

Gestión de Datos

1° Cuatrimestre 2024



Supermercado

Grupo: "Los_Migradores"

Integrantes: Tobías Castañeda

Gastón Ramírez

Brenda Torrico

PROFESOR: Enrique Reinoso, Marcelo Moscuzza

Curso: K3012 /K3051/ K3151

INDICE:

Diagramas -----	2
Tablas modelo transaccional-----	2
Correcciones primera entrega-----	6
Correcciones segunda entrega-----	6
Funciones y procedures auxiliares-----	7
Tablas modelo BI -----	8
Views -----	10
ANEXO -----	11

Objetivo

El propósito de este documento es expresar las justificaciones y decisiones tomadas en la entrega del Diagrama de Entidad Relación y en la migración de datos realizada.

Diagrama de Entidad Relación:

La imagen del diagrama se encuentra en el [Anexo](#) de archivos.

Entidades:

TABLA MAESTRA:

Inicialmente partimos de una tabla maestra que contiene los siguientes atributos:

SUPER_NOMBRE	SUPER_RAZON_SOC	SUPER_CUIT
SUPER_IIBB	SUPER_DOMICILIO	SUPER_FECHA_INI_ACTIVIDAD
SUPER_CONDICION_FISCAL	SUPER_LOCALIDAD	SUPER_PROVINCIA
SUCURSAL_NOMBRE	SUCURSAL_DIRECCION	SUCURSAL_LOCALIDAD
SUCURSAL_PROVINCIA	TICKET_NUMERO	TICKET_FECHA_HORA
TICKET_TIPO_COMPROBANTE	TICKET_SUBTOTAL_PRODUCTOS	TICKET_TOTAL_DESCUENTO_APLICADO
TICKET_TOTAL_DESCUENTO_APLICADO_MP	TICKET_TOTAL_ENVIO	TICKET_TOTAL_TICKET
EMPLEADO_NOMBRE	EMPLEADO_APELLIDO	EMPLEADO_DNI
EMPLEADO_FECHA_REGISTRO	EMPLEADO_TELEFONO	EMPLEADO_MAIL
EMPLEADO_FECHA_NACIMIENTO	CAJA_NUMERO	CAJA_TIPO
TICKET_DET_CANTIDAD	TICKET_DET_PRECIO	TICKET_DET_TOTAL
PRODUCTO_NOMBRE	PRODUCTO_DESCRIPCION	PRODUCTO_PRECIO
PRODUCTO_MARCA	PRODUCTO_SUB_CATEGORIA	PRODUCTO_CATEGORIA
PROMO_APLICADA_DESCUENTO	PROMO_CODIGO	PROMOCION_DESCRIPCION
PROMOCION_FECHA_INICIO	PROMOCION_FECHA_FIN	REGLA_APLICA_MISMA_MARCA
REGLA_APLICA_MISMO_PROD	REGLA_CANT_APLICA_DESCUENTO	REGLA_CANT_APLICABLE_REGLA
REGLA_CANT_MAX_PROD	REGLA_DESCRIPCION	REGLA_DESCUENTO_APLICABLE_PROD

PAGO_FECHA	PAGO_IMPORTE	PAGO_MEDIO_PAGO
PAGO_TIPO_MEDIO_PAGO	PAGO_TARJETA_NRO	PAGO_TARJETA_CUOTAS
PAGO_TARJETA_FECHA_VENC	PAGO_DESCUENTO_APLICADO	DESCUENTO_CODIGO
DESCUENTO_DESCRIPCION	DESCUENTO_FECHA_INICIO	DESCUENTO_FECHA_FIN
DESCUENTO_PORCENTAJE_DESC	DESCUENTO_TOPE	ENVIO_COSTO
ENVIO_FECHA_PROGRAMADA	ENVIO_HORA_INICIO	ENVIO_HORA_FIN
ENVIO_FECHA_ENTREGA	ENVIO_ESTADO	CLIENTE_NOMBRE
CLIENTE_APELLIDO	CLIENTE_DNI	CLIENTE_FECHA_REGISTRO
CLIENTE_TELEFONO	CLIENTE_MAIL	CLIENTE_FECHA_NACIMIENTO
CLIENTE_DOMICILIO	CLIENTE_LOCALIDAD	CLIENTE_PROVINCIA

Decidimos agruparlos en 23 entidades diferentes

A continuación, se documentarán las decisiones de diseño tomadas que consideramos más importantes durante la creación y migración de tablas.

TICKET, DETALLE_TICKET, PROMOCION_POR_TICKET, PROMOCION, PROMOCION_POR REGLA, REGLA, PRODUCTO_PROMO, PRODUCTO, SUBCATEGORIA, CATEGORIA, CLIENTE, ENVIO, PAGO, PAGO_POR_DESCUENTO, DETALLE_PAGO, DESCUENTO_MP, MEDIO_DE_PAGO, EMPLEADO, SUCURSAL, CAJA, CAJA_POR_TIPO_DE_CAJA, SUPER, TIPO_DE_CAJA

- **Migración de datos de la tabla Producto_por_Promo**

Durante la migración de datos hacia la tabla Producto_por_Promo, encontramos una complicación significativa: los campos producto_id y subcategoria_id no están presentes en la tabla maestra (gd_esquema.Maestra). Estos campos son autoincrementales y solo se encuentran en las tablas Producto y Subcategoria. Para resolver este problema, decidimos realizar un JOIN con estas tablas para recuperar producto_id.

Para recuperar producto_id, unimos la tabla maestra con la tabla Subcategoria usando subcategoria_nombre, y luego con la tabla Producto utilizando una combinación de campos (descripción, nombre, precio, marca y subcategoría). Esta estrategia, aunque necesaria, introduce la desventaja de que el JOIN inicial con Subcategoria no se realiza sobre la clave primaria (subcategoria_id). Esto podría haber afectado potencialmente la integridad de la migración.

Para mitigar este riesgo, tomamos las siguientes decisiones: La consulta incluye un DISTINCT para evitar duplicados y un filtro para considerar solo registros con códigos de promoción válidos. Este enfoque, aunque imperfecto, permite una migración precisa y consistente de los datos a la tabla Producto_por_Promo, asegurando que la integridad de los datos se mantenga lo más posible dentro de las limitaciones dadas.

- **Diseño de tabla Subcategoría**

Durante el proceso de migración de datos hacia la tabla Subcategoría, enfrentamos un desafío significativo: la presencia de múltiples filas con el mismo valor en el campo PRODUCTO_SUB_CATEGORIA pero con diferentes valores en el campo PRODUCTO_CATEGORIA. Este escenario no cumplía inicialmente con la cardinalidad establecida en el Diagrama de Entidad-Relación (DER), lo que podría comprometer la integridad de los datos migrados.

Para abordar esta situación sin perder información crucial de la tabla maestra, decidimos realizar un rediseño en la estructura de la tabla Subcategoría. Cambiamos la clave primaria de la tabla de subcategoria_nombre a subcategoria_id, implementando un campo autoincremental. Este cambio nos permitió tener varias filas con la misma subcategoria_nombre pero con diferentes valores en subcategoria_categoria, preservando así la integridad de los datos y evitando la pérdida de información.

- **Diseño y migración de datos de tabla Producto.**

Durante la migración de datos hacia la tabla Producto, nos enfrentamos a un desafío relacionado con la duplicación de filas en la tabla maestra. Observamos que múltiples filas tenían el mismo PRODUCTO_NOMBRE y PRODUCTO_DESCRIPCION, pero con valores diferentes en los campos PRODUCTO_PRECIO y PRODUCTO_SUB_CATEGORIA. La razón detrás de estos datos distintos no estaba clara, lo que planteaba una dificultad para mantener la integridad de los datos durante la migración.

Para evitar la pérdida de información, decidimos cambiar la clave primaria de la tabla Producto de producto_nombre a producto_id, utilizando un campo autoincremental. Esta modificación nos permitió almacenar filas con el mismo nombre y descripción, pero con valores diferentes en precio y subcategoría, asegurando así la conservación de todos los datos relevantes.

Sin embargo, este cambio afectó el proceso de migración, ya que ahora necesitábamos guardar el campo producto_subcategoria, que ahora sería subcategoria_id. Dado que la clave primaria de Subcategoría no estaba presente en la tabla maestra, no pudimos realizar un JOIN directo. La solución adoptada fue realizar una subconsulta en la tabla Subcategoría, buscando la fila que coincidiera con los valores de subcategoria_nombre y subcategoria_categoria, que sí están presentes en la tabla maestra.

Es importante destacar que esta estrategia introdujo un aumento en el tiempo de procesamiento, elevándolo a 17 segundos. Pese a esta desventaja en rendimiento, la prioridad fue mantener la integridad y la consistencia de los datos.

- **Creación y migración de datos de la tabla Ticket**

La consulta realizada tiene como objetivo migrar datos de la tabla "Maestra" al esquema "LOS_MIGRADORES", específicamente a la tabla "Ticket". Durante el proceso de migración, se identificó una discrepancia entre el número total de tickets esperados y el número real de registros en la tabla de origen. Inicialmente, se esperaba migrar un total de 70068 filas de tickets. Sin embargo, al realizar una consulta para seleccionar registros distintos por el campo "TICKET_NUMERO" en la tabla "Maestra" del esquema "gd_esquema", se obtuvieron solo 70025 filas. Esta discrepancia se atribuye a la presencia de 23 filas duplicadas que comparten el mismo número de ticket pero presentan diferencias significativas en otros campos como fecha, monto, número de caja e incluso clientes asociados. Un ejemplo de esta duplicidad se encuentra en el número de ticket **1351465859**, el cual registra dos filas con distintas fechas, números de cajas, montos y hasta clientes, si aclarar la razon detrás Para preservar la integridad de los datos y evitar la pérdida de información, se tomó la decisión de modificar la clave primaria (PK) de la tabla "Ticket" en el esquema "LOS_MIGRADORES". En lugar de basarse en el número de ticket, la nueva PK se estableció como una clave autoincremental denominada "TICKET_ID". Esta modificación permitió la inclusión de las 70068 filas esperadas, aunque con la posibilidad de que se repitan números de tickets. Es importante destacar que esta modificación conlleva una disminución en el rendimiento de las consultas, especialmente en la velocidad de ejecución, pudiendo alcanzar hasta 30 segundos.

Correcciones en el diagrama a partir de la primera entrega:

A partir de las correcciones y sugerencias recibidas se realizaron las siguientes modificaciones

- Decidimos crear la entidad “Promoción_por_ticket” para plasmar la relación de muchos a muchos entre Ticket_Detalle y Promoción ya que Ticket_Detalle ya se encuentra relacionada anteriormente con la entidad Producto.
- Se agrego la tabla “Categoría” para poder generar la relación con la tabla Subcategoría.
- Se agrega la entidad “Tipo_caja” para relacionarse con caja_por_tipo_caja, anteriormente no identificada en la primera entrega.
- Se crea la entidad “Detalle_Pago” que se termina relacionando con la entidad Pago.

- Se genera una nueva entidad “Pago_Por_Descuento” a partir de la relación de muchos a muchos entre la tabla Descuento_MP y Pago

Correcciones en el diagrama a partir de la segunda entrega:

Migración de la tabla Ticket_Detalle

Para recuperar los cinco campos de la tabla, se utilizo:

- Para ticket_id, una subquery a la tabla Ticket
- Para producto_id, una subquery a la tabla Producto
- Para la cantidad, monto total y descuentos, se saco de la tabla maestra.

Surgió el inconveniente de tener mas una fila con el mismo numero de ticket y el mismo producto comprado, con distintas cantidades y montos.

Interpretamos esta situación como el de una factura del mismo ticket con el mismo producto, por separado, por lo que vemos la necesidad de unificarlo en una sola fila. La solución fue hacer una sumatoria de las cantidades, el monto total, y los descuentos, para así respetar la unicidad de las PK.

Modelo de Inteligencia de Negocios (BI)

Implica la migración de datos desde el esquema existente hacia un nuevo esquema denominado LOS_MIGRADORES, optimizado para soportar análisis de BI (Business Intelligence) eficiente y procesos de consulta avanzados. Se incluye la redefinición de tablas, vistas, funciones y restricciones para asegurar coherencia y rendimiento en la base de datos.

Decisiones Tomadas

Eliminación de Objetos Existentes

Se eliminaron las siguientes funciones y vistas del esquema anterior para simplificar la estructura y reducir la complejidad:

Modificación de Restricciones de Claves Foráneas

Se ajustaron las restricciones de claves foráneas en las tablas BI_Venta, BI_Pago y BI_Envio para asegurar integridad referencial y optimizar consultas:

- Se eliminaron y recrearon las siguientes restricciones: FK_Venta_Empleado, FK_Venta_Tiempo, FK_Venta_Sucursal, FK_Venta_Turno, FK_Venta_TipoCaja, FK_Venta_Producto, FK_Pago_MedioDePago, FK_Pago_Cliente, FK_Pago_Descuento, FK_Envio_FechaProgramada, FK_Envio_FechaEntrega, FK_Envio_Cliente.

Creacion de funciones auxiliares

- Funciones: devolverCuatri, devolverRangoEtareo, devolverTurno, entregaATiempo, devolverTiempoDeFecha, devolverImporteDeCuotas

Creacion de procedimientos auxiliares

- Procedimiento: MigrarTiemposSinRepetir

Creación de Nuevas Tablas Dimensionales

Se diseñaron nuevas tablas dimensionales en el esquema LOS_MIGRADORES para almacenar información clave utilizada en análisis de BI:

BI_Empleado

- EMPLEADO_DNI
- EMPLEADO_NOMBRE
- EMPLADO_APELLIDO
- EMPLEADO_TELEFONO
- EMPLEADO_MAIL
- EMPLEADO_FECHA_NACIMIENTO
- EMPL_SUCURSAL_NOMBRE
- EMPL_RANGO_ETAREO

BI_Tipo_Caja

- CAJA_TIPO

BI_Sucursal

- SUCURSAL_NOMBRE
- SUCURSAL_DIRECCION
- SUCURSAL_LOCALIDAD
- SUPER_NOMBRE
- SUCURSAL_PROVINCIA

BI_Producto

- PRODUCTO_ID
- PRODUCTO_NOMBRE
- PRODUCTO_DESCRIPCION
- PRODUCTO_PRECIO
- PRODUCTO_MARCA
- PRODUCTO_SUBCATEGORIA
- PRODUCTO_CATEGORIA

BI_Cliente

- CLIENTE_DNI
- CLIENTE_NOMBRE
- CLIENTE_APELLIDO
- CLIENTE_FECHA_REGISTRO
- CLIENTE_TELEFONO
- CLIENTE_MAIL
- CLIENTE_FECHA_NACIMIENTO
- CLIENTE_DOMICILIO
- CLIENTE_LOCALIDAD
- CLIENTE_PROVINCIA
- CLIENTE_RANGO_ETAREO

BI_Tiempo

- TIEMPO_ID
- TIEMPO_MES
- TIEMPO_ANIO
- TIEMPO_CUATRI

BI_Medio_de_Pago

- MEDIO_DE_PAGO_NOMBRE
- MEDIO_DE_PAGO_TIPO

Creación de Nuevas Tablas Hechos

Se diseñaron nuevas tablas dimensionales en el esquema LOS_MIGRADORES para almacenar información clave utilizada en análisis de BI:

BI_Venta

- EMPLEADO_ID
- TIEMPO_ID
- PRODUCTO_ID
- SUCURSAL_ID
- TURNO_ID
- TIPO_DE_CAJA_ID
- IMPORTE_TOTAL_VENTAS
- CANTIDAD_ARTICULOS
- CANTIDAD_VENTAS

BI_PAGO

- MEDIO_DE_PAGO_ID
- SUCURSAL_ID
- TIEMPO_ID
- CLIENTE_RANGO_ETAREO
- IMPORTE_EN_CUOTAS
- IMPORTE_TOTAL
- DESCUENTO_TOTAL

BI_ENVIO

- ENVIO_FECHA_ENTREGA
- ENVIO_FECHA_PROGRAMADA
- SUCURSAL_ID
- ENVIO_CLIENTE
- COSTO_ENVIO
- CANTIDAD_DE_ENVIOS_CUMPLIDOS
- CANTIDAD_DE_ENVIOS

BI_PROMOCION

- TIEMPO_ID
- PRODUCTO_ID
- DESCUENTO_TOTAL_APLICADO_POR_PROMOCION
- VENTA_TOTAL_TICKETS
- CANTIDAD_TICKETS
- PORCENTAJE_DESCUENTO_RESPECTO_TOTAL_TICKETS

Migración de Datos

Se realizó la migración de datos desde las tablas existentes del archivo 'script.creacion.inicial.sql' hacia las nuevas tablas del esquema LOS_MIGRADORES en el archivo 'script.creacion.inicial_BI.sql' para asegurar consistencia y completitud de la información necesaria para análisis:

- Se ejecutaron scripts de migración específicos para cada tabla utilizando consultas
INSERT INTO ... SELECT DISTINCT ... FROM

- El orden de migración fue en primer lugar las tablas dimensionales, y el segundo lugar las tablas de hechos

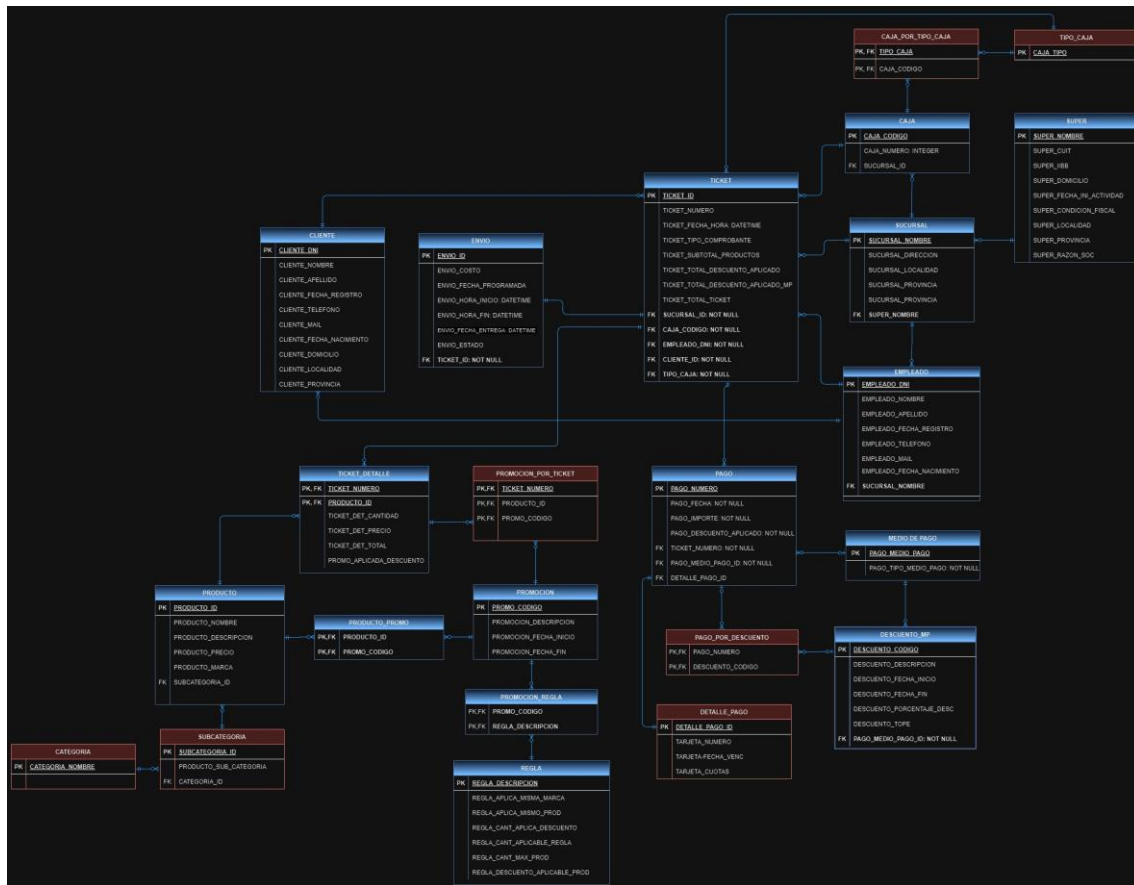
Cambios y aclaraciones de la migración de datos BI

- Se decidió guardar el campo sucursal_id en las tablas BI_Pago y BI_Envio, haciendo un JOIN con la tabla Ticket
- Se decidió guardar campo cliente_id y cliente_rango_etareo en BI_Pago como un campo común, ya que al obtener este valor de joinear con Ticket, cabe la posibilidad de que el campo sea NULL si el medio de pago fue con efectivo, en dicho caso el rango etareo devolvera el valor 'Desconocido'
- Los campos BI_cliente y BI_empleado almacenaran un campo llamado rango_etareo, cuyo valor se obtendrá utilizando la función devolverRangoEtareo.
- Se decidió crear una tabla Tiempo donde guardara las fechas. Tendrán un PK autoincremental, y almacenará el día, año y cuatrimestre. El cuatrimestre se obtendrá con la función devolverCuatrimestre. Para referenciar a la tabla BI_Tiempo, se utilizará la función devolverTiempoParaFecha .
- Se decidió que la tabla BI_Envio almacene los envíos que no se cumplieron a tiempo, filtrándolos con una función llamada entregaATiempo
- Se decidió agregar en la tabla Ticket el campo tipo_de_caja en el que se registra, para así facilitar su obtención.

Creación de vistas

- Vista_TicketPromedioMensual
- Vista_CantidadUnidadesPromedio
- PorcentajeAnualDeVentas
- CantidadVentasRegistradasPorTurno
- v_PorcentajeDescuentoAplicadoEnFuncionDelTotal
- v_tres_categorias_productos_mayor_descuentos_aplicado
- v_porcentaje_cumplimiento_envios
- v_cantidad_envios_por_rango_etario_cuatrimestre
- v_5LocalidadesMayorCosto
- v_tres_sucursales_mayor_importe_en_cuotas
- v_promedio_importe_rango_etareo
- v_porcentaje_desc_aplicado

MODELO OLTP (Actualizado tercera entrega)



MODELO OLAP

