



UNIVERSIDAD NACIONAL DE LA MATANZA

INTELIGENCIA DE NEGOCIOS

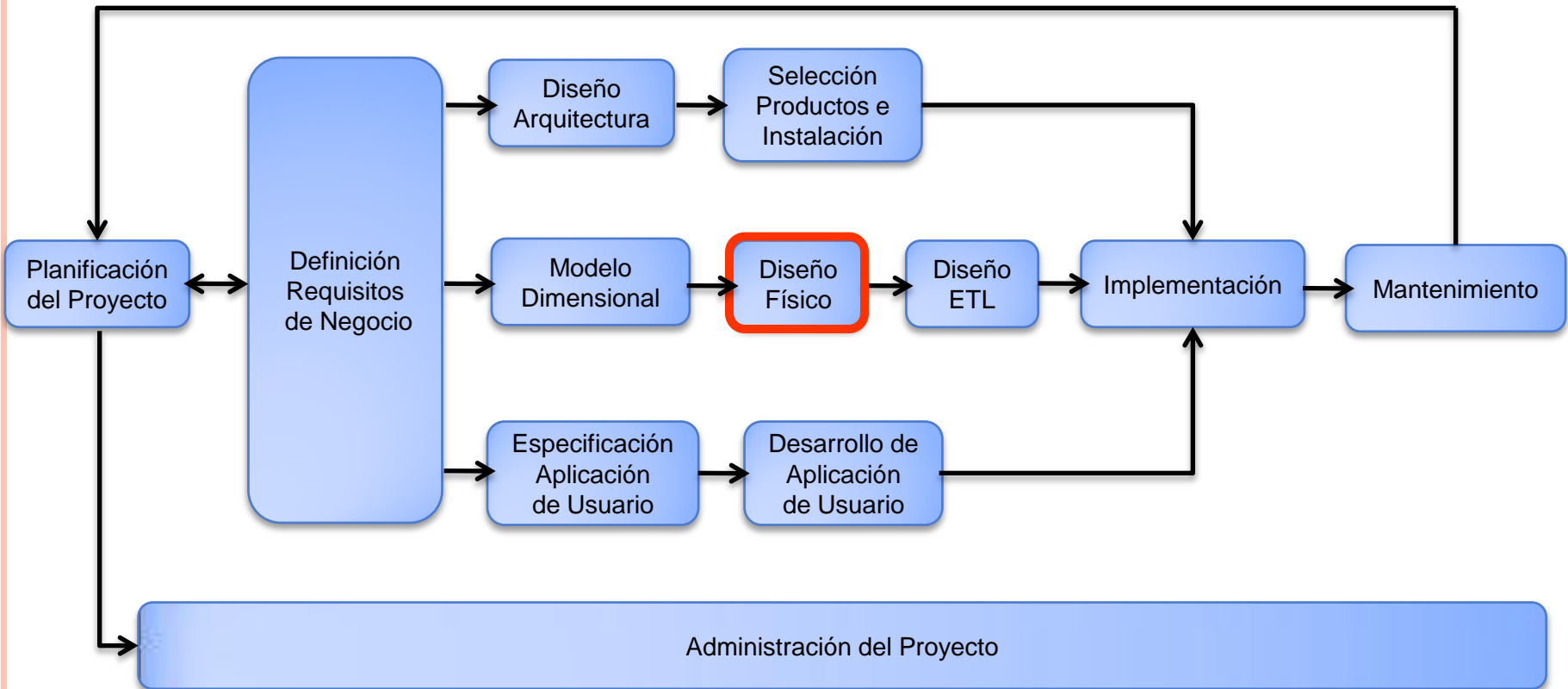
Modelo Físico

Técnicas de Modelado - Parte 1

Profesor: Mg. Diego Basso

Curso 2017

CICLO DE VIDA DE UN PROYECTO DE BI



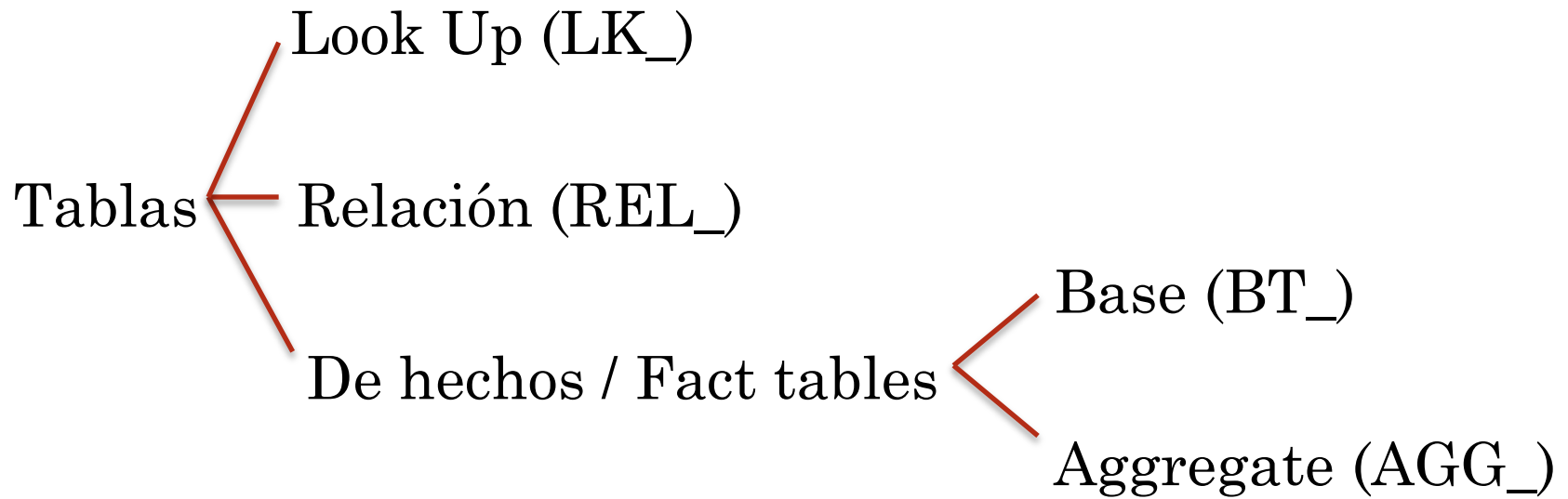
MODELO FÍSICO



- 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* mas una columna por cada *padre* del atributo.



EJEMPLO TABLA LOOK UP



○ Provincia

- Provincia_ID
 - Desc_Provincia
- } *Attribute form*

1, Buenos Aires
2, Córdoba
3, La Pampa

} *Elementos*

LK_PROVINCIA

Provincia_ID	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

- Ciudad_ID
- Desc_Ciudad

Provincia



Jerarquía de atributos

LK_CIUADAD

Ciudad_ID	Desc_Ciudad	Provincia_ID
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

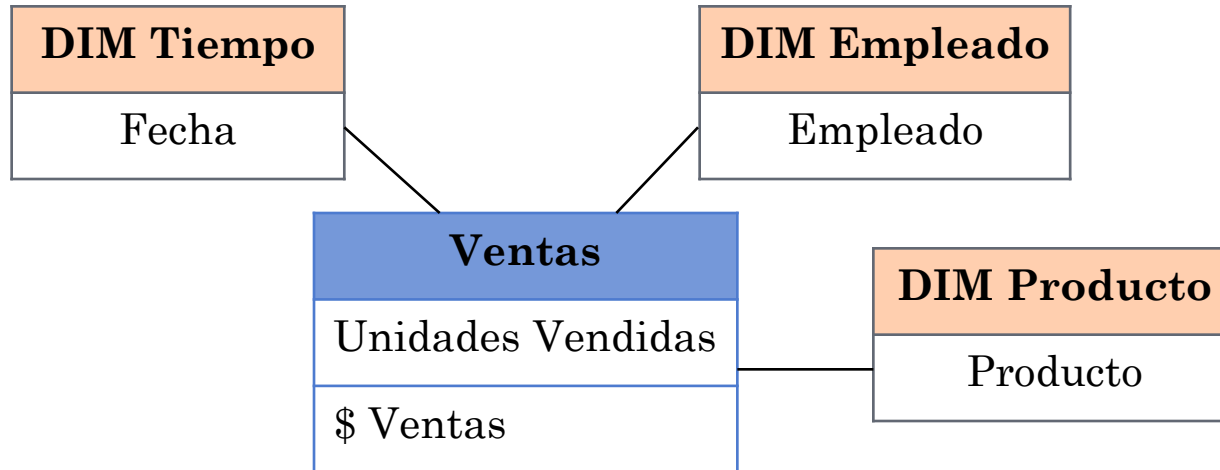


Tabla de hechos **Base**

- Almacena los **valores** de las medidas de los hechos al *máximo nivel de detalle*.
 - 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

Fecha_ID	Empleado_ID	Producto_ID	Unid_Vend	\$ Ventas
01/04/2016	111	17	20	\$ 400
01/04/2016	112	17	45	\$ 900
01/04/2016	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

Fecha_ID	Empleado_ID	Producto_ID	Unid_Vend	\$ Ventas
01/04/2016	111	17	20	\$ 400
08/04/2016	112	17	45	\$ 900
15/04/2016	112	17	30	\$ 450
19/04/2016	111	18	15	\$ 300

AGG_VENTAS_EMP_MES

Mes_ID	Empleado_ID	Unid_Vend	\$ Ventas
201604	111	35	\$ 700
201604	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

- Producto_ID
- Desc_producto

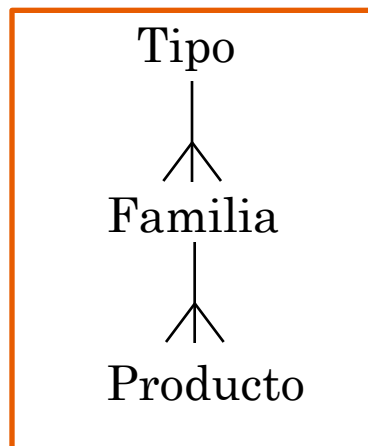
○ Familia

- Familia_ID
- Desc_familia

○ Tipo

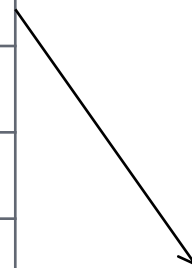
- Tipo_ID
- Desc_tipo

DIM_PRODUCTO

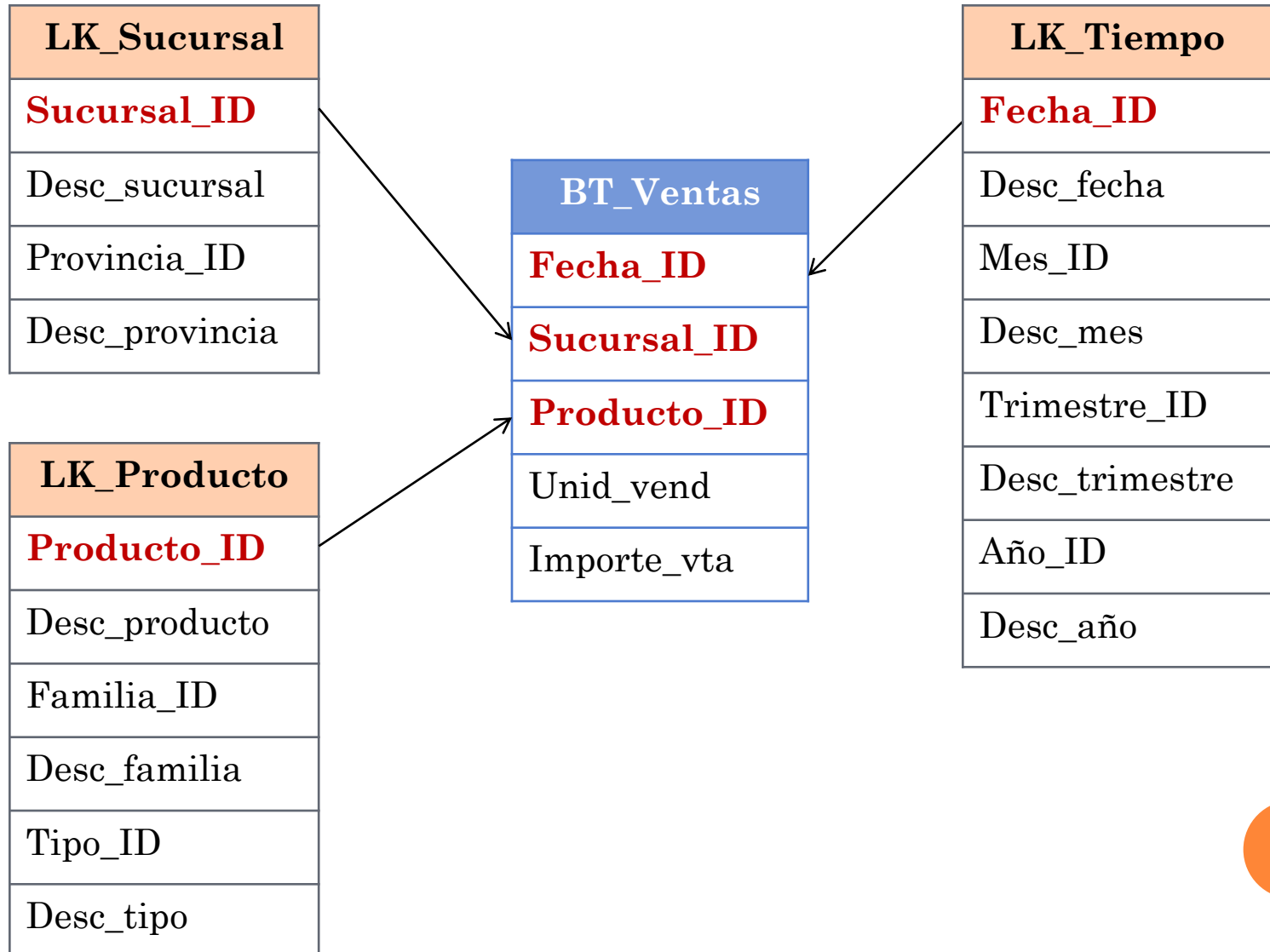


LK_Producto
Producto_ID
Desc_producto
Familia_ID
Desc_familia
Tipo_ID
Desc_tipo

BT_Ventas
Fecha_ID
Sucursal_ID
Empleado_ID
Producto_ID
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 2017
- Dimensiones para:
 - Mostrar, filtrar, agrupar
- 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 suma
FROM
    BT_Ventas v,
    LK_Sucursal s,
    LK_Producto p,
    LK_Tiempo t
WHERE
    v.Fecha_ID = t.Fecha_ID and
    v.Sucursal_ID = s.Sucursal_ID and
    v.Producto_ID = p.Producto_ID and
    s.Desc_provincia = 'CABA' and
    t.Mes_ID = '022017' and
    p.Desc_tipo = '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

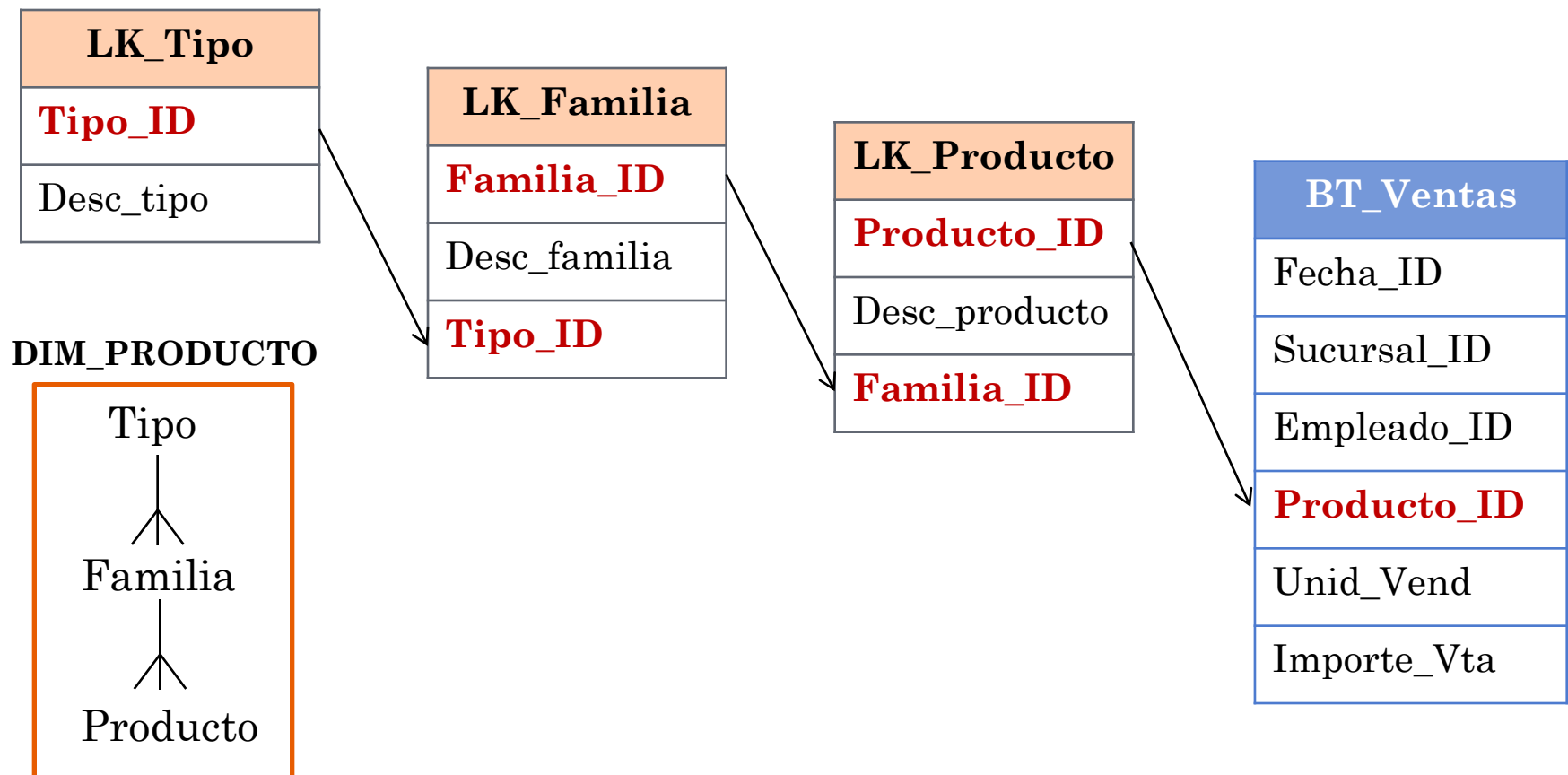
- Producto_ID
- Desc_producto

○ Familia

- Familia_ID
- Desc_familia

○ Tipo

- Tipo_ID
- Desc_tipo

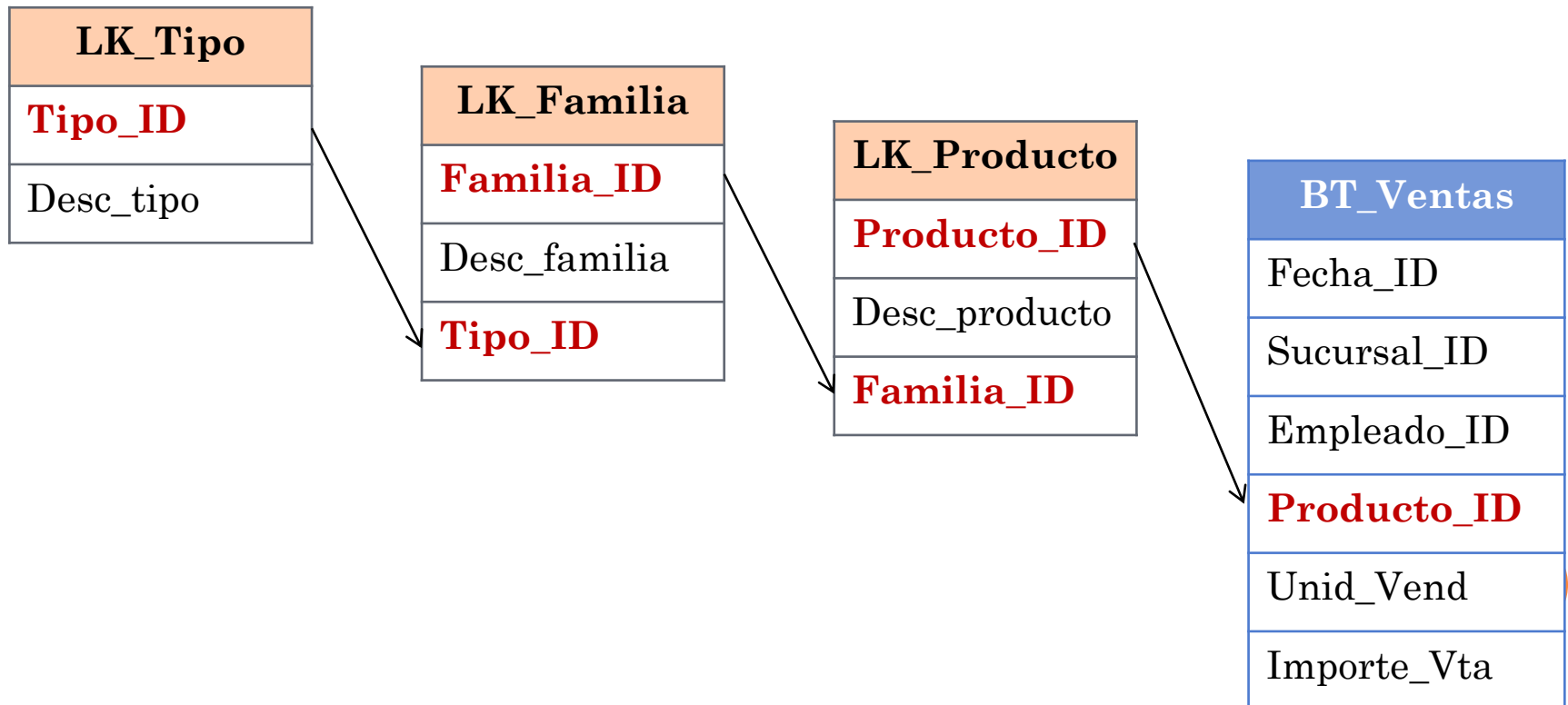


NORMALIZACIÓN



○ 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.

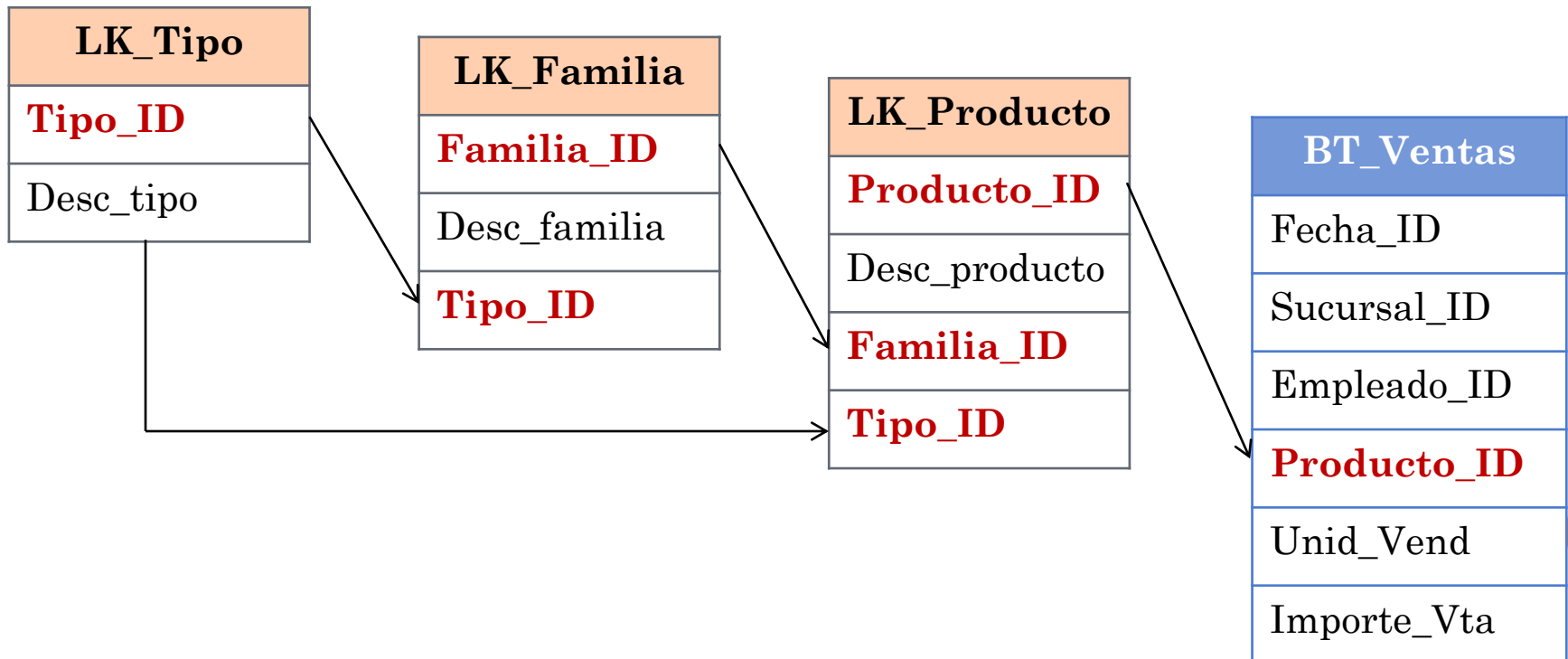




NORMALIZACIÓN

○ 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.

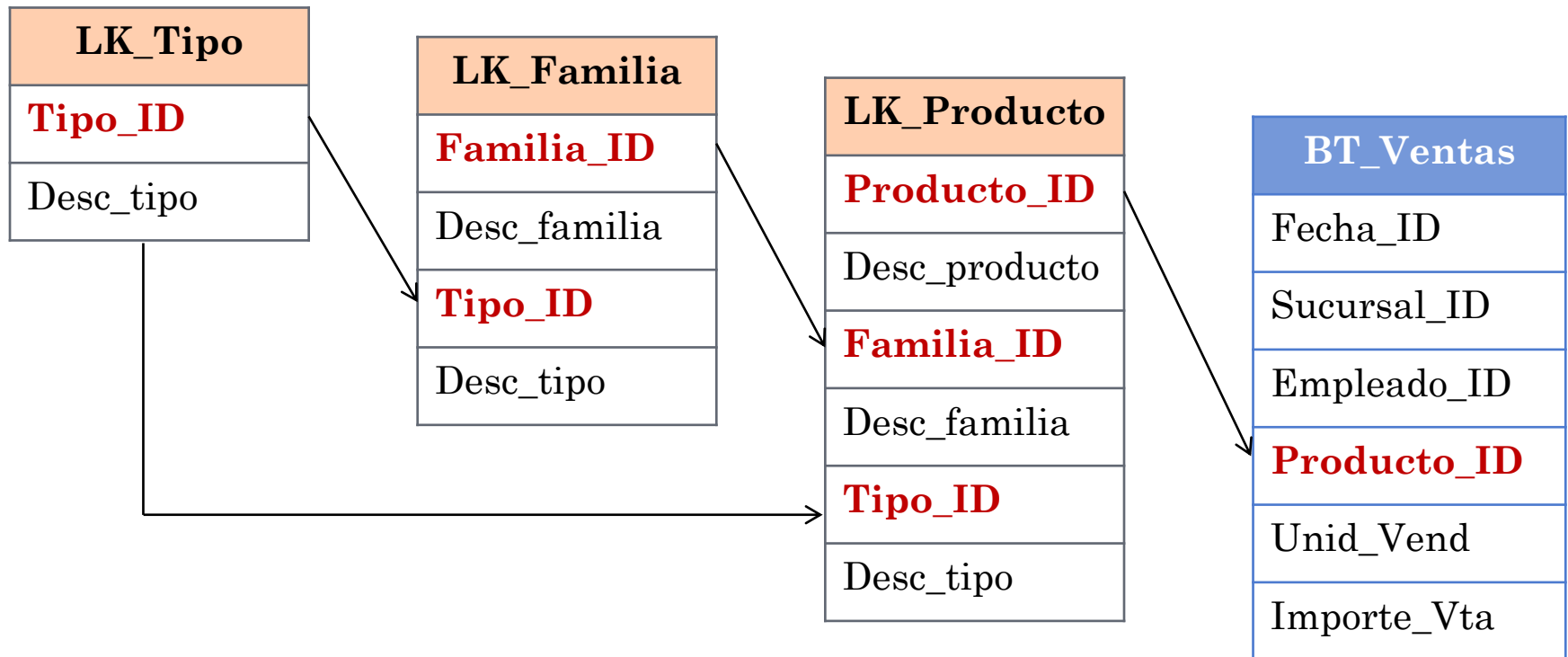




NORMALIZACIÓN

○ 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



¿CÓMO 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
Producto_ID
Cod_producto
Desc_producto
....

BT_Ventas
Fecha_ID
Sucursal_ID
Empleado_ID
Producto_ID
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.



