

Documento de Estrategia

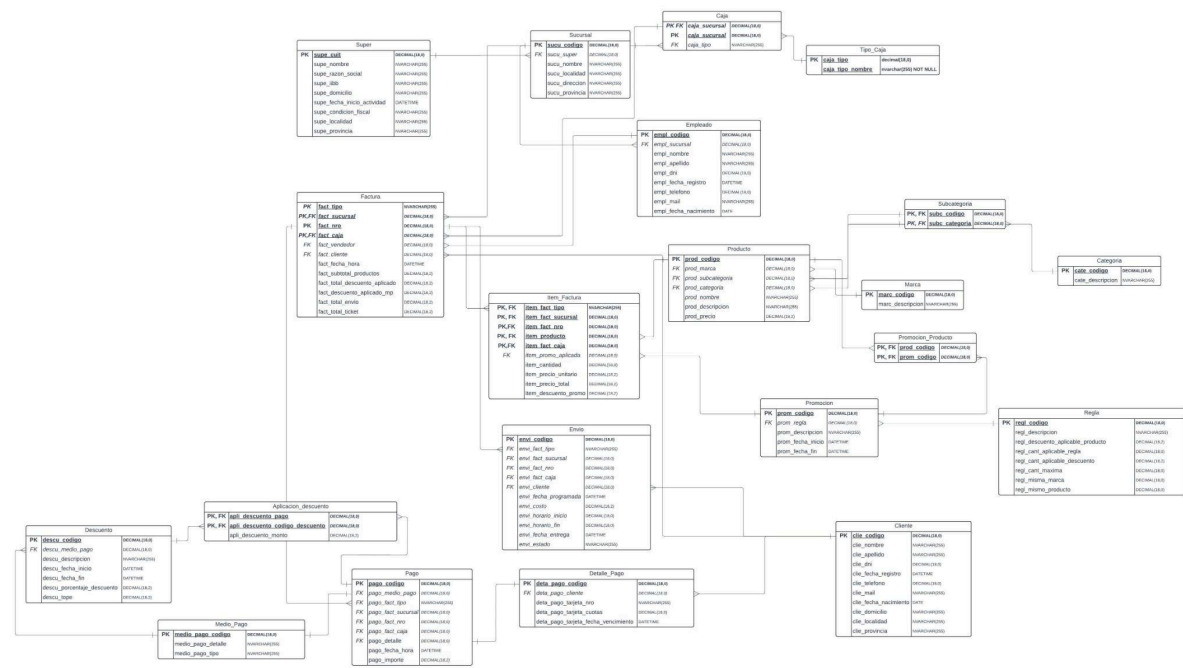
Gestión de Datos 1C 2024

Curso K3051| Grupo 30: GeDeDe

Índice

Modelo DER	3
Tablas	3
Relaciones más importantes	5
Creación de Tablas	5
Migración de datos	6
Modelo BI	7
Índices	7
Otras consideraciones	8

Modelo DER



El Diagrama de Entidad-Relación (DER) presentado representa un sistema de gestión para una cadena de supermercados. A continuación, se describen brevemente las funciones de cada tabla y sus relaciones clave

Tablas y su explicación:

Super: Esta tabla almacena la información sobre los supermercados, incluyendo su nombre, razón social, domicilio, y detalles fiscales. La clave primaria (PK) es **supe_cuit** que identifica de manera única a cada supermercado.

Sucursal: Contiene datos sobre las diferentes sucursales de la cadena, como nombre, dirección y localidad. La PK es **sucu_codigo**, generado por nosotros y hay una clave foránea (FK) sucu_super que se relaciona con Super, indicando a qué cadena pertenece cada sucursal.

Caja: Representa las cajas registradoras en cada sucursal. La PK compuesta es **caja_sucursal** y **caja_numero**. La FK **caja_sucursal** se relaciona con Sucursal, mientras que **caja_tipo** se refiere a la tabla **Tipo_Caja**.

Tipo_Caja: Define los diferentes tipos de cajas registradoras. La PK es **caja_tipo**. Y esos tipos de caja se diferencian por: Prioridad, Envío y Rápida

Empleado: Almacena información sobre los empleados, como nombre, apellido y fecha de nacimiento. La PK es **empl_codigo**, y la FK **empl_sucursal** indica en qué sucursal trabaja cada empleado.

Factura: Registra las facturas o tickets emitidos, con detalles como fecha, hora, subtotal de productos y total del ticket. La PK compuesta es **fact_tipo**, **fact_numero**, y **fact_sucu**.

Item_Factura: Detalla los productos incluidos en cada factura. La PK compuesta es **item_fact_tipo**, **item_fact_numero**, **item_fact_sucursal**, **item_fact_caja**, e **item_producto**. Las FKs relacionan esta tabla con **Factura** y **Producto**.

Envío: Contiene información sobre los envíos de productos, incluyendo fecha de registro y estado del envío. La PK es **envi_codigo**, y las FKs relacionan esta tabla con **Factura** y **Cliente**.

Cliente: Almacena datos de los clientes, como nombre, apellido, dirección y fecha de registro. La PK es **clie_codigo**.

Producto: Detalla los productos, incluyendo nombre, descripción y precio. La PK es **prod_codigo**, y las FKs **prod_marca** y **prod_subcategoria** se relacionan con **Marca** y **Subcategoria**, respectivamente.

Subcategoría: Define subcategorías de productos. Las subcategorías consideramos que es una relación muchos a muchos de la tabla categorías. Entonces subc_codigo y subc_categoria son ambos códigos de **categoría**. Por eso la PK compuesta es **subc_codigo** y **subc_categoria**.

Categoría: Define las categorías principales de productos. La PK es **cate_codigo**.

Marca: Almacena información sobre las marcas de productos. La PK es **marc_codigo**.

Promoción: Contiene detalles sobre las promociones, como descripción y fechas de inicio y fin. La PK es **prom_codigo**, y hay una FK **prom_regla** que se relaciona con Regla que aplica para esa promoción.

Promocion_Producto: Relaciona productos con promociones. La PK compuesta es **prom_codigo** y **prod_codigo**, relacionando esta tabla con Producto y Promoción.

Regla: Define las reglas aplicables a las promociones. La PK es **regl_codigo**.

Descuento: Almacena detalles sobre los descuentos aplicables a los pagos, incluyendo descripción y porcentaje aplicable. La PK es **descu_codigo**.

Aplicacion_Descuento: Aplicaciones de descuentos en pagos. La PK compuesta es **apl_descuento_pago** y **apl_descuento_codigo_descuento**.

Medio_Pago: Define los medios de pago aceptados. La PK es **medio_pago_codigo**.

Pago: Registra los pagos realizados, incluyendo importe y fecha. La PK es **pago_codigo**, y las FKs relacionan esta tabla con **Factura**.

Detalle_Pago: Contiene detalles específicos de los pagos, como fecha de vencimiento de la tarjeta. La PK es **deta_pago_codigo**.

Relaciones más importantes:

- **Super-Sucursal:** Una cadena de supermercados (Super) puede tener múltiples sucursales (Sucursal), creando una relación uno a muchos.
- **Sucursal-Caja:** Cada sucursal (Sucursal) puede tener varias cajas registradoras (Caja).
- **Sucursal-Empleado:** Cada sucursal (Sucursal) puede emplear a varios empleados (Empleado).
- **Factura-Item_Factura:** Una factura (Factura) puede incluir múltiples ítems (Item_Factura).
- **Factura-Envio:** Una factura (Factura) puede tener un solo envío (Envio).
- **Producto-Item_Factura:** Un producto (Producto) puede estar en múltiples ítems de factura (Item_Factura).
- **Producto-Promocion_Producto:** Un producto (Producto) puede estar en múltiples promociones (Promocion_Producto) y una promoción puede aplicarse a múltiples productos.
- **Cliente-Factura:** Un cliente (Cliente) puede tener múltiples facturas (Factura).
- **Promocion-Regla:** Una promoción (Promocion) puede tener múltiples reglas (Regla).
- **Descuento-Aplicacion_Descuento:** Un descuento (Descuento) puede aplicarse en múltiples situaciones (Aplicacion_Descuento).

Creación de Tablas

En la creación del modelo de base de datos para la cadena de supermercados, hemos seguido una metodología estructurada como indica el enunciado que incluye tanto el Lenguaje de Manipulación de Datos (DML) como el Lenguaje de Definición de Datos (DDL). Todo el proceso se ha automatizado y documentado en un script específico que se encuentra en la carpeta [/data](#).

Este script incluye todas las instrucciones necesarias para definir la estructura de la base de datos, crear el nuevo esquema y establecer las tablas correspondientes con sus relaciones, como también la migración de datos (ver en apartado Migración de datos).

El script automatiza la creación del nuevo esquema de la base de datos, asegurando que todas las tablas y sus relaciones se establezcan correctamente de acuerdo con el modelo diseñado. Al crear el script consideramos que sea determinístico para que independientemente si la persona que ejecuta el script tenga ya tablas creadas con esos nombres, el script considera y borra aquellas tablas antes de realizar los nuevos **create table**. Hay consideraciones o requerimientos iniciales (como tener la base de datos con la tabla maestra) que son consideradas obligatorias para correr el script.

Migración de datos

El script SQL realiza la creación y configuración de un esquema de base de datos llamado **[GeDeDe]** dentro de la base de datos **GD1C2024**. Aquí hay un resumen de los pasos principales que realiza:

Crear Esquema y Procedimientos:

Crea el esquema [GeDeDe].

Define dos procedimientos almacenados: [GeDeDe].[CREATE_DDL] y [GeDeDe].[CREATE_DML].

Definición del Procedimiento [CREATE_DDL]:

Dentro de este procedimiento se crean varias tablas, cada una con sus respectivas columnas y restricciones, incluyendo claves primarias y foráneas para asegurar la integridad referencial.

Las tablas creadas incluyen: Super, Sucursal, Tipo_Caja, Caja, Empleado, Factura, Marca, Categoria, Subcategoria, Producto, Regla, Promocion, Promocion_Producto, Item_Factura, Cliente, Envio, Medio_Pago, Descuento, Pago, Detalle_Pago, y Aplicacion_Descuento.

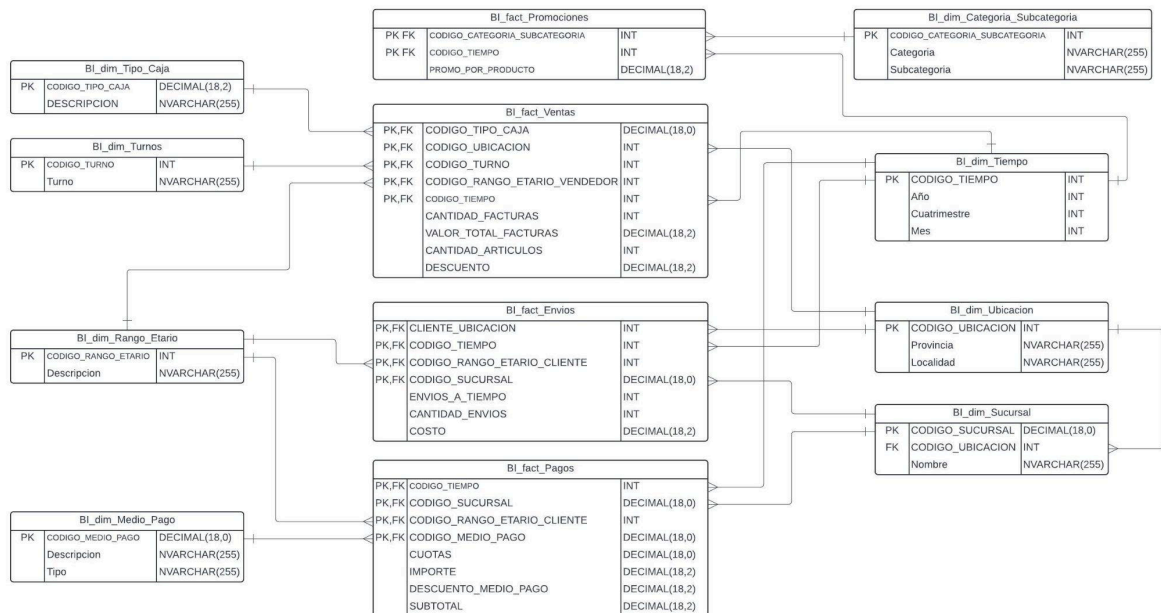
Definición del Procedimiento [CREATE_DML]:

Este procedimiento se encarga de insertar datos en las tablas del esquema [GeDeDe] seleccionándolos desde una tabla llamada [gd_esquema].[Maestra] que ya contiene datos. Los datos se insertan en las tablas Super, Sucursal, Tipo_Caja, Caja, Marca, Categoria, Subcategoria, Producto, y Empleado.

Cada inserción incluye una serie de operaciones INSERT INTO y SELECT que seleccionan y transforman los datos necesarios desde la tabla **[Maestra]** para adaptarlos a la estructura de las nuevas tablas.

Este script establece la estructura de una base de datos con múltiples entidades relacionadas, y luego pobla esas tablas con datos iniciales provenientes de una fuente maestra.

Modelo BI



El Diagrama de Entidad-Relación (DER) presentado representa un modelo multi-cubo, que posee 4 tablas de hechos (BI_fact_Promociones, BI_fact_Ventas, BI_fact_Envios y BI_fact_Pagos) y 8 tablas dimensionales (BI_dim_Categoria_Subcategoria, BI_dim_Tiempo, BI_dim_Ubicacion, BI_dim_Sucursal, BI_dim_Tipo_Caja, BI_dim_Turnos, BI_dim_Rango_Etario y BI_dim_Medio_Pago)

Tablas y su explicación:

BI_dim_Categoria_Subcategoria: esta tabla se migra directamente del modelo OLTP y posee los campos CODIGO_CATEGORIA_SUBCATEGORIA (PK), Categoria y Subcategoria.

BI_dim_Tiempo: esta tabla se migra junto a la de hechos "BI_fact_Ventas", donde tomamos la fecha de las facturas, calculamos el mes, cuatrimestre y año correspondiente a la misma y en caso de no existir dentro de esta tabla, lo insertamos. Posee los campos CODIGO_TIEMPO (PK), Año, Cuatrimestre y Mes.

BI_dim_Ubicacion: esta tabla se migra en conjunto a "BI_dim_Sucursal", tomamos la localidad y la provincia de las sucursales del modelo OLTP, y en caso de no existir dentro de esta tabla, las insertamos. Posee los campos CODIGO_UBICACION (PK), Provincia y Localidad.

BI_dim_Sucursal: esta tabla se migra en conjunto a “BI_dim_Ubicacion”, tomamos las sucursales del modelo OLTP y en base a su localidad y provincia, le asignamos un CODIGO_UBICACION. Posee los campos CODIGO_SUCURSAL (PK), CODIGO_UBICACION (FK que hace referencia a la PK de “BI_dim_Ubicacion”) y Nombre.

BI_dim_Tipo_Caja: esta tabla se migra directamente del modelo OLTP y posee los campos CODIGO_TIPO_CAJA (PK) y DESCRIPCION.

BI_dim_Turnos: esta tabla se migra junto a la de hechos “BI_fact_Ventas”, donde a partir de una función “obtenerTurnos()”, le pasamos por parámetro la fecha/hora de la factura y en caso de devolver un turno que aún no exista en esta tabla, lo inserta. Posee los campos CODIGO_TURNO (PK) y el turno.

BI_dim_Rango_Etario: esta tabla se migra junto a la de hechos “BI_fact_Ventas”, donde a partir de una función “obtenerRangoEtario()”, le pasamos por parámetro la edad del vendedor y en caso de devolver un rango etario que aún no exista en esta tabla, lo inserta. Posee los campos CODIGO_RANGO_ETARIO (PK) y su descripción.

BI_dim_Medio_Pago: esta tabla se migra directamente del modelo OLTP y posee los campos CODIGO_MEDIO_PAGO(PK), descripción y tipo.

BI_fact_Ventas: esta tabla contiene información agrupada según sus dimensiones, que conforman la PK Compuesta (CODIGO_TIEMPO, CODIGO_UBICACION, CODIGO_RANGO_ETARIO_VENDEDOR, CODIGO_TURNO y CODIGO_TIPO_CAJA). Además, posee los campos CANTIDAD_FACTURAS, VALOR_TOTAL_FACTURAS, CANTIDAD_ARTICULOS y DESCUENTO, que son necesarios para crear las vistas 1, 2, 3, 4 y 5.

BI_fact_Promociones: esta tabla contiene información agrupada según sus dimensiones, que conforman la PK Compuesta (CODIGO_TIEMPO, CODIGO_CATEGORIA_SUBCATEGORIA). Además, posee el campo PROMO_POR_PRODUCTO, necesario para crear la vista 6.

BI_fact_Envios: esta tabla contiene información agrupada según sus dimensiones, que conforman la PK Compuesta (CODIGO_TIEMPO, CODIGO_SUCURSAL, CODIGO_RANGO_ETARIO_CLIENTE, CLIENTE_UBICACION). Además, posee los campos ENVIOS_A_TIEMPO, CANTIDAD_ENVIOS y COSTO, que son necesarios para crear las vistas 7, 8 y 9.

BI_fact_Pagos: esta tabla contiene información agrupada según sus dimensiones, que conforman la PK Compuesta (CODIGO_TIEMPO, CODIGO_SUCURSAL, CODIGO_RANGO_ETARIO_CLIENTE, CODIGO_MEDIO_PAGO). Además, posee los campos CUOTAS, IMPORTE, DESCUENTO_MEDIO_PAGO y SUBTOTAL, que son necesarios para crear las vistas 10, 11 y 12.

Para la migración de las tablas de hechos, utilizamos tablas temporales (intermedias), para poder insertar duplicadas lo que vendría a ser las PK de las tablas de hechos, y luego

agrupar sobre estos campos y hacer los pre-cálculos necesarios para luego crear las vistas correspondientes.

Otras consideraciones

Encontramos 4 tickets, que fueron emitidos por Cajas diferentes en la misma sucursal

```

select count(distinct SUCURSAL_NOMBRE) 'DISTINTAS SUCURSALES',
TICKET_NUMERO,
count(distinct TICKET_TIPO_COMPROBANTE) 'DISTINTOS TIPOS',
count(distinct CAJA_NUMERO) 'CAJAS DISTINTAS'
from gd_esquema.Maestra
where CAJA_NUMERO is not null
group by TICKET_NUMERO, SUCURSAL_NOMBRE, TICKET_TIPO_COMPROBANTE--, CAJA_NUMERO
order by 3 desc, 1 desc, 4 desc

```

	DISTINTAS SUCURSALES	TICKET_NUMERO	DISTINTOS TIPOS	CAJAS DISTINTAS
1	1	1351465859	1	2
2	1	1353429485	1	2
3	1	1352673365	1	2
4	1	1352525134	1	2
5	1	1352525135	1	1
6	1	1352525137	1	1
7	1	1352525140	1	1
8	1	1352525144	1	1
9	1	1352525149	1	1
10	1	1352525155	1	1
11	1	1352525162	1	1
12	1	1352525170	1	1
13	1	1352525179	1	1
14	1	1352525189	1	1
15	1	1352525200	1	1
16	1	1352525212	1	1
17	1	1352525225	1	1
18	1	1352525239	1	1

Query executed successfully. DESKTOP-J2HAKF4\SQLEXPRESS ... DESKTOP-J2HAKF4\FedeT ... GD1C2024 00:00:01 17,064 rows

Por esto mismo, tomamos la decisión de agregar la Caja como parte de la PK de Factura.

Por otro lado encontramos un ticket (factura para nosotros) que no tiene asociado un cliente cuando es pagado con tarjeta de crédito. Esto hace que luego en el modelo BI, no coincida la cantidad de Facturas, con la Cantidad de Pagos de Clientes (porque no tenemos forma de relacionar este ticket con un cliente).

```

select PAGO_TIPO_MEDIO_PAGO, CLIENTE_DNI
FROM [GD1C2024].[gd_esquema].[Maestra]
where ticket_numero = '1351318319'

```

	PAGO_TIPO_MEDIO_PAGO	CLIENTE_DNI
1	NULL	NULL
2	NULL	NULL
3	NULL	NULL
4	NULL	NULL
5	NULL	NULL
6	NULL	NULL
7	NULL	NULL
8	NULL	NULL
9	NULL	NULL
10	NULL	NULL
11	Tarjeta Crédito	NULL

Query executed successfully. DESKTOP-J2HAKF4\SQLEXPRESS ... DESKTOP-J2HAKF4\FedeT ... GD1C2024 00:00:00 11 rows

