGrlIdiomas (Central, Grl, Vista, Idiomas = Idiomas del sistema

**Vistas**

Todas las vistas del sistema deben tener la cláusula WITH (NOLOCK)

**Campo Clave ID = uniqueidentifier**

**ID** = Todas las tablas del sistema representan el valor único con el nombre ID. Por lo tanto, para saber de qué tabla es originario el valor, solo hay que buscar el valor ID de cada tabla y en la que se encuentre pertenece el valor.

Estos ID se colocan en otras tablas poniendo el ID seguido del nombre de la tabla: ej: ID GDicDescPorIdiomas

**Campo Codificaciones**

**ID** = valor único de la tabla. Uniqueidentifier.

**IDgdicTCodigos** = vínculo con otra tabla, es el ID seguido del nombre de la tabla.

**Cod** = código, es el código del ID. Este valor es único para ese ID, no se puede repetir. Luego de la palabra Cod se debe poner el tipo de código al que se refiere (esto está asociado a la tabla). Ej: CodIdioma, es el código único del idioma.

**Desc** = descripción, es la descripción detallada del ID. Es una explicación o la palabra completa del código. Luego de la palabra Desc se debe poner a qie descripción se refiere (esto está asociado a la tabla). Ej: CodIdioma = ita, DescIdioma = italiano.

**Usuarios:**

**GUsuTDatAcceso** Gus01\_UsuAct

|  |  |  |  |
| --- | --- | --- | --- |
| Módulo | Sección | Elemento | Nombre |
| G = General | Usu = Usuario | T = Tabla | DatAcceso |

En esta tabla está la información del usuario y sus claves.

**GUsuTPerfil** Gus03\_Perfil

|  |  |  |  |
| --- | --- | --- | --- |
| Módulo | Sección | Elemento | Nombre |
| G = General | Usu = Usuario | T = Tabla | Perfil |

En esta tabla está la información de los distintos perfiles que tiene el sistema. Cada perfil, tiene distintas clasificaciones, estas son: Sistema al que ingresa, Tipo de Perfil es si para ingreso, proceso, etc., (ej: un usuario puede tener un perfil de ingreso para un proceso y para otro de autorización).

**GUsuTUsuPerfil** Gus02\_UsuPer

|  |  |  |  |
| --- | --- | --- | --- |
| Módulo | Sección | Elemento | Nombre |
| G = General | Usu = Usuario | T = Tabla | UsuPerfil |

En esta tabla se vincula al usuario de la tabla de usuarios GUsuTDatAcceso y la tabla de los perfiles GUsuTPerfil.

Aquí se determina que puede ver y hacer cada usuario.

Un usuario puede tener varios perfiles y un perfil lo pueden tener varios usuarios (ej: perfil del call, pueden tenerlo varias personas.)

**Módulo General**

**Idiomas**

El sistema es multilenguaje, quiere decir que para determinar lo que significa el código de un item, se debe hacer una consulta al procedimiento almacenado GDicPCodxIdio donde está la información de cada código y el significado en el idioma elegido. En caso de que no se elija un idioma, el predeterminado es el español.

Por lo tanto para determinar el contenido de cada tabla, se debe consultar al procedimiento almacenado respectivo. En este procedimiento almacenado, está vinculada la información de la tabla buscada con el GDicPCodxIdio, y de esa relación cada código tiene su descripción en el idioma indicado.

GDicTIdioma: Tabla con la información de los idiomas.

Están los distintos idiomas que maneja el sistema, el ID del idioma, su Código y su Descripción. Esta es la única tabla donde en la misma esta toda la información de código y descripción en el idioma original.

También la misma tiene un campo con el código del estado del idioma.

GDicPIdioma: Procedimiento Almacenado con la información de los idiomas.

En este procedimiento almacenado, está la información de la tabla GDicTIdioma, vinculada con el procedimiento almacenado GDicPCodxIdio, donde está la información de los códigos y su descripción del idioma elegido. Si no se elige idioma va el por defecto, es el español.

Eliminar Tabla idiomas y renombrar y cambiar todas las consultas.

**Idiomas Excepciones**

La información que se guarda en el sistema que no está clasificada por idioma es la ingresada por el usuario. Esto significa que cualquier campo de comentarios, aclaraciones, etc., no se clasifican por idioma.

**Códigos**

Los distintos **códigos** que maneja el sistema son únicos, independiente de la lengua. Eje: el valor ID c6fe2749-0cb8-49cd-97df-03c299c0c6cf, es codigo: HAB, descripción: habilitado, pero depende del idioma el código y su descripción. Podemos concluir, como el código y su descripción varían según el idioma, debemos hacer dos tablas para saber cuál es su descripción de acuerdo al idioma del usuario.

GDicTCodigos: están los códigos únicos de los ítems.

GDicTCodIdiomas: están los códigos únicos, el idioma, el código y su descripción. Esta clave de tres campos hace que un código, para un idioma solo puede tener un único código de hasta 50 caracteres. Así por ejemplo para el idioma español, tiene un código distinto para cada idioma.



Campos a destacar:

Descripcion: varchar(250), es la descripción del ítem.

GDicPCodxIdio: Procedimiento almacenado donde está la descripción de cada código con su idioma.

**Descripciones**

Las distintas **descripciones** que maneja el sistema son únicos, independiente de la lengua. Eje: el valor ID **8dce0539-b6c6-4e4c-b6ff-d52701fb5483**, descripción: **Tipo de Carrocería**, pero depende del idioma su descripción. Podemos concluir, como la descripción varían según el idioma, debemos hacer dos tablas para saber cuál es su descripción de acuerdo al idioma del usuario.

GDicTDescripciones: están los ID únicos de las descripciones.

GDicTDescIdiomas: están los ID únicos y el ID idioma. Esta clave de dos campos hace que un código, para un idioma solo puede tener una descripción única de hasta 250 caracteres.

GDicPDescxIdio: Procedimiento almacenado donde está la descripción de cada código con su idioma. En este procedimiento si no se elige el idioma la información que surge es la del idioma por defecto, el español.

**Módulo Vehículos:**

**Datos de los Vehículos de Fábrica:**

La información de fábrica, se registra tal cual se extrae de la base de datos. Esta información tiene las especificaciones técnicas y los submodelos comerciales. Luego de que se carga un modelo se verifica si la especificación técnica que está en la Placa de PostVenta PPV está registrada en la tabla ATVehDatTecn, si no la registra y si está actualiza el IDvdt de la tabla ATVehBasTerm.

**ATVehBasTerm:** están la información de los vehículos de la terminal.

Clave primaria:

ID: código único del vehículo uniqueidentifier.

VIN: vehícule identification number, es único (Chasis).

Campos a destacar:

IDvdt: es el código ID de la Placa Post Venta PPV.

**ATVehDatTecn:** está la información de la Placa Post Venta y su desagregación.

Clave primaria:

ID: código único ID de la PPV.

Campos a destacar:

IDvdt: es el código ID de la Placa Post Venta PPV.

La PPV es un código multivaluado, por lo tanto luego de la carga se debe discriminar su contenido y obtener la información indicada.

Proceso de busqueda

1.- Primero hago la consulta para obtener el código ID de la pieza o la estructura del arbol

2.- Ya con la pieza debo ver la ubicación en el Arbol de piezas y seleccionar desde el padre hasta los hijos todas las piezas afectadas.

La búsqueda por vehículo, puede generarse de dos formas:

1. De la tabla de **ATVehCodBusq**, donde se puede encontrar con el VIN o el Dominio. De esta búsqueda obtenemos el CBU (código de búsqueda único).
2. De la consulta **APVehCodArbBucle**, donde está la estructura desde la Marca hasta la Codigo de Busqueda Unico (PPV). Cuando el cliente elige una de estas opciones, debe serguir buceando hasta llegar al CBU (código de búsqueda único).

El código de búsqueda único CBU, surge de la Placa de Post Venta, sin algunos puntos que no son relevantes para la búsqueda.

La búsqueda por Pieza, puede generarse desde la

De la tabla de **ATParCodBusq** debo cargar la búsqueda que se hace en el árbol de piezas, En este árbol está la combinación única desde la CBU (código de búsqueda único) hasta las piezas. Es por ello que Yo busco en la tabla de CodBusq y luego con ese ID voy al árbol de producto a encontrar las piezas.

Por lo tanto, el código, deben tener una tabla ATParCodBusq, donde estan las distintas posibilidades de busqueda de un código, esto se debe unir con la tabla ATParCodxIdi, donde esta el nombre original que sería la primer forma de busqueda. De esta forma se encuentra una misma busqueda, si es la pieza o el arbol lo que esta buscando.

**Busqueda por Siñiestro:**

Para realizar una búsqueda por siniestro, primero se debe cargar un árbol que tenga la estructura de los siniestros:

**Zona de impacto**: abarcan las partes en las cuales se presentan los daños producidos en la unidad. Para poder producir la búsqueda es necesario ubicar la zona afectada, para ello disponemos en el gráfico las distintas partes que intervienen (se puede seleccionar una o varias zona simultaneamente).

Lateral Derecho

Delantero Derecho

Trasero Derecho

Trasero Central

Delantero Central

Trasero Izquierdo

Delantero Izquierdo

Lateral Izquierdo

Se debería poder seleccionar las zonas en forma gráfica y multiple, eje Paragolpe capot guardabarros, etc.

Consulta SQL que muestra la zona zona de impacto es: SQL\_ArbolyTablaParPiezas.sql

**Grado de impacto:** intensidad de la colisión, determina el tipo de piezas dañadas, pueden ser uno o varios de los 4 rubros: Carrocería, Electricidad, Tapicería, Mecánica. Si se seleccionan todos afectan a todas las piezas del vehículo.

Una vez seleccionada la Zona y Grado de Impacto, se podrá visualizar el conjunto de partes involucradas accediendo por el botón “Buscar”

**Ilustración de la Búsqueda por Siniestro**: Ilustración de las partes, se muestran todas las áreas que se encuentran involucradas en la colisión.

A partir de ellas localizamos las piezas, todas ellas estarán contempladas en el siniestro. Se puden deseleccionar uno o varios rubros, uno o varias piezas o conjunto de piezas.

Con las teclas zoom se puede avanzar o retroceder en los gráficos de los vehículos.

También se puede hacer estas selecciones del gráfico de arbol.

**NOTAS**

De cada tipo de vehículo debo tener el Despiece de carrocería. De la estructura a detalle de pieza cada modelo tiene una imagen única para la estructura, eso hace que se puedan grabar las coordenadas y automatizar la extracción de información, pero si o si hay que hacer manualmente cada modelo 1 vez. Aquí se hace lo que no es automatizable.

Explicación:

Surge de la tabla GDicTIdioma, filtrando el itemID = Idioma, en el único caso que coinciden es cuando se agrega un nuevo idioma.