



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

**Trabajo Práctico Grupal**

**Tema:** Estrategia

**Curso:** K3012

**Profesor:** Enrique José Reinosa

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

**Grupo:** LAMBDA (35)

| Alumno                       | Legajo  |
|------------------------------|---------|
| Matias Goldfarb              | 2045898 |
| Theo Dyzenchauz              | 2035832 |
| Uriel Damian Liahaff         | 2027203 |
| Lautaro Matías Romero Galván | 1762783 |

|            | Fecha | Nota | Observaciones | Fecha Devol. | Firma Docente |
|------------|-------|------|---------------|--------------|---------------|
| Entrega    | 30-06 |      |               |              |               |
| 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

    1. Funciones Auxiliares..... 3

    2. Tablas Dimensionales..... 3

    3. Tablas Puente ..... 4

    4. Tablas Fácticas..... 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

### 1. Funciones Auxiliares

Las funciones auxiliares (`getAge`, `getAgeRange`, `getTurn`, `fn_Cuatrimestre`) se crean para facilitar cálculos y transformaciones comunes, asegurando consistencia y simplicidad en el código de la base de datos:

- **getAge**: Calcula la edad actual de una persona a partir de su fecha de nacimiento.
- **getAgeRange**: Determina el rango etario al que pertenece una persona dada su edad.
- **getTurn**: Identifica el turno del día (mañana, tarde, noche) basado en una hora específica.
- **fn\_Cuatrimestre**: Determina el cuatrimestre del año basado en una fecha específica.

Estas funciones permiten realizar transformaciones comunes directamente en las consultas SQL, simplificando el proceso de extracción, transformación y carga (ETL).

### 2. Tablas Dimensionales

Las tablas dimensionales contienen atributos descriptivos relacionados con las entidades del negocio, proporcionando contexto a los datos fácticos:

- **BI\_dim\_tiempos**: Contiene información temporal (año, cuatrimestre, mes).
- **BI\_dim\_localidades** y **BI\_dim\_provincias**: Detallan la ubicación geográfica.
- **BI\_dim\_sucursales**: Describe las sucursales, incluyendo direcciones y relaciones con localidades y provincias.
- **BI\_dim\_clientes**: Información detallada de los clientes, incluyendo datos personales y de ubicación.
- **BI\_dim\_estados**: Estado de los envíos.
- **BI\_dim\_rangos\_etarios**: Rango de edad.
- **BI\_dim\_turnos**: Detalles de los turnos horarios.
- **BI\_dim\_medios\_pago**: Métodos de pago.
- **BI\_dim\_descuentos**: Detalles de descuentos.

- **BI\_dim\_productos, BI\_dim\_categorias, BI\_dim\_subcategorias:** Información de productos y sus clasificaciones.

### **3. Tablas Puente**

Las tablas puente (BI\_dim\_categoria\_subcategoria, BI\_dim\_producto\_categoria, BI\_dim\_producto\_subcategoria) se utilizan para gestionar las relaciones muchos-a-muchos entre las dimensiones:

- **BI\_dim\_categoria\_subcategoria:** Relaciona categorías y subcategorías.
- **BI\_dim\_producto\_categoria:** Relaciona productos y categorías.
- **BI\_dim\_producto\_subcategoria:** Relaciona productos y subcategorías.

Estas tablas permiten una mayor flexibilidad en las consultas analíticas y mantienen la integridad de las relaciones complejas.

### **4. Tablas Fácticas**

Las tablas fácticas almacenan las medidas cuantitativas del negocio y están relacionadas con las dimensiones para proporcionar contexto a estos datos:

- **BI\_fact\_ventas:** Registra las ventas, incluyendo detalles del producto, sucursal, tiempo, turno, y rangos etarios de los empleados.
- **BI\_fact\_envios:** Detalla los envíos, incluyendo información del cliente, estado del envío, y costos.
- **BI\_fact\_pagos:** Registra los pagos, incluyendo detalles del medio de pago, descuentos aplicados, y rangos etarios de los clientes.

Estas tablas permiten realizar análisis detallados de las transacciones y el comportamiento de los clientes.