



**Universidad Tecnológica Nacional - Facultad Regional Buenos Aires**

**Trabajo Práctico Gestión de Datos**

**Tema:** Estrategia

**Curso:** K3012

**Profesor:** Enrique José Reinosa

**Fecha Estipulada de Entrega: 30/06/2024**

Alumno	Legajo
Lautaro Matías Romero Galván	1762783

	Fecha	Nota	Observaciones	Fecha Devol.	Firma Docente
Entrega	30-06	Aprobado		8/7	
Corrección					

Tabla de contenido

DIAGRAMA ENTIDAD RELACION ..... 1

    PRODUCTOS ..... 1

    TICKETS..... 1

    PAGOS ..... 1

    DESCUENTOS POR MEDIOS DE PAGO ..... 1

    MEDIOS DE PAGO..... 1

    PRODUCTO POR CATEGORIA/SUBCATEGORIA ..... 1

    PRODUCTO POR MARCA ..... 1

    PROVINCIA, LOCALIDAD, ESTADO: ..... 1

    TICKET DETALLE:..... 1

    PROMO\_APLICADA..... 2

SCRIPT MIGRACION INICIAL ..... 2

    SCRIPT..... 2

    Stored Procedures..... 2

SCRIPT MIGRACION BI ..... 3

    Funciones: ..... 3

    Tablas Dimensionales:..... 3

    Tablas Fácticas:..... 3

    Procedimientos de Migración: ..... 3

    Vistas: ..... 4

## DIAGRAMA ENTIDAD RELACION

**PRODUCTOS:** Al analizar la tabla maestra nos dimos cuenta que el nombre de un producto puede tener distintas marcas y, a su vez, distintas categorías/subcategorías. Debido a esto, tuvimos que hacer una relación de “muchos a muchos” entre ellas. Y a la relación entre producto y marca le definimos un código que será el código único de ese producto de esa marca. Por ejemplo, para el momento de pasarlo por caja que automáticamente el sistema se dé cuenta de que marca es el mismo. De la misma forma, implementamos dos tablas intermedias entre categoría, subcategoría y productos debido a su “relación de muchos a muchos” con éstas.

**TICKETS:** Al principio quisimos poner como PK la combinación de PK(numero\_ticket, ticket\_sucursal, ticket\_tipo\_comprobante) pero tuvimos problemas ya que, en ciertos tickets, se repetía la combinación cuando cada repetición contiene información distinta, por lo que decidimos sumar a caja\_numero a la combinación final de PK.

**PAGOS:** En cuanto a los pagos, partiendo de la base que un ticket puede tener distintos pagos, decidimos a cada pago individual asignarle un código único y así poder discriminar entre ambos al momento de necesitarlos. Por ejemplo, al hacer un pago en efectivo y uno con tarjeta, la tabla “pago\_tarjeta” tendría la FK del número de pago específico que se realizó con la tarjeta y no del de efectivo.

**DESCUENTOS POR MEDIOS DE PAGO:** Cada pago realizado con un medio se puede relacionar con un descuento si el medio de pago esta relacionado con el medio descuento a aplicar.

**MEDIOS DE PAGO:** Creamos una tabla aparte para medios de pago a la que le asignamos un código para poder usarlo como una FK en distintas tablas que lo contienen y no repetir 2 filas (medio\_pago, tipo\_medio\_pago).

**PRODUCTO POR CATEGORIA/SUBCATEGORIA:** En principio no creamos esta “relación de muchos a muchos” pero nos vimos obligados al notar que un mismo producto podía tener distintas categorías/subcategorías. A su vez, una categoría podía ser padre de muchas subcategorías y la misma subcategoría hija de varias categorías.

**PRODUCTO POR MARCA:** En la tabla maestra un mismo producto\_nombre podía tener varias marcas asique, supusimos que con “producto\_nombre” se referia a un nombre general (por ejemplo: “manteca”). De esta forma, decidimos que cada combinación de producto y marca sea una tabla distinta a la cual le asignamos un código a cada combinación. De esta manera, un producto de una marca especifica tendrá asignado un código y así el sistema detecte automáticamente que el código del producto es de una marca especifica (por ejemplo: Codigo: 1, “manteca LaPaulina”).

**PROVINCIA, LOCALIDAD, ESTADO:** Decidimos pasar cada uno a una tabla independiente ya que eran, por defecto, un varchar(255). Dentro de cada tabla le asignamos un código decimal(18,0) para que, cada vez que se quiera referenciar a una de ella desde otra tabla usen un decimal(18,0) y no un varchar(255) que ocupa más memoria por fila.

**TICKET DETALLE:** Aquí solo referenciamos al ticket al cual pertenece y, como agregado, se mostrarán los ProductosXmarca comprados, la cantidad, el precio y el total de los mismos.

**PROMO APLICADA:** Para cada producto se pueden aplicar una/varias promoción/es por lo que en “promo\_aplicada” se registra el ticket al que pertenece, el producto al cual se le aplico la promoción y la cantidad descontada. De esa manera logramos que un producto tenga varias promociones aplicadas y, a su vez, una promoción sea aplicada en distintos productos en distintos tickets.

## SCRIPT MIGRACION INICIAL

**SCRIPT:** El script procede a crear el schema LAMBDA y, en caso de existir tablas con los nombres que crearemos, las elimina para darle paso a la creación de las mismas nuevamente.

Luego se crean las tablas con su estructura correspondiente (proporcional al DER) y los stored procedures para la migración:

### Stored Procedures:

1. **migrar\_localidades:** Inserta localidades únicas en la tabla **LAMBDA.Localidad** que no existan ya en la misma.
2. **migrar\_provincias:** Inserta provincias únicas en la tabla **LAMBDA.Provincia** que no existan ya en la misma.
3. **migrar\_supers:** Migra datos de entidades 'super' desde **gd\_esquema.Maestra** a **LAMBDA.Super**, asegurando que los códigos de localidad y provincia se correspondan con los nombres.
4. **migrar\_sucursales:** Inserta sucursales en **LAMBDA.Sucursal**, ajustando el formato del nombre de la sucursal y enlazando con los códigos de localidad y provincia.
5. **migrar\_empleados:** Migra datos de empleados ordenados por la fecha de registro a **LAMBDA.Empleado**.
6. **migrar\_cajas:** Utiliza la instrucción **MERGE** para insertar cajas en **LAMBDA.Caja**, evitando duplicados basados en el número de caja y sucursal.
7. **migrar\_estados:** Inserta estados únicos de envíos en **LAMBDA.Estado**.
8. **migrar\_tickets:** Inserta tickets en **LAMBDA.Ticket**, enlazando con empleados y asegurando la integridad referencial.
9. **migrar\_clientes:** Migra datos de clientes a **LAMBDA.Cliente**, enlazando con códigos de localidad y provincia y ordenando por la fecha de registro.
10. **migrar\_envios:** Inserta envíos en **LAMBDA.Envio**, asegurando la correlación con estados, clientes y tickets existentes.
11. **migrar\_categorias:** Inserta códigos únicos de categorías en **LAMBDA.Categoria**.
12. **migrar\_subcategorias:** Inserta códigos únicos de subcategorías en **LAMBDA.SubCategoria**.
13. **migrar\_categorias\_subcategorias:** Relaciona categorías y subcategorías en **LAMBDA.Categoria\_SubCategoria**.
14. **migrar\_productos\_categorias:** Relaciona productos con sus categorías en **LAMBDA.Producto\_Categoria**.
15. **migrar\_productos\_subcategorias:** Relaciona productos con sus subcategorías en **LAMBDA.Producto\_SubCategoria**.
16. **migrar\_marcas:** Inserta códigos únicos de marcas en **LAMBDA.Marca**.
17. **migrar\_productos:** Inserta nombres y descripciones de productos en **LAMBDA.Producto**.
18. **migrar\_productos\_marcas:** Relaciona productos con sus marcas y precios máximos en **LAMBDA.Producto\_Marca**.

19. **migrar\_ticket\_detalle**: Inserta detalles de tickets en **LAMBDA.Ticket\_Detalle**, agrupando y sumando cantidades y totales por producto.
20. **migrar\_promociones**: Inserta promociones únicas en **LAMBDA.Promocion**.
21. **migrar\_reglas**: Inserta reglas de promociones en **LAMBDA.Regla**.
22. **migrar\_promos\_productos**: Relaciona productos con sus promociones en **LAMBDA.Promo\_Producto**.
23. **migrar\_promos\_aplicadas**: Inserta promociones aplicadas a tickets en **LAMBDA.Promo\_Aplicada**.
24. **migrar\_medios\_pago**: Inserta medios de pago únicos en **LAMBDA.Medio\_Pago**.
25. **migrar\_descuentos**: Inserta descuentos en **LAMBDA.Descuento**, relacionándolos con los medios de pago.
26. **migrar\_pagos**: Inserta pagos en **LAMBDA.Pago**, asegurando la integridad con tickets y medios de pago.
27. **migrar\_pagos\_tarjetas**: Relaciona los pagos que están hechos con tarjeta a la misma junto con la información del cliente que hizo el pago.

## SCRIPT MIGRACION BI

### Funciones:

1. **getAge**: Calcula la edad actual a partir de la fecha de nacimiento.
2. **getAgeRange**: Devuelve un rango de edad basado en la edad proporcionada.
3. **getTurn**: Determina el código de turno según la hora del día.
4. **fn\_Cuatrimestre**: Asigna un número de cuatrimestre a una fecha dada.

### Tablas Dimensionales:

1. **BI\_dim\_tiempos**: Detalla años, cuatrimestres y meses.
2. **BI\_dim\_localidades**: Lista de localidades.
3. **BI\_dim\_provincias**: Lista de provincias.
4. **BI\_dim\_sucursales**: Detalle de sucursales con referencias a localidades y provincias.
5. **BI\_dim\_cajas**: Información sobre cajas en sucursales.
6. **BI\_dim\_clientes**: Detalle de clientes con datos personales y referencias geográficas.
7. **BI\_dim\_rangos\_etarios**: Define rangos de edad.
8. **BI\_dim\_turnos**: Descripción de turnos laborales.
9. **BI\_dim\_medios\_pago**: Tipos y detalles de medios de pago.
10. **BI\_dim\_categorias**: Categorías de productos.

### Tablas Fácticas:

1. **BI\_fact\_ventas**: Datos de ventas por sucursal, tiempo, turno, empleado y caja.
2. **BI\_fact\_envios**: Información de envíos por sucursal, tiempo, cliente y cumplimiento.
3. **BI\_fact\_pagos**: Detalle de pagos por sucursal, tiempo, método de pago y cliente.
4. **BI\_fact\_descuentos**: Descuentos aplicados en ventas por tiempo, categoría de producto y medio de pago.

### Procedimientos de Migración:

1. Procedimientos para migrar datos de tablas operativas a tablas dimensionales:

- **BI\_migrar\_tiempos**
- **BI\_migrar\_localidades**
- **BI\_migrar\_provincias**
- **BI\_migrar\_sucursales**
- **BI\_migrar\_cajas**
- **BI\_migrar\_clientes**
- **BI\_migrar\_rangos\_etarios**
- **BI\_migrar\_turnos**
- **BI\_migrar\_medios\_pago**
- **BI\_migrar\_categorias**

#### Vistas:

1. **v\_ticketPromedioMensual:** Promedio mensual de precios de tickets por localidad.
2. **v\_cantUnidadesPromedio:** Promedio de cantidad de productos vendidos por turno y cuatrimestre.
3. **v\_porcVentasxEdad:** Porcentaje de ventas por rango de edad y tipo de caja.
4. **v\_cantVentasTurnoLocalidad:** Cantidad de ventas por turno y localidad mensualmente.
5. **v\_porcentajeDescuentoMes:** Porcentaje promedio de descuento aplicado mensualmente.
6. **v\_top3CategsProdsMayorDesc:** Top 3 de categorías de productos con mayor descuento aplicado.
7. **v\_porcEnviosCumplidos:** Porcentaje de envíos cumplidos por sucursal mensualmente.
8. **v\_cantEnvRangoEtCuatriAño:** Cantidad de envíos por rango de edad, cuatrimestre y año.
9. **v\_top5LocalidadesEnvioCaro:** Top 5 de localidades con mayor costo total de envío.
10. **v\_top3SucursalesMayorPagoCuotas:** Top 3 de sucursales con mayor importe total de cuotas pagadas.
11. **v\_promedioCuotaEdadCliente:** Promedio de cuotas por edad y cliente.