



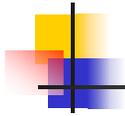
Diseño lógico relacional de DW

■ Metodologías

- Pasaje / traducción del esquema conceptual multidimensional al esquema lógico (ya visto)
- Diseño a partir de las fuentes
 - Partiendo del esquema **conceptual** integrado de las fuentes (MER)
 - Partiendo del esquema **lógico** integrado de las fuentes (Modelo Relacional)

■ Refinamientos

- Estudio de problemas frecuentes y sus soluciones



Refinamientos al esq. relacional

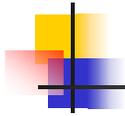
- **Agregaciones**
- **Dimensiones que cambian lentamente**
- **Minidimensiones**
- **Roles**
- **Dimensiones “Many-to-Many”**
- **Fragmentación de tabla de hechos**

Referencias: [Kim96] [Kim98] [Son01] [Gol00]



Agregaciones

- Resúmenes pre-calculados
- **agregación**: registro de tabla de hechos representando un resumen de registros de nivel básico de esa tabla
- Un registro de hechos *agregado* siempre está asociado con uno o más registros de dimensión *agregados*
- Tablas de hechos **derivadas**
- Hay que crear claves artificiales en las dimensiones derivadas



Dim. que cambian lentamente

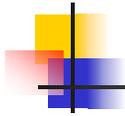
- **“Slowly changing dimensions” [Kim96]**
 - dimensiones “casi” constantes a través del tiempo
 - Alternativas para guardar la historia
 - 1. Crear un registro adicional cuando hay un cambio, con los nuevos valores
 - 2. Crear nuevos campos para guardar los nuevos valores conservando los viejos



Dim. que cambian lentamente

■ Alternativa 1

- Separar una relación de historia y una actual ?
- Manejo de la clave
 - Generalizar la clave existente
 - Agregando dígitos
 - Agregando otros atributos
 - Agregando atributo de tiempo
- Agregar fecha de todos modos ?



Dim. que cambian lentamente

■ Ejemplo

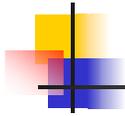
- Cliente 'Maria Perez'. Hasta 15/1/94 no era casada entonces el campo de ec tenia valor 'S'. En esa fecha se caso.
- Existen 2 descripciones correctas de Maria Perez.
- Los registros de hechos hasta el 15/1/94 tendrán a M.P. como 'S' y los reg. de hechos luego de esa fecha tendrán a M.P. como 'C'.
- Creamos un 2o. registro de cliente para Maria Perez.
- Generalizamos la clave de la dimensión: clave original + 2 dígitos de versión.
- No es necesario guardar la fecha de cambio en la dimensión. Los datos de los hechos se relacionan con la versión correspondiente.
- No se puede usar el nuevo valor con historia anterior.



Dim. que cambian lentamente

■ Alternativa 2

- Agregamos un nuevo campo para el atr. afectado.
- Ver los valores viejos y los nuevos hacia adelante y hacia atrás en el tiempo.
- Tiene sentido agregar también un campo de fecha de efectividad del valor actual.
- La única forma de partir la historia es usando la fecha de efectividad.
- No se guardan los valores intermedios. No se puede usar para describir con exactitud los cambios en un objeto.



Dim. que cambian lentamente

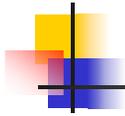
■ El mismo ejemplo ...

- Agregamos campos *ec_actual* y *fecha_ec*.
- Cada vez que el estado civil de 'Maria Perez' cambia, sobrescribimos el campo *ec_actual* y cambiamos la fecha de efectividad (*fecha_ec*). Dejamos siempre igual el original, campo *ec*.



Minidimensiones

- Cambios en grandes dimensiones
 - Separar una o mas **minidimensiones** de la tabla de dimensión, c/u consistente de pequeños grupos de atributos que han sido administrados para tener una cantidad limitada de valores.
- Ganancia significativa en performance



Minidimensiones

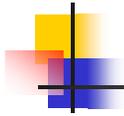
- **Ejemplo: Dimensión Cliente**
 - Minidimensión demográfica
 - guardamos solo las distintas combinaciones de atributos demográficos que realmente ocurren
 - edad o ingresos se agrupan en franjas
 - Se puede hacer join directamente a la fact table o a la dimensión Cliente (puede existir la foreign key en las 2)



Ejemplo

CLIENTES

NOMBRE	EDAD	NIVEL-INGRESOS	DIRECCION	SEXO	CIUDAD	EC
R. Mendez	20	10000	Bvar. Artigas 3	F	Mont.	S
S. Nunez	30	15000	J. Herrera y Ob	M	Mont.	C
M. Garcia	20	10000	Garzon 2125	F	Salto	S
L. Lopez	50	5000	18 de Julio 643	M	Colonia	C



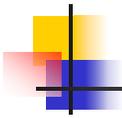
Ejemplo

DEMOGRAFICOS

COD_DEM	EDAD	NIVEL-INGRESOS	SEXO	EC
100	20	10000	F	S
200	30	15000	M	C
300	50	5000	M	C

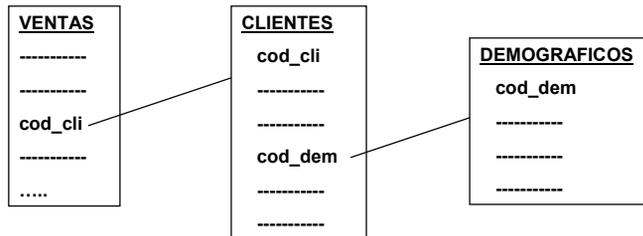
CLIENTES

NOMBRE	DIRECCION	CIUDAD	COD_DEM
R. Mendez	Bvar. Artiga	Mont.	100
S. Núñez	J. Herrera y	Mont.	200
M. Garcia	Garzon 2125	Salto	100
L. Lopez	18 de Julio	Colonia	300

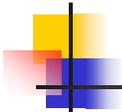


Ejemplo: historia

Opción A:

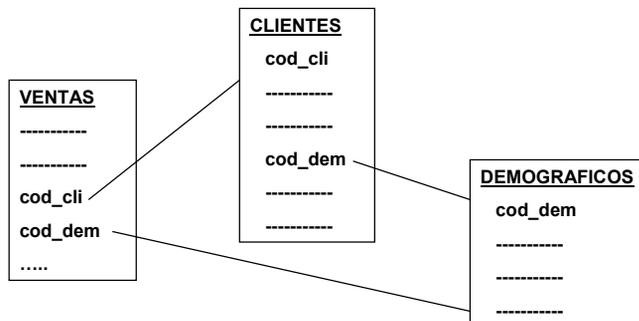


Si hay un cambio: Genero otro registro en Clientes con la clave foránea correspondiente



Ejemplo: historia

Opción B:



Si hay un cambio: Cambio el valor de la clave foránea en Clientes.

Roles

- **Problema**
 - Una misma dimension puede tener varios roles al usarse en una fact table.
- **Ejemplo**

Diagram illustrating a fact table with multiple roles for the same dimension:

- Fact Table Attributes: Fecha orden, Fecha empacado, Fecha envio, Fecha pago, Cliente, Producto, Almacen, Cant Unidades, Precio.
- Dimensions and Roles:
 - Dimension Fecha: Connected to Fecha orden, Fecha empacado, Fecha envio, and Fecha pago.
 - Dimension Cliente: Connected to Cliente.
 - Dimension Producto: Connected to Producto.
 - Dimension Almacen: Connected to Almacen.

Facultad de Ingeniería - In.Co. Sistemas de Data Warehousing - 2003 15

Roles

- **Solucion**
 - No tendremos varias tablas físicas para esa dimension
 - 2 opciones
 - Usar alias en la consulta SQL
 - Usar Vistas
 - Nos permiten darle nombres distintos a los atributos de las distintas vistas

Facultad de Ingeniería - In.Co. Sistemas de Data Warehousing - 2003 16

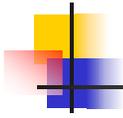
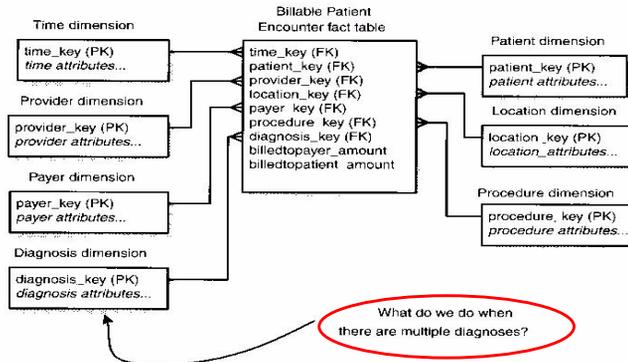


Dimensiones "Many-to-Many"

■ Problema

- En algunos casos la fact table tiene una relacion N-N con una de sus dimensiones

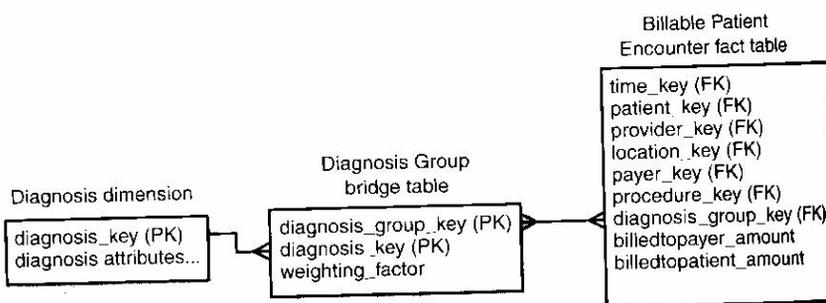
■ Ejemplo



Dimensiones "Many-to-Many"

■ Solucion

- Agregar una "bridge table" entre la fact table y la dimension

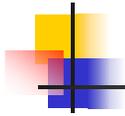




Dimensiones "Many-to-Many"

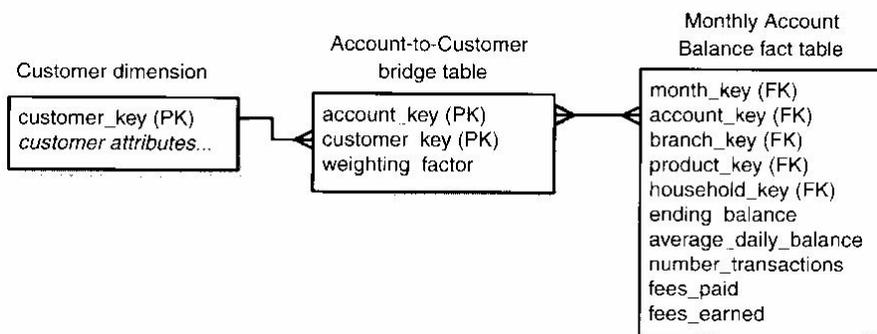
■ Implica

- Generalizar la clave en la fact table:
diagnosis_key → diagnosis_group_key
- Bridge table contiene:
 - Clave del grupo
 - Clave de la dimension (la original)
 - Peso (Weighting factor)
- Como se agrupa por esa dimension?
 - Se hace join con la bridge table, y se multiplica la medida por el peso antes de sumar.
- Que sucede si no hay pesos?



Dimensiones "Many-to-Many"

■ Otro ejemplo

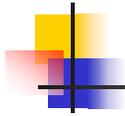




Dimensiones "Many-to-Many"

- **Artículo [Son01]:**

An Analysis of Many-to-Many Relationships Between Fact and Dimension Tables in Dimensional Modeling.
Song, Rowen, Medsker, Ewen. DMDW 2001.



Fragmentación tabla de hechos

- **Artículo [Gol00]**

"Applying Vertical Fragmentation Techniques in Logical Design of Multidimensional Databases". M. Golfarelli, D. Maio, S. Rizzi. DAWAK 2000.

- **Materialización de agregaciones**

- Dado *workload*, para mejorar performance.
- Totalizaciones de medidas segun *patrones de agregaciones*

- **Fragmentación**

- Particionar las medidas de una *vista* en varias tablas
- Unificar varias vistas en una sola tabla.



Fragmentación tabla de hechos

■ Notación/ Definiciones

- Cubo f , $\text{Pat}(f)$, $\text{Med}(f)$, $\text{Atr}(f)$, $\text{Pat}(f) \subset \text{Atr}(f)$
- P_i y P_j 2 patrones. $P_i \geq P_j$ si $\forall a_h \in P_i (a_h \in P_j) \vee (\exists a_k \in P_j / a_k \rightarrow a_h)$. P_i mas resumido que P_j .
- *workload*. Conjunto de pares (q_i, f_i) , consulta y frecuencia.
 - q_i se caracteriza por $\text{Pat}(q_i)$, $\text{Med}(q_i)$, $\text{sel}(q_i)$



Fragmentación tabla de hechos

■ Problema de la fragmentación

- Considera
 - $\text{Med}(q_i)$, $\text{Med}(\text{particiones})$
 - $\text{Pat}(q_i)$, $\text{Pat}(\text{particiones})$
- Factores a tener en cuenta
 - Cantidad de medidas a leer
 - Cantidad de tuplas a acceder
 - Numero de atributos de la clave de la tabla de hechos



Fragmentación tabla de hechos

■ Ejemplo

TABLA HECHOS:

LinItem (PartId, SupplierId, OrderId, ShipDateId, Price, Qty, ExtPrice, Discount, DiscPrice, SumCharge, Tax)

VISTAS v1 y v2:

Med(v1) = Med(LinItem)

Med(v2) = Med(LinItem)

Pat(v1) = {SNation, Brand}

Pat(v2) = {SNation, Part, ODate}



Fragmentación tabla de hechos

■ Ejemplo – cont.

PARTICIONES v1', v1'', v2', v2'':

Med(v1') = {Price, Qty, Discount, ExtPrice, DiscPrice}

Med(v1'') = {Tax, SumCharge}

Med(v2') = {DiscPrice}

Med(v2'') = {Price, Qty, Discount, Ext Price, Tax, SumCharge}

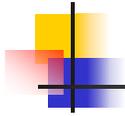
CONSULTAS q1 y q2:

Med(q1) = {Price, Qty, Discount, ExtPrice, DiscPrice}

Med(q2) = {Tax, DiscPrice, SumCharge}

Pat(q1) = {SNation, Brand}

Pat(q2) = {SNation, Brand}



Conclusiones

■ Refinamientos

- Se pueden aplicar como paso siguiente a alguna metodología de diseño de esquema dimensional relacional básico
- Vimos algunos problemas frecuentes y propuestas de diseño para resolverlos
 - En base a ejemplos (Kimball)
 - Focalizando problemas muy específicos (artículos)
- Todavía no existe una teoría formal que sustente técnicas de diseño lógico relacional de DWs