

# TEMA 2

## DISEÑO MULTIDIMENSIONAL

Natalia Padilla Zea, Eladio GarvÍ, José Samos

## 2.1. Ciclo de vida

2

- Para un sistema informático en general (p.e. sistema de gestión de ventas):
  - ▣ Análisis de requisitos
    - Usuario y desarrollador (jefe del proyecto)
    - Entrevistas: usuario indica lo que necesita y desarrollador lo interpreta y registra
  - ▣ Diseño conceptual
    - E/R
    - Casos de uso
    - ...

## 2.1. Ciclo de vida

3

- Para un sistema informático en general (p.e. sistema de gestión de ventas):
  - ▣ Diseño lógico
    - Implementación
  - ▣ Pruebas
  - ▣ Presentación al cliente
    - Iterativo hasta completar requisitos
  - ▣ Implantación
    - Puesta en producción

## 2.1. Ciclo de vida

4

- En Sistemas Multidimensionales: dos enfoques
  - ▣ Dirigido por requerimientos
  - ▣ Dirigido por datos

## 2.1. Ciclo de vida

5

- Enfoque dirigido por requerimientos
  - ▣ Entrevista con el cliente (requisitos)
  - ▣ Modelo conceptual
  - ▣ Diseño lógico
  - ▣ Generar datos de prueba
  - ▣ Darlo al cliente para hacer pruebas, ¿qué implica?

## 2.1. Ciclo de vida

6

- Enfoque dirigido por requerimientos
  - ▣ Darlo al cliente para hacer pruebas
    - ¿Con qué datos va a funcionar el sistema?
      - Con los de la empresa
    - ¿De qué va a depender el sistema?
      - De los datos que pueda aportar el usuario o se puedan recopilar de otros sitios
  - Entonces...
    - Hay que mirar los datos y el contenido de dichos datos antes de asegurarle al usuario que se puede hacer lo que pide
  - Si los datos no son válidos para lo que el usuario pide, esta aproximación no vale

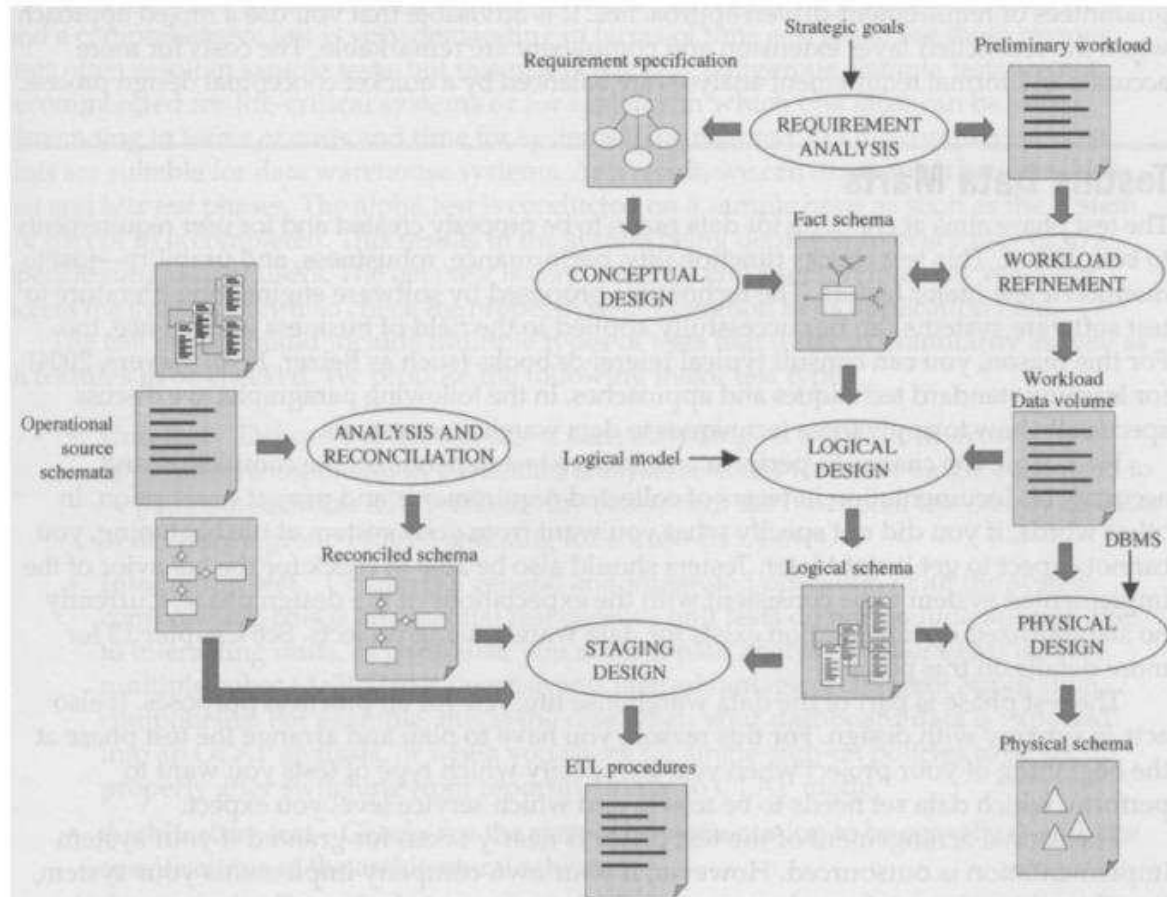
## 2.1. Ciclo de vida

7

### □ Enfoque dirigido por requerimientos

(Golfarelli y Rizzi, 2009)

*Requirement-driven approach*



## 2.1. Ciclo de vida

8

- Enfoque dirigido por datos
  - ▣ Ver los datos de la empresa y hacer una propuesta
    - Necesidad de mucho esfuerzo
    - La propuesta puede no ser útil para el decisor
    - Problemas “políticos” para acceder a los datos
    - El diseño está condicionado por los datos (extremo contrario)

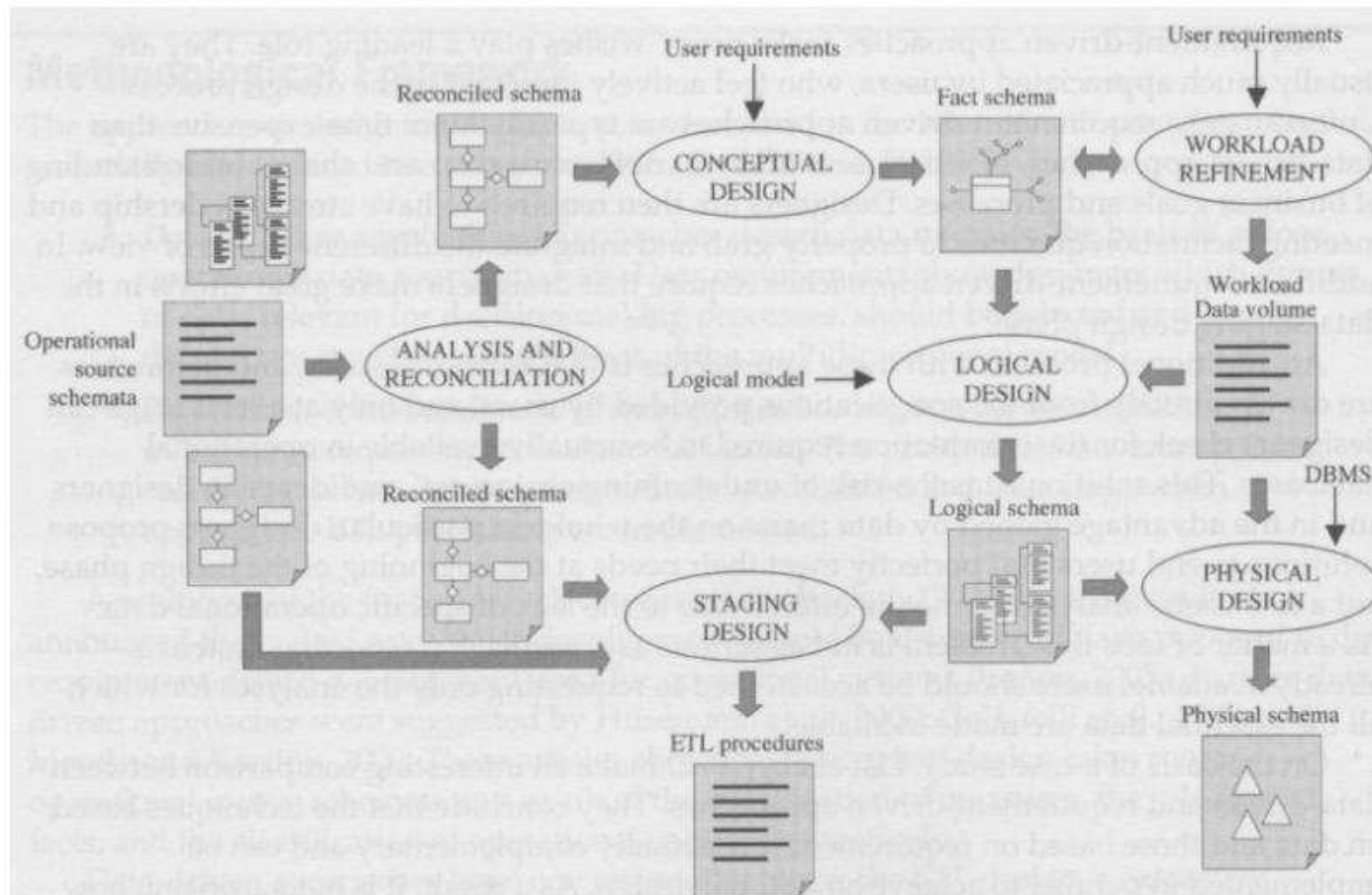


## 2.1. Ciclo de vida

9

### □ Enfoque dirigido por datos

(Golfarelli y Rizzi, 2009)  
*Data-driven approach*



## 2.1. Ciclo de vida

10

- Lo más razonable: enfoque mixto
  - ▣ Combinar requerimientos de decisor y datos disponibles
  - ▣ Propuesta útil teniendo en cuenta los datos que hay

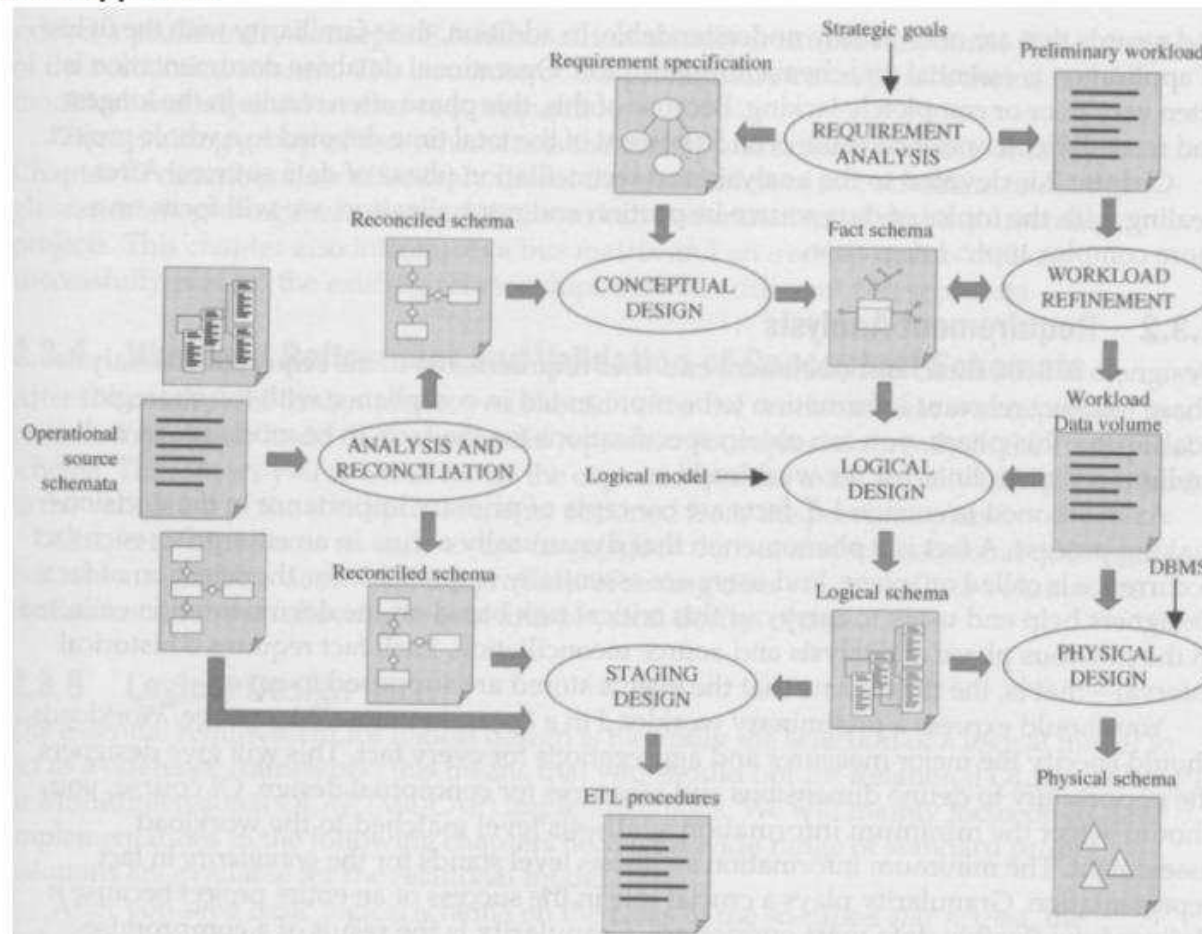
## 2.1. Ciclo de vida

11

- Enfoque mixto

*(Golfarelli y Rizzi, 2009)*

### Mixed approach



## 2.2. Diseño Conceptual

12

- Metodología en 4 pasos (Kimball):
  - ▣ Seleccionar el proceso de negocio a modelar
  - ▣ Establecer la granularidad
  - ▣ Determinar las dimensiones de análisis
  - ▣ Identificar las mediciones (hechos)
- Es una metodología antigua que sigue vigente

## 2.2. Diseño Conceptual

13

- PASO 1: Seleccionar el proceso de negocio
  - ▣ Estudiar la actividad principal de la empresa
    - Ejemplos: Universidad, hoteles
  - ▣ Si hay varias opciones, elegir la más importante como foco de atención
    - Cada diseño conceptual tiene un único foco de atención

## 2.2. Diseño Conceptual

14

- PASO 2: Establecer la granularidad
  - ▣ Dependerá de las necesidades del analista (decisor)
  - ▣ Considerar el rendimiento y la cantidad de espacio
    - Si basta un resumen, no almacenarlo todo
    - El espacio ocupado influirá en el rendimiento
  - ▣ Plantearse:
    - ¿Todos los decisores necesitan el mismo nivel de detalle?
    - ¿Necesitan datos externos?

## 2.2. Diseño Conceptual

15

- ▣ Si se puede, se elige:
  - Datos detallados, con menos datos externos y menos complejidad de implementación
  
- ▣ IMPORTANTE:
  - El nivel de detalle indica lo que significa un hecho
  - Hay que seleccionar el nivel de detalle que interesa al DECISOR

## 2.2. Diseño Conceptual

16

- PASO 3: Diseño de las dimensiones
  - ▣ Ver las dimensiones iniciales
    - Ejemplo: “Resumen de ventas a nivel de producto, tienda y fecha” (CASO 1)
  - ▣ Ver otros componentes en cada dimensión (hacer lista)
    - Entre ellos tiene que haber una relación parte-todo exclusiva (1-N)
  - ▣ Ordenar los niveles y colocar el TODO al final
  - ▣ Añadir “descriptores” si fuese necesario



## 2.2. Diseño Conceptual

17

- ▣ Puede haber dimensiones conformadas
  - Dimensiones compartidas entre cubos
    - Facilitan la reutilización → Ahorra trabajo
    - Permiten realizar Drill-Across
- ▣ En el diseño conceptual se incluyen todos los cubos
  - Cubo base y todos los que se puedan obtener
- ▣ Número de cubos con distinto nivel de detalle:
  - Multiplicar entre sí el número de niveles de cada dimensión
  - ¿Qué supone añadir un nivel en una dimensión?

## 2.2. Diseño Conceptual

18

- ▣ Para añadir un nivel:
  - El decisor tiene que estar interesado en dicho nivel
  - Existir una relación parte-todo exclusiva con otro
  
- ▣ Niveles (Para el CASO 1)
  - Ejemplo: Marca, Importe
    - Nivel: Marca, Todo, Todo
  - Ejemplo: Marca, Departamento, Importe
    - Nivel: Marca-Departamento, Todo, Todo (drill-down)
  - Ejemplo: Marca, Departamento, Producto, Importe
    - Nivel: Producto, Todo, Todo (drill-down)

## 2.2. Diseño Conceptual

19

- PASO 4: Seleccionar mediciones (Hechos)
  - ▣ ¿Qué queremos medir?
  - ▣ Estudiar la aditividad en cada dimensión
    - Comprobarlo en un nivel (TODO)
    - Para todas las dimensiones: ver el significado y si tiene sentido sumar
  - ▣ Podemos tener mediciones calculadas
    - Las que no se pueden sumar
    - Se marcan como (\*)

## 2.2. Diseño Conceptual

20

### □ Ejemplo: Aditividad de N° Clientes en el CASO 1

	QUÉ	DÓNDE	CUÁNDO	CONCLUSIÓN
Importe	SUM	SUM	SUM	ADITIVA
Cantidad	SUM	SUM	SUM	ADITIVA
PVP	<del>SUM</del>	<del>SUM</del>	<del>SUM</del>	NO ADITIVA
N° Clientes	<del>SUM</del> (1)	SUM (2)	SUM	SEMI-ADITIVA

- (1) Si en dos tickets distintos aparece el mismo producto y en uno de ellos aparecen dos productos más, al agrupar por producto nos saldrían tres productos, lo que nos llevaría a pensar que hay tres clientes, pero en realidad sólo hay 2 clientes (tickets).

## 2.2. Diseño Conceptual

21

### □ Ejemplo: Aditividad de N° Clientes en el CASO 1

	QUÉ	DÓNDE	CUÁNDO	CONCLUSIÓN
Importe	SUM	SUM	SUM	ADITIVA
Cantidad	SUM	SUM	SUM	ADITIVA
PVP	<del>SUM</del>	<del>SUM</del>	<del>SUM</del>	NO ADITIVA
N° Clientes	<del>SUM</del> (1)	SUM (2)	SUM	SEMI-ADITIVA

- (2) Si tenemos tickets diferentes (no se tienen en cuenta los productos) en una misma tienda, al contar, se suma el número de tickets, que sería el número de ocurrencias de clientes, es decir, el número de ventas.

## 2.2. Diseño Conceptual

22

- Notas de repaso (I)
  - ▣ Se dan 4 pasos para diseñar un sistema multidimensional
  - ▣ Terminología:
    - Instancia: ocurrencia del hecho a nivel conceptual
    - Registro: ocurrencia del hecho a nivel lógico
  - ▣ Para establecer la granularidad (Paso 2):
    - ¿Cuáles son las bases del sistema?
    - Estimar el número de registros para la carga del sistema
  - ▣ Copo de nieve tiene más tablas pero ocupa menos que estrella porque sus tablas están normalizadas

## 2.2. Diseño Conceptual

23

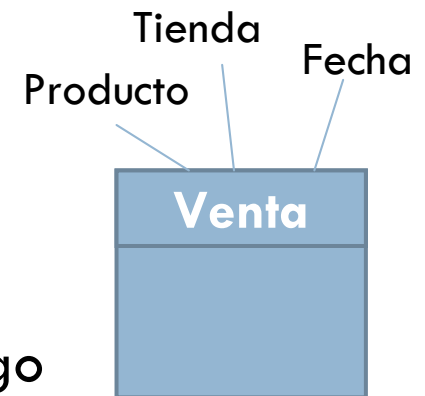
- Notas de repaso (II)
  - ▣ Hay que calcular el espacio que ocupa el diseño
    - Hay que considerar las bases y las mediciones
    - Bases: Conjunto de dimensiones que identifican unívocamente los hechos (se considera el nivel más bajo)
    - Multiplicar por el número de instancias

## 2.2. Diseño Conceptual

24

- En el CASO 1: “Resumen de ventas por producto, tienda y fecha”

- $N^{\circ} \text{ tiendas} * N^{\circ} \text{ fechas} * N^{\circ} \text{ productos vendidos}$



- Con cifras...

- Empresa con 8 tiendas, 80.000 productos en catálogo

- Tamaño de la dimensión Tienda

- $8 * 1 \text{ KB (aprox)}$

- Tamaño de la dimensión Fecha (5 años)

- $365 - 52 \text{ (festivos)} = 313 * 5 = 1565 * 64 \text{ B}$

- Tamaño de la dimensión Producto

- $80.000 * 1 \text{ KB (aprox)}$

**≈ 80 MB**  
(el resto es despreciable)



## 2.2. Diseño Conceptual

25

### ■ Tabla de Hechos

- ¿Cuántos productos se venden en un día? Todos los del catálogo seguro que no... Suponemos que 1000 distintos por tienda.
- $1565 \text{ fechas} * 8 \text{ tiendas} * 1000 \text{ productos} \approx 12.520.000 \text{ ventas}$
- ¿Cuánto ocupa cada registro?

Tienda + Fecha + Producto + Medición 1 + Medición 2  
10+      10+      10+              2+                      2   = 34 B

$12.520.000 \text{ ventas} * 34 \text{ B} \approx 425 \text{ MB}$



**Tenemos que reducir  
este espacio**

## 2.2. Diseño Conceptual

26

- Para reducir el tamaño se añaden las claves generadas
  - CodTienda: 1 B
  - CodFecha: 2 B
  - CodProducto: 3 B
  
- Así tenemos:  $1\text{ B} + 2\text{ B} + 3\text{ B} + 2\text{ B (Med. 1)} + 2\text{ B (Med. 2)} = 10\text{ B}$
  
- $12.520.000\text{ ventas} * 10\text{ B} \approx 125\text{ MB}$
  
- Se ha reducido el tamaño de la tabla de hechos un 70%

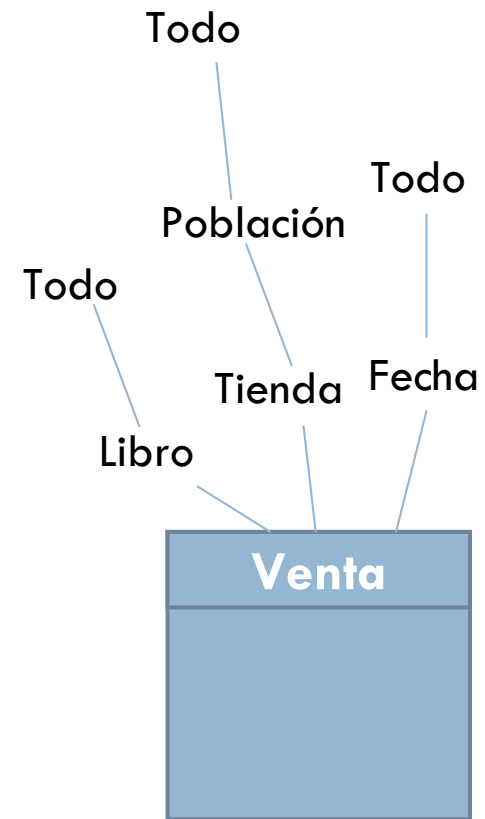
## 2.2. Diseño Conceptual

27

### □ Identificación de bases

#### ▣ ¿Cuáles son las bases?

Libro, Tienda, Fecha

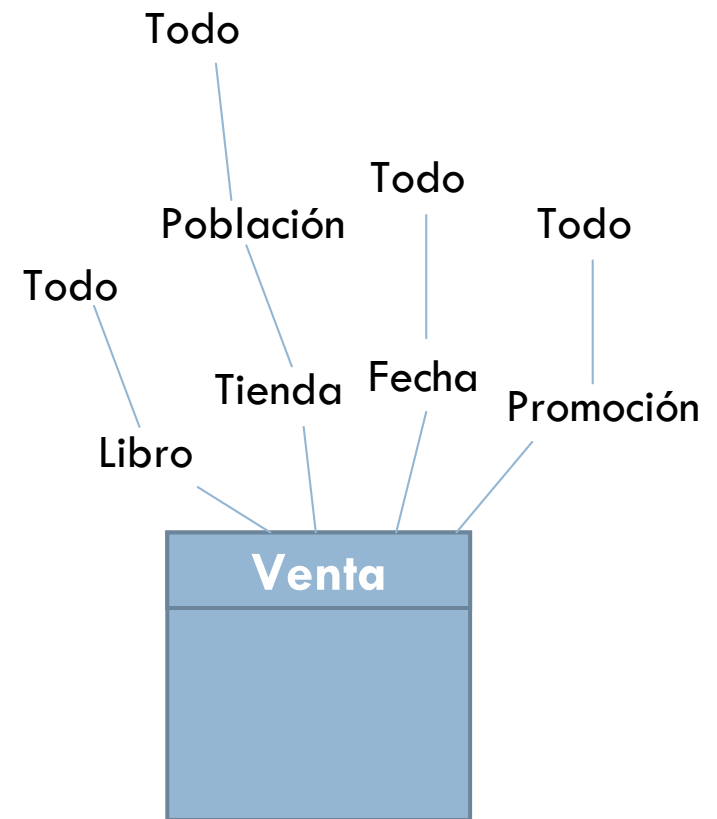


## 2.2. Diseño Conceptual

28

- Identificación de bases
  - ▣ ¿Y si añadimos otra dimensión?

Libro, Tienda, Fecha, Promoción



## 2.2. Diseño Conceptual

29

### □ Identificación de bases

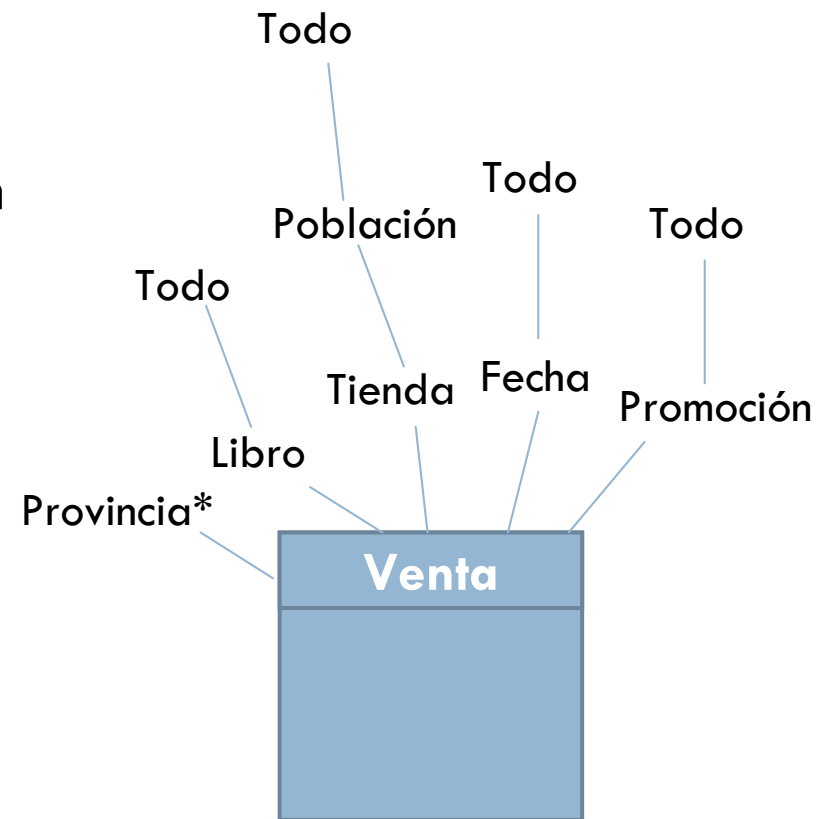
#### ▣ ¿Y si añadimos otra más?

Libro, Tienda, Fecha, Promoción

La Provincia no sería base

Añadir una base implica que los datos se disgregan, es decir, hay más instancias en los hechos

Añadir una dimensión que no es base no añade instancias

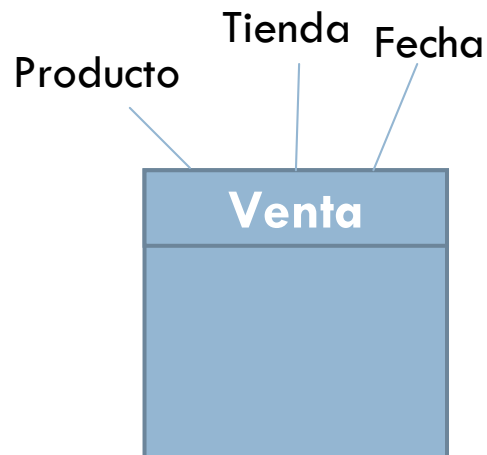


\* Es un mal diseño... ¡Está mal!

## 2.2. Diseño Conceptual

30

- CASO 2: “Ventas a nivel de línea de ticket”
  - ▣ El decisor indica que necesita “Analizar los productos que se venden conjuntamente” (p.e., pañales y toallitas, o libro1 y libro2)

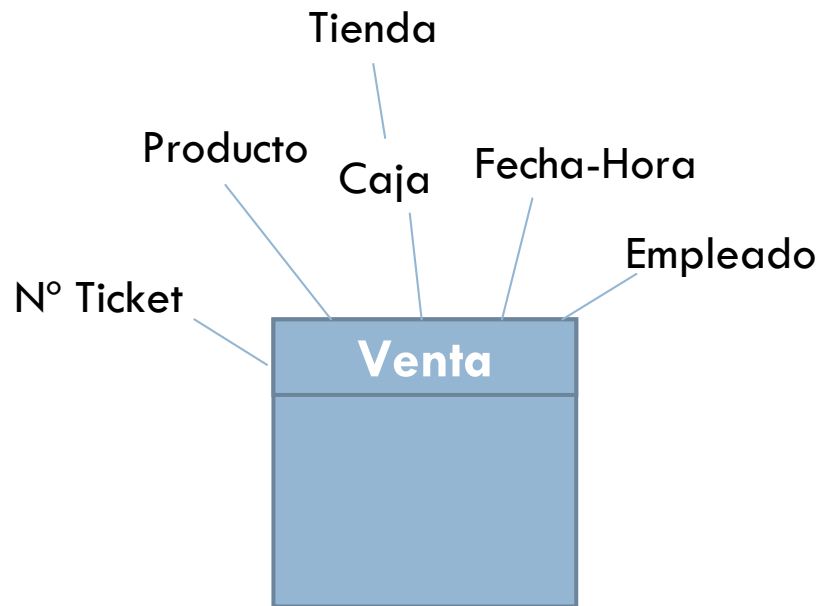


En este caso, no se puede saber lo que quiere el decisor porque no hay información a nivel de cliente

## 2.2. Diseño Conceptual

31

- Hay que plantear otra solución (diseño conceptual)



¿Cuál es el foco de atención? Venta

¿Qué es un hecho? Una línea de ticket

¿Cuál es la granularidad? Línea de ticket

## 2.2. Diseño Conceptual

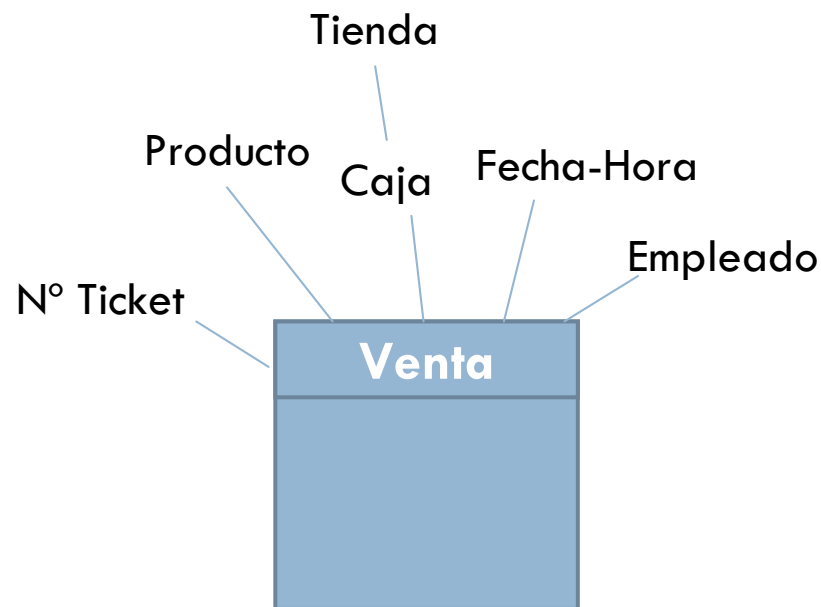
32

▣ ¿Cuáles son las bases?

▣ Varias opciones:

■ N° Ticket, Producto

- Línea dentro del ticket
- ¿El ticket es único en el sistema? ¿Es único por tienda?
- Si es único en el sistema, las bases son: N° Ticket y Producto
- Asumimos que el n° de ticket va a estar. ¿Se puede cambiar el producto por otra base? No





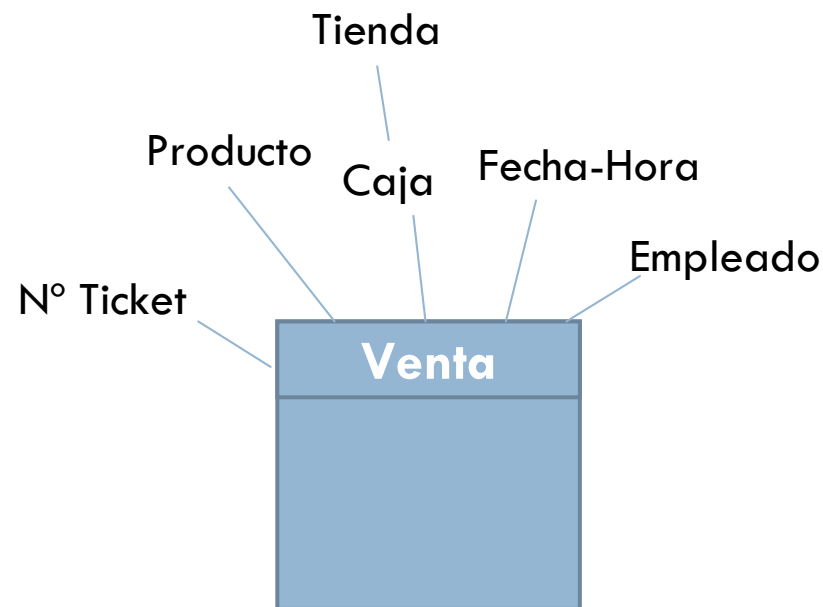
## 2.2. Diseño Conceptual

33

▣ ¿Cuáles son las bases?

▣ Varias opciones:

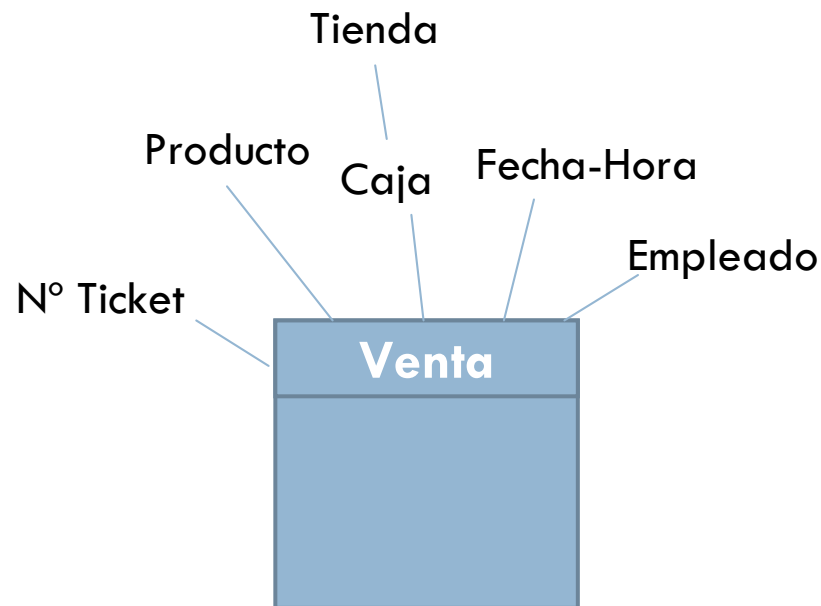
- Si no ponemos ticket y sí el producto, ¿qué opciones tenemos?



## 2.2. Diseño Conceptual

34

### ▣ ¿Cuáles son las bases?



### ▣ Varias opciones:

#### ■ Si incluimos el Producto:

- Caja, Fecha-Hora, Producto
- Empleado, Fecha-Hora, Producto

#### ■ ¿Y si añadimos algún atributo más?

- No aporta información → No es una base

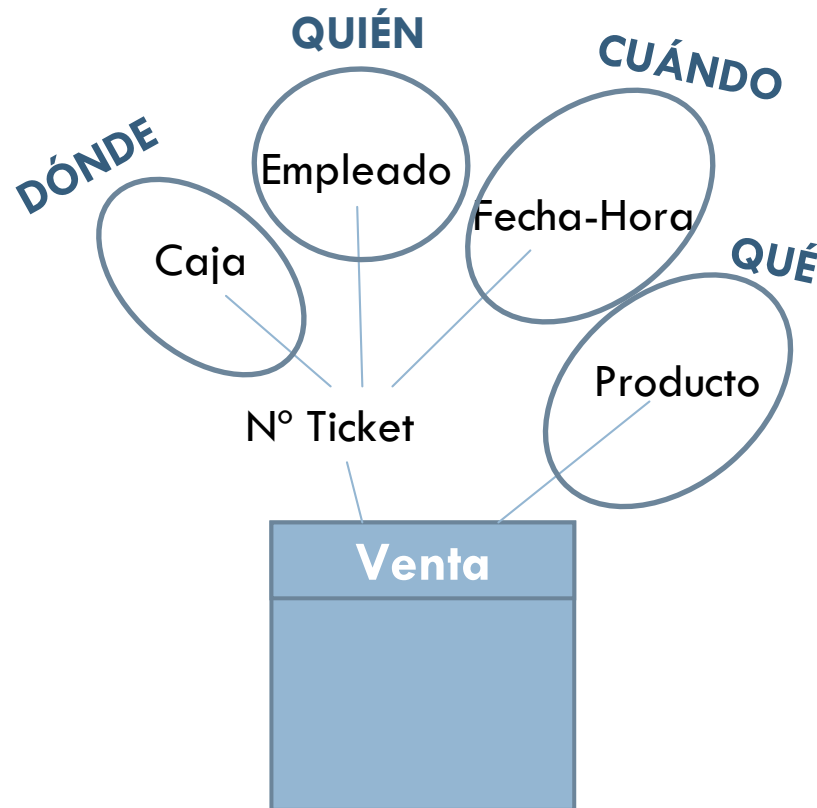
#### ■ ¿Cómo reducimos el número de instancias?

- ¿Quitar el Producto? No se puede
- Con poco coste se puede tener más información si añadimos Empleado, Fecha-Hora, Caja

## 2.2. Diseño Conceptual

35

### □ CASO 2: Diseño alternativo ¿Qué os parece?



*Mezcla los significados de las dimensiones*

#### Consideraciones de diseño:

- Hay que diferenciar el significado de las dimensiones
- Poner el N° ticket (n° expediente, n° factura...) aparte [Patrón de diseño]

## 2.2. Diseño Conceptual

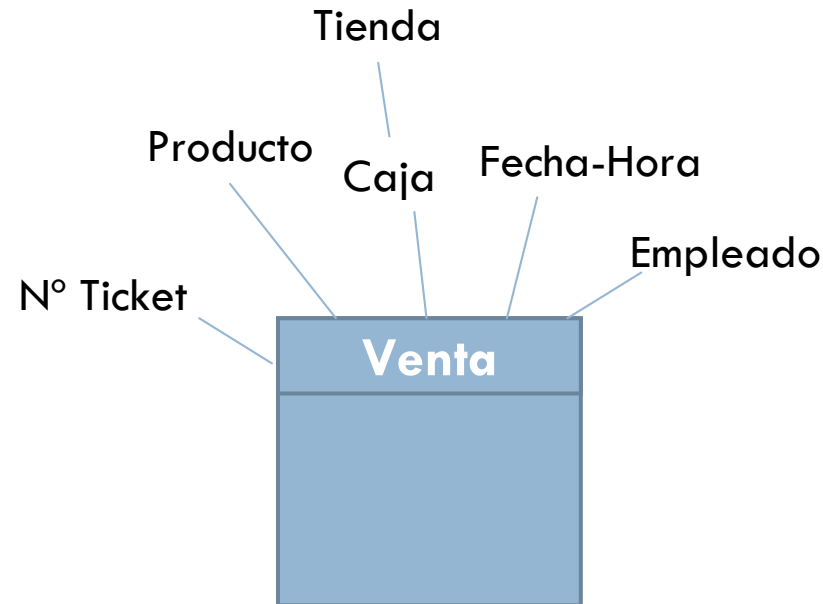
36

- ▣ ¿Sabemos cuántas instancias hay de Caja?
  - Sí
- ▣ ¿Sabemos cuántas instancias hay de Producto?
  - Sí
- ▣ ¿Sabemos cuántas instancias hay de Fecha-Hora?
  - Sí
- ▣ Podemos tener esta información preparada y controlada (con todas sus instancias)

## 2.2. Diseño Conceptual

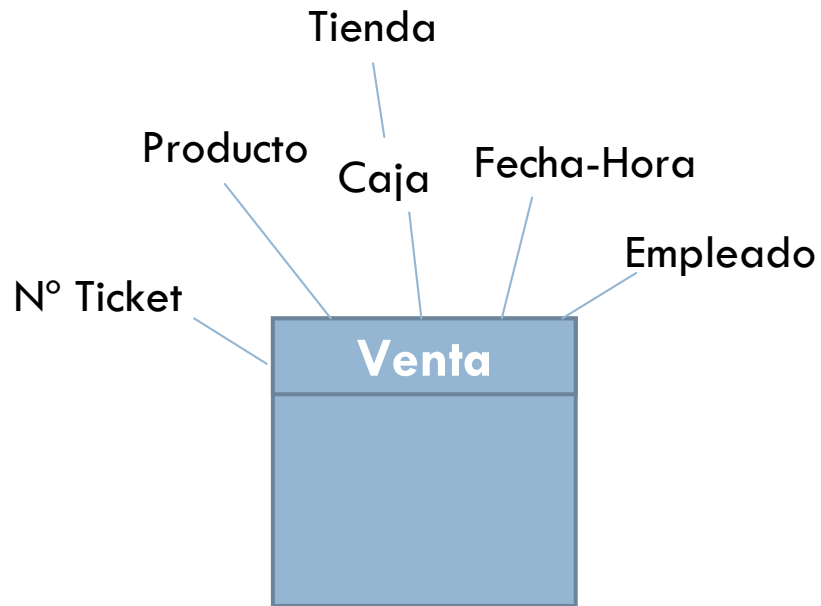
37

- ▣ ¿Sabemos cuántas instancias hay de N° Ticket?
  - No... dependerá de las ventas que se hagan
  - Para controlar mejor el tamaño de las dimensiones es preferible la primera versión del diseño



## 2.2. Diseño Conceptual

38



□ ¿Se puede enriquecer la información del Empleado?

▣ Ver la información de la BD

■ (Cod1, Empl1; Cod2, Empl2)

■ (0001, José Pérez Rus;...)

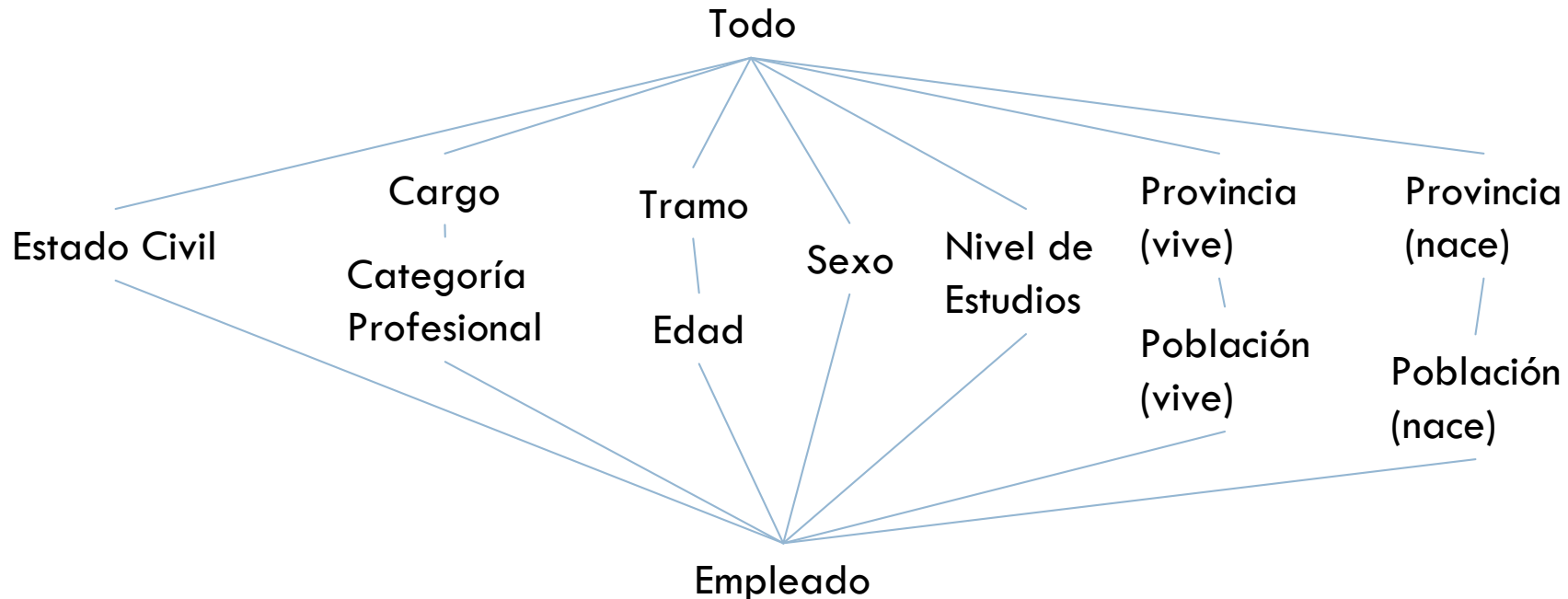
■ Si hay más información:

- BD de personal de la empresa
- Edad, Categoría Profesional, Antigüedad, Nivel de Estudios, Estado Civil, Nº Hijos, Domicilio, Lugar de Nacimiento...

## 2.2. Diseño Conceptual

39

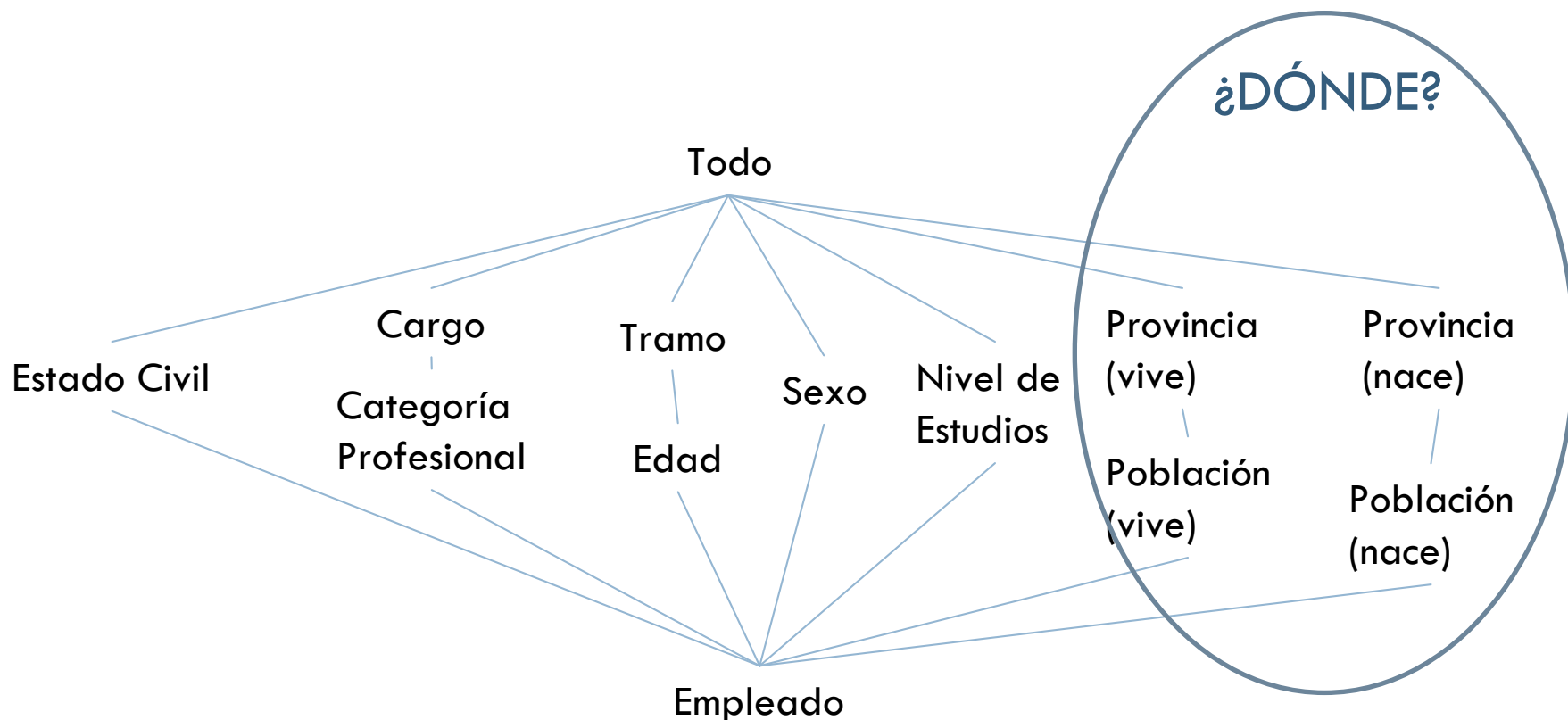
- Supongamos que al decisor le interesa saber la edad, sexo, categoría profesional, estado civil, nivel de estudios, residencia y lugar de nacimiento
- ¿Cómo se relacionan en la dimensión?



## 2.2. Diseño Conceptual

40

- Desde el punto de vista de los hechos, NO representa “dónde”, sino “quién” → Describe al Empleado

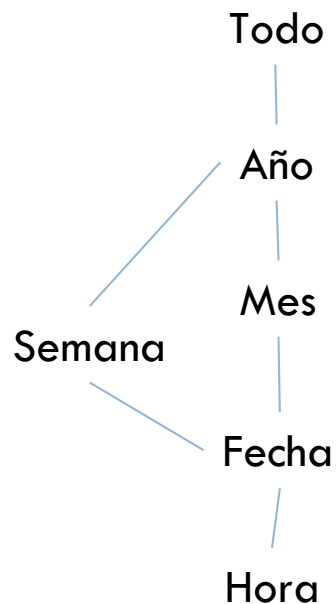




## 2.2. Diseño Conceptual

41

- ¿Cómo sería la dimensión Fecha-Hora?

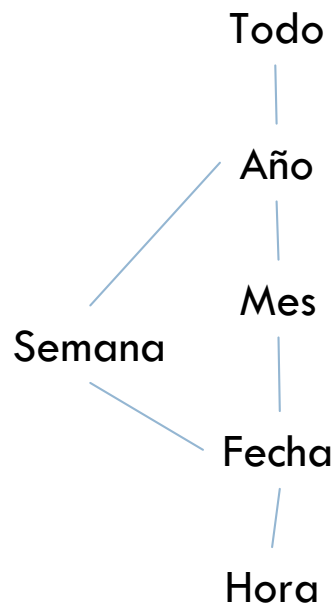


- 22/04/2015
- 23/04/2015
- 17:00:00
- 17:00:12
- ¿Cardinalidad entre ellos?
- 22/04/2015-17:00:00
- 22/04/2015-17:00:12
- 23/04/2015-17:00:00
- 23/04/2015-17:00:12

## 2.2. Diseño Conceptual

42

### □ ¿Cómo sería la dimensión Fecha-Hora?



### □ ¿Cuántas instancias hay?

- 5 años, 365 días/año=1825
- 24 horas / día
- 60 minutos / hora
- 60 segundos / minuto
- → 157.680.000 segundos
- Tener esto almacenado ocupa mucho (¡y no hay ventas en todos esos instantes de tiempo!)

## 2.2. Diseño Conceptual

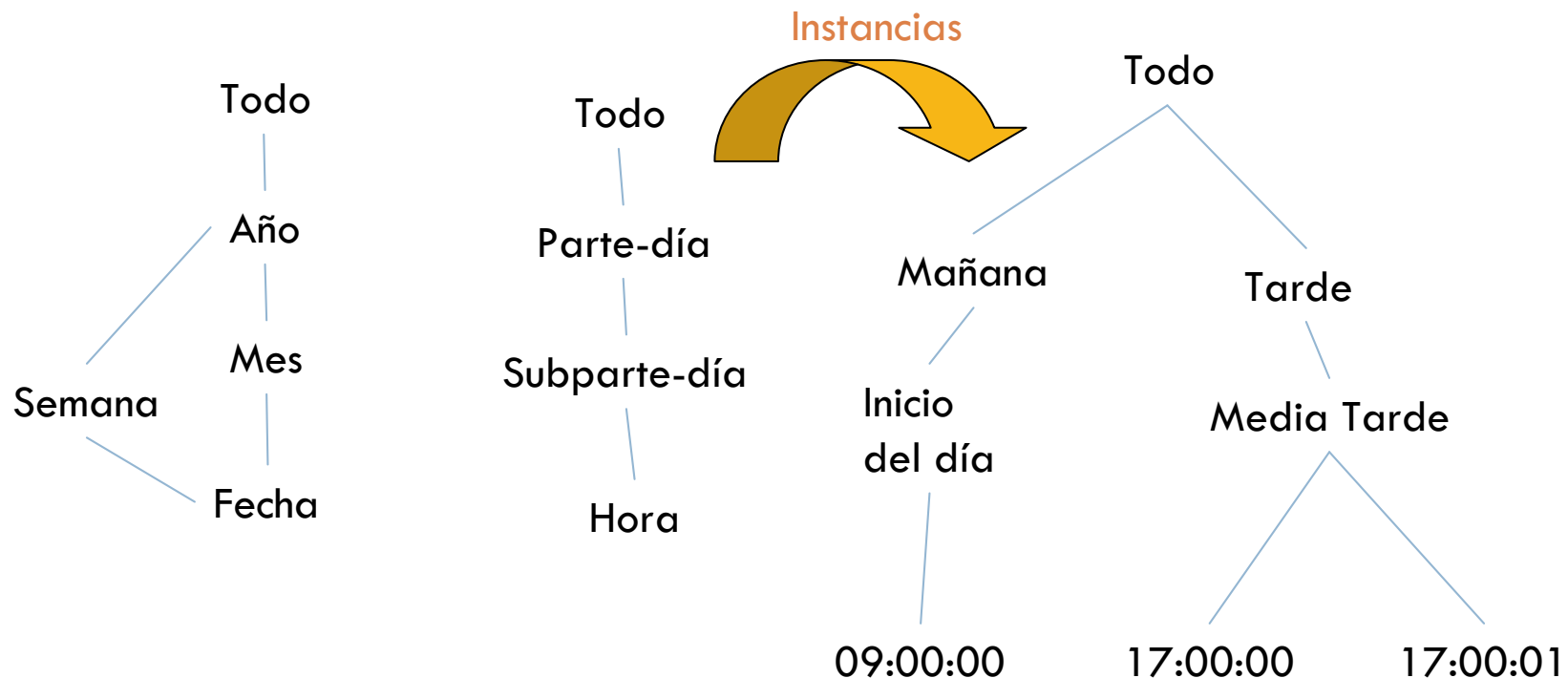
43

- Posibilidades para la dimensión Cuándo:
  - ▣ Anotar sólo las instancias en las que haya ventas
    - No podemos tener preparada la dimensión antes
  - ▣ No guardar a nivel de segundo
    - Entonces, ¿a qué nivel? Minuto, 10 minutos...
    - Sí permite preparar la dimensión previamente
  - ▣ Separar Hora y Fecha como dimensiones diferentes
    - Inconveniente: El diseño es más complicado
    - Ventajas:
      - La fecha puede ser una dimensión conformada con otros diseños
      - La hora se puede agrupar con más facilidad

## 2.2. Diseño Conceptual

44

- ▣ Separar Hora y Fecha como dimensiones diferentes



- ▣ En este caso, para la base hay que tomar Fecha y Hora

## 2.2. Diseño Conceptual

45

- Selección de mediciones
  - ▣ Importe, cantidad → Aditivas
  - ▣ PVP → Calculada
  - ▣ N° clientes
    - Versión de resumen por ticket: N° clientes = 1 (uno cada ticket) → Semi-aditiva
    - Versión línea de ticket: No se puede sumar por producto → Semi-aditiva

## 2.2. Diseño Conceptual

46

- Estimación del número de instancias de los hechos
  - ¿Depende de la elección de las bases? Las opciones son:
    - Caja, Fecha-Hora, Producto
    - Empleado, Fecha-Hora, Producto
    - N° Ticket, Producto
  - Depende siempre del número de líneas del ticket (productos)
  - ¿Cuántos tickets se emiten en una caja al día? ¿Cuántas líneas tiene de media un ticket? ¿Cuántas cajas hay?
    - Ej. 1000 cajas \* X horas \* Y tickets \* Z líneas
  - Para la base con Empleado se hace un razonamiento similar

## 2.2. Diseño Conceptual

47

### ■ N° de instancias de los hechos para el CASO 2

#### ■ Supongamos

- T: N° tickets
- C: Cantidad de cajas (500)
- V: Cantidad de empleados (700)
- P: Cantidad de productos (60.000)
- F: Cantidad de fechas
- H: Cantidad de horas

## 2.2. Diseño Conceptual

48

- ▣ N° instancias de los hechos
  - T\*P no puede ser porque significaría que en cada ticket vendemos todos los productos
  - T\*n sería más correcto, donde  $n = n^{\circ}$  medio de líneas por ticket (p. e. 10). Hay que estimar T:
    - F: 5 años = 1825 días  
Quitamos 325 días que está cerrado  $\rightarrow F = 1500$  días
    - N° tickets al día en cada caja  $\approx 600$
    - $T \approx C * N^{\circ} \text{ tickets} * N^{\circ} \text{ días}$
    - $T \approx 500 * 600 * 1.500 = 450.000.000$
  - Entonces, la tabla de hechos tiene T\*n instancias
    - $450.000.000 * 10 = 4.500.000.000$  instancias



## 2.2. Diseño Conceptual

49

### □ CASO 3: “Promociones en ventas”

#### ▣ Ejemplos de promociones:

- Llévese 3 y pague 2
- 2ª unidad con 50% de descuento
- Sin promoción

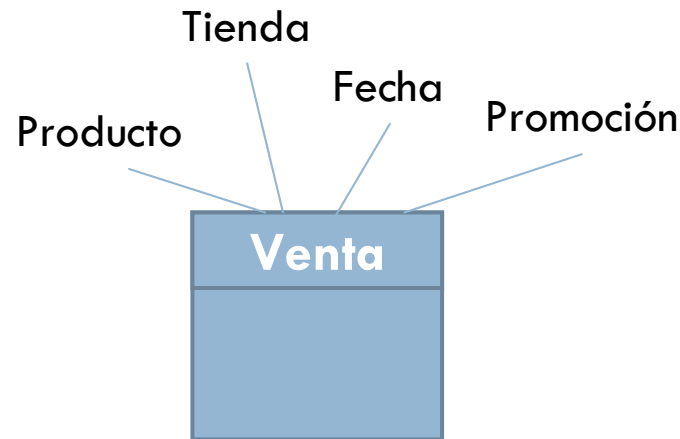
#### ▣ Hacemos el diseño conceptual

- Selección del proceso de negocio: Ventas
- Granularidad: El decisor indica necesidad de analizar por promoción
- ¿Cómo sale el diseño?
  - Partimos del DC de resumen de ventas del Caso 1

## 2.2. Diseño Conceptual

50

- Diseño conceptual para el CASO 3 (versión 1)



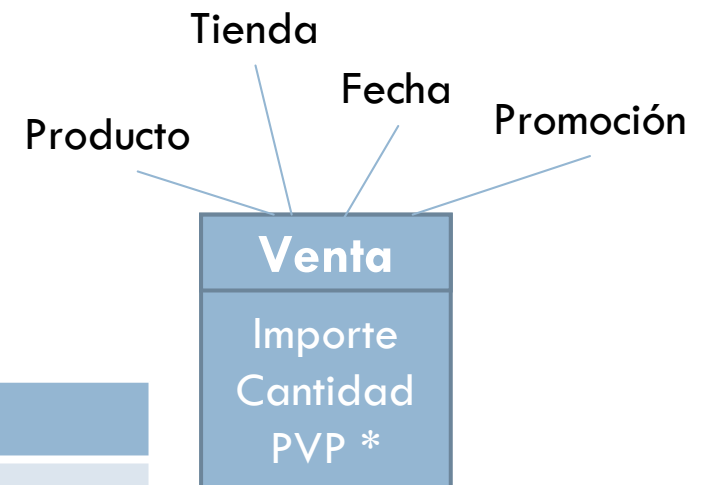
- Necesitamos identificar la Promoción porque desdoblará los hechos

## 2.2. Diseño Conceptual

51

- Ejemplo de informe típico de ventas con promoción: “Productos vendidos en Granada, en enero de 2014, bajo la promoción ‘lleve 3 y pague 2’”

PROMOCIÓN 3X2, GRANADA, ENERO 2014	
PRODUCTO	CANTIDAD
P1	200
P2	150
...	...
...	...
P100	6

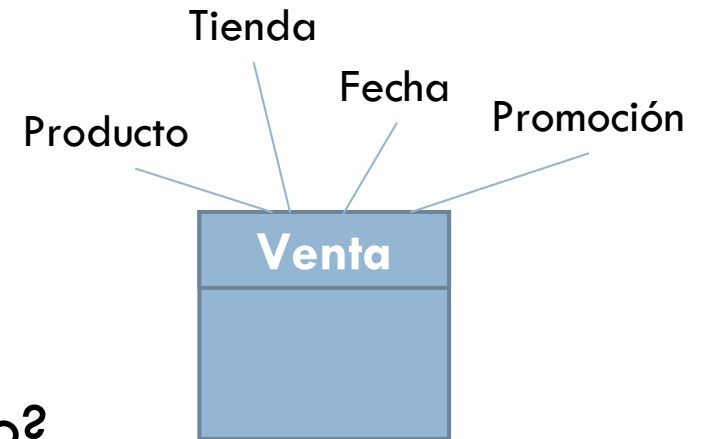


## 2.2. Diseño Conceptual

52

- Bases para el CASO 3 (versión 1)

- Producto, Tienda, Fecha, Promoción



- ¿Qué problemas plantea el diseño?

- ¿Cómo se sabe el resultado de las promociones?
  - “Venta” sólo recoge los hechos ocurridos
  - Si hay una promoción de la que no se ha vendido nada, no hay hechos asociados → No se puede analizar
- Este diseño refleja las ventas con promoción y sin ella, pero no podemos saber si una promoción ha estado vigente y no se ha vendido nada de ella

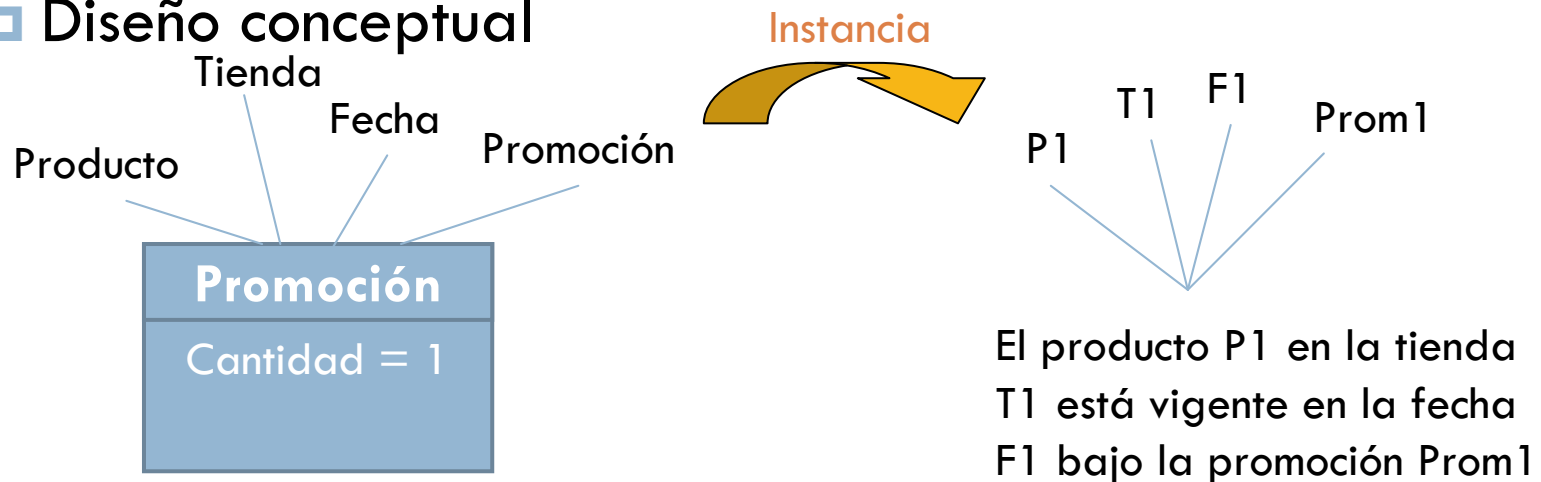
## 2.2. Diseño Conceptual

53

### □ CASO 4: “Promociones sin ventas”

▣ Solución a los problemas del Caso 3: un nuevo cubo

▣ Diseño conceptual



### Consideraciones de diseño:

- No hay mediciones → Son hechos sin mediciones
- Se incluye la medición por defecto (Cantidad = 1) para poder sumar (aditiva) (“Factless-Fact”)

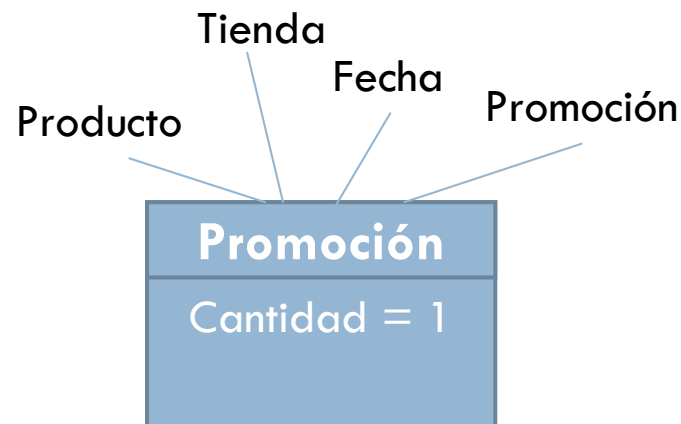
## 2.2. Diseño Conceptual

54

### □ Estimación del tamaño para el CASO 4

#### ▣ Hechos:

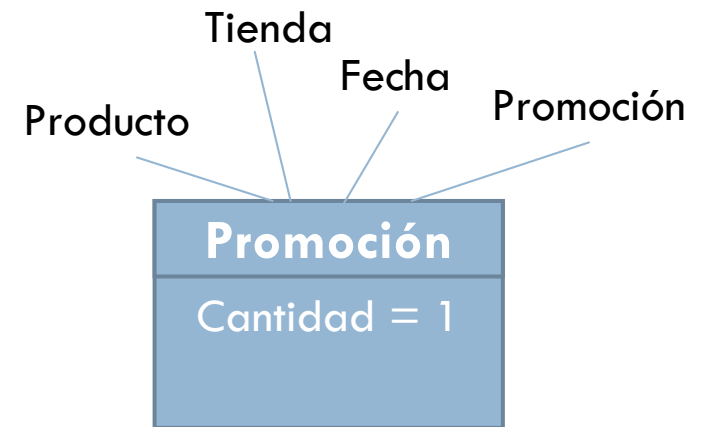
- “Sin promoción” no se considera instancia de Promoción
- Suponemos que cada día, cada tienda tiene algún producto en promoción
- $\text{Tamaño} = \text{N}^\circ \text{ Tiendas} * \text{N}^\circ \text{ días} * \text{N}^\circ \text{ Productos en promoción} * \text{N}^\circ \text{ promociones del producto}$



## 2.2. Diseño Conceptual

55

- Ejemplo de informe típico de sobre promociones: “Productos bajo la promoción `lleve 3 y pague 2` en Granada, en enero de 2014”

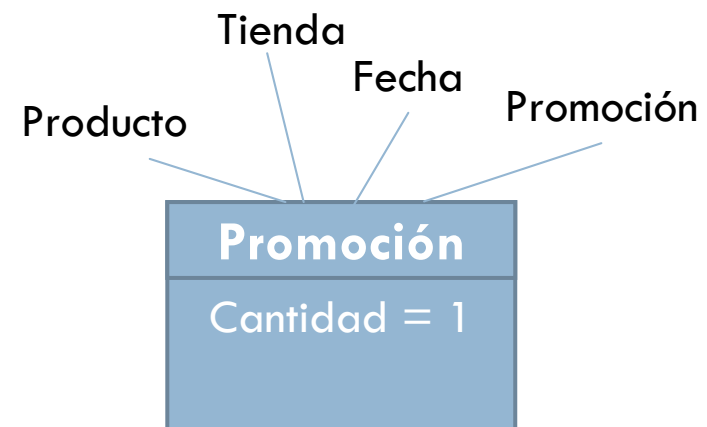
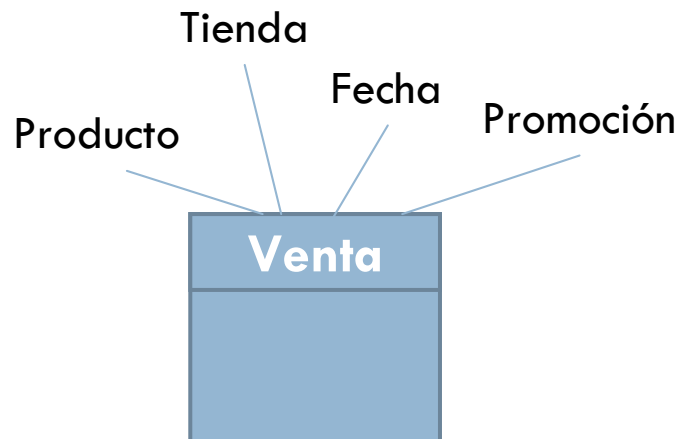


PROMOCIÓN 3X2, GRANADA, ENERO 2014	
PRODUCTO	CANTIDAD
P1	8
P2	7
...	...
...	...
P100	6

## 2.2. Diseño Conceptual

56

- ¿Cómo sabemos los productos en promoción que no se han vendido? Realizando Drill-Across con Venta
- Para hacer Drill-Across entre dos cubos es necesario que estos cubos tengan alguna dimensión o nivel en común





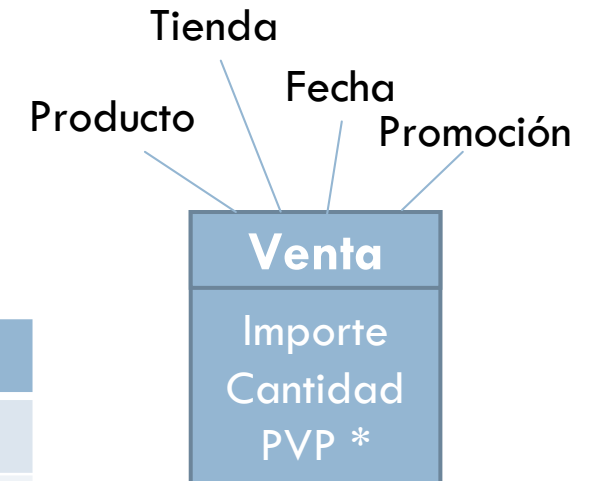
## 2.2. Diseño Conceptual

57

- Ejemplo de informe de promociones sin ventas

PROMOCIÓN 3X2, GRANADA, ENERO 2014		
PRODUCTO	CANTIDAD (Promoción)	CANTIDAD (Venta)
P1	8	200
P2	7	150
P3	8	0
...	...	...
P100	6	100

El producto P3 ha estado 8 veces en promoción en el mes de Enero en Granada y se ha vendido 0 veces



## 2.2. Diseño Conceptual

58

- CASO 5: Almacén de una cadena de tiendas
  - Caso a: Movimientos
    - ▣ Tenemos una aplicación OLTP de almacén que nos dice los movimientos del almacén y el motivo de los movimientos:
      - 300 unidades de entrada desde el distribuidor
      - 200 unidades de salida a tienda
      - 50 unidades estropeadas por agua

## 2.2. Diseño Conceptual

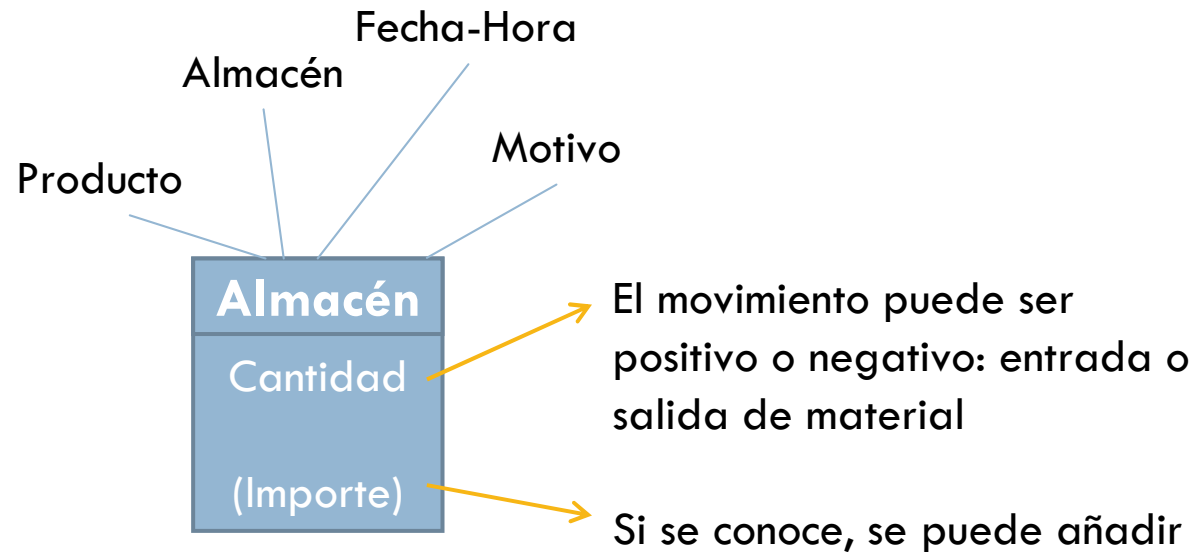
59

- CASO 5.a: Movimientos
  - ▣ ¿Sobre qué se produce el movimiento? Producto
  - ▣ ¿Dónde se produce el movimiento? Almacén
  - ▣ ¿Cuándo se produce? En una Fecha-Hora
  - ▣ ¿Por qué se produce? Por un motivo
  - ▣ ¿Cuánto valen los productos? Si se sabe, podremos añadir el importe como medición

## 2.2. Diseño Conceptual

60

- CASO 5.a: Movimientos
  - Es un caso similar al de Ventas, salvo porque la cantidad puede ser positiva o negativa



## 2.2. Diseño Conceptual

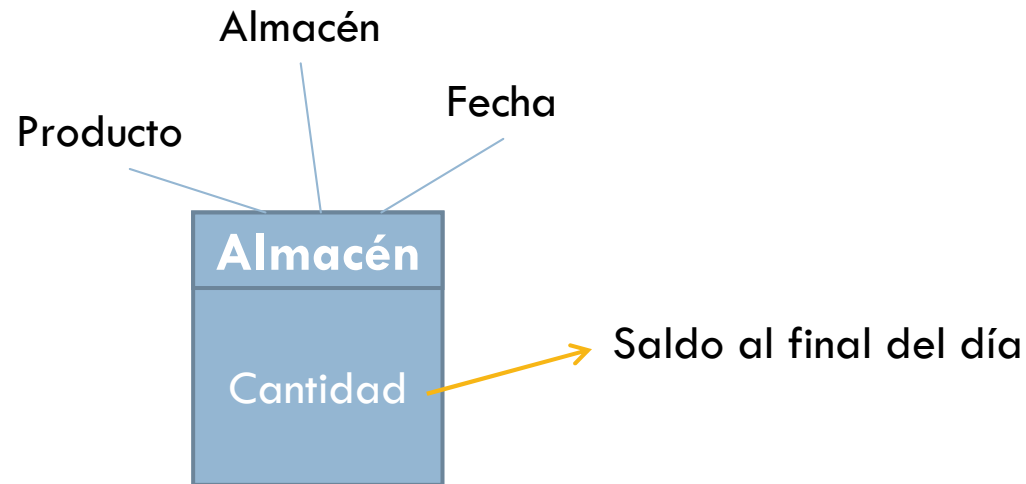
61

- CASO 5: Almacén de una cadena de tiendas
  - Caso b: Inventario
    - ▣ Supongamos que el decisor necesita saber la evolución del stock de productos. Ahora necesitamos:
      - Cantidad inicial en una fecha concreta
      - Sumar las entradas
      - Restar las salidas
      - ... para un periodo de tiempo, por ejemplo, un mes.
    - ▣ ¿Cómo sería el modelo conceptual? ¿Podría hacerse con el diseño conceptual del caso 5.a?

## 2.2. Diseño Conceptual

62

### □ CASO 5.b: Inventario



- Con esta solución podemos analizar la evolución del stock, pero no los motivos
- La “mejor” solución depende de lo que quiera el decisor

## 2.2. Diseño Conceptual

63

- CASO 5.b: Inventario de almacén de una cadena de tiendas
  
- Ejercicio propuesto
  - Indicar las bases y la aditividad del sistema
  - Hacer una estimación del número de instancias de los hechos

## 2.2. Diseño Conceptual

64

- CASO 6: Matriculación de vehículos
  - ▣ Cada mes, la DGT genera un archivo de matriculación de vehículos (ver T2-Anexo-1)
  - ▣ Solución:
    - Seleccionar el proceso de negocio: Matriculación de vehículos
      - Suponiendo que estamos en un concesionario de una marca de vehículos, ¿para qué puede interesar esta información?
      - “Este año hemos vendido un 30% menos que el año pasado”
      - “Este mes hemos vendido un 30% menos que el mismo mes del año pasado”
      - ¿Son malos datos? Depende... Si la matriculación total en el sector ha bajado un 50%, no está tan mal.



## 2.2. Diseño Conceptual

65

### □ CASO 6: Matriculación de vehículos

#### ▣ Solución:

##### ■ Granularidad

- No tenemos un “DNI” ni la matrícula propiamente dicha → Resumen de matriculaciones
- Si se matriculan varios coches del mismo tipo en las mismas condiciones, los datos nos los darán agrupados

## 2.2. Diseño Conceptual

66

### □ CASO 6: Matriculación de vehículos

#### ▣ Solución:

- Dimensiones: cuándo, qué, quién, dónde
- ...
- Nota: fecha de matriculación y fecha de 1ª matriculación no se pueden poner en la misma dimensión porque representan dos cuándo diferentes

## 2.2. Diseño Conceptual

67

### □ CASO 6: Matriculación de vehículos

#### ▣ Solución:

- Nos quedan varios campos que parecen no tener relación con ningún otro:
  - Clave (8 valores)
  - Procedencia (4 valores)
  - Servicio (10 valores)
  - Indicador (Nuevo | Usado) (2 valores)
- Opciones para los valores sin relación:
  - Poner una dimensión para cada dato: maneja muchas dimensiones y pocas instancias en cada una
  - “Combinar” esos datos en una nueva dimensión

## 2.2. Diseño Conceptual

68

### □ CASO 6: Matriculación de vehículos

#### □ Solución:

##### ■ “Cajón de trastos”

- Combinación de diferentes elementos sin relación aparente
- Si salen muchas combinaciones, se pueden eliminar de la BD las combinaciones que no se dan para que sea más eficiente
- En nuestro caso:  $8 \text{ claves} * 4 \text{ procedencias} * 10 \text{ servicios} * 2 \text{ indicadores} = 640 \text{ instancias}$
- Si salen muchas instancias, poner 2 cajones de trastos haciendo en cada uno las combinaciones más probables

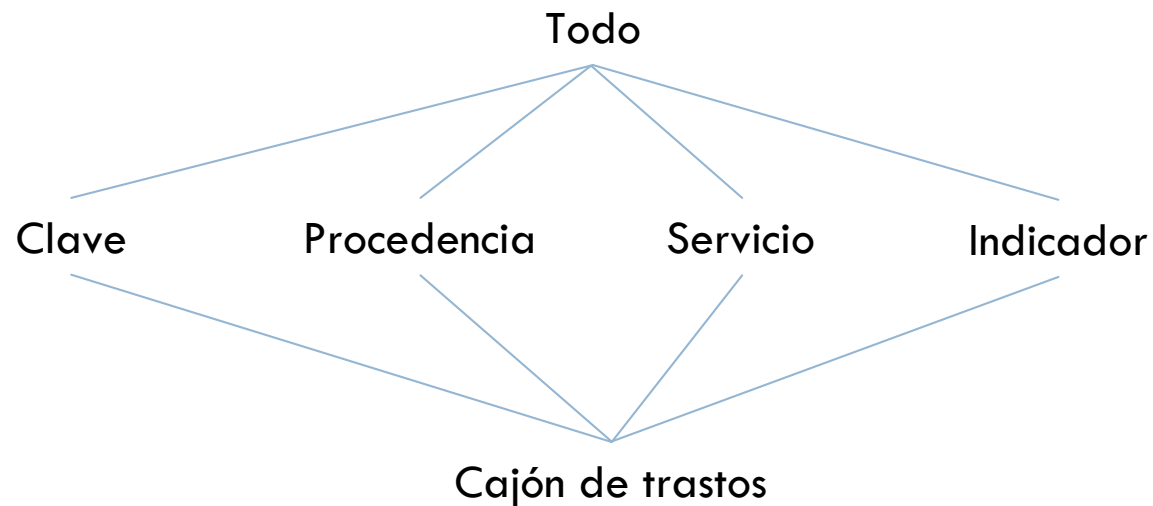
## 2.2. Diseño Conceptual

69

### □ CASO 6: Matriculación de vehículos

#### ▣ Solución:

##### ■ “Cajón de trastos”



## 2.2. Diseño Conceptual

70

### □ CASO 6: Matriculación de vehículos

#### ▣ Solución:

- “Cajón de trastos” (si hay muchas instancias)
  - Se dividen las jerarquías en dos dimensiones diferentes
  - Se agrupan en la misma dimensión los datos que tenga más sentido combinar
  - NO se hacen particiones tipo “la mitad de los valores en uno y la mitad en otro”

## 2.2. Diseño Conceptual

71

### □ CASO 6: Matriculación de vehículos

#### ▣ Solución:

##### ■ Mediciones: Cantidad

- Será la cantidad de matriculaciones que se han producido
- Valores posibles: 1, 100, 345... lo que sea

##### ■ Bases:

- Hay una única base que está formada por el identificador de TODAS las dimensiones

## 2.2. Diseño Conceptual

72

- CASO 7: Encuestas de calidad de la UGR
  - ▣ Datos que tiene la encuesta (ver T2-Anexo-2)
    - Profesor
    - Asignatura
    - Sexo
    - Edad
    - Grado de interés
    - Grado de dificultad
    - Nivel de asistencia a clase
    - Nivel de asistencia a tutorías
    - Fecha (no está, pero se sabe)
    - 25 preguntas valoradas de 1 a 5 ó NS/NC



## 2.2. Diseño Conceptual

73

### □ CASO 7: Encuestas de calidad de la UGR

#### ▣ Datos que tiene la encuesta:

- Profesor → ¿A quién se evalúa?
- Asignatura → ¿Qué se evalúa?
- Sexo → ¿Quién evalúa?
- Edad → ¿Quién evalúa?
- Grado de interés → ¿Quién evalúa?
- Grado de dificultad → ¿Quién evalúa?
- Nivel de asistencia a clase → ¿Quién evalúa?
- Nivel de asistencia a tutorías → ¿Quién evalúa?
- Fecha → ¿Cuándo se evalúa?
- 25 preguntas valoradas de 1 a 5 ó NS/NC

## 2.2. Diseño Conceptual

74

- CASO 7: Encuestas de calidad de la UGR
  - ▣ Foco de atención: Encuesta
  - ▣ Granularidad: Encuesta individual
  - ▣ Dimensiones:
    - Quién evalúa (Alumno)
    - Cuándo se evalúa (Curso académico)
    - A quién se evalúa (Profesor)
    - Qué se evalúa (Asignatura)

## 2.2. Diseño Conceptual

75

### □ CASO 7: Encuestas de calidad de la UGR

#### ▣ Dimensiones:

##### ■ Quién evalúa (Alumno)

- No tenemos información del alumno particular que hace la encuesta (son anónimas)
- Tendremos que analizar los resultados a nivel de “Perfil de Alumno”

##### ■ Cuándo (Curso académico)

- La opción habitual de Fecha y Hora no aporta nada para este análisis
- Tendremos que considerar cuatrimestres y curso académico

## 2.2. Diseño Conceptual

76

### □ CASO 7: Encuestas de calidad de la UGR

#### ▣ Dimensiones:

- A quién se evalúa (Profesor). Preguntarse si influyen:
  - El sexo del profesor
  - El tramo de edad en el que se encuentra
  - Antigüedad
- Qué se evalúa (Asignatura)
  - Asumimos que cada asignatura es de un único departamento
  - ¿La imparte un único profesor?

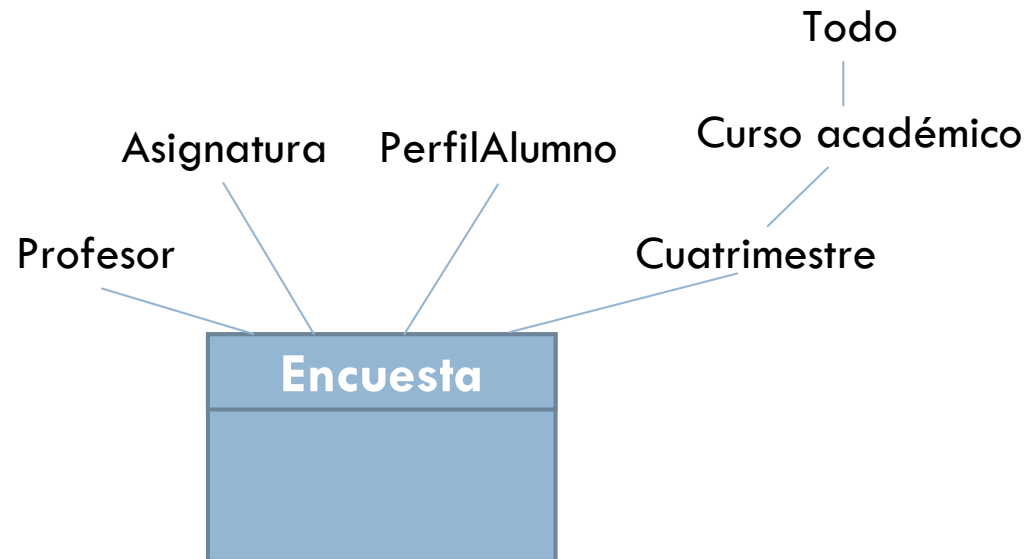
## 2.2. Diseño Conceptual

77

### □ CASO 7: Encuestas de calidad de la UGR

#### ▣ Dimensiones:

#### ■ Modelo conceptual:

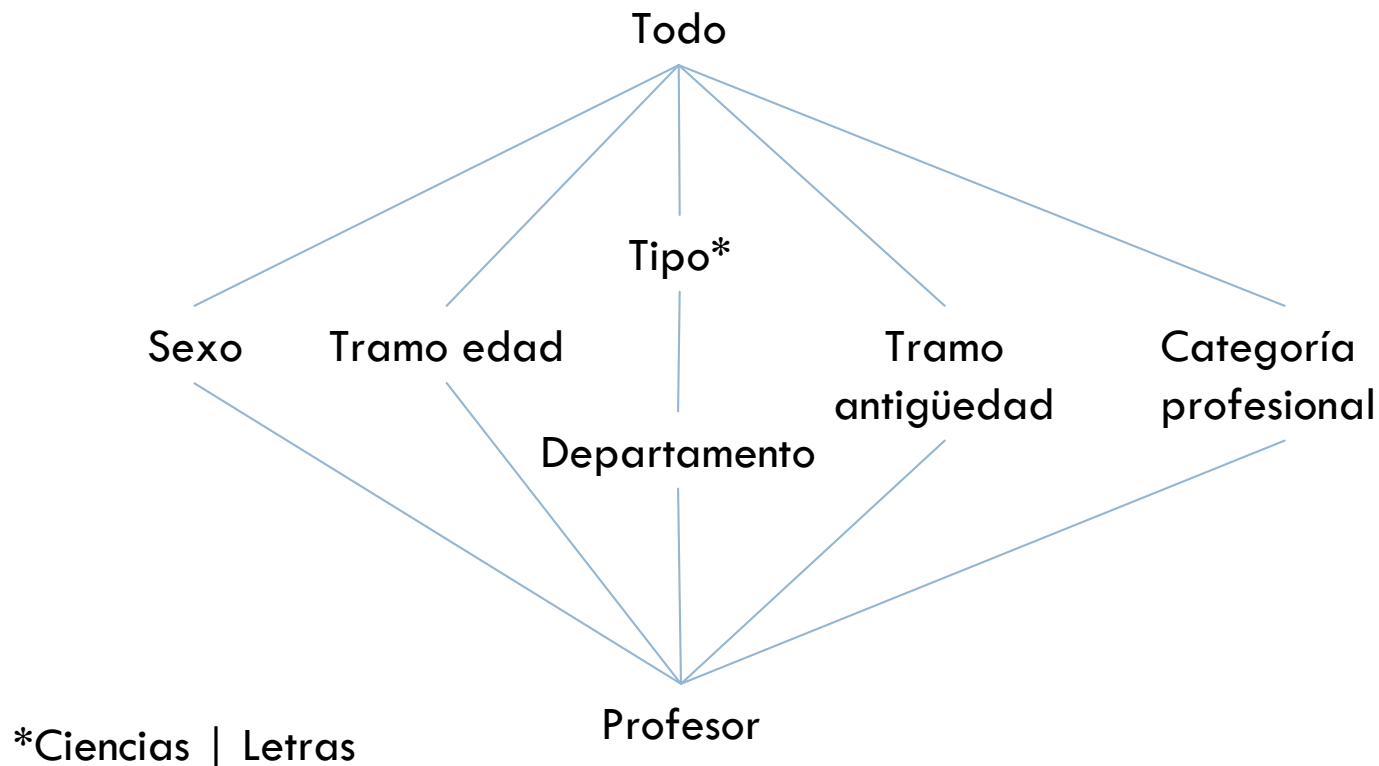


- ¿Pueden estar en la misma dimensión profesor y asignatura?

## 2.2. Diseño Conceptual

78

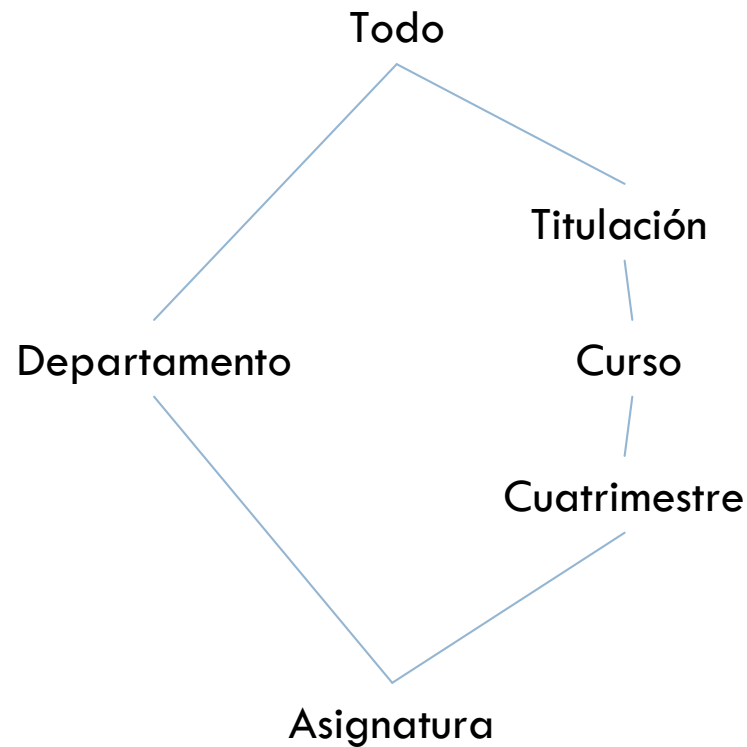
- CASO 7: Encuestas de calidad de la UGR
  - ▣ Modelo conceptual: dimensión Profesor



## 2.2. Diseño Conceptual

79

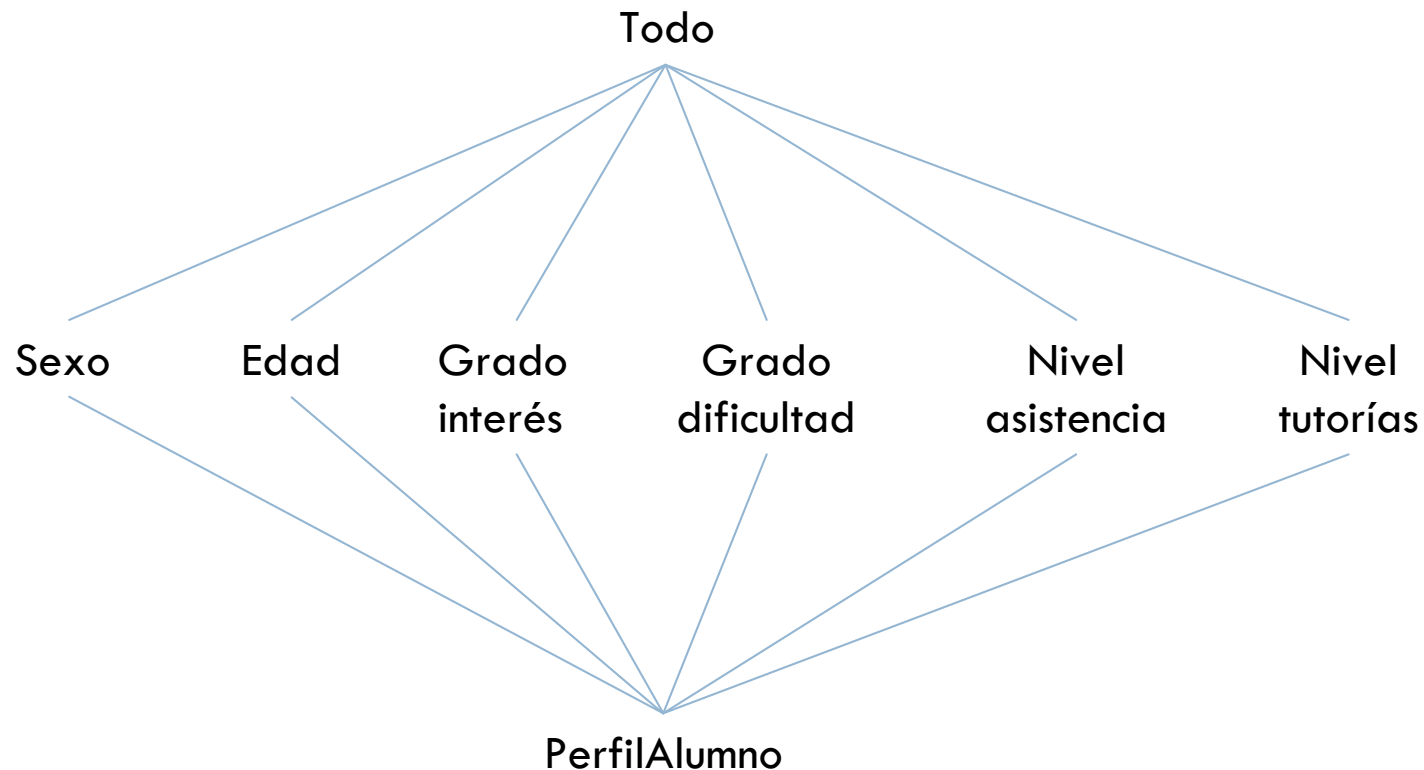
- CASO 7: Encuestas de calidad de la UGR
  - ▣ Modelo conceptual: dimensión Asignatura



## 2.2. Diseño Conceptual

80

- CASO 7: Encuestas de calidad de la UGR
  - ▣ Modelo conceptual: dimensión PerfilAlumno





## 2.2. Diseño Conceptual

81

### □ CASO 7: Encuestas de calidad de la UGR

#### ▣ Dimensiones:

- Si queda información sin clasificar → Cajón de trastos

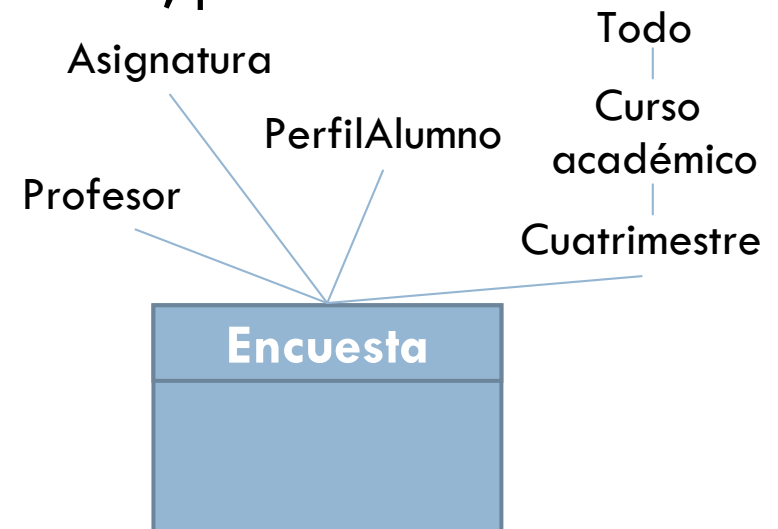
#### ▣ Bases:

- Profesor, asignatura, cuatrimestre-curso, perfilAlumno

- Puede haber varias encuestas para la base anterior

- Dos opciones:

- “Resumen de encuestas”
- Añadir N° encuesta



## 2.2. Diseño Conceptual

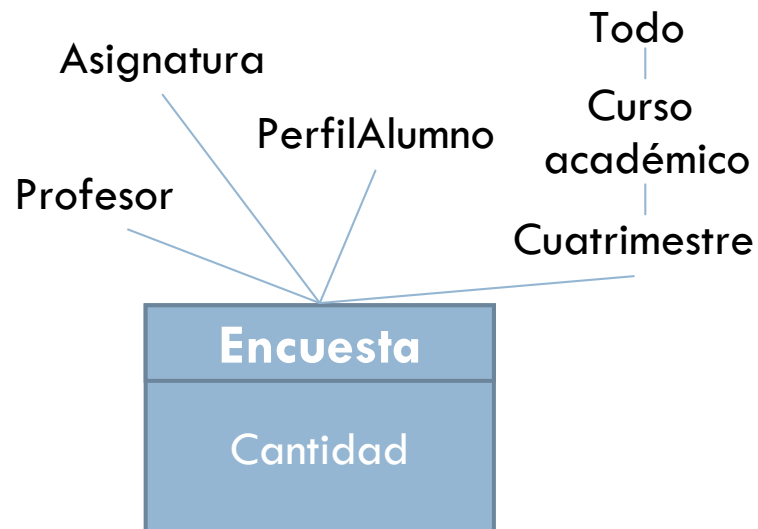
82

### □ CASO 7: Encuestas de calidad de la UGR

#### ▣ Bases:

##### ■ Opción 1: “Resumen de encuestas”

- Hay que añadir la medición Cantidad para indicar cuántas encuestas hay del mismo tipo (Cantidad  $\geq 1$ )
- Se pierde algo de información (no distinguimos cada encuesta)



## 2.2. Diseño Conceptual

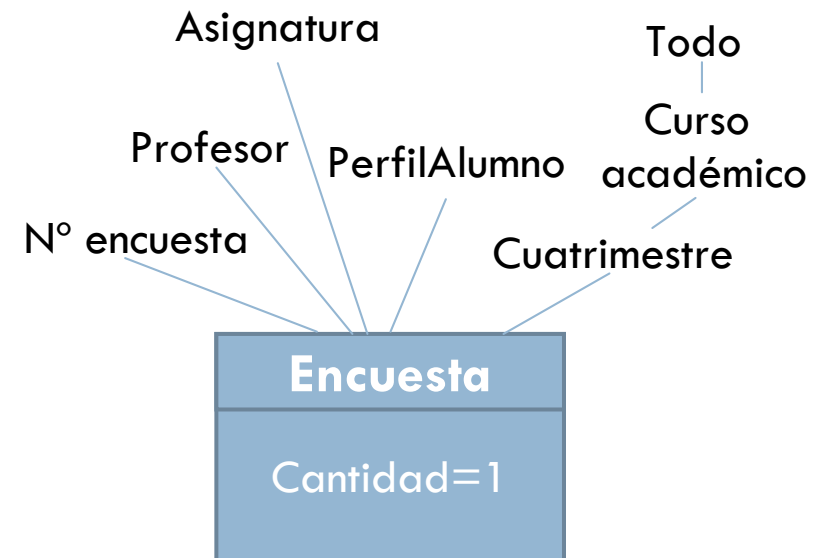
83

### □ CASO 7: Encuestas de calidad de la UGR

#### □ Bases:

#### ■ Opción 2: Añadir N° de encuesta (identificador)

- Ver si es único para toda la Universidad, cada profesor, cada asignatura...
- Elegir la base en consecuencia
- El N° de encuesta iría en una dimensión aparte
- Mediciones → Cantidad = 1
- El N° de encuesta se asigna al procesar la encuesta



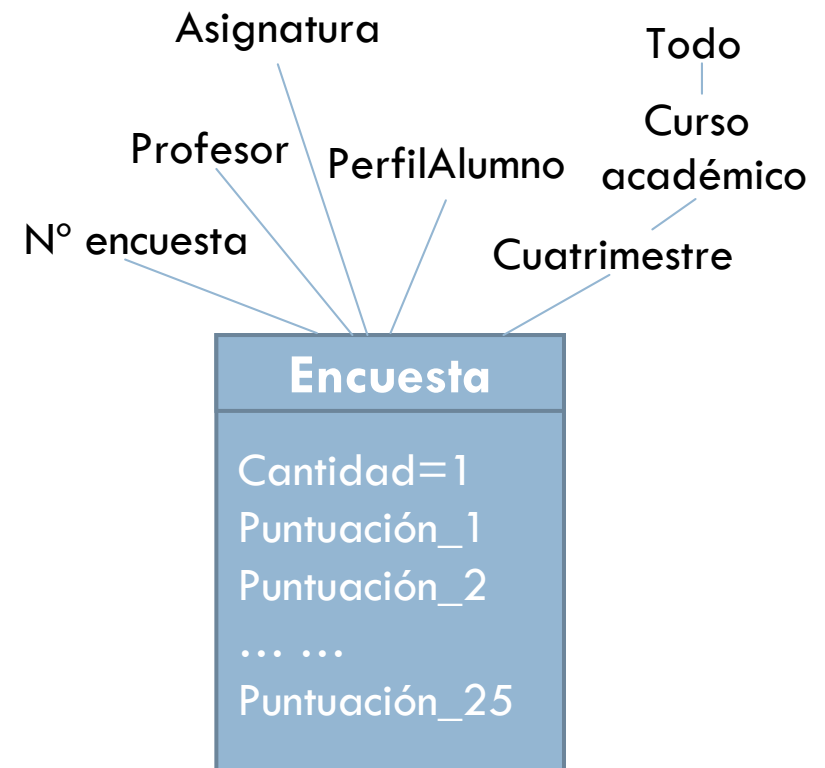
## 2.2. Diseño Conceptual

84

### □ CASO 7: Encuestas de calidad de la UGR

#### ▣ Mediciones (Opción a):

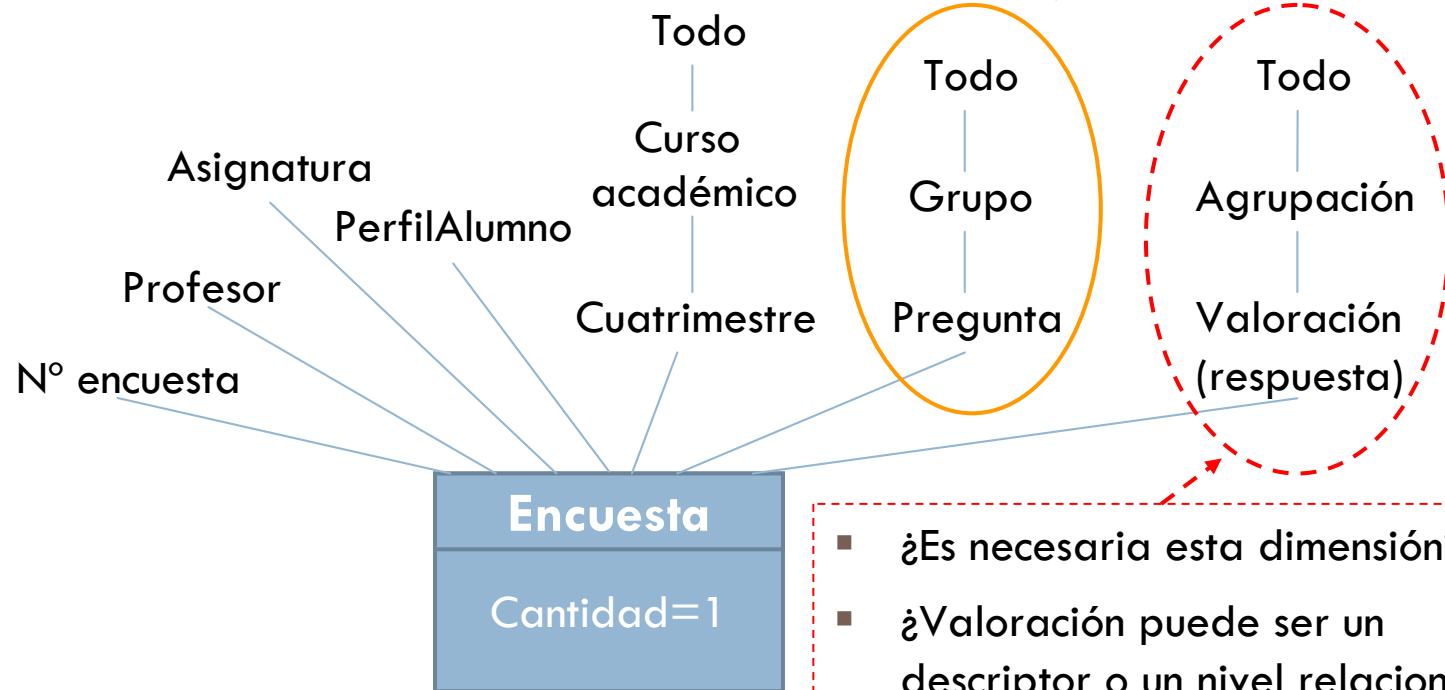
- Una medición para cada pregunta (25 preguntas → 25 mediciones)
- La respuesta NS/NC no es numérica → No puede estar en los hechos
- Si el cuestionario tiene muchas preguntas no es una solución válida



## 2.2. Diseño Conceptual

85

- CASO 7: Encuestas de calidad de la UGR
  - ▣ Mediciones (Opción b): Encuestas con datos no numéricos
    - Añadir dimensión de mediciones



## 2.2. Diseño Conceptual

86

- CASO 7: Encuestas de calidad de la UGR
  - ▣ Cuestiones sin resolver
    - ¿Cuáles son las bases en cada caso?
    - ¿Cuál es la aditividad de las mediciones?
    - ¿Cuál sería el tamaño estimado del sistema multidimensional?

## 2.2. Diseño Conceptual

87

### □ OTROS CASOS (ver T2-Anexo-3)

#### ▣ Encuestas con preguntas relacionadas

- Añadir dimensión de mediciones

#### ▣ Análisis clínicos

- Tratar el caso de forma similar a una encuesta
- Distinguir entre preguntas con valores numéricos y no numéricos

## 2.3. Diseño Lógico

88

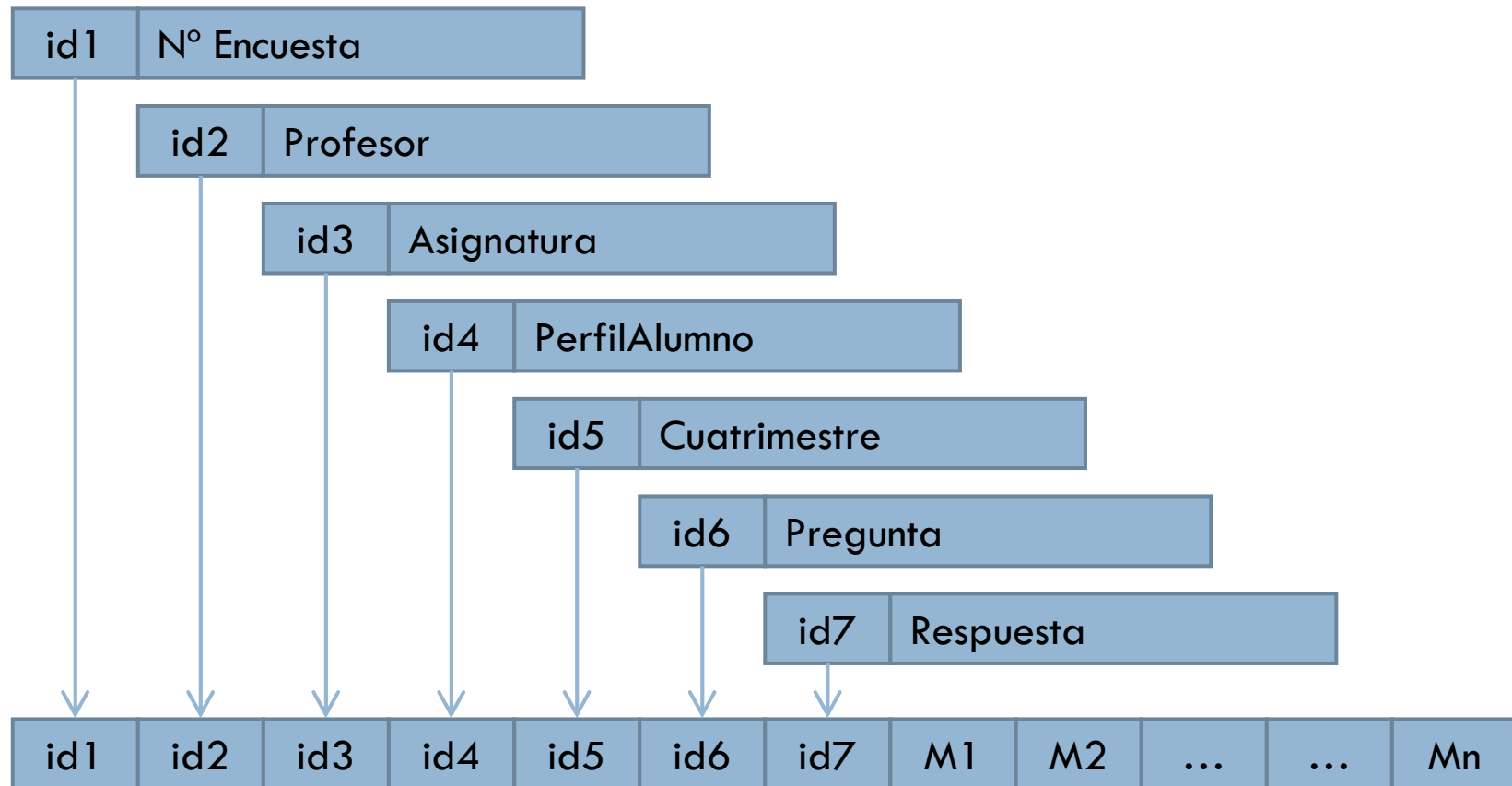
- Objetivo del diseño lógico: Implementar el diseño conceptual
- Habitualmente se sigue el modelo ROLAP en estrella; en este caso la regla general es:
  - ▣ Una tabla para cada dimensión
  - ▣ Una tabla para los hechos
  - ▣ Llaves generadas para relacionar las tablas



## 2.3. Diseño Lógico

89

- Para el ejemplo de las encuestas:

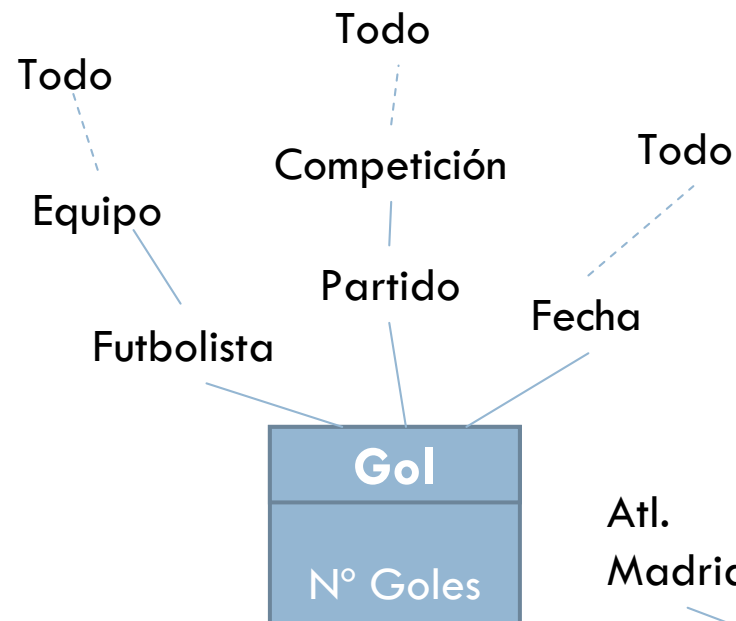


- ¿Se necesita la tabla del N° de encuesta? → **Excepción:** Dimensión degenerada

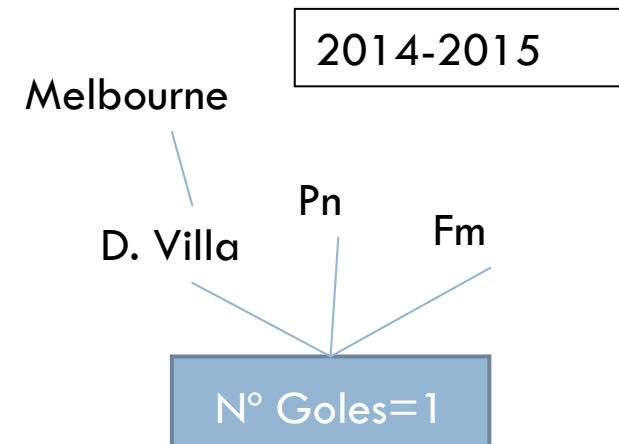
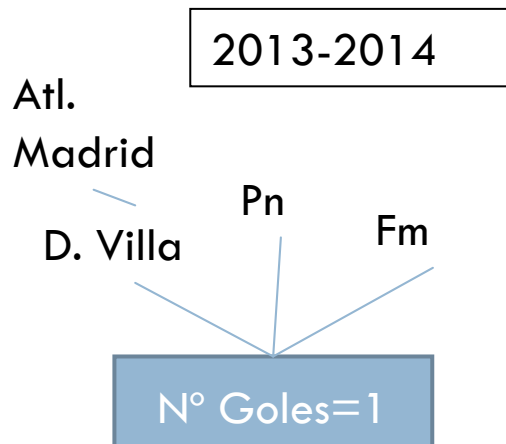
## 2.3. Diseño Lógico

90

### □ DIMENSIONES CAMBIANTES



Ejemplo: David Villa, que en la temporada actual está en el Melbourne City y en la anterior estaba en Atl. de Madrid



## 2.3. Diseño Lógico

91

- En la temporada anterior tendríamos una implementación en una tabla:

id	dni	futbolista	equipo	...
----	-----	------------	--------	-----

(o nº de registro en  
la Federación...)

1	dni1	futbolista1	equipo1	...
2	dni2	futbolista2	equipo2	...

x	zzz	David Villa	Atl.Madrid	...
---	-----	-------------	------------	-----

En una BD haríamos UPDATE... ¿y en un SMD?

## 2.3. Diseño Lógico

92

- En un SMD sería un error hacer UPDATE porque cambiaríamos la historia → Asignaríamos los goles del equipo anterior como si fuesen del equipo actual
  - ▣ Habría que dejar lo que había
  - ▣ Crear un nuevo jugador con el mismo nombre y distinto equipo

## 2.3. Diseño Lógico

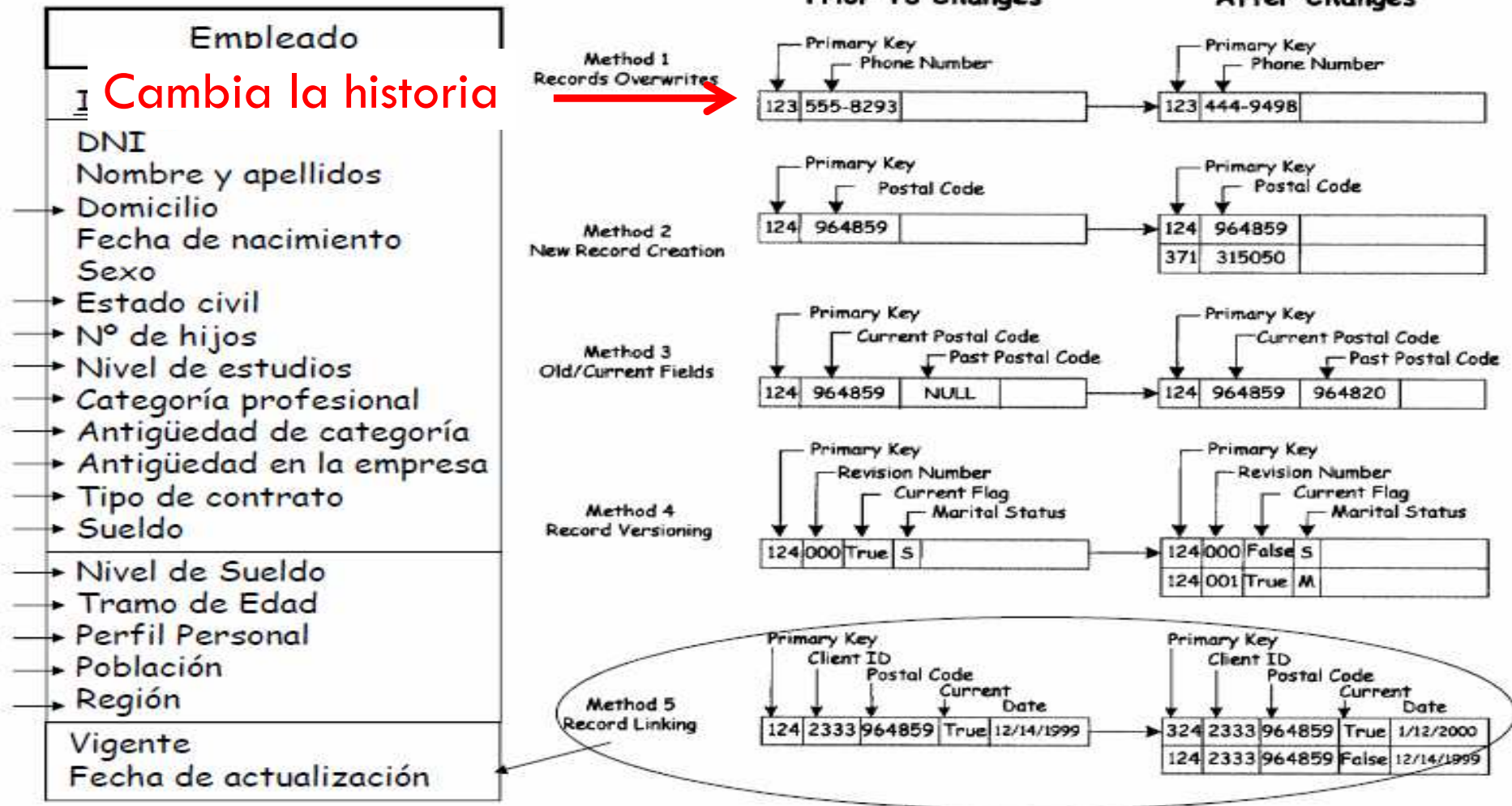
93

- Características de las dimensiones cambiantes
  - ▣ Al cambiar los datos habrá que añadir nuevos registros con nuevas claves (generadas); estas claves se usan como llaves primarias
  - ▣ Se mantienen otros datos repetidos; se pueden utilizar como llaves alternativas para agrupar (p.e., dni, nombre)

## 2.3. Diseño Lógico

94

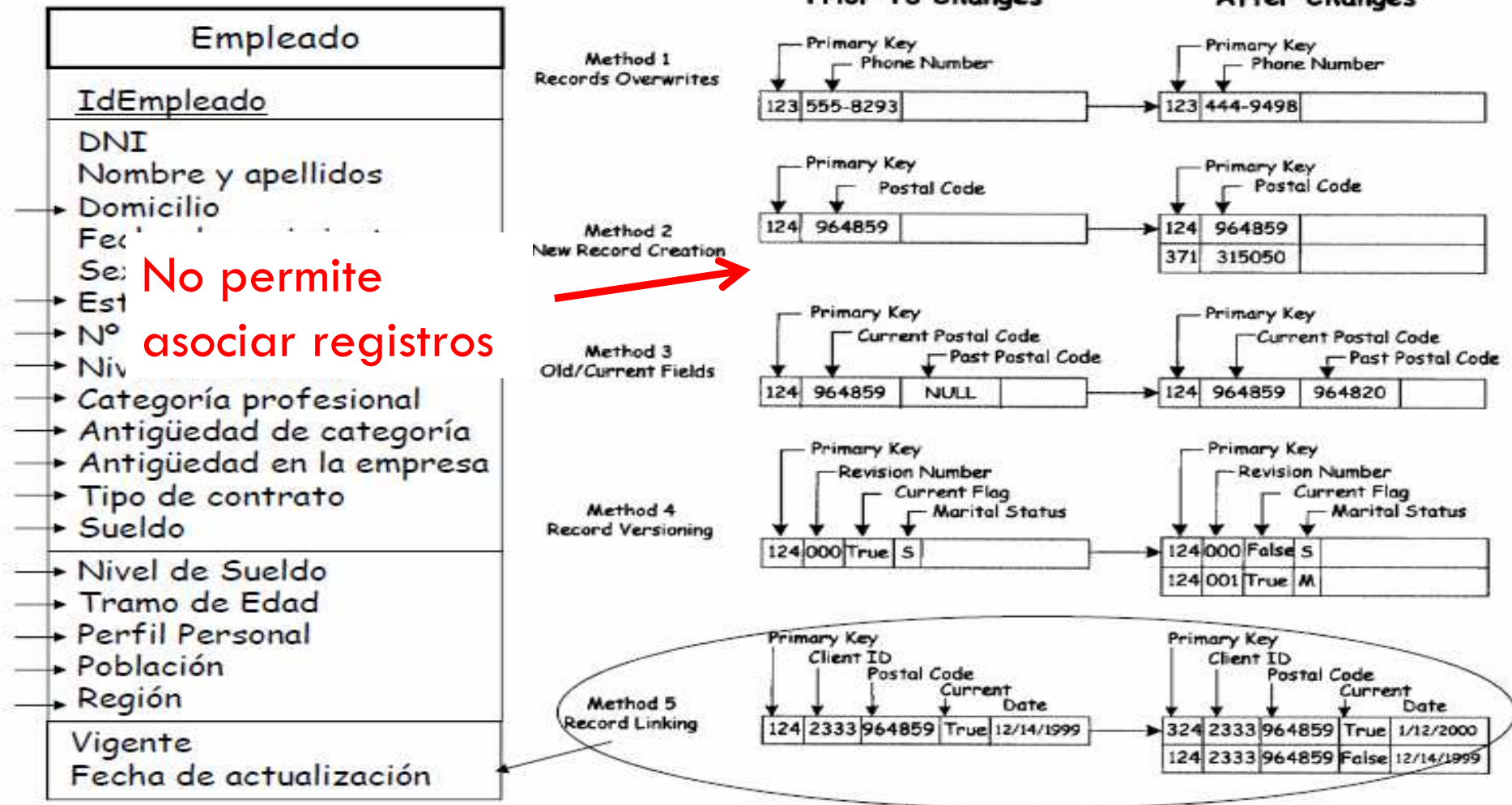
Dimensiones cambiantes



## 2.3. Diseño Lógico

95

### Dimensiones cambiantes

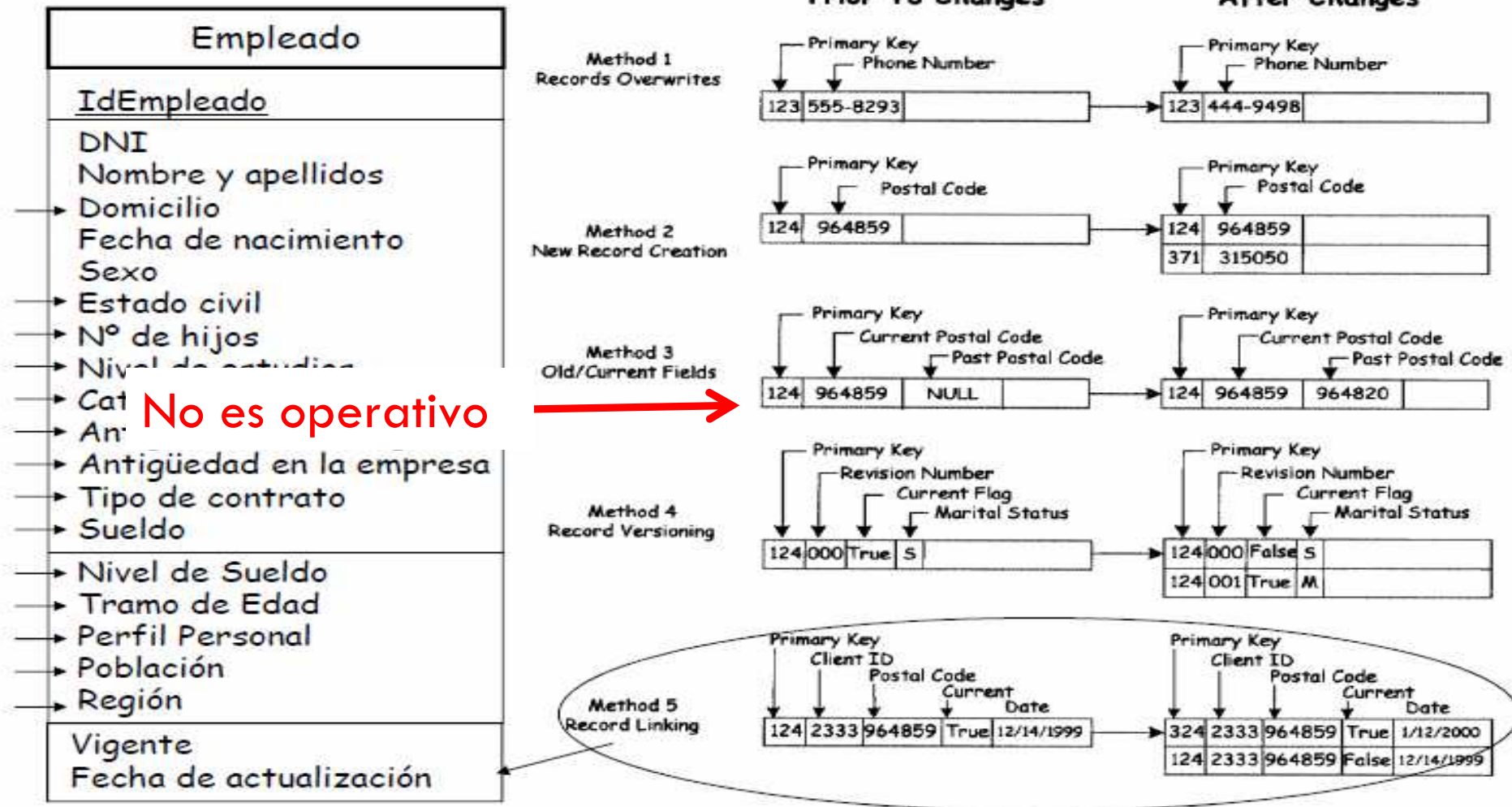




## 2.3. Diseño Lógico

96

### Dimensiones cambiantes

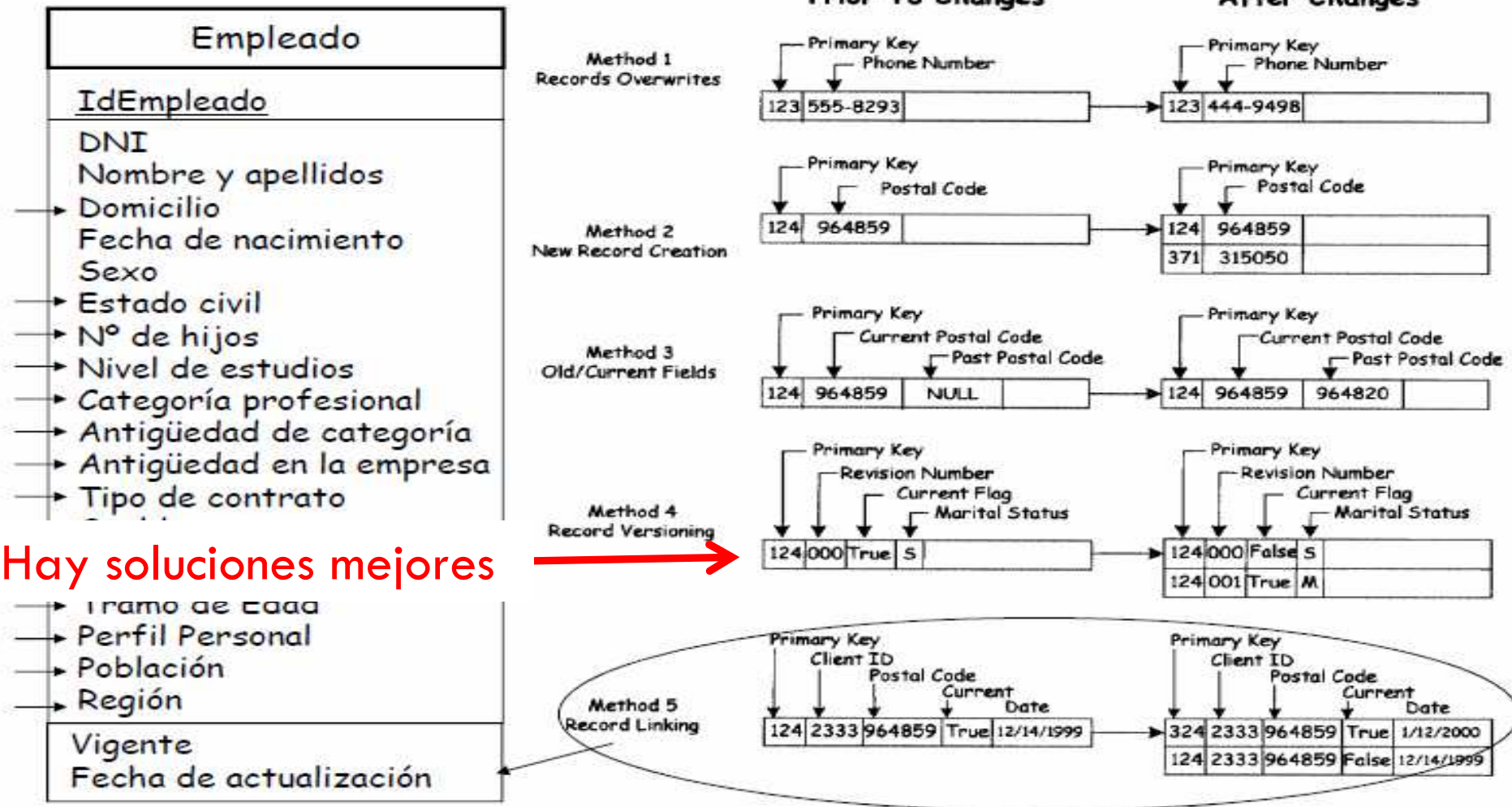




## 2.3. Diseño Lógico

97

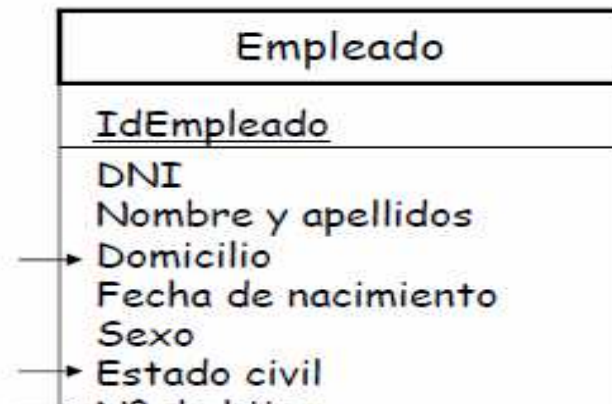
Dimensiones cambiantes



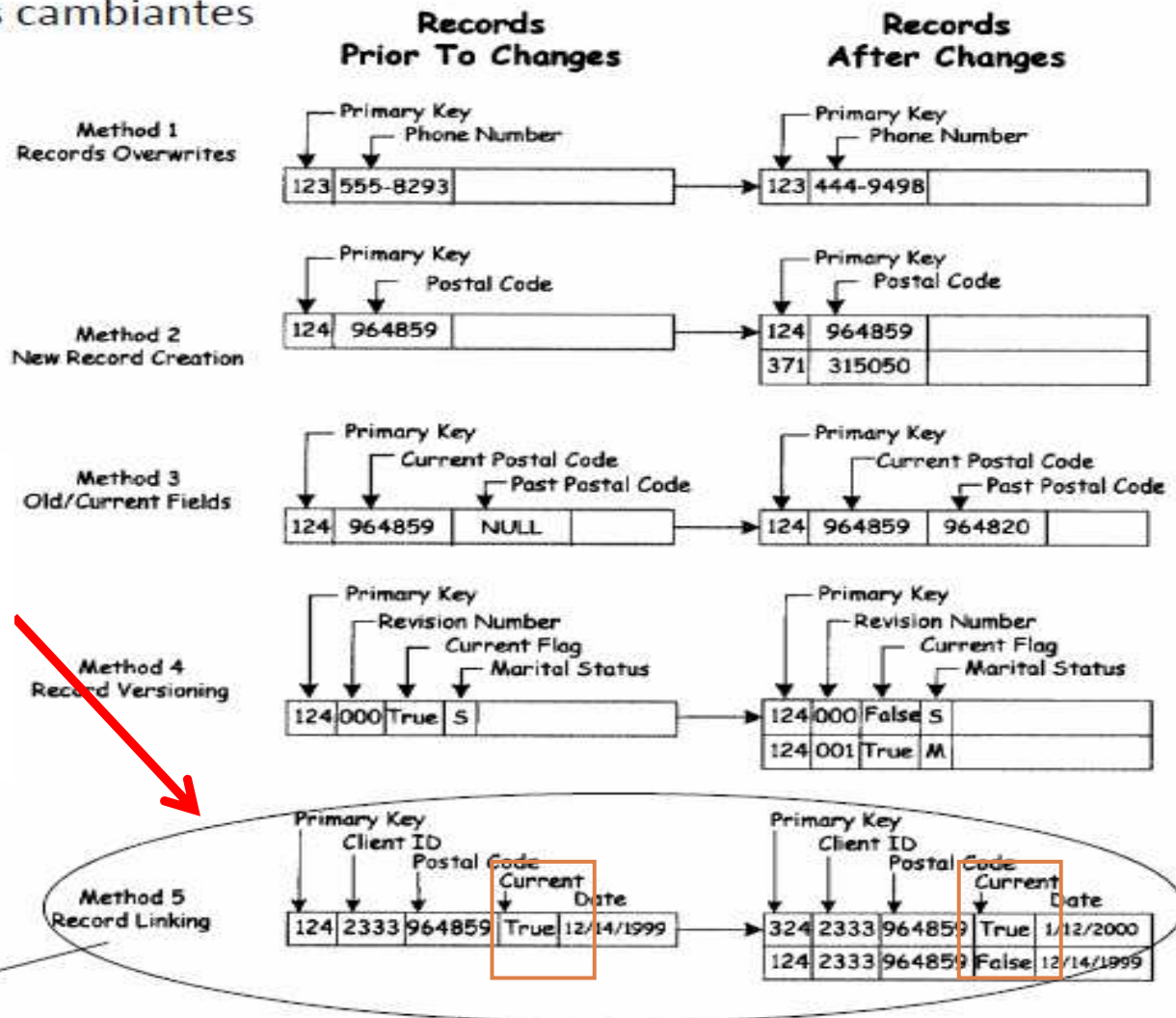
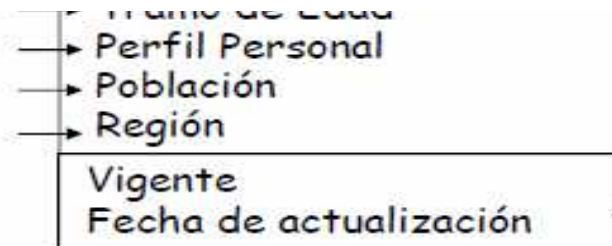
## 2.3. Diseño Lógico

98

### Dimensiones cambiantes



Una buena solución:  
se añade un campo que  
indica la situación  
actual (vigente)



## 2.3. Diseño Lógico

99

- Considerando el método 5, varios registros representan a un futbolista
  - ▣ Problema:
    - Cambios frecuentes → Muchos registros
    - Muchos registros → Dimensión muy grande
    - Ejemplo: Cambios en nivel de estudios, número de hijos...
  - ▣ Esta solución es buena para dimensiones lentamente cambiantes (donde hay pocos cambios)

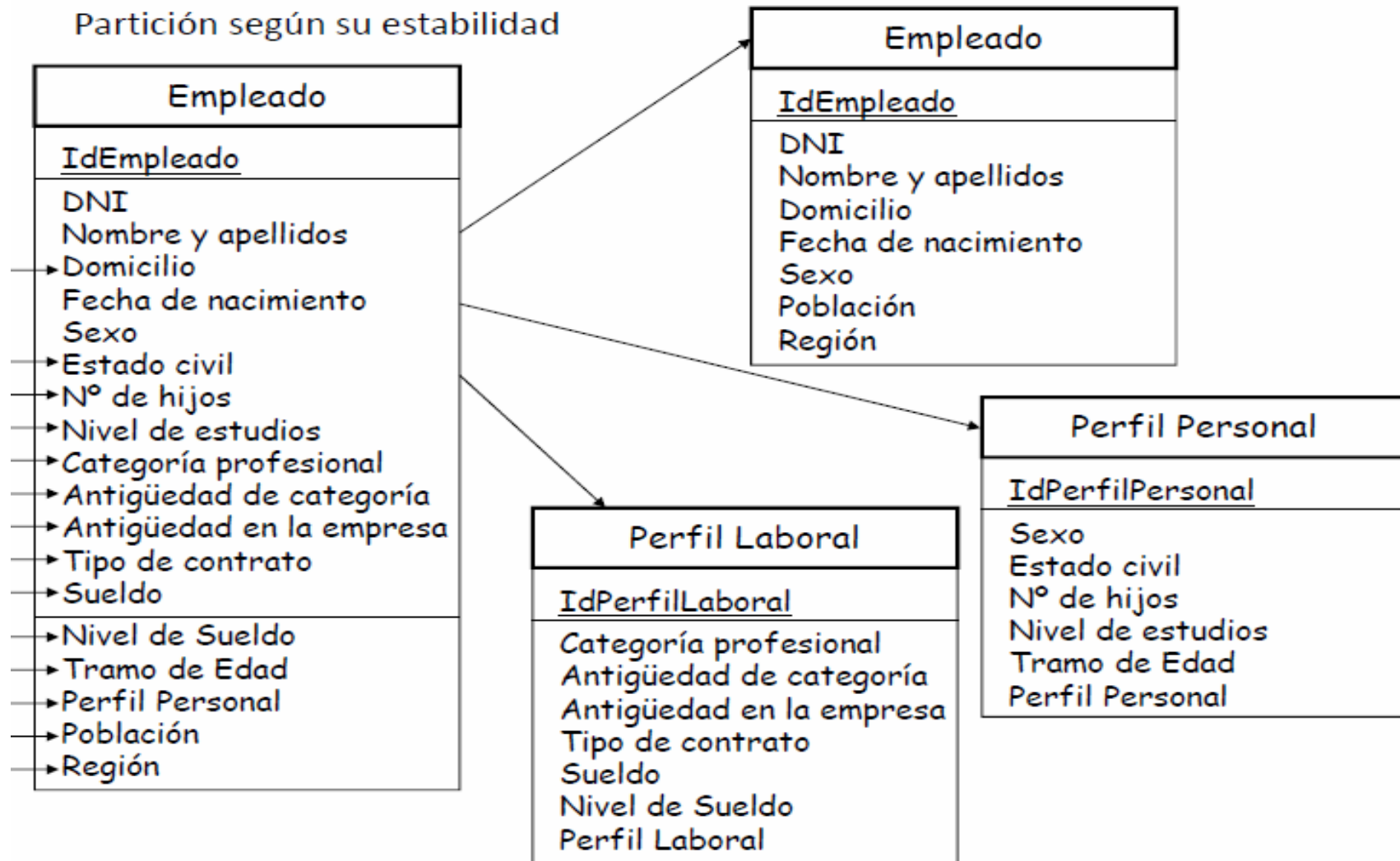
## 2.3. Diseño Lógico

100

- ¿Y si hay muchos cambios? ¿es necesario conservarlos?
  - ▣ Hablar con el decisor
- Si decide que los necesita:
  - ▣ Crear una tabla con la parte que cambia
  - ▣ Conceptualmente, NO es una dimensión nueva
  - ▣ Los datos **se particionan según su estabilidad**
  - ▣ **Excepción a la regla general:**
    - Las dimensiones cambiantes se implementan como varias tablas → Desdoblamiento de dimensiones

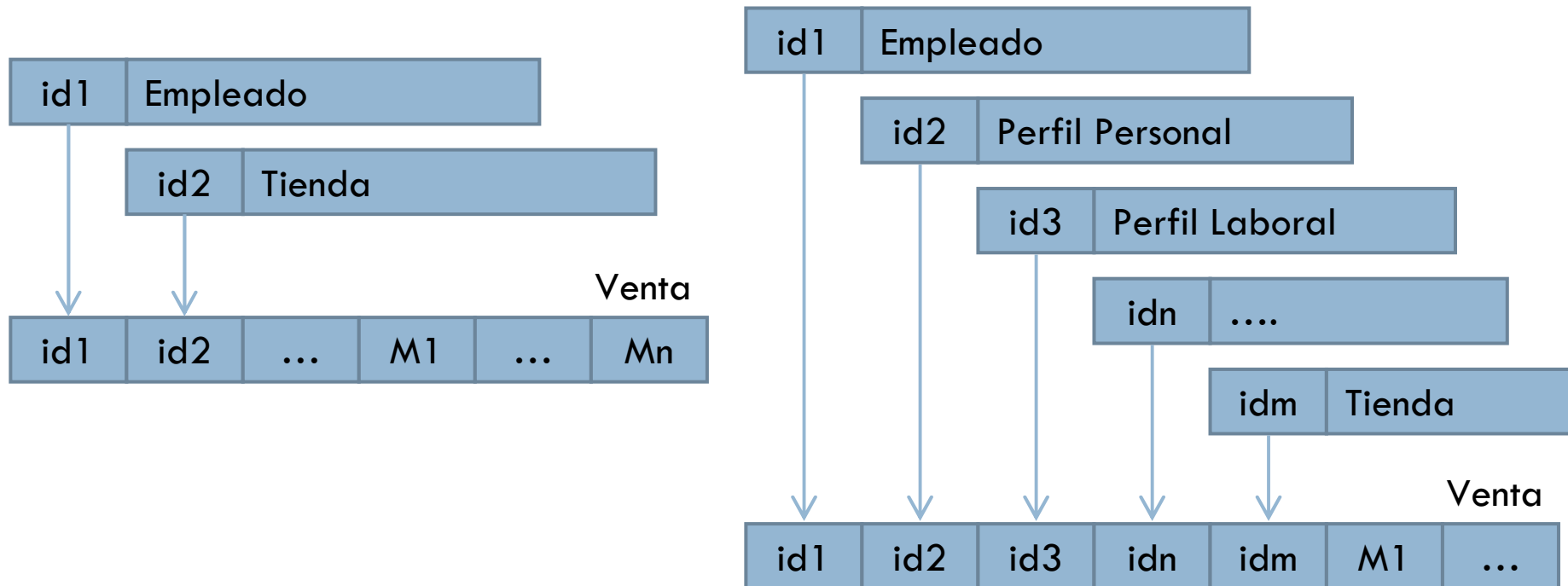
## 2.3. Diseño Lógico

101



## 2.3. Diseño Lógico

102

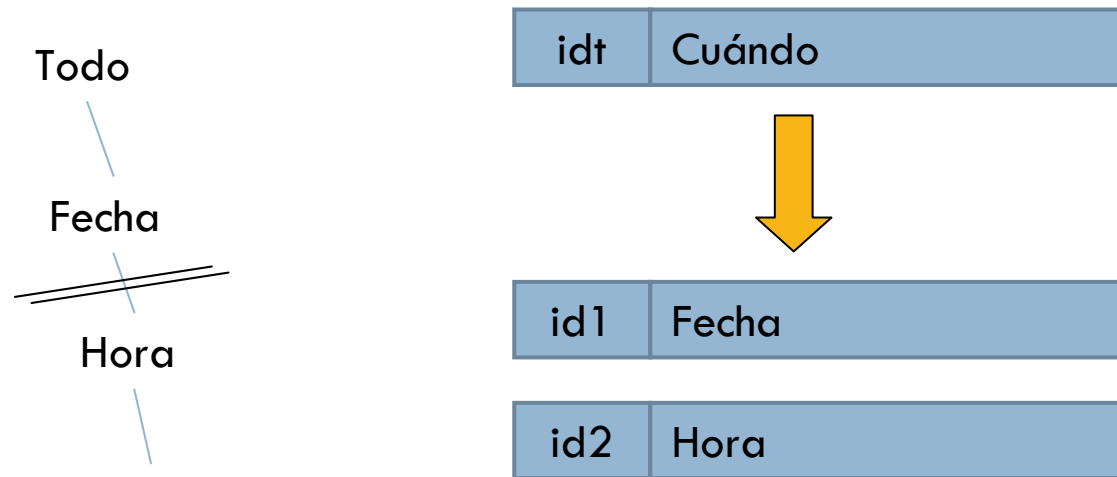


- ¿Sería estrella o copo de nieve?
  - ▣ Estrella, porque la relación viene dada por la tabla de hechos

## 2.3. Diseño Lógico

103

- Un ejemplo de desdoblamiento de dimensiones sería la Fecha-Hora, donde las combinaciones se disparan



## 2.3. Diseño Lógico

104

### □ MEDICIONES

#### ▣ Dos posibilidades de implementación

- Almacenadas: requiere espacio
- Calculadas: requiere tiempo

### □ Ejemplo Censo

- ▣ Población = Hombres + Mujeres
- ▣ Hombres: aditiva, almacenada
- ▣ Mujeres: aditiva, almacenada
- ▣ Población: aditiva, ¿almacenada o calculada?



## 2.3. Diseño Lógico

105

- Ejemplo Censo: medición Población
  - ▣ Se puede hacer de las dos formas, porque siempre hay que sumar
  - ▣ Preferiblemente calculada (depende de si su uso es frecuente o no)
- Ejemplo Venta
  - ▣  $\text{Importe} = \text{Cantidad} * \text{PVP}$  (aditiva)
  - ▣  $\text{Cantidad} = \text{Importe} / \text{PVP}$  (Aditiva)
  - ▣  $\text{PVP} = \text{Importe} / \text{Cantidad}$  (No aditiva)
- **Regla:** Almacenar las aditivas y calcular el resto, si se puede

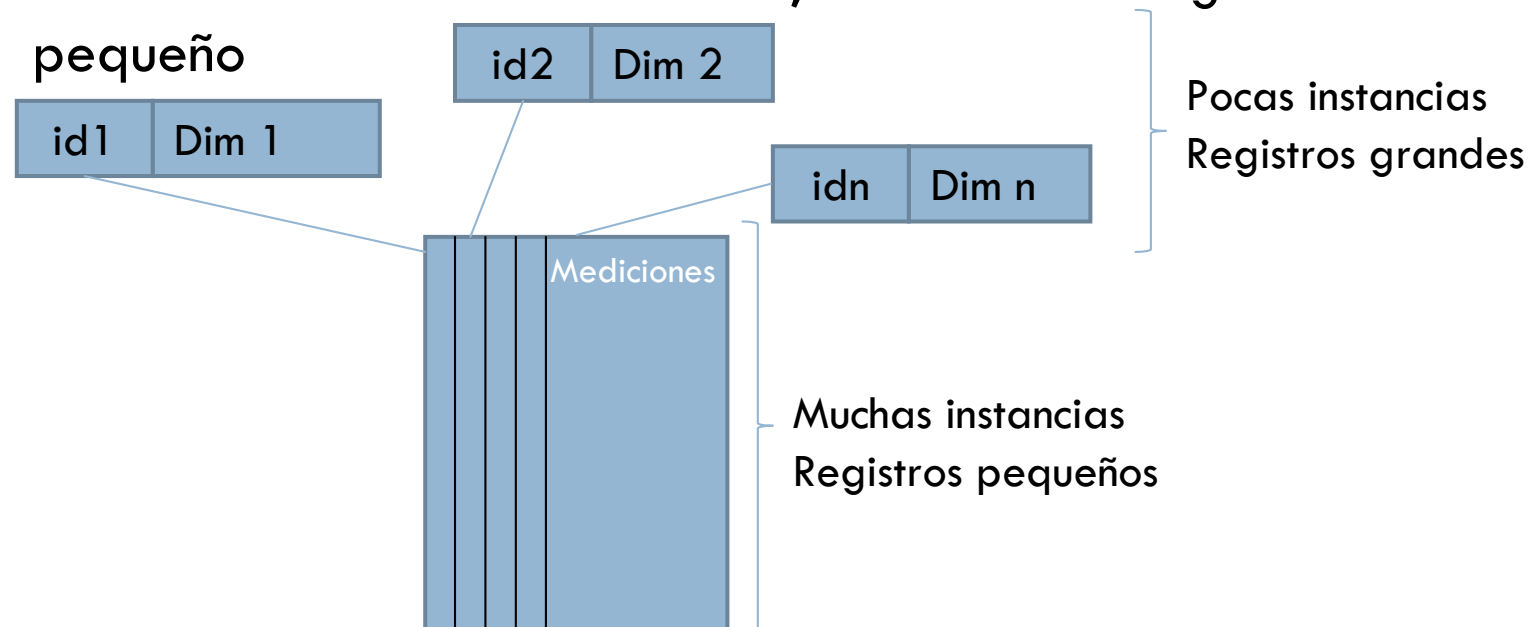
## 2.4. Diseño Físico

106

### □ ÍNDICES

□ Básicamente, lo que se tiene es lo siguiente:

- Pocas instancias en las dimensiones, donde cada registro es grande
- Muchas instancias en los hechos, donde cada registro es pequeño



## 2.4. Diseño Físico

107

### □ ÍNDICES

#### ▣ Definimos índices para:

##### ■ Las dimensiones:

- Llaves primarias (generadas)
- En los niveles (para HOLAP o ROLAP, no en MOLAP)
- En los descriptores que se usen mucho en las consultas

##### ■ Los hechos:

- Llaves externas

## 2.4. Diseño Físico

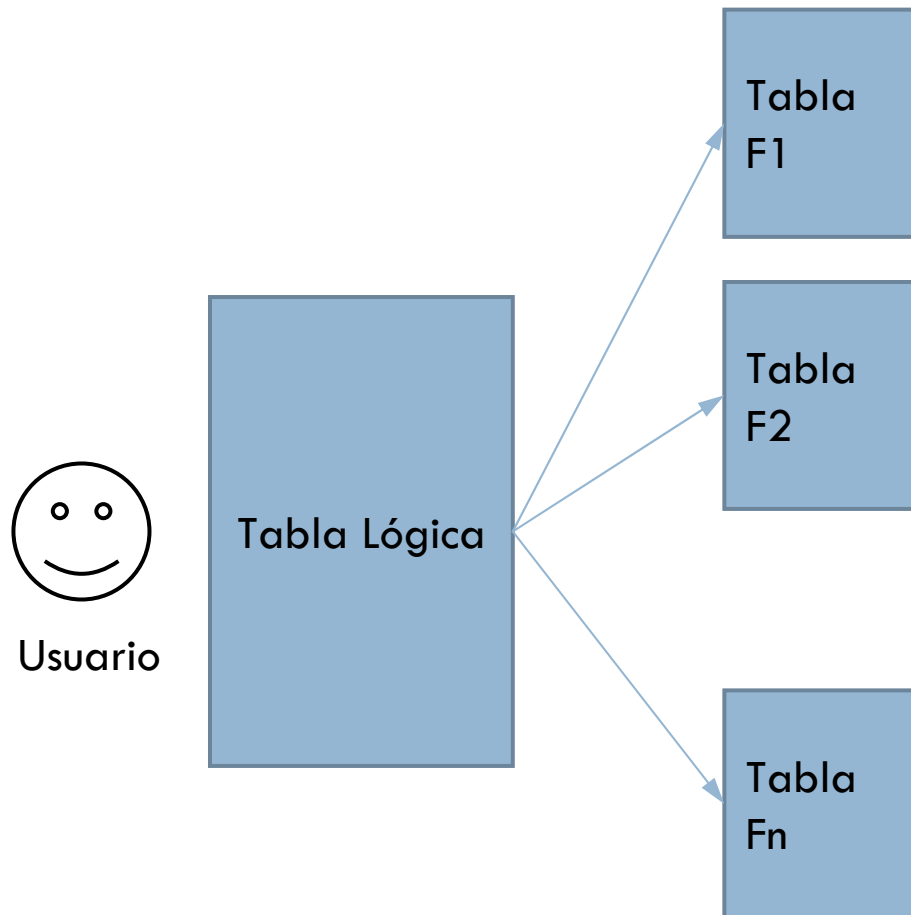
108

### □ GESTIÓN DE TABLAS

- ▣ Supongamos que se requiere una copia de seguridad
  - Dimensiones no hay problema
  - Hechos, complicado por el número de registros
- ▣ Solución: particionado de tablas
  - Es una funcionalidad del SGBD
  - Tenemos una tabla lógica implementada en varias físicas
  - Es transparente al usuario
  - El administrador de la BD pueden redefinir las tablas físicas en función de claves de partición

## 2.4. Diseño Físico

109

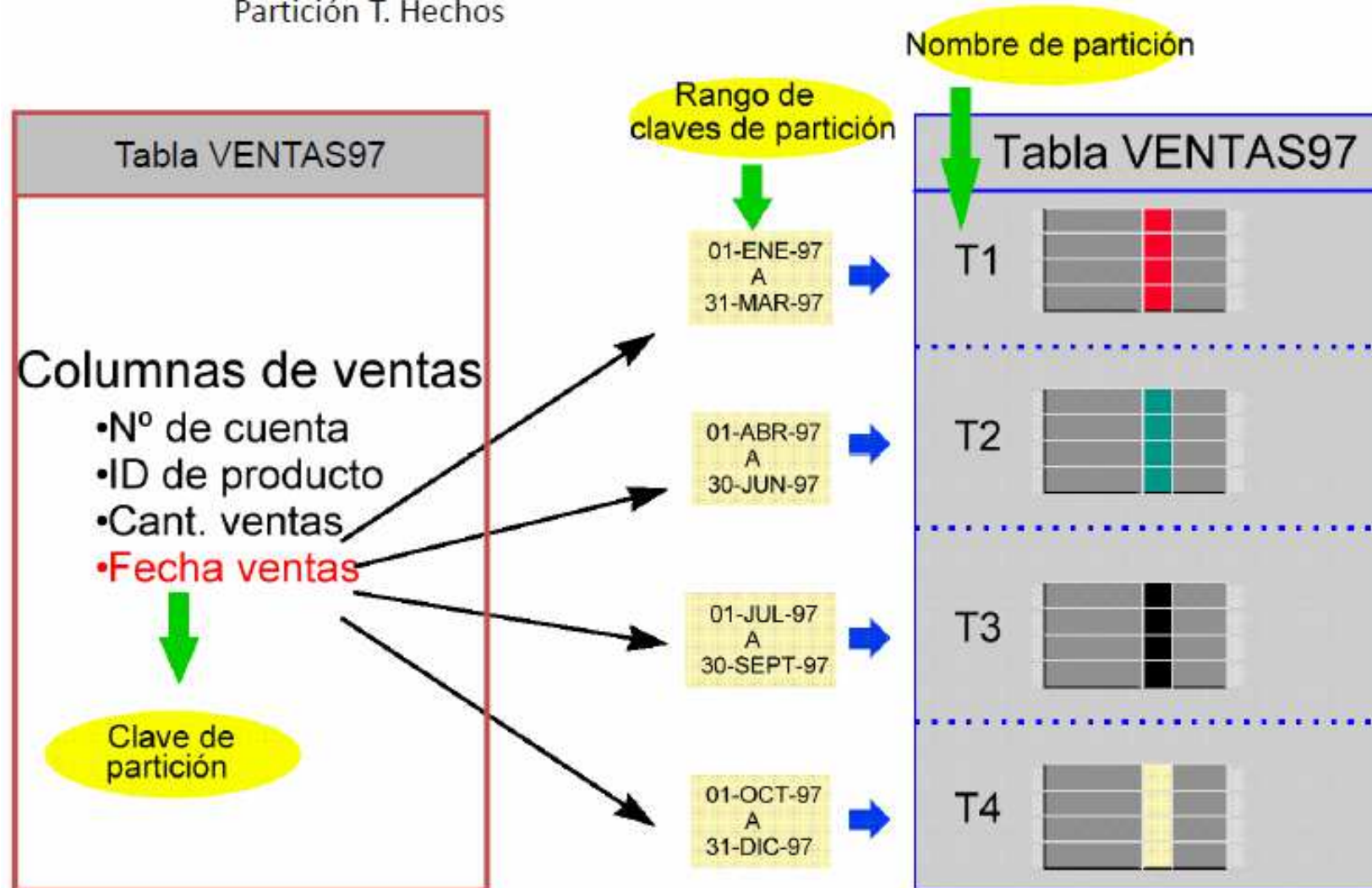


- La dimensión “cuándo” guarda la historia
- La clave de partición podría depender del tiempo
- Buena opción para la tabla de hechos y copias de seguridad
  - ▣ Sólo se guarda la última tabla

## 2.4. Diseño Físico

110

Partición T. Hechos



## 2.4. Diseño Físico

111

- Adicionalmente, se podrían usar nodos de distribución

