



# **UNIVERSIDAD NACIONAL DE LA MATANZA**

## **INTELIGENCIA DE NEGOCIOS**

---

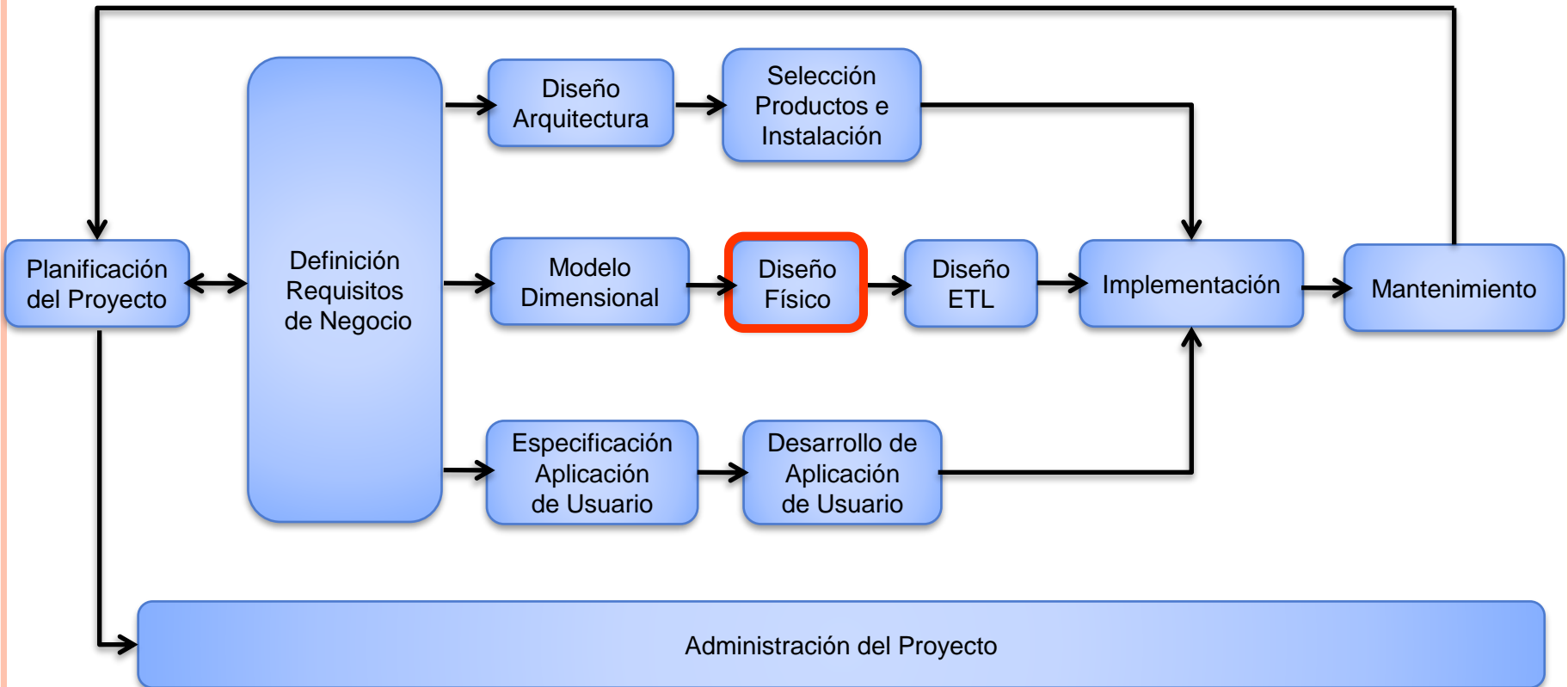
### **Modelo Físico**

### **Técnicas de Modelado - Parte 1**

**Docentes: ING. LORENA R. MATTEO**  
**Autores ppt orig.: Lic. HUGO M. CASTRO / MG. DIEGO BASSO**



# CICLO DE VIDA DE UN PROYECTO DE BI

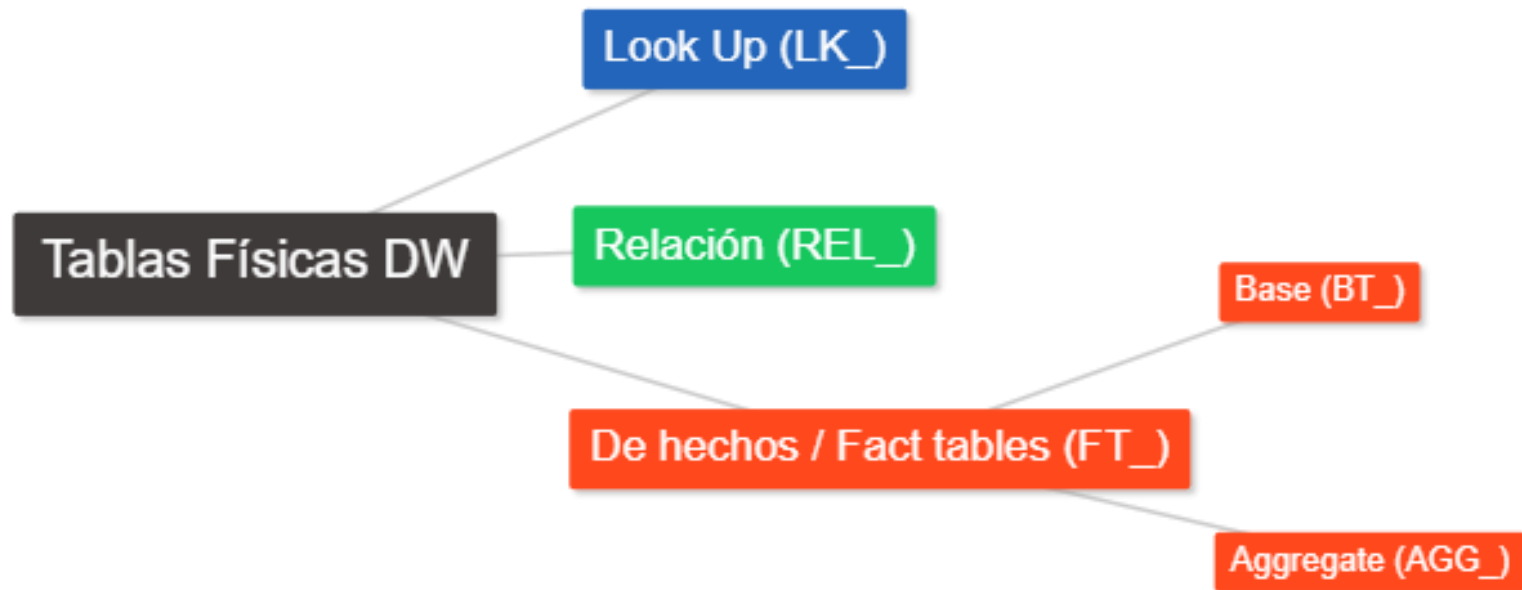




# MODELO FÍSICO

- A partir del modelo dimensional lógico debemos armar el modelo físico.
- Son las tablas en las que se van a guardar los datos en el Data Warehouse.
  - Soporte a las dimensiones y grupo de hechos del modelo dimensional.
  - Se compone de tablas, registros y columnas.
- Diferentes técnicas de modelado, según la realidad del negocio.

# ESTRUCTURA DE LAS TABLAS



# ESTRUCTURA DE TABLAS



## Tablas Look Up

- Son las tablas maestras donde se almacenan los elementos de un atributo descriptivo.
  - De tipo textual y discreto
  - Para seleccionar
  - Para agrupar
  - Para mostrar
- Se puede crear una tabla Look Up por cada **atributo** o una tabla por **dimensión**.
- No contiene valores que intervengan en cálculos.
- Valores numéricos categorizados.
- Tienen una columna por *attribute form* más una columna por cada *padre* del atributo.



# EJEMPLO TABLA LOOK UP

## ○ Provincia

- ID\_Provincia
  - Desc\_Provincia
- } *Attribute form*

1, Buenos Aires  
2, Córdoba  
3, La Pampa

} *Elementos*

LK\_PROVINCIA

ID_Provincia	Desc_Provincia
1	Buenos Aires
2	Córdoba
3	La Pampa
...	...

Una fila por **elemento**  
del atributo

Una columna por **attribute form**



# EJEMPLO TABLA LOOK UP

## ○ Ciudad

- ID\_Ciudad
- Desc\_Ciudad

Provincia



*Jerarquía de atributos*

## LK\_CIUDAD

ID_Ciudad	Desc_Ciudad	ID_Provincia
11	Junín	1
12	Lobos	1
23	La Falda	2
35	Cosquín	2
38	Santa Rosa	3

Una columna por  
**attribute form**

Una columna por  
cada **padre**

# ESTRUCTURA DE TABLAS



## Tablas Relación

- Se utilizan cuando hay una relación *muchos a muchos* entre dos atributos.

## Tabla de hechos (Fact table)

- Almacena los **valores** de las medidas de los hechos.
- Contiene una fila por cada acontecimiento que debe reflejar.
- Es la tabla de mayor cantidad de filas.
  - Ocupan más del 90% del Data Warehouse
- Tiene una columna por cada *medida* mas una columna por cada *referencia a las dimensiones* al cual se conocen los hechos.





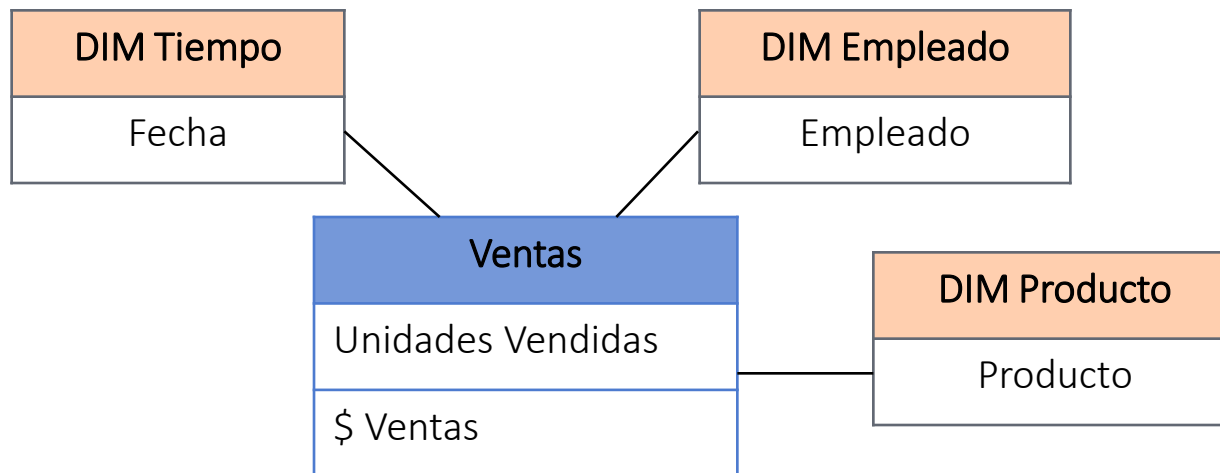
# ESTRUCTURA DE TABLAS

## Tabla de hechos **Base**

- Almacena los **valores** de las medidas de los hechos al *máximo nivel de detalle (Granularidad)*.
  - Esta tabla es imprescindible
- Sus filas se llenan a partir de los datos que provienen de los sistemas OLTP.



# EJEMPLO DE TABLA DE HECHOS BASE



## BT\_VENTAS

ID_Fecha	ID_Empleado	ID_Producto	Unid_Vend	\$ Ventas
01/04/2020	111	17	20	\$ 400
01/04/2020	112	17	45	\$ 900
01/04/2020	113	18	30	\$ 450
...	...	...	...	...

Dimensiones

Medidas

# ESTRUCTURA DE TABLAS



## Tabla de hechos **Agregadas**

- Se generan exclusivamente para reducir el tiempo de respuesta de las consultas.
- Su contenido se calcula a partir de la tabla de hechos base.
- Son prescindibles en términos de información.
  - No dicen nada que no sea dicho en la tabla de hechos base, pero lo dicen más rápido.
- En general, tienen pocas filas (respecto a la tabla de hechos base).



# EJEMPLO DE TABLA DE HECHOS AGREGADA

**BT\_VENTAS**

ID_Fecha	ID_Empleado	ID_Producto	Unid_Vend	\$ Ventas
01/04/2020	111	17	20	\$ 400
08/04/2020	112	17	45	\$ 900
15/04/2020	112	17	30	\$ 450
19/04/2020	111	18	15	\$ 300

**AGG\_VENTAS\_EMP\_MES**

ID_Mes	ID_Empleado	Unid_Vend	\$ Ventas
202004	111	35	\$ 700
202004	112	75	\$ 1350
...	...	...	...



Contiene un atributo de alguna dimensión, por ejemplo TIEMPO



# ESQUEMA ESTRELLA (STAR)

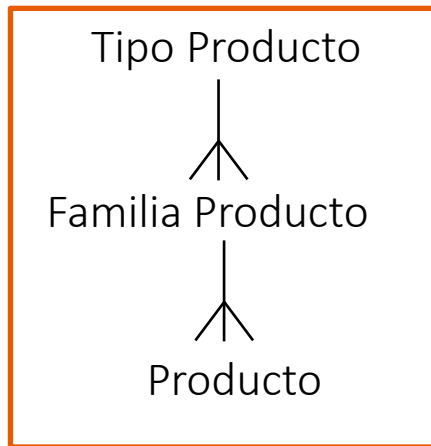
- Una tabla por cada dimensión.
- Menor cantidad de tablas involucradas.
- La tabla de hechos está en 3FN.
  - No tiene filas repetidas.
- Las tablas de dimensión están en 2FN.
  - Todos los productos de una misma familia llevan como atributo el nombre de la familia.
- Mayor espacio de almacenamiento.
- Las consultas son más fáciles de construir.
  - Joins entre tabla de hechos con cada tabla de dimensiones.
- Mejor rendimiento en consultas.
  - Se necesita acceder a menos tablas.

# ESQUEMA ESTRELLA (STAR)



- Producto
  - ID\_Producto
  - Desc\_Producto
- Familia\_Producto
  - ID\_FamiliaProducto
  - Desc\_Familia
- Tipo\_Producto
  - ID\_TipoProducto
  - Desc\_TipoProducto

DIM\_PRODUCTO

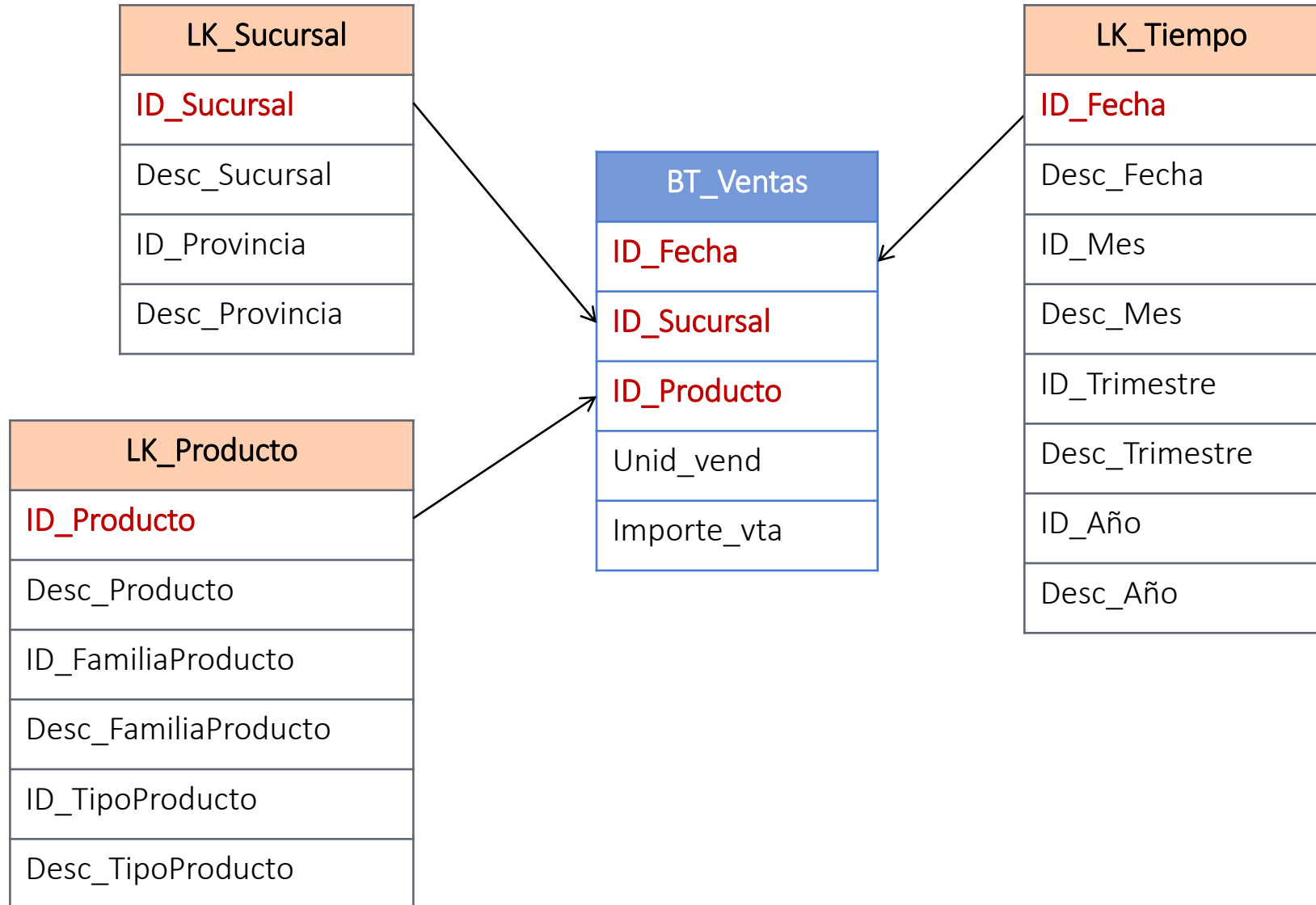


LK_Producto
ID_Producto
Desc_Producto
ID_FamiliaProducto
Desc_FamiliaProducto
ID_TipoProducto
Desc_TipoProducto

BT_Ventas
ID_Fecha
ID_Sucursal
ID_Empleado
ID_Producto
Unid_Vend
Importe_Vta



# EJEMPLO ESQUEMA ESTRELLA





# CONSULTA SQL A UN MODELO ESTRELLA

- Ejemplo: Para cada producto de tipo bazar calcular el importe vendido en las sucursales de CABA en el mes de febrero de 2020
- Dimensiones para:
  - Mostrar, filtrar, agrupar
- Medidas o Hechos para:
  - Calcular
- Join: Sólo entre tabla de hechos y tabla de dimensión





# CONSULTA SQL A UN MODELO ESTRELLA

```
SELECT p.Desc_Producto, sum(v.Importe_vta) AS Importe
FROM
    BT_Ventas v,
    LK_Sucursal s,
    LK_Producto p,
    LK_Tiempo t
WHERE
    v.ID_Fecha = t.ID_Fecha and
    v.ID_Sucursal = s.ID_Sucursal and
    v.ID_Producto = p.ID_Producto and
    s.Desc_Provincia = 'CABA' and
    t.ID_Mes = '022020' and
    p.Desc_TipoProducto = 'Bazar'
GROUP BY p.Desc_Producto
```

# ESQUEMA COPO DE NIEVE (SNOWFLAKE)



- Una tabla por atributo.
- Todas las tablas (dimensiones y hechos) están en 3FN.
  - No tiene filas repetidas.
- Menor espacio de almacenamiento.
- Más fácil de entender para el usuario final.
- Más fáciles de llenar las tablas.
- Menor rendimiento de las consultas por los join.
- Aplicable para tablas de dimensiones con una gran cantidad de filas.
- Ampliamente usado en implementaciones ROLAP.

# ESQUEMA COPO DE NIEVE (SNOWFLAKE)



- Producto
  - Producto\_ID
  - Desc\_producto
- Familia\_Producto
  - ID\_FamiliaProducto
  - Desc\_Familia
- Tipo\_Producto
  - ID\_TipoProducto
  - Desc\_TipoProducto

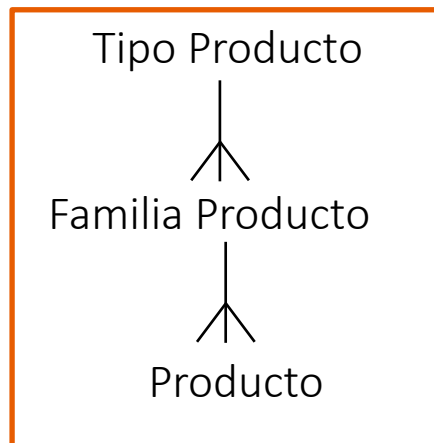
LK_TipoProducto
ID_TipoProducto
Desc_TipoProducto

LK_FamiliaProducto
ID_FamiliaProducto
Desc_FamiliaProducto
ID_TipoProducto

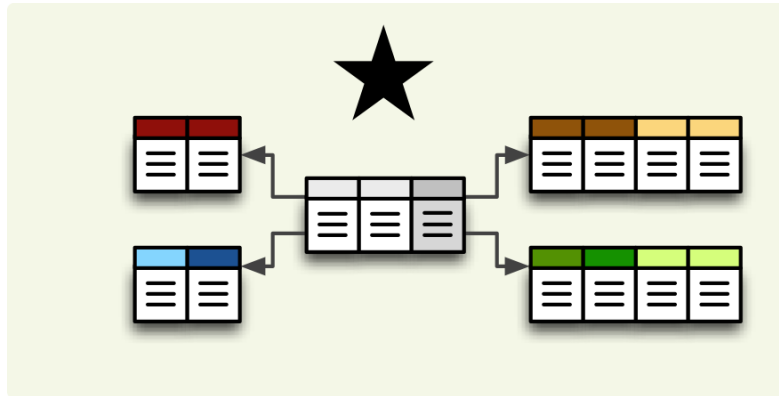
LK_Producto
ID_Producto
Desc_Producto
ID_FamiliaProducto

BT_Ventas
ID_Fecha
ID_Sucursal
ID_Empleado
ID_Producto
Unid_Vend
Importe_Vta

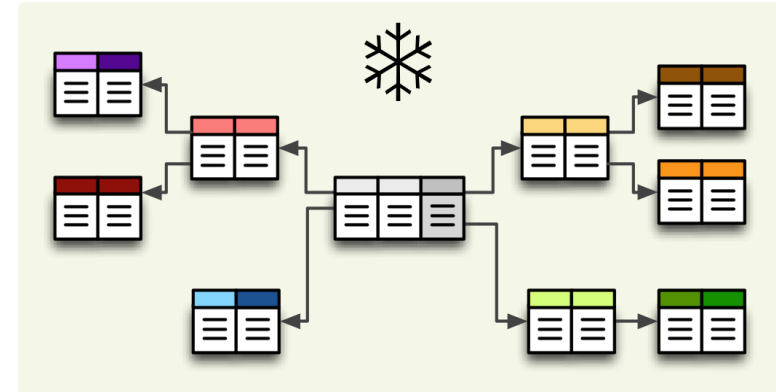
## DIM\_PRODUCTO



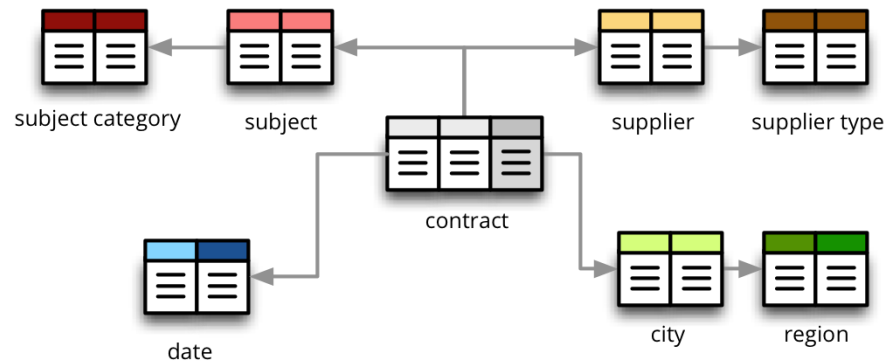
# ESQUEMAS STAR VS SNOWFLAKE



Modelo Estrella



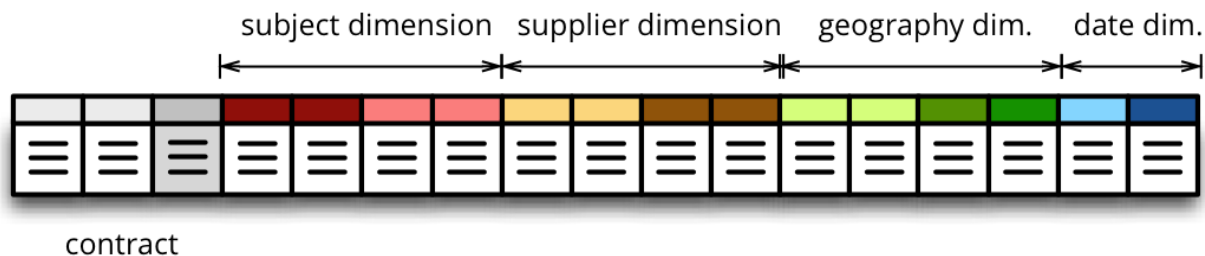
Modelo Copo de Nieve



Es posible pasar de este esquema **Copo de Nieve** ...



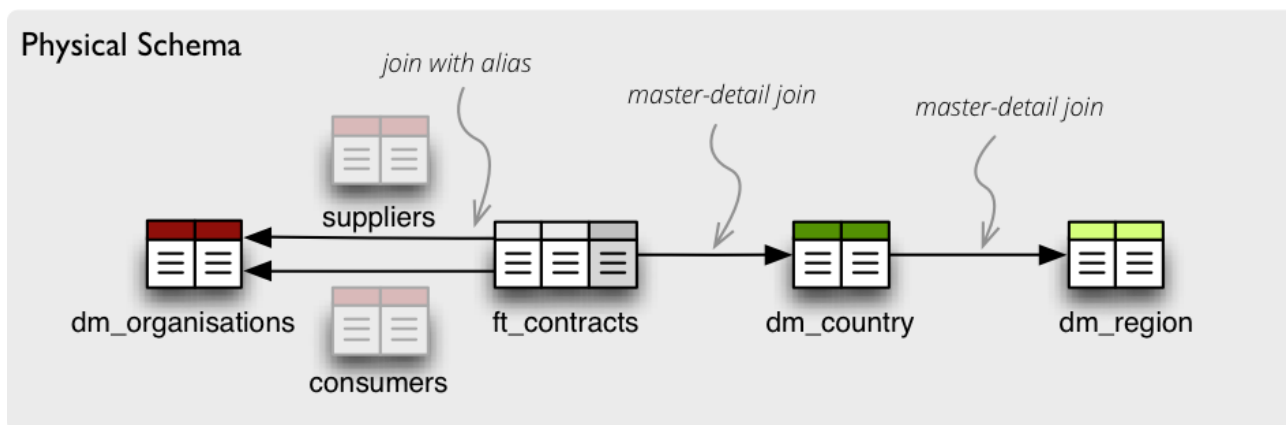
... a este en **Estrella** (desnormalizado) con todos los atributos del cubo entendiendo como relacionan las tablas a través de las **SKs**.





# ESQUEMAS STAR VS SNOWFLAKE - ALIAS

- ¿Qué sucede si necesita unirse a la misma tabla dos o más veces?
  - Por ejemplo, tiene una lista de organizaciones y es necesario usarse como proveedor y consumidor de servicios.



Especificar Alias en los joins (pseudo):

```
"joins" =
[
  { "master" : "contratos.supplier_id",
    "detalle" : "organizaciones.id",
    "alias" : "proveedores"
  },
  { "master" : "contratos.consumer_id",
    "detalle" : "organizaciones.id",
    "alias" : "consumidores"
  }
]
```

Fuente: <http://databrewery.org/cubes/doc/mapping.html#joins>

Usar el alias especificado en los joins, y no el nombre real de la tabla.

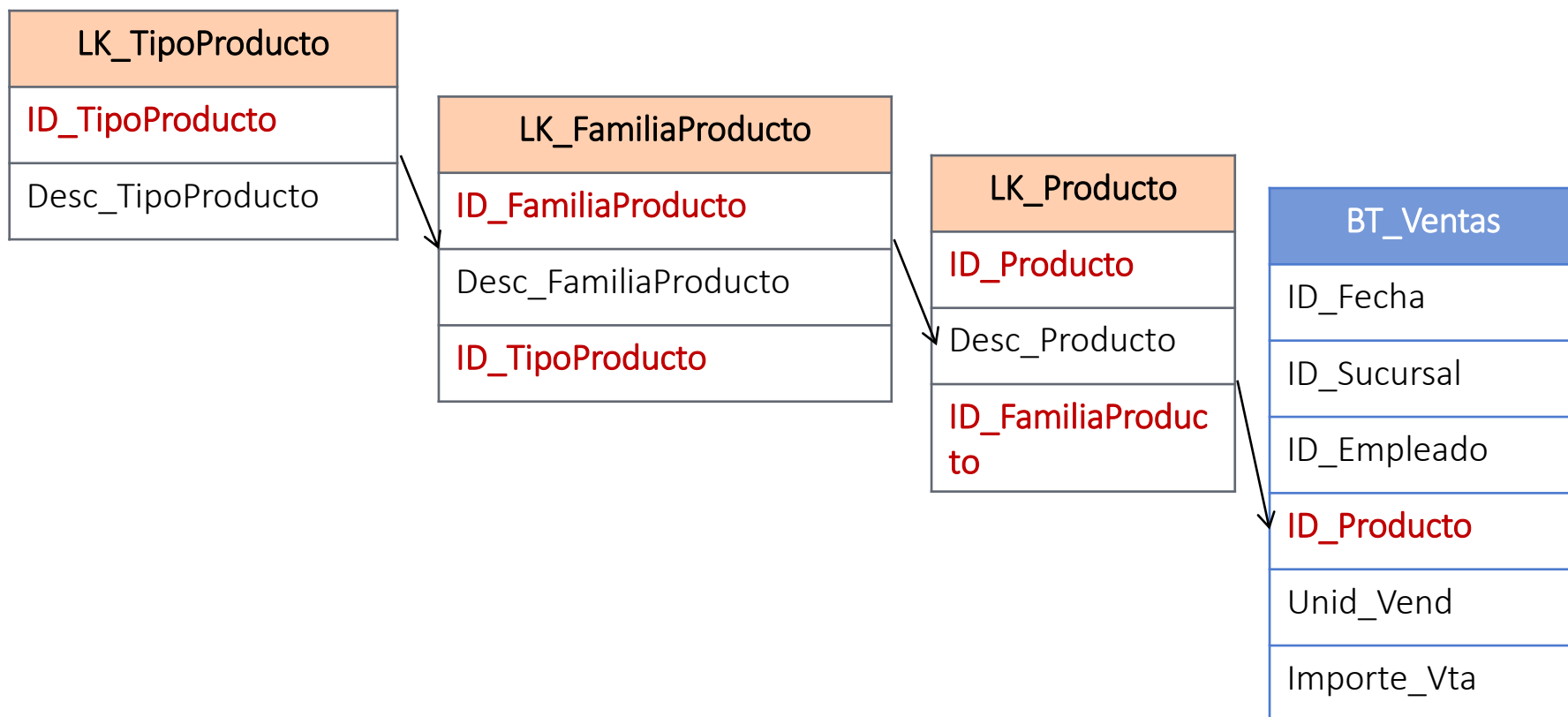
```
"asignaciones" :
{
  "proveedor.nombre" :
  "proveedores.org_nombre",
  "consumidor.nombre" :
  "consumidores.org_nombre"
}
```



# NORMALIZACIÓN

## ○ Completamente Normalizado

- Las tablas Look Up tienen el ID propio, la descripción y el ID del padre.
- Minimiza la redundancia.
- Muchos joins para acceder a las tablas de más alto nivel.

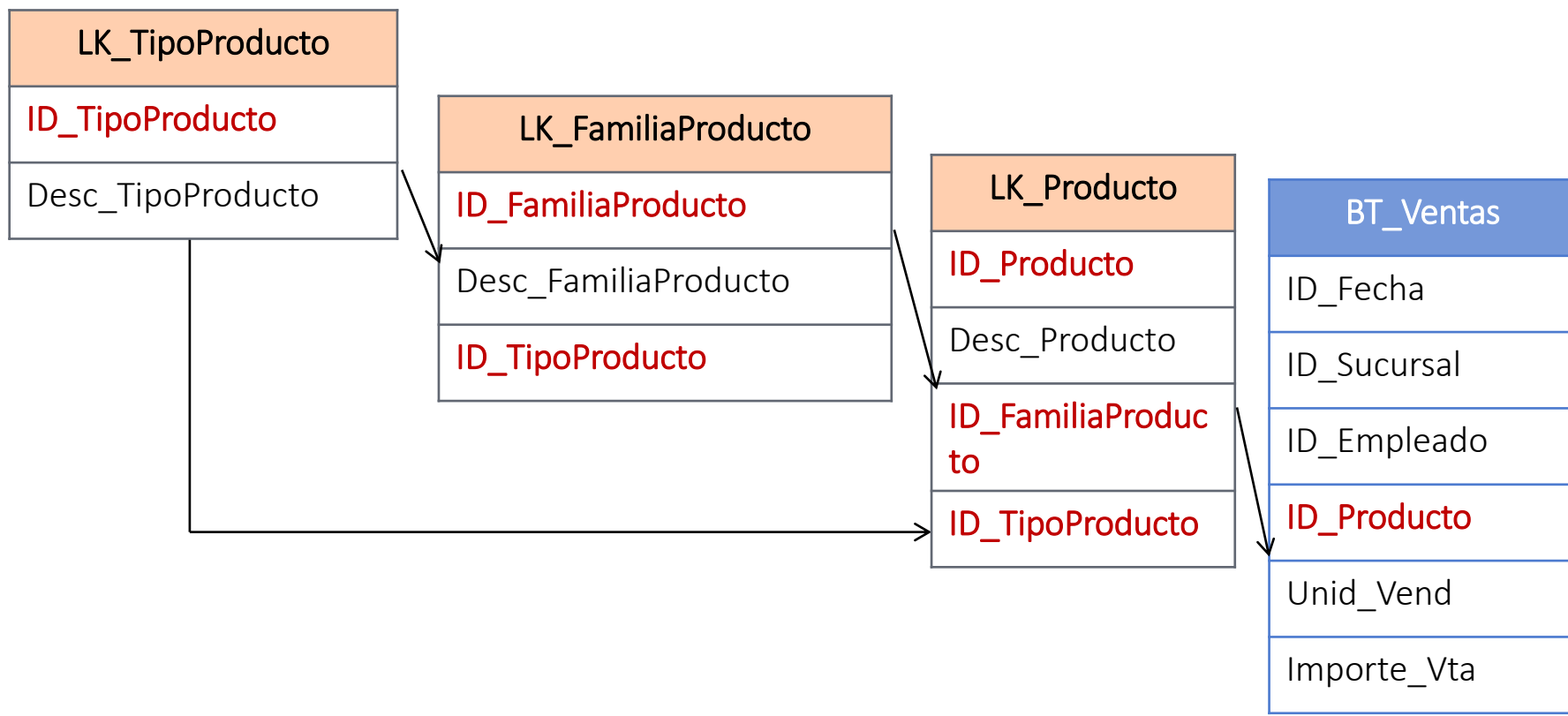




# NORMALIZACIÓN

## ○ Moderadamente Normalizado

- Las tablas Look Up tienen el ID propio, la descripción y todos los ID de sus ancestros.
- Algo de redundancia.
- Reduce significativamente los joins para consultar datos dentro de la jerarquía.

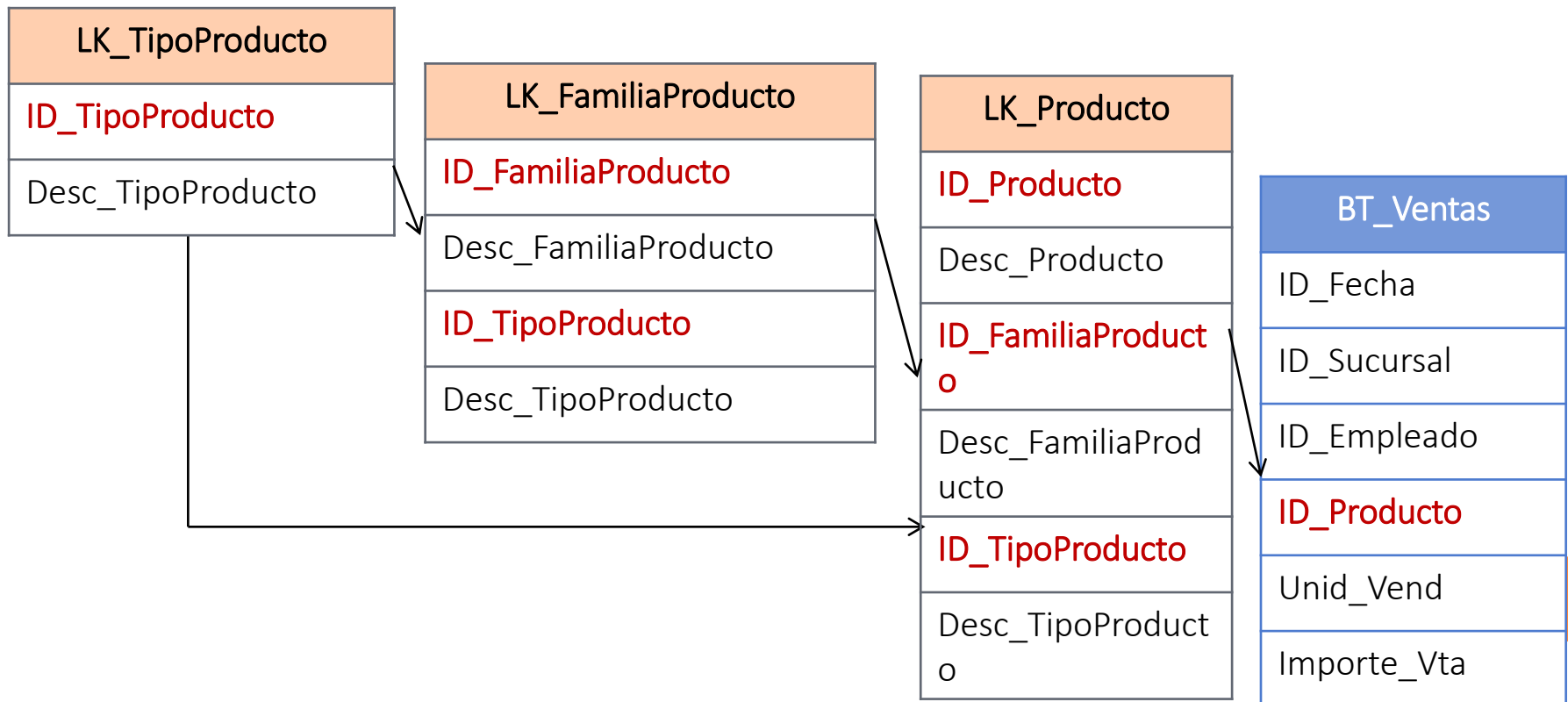




# NORMALIZACIÓN

## ○ Completamente Desnormalizado

- Las tablas Look Up tienen el ID propio, la descripción y todos los ID y descripción de sus ancestros.
- Mucha redundancia y espacio de almacenamiento.
- Elimina los joins para consultar datos dentro de la jerarquía.







# TIPOS DE MEDIDAS O HECHOS

## ○ Aditivas

- Se pueden sumar (SUM) por todas las dimensiones.

## ○ Ejemplos

- **Importe de venta**
  - Tiene sentido sumar el importe por producto, por sucursal, por fecha, por empleado.
- **Costo de venta**
- **Ganancia o beneficio**
- **Cantidad de ventas (en el día)**
- **Cantidad de expedientes procesados (en una semana)**
- **Cantidad de nacimientos (por mes)**



# TIPOS DE MEDIDAS O HECHOS

## ○ Semiaditivas

- Se pueden sumar en algunas dimensiones y en otras no.
- Carece de sentido sumarla en otras dimensiones.

## ○ Ejemplos

- Unidades Vendidas
  - Tiene sentido sumar sólo si está la dimensión *Producto*.
- Unidades en Stock (o inventarios de cualquier tipo)
  - Sólo se pueden sumar si son del mismo día, mes, año, etc.
  - No se puede sumar por *producto*  $\Rightarrow$  semánticamente incorrecto.
- Saldo de Cuentas
  - Sólo se puede sumar si son del mismo día. Se obtiene el saldo x día.
  - Normalmente se obtiene a nivel del último día del mes.
- Número de clientes
  - No se puede sumar por *producto*  $\Rightarrow$  semánticamente incorrecto.



# TIPOS DE MEDIDAS O HECHOS

- No aditivas
  - No tiene sentido sumarlas por ninguna dimensión.
  - Usan otras formas de consolidación o agregación.
- Ejemplos
  - Precio Unitario
    - No tiene sentido si lo sumo por día, producto o sucursal.
  - Margen o porcentaje de ganancia
    - No se puede sumarizar por ninguna dimensión.
    - Es una medida calculada obtenida con la herramienta OLAP.
  - Temperatura
  - Edad
  - Notas de exámenes
  - Tipo de cambio de moneda



# FORMAS DE CONSOLIDACIÓN DE MEDIDAS

- Suma
- Promedio
- Máximo
- Mínimo
- Cantidad de casos
- Cantidad de casos distintos

# ¿COMO SE CONECTA LA TABLA DE HECHOS A LAS DE DIMENSIÓN?



- Verificar **integridad referencial** entre la tabla de hechos y las tablas de dimensión.
- En la tabla de dimensión:
  - La clave tiene que ser una clave primaria (PK)
- En la tabla de hechos:
  - Cada dimensión tiene su clave foránea (FK) que apunta a la fila que corresponde en la tabla de dimensión.
  - Todas las FK de las dimensiones forman la PK de la tabla de hechos.

# CLAVE SUBROGADA (SURROGATE KEY - SK)



## Problemas con las claves

- Los sistemas fuentes utilizan sus propias claves (código de artículo, código de cliente, etc.), con un determinado formato y sentido para el negocio.
- Podrían cambiarse los formatos de las claves de los sistemas fuentes.
- Los datos de un DW vienen de fuentes heterogéneas.
- Las SK generan dentro del ámbito del DW una clave numérica única sin significado para el negocio.
  - Número entero asignado en forma secuencial.
- Aconsejable no usar como claves los códigos de los sistemas fuentes.

# CLAVE SUBROGADA (SURROGATE KEY - SK)



## Se necesita crear una SK en un DW

- Los sistemas fuentes pueden cambiar la descripción de un producto o cliente, sin cambiar el código del DW que lo representa.
- Si se cambian los formatos de las claves en los sistemas fuentes, la clave SK no se altera.
- Los sistemas fuentes pueden reutilizar códigos que se habían depurado y aún existen en el DW.
- Rendimiento
  - Las claves de las tablas ocuparán menos espacio.
- Más fáciles de mantener.

# CLAVE SUBROGADA (SURROGATE KEY - SK)



LK_Producto
ID_Producto
Cod_Producto
Desc_producto
....

BT_Ventas
ID_Fecha
ID_Sucursal
ID_Empleado
ID_Producto
Unid_Vend
Importe_Vta

Sólo se usa para  
conectar las tablas.

No tiene significado  
para el negocio



# CLAVE SUBROGADA (SURROGATE KEY - SK)



## Ventajas

- El DW se independiza de cambios en el manejo de claves de los sistemas fuente.
- Permite manejar dimensiones de cambio lento.

## Desventajas

- Hay que manejar y administrar estas claves en el proceso de ETL.



# LAS 10 REGLAS ESENCIALES DEL MODELADO

## "10 Mandamientos de Kimball"

- Regla #1: Cargar los datos atómicos en estructuras dimensionales
- Regla #2: Estructura de los modelos dimensionales en función de los procesos de negocio
- Regla #3: Asegurarse de que cada tabla de hechos tiene una tabla de dimensión tiempo asociada
- Regla #4: Asegurarse de que todos los hechos de la tabla de hechos tienen el mismo nivel de detalle
- Regla #5: Resolver correspondencias muchos a muchos en tablas de hechos
- Rule #6: Resolver correspondencias muchos a muchos en tablas de dimensiones
- Regla #7: Almacenar las descripciones en las tablas de dimensión
- Regla #8: Asegurarse de que las tablas dimensionales usan claves subrogadas
- Regla #9: Crear dimensiones conformadas para integrar los datos de toda la empresa
- Regla #10: Valora constantemente los requerimientos y las realidades para proporcionar una solución DW/BI que sea aceptada por los usuarios de negocios y que apoye su proceso de decisiones



# CONSEJOS NOMBRES DE OBJETOS DWH

... deben estar orientados a los usuarios de negocios, no al personal técnico.

- Regla #1: Sigue la convención de nomenclatura.
- Regla #2: Cada objeto tiene un nombre.
- Regla #3: Los nombres de los objetos son descriptivos.
- Regla #4: No se recomiendan abreviaciones y acrónimos.
- Regla #5: Los nombres de los objetos son atractivos.
- Rule #6: Los nombres de los objetos son únicos.
- Regla #7: Nombres de objetos no demasiado largos.
- Regla #8: Considera anteponer a las columnas de nombre, una tabla abreviada del nombre.
- Regla #9: Cambia los nombres en la capa de visualización si es necesario.
- Regla #10: ¡Se constante!



# DIMENSIONAMIENTO DE UN MODELO

## EJEMPLO

Dimensión	
Tiempo	2 Años x 365 días = <b>730 días</b>
Sucursal	<b>300 sucursales</b> Informan ventas x día
Producto	30000 en cada sucursal, <b>4000 vendidos x día en todas las sucursales</b>
Empleado	Solo <b>1 empleado x sucursal x día</b>

**Número Base de Registros en las Fact Tables**  
 $730 \times 300 \times 4000 \times 1 = \mathbf{876 \text{ Millones de registros}}$

**Número de los Campos Indices = 4**  
**Número de los campos de la Fact Table = 4**  
**Total de Campos = 8**

**Tamaño Base de la Fact Table**  
 $876 \text{ Millones} \times 8 \text{ Campos} \times 4 \text{ bytes}^{(*)} = \mathbf{28 \text{ Gb}}$

(\*) caso hipotético en que todos ocupen 4bytes.

# TP CASO ESTUDIO: PLATAFORMA STREAMING SOUNDWAVE

## ENTREGA OPCIONAL (\*)



- Listar las posibles dimensiones a partir de las necesidades de información enunciadas.
- Identificar los hechos con las medidas básicas y calculadas.
- Identificar el proceso de negocio a modelar.
- Construir el modelo dimensional conceptual.
- Identificar atributos y jerarquías para cada dimensión.
- Construir el modelo dimensional lógico.
- Construir el modelo físico (en esquema Estrella y Copo de Nieve).

(\*)

- *Los TPs Opcionales sirven para afianzar conceptos necesarios para realizar los TPs de Aplicación, serán corregidos en clase y/o mediante Autoevaluación.*
- *Los TPs de Aplicación tiene una Fecha Límite de Entrega que deberá ser cumplida sin excepción, serán corregidos en detalle por los docentes.*
- *Ver documentos: "Condiciones de Cursada en MIeL - sección: Plazos y condiciones de Entrega Trabajos Prácticos y Casos de Estudio" + "Circuito Entrega TPs Teams/MIeL"*

# TRABAJO PRÁCTICO DE APLICACIÓN

## ENTREGA PRÓXIMA CLASE (\*)



### Parte 1 (Modelo Dimensional)

- Definir el contexto de la empresa.
- Listar dimensiones, hechos con medidas (básicas y calculadas), atributos y jerarquías.
- Construir los modelos dimensional conceptual y lógico.
- Buscar nuevas preguntas de negocio que podría responder el modelo construido.

### Parte 2 (Modelo Físico) (Próxima clase + 1)

- Construir el modelo físico (en esquema Estrella y Copo de Nieve) + identificar cada dimensión del modelo dimensional.

*(\*) Cada trabajo práctico tiene una Fecha Límite de Entrega que deberá ser cumplida sin excepción. (Ver Condiciones de Cursada en MIeL – Sección: "Plazos y condiciones de Entrega Trabajos Prácticos y Casos de Estudio")*

# RESUMEN TEMA 3 PARTE 1

