

Análisis y Diseño de una Base de Datos para la Gestión de Supermercados

Asignatura:

- Gestión de Datos

Profesor:

- Enrique Reinoso

Estudiantes:

- Juan Manuel Furlan Hermida
- Tobias Ducrot
- Juan Cruz Neira
- Nicolas Pinto

Fecha de Entrega:

- 30 de Mayo del 2024

Institución:

- Universidad Tecnológica Nacional Facultad Regional
Buenos Aires

Índice:

Introducción	3
Estrategias de Diseño SQL	3
Estrategias de Diseño BI	6
Conclusion	10

Introducción

Este documento describe la estrategia seguida para diseñar una base de datos relacional destinada a gestionar la información de supermercados, sus sucursales, empleados, clientes, transacciones y otros aspectos relacionados. El diseño busca garantizar la integridad de los datos, la eficiencia en el almacenamiento y la migración, así como la facilidad para realizar consultas.

Estrategia de Diseño SQL

Creación de Tablas

Decisión: Se han definido tablas para las entidades clave del sistema y entidades auxiliares, incluyendo: *Provincia, Localidad, Supermercado, Sucursal, Empleado, Tipo_Caja, Caja, Cliente, Ticket, Medio_De_Pago, Descuento_Medio_Pago, Descuento_Aplicado_X_MP, Pago, Detalle_Pago, Estado_Envio, Envio, Categoria, Subcategoria, Categoria_X_Subcategoria, Marca, Regla, Promocion, Producto, Marca_X_Producto, Producto_X_Subcategoria, Rebaja_Producto, Item_Ticket, Promocion_X_Ticket, y Tipo_Medio_Pago.*

Justificación: Estas tablas representan las entidades principales del sistema y cubren todos los aspectos necesarios para la gestión de información en el ámbito de los supermercados. Cada tabla se enfoca en una entidad específica, permitiendo una mayor claridad y facilidad de mantenimiento.

Definición de Llaves Primarias y Foráneas

Decisión: Se han definido llaves primarias para cada tabla y se han establecido llaves foráneas para mantener la integridad referencial entre las tablas. En algunas tablas se han utilizado llaves primarias concatenadas para garantizar la unicidad de los datos.

Justificación: Las llaves primarias garantizan la unicidad de los registros en cada tabla. Las llaves foráneas aseguran que las relaciones entre tablas sean consistentes, evitando la entrada de datos huérfanos y manteniendo la integridad de los datos.

Tipos de Datos y Longitud de Campos

Decisión: Se han utilizado tipos de datos adecuados para cada campo y se han definido longitudes específicas donde sea necesario.

Justificación: Seleccionar tipos de datos adecuados optimiza el almacenamiento y la eficiencia en las consultas y la migración de la base. Definir longitudes específicas evita el desperdicio de espacio y mejora el rendimiento.

Integridad Referencial y Relacional

Decisión: Se han establecido relaciones entre tablas utilizando llaves foráneas.

Justificación: Mantener la integridad referencial es crucial para garantizar que los datos sean consistentes y que las relaciones entre entidades se respeten. Esto evita errores y asegura que la base de datos refleje correctamente la realidad del negocio.

Funciones de Extracción de Datos

Se han creado una serie de funciones de extracción de datos para procesar y limpiar las cadenas de texto provenientes de la tabla maestra. Estas funciones se utilizan para eliminar prefijos no deseados o caracteres especiales de los campos de texto, asegurando consistencia en los datos insertados en las tablas del sistema.

ExtractSucursal: Elimina el prefijo "Sucursal N°:" del nombre de la sucursal.

ExtractIIBB: Elimina el prefijo "Ingr. Brut. N°:" del registro de Ingresos Brutos.

ExtractCategoria: Elimina el prefijo "Categoria N°" del código de categoría.

ExtractSubCategoria: Elimina el prefijo "SubCategoria N°" del código de subcategoría.

ExtractProductoNombre: Elimina el prefijo "Codigo:" del nombre del producto.

ExtractProductoMarca: Elimina el prefijo "Marca N°" del código de marca.

Orden de Inserción de los Datos

Los inserts se han dispuesto en un orden específico para asegurar la integridad referencial y evitar errores de inserción debido a restricciones de clave foránea. Este orden se ha determinado considerando las dependencias entre las tablas y las funciones de extracción de datos.

Provincias y Localidades: Se insertan primero las provincias y luego las localidades asociadas ya que estas últimas dependen de las provincias.

Localidades y Supermercados: Se insertan antes los Supermercados ya que dependen de las Localidades.

Supermercados y Sucursales: Los supermercados se insertan antes que las sucursales, ya que las sucursales dependen de los supermercados y de las localidades. Además, se utiliza la función `dbo.ExtractSucursal` para limpiar los nombres de las sucursales.

Empleados y Sucursales: Se insertan los empleados y cómo estos están asignados a las sucursales, las sucursales se insertaron con anterioridad.

Tipo de Caja y Caja: Se insertan los tipos de caja y luego las cajas, ya que las cajas dependen de las sucursales y de los tipos de caja.

Clientes y Localidades: Se insertan los clientes junto con su información básica, utilizando las funciones de extracción de datos para limpiar los campos relevantes. Como estos dependen de las localidades, las mismas fueron insertadas con anterioridad en el esquema.

Tickets y Empleados, Cajas y Sucursales: Se insertan los tickets luego de haber insertado las tablas de Empleados, Cajas y Sucursales para evitar conflictos con las claves foráneas. Además se utilizan las funciones de extracción de datos para obtener datos coherentes.

Medios de Pago y Descuentos por medio de pago: Se insertan los medios de pago y luego los descuentos asociados a estos, utilizando las funciones de extracción de datos para limpiar los campos relevantes.

Pagos y Detalles de Pagos: Primero se insertan los pagos y luego los detalles ya que estos últimos dependen del pago y del cliente.

Envíos y Estados de Envío: Se insertan los envíos luego de sus estados asociados, utilizando las funciones de extracción de datos para limpiar los campos relevantes.

Categorías, Subcategorías y Categorías por Subcategorías: Se insertan las categorías y subcategorías antes que las Categorías por Subcategorías ya que, está última hace como intermediario entre las tablas ya que es una relación de muchos a muchos. Se utilizan las funciones de extracción de datos para obtener datos coherentes.

Reglas y Promociones: Se insertan las reglas antes que las promociones, ya que estas últimas pueden referirse a reglas específicas. Se utilizan las funciones de extracción de datos para obtener datos coherentes.

Marcas, Productos y Marcas por Producto: Las primeras dos tablas se insertan con antelación para evitar conflictos con las claves foráneas de marcas por productos.

Productos, Promociones y Rebajas de Productos: Dado a que las rebajas por producto están relacionadas a las promociones y los productos, primero se migran los datos de estas tablas para luego migrar las rebajas por cada producto

Producto, Ticket e Ítems de Tickets: Los ítems de los tickets son una de las últimas tablas en migrar ya que antes se necesita tener los datos de los Productos y los Tickets.

Promociones, Tickets y Promociones Aplicadas a Tickets: La última tabla en migrar es la de Promociones por ticket ya que está depende de las promociones y los tickets.

Este orden garantiza que todas las dependencias entre las tablas se respeten durante el proceso de inserción de datos, minimizando así los errores y asegurando la integridad de la base de datos.

Utilización de la cláusula max

Durante el proceso de migración de la base de datos, se identificó que tanto en la tabla Ticket como en la de Pago, existían múltiples ocurrencias para un mismo ticket o pago. Estas ocurrencias estaban distribuidas en varias líneas, donde cada línea contiene información parcial o complementaria de un mismo registro. Esto se debía a que ninguna línea tenía todos los campos completos, pero la suma de todas las líneas referidas a un mismo ticket/pago contenía toda la información necesaria.

Para consolidar esta información y asegurar que cada ticket o pago tenga una sola ocurrencia con todos los datos correspondientes, se decidió utilizar la función de agregación MAX en las consultas de inserción. Esta función permite seleccionar el valor máximo de cada campo dentro de un grupo de registros que comparten el mismo identificador de ticket/pago. Al hacerlo, garantizamos que se capture el dato más significativo para cada campo.

Estrategia de Diseño BI

Diseño de Dimensiones

Se crearon dimensiones para Ubicación, Tiempo, Cliente, Sucursal, Rango_Etario, Turno, Empleado, Categoría, Tipo Caja y Medio Pago.

Justificación: Estas dimensiones permiten un análisis multidimensional de los datos, facilitando la segmentación y la posibilidad de ver los datos con una mayor granularidad. Cada dimensión representa un aspecto importante del negocio que puede influir en las métricas de rendimiento, proporcionando una visión completa y detallada de las operaciones.

BI_Dim_Ubicacion

- ubicacion_id
- ubicacion_provincia
- ubicacion_localidad

BI_Dim_Tiempo

- tiempo_id
- tiempo_anio
- tiempo_mes
- tiempo_cuatrimestre

BI_Dim_Cliente

- cliente_id
- cliente_ubicacion
- cliente_rango_etario

BI_Dim_Sucursal

- sucursal_id
- sucursal_ubicacion

BI_Dim_Rango_Etario

- rango_id
- rango_descripcion

BI_Dim_Turno

- turno_id
- turno_hora_inicio
- turno_hora_fin

BI_Dim_Empleado

- empleado_id
- empleado_rango_etario
- empleado_sucursal

BI_Dim_Categoria

- categoria_id
- categoria_subcategoria
- categoria_nombre

BI_Dim_Tipo_Caja

- tipo_caja_id

BI_Dim_Medio_Pago

- medio_pago_id
- Medio_pago_tipo

Diseño de Tablas de Hechos

Se crearon tablas de hechos para Ventas, Promociones, Envíos y Pagos.

Justificación: Estas tablas capturan las métricas clave del negocio y las relacionan con las dimensiones relevantes. Esto permite un análisis completo y detallado de las operaciones del supermercado, facilitando la toma de decisiones estratégicas basadas en datos precisos y confiables.

BI_Fact_Ventas

- tiempo_id
- sucursal_id
- cliente_id
- empleado_id
- turno_id
- tipo_caja_id
- importe_total
- cantidad_ventas
- cantidad_unidades

BI_Fact_Promociones

- categoria_id
- tiempo_id

- monto_descuento_promocion
- ticket_total
- cantidad_tickets
- porcentaje_descuento

BI_Fact_Envios

- tiempo_id
- sucursal_id
- rango_id
- cliente_id
- costo_envio_total
- cantidad_envios
- envios_cumplidos

BI_Fact_Pagos

- tiempo_id
- medio_pago_id
- sucursal_id
- cliente_id
- rango_id
- importe_pago_en_cuotas
- importe_pago_total
- descuento_total

Granularidad de los Datos

Decisión: Se mantuvo una granularidad detallada en las tablas de hechos, llegando al nivel de ticket individual en algunos casos.

Justificación: Esta granularidad permite análisis más detallados y flexibles, pudiendo agregar los datos a niveles superiores según sea necesario para diferentes tipos de análisis. Esto es esencial para entender patrones y tendencias en las operaciones diarias y a lo largo del tiempo.

Uso de Identificadores Compuestos

Decisión: Se utilizaron identificadores compuestos en algunas dimensiones y hechos.

Justificación: Los identificadores compuestos permiten una fácil interpretación y manejo de los datos. Además, generan una clave primaria única para cada dimensión, asegurando la unicidad y consistencia de los datos en el modelo de BI.

Cálculos Precargados

Decisión: Se utilizan funciones para pre calcular ciertos datos.

Justificación: Esto mejora la eficiencia del sistema al reducir la carga de procesamiento en tiempo real. Los datos pre calculados permiten respuestas más rápidas a las consultas y mejoran el rendimiento general del sistema.

Las funciones realizadas son:

- ExtractSucursal;
- ExtractIIBB;
- ExtractSubCategoria;
- ExtractProductoNombre;
- ExtractProductoMarca;
- ExtractCategoria;
- ExtractRangoEtario;

Diseño de Vistas

Decisión: Se crearon vistas para responder a preguntas específicas del negocio.

Justificación: Las vistas encapsulan lógica de negocio compleja y proporcionan una capa de abstracción que facilita el análisis y la generación de informes sin necesidad de escribir consultas complejas cada vez. Esto agiliza el proceso de toma de decisiones y mejora la accesibilidad a la información relevante.

Vistas creadas:

- BI_VW_PorcentajeDescuentoPorMedioPago;
- BI_VW_PromedioImporteCuotaPorRangoEtarioCliente;
- BI_VW_Top3SucursalesPagosCuotas;
- BI_VW_Top5LocalidadesCostoEnvio;
- BI_VW_EnviosPorRangoEtarioCliente;
- BI_VW_CumplimientoEnvios;
- BI_VW_Top3CategoriasDescuento;
- BI_VW_PorcentajeDescuentoAplicado;
- BI_VW_VentasPorTurnoLocalidad;
- BI_VW_PorcentajeVentasRangoEtarioEmpleado;
- BI_VW_UnidadesPromedioPorTurno;
- BI_VW_TicketPromedioMensual;

Manejo de Temporalidad

Decisión: Se incluyó una dimensión de tiempo que permite análisis por año, mes y cuatrimestre.

Justificación: Esta estructura temporal facilita el análisis de tendencias y la comparación de rendimiento en diferentes períodos, crucial para la toma de decisiones estratégicas. Permite un análisis detallado y preciso del desempeño a lo largo del tiempo.

Conclusión

El diseño de esta base de datos se ha realizado con un enfoque en la integridad de los datos, la eficiencia en el almacenamiento y la claridad en las relaciones entre entidades. Las decisiones tomadas están alineadas con las buenas prácticas en el diseño de bases de datos relacionales y buscan proporcionar una base sólida para las operaciones y el análisis de datos en el contexto de la gestión de supermercados.

La estructura de Business Intelligence (BI) desarrollada sobre esta base de datos operacional permite un análisis profundo y multidimensional de los datos del negocio. Las dimensiones y tablas de hechos diseñadas facilitan la generación de informes detallados y el descubrimiento de insights valiosos para la toma de decisiones estratégicas. Las vistas creadas proporcionan respuestas rápidas a preguntas de negocio comunes, mientras que la granularidad detallada de los datos permite análisis ad-hoc más específicos cuando sea necesario.

Este diseño no solo cumple con los requisitos actuales del negocio, sino que también proporciona la flexibilidad necesaria para adaptarse a futuros cambios y expansiones en las necesidades analíticas de la empresa.