Tercera Parte - Reflexiones sobre el estado actual

- "Semántica" de las correspondencias (15 T → 2 T)
- Resumen/Conclusion (4 T)
- Resultado de esta serie de presentaciones (6 T)

"Semántica" de una correspondencia

- El *esquema relacional y la instancia asociada a una correspondencia* "debe pensarse como una vista SQL".
  - "La instancia de la vista debe coincidir con la instancia esperada para el fragmento/cubo".
  - Consulta SQL derivada de la función de correspondencia
  - Extension (mediante "and") del "where" con condiciones correspondientes a las restricciones sobre la bdf
- Para el caso de fragm. se define en la tesis de Vero (pag. 34):

select	*expresiones de correspondencia*
from	*tablas en expresiones de corresp.*
where	*condiciones de join derivadas de links*
and	*condiciones sobre la bdf*

Resumiendo

- Qué estamos haciendo ?
  - Buscando la forma interpretar/pensar/verificar/... lo que estoy "dibujando" con las correspondencias
    - » Una sola consulta SQL ?
  - Mi conclusión: la "semantica" no está completa o la está dando el algoritmo.
- Hay una forma más clara de dar la semántica sin usar sentencias sql ?
  - En qué nos puede ayudar ?
- Si las respuestas al punto anterior son: "no" y "en nada" o "nos estamos complicando la vida" o "la semantica coincide con el método" entonces
  - disculpen el tiempo (3 min.) y olvidemos la pregunta anterior.

Correspondencia de fragm. de dim. (revised)

- 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
- Dudas:
  1. Es equivalente lo que genera el algoritmo con respecto a la semantica cuando hay atributos en las tablas fuentes involucradas que no intervienen en la función de correspondencia de items ?.
     - » atributos no primos
     - » atributos primos
  2. Semantica cuando se tiene expresiones Ncalc
  3. Interrelacion entre 1 y 2

Fragm. dim. /"Semántica" de una corresp. (revised)

- Sin NCalc
- Semantica ↔ Algoritmo: estudiar la secuencia

  E = Relation join

  E1 = (Attribute Adding | Temporalization | Version Digits | | DD-Adding1-1)*

  E2 = InstanceFilter (E1, ...)

  E3 = Aggregate Generation (E2, ...)
- Ejemplo:

  E1 = DD-Adding1-1 (E, ...)

  E2 = InstanceFilter (E1, ...)

  E3 = Aggregate Generation (E2, ...)
  - En SQL, estaríamos hablando de:

Fragm. dimension / Secuencia de SQL

- DD Adding 1-1

E1 =	select	att(E), f(A ⊆ att(E))
	from	E

- Instance Filter

E2 =	select	*
	from	E1
	where	condiciones que pueden o no involucrar atributos que estan siendo correspondidos

- Aggregate Generation

E3 =	select	att(E2) – X /* sin agg */
	from	E2
	group by	att(E2) – X

Fragm. dimension / Ejemplo 1 (revised)

- **Se eliminan atributos no primos y no hay Ncalc**
- **Aplicación del algo a la Figura 18, pag. 35**
  - **Detalle:** *"Uruguay" as Pais* **no sería generado del attribute adding**
  - **Problema en el paso 4**
- **Consulta SQL a mano**

```
select "Uru" ++ D.zona as zona, "Uruguay" as pais, D.iddepto as iddepto,
       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"
group by D.iddepto, D.nomdepto, C.idciudad, C.nomciudad, "Uru"++D.zona,
       "Uruguay"
```


Algoritmo / Tablas de dimensión (revised)

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

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

Fragm. dimension / Ejemplo 2

- **Se eliminan atributos primos y no hay Ncalc**
- **Semantica ↔ Algoritmo**
  - **Ver hojita**
  - **Semantica**
  - **Algoritmo**
    - » **el group by ?**

Fragm. dimension / Ejercicios pendientes

- **Ejemplo 3**
  - **Un ejemplo con Ncalc y sin excluir atributos**
  - **Consulta SQL**
- **Ejemplo 4**
  - **Un ejemplo con Ncalc y excluyendo atributos no primos**
  - **Consulta SQL**
- **Ejemplo 5**
  - **Un ejemplo con Ncalc y excluyendo atributos primos**
  - **Consulta SQL**

Fragm. dim. / "Semántica" de una correspondencia (revised)

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

select	**t(*expresiones de correspondencia)* **
from	***tablas en expresiones de corresp.***
where	***condiciones de join derivados de link***
and	***condiciones sobre la bdf***
group by	***atributos que participan en Direct y1Calc ?***

"Semántica" de una correspondencia (revised)

- **Con Ncalc**
- **Semantica ↔ Algoritmo**
  - **T1 = DD-AddingN-N (σE, R, ...)**
  - **T2 = Aggregate Generation (T1, ...)**
- **En SQL, estaríamos hablando de:**

Fragm. dimension / Semantica ↔ Algoritmo

- **DD Adding N-N**

```
T1 =    select      attr(σE), agg(A ⊆ attr(R))
        from        σE, R
        where       condicion-join(σE, R)
        group by    attr(σE)
```

- **Aggregate Generation**

```
T3 =    select      attr(T1) - X /* sin agg !!!! */
        from        T1
        group by    attr(T1) - X
```


Correspondencia base de cubos (revised)

- **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.**

Cubo base /Ejemplo (1)

- **Figura 20, pag. 39**
- **"Semantica" ?**

```
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 ?**

Cubo base / Secuencia de SQL

- **Semantica ↔ Algoritmo**

DD Adding N-N

```
T1 =    select      attr(σE), agg (A ⊆ attr(R))
        from        σE, R
        where       condicion-join(σE, R)
        group by    attr(σE)
```

Aggregate Generation

```
T3 =    select      attr(T1) - (X ∪ M),rup (M)
        from        T1
        group by    attr(T1) - (X ∪ M)
```


Cubo base / Ejercicio pendiente

- **Con Ncalc**
- **Consulta SQL**

Resumiendo (repetición)

- **Qué estamos haciendo ?**
  - **Buscando la forma interpretar/pensar/verificar/... lo que estoy "dibujando" con las correspondencias**
    - » **Una sola consulta SQL ?**
  - **Mi conclusión: la "semantica" no está completa o la está dando el algoritmo.**
- **Hay una forma más clara de dar la semántica sin usar sentencias sql ?**
  - **En qué nos puede ayudar ?**
- **Si las respuestas al punto anterior son: "no" y "en nada" o "nos estamos complicando la vida" o "la semantica coincide con el método" entonces**
  - **disculpen el tiempo (3 min.) y olvidemos la pregunta anterior.**

Estado actual

- **A mi entender este es el estado del trabajo realizado en csi sobre diseño de dw relacionales.**

Casos de estudio

- **Anexo 7 tesis Veronika**
  - **Se aplica el algoritmo (no hay expresiones Ncalc)**
- **Ejercicio "Arte Espectacular" del curso Sist DW**
  - **Se modela con CMDM**
  - **Pasaje directo a esquema estrella a mano**
  - **Aplicación a mano de primitivas para pasar de BDF a los esquemas estrella.**
  - **Diseño a mano de la carga inicial obtenida mediante:**
    - » **Secuencias de consultas SQL obtenidas directamente de la def. de las primitivas**
    - » **Simplificacion intuitiva de las secuencias**
  - **Arquitectura del sist. de DW utilizado (comentar)**

Casos de estudio (1)

- **Sistema gerencial de apoyo a la enseñanza en la Fing - UdelaR**
  - **en elaboración**
- **Sistema de DW para el INAPE**
  - **en elaboración**
- **Hasta ahora casos de estudio sin implementacion**
  - **A mi conocimiento, el más cercano "Arte Espectacular"**

Objetivo de esta presentación (repetición)

- **Contexto**
  - **Tomar el trabajo de la tesis de Verónika 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.**

Resultado

- **2 grandes preguntas**
  - **qué enfoque seguir para definir un mecanismo de carga y actualizacion ?**
  - **cómo ubicamos el trabajo hecho en un sistema de dw ?**

Carga y Actualización de datos

- **Partir de la "semantica" de las correspondencias o de la traza generada por el algoritmo ?**
- **Con el enfoque de la tesis de Vero, hay dos caminos:**
  1. **generar directamente las consultas sql,**
  2. **generar secuencia de primitivas de diseño de RDW (cap. 4 de la tesis de Veronika).**
- **Camino 1 necesita:**
  - **Completar la "semantica"**
  - **Analizarsi el proceso de carga son las consultas sql directamente (tarea abierta).**
    - » **Se está en las mismas condiciones que los trabajos sobre mantenimiento de vistas materializadas ?.**

Carga y Actualización de datos (1)

- **Camino 2 necesita**
  - **Decidir que es la "semantica"**
  - **Si la semántica NO es el algoritmo**
    - » **Completar la semantica**
    - » **Mostrar que el algoritmo produce un resultado equivalente**
  - **Estudiar el paso de filtrado del algoritmo.**
    2 subcaminos
    1. Modificar las restricciones a cada paso como se hace con las corresp.
    2. Estudiar modificación de orden del paso de filtrado en el algo. actual.
  - **Transformación de secuencia de consultas/vistas.**
  - **(idem camino 1) Analizar si el proceso de carga son las consultas sql directamente (tarea abierta).**
    - » **Se está en las mismas condiciones que los trabajos sobre mantenimiento de vistas materializadas ?.**

Carga y Actualización de datos (2)

- **Motivación por el camino 2**
  - **Mostrar que el método propuesto automatiza el uso de las primitivas cuando los req. Olap están bien definidos.**
  - **Facilitar la definición del algoritmo de generación (técnica de 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 ?**

Carga y Actualización de datos (3)

- **Carga inicial y actualización de datos ?**
  - **Recuperación en carga inicial ?**
  - **Paralelización ?**
  - **Técnicas de materialización de vistas ?**
- **Controles de consistencia en el RDW**
  - **Manejo de errores**
- **Actualización y Consistencia BDF y RDW**
  - **Técnicas de materialización de vistas ?**
  - **Algo. Strobe ?**
- **Qué ofrecen los "grandes" fabricantes de DBMS ?**
  - **Oracle, IBM/Informix, Microsoft, . . .**

Aplicación en Sist. de DW

- **Contexto de aplicación de lo hecho**
  - **Este es un buen momento para/nos interesa hacer esta pregunta ?.**
  - **Comportamiento en casos de estudio ?**
  - **Pregunta: Cuál es la arquitectura de Sist de DW que se beneficiaría con esta propuesta ?**
    - » **Consultas directas al RDW resultado del algoritmo ? Rolap?**
  - **Qué tan "comun" es una arquitectura donde la carga de la porción del DW -> Olap sea siempre completa ?**
  - **Complemento a la pregunta: Cómo aprovecho lo diseñado en una arquitectura concreta ?**
    - » **Tengo el RDW resultado del algoritmo, cómo exploto los fragmentos, los cubos materializados ?.**
    - » **Consultas en un RDBMS+ ?**
    - » **Molap ?**

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**

Algoritmo / Tablas de dimensión

Para cada framento, 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 / σ (E^+)
5. **Eliminar atributos sin correspondencia**
   Fragment Group / Aggregate Generation / $\xi_{att(E+)\text{-}X, \Phi}$ (E^+)
6. **Ajustar las claves**
   Primary Key / Primary Key Modification / "identidad"

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.

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

3. Generar atributos para items con Extern, 1calc y Ncalc

Extern

ConstantExternValue / Attribute Adding/ π^+ con null

TimestampE..V.. / Temporalization / π^+ con función de tiempo

VersionE...V... / Version Digits / π^+ con attr. calculado

1calc

SimpleCalculate / DDAdding1-1 / π^+ con atributo calculado

Ncalc

AggregateCalculate / DDAddingN-N / $\xi_{att(E),\ agg(a \in att(R))}(E*R)$

Volver al algo.

Algoritmo / Tablas de dimensión (1)

- **Interesante**
  - **analizar la "mezcla" del paso 3 y el paso 5 (<u>volver al algo.</u>)**

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 items con Direct

Idem dimensión

3. Generar atributos para items con Extern, 1calc y Ncalc

Idem dimension

4. Aplicar filtros

Idem dimensión

5. Eliminar atributos sin correspondencia

CubeGroup / Aggregate Generation / $\xi_{att(E+)-X,\ Rup}(E^+)$

6. Ajustar las claves

Idem dimensión

Algoritmo / Tablas de dimensión

Algoritmo / Tablas de hecho para cubos base (1)

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

Restricciones sobre la BDF (revised) (pag. 35)

- ***Condiciones sobre los atributos de las tablas fuentes***
  - **pueden deberse a**
    - » ***las restricciones del modelo conceptual***
    - » ***restricciones que se deseen al hacer la correspondencia***
  - **el diseñador las define "*a mano*"**
- **Trabajo futuro**

  Automatizar el pasaje

  Observación (a tener en cuenta en la automatización ?):
  - **Al obligar a esta altura el pasaje de las restricciones del conceptual a restricciones sobre las fuentes, no permite expresar restricciones sobre items que tienen asociada una expresion de corresp.de tipo Ncalc**
    - » **en su forma más simple serían predicados para el "having" ?**

Observaciones finales

- **Primitivas que se agregan**
  - **Relation join !!!!!! ("flexible" pero poco conceptual desde el punto de vista del espiritu inicial de las primitivas)**
    - » **(a verificar) primitivas que pueden escribirse en función de ella:**
      - **hierarchy generation**
      - **minidimension**
      - **new dimension crossing**
  - **Attribute renaming**
  - **Primary key modification**
  - **Instance filter !!!**
    - » **trabaja puramente con instancias**
- **Consecuencias en la técnica de evolución de esquema ?**

Observaciones finales (1)

- **Qué primitivas NO se utilizan en la presente propuesta ?**

Afinamiento

1. Mini dimension

Para mejoras en el tratamiento de cambios y consultas sobre dimensiones muy grandes

2. Partition by stability

Para organizar datos históricos

Datos en metadatos

3. Data array creation

Generación de tablas de jerarquias

4. Hierarchy generation

???

5. New dimension crossing