MAXICOMERCIO ®

Manual de referencia de la base de datos

MaxiComercio Business Platform

2010

MAXICOMERCIO ERP & POS

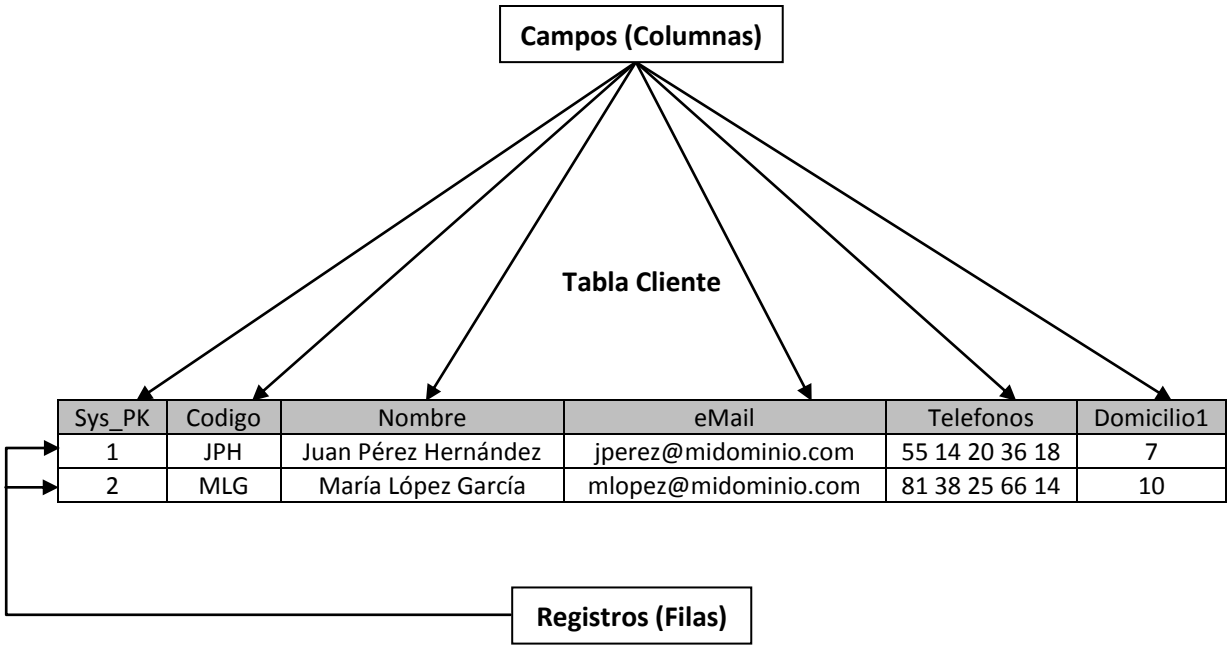
Contenido

La base de datos

Conocimientos básicos acerca de bases de datos relacionales y SQL

Tablas (campos y registros)

Una tabla es un conjunto de datos organizados en filas y columnas, a las filas se les denomina registros y a las columnas campos. Vea el siguiente ejemplo de una tabla.



Sys_PK	Codigo	Nombre	eMail	Telefonos	Domicilio1
1	JPH	Juan Pérez Hernández	jperez@midominio.com	55 14 20 36 18	7
2	MLG	María López García	mlopez@midominio.com	81 38 25 66 14	10

Como se puede observar, en esta tabla *Cliente* se tienen 6 campos y 2 registros.

En la definición de cada campo, debe existir un nombre único, con su tipo de dato correspondiente. Esto es útil a la hora de manejar varios campos en la tabla, ya que cada nombre de campo debe ser distinto entre sí.

A los campos se les puede asignar, además, propiedades especiales que afectan a los registros insertados. El campo puede ser definido como *índice* o *autoincrementable*, lo cual permite que los datos de ese campo cambien solos o sean el principal indicar a la hora de ordenar los datos contenidos.

Cada tabla creada debe tener un nombre único en la Base de Datos, haciéndola accesible mediante su nombre o su seudónimo (Alias) (dependiendo del tipo de base de datos elegida).

Para construir sistemas de información es necesario que exista una manera de identificar de manera única un registro y en consecuencia poder localizarlo fácilmente. Esto se logra mediante

un campo clave, el cual tiene valores únicos. Un ejemplo de campo clave sería el campo Sys_PK del ejemplo anterior.

Ahora bien, para que se pueda crear un verdadero sistema es necesario tener más de una tabla y establecer relaciones entre las mismas de tal forma que permitan almacenar y procesar la información de acuerdo al objetivo de la organización.

Relaciones entre tablas (claves primarias y foráneas)

- **Clave principal**

Uno o más campos cuyo valor o valores identifican de manera única a cada registro de una tabla. En una relación, se usa una clave principal para hacer referencia a registros específicos de una tabla desde otra tabla. Una clave principal se denomina clave externa cuando es referenciada desde otra tabla.

- **Clave foránea (externa)**

Uno o más campos que hacen referencia al campo o campos de la clave principal de otra tabla. Una clave foránea o externa indica como están relacionadas las tablas: los datos en los campos de la clave externa y en la clave principal deben coincidir.

- **Índice**

Una característica del motor de base de datos que acelera la búsqueda y ordenación en tablas, La clave principal de una tabla se indexa automáticamente. Los campos cuyos tipos de datos sean Memo, Vínculo u Objeto OLE no pueden indexarse.

Puede usar la propiedad Indexado para buscar y ordenar registros con un solo campo de una tabla. Este campo puede contener valores únicos o no. Por ejemplo, puede crear un índice con un campo llamado IdEmpleado para una tabla Empleados en la que cada identificador de empleado sea único, o también un índice con el campo Nombre, en el que algunos nombres pueden estar duplicados.

  - **Índice principal**

    Un índice principal contiene uno o más campos que identifican de forma única a todos los registros de la tabla en un orden predefinido. Puesto que el campo índice debe ser único, la propiedad Unique del objeto Index se establece como True. Si el índice principal contiene más de un campo, cada campo puede contener valores repetidos, pero cada combinación de valores de todos los campos indexados debe ser única. Un índice principal contiene una clave para la tabla y normalmente contiene los mismos campos que la clave principal.

  - **Índice candidato**

    Una clave CANDIDATO es una expresión de índice que no permite valores duplicados o NULOS.

Por definición, una clave candidato es un solo campo o una expresión compuesta por definición que cumple los requisitos de un clave principal. Es un requisito de diseño de bases de datos del que tienen uno y sólo uno clave principal las tablas. Sin embargo, una tabla podría contener varios identificadores únicos. Una vez que se haya identificado el clave principal, a otras claves únicas se les denomina claves candidato.

Relaciones entre tablas

Uno de los objetivos de un buen diseño de base de datos es eliminar la redundancia de los datos (datos duplicados). Para lograr dicho objetivo, conviene desglosar los datos en muchas tablas (esto se realiza colocando campos comunes en tablas que están relacionadas) basadas en temas para que cada hecho esté representado sólo una vez.

Tipos de relaciones entre tablas

Existen tres tipos de relaciones de tabla:

- Relación 1 a muchos

Considere una base de datos de seguimiento de pedidos que incluya una tabla Clientes y una tabla Pedidos. Un cliente puede realizar cualquier número de pedidos. Por lo tanto, para cualquier cliente representado en la tabla Clientes puede haber representados muchos pedidos en la tabla Pedidos. Por consiguiente, la relación entre la tabla Clientes y la tabla Pedidos es una relación de uno a varios.

Para representar una relación de uno a varios en el diseño de la base de datos, tome la clave principal del lado "uno" de la relación y agréguela como un campo o campos adicionales a la tabla en el lado "varios" de la relación. En este caso, por ejemplo, agregaría un nuevo campo: (el campo Id. de la tabla Clientes) a la tabla Pedidos y le denominaría Id. de cliente. Access utilizaría entonces el número de identificador del cliente de la tabla Pedidos para localizar el cliente correcto de cada producto.

- Relación muchos a muchos

Considere la relación entre una tabla Productos y una tabla Pedidos. Un solo pedido puede incluir varios productos. Por otro lado, un único producto puede aparecer en muchos pedidos. Por tanto, para cada registro de la tabla Pedidos puede haber varios registros en la tabla Productos. Además, para cada registro de la tabla Productos puede haber varios registros en la tabla Pedidos. Este tipo de relación se denomina relación de varios a varios porque para un producto puede haber varios pedidos, y para un pedido puede haber varios productos. Tenga en cuenta que para detectar las relaciones de varios a varios existentes entre las tablas, es importante que considere ambas partes de la relación.

Para representar una relación de varios a varios, debe crear una tercera tabla, a menudo denominada tabla de unión, que divide la relación de varios a varios en dos relaciones uno a varios. Debe insertar la clave principal de cada una de las dos tablas en la tercera. Como resultado, la tercera tabla registra cada ocurrencia, o instancia, de la relación. Por ejemplo, la

tabla Pedidos y la tabla Productos tienen una relación varios a varios que se define mediante la creación de dos relaciones uno a varios con la tabla Detalles de pedidos. Un pedido puede incluir muchos productos, y cada producto puede aparecer en muchos pedidos.

- Relación uno a uno

En una relación uno a uno, cada registro de la primera tabla sólo puede tener un registro coincidente en la segunda tabla y viceversa. Este tipo de relación no es común porque, muy a menudo, la información relacionada de este modo se almacena en la misma tabla. Puede utilizar la relación uno a uno para dividir una tabla con muchos campos, para aislar parte de una tabla por razones de seguridad o para almacenar información que sólo se aplica a un subconjunto de la tabla principal. Cuando identifique esta relación, ambas tablas deben compartir un campo común.

La instrucción SELECT

La instrucción SELECT se utiliza para recuperar registros de una base de datos en forma de un conjunto de registros, almacenándolos en un nuevo objeto Recordset. Posteriormente, su aplicación puede manipular este Recordset presentando, agregando, cambiando o eliminando registros según sea necesario. Su aplicación también puede presentar y crear informes a partir de los datos.

SELECT suele ser la primera palabra de una instrucción SQL. La mayoría de las instrucciones SQL son SELECT o SELECT...INTO. Las instrucciones SELECT no modifican los datos de la base de datos, sólo los recuperan.

La estructura general de la consulta SELECT es la siguiente:

```
SELECT lista_campos
FROM nombre_tabla
WHERE condiciones_búsqueda
GROUP BY lista_campos
HAVING criterios grupo
ORDER BY lista_campos
```

Ejemplo:

```
SELECT Sys_PK, Nombre FROM Cliente WHERE Nombre LIKE 'A*' ORDER BY Nombre
```

Esta instrucción creará una consulta con las columnas Sys_PK y Nombre con todos los clientes cuyo nombre empiecen con la letra A.

Consultas de unión

Consultas de Unión Internas

- **INNER JOIN**

Las vinculaciones entre tablas se realizan mediante la cláusula INNER que combina registros de dos tablas siempre que haya concordancia de valores en un campo común. Se puede utilizar una operación INNER JOIN en cualquier cláusula FROM. Esto crea una combinación por equivalencia, conocida también como unión interna. Las combinaciones equivalentes son las más comunes; éstas combinan los registros de dos tablas siempre que haya concordancia de valores en un campo común a ambas tablas.

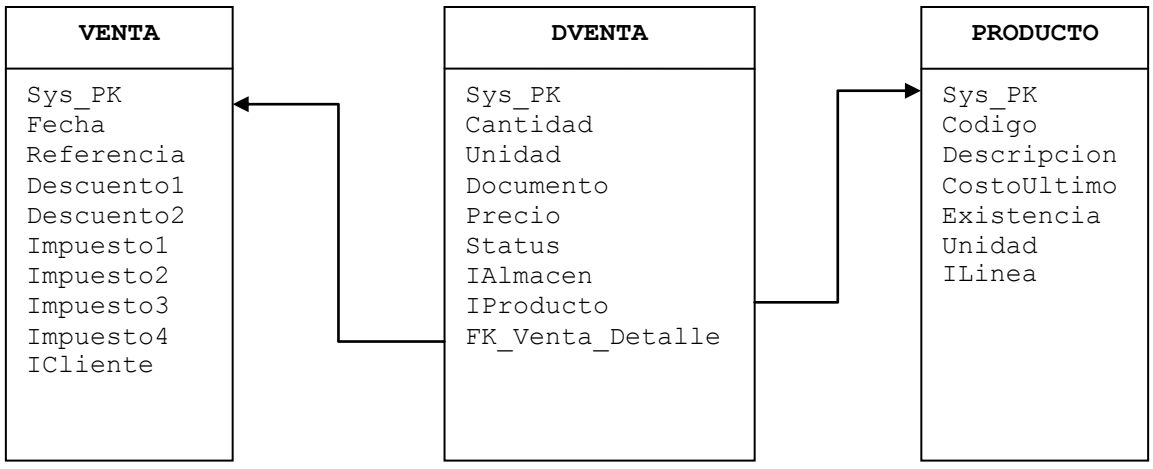
Su sintaxis es:

```
SELECT lista_campos
FROM tablaA
INNER JOIN tablaB
ON tablaA.campox operador tablaB.campoy
```

En donde:

lista_campos	Son los nombres de campos a obtener en la consulta
tablaA, tablaB	Son los nombres de las tablas que se unirán para obtener los registros
tablaA.campox	Es el nombre del campo de la tablaA. Si no son numéricos, los campos deben ser y contener el mismo tipo de datos. No es necesario que se llame igual que el campo de la tablaB con la que se unirá
tablaB.campoy	Es el nombre del campo de la tablaB. Si no son numéricos, los campos deben ser y contener el mismo tipo de datos. No es necesario que se llame igual que el campo de la tablaA con la que se unirá
Operador	Es cualquier operador de comparación relacional: =, <,<>, <=, =>, ó >

Ejemplo:



```
SELECT DVenta.Unidad, Producto.Codigo, Producto.Descripcion, Venta.Fecha FROM Venta INNER JOIN (DVenta INNER JOIN Producto ON DVenta.IProducto=Producto.Sys_PK) ON Venta.Sys_PK=DVenta.FK_Venta_Detalle
```

Consultas de Unión Externas

- **LEFT JOIN Y RIGHT JOIN**

Son otro tipo de unión de tablas, son una extensión del INNER JOIN. Se utilizan en los casos que se requiera que también aparezcan las filas que no tienen una fila coincidente en la otra tabla. La sintaxis es similar que la del INNER JOIN, lo único que cambia es la palabra INNER por LEFT o RIGHT según sea el caso.

LEFT JOIN. Esta operación consiste en añadir al resultado de la unión las filas de la tabla de la izquierda que no tienen correspondencia en la otra tabla, y rellenar en esas filas los campos de la tabla de la derecha con valores nulos.

Ejemplo:

Si se quiere obtener el código, la descripción y la marca de la tabla producto, pero se quiere obtener también aquellos productos que no tiene marca, es decir, que el registro marca venga nulo, la sentencia sería la siguiente:

```
SELECT Producto.Codigo, Producto.Descripcion, Marca.Descripcion FROM Producto LEFT JOIN Marca ON Producto.Imarca=Marca.Sys_PK
```

RIGHT JOIN. Esta operación consiste en añadir al resultado de la unión las filas de la tabla de la derecha que no tienen correspondencia en la otra tabla, y rellenar en esas filas los campos de la tabla de la izquierda con valores nulos.

Ejemplo:

Si se quiere obtener las referencias de los documentos de ventas y el agente (vendedor) que realizó la transacción, pero se quiere obtener también aquellas transacciones que no tengan asociadas un agente, es decir, que el registro agente venga nulo, la sentencia sería la siguiente:

```
SELECT Agente.Nombre, Venta.Referencia FROM Agente RIGHT JOIN Venta ON Agente.Sys_PK=Venta.IAgente
```

- **UNION**

Se utiliza la operación UNION para crear una consulta de unión, combinando los resultados de dos o más consultas “Select” o tablas independientes. Ambas consultas “Select” deben tener el mismo número de campos. Su sintaxis es:

```
SELECT lista_campos
FROM nombre_tabla
UNION ALL
SELECT lista_campos
FROM nombres tablas
```

Ejemplo:

```
SELECT Sys_PK FROM Cliente UNION SELECT ICliente FROM Venta
```

En esta instrucción, si la clave primaria del cliente aparece en las tablas Cliente y Venta, aparecerá solamente una vez en el resultado. Esta operación elimina duplicados.

- **UNION ALL**

La consulta UNION ALL le permite combinar los conjuntos de resultados de 2 o más consultas “Select”. Devuelve todos los registros (incluso si la fila existe en más de uno de las declaraciones “Select”.

Cada instrucción SQL dentro de la consulta UNION ALL debe tener el mismo número de campos en los conjuntos de resultados con similares tipos de datos.

Su sintaxis es:

```
SELECT lista_campos
FROM nombre_tabla
UNION
SELECT lista_campos
FROM nombres tablas
```

Ejemplo:

```
SELECT Sys_PK FROM Cliente UNION ALL SELECT ICliente FROM Venta
```

En esta instrucción, si la clave primaria del cliente aparece en las tablas Cliente y Venta, aparecerá múltiples veces en el resultado. Esta operación no elimina duplicados.

Funciones SQL estándar y de agrupación (SUM, AVG, COUNT, MIN, MAX)

Las funciones de agrupación se usan dentro de una cláusula SELECT y operan sobre conjuntos de filas para dar un resultado por grupo y devolver un único valor que se aplica a un grupo de registros. Las funciones de agrupación ignoran los valores NULL en las consultas.

La siguiente tabla enumera las funciones agregadas.

Función agregada	Descripción
SUM	Utilizada para devolver la suma de todos los valores de un determinado campo
AVG	Utilizada para calcular el promedio de los valores de un determinado campo
COUNT	Utilizada para devolver el número de registros de la selección
MIN	Utilizada para devolver el valor más bajo de un campo especificado
MAX	Utilizada para devolver el valor más alto de un campo especificado

- **GROUP BY**

Mediante la cláusula GROUP BY se dividen las filas que devuelve una consulta en pequeños grupos. Cuando se utilice hay que tener en cuenta que todas las columnas que se pongan en el SELECT que no estén en el GROUP BY, han de estar en una función de agrupación.

Patrones de diseño

Los objetos de la base de datos de MaxiComercio/Déminus 2008/ERA siguen alguno de tres patrones de diseño específicos que permiten un mejor control, mantenimiento y optimización.

Patrones para tablas

- Entidad. Contiene datos que corresponden con algún objeto (cliente, factura, cheque, etc.)
- Asociación. Mantienen la información de una asociación muchos a muchos entre dos asociaciones. Por ejemplo la asociación entre grupos de productos y productos.
- Constante. Contienen valores que son usados a la manera de constantes por el sistema.

Ejemplos de tablas y su correspondiente patrón de diseño

Tabla	Patrón	Descripción
Cliente	Entidad	Contiene el catálogo de clientes
Cliente_GruposClientes_	Asociación	Mantiene la información de la asociación entre los clientes y los grupos a los que pertenecen
cDocumentos	Constante	Contiene una lista de los tipos de documentos que maneja el sistema

Tablas Entidad

Campos comunes

Campo	Tipo	Descripción
Sys_PK	Entero	Clave primaria para la tabla (alcance local)
Sys_GUID	Texto(32)	GUID que puede usarse como clave primaria de alcance global
Sys_DTCreated	Fecha/Hora	Fecha y hora de creación del registro
Sys_TimeStamp	Fecha/Hora	Fecha y hora de la última actualización del registro (marca de tiempo con precisión hasta segundos)
Sys_Exported	Cierto/falso	Indica si el registro ha sido exportado por las utilidades de transporte de datos
Sys_DTExported	Fecha/Hora	Indica la fecha en que se exportó por última vez

Clave primaria

Siempre tiene el nombre Sys_PK y se trata de un entero (de 32 bits o 64 bits dependiendo la plataforma y la configuración para el tipo SQL INT). Tiene los atributos incremental, único y sin permisión de valores nulos. Indexado ascendentemente.

Claves foráneas

Todas las claves foráneas son de tipo Entero (SQL INT) y contienen el valor del campo Sys_PK de la tabla a la que apuntan. Generalmente inician con la letra "I" seguida del nombre de la tabla. Por ejemplo, en la tabla "Venta" existe un campo clave foránea para el cliente que tiene el nombre "ICliente".

Durante la generación de la base de datos el software de instalación de MaxiComercio/Déminus crea restricciones o relaciones dependientes del soporte de la plataforma para mantener la integridad referencial.

GUID's

¿Qué es una GUID?

Es la abreviatura de Global Unique Identifier (Identificador Global Único) y constituye un valor numérico extremadamente alto (32 dígitos hexadecimales) generado por un algoritmo computacional que prácticamente asegura que es irrepetible independientemente del lugar y momento en que se cree.

¿Para qué una GUID en cada tabla?

Las claves primarias usadas en todas las tablas son un entero auto-incremental generado por el motor de la base de datos, esto garantiza el mejor rendimiento del sistema y obedece a las mejores prácticas de diseño. Este valor es irrepetible en la misma tabla y base de datos.

No obstante, un cliente "X" en la tabla "Cliente" de la base de datos "A" puede tener la misma clave primaria (valor para Sys_PK) que un cliente "Y" en la tabla "Cliente" pero en la base de datos "B". Esta situación no reviste ningún problema salvo si quisiéramos comprobar que el cliente "X" de la base de datos "A" no existe también en la base de datos "B". (O si quisiéramos copiarlo).

Es decir, cada registro solo es único en la tabla y base de datos que lo contiene (alcance local de la clave primaria).

Aunque la mayoría de las entidades por su naturaleza implementan códigos alfanuméricos (el campo Código de la tabla Cliente por ejemplo), no existe la garantía de que en otras bases de datos no se repitan.

Esto se soluciona con la implementación de GUIDs que son únicas universalmente.

¿Por qué una GUID de 32 caracteres?

La mayoría de los motores de bases de datos implementan el tipo GUID, que está optimizado para contener este tipo de valores, sin embargo utilizarlo cuando esté disponible supondría la limitante de usar algún determinado motor en particular.

Por esta razón se implementa el campo GUID como texto (generalmente Varchar(32).

Tablas Entidad Definidas por el Usuario (TEDeU)

El asistente de creación de tablas que se encuentra en el "Panel de control de MaxiComercio/Déminus" permite extender la estructura de la base de datos para soportar nuevas entidades que requiera la organización.

Genera los mismos campos comunes y sigue las convenciones anteriores con las siguientes adiciones:

- Antepone el prefijo "ut_" (User Table) a los nombres asignados por el usuario.
- Antepone el prefijo "uf_" (User Field) a los nombres de los campos definidos por el usuario.
- Solo implementa restricciones de valor único y no vacío para los campos definidos por el usuario.

De acuerdo a lo anterior, si el usuario define una nueva tabla "Cobrador", el sistema generará "ut_Cobrador" y si se definió el campo "Nombre", realmente tendrá la columna "uf_Nombre". Quedando una consulta SQL correcta con la siguiente sintaxis: **SELECT Sys_PK, uf_Nombre FROM ut_Cobrador**

Tablas Asociación

Campos comunes

Campo	Tipo	Descripción
Sys_PK	Entero	Clave primaria para la tabla (alcance local)
Sys_TimeStamp	Fecha/Hora	Fecha y hora de la última actualización del registro (marca de tiempo con precisión hasta segundos)

Claves foráneas

Las tablas asociación contendrán siempre dos campos clave foránea de tipo Entero.

Adicionalmente es posible que contengan otros campos no indexados con atributos de la asociación dependiendo del diseño o implementación de la base de datos.

Generalmente estas tablas tienen el nombre con la siguiente sintaxis:

> *[TablaEntidad1]_[TablaEntidad2]_*
>
> *Donde [TablaEntidad1] y [TablaEntidad2] son los nombres de las tablas cuya asociación representa.*
>
> *Ejemplo: Cliente_GrupoClientes_*

Tablas Constante

Solo contienen dos campos:

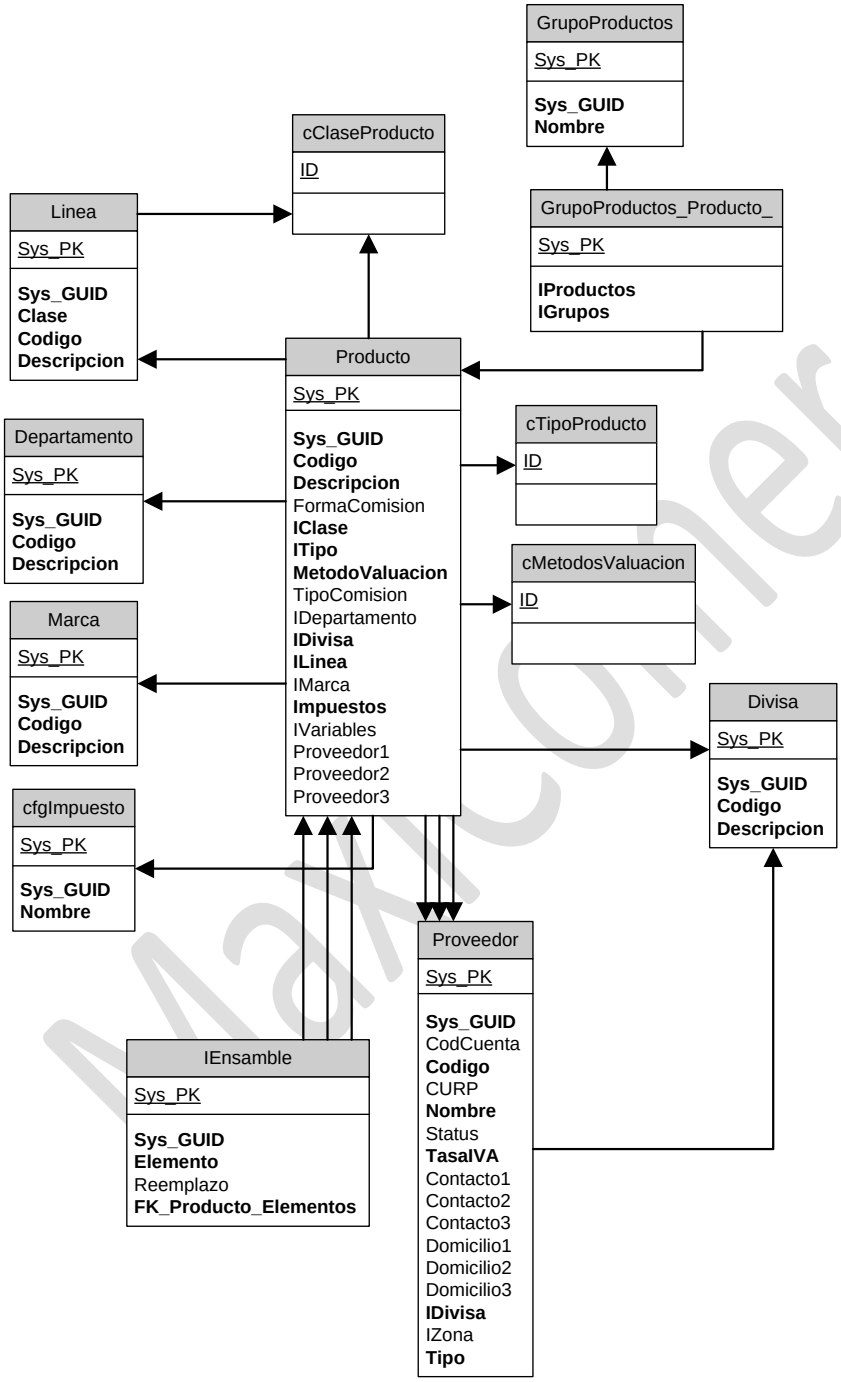
- ID. Entero único (clave primaria) aunque no es auto-incremental.
- Const. Texto (32 caracteres) que contiene un texto descriptivo para el valor ID

Generalmente este tipo de tablas inician con el prefijo 'c'. *Ejemplo "cDocumentos"*

Modelo relacional

Inventario y productos

Catálogo de productos



Acerca de la configuración de impuestos a los productos

La configuración de impuestos se encuentra en la tabla cfgImpuesto.

Se admiten 4 impuestos a la compra y 4 impuestos a la venta.

Al mismo tiempo, cada impuesto puede tener dos tasas de cálculo.

Ejemplo:

> ***En México, existe el Impuesto al Valor Agregado (IVA) con una tasa del 16% sobre el precio de venta. No obstante esta tasa cambia en las denominadas zonas fronterizas a solo el 11%.***
>
> ***Una empresa ubicada en alguna zona fronteriza debe cobrar solo el 11% de IVA, sin embargo en el caso de que tenga proveedores en el interior del país pagará 16%.***
>
> ***En el mismo sentido, si una empresa se encuentra en el interior del país y su proveedor en zona fronteriza, pagará el 11% de IVA aunque deberá cobrar el 16%.***

Los impuestos 1 y 2 se calculan en cascada, es decir que la base de cálculo para el impuesto 2 es la suma del impuesto 1 y el subtotal. Los impuestos 3 y 4 se calculan directamente de la base.

> ***Ejemplo:***
>
> ***Impuesto1=10%, Impuesto2=10%, Impuesto3=10%, Impuesto4=10%***
>
> ***Subtotal: 100***
>
> ***Impuesto1=Subtotal * 10%=10***
>
> ***Impuesto2=(Subtotal+Impuesto1)*10%=11***
>
> ***Impuesto3=Subtotal*10%=10***
>
> ***Impuesto4=Subtotal*10%=10***
>
> ***100+10+11+10+10=141***

Campos de la tabla cfgImpuesto

I1Compra, I2Compra, I3Compra I4Compra. Son los campos con los impuestos a la compra.

I1Venta, I2Venta, I3Venta, I4Venta. Son los campos con los impuestos a la venta.

Estos campos son de texto y tienen el siguiente formato: %Tasa|%Tasa.

Donde Tasa es el porcentaje de impuesto a aplicar.

El valor a la izquierda del Pipe | corresponde al interior y el de la derecha a la frontera.

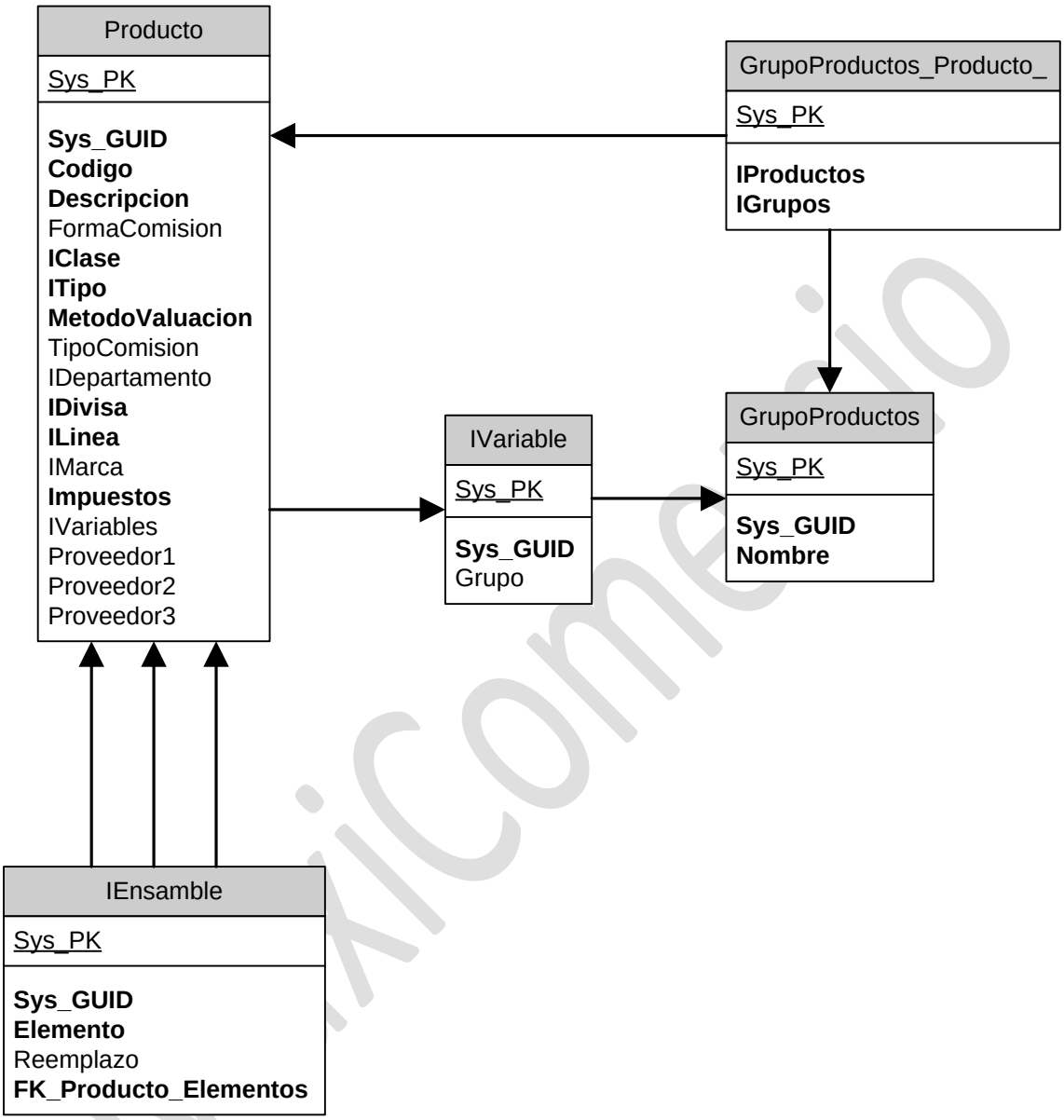
Determinación de la tasa en tiempo de ejecución

La determinación de que tasa de impuesto (frontera o interior) aplicar corresponde en el caso de las ventas a un valor de la configuración local del equipo.

En el caso de las compras, cada proveedor tiene definido un campo que permite indicar si las operaciones realizadas con el usaran el impuesto de frontera.

Este campo es TasaIVA de la tabla Proveedor.

Ensambles y componentes variables



Los componentes variables son usados para las recetas variables de alimentos o bebidas por ejemplo.

Acerca de los ensambles

Un ensamble es un producto que contiene la especificación de sus componentes (otros productos).

La tabla IEnsamble contiene la especificación de cada componente.

Existen 3 claves foráneas importantes en la tabla IEnsamble:

- FK_Producto_Elementos. Contiene el valor de la clave primaria (Sys_PK) del ensamble (producto) al cual pertenece el componente.
- Elemento. Contiene el valor de la clave primaria del producto considerado como componente.
- Reemplazo. Contiene la clave primaria del producto que podría usarse como reemplazo del componente especificado en el campo “Elemento”.

Otros campos de la tabla IEnsamble:

- Cantidad. Indica la cantidad requerida en unidad estándar del componente para el ensamble.

Campos usados para órdenes de producción:

- Omitible. Es un valor booleano que permite establecer si el componente puede omitirse.
- CantPoco. Indica la cantidad a utilizar del componente en caso de hacer la especificación “poco”, este valor es útil en el caso de recetas de alimentos o bebidas por ejemplo.
- CantMucho. Indica la cantidad a utilizar del componente en caso de hacer la especificación “mucho”, este valor es útil en el caso de recetas de alimentos o bebidas por ejemplo.
- CantDemasiado. Indica la cantidad a utilizar del componente en caso de hacer la especificación “demasiado”, este valor es útil en el caso de recetas de alimentos o bebidas por ejemplo.

Componentes variables

Un componente variable está definido por un “grupo de productos” del cual se puede elegir cierta cantidad de elementos (productos) para usarlos en la producción de otro.

Esta función es muy utilizada para la elaboración de alimentos o bebidas u órdenes de producción bajo especificaciones del consumidor.

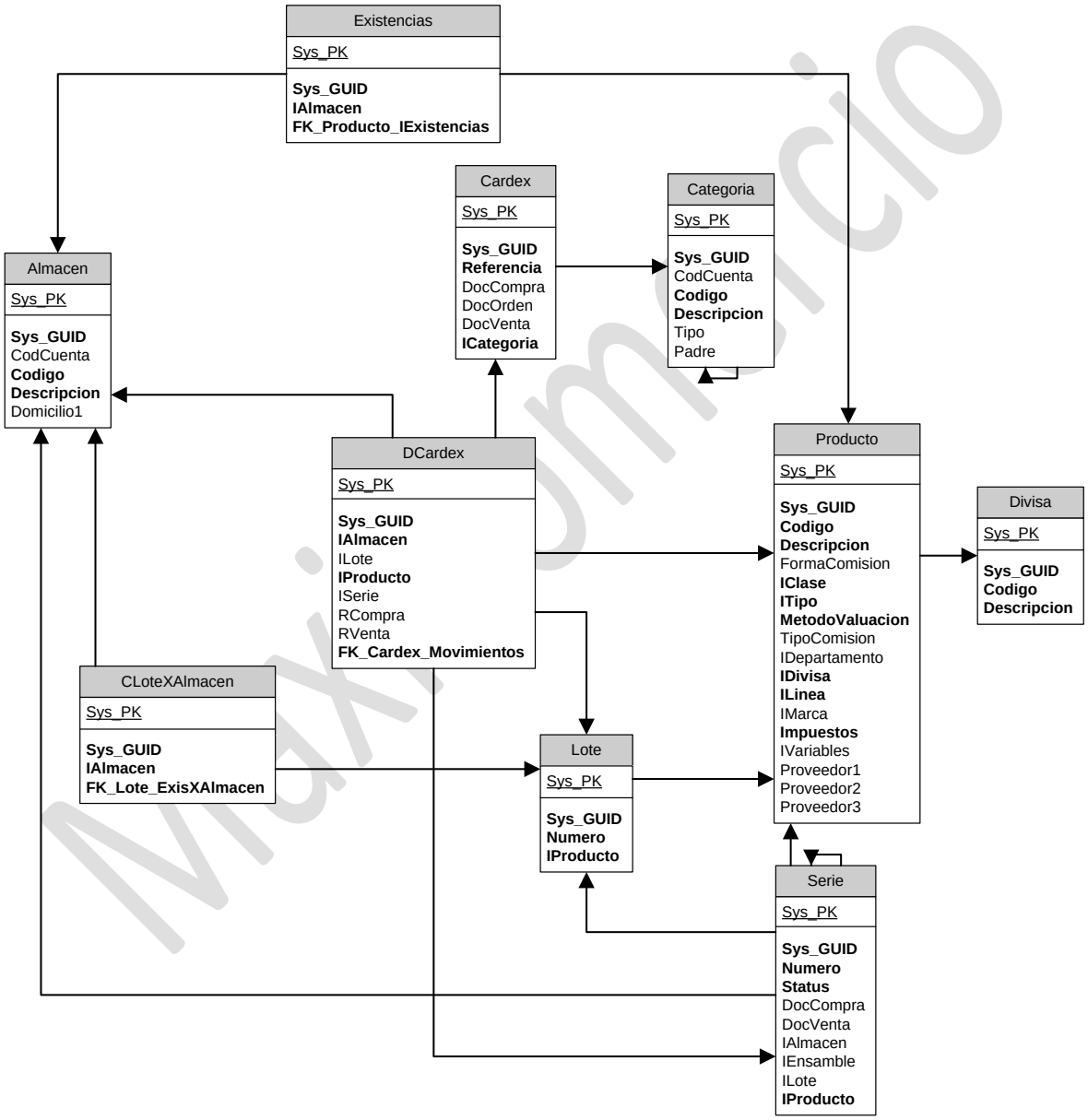
Campos clave foránea de la tabla IVariable:

- FK_Producto_IVariables. Contiene el valor de la clave primaria (Sys_PK) del producto a producir.
- Grupo. Identifica al “Grupo de productos” (Sys_PK de la tabla GrupoProductos) que puede usarse como componente.

Campos de especificación de cantidades:

- Minimo. Establece la cantidad mínima de elementos del grupo de productos que se pueden elegir.
- Maximo. Establece la cantidad máxima de elementos del grupo de productos que se pueden elegir.
- Factor. Indica un factor multiplicativo para considerar la cantidad a emplear en función de la cantidad de elementos seleccionada.

Existencias



Movimientos al inventario

La existencia de un producto se establece como la sumatoria de todas sus entradas menos la sumatoria de todas sus salidas.

La existencia está expresada y valuada en función de la Unidad estándar del producto.

Las entradas/salidas se registran mediante las tablas “Cardex” y “DCardex” que representan un Movimiento de almacén.

La tabla “Cardex” contiene los datos generales del movimiento, por su parte la tabla relacionada DCardex contiene cada partida del movimiento (la entrada o salida de un producto).

El campo FK_Cardex_Movimientos de la tabla DCardex es una clave foránea que contiene el valor de la clave primaria del movimiento en la tabla Cardex al que pertenece la partida. (Maestro/detalle).

Consular Existencias y saldos

Aunque la existencia y el saldo pueden calcularse, por razones de rendimiento se acumulan en las tablas Existencias y Producto.

La tabla Existencias contiene el saldo y existencia de cada producto en cada almacén. Por su parte en la tabla Producto se guarda en cada registro el saldo y existencia de todos los almacenes de ese producto.

Si requiere conocer la existencia o saldo total al momento puede consultar directamente los campos Existencia y Saldo de la tabla Producto.

Para conocer la existencia al momento de un almacén en particular debe consultar la tabla Existencias identificando el producto y almacén en cuestión.

Si requiere conocer existencias o saldos en distintos momentos en el tiempo deberá realizar la sumatoria de entradas-salidas o cargos-abonos de la tabla DCardex hasta una fecha dada.

ADVERTENCIA: NO MODIFIQUE DIRECTAMENTE LOS VALORES ACUMULADOS DE EXISTENCIAS Y SALDOS, SOLO ÚSELOS PARA CONSULTAS.

> ***Nota: La integridad de los datos acumulados está protegida mediante procedimientos almacenados, disparadores y transacciones***
>
> ***(ACID compliant http://es.wikipedia.org/wiki/ACID).***

Valuación del inventario

La valuación del inventario por los métodos de costo promedio e identificado se realiza directamente en la tabla DCardex calculando el valor para el campo “Abonos”.

En el caso de los métodos PEPS y UEPS se utiliza el campo “Resto” de la tabla “Cardex” para mantener la cantidad restante de unidades de un movimiento de entrada en específico y adicionalmente la tabla DCapa describe como se han tomado los costos provenientes de distintas entradas para una salida en particular.

En la tabla DCapa el campo FK_DCardex_Capas contiene el valor de clave primaria del registro de la tabla DCardex que contiene la salida para la que fue necesario usar unidades de distintas entradas.

El campo “PKEntrada” contiene el valor de clave primaria del registro de la tabla DCardex que contiene la entrada de la que se tomaron unidades.

La tabla DCapa contiene también la cantidad de la entrada tomada y el costo correspondiente.

Aplicar, valuar y completar movimientos

Un movimiento es un conjunto de operaciones (entrada/salida) de mercancías, al aplicarse mueve las existencias (cantidades disponibles) del producto, al valuarse se calculan los costos de las salidas.

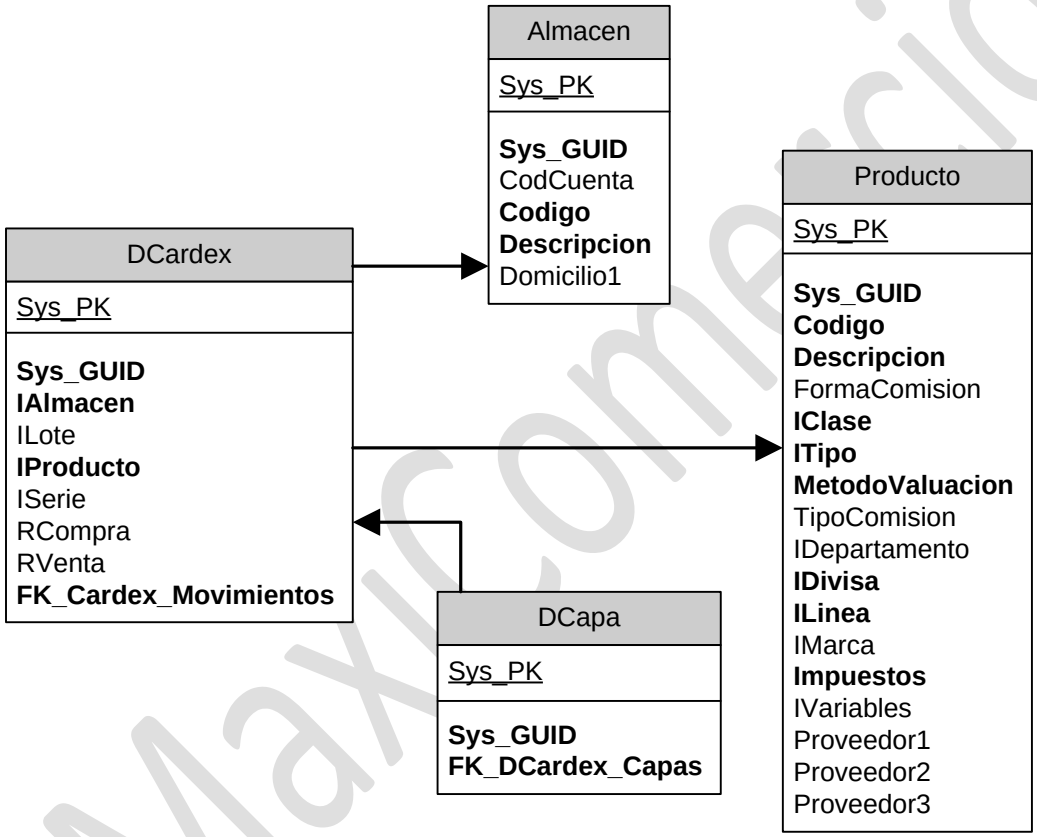
Lo ideal es completar el movimiento, esto es aplicar y valuar todas las operaciones, esta debe ser la configuración predeterminada de la aplicación, sin embargo en la práctica puede ser necesario desacoplar este proceso.

Reglas de desacoplamiento del movimiento

Una operación de entrada debe ser siempre completada, es una operación acoplada, es decir solo se podrá aplicar si se puede valuar, lo cual implica que se especifican números de serie y lote de ser necesarios.

Las operaciones de salida pueden desacoplarse es decir pueden aplicarse y luego valuarse, para que se valúen es necesario especificar números de serie y lote si el producto los requiere.

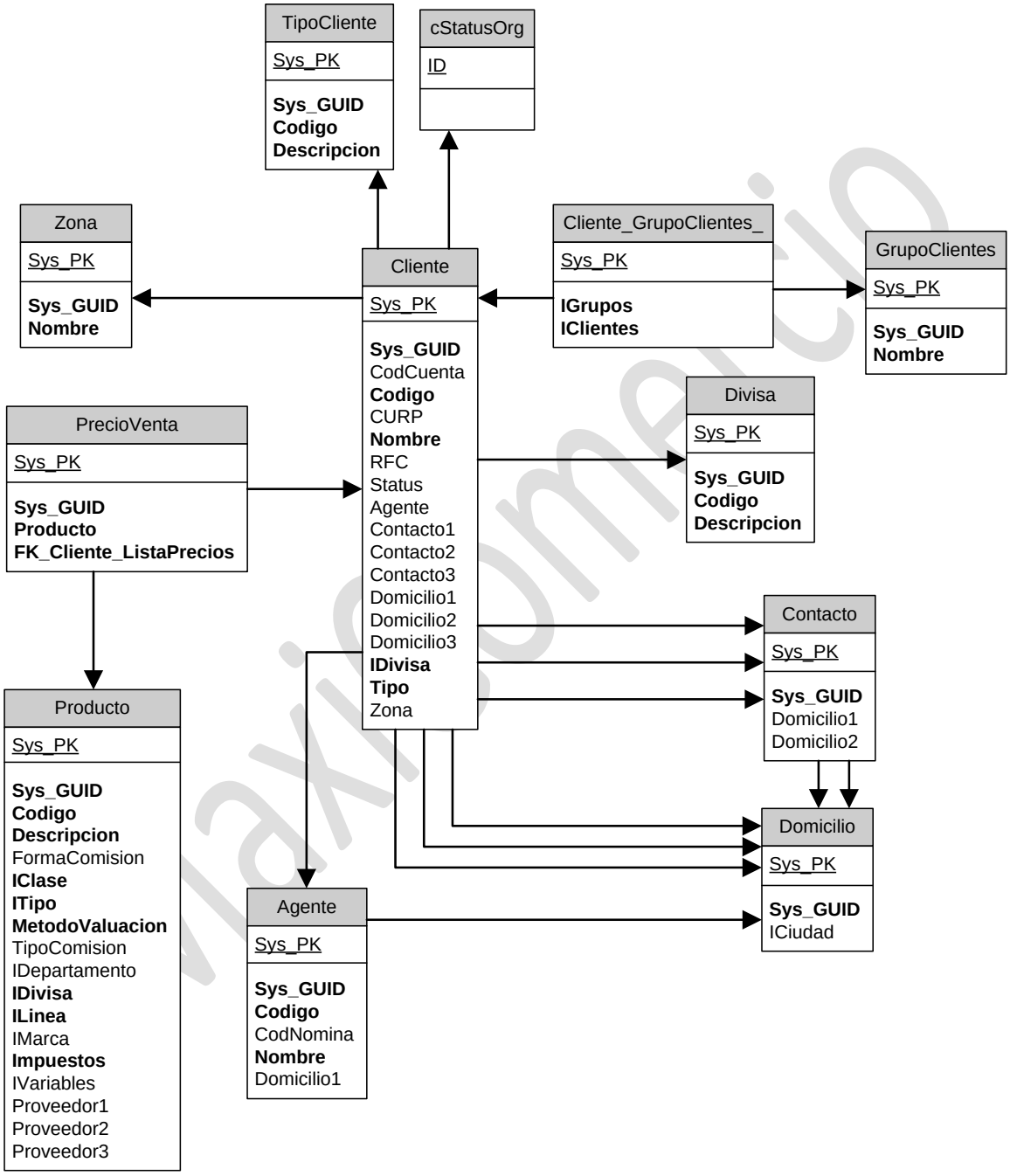
Un movimiento puede ser desacoplado si solo implica salidas, caso contrario solo puede estar no aplicado y aplicado y valuado. Al no ser aplicado no se mueve las existencias.



Observe que cada registro de DCardex puede aplicar a un almacén distinto aunque pertenezcan a un mismo movimiento de almacén (registro de la tabla Cardex).

Clientes y cuentas por cobrar

Catálogo de clientes



Observaciones generales a la estructura del catálogo de clientes

- Los domicilios se almacenan en la tabla Domicilio existiendo solamente la referencia en la tabla Cliente mediante los campos Domicilio1,Domicilio2 y Domicilio3.

- Las listas de precios exclusivas para cada cliente utilizan la tabla PrecioVenta que está relacionada con “Producto”. Aquí se establece un precio en especial directo o en función de un límite (volumen de venta).

Monedero electrónico y puntos

Los saldos del monedero electrónico se acumulan en el campo Monedero de la tabla Cliente.

El saldo en puntos del cliente se mantiene en el campo SaldoPuntos de la tabla Cliente.

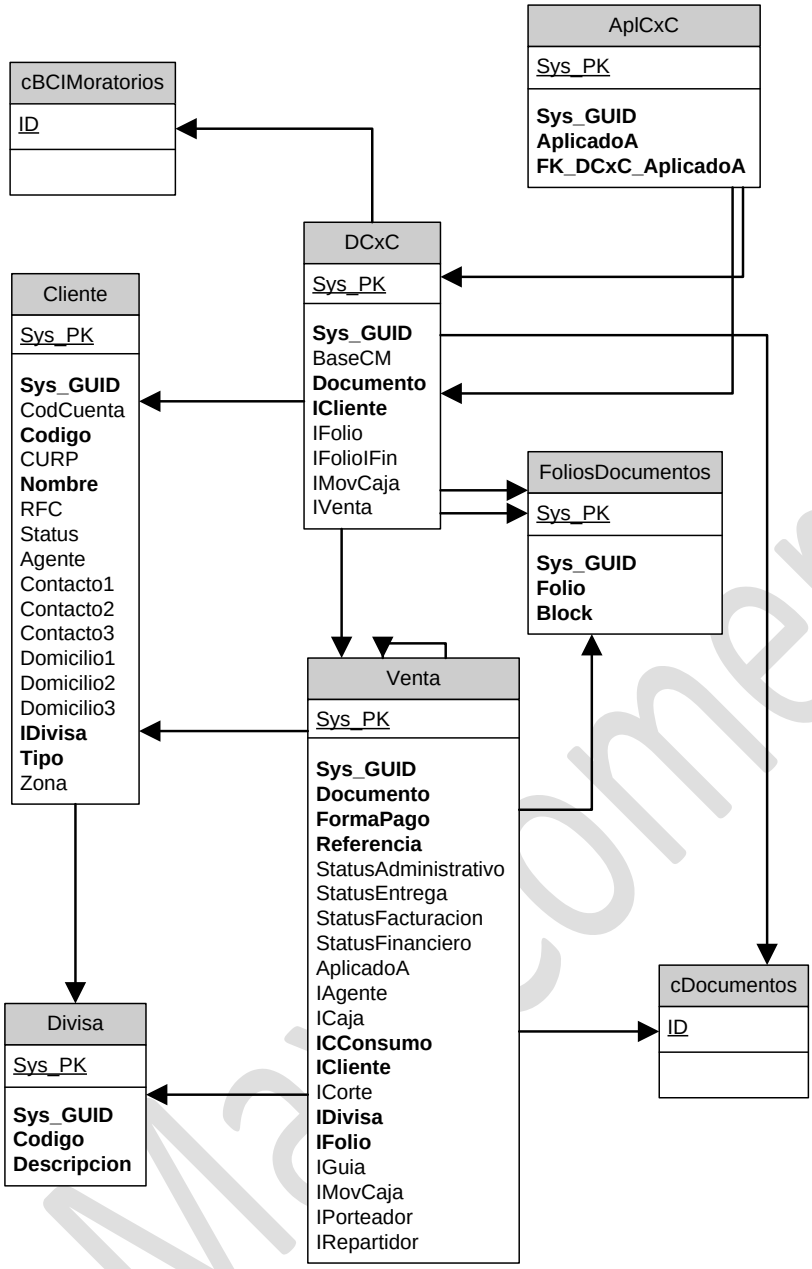
El campo Puntos de la tabla Cliente mantiene un acumulado de todos los puntos que ha sumado el cliente por sus compras.

Se mantiene una bitácora de los puntos y/o dinero obtenido en la tabla ut_BitPromociones que relaciona la venta, los importes y el cliente.

El registro de la utilización de los puntos y el dinero del monedero se mantiene en la tabla DVales relacionada a la tabla MovCaja que contiene el registro de operaciones de efectivo en cajas.

> ***Importante. No modifique directamente los valores acumulados de los campos Puntos, SaldoPuntos y Monedero de la tabla Cliente***

Cuentas por cobrar



Saldos y movimientos a cuentas por cobrar

La tabla DCxC contiene los movimientos (debe y haber) a la cuenta de un cliente.

Nota. Las cuentas de clientes pertenecen al Activo por lo que son deudoras.

El saldo por cobrar a un cliente se puede determinar por la sumatoria del campo “Debe” menos la sumatoria del campo “Haber” de la tabla DCxC.

Por cuestiones de rendimiento el saldo total se mantiene en el campo “Saldo” de la tabla Cliente.

Para consultar el saldo total de un cliente use el campo Saldo de la tabla Cliente.

Para consultar el saldo vencido o por vencer realice las sumatorias correspondientes en la tabla DCxC filtrando por el campo “Aplicación” que contiene la fecha de vencimiento/aplicación del movimiento.

Intereses, pagos y bonificaciones a cargos

Estos datos están descritos en la tabla DCxC por los siguientes campos y se usan en registros donde el valor de debe es mayor que cero (cargos):

- Pagos. El importe total de pagos que se han aplicado al cargo
- Bonificaciones. El importe total de bonificaciones aplicadas al cargo
- BaseCM. Apunta a la tabla de constantes cBCIMoratorios para indicar la base de cálculo para intereses moratorios para este cargo.
- IntFinancieros. Contiene el importe de intereses financieros para el cargo. Este importe junto al capital representan el valor del campo Debe.
- IntMoratorios. El importe de intereses moratorios que se han generado para el cargo, su valor se corresponde con el de otro registro de cargo que se agrega a la tabla cuando se realiza el cálculo.
- PorImpuestoCap. Es el porcentaje de impuestos correspondiente al capital
- PorImpuestoFin. Es el porcentaje de impuestos correspondiente a los intereses financieros
- PorImpuestoMor. Es el porcentaje de impuestos correspondiente a los intereses moratorios.
- TasaMoratorios. Indica el porcentaje de intereses moratorios para el cargo.

Detalle de aplicación de pagos/bonificaciones a cargos específicos

La identificación de los pagos o bonificaciones aplicadas a cargos específicos se encuentra registrada en la tabla AplCxC.

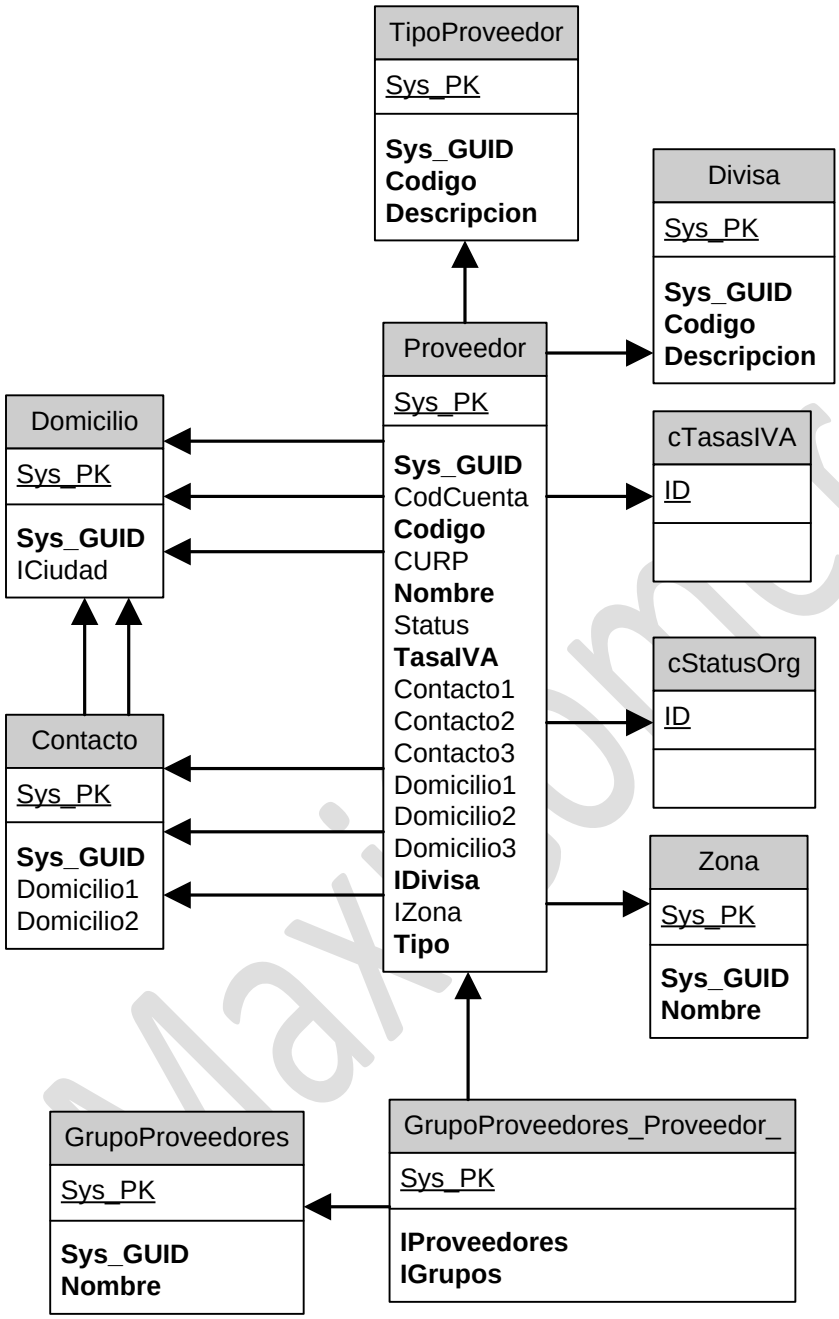
Claves foráneas de AplCxC:

- FK_DCxC_AplicadoA contiene el valor de la clave primaria del registro en DCxC que corresponde a un pago o bonificación
- AplicadoA contiene el valor de clave primaria del registro en DCxC que corresponde al cargo al que se aplica el pago o bonificación

En la tabla DCxC el campo XAplicar indica la cantidad restante por aplicar de un abono.

Proveedores y cuentas por pagar

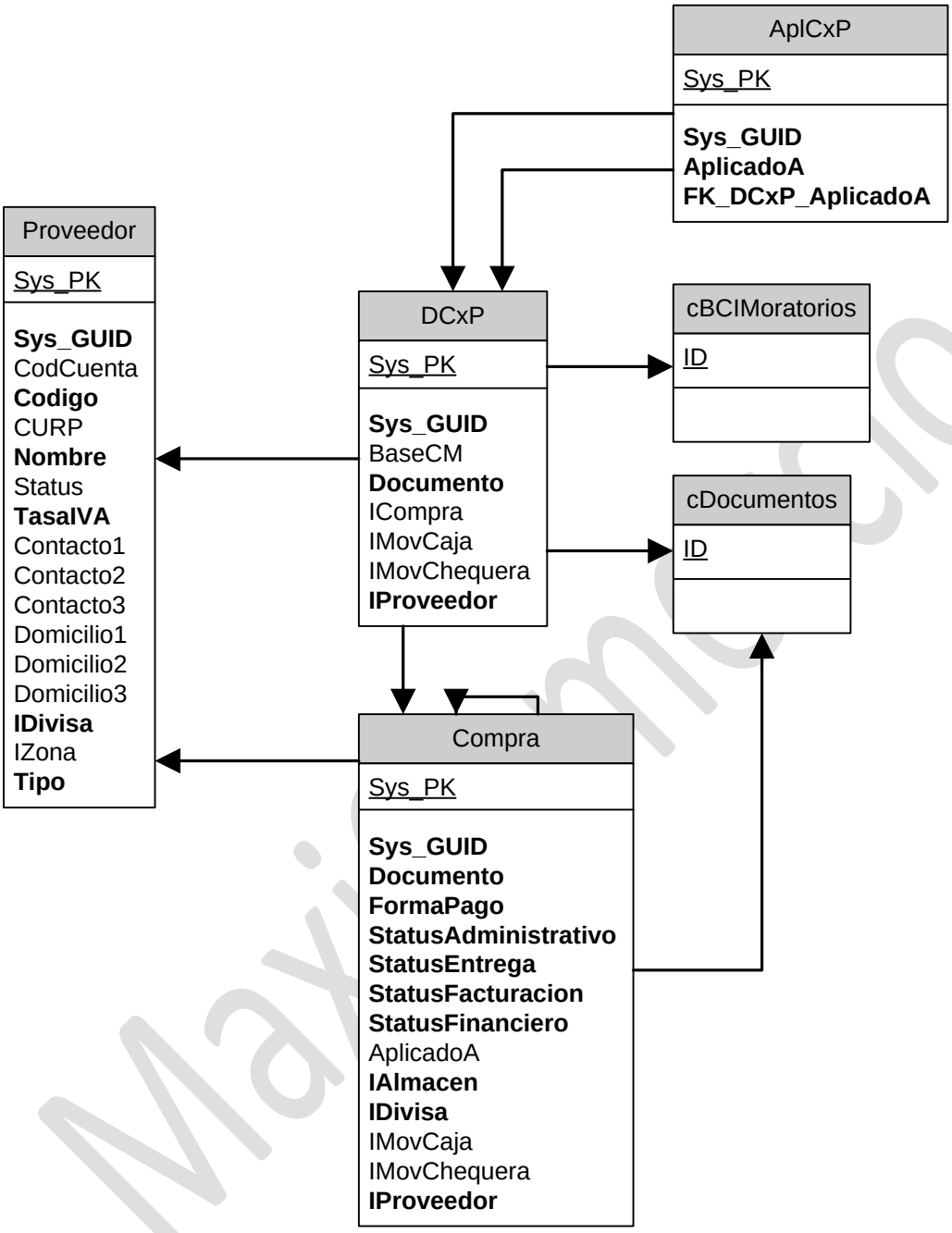
Catálogo de proveedores



Observaciones:

- La tasa de IVA (México) a usar (frontera o interior) se define por el campo TasaIVA de la tabla Proveedor relacionado con cTasasIVA

Cuentas por pagar



Saldos y movimientos a cuentas por pagar

La tabla DCxP contiene los movimientos (debe y haber) a la cuenta de un proveedor.

Nota. Las cuentas de clientes pertenecen al Pasivo por lo que son acreedoras.

El saldo por pagar a un proveedor se puede determinar por la sumatoria del campo “Haber” menos la sumatoria del campo “Debe” de la tabla DCxP.

Por cuestiones de rendimiento el saldo total se mantiene en el campo “Saldo” de la tabla Proveedor.

Para consultar el saldo total de un proveedor use el campo Saldo de la tabla Proveedor.

Para consultar el saldo vencido o por vencer realice las sumatorias correspondientes en la tabla DCxP filtrando por el campo “Aplicación” que contiene la fecha de vencimiento/aplicación del movimiento.

Intereses, pagos y bonificaciones

Estos datos están descritos en la tabla DCxP por los siguientes campos y se usan en registros donde el valor de Haber es mayor que cero:

- Pagos. El importe total de pagos que se han aplicado
- Bonificaciones. El importe total de bonificaciones aplicadas
- BaseCM. Apunta a la tabla de constantes cBCIMoratorios para indicar la base de cálculo para intereses moratorios.
- IntFinancieros. Contiene el importe de intereses financieros. Este importe junto al capital representan el valor del campo Haber.
- IntMoratorios. El importe de intereses moratorios que se han generado, su valor se corresponde con el de otro registro del Haber que se agrega a la tabla cuando se realiza el cálculo.
- PorImpuestoCap. Es el porcentaje de impuestos correspondiente al capital
- PorImpuestoFin. Es el porcentaje de impuestos correspondiente a los intereses financieros
- PorImpuestoMor. Es el porcentaje de impuestos correspondiente a los intereses moratorios.
- TasaMoratorios. Indica el porcentaje de intereses moratorios.

Detalle de aplicación de pagos/bonificaciones

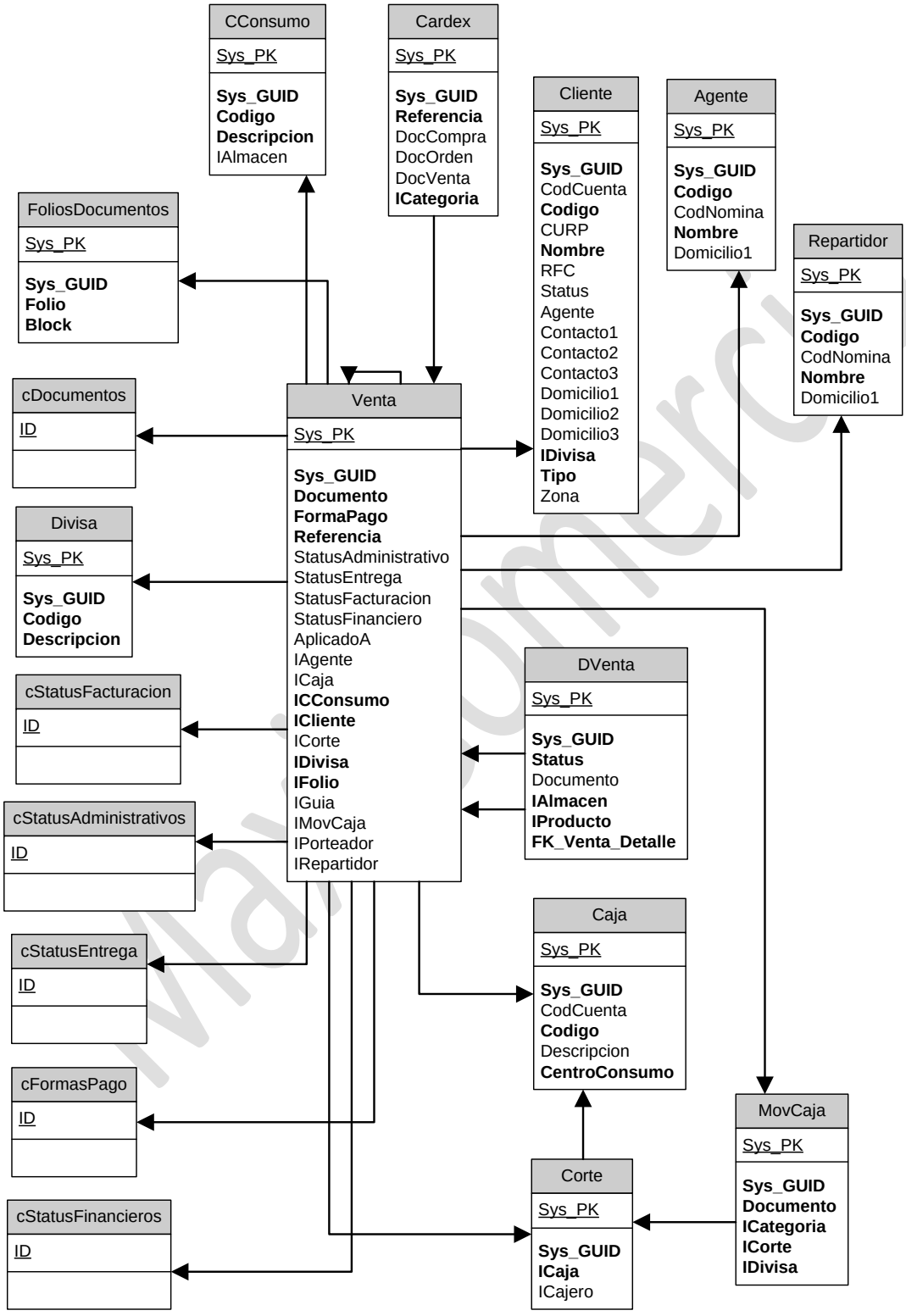
La identificación de los pagos o bonificaciones aplicadas se encuentra registrada en la tabla AplCxP.

Claves foráneas de AplCxP:

- FK_DCxP_AplicadoA contiene el valor de la clave primaria del registro en DCxP que corresponde a un pago o bonificación
- AplicadoA contiene el valor de clave primaria del registro en DCxP que corresponde al movimiento al que se aplica el pago o bonificación

En la tabla DCxP el campo XAplicar indica la cantidad restante por aplicar.

Ventas



Acerca de la estructura de Ventas

La tabla Venta contiene los documentos relacionados con la venta: Cotizaciones, Pedidos, Remisiones, Tickets, Facturas y Notas de crédito.

Esta tabla mantiene el detalle del documento (partidas) mediante la tabla DVenta.

Si la venta se cobró en efectivo mantiene una relación con MovCaja.

Si la venta se realizó en el punto de venta se guarda relación con la tabla Caja y Corte.

Cuando una venta afecta al inventario el registro de la tabla Cardex que le corresponde apunta a la tabla Venta a través del campo Cardex.DocVenta.

Cambios en los campos que controlan estados en las tablas Venta y DVenta

Cotización

	Inicial	**Pedido**	**Remitida/facturada**
Venta.AplicadoA	Nada	**Una referencia al documento en que se incluyó**	
Venta.StatusFinanciero	0-No aplica	**0-No aplica**	
Venta.StatusAdministrativo	2-Cerrado	**3-Procesado**	
Venta.StatusEntrega	0-No aplica	**0-No aplica**	
Venta.StatusFacturacion	0-No aplica		**3-Facturado**
DVenta.XFacturar	**0**		
DVenta.Status	**0-No aplica**		
DVenta.Documento	**Nada**		

Pedido

	Inicial	**Remitido**	**Facturado**	**Cancelado**
Venta.AplicadoA	Nada	Una referencia		**Nada**
Venta.StatusFinanciero	**0-No aplica**			
Venta.StatusAdministrativo	2-Cerrado	3-Procesado		**99-Cancelado**
Venta.StatusEntrega	1-Por entregar	3-Entregado		**0-No aplica**
Venta.StatusFacturacion	0-No aplica		3-Facturado	**0-No aplica**
DVenta.XFacturar	0	0 si se entregó completo >0 es la cantidad pendiente por entregar		**0**
DVenta.Status	1- Por entregar	2-Entregado de ser así 0-No aplica si no se entregó o si no se entregó completo		**0-No aplica**
DVenta.Documento	**Nada**	**Si se entregó (Status=2) Nada** **Si Status=0 Referencia del nuevo pedido si se generó**		**Nada**

Remisión

	Inicial	**Facturada**	**Cancelada**

Venta.AplicadoA	Nada		Referencia a la primer factura,	**Nada**
Venta.StatusFinanciero	No aplica	1-Con adeudo 2-Saldado	2-Saldado (La factura adquirirá el adeudo si existe)	**0-No aplica**
		3-Consignado		
Venta.StatusAdministrativo	1-Abierto 2-Cerrado	3-Procesado		**99-Cancelado**
Venta.StatusEntrega	No aplica	Entregado		**0-No aplica**
Venta.StatusFacturacion	No aplica	1-Por facturar	2-Parcialmente facturado 3-Facturado	**0-No aplica**
DVenta.XFacturar	0	=Cantidad	=Pendiente de facturar	**0**
DVenta.Status	No aplica	Entregado	2-Entregado 3-Facturado	**0-No aplica**
DVenta.Documento	**Nada**		**Factura**	**Nada**

Ticket

	Inicial		**Facturado**	**Cancelado**
Venta.AplicadoA	Nada		Referencia a la primer factura,	**Nada**
Venta.StatusFinanciero	No aplica	1-Con adeudo 2-Saldado	2-Saldado (La factura adquirirá el adeudo si existe)	**0-No aplica**
Venta.StatusAdministrativo	1-Abierto 2-Cerrado	3-Procesado		**99-Cancelado**
Venta.StatusEntrega	No aplica	Entregado		**0-No aplica**
Venta.StatusFacturacion	No aplica	1-Por facturar	2-Parcialmente facturado 3-Facturado	**0-No aplica**
DVenta.XFacturar	0	=Cantidad	=Pendiente de facturar	**0**
DVenta.Status	No aplica	Entregado	2-Entregado 3-Facturado	**0-No aplica**
DVenta.Documento	**Nada**		**Factura**	**Nada**

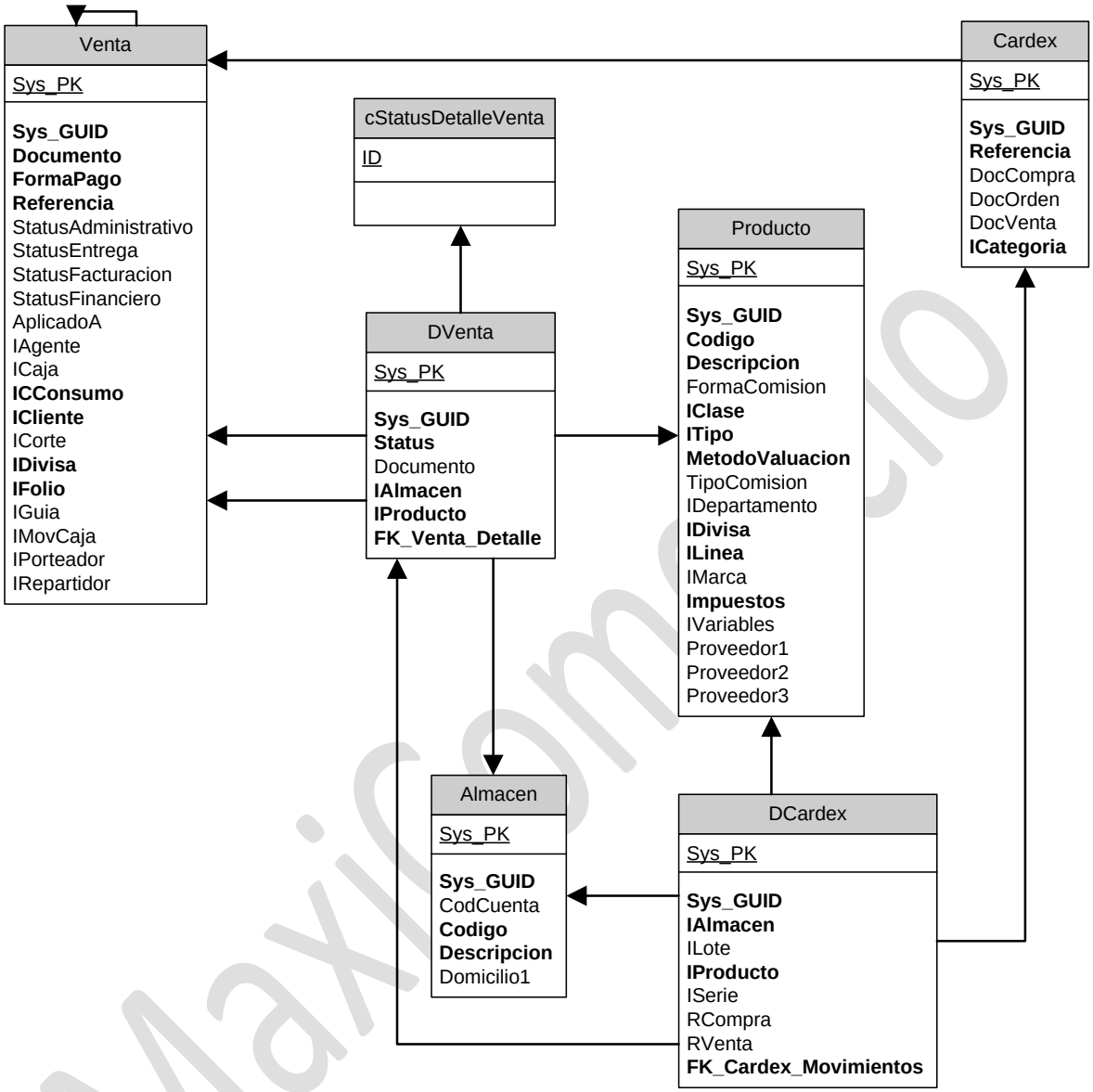
Factura

	Inicial		Cancelada
Venta.AplicadoA	Nada		**Nada**
Venta.StatusFinanciero	No aplica	1-Con adeudo 2-Saldado	**No aplica**
Venta.StatusAdministrativo	1-Abierto 2-Cerrado	3-procesado	**99-cancelada**
Venta.StatusEntrega	No aplica	Entregado	**No aplica**
Venta.StatusFacturacion	No aplica		**No aplica**
DVenta.XFacturar	0		**0**
DVenta.Status	Facturado		**No aplica**
DVenta.Documento	**Nada**		**Nada**

Notas de crédito

	Inicial	Cancelado
Venta.AplicadoA	Nada	**nada**
Venta.StatusFinanciero	2-Saldado	**0-No aplica**
Venta.StatusAdministrativo	3-Procesado	**99-cancelado**
Venta.StatusEntrega	No aplica	**No aplica**
Venta.StatusFacturacion	No aplica	**No aplica**
DVenta.XFacturar	0	**0**
DVenta.Status	No aplica	**No aplica**
DVenta.Documento	**Nada**	**Nada**

Detalle de cada partida de una venta



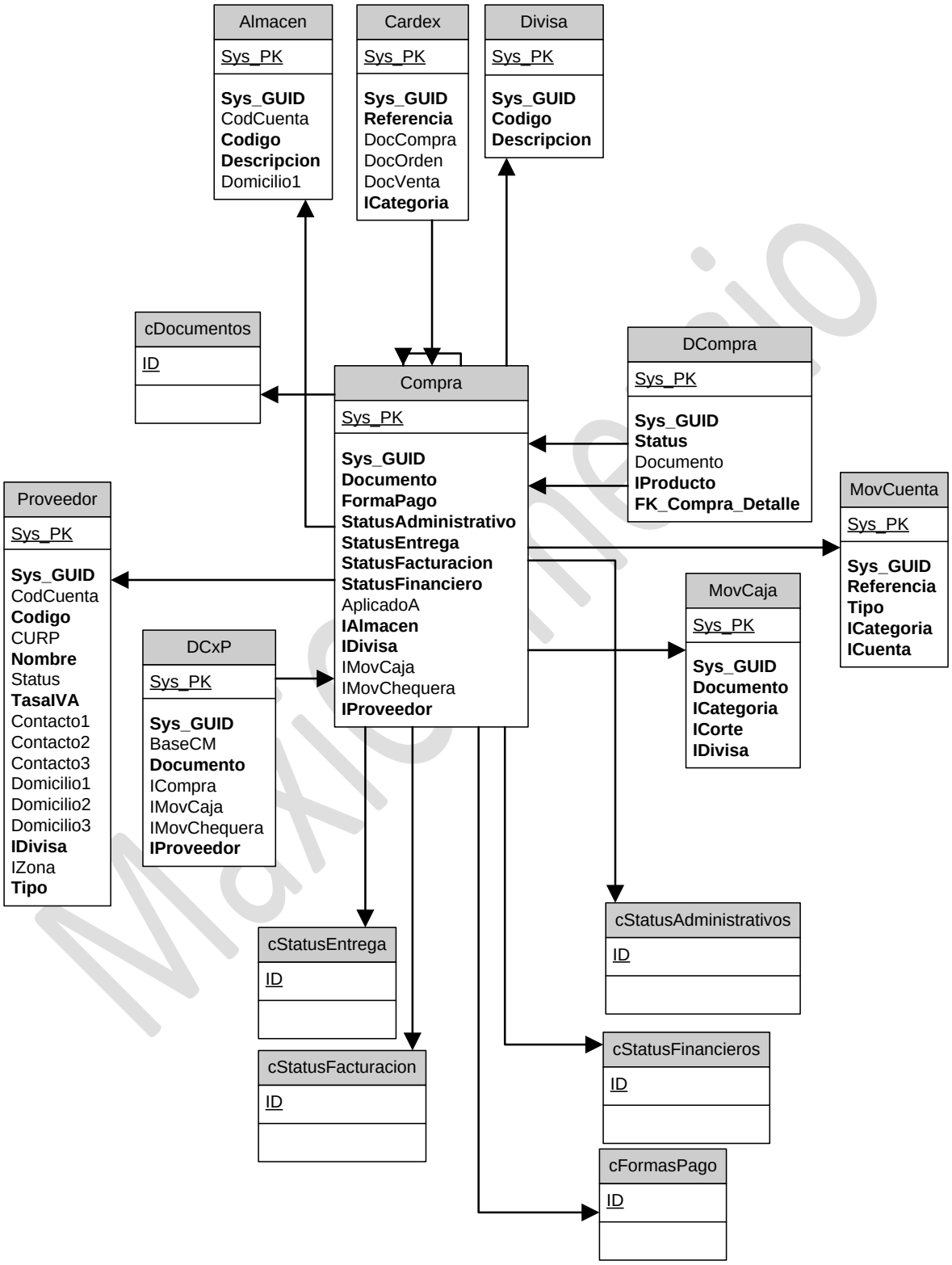
La tabla DVenta mantiene el detalle de las partidas de un documento de venta, existe un estatus de entrega cuyos valores provienen de la tabla cStatusDetalleVenta.

Una partida puede provenir de otro documento (por ejemplo una factura de una cotización), en ese caso la relación con el documento origen se establece a través del campo DVenta.Documento.

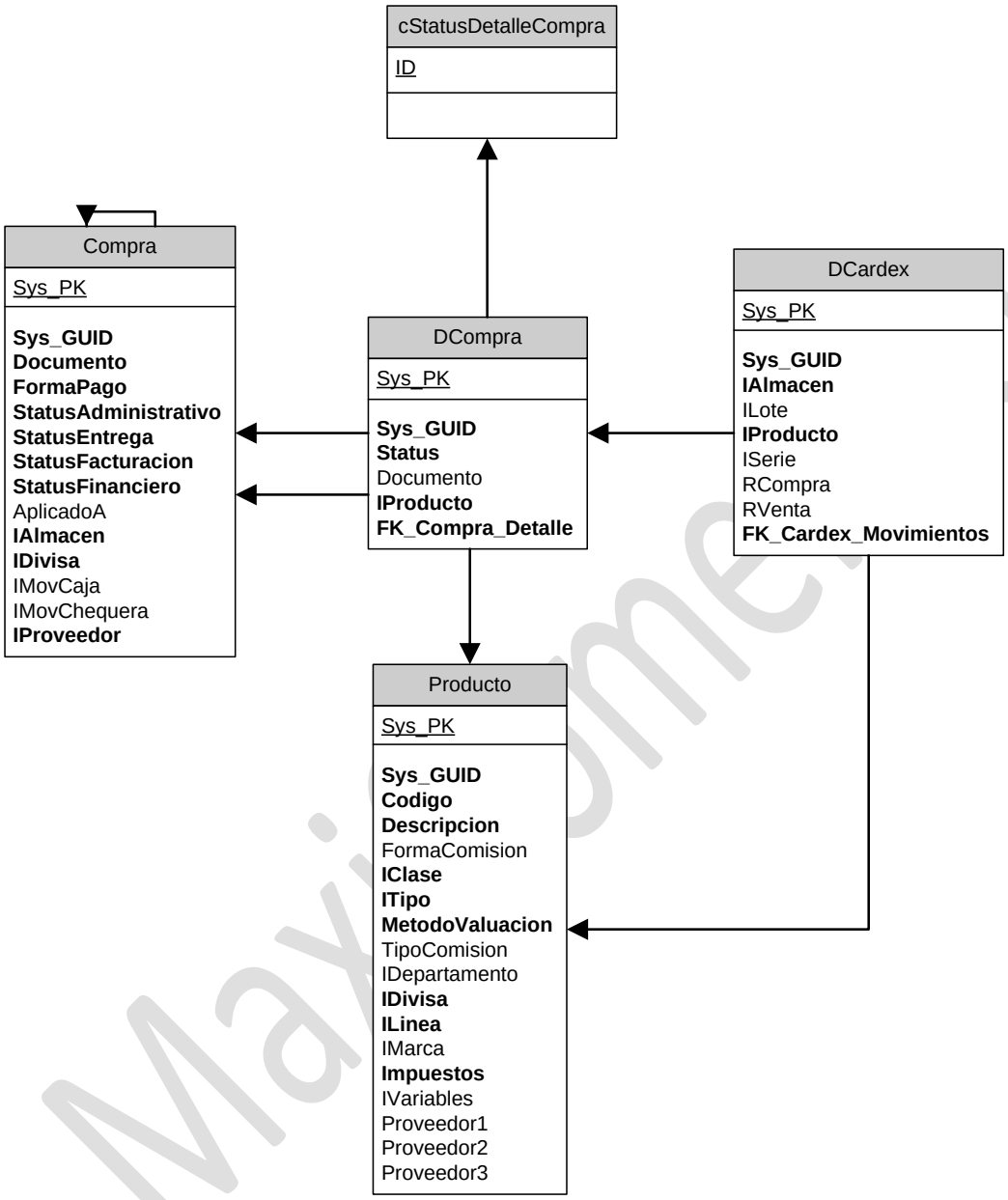
La clave foránea que apunta al encabezado de la venta (Sys_PK de Venta) es el campo FK_Venta_Detalle.

Si se han procesado salidas de inventario el campo RVenta de DCardex contiene la clave primaria de DVenta (Sys_PK).

Compras



Detalle de cada partida de compra



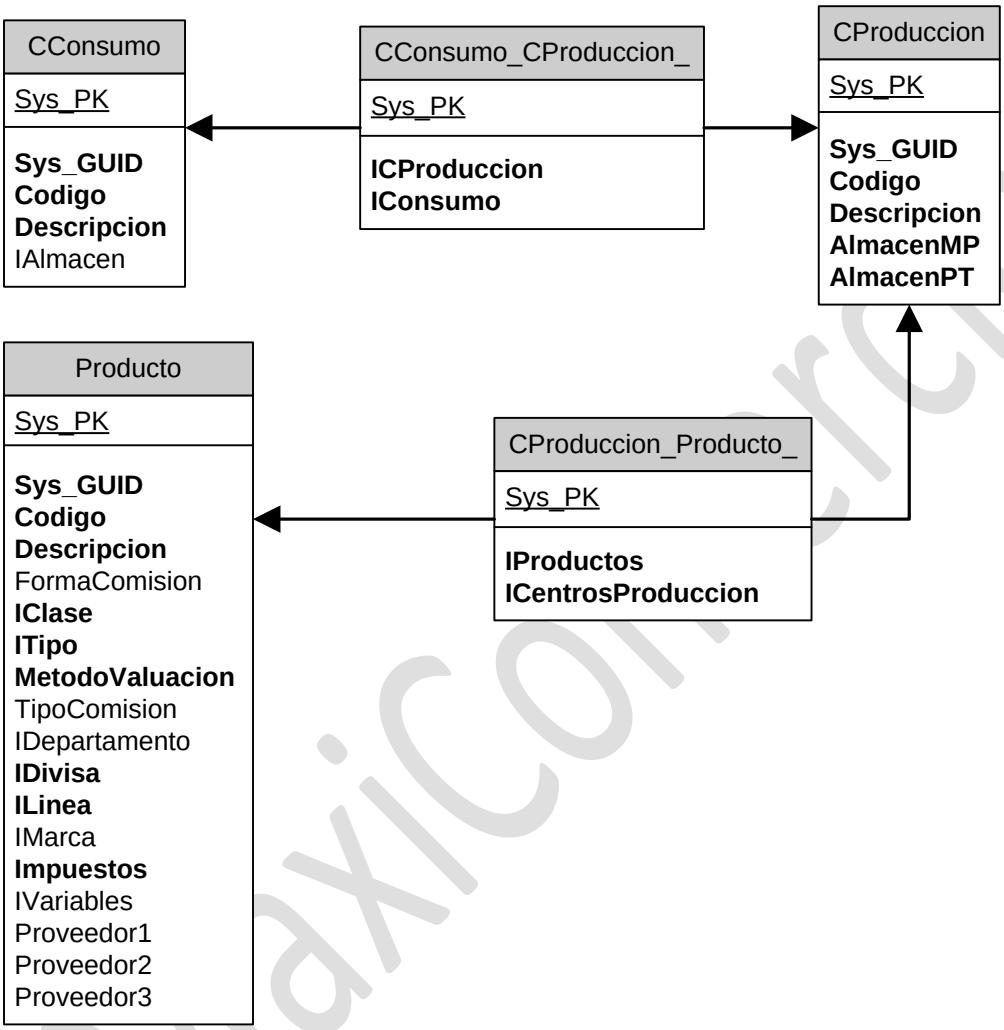
Los documentos comerciales de compras (pedido, remisión, factura y nota de crédito) se mantienen mediante las tablas Compra y DCompra.

Existe una relación con MovCaja y MovCuenta (movimientos de la chequera) dependiendo si el documento se pagó con efectivo de una caja o un cheque.

Los estados y relaciones con Cardex y DCardex son similares a los de ventas.

Control de producción

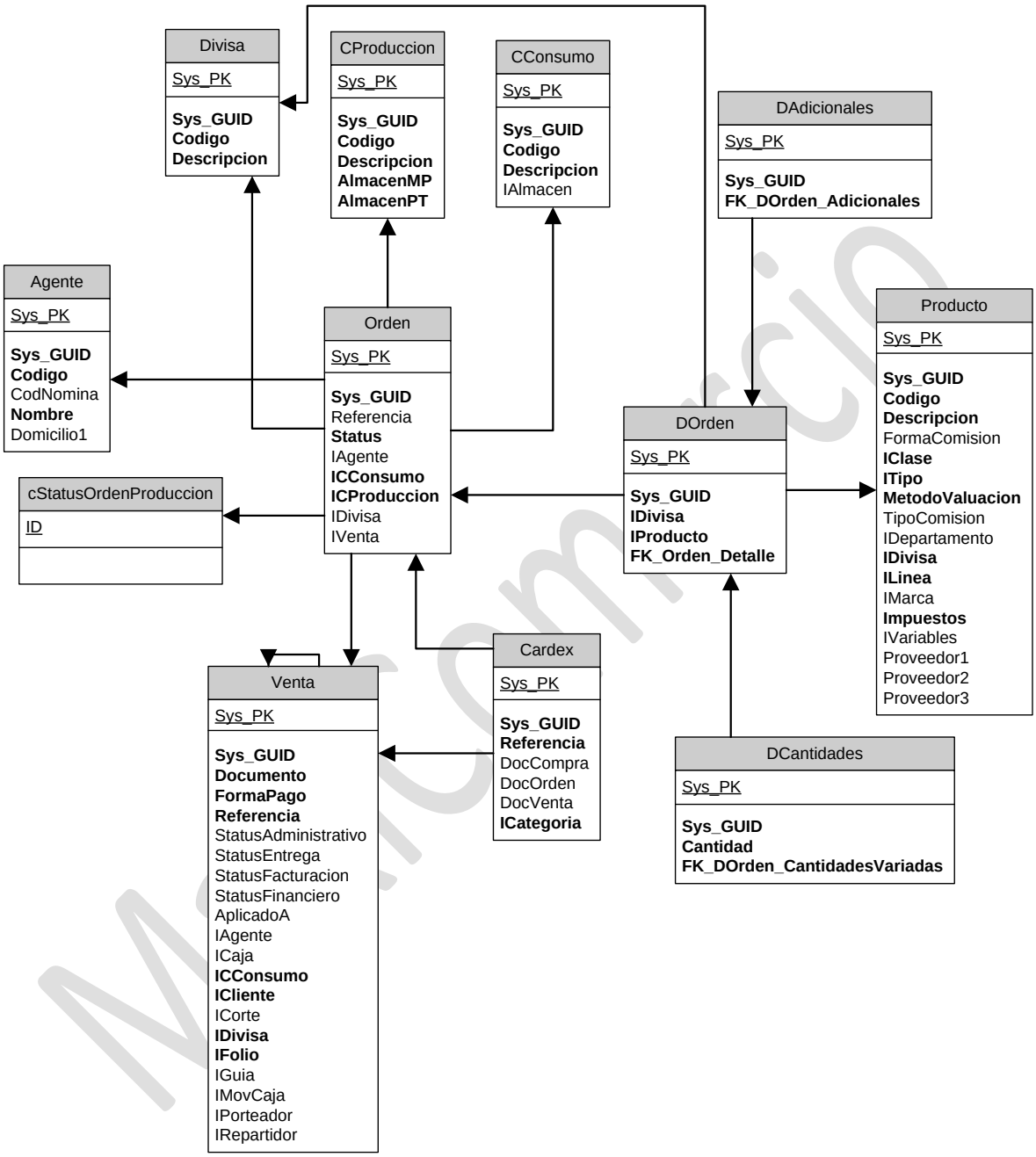
Relación entre productos, centros de producción y centros de consumo



Distintos centros de consumo pueden utilizar distintos centros de producción, del mismo modo un producto puede producirse en diversos centros de producción.

Observe el diagrama para entender esta relación que controla el aprovisionamiento de materiales para la producción.

Órdenes de producción

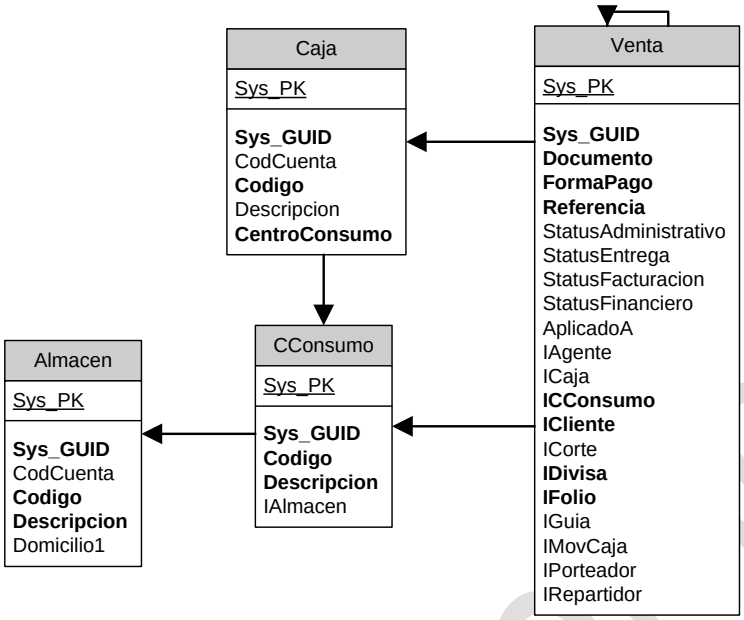


Las órdenes de producción se mantienen mediante las tablas Orden y DOrden, dependiendo el estao (producida, cancelada o por producir) se establecen relaciones con la tabla Cardex y en caso de atender al requerimiento de una venta también se establece la relación con Venta.

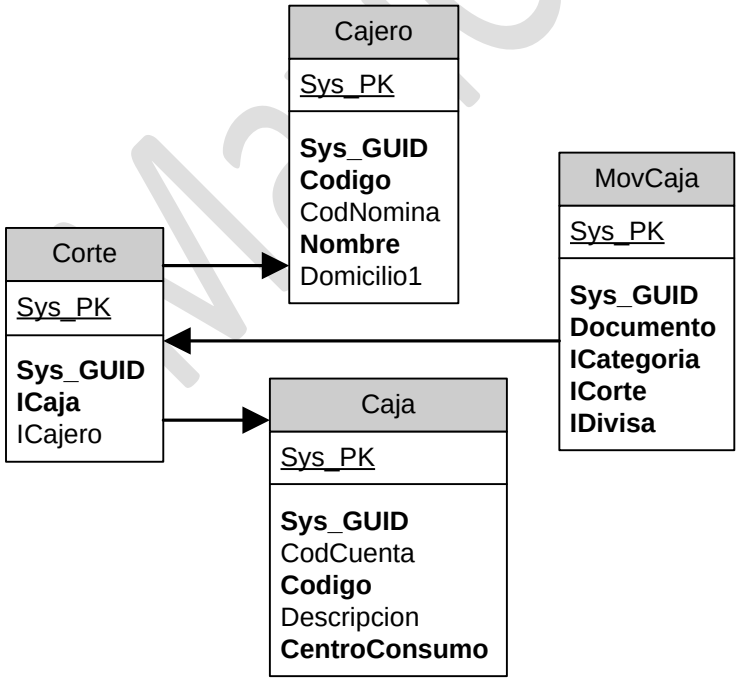
Por su parte, el detalle de la orden en la tabla DOrden utiliza las tablas DCantidades y DAdicionales para cantidades distintas a la especificación del producto (receta) o componentes adicionales.

Cajas y efectivo

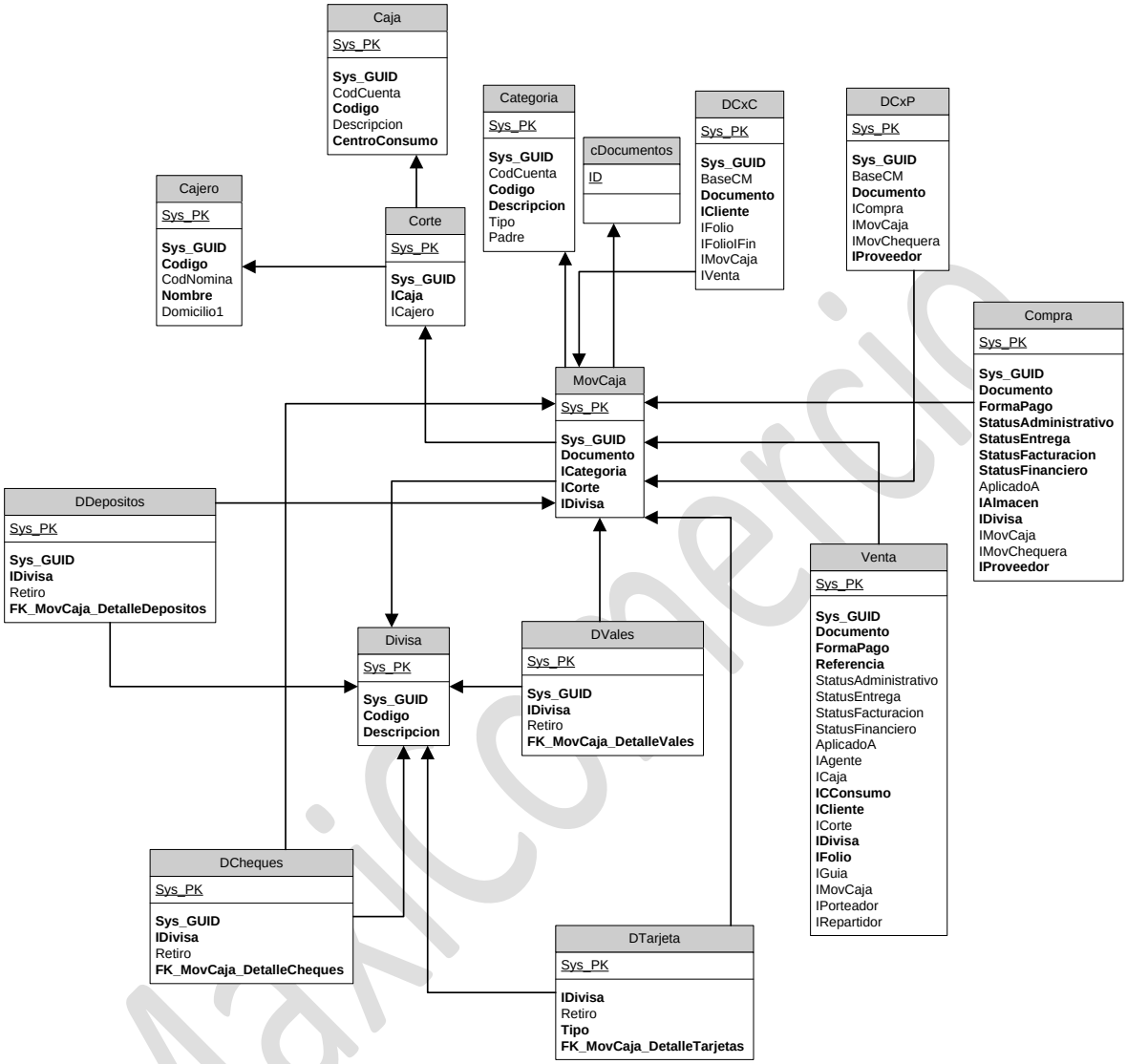
Relación entre las cajas y los centros de consumo



Cajas, cajeros, cortes y movimientos



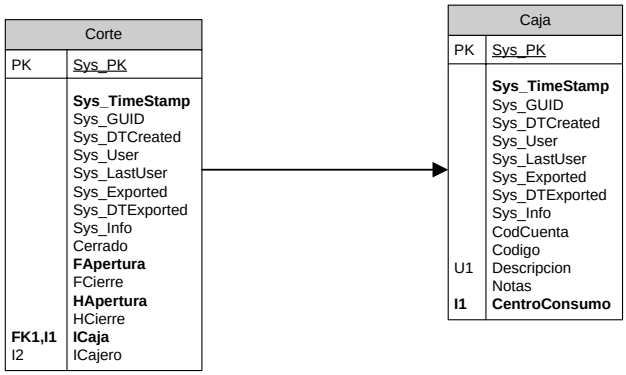
Relaciones de los movimientos de caja



Conceptos clave

- Corte o sesión de caja. Engloba un conjunto de operaciones en una caja en un periodo de tiempo por un cajero.
- Movimiento. Cargo o abono a la cuenta de una caja, se requiere una sesión abierta
- Transferencia entre cajas. Implica el traslado de recursos desde una caja a otra.

Corte. Una caja tiene cero o muchos cortes (sesiones) relacionadas.



Movimientos

Cada corte tiene un conjunto de movimientos. En general el movimiento es un ingreso o egreso de efectivo que considera las siguientes formas:

- Efectivo. (Metálico) Monedas y billetes en circulación.
- Tarjeta de crédito. Cobros con tarjetas de crédito, en este caso se mantiene un registro de las tarjetas y montos individuales que forman la transacción.
- Cheques. El importe recibido en cheques, se mantiene un detalle de los bancos, números y montos individuales.
- Vales. Cualquier importe cubierto mediante vales aceptables.
- Fichas de depósito. Considera ingresos los comprobantes de depósitos hechos a cuentas bancarias.

Cada movimiento es respaldado por un **documento** y puede ser de cualquier divisa.

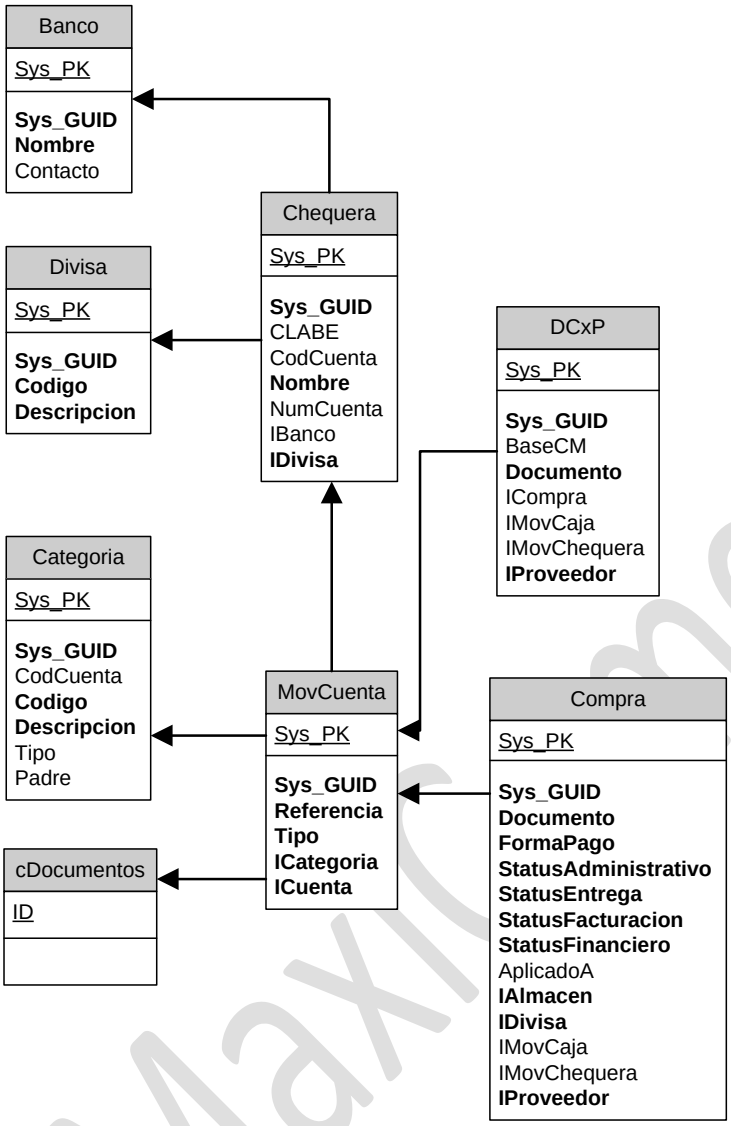
Por otro lado y con fines de obtención de reportes se le vincula una categoría.

Tipos (o formas) de efectivo

Enumeración: cTiposEfectivo

ID	Valor
1	cMetalico
2	cCheques
3	cDepositos
4	cTarjetas
5	cVales
99	cTodos

Chequeras

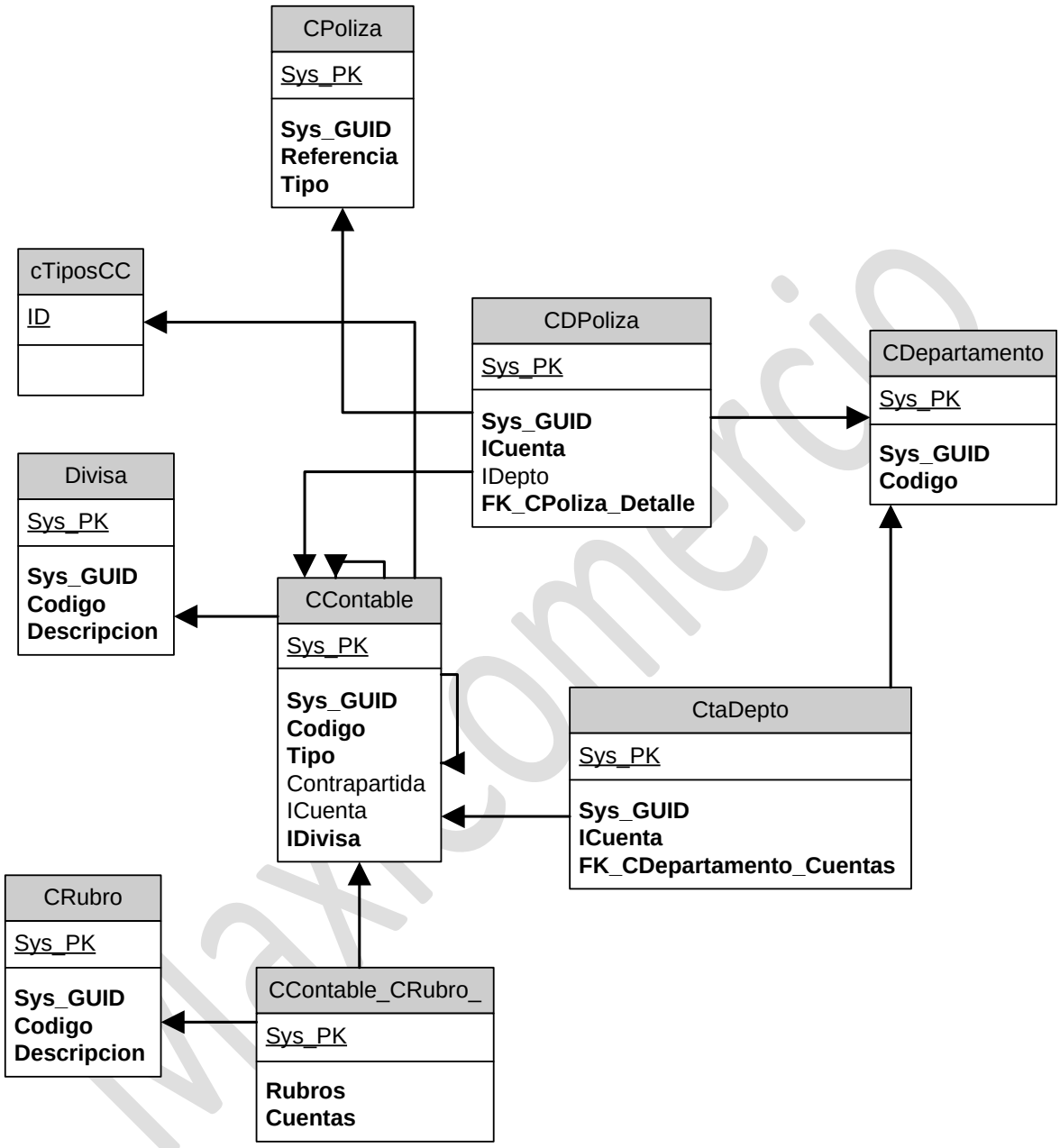


Conceptos clave

- Cada chequera (cuenta de cheques) puede ser de una divisa diferente
- El orden de los movimientos en la tabla MovCuenta será Fecha de aplicación, Egreso para asegurar que primero vayan los ingresos y luego los egresos (que serán cero en el caso de depósitos)
- Cada movimiento tiene relacionada una categoría para obtener reportes.

> ***Importante:***
>
> ***Fecha y Fecha de aplicación, la primera es la que corresponde al registro del movimiento en el sistema y solo es informativa, la segunda es la de aplicación y afecta el saldo a una fecha dada***

Contabilidad

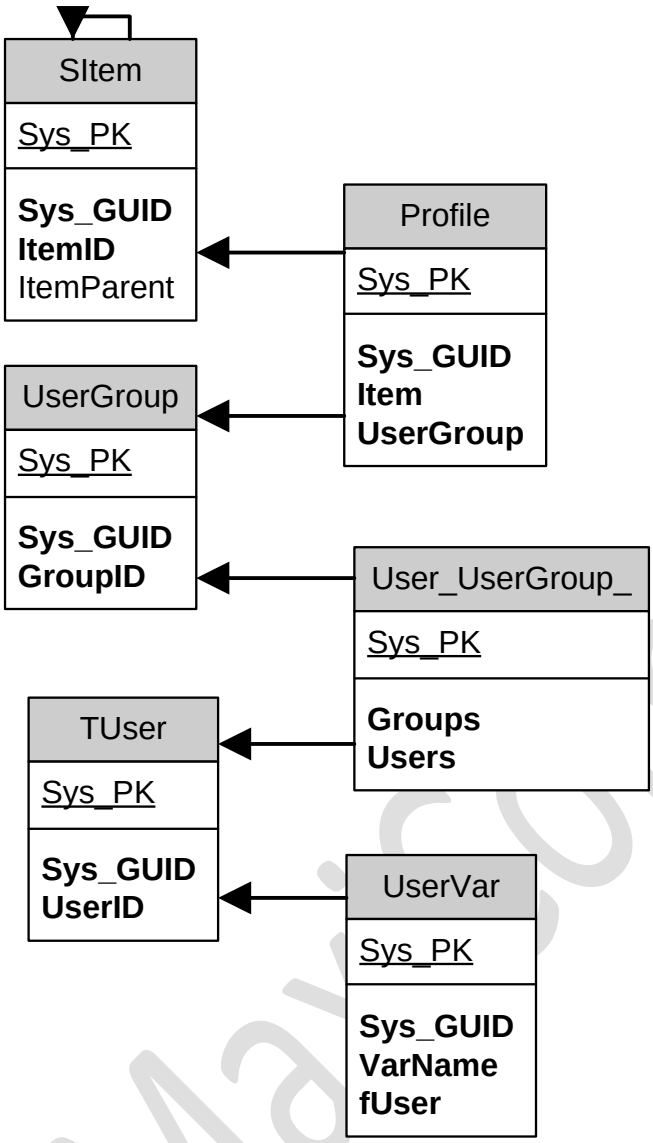


Las pólizas están contenidas en las tablas CPoliza y CDPoliza (principal/detalle respectivamente).

El catálogo de cuentas está en la tabla CContable.

Los saldos de las cuentas están en la tabla CContable y se acumulan cuando las pólizas se aplican (vea el campo Aplicado de la tabla CPoliza)

Usuarios, grupos perfiles y permisos

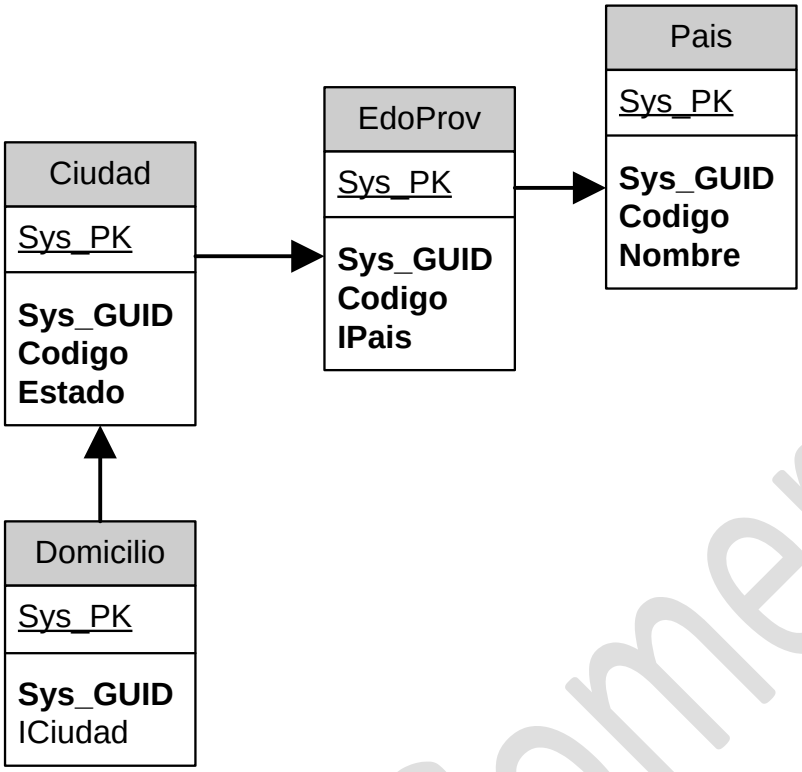


Los usuarios están en la tabla TUser, los grupos en la tabla UserGroup y la relación entre ellos definida en Use_UserGroup_.

El esquema de seguridad, es decir la lista de permisos disponibles está en la tabla SItem, los permisos que tiene un grupo de usuarios se encuentran en la tabla Profile.

La tabla UserVar guarda valores del usuario.

Catálogo de ciudades, estados y países



La tabla Pais contiene un catálogo de países.

La tabla EdoProv contiene a las provincias o estados de un país manteniendo una relación a través de la clave foránea IPais.

Ciudad mantiene los nombres de las ciudades de un estado o provincia, se relaciona con la tabla EdoProv mendiante el campo Estado.

Por su parte, la tabla Domicilio tiene los domicilios de clientes, proveedores y contactos. Cada domicilio está relacionado con una ciudad mediante ICiudad.

Consideraciones importantes

Tablas y campos que no se deben afectar directamente

Si va a usar instrucciones SQL o ejecutará programas que alteren los datos del sistema tome en cuenta las siguientes previsiones.

Tabla: Existencias

Contiene la existencia y el valor correspondiente por almacén.

- No agregar, modificar ni eliminar registros

Campos de la tabla Producto: Saldo y Existencia

Contienen los valores correspondientes acumulados de todos los almacenes.

- No alterar directamente estos valores

Campos de la tabla Cliente: Saldo, Puntos, SaldoPuntos y Monedero

Contienen valores acumulados.

- No alterar directamente estos valores

Campos de la tabla Proveedor: Saldo

Contiene el saldo por pagar acumulado.

- No alterar este valor

Campos de la tabla CContable: SInicial, SFinal, S01…S12, D01…D12,H1…H12

Contienen los valores acumulados de saldos, debe y haber de las cuentas contables.

- No alterar estos valores

Justificación para el uso de valores acumulados que se pueden calcular

Contraviniendo los principios de normalización de bases de datos, se acumulan los valores de saldos (clientes, proveedores y productos), existencias y algunos totales de documentos de compra y venta.

Se ha tomado esta decisión de diseño para favorecer el desempeño general del sistema.

> ***Ejemplo: El valor de la existencia de un producto se puede calcular sumando todas las entradas menos las salidas que se encuentran en el detalle de cardex (dcardex), sin embargo realizar esta operación cada vez que se requiriese volvería extremadamente lento al sistema. Es más rápido consultar el valor acumulado.***

Para prevenir inconsistencias de datos, la afectación de los valores acumulados se realiza mediante procedimientos almacenados y/o disparadores siempre en el contexto de una transacción.