Capítulo 5

Dependencias Funcionales

Parte 2

Hallando las dependencias funcionales candidatas

- **Problema:** tenemos un esquema universal *R* y un conjunto de DF *F*. **¿Cómo hallar DFs candidatas?**
- **Algunas heurísticas:**
  1. **Uso de conceptos**: Identificar a partir de *R* un *concepto*;
     - buscar atributos *S* del concepto (subconjunto de R);
     - identificar subconjunto de atributos de *S* que determina los demás atributos de *S* (identificador del concepto)*; esto da una DF* α -> β.
     - Se parece a hallar las diferentes claves de un conjunto de entidades o concepto.
  2. **Relaciones entre conceptos**: Identificar a partir de *R* una relación entre atributos vinculando conceptos; esa relación tiene atributos S; identificar subconjunto de atributos de *S* que determina los demás atributos de *S; esto da una DF* α -> β.

Hallando las dependencias funcionales candidatas

- **Algunas heurísticas - cont:**
  3. Relaciones uno-varios o varios – uno entre conceptos
  4. Relaciones uno-uno entre conceptos
  5. **Reducir el lado izquierdo de una DF**: α -> β candidata a DF; si α – {A} -> β sigue siendo candidata, entonces A es redundante en α.
  6. **Redundancia de información como indicio de DF**: Si conjunto de atributos β en *R* generan redundancia de información: encontrar candidata a DF α -> β (α no determina *R*).
  7. **Atributos Huérfanos sin DF**: Si un atributo *A* de *R* no fue cubierto por ninguna DF,
     - pensar por qué está ese atributo en *R*;
     - o sea, ver si forma parte de un concepto o de relación entre conceptos.
     - Luego intentar aplicar las una de las dos primeras heurísticas para encontrar DF que involucra a *A*.
  8. **Atributos que pertenecen a mas de un concepto**: Fijarse si un atributo debe pertenecer a otro concepto también y esto da lugar DFs adicionales.
     - **Ejemplo**: nombreBib pertenece a copia de libro y a biblioteca.

Hallando las dependencias funcionales candidatas

- **Algunas heurísticas – cont:**
  9. **Examen de regla de dominio**: ¿Una regla de dominio se puede expresar como una dependencia funcional?
     - **Ejemplo**: en una dirección no puede haber más de dos bibliotecas (calle, número → nombreBib)
  10. **Examen de consulta**: ¿Una consulta tiene implícita una dependencia funcional?
     - **Ejemplo**: encontrar el dueño de un teléfono (TE → DNI)
- Para cada heurística se puede buscar **cubrir todos los casos**.
  - **P.ej:** todos los conceptos, todas las relaciones entre atributos, todas las reglas del dominio, todas las consultas, etc.

Chequeo de la validez de una dependencia funcional candidata

- Se divide el **chequeo de la validez de una DF candidata** en dos partes:
  - **aplicar reglas de descarte**: si se viola una regla de estas, entonces la DF candidata no vale.
    - El chequeo aquí considera una **resistencia a la ruptura** de las reglas de descarte.
  - **Validación positiva**: Viendo **razones válidas** para usar esa DF candidata.

Chequeo de la validez de una dependencia funcional candidata

- **Reglas de descarte:** consideramos los siguientes ejemplos de reglas de descarte:
  - La DF se rompe en uno o más casos reales (documentados o no).
  - La DF no refleja como funciona el sistema hoy (falta de sentido en el dominio).
  - Los usuarios, expertos, clientes del sistema ya no reconocen la DF como válida.
  - La DF contradice reglas de negocio (políticas o prácticas adoptadas)
  - La DF se rompe frecuentemente por cambios en el sistema o en los datos.
  - La DF no puede sostenerse ante evaluadores externos.

Chequeo de la validez de una dependencia funcional candidata

- **Validación positiva**: Consideramos los siguientes ejemplos de **razones de validez**:
  - La DF representa una relación real en el dominio.
  - La DF es recomendada como válida por diferentes roles (usuarios, expertos, clientes, etc.)
  - La DF es útil en el diseño: permite manejar problemas de diseño (como redundancia de información, manejo de nulos).
  - La DF captura reglas de negocio.
  - La DF es estable (se espera que se mantenga en el tiempo).

Reformulación de DF candidatas

- Incluso si una DF fue descartada por cumplir una regla de descarte, muchas veces se **puede rescatar el valor del intento**.
- Una DF incluso fallida puede revelar caminos o intuiciones que merecen ser exploradas.
- Aunque una DF no sea válida, puede sugerir una DF cercana que sí lo sea.
  - **Por ejemplo**: producto → precio, tal vez podría arreglarse con producto, proveedor → precio.
- Podría ser que una DF no sea válida, porque falta algún atributo en el esquema universal que debe ser considerado.
  - **Por ejemplo**: en el esquema universal del ejemplo anterior no se tenía proveedor.

Chequeo de la validez de una dependencia funcional candidata

- A veces, la DF no se valida, pero **va en el sentido correcto** porque revela una intención, una lógica del sistema, una expectativa de los actores.
  - En otras palabras hay razones de validez.
- **Por lo tanto, podemos considerar los siguientes pasos:**
  1. Presentar DF candidata
  2. Dar motivo de descarte
  3. Dar motivo para hacer reformulación
     - **P.ej**: Intuición revelada, cercana a la realidad, atributos faltantes en esquema universal.
  4. Reformulación sugerida
  5. Chequeo de validez de la reformulación

Pulido del conjunto F del problema del mundo real

- Una vez que hallamos un conjunto de DF *F* para un problema del mundo real, puede venir bien **pulir el conjunto** para hacer **más eficiente su chequeo**.
- Damos algunas **heurísticas** para hacerlo:
  - unir DFs que pueden expresarse como una sola más potente.
    - **Ejemplo**: si tenemos A → B y A → C, puede consolidarse como A → B, C.
  - asegurar que los lados izquierdos de las DF no tengan atributos innecesarios.
    - **Ejemplo**: si A, B → C pero A → C ya es válida, entonces B es redundante
- También puede venir bien organizar *F* en **subconjuntos temáticos** (por área, por actor, por tipo de relación).

Hallando las dependencias funcionales de un problema del mundo real

- **Ejercicio**: Sea el esquema relacional:
  - SmaAutomotor = (DNI, nombre, marca, modelo,
    patente, numSeguro, compañíaSeguro, direcciónCS)
  - Algunas reglas del dominio:
    - Un seguro cubre un solo vehículo
    - Los números de seguro se pueden repetir entre diferentes compañías
    - Cada vehículo tiene un único titular
    - Cada vehículo tiene un único seguro vigente
    - Una compañía de seguros tiene una única dirección registrada (la de la casa matriz)
  - Encontrar un conjunto de DF adecuado para el problema.

Algunas observaciones

- **Agregar una nueva dependencia funcional puede llevar a**:
  - **redefinir tablas, modificar claves… e incluso eliminar tuplas que contradicen la nueva DF**.
  - Pero esto no es solo un problema técnico: es una señal de que el sistema está **madurando**, y que la estructura debe acompañar esa revelación.
  - A veces, la DF no existía porque el sistema no la necesitaba… pero ahora **el negocio cambió, el rol evolucionó, una práctica se ritualizó**.
    - Entonces, **la estructura debe acompañar esa legitimación**.
    - Y si hay datos que contradicen esa nueva legitimidad, **pueden ser vistos como ruido, error o vestigio de un ciclo anterior**.

Algunas observaciones

- **Generalizando**: **añadir una restricción de integridad nueva implica revisar los datos existentes, y muchas veces redefinirlos**.
  - Y esto no solo vale para el modelo relacional.
  - Se aplica a otros modelos como basados en documentos, en grafos, etc.
- Por esto, mejor pensar bien las restricciones de integridad desde un comienzo,
  - porque puede ser muy duro arreglar las cosas una vez que se tienen las tablas, o lo que quiera que sea.
  - Al menos arrancar bien.

Algunas observaciones

- **Problema**: **rigidez estructural** frente a **evolución del sistema**.
  - El modelo relacional **normaliza** en función de un conjunto *F* que se supone estable.
  - Pero en sistemas vivos, *F* **evoluciona**:
    - se descartan DF, se reformulan, se consagran nuevas.
  - Esto implica que las **tablas, claves y relaciones** deben rediseñarse, lo que puede:
    - Romper compatibilidad con aplicaciones existentes
    - Requerir migración de datos
    - Generar resistencia institucional

Algunas observaciones

- **Idea de solución**:
  - El modelo relacional **sirve para consagrar lo que ya está claro**,
  - pero que necesita ser acompañado por modelos que **honren lo que aún está en proceso**.
  - Así, la normalización no es un dogma, sino una herramienta que se activa cuando el sistema lo permite.

Otros conceptos

- **Definición**: α -> β se cumple en r(R) si para todo t1 ≠ t2 en r:

$$t1[\alpha] = t2[\alpha] \Rightarrow t1[\beta] = t2[\beta].$$

- **Ejercicio**: Listar algunas DF que no se cumplen y algunas DF que se cumplen en la siguiente tabla:

A	B	C
a1	b1	c1
a2	a1	c1
a1	b2	c2

- No confundir las DF que se cumplen en una tabla con las DF de un problema del mundo real para el cual la tabla es legal.
  - El problema suele cumplir menos DF que la tabla.

Otros conceptos

- Una DF **trivial** se cumple en todas las tablas de un esquema.
  - **Ejemplo***:* En *SmaBibliotecas*
    - *nombreBib, calle* $\rightarrow$ *nombreBib*
    - *calle* $\rightarrow$ *calle*
- Después veremos que las DF triviales juegan su papel en los algoritmos de normalización.
- **Ejercicio** (de la práctica)**:** probar la siguiente
  - **Proposición**: $\alpha \rightarrow \beta$ es **trivial** si y solo si $\beta \subseteq \alpha$.