

## Objetivo de esta presentación

- **Contexto**
  - Tomar el trabajo de la tesis de Verónica como punto de partida para proponer un mecanismo de carga y actualización en DWD.
- **Objetivo**
  - Re analizar (sin stress de "defensa de tesis") lo hecho.
  - Identificar puntos a profundizar en vista al mecanismo que queremos definir.

InCo - Facultad de Ingeniería

1

## Diseño Lógico de RDW desde Conceptual MD

- **Tesis de maestría [Peralta, 2001]**
  - Continuación de
    - » Tesis de maestría [Carpani, 2000]: CMDM
    - » Tesis de maestría [Marotta, 1999]: Primitivas de diseño lóg. RDW
- **Def. problema**
  - Generación de esquemas lógicos para DW relacionales a partir de:
    - » esquemas conceptuales de DW en CMDM, y
    - » esquemas lógicos integrados de las bds fuentes

InCo - Facultad de Ingeniería

2

## Enfoque de la solución

- **Un algoritmo**
  - Entrada
    - Esquema conceptual del DW en CMDM refinado y con lineamientos de diseño (esquema intermedio)
    - Esquema lógico de la bdf
    - Correspondencia entre el esquema intermedio y la bdf
  - Salida
    - Esquema lógico relacional del DW
    - Correspondencia (traza) entre el esquema del DW y el esquema de la bdf.
- **El algoritmo se apoya en reglas de diseño**

InCo - Facultad de Ingeniería

3

## Primera Parte - Parámetros de entrada

- **Refinamiento del esquema conceptual CMDM**
- **Lineamientos de diseño**
- **Esquema intermedio**
- **Especificación del esquema lógico de la bdf**
- **Correspondencia entre EI - BDF**

InCo - Facultad de Ingeniería

4

## Refinamiento del esquema conceptual MD

- **Instanciar parámetros de CMDM**
  - Niveles compuestos por atributos de tipos simples ("niveles de tipo tupla").
    - » A estos atributos se les llama *items*.
- **Agregar restricciones estructurales a CMDM**
  - Unidad de items y niveles dentro del esquema MD
  - Claves para todos los niveles
    - » absolutas
    - » relativas

InCo - Facultad de Ingeniería

5

## Lineamientos de diseño (pag. 24)

- **Se tratan de decisiones que el diseñador debe especificar**
  - El diseñador se guía por sus objetivos de performance de consultas y límites de espacio.
  - No se provee ayuda para tomar estas decisiones (trabajo futuro)
- **Sobre las dimensiones**
  - Fragmentación vertical
    - » niveles a almacenar juntos
- **Sobre las relaciones dimensionales**
  - Materialización de relaciones dimensionales
    - » cubos a materializar
  - Fragmentación horizontal de cubos
    - » "franjas" definidas por predicados sobre los items de los niveles del cubo.

InCo - Facultad de Ingeniería

6

## Esquema intermedio - EI

- **Consiste de**
  - Items (SCHITEMS)
  - Niveles (SCHLEVELS)
  - Dimensiones (SCHDIMENSIONS)
  - Cubos (SCHCUBES)
  - Fragmentación de dimensiones (SCHDFRAGMENTATION)
  - Fragmentación de cubos (SCHCFRAGMENTATION)
- **Nota**

Los niveles, dimensiones, cubos y el esquema MD pueden tener restricciones.

InCo - Facultad de Ingeniería

7

## Especificación de la BDF (pag. 31)

- **BDF = bd relacional integrada [y consistente]**
- **Elementos de la especificación**
  - tablas y su clave primaria (SCHTABLES)
  - atributos con sus tipos (SCHATTRIBUTES)
  - links (SCHLINKS)
- **Link**
  - especifica la forma de vincular dos tablas mediante un join
  - se introduce con el objetivo de automatizar la vinculación entre las tablas fuentes dentro del algoritmo de generación.
- **El grafo de tablas y links debe ser: conexo, acíclico y simple**

InCo - Facultad de Ingeniería

8

## Correspondencia EI – BDF

- **En su forma más básica, se expresa mediante**
  - *funciones de correspondencias de items, y restricciones sobre la bdf*
- **Una correspondencia por cada:**
  - Fragmento de dimensión
  - Cubo
    - » correspondencia base
    - » correspondencia recursiva

InCo - Facultad de Ingeniería

9

## Función de correspondencia

- Una función de correspondencia hace corresponder una *expresión de correspondencia* a cada *item*, controlando que coincidan sus tipos.

{ (item, expresión de correspondencia) }
- Una función de correspondencia "representa" una consulta SQL (pag. 34).
  - En "select": expresiones de correspondencia
  - En "from": tablas referenciadas en las expresiones de corresp.
  - En "where": condiciones derivadas de los links

{ (item, expresión de correspondencia) }  Consulta SQL  
"se entiende"  
("semántica")

InCo - Facultad de Ingeniería

10

## Expresiones de correspondencia (pag. 32)

- **Las expresiones de correspondencia se definen sobre los elementos de la BDF**
  - En particular sobre SCHTABLES
- **4 clases de expresiones**
  - **Extern**
    - » constante, estampilla de tiempo o dígitos de versión
  - **Direct**
    - » atributo de una tabla fuente
  - **1cale**
    - » cálculo involucrando varios atributos de UNA tupla
  - **Ncale**
    - » resumen involucrando varios atributos de VARIAS tuplas

InCo - Facultad de Ingeniería

11

## Restricciones sobre la BDF (pag. 35)

- **Condiciones sobre los atributos de las tablas fuentes**
  - se trata de un pasaje de las restricciones del modelo conceptual
  - el diseñador las define "a mano"
- **Trabajo futuro**
  - Automatizar el pasaje

InCo - Facultad de Ingeniería

12

### "Semántica" de una correspondencia

- El *esquema relacional y la instancia asociada a una correspondencia* "puede pensarse como" una vista (consulta) SQL (deducido de comentarios pag. 34)
  - Consulta SQL derivada de la función de correspondencia
  - Extendiendo (mediante "and") el "where" con:
    - » Condiciones correspondientes a las restricciones sobre la bdf

```
select      expresiones de correspondencia
from        tablas en expresiones de corresp.
where       condiciones de join derivados de link
and        condiciones sobre la bdf
```

InCo - Facultad de Ingeniería

13

### Correspondencia de fragmentos de dimensión

- Para cada fragmento de dimensión:
  - una función de correspondencia de sus items
    - » puede incluir items de niveles superiores en el caso de que sus niveles raíz tengan clave relativa
  - (si corresponde), restricciones sobre la bdf

InCo - Facultad de Ingeniería

14

### Fragm. dimension / Ejemplo

- Figura 18, pag. 35
- Consulta SQL (derivado de comentarios pag. 37 y 38)

```
select "Uru" ++ D.zona as zona, "Uruguay" as Pais, D.iddepto as IdDept,
       D.nomdepto as Depto, C.idciudad as IdCiudad, C.nomciudad as Ciudad
  from Departamentos D, Ciudades C
 where D.iddepto = C.iddepto
   and  D.zona < 10
   and  C.clasificacion = "R"
```
- Observación:
  - eliminación de los atributos población y clasificación

InCo - Facultad de Ingeniería

15

### Correspondencia base de cubos

- Dado un cubo, se define mediante
  - una función de correspondencia de los items que identifican a los niveles
  - (si corresponde), restricciones sobre la bdf
  - un conjunto de funciones de rollup sobre los items de la medida.

InCo - Facultad de Ingeniería

16

### Cubo base /Ejemplo

- Figura 20, pag. 39
- Consulta SQL (usando transp. 13)

```
select month(F.Fecha) as mes, RF.Articulo as articulo, F.cliente as cliente,
       F.vendedor as vendedor, ???
  from Facturas F, Registros-Facturas RF
 where F. Factura = RF.Factura
      ???
```
- Duda:
  - las restricciones sobre las dimensiones no pueden afectar las instancias que se quieren de la "tabla de hechos" ?
    - » e.g. problemas de FK ?

InCo - Facultad de Ingeniería

17

### Correspondencia recursiva de cubos

- (pendiente para una segunda pasada)

InCo - Facultad de Ingeniería

18

## Observaciones y Preguntas

- Con los elementos definidos hasta aqui, hay dos caminos
  1. generar directamente las consultas sql (luego, de completar la "sem")
  2. generar secuencia de primitivas de diseño de RDW (cap. 4 de la tesis de Veronika).
- Motivación por el camino (2)
  - Mostrar que el método propuesto automatiza el uso de las primitivas cuando los req. Olap están bien definidos.
  - Facilitar el algoritmo de generación (refinamiento sucesivo).
  - Evolución de esquema como valor agregado.
    - » Qué tanto dependen las técnicas usadas de las primitivas ?.
    - » El "grafo de atributos" es el principal elemento de las técnicas ?.
    - » Dicho grafo puede ser generado a partir de las consultas sql ?

InCo - Facultad de Ingeniería

19

## Segunda Parte - Algoritmo

- Algoritmo de generación del esquema lógico del RDW
- Reglas de diseño

InCo - Facultad de Ingeniería

20

## Algoritmo

- 2 partes:
  - Parte 1: Construcción de las tablas de dimensión
  - Parte 2: Construcción de las tablas de hecho
    - » Construcción de las tablas de hecho para cubos base
    - » Idem cubos recursivos
    - » Construcción de tablas de hecho para franjas de cubos

InCo - Facultad de Ingeniería

21

## Algoritmo / Tablas de dimensión

Para cada fragmento, se aplican los pasos: (regla / primitiva / instancia)

1. Construir esqueleto E (def.)  
Join / Relation join / join
2. Renombrar atributos para items con Direct  
Rename / Attribute renaming / "identidad"
3. Generar atributos para items con Extern, 1calc y Ncalc  
Detalle del paso
4. Aplicar filtros  
Filter / Instance Filtering /  $\sigma$  (E\*)
5. Eliminar atributos sin correspondencia  
Fragment Group / Aggregate Generation /  $\xi_{AS, \Phi}$  (E\*)
6. Ajustar las claves  
Primary Key / Primary Key Modification / "identidad"

InCo - Facultad de Ingeniería

Algoritmo / Tablas de hecho

22

## Algoritmo / Esqueleto

- Esqueleto
  - Tabla E resultante del join de tablas fuentes usando los links tal que la función de correspondencia del fragmento contenga todas sus expresiones de correspondencia de tipo Direct y 1calc sólo sobre atributos de E.

Volver al algo.

InCo - Facultad de Ingeniería

23

## Algoritmo / Paso 3 (item con Extern, 1calc, Ncalc)

3. Generar atributos para items con Extern, 1calc y Ncalc
  - Extern  
ConstantExternValue / Attribute Adding/  $\pi$  con null
  - TimestampE..V.. / Temporalization /  $\pi$  con función de tiempo
  - VersionE...V... / Version Digits /  $\pi$  con attr. calculado
- 1calc  
SimpleCalculate / DDAAdding1-1 /  $\pi$  con atributo calculado
- Ncalc  
AggregateCalculate / DDAAddingN-N /  $\xi_{E,A1, ..., E.An, agg(R.Bs))}(E^*R)$

Volver al algo.

InCo - Facultad de Ingeniería

24

## Algoritmo / Tablas de dimensión (1)

- Interesante
  - analizar la "mezcla" del paso 3 y el paso 5 ([volver al algo.](#))

## Algoritmo / Tablas de hecho para cubos base

Para cada cubo, se aplican los siguientes pasos

1. Construir esqueleto E
  - Idem dimensión
2. Renombrar atributos para ítems con Direct
  - Idem dimensión
3. Generar atributos para ítems con Extern, 1calc y Nealc
  - Idem dimensión
4. Aplicar filtros
  - Idem dimensión
5. Eliminar atributos sin correspondencia
  - CubeGroup / Aggregate Generation /  $\xi_{As, Rup}(E^+)$
6. Ajustar las claves
  - Idem dimensión

## Algoritmo / Tablas de hecho para cubos base (1)

- Más interesante
  - estudiar "mezcla" de los pasos 3 y 5.

## Algoritmo / Tablas de hecho para cubos recursivos

- Pendiente para una segunda pasada

## Reglas de diseño

- Objetivo
  - Determinar cuando puede aplicarse una determinada primitiva de diseño
- Las reglas utilizan y modifican tablas relacionales y las funciones de correspondencia asociadas.