Contenido:

5 Restricciones de integridad

En este tema se trata uno de los aspectos más importantes para añadir consistencia a los diseños de bases de datos: son las restricciones de integridad que ayudan a mantener la consistencia semántica de los datos. Además de las restricciones de integridad definidas por las claves y las restricciones de cardinalidad y participación estudiadas en el modelo entidad-relación, se tratan las restricciones de los dominios, la integridad referencial, las dependencias funcionales y las dependencias multivaloradas, así como la forma de implementarlas mediante asertos y disparadores.

Las restricciones de integridad proporcionan un medio de asegurar que las modificaciones hechas a la base de datos por los usuarios autorizados no provoquen la pérdida de la consistencia de los datos.
Protegen a la base de datos contra los daños accidentales.

Tipos de restricciones de integridad:

- Declaración de claves (primarias, candidatas).
- Cardinalidad de la relación – de varios a varios, de uno a varios, de uno a uno.
- Participación mín/máx.
- Restricciones de los dominios.
- Integridad referencial.
- Dependencias funcionales.
- Dependencias multivaloradas.

Los asertos y disparadores permiten implementar restricciones de integridad.

5.1 Restricciones de los dominios

Las restricciones de los dominios son la forma más simple de restricción de integridad.
Se especifica para cada atributo un dominio de valores posibles.
Una definición adecuada de las restricciones de los dominios no sólo permite verificar los valores introducidos en la base de datos sino también examinar las consultas para asegurarse de que tengan sentido las comparaciones que hagan.
Por ejemplo, normalmente no se considerará que la consulta “Hallar todos los clientes que tengan el nombre de una sucursal” tenga sentido. Por tanto, *nombre-cliente* y *nombre-sucursal* deben tener dominios diferentes.

La cláusula **check** de SQL:1999 permite restringir los dominios de maneras poderosas que no permiten la mayor parte de los sistemas de tipos de los lenguajes de programación.

La cláusula **check** permite especificar un predicado que debe satisfacer cualquier valor asignado a una variable cuyo tipo sea el dominio. Por ejemplo:

Número de cifras
Número de decimales

create domain *sueldo-por-hora* **numeric**(4,2)

constraint *comprobación-valor-sueldo* **check**(**value** ≥ 6.00)

Restricciones de existencia

Dentro de las restricciones de los dominios, un tipo especial de restricción que se puede aplicar a cualquier dominio es la restricción de existencia.

Esta restricción evita la aparición de valores *nulos* en las columnas.

Restricciones de unicidad

Otro tipo especial de restricción que se puede aplicar a cualquier dominio es la restricción de unicidad.

Esta restricción evita la aparición de valores *duplicados* en las columnas.

Por ejemplo:
Sólo se admite una sucursal en cada ciudad.

```
CREATE TABLE Sucursales
(nombre-sucursal VARCHAR(20),
 ciudad-sucursal VARCHAR(20) NOT NULL,       - Restricción de existencia
PRIMARY KEY(nombre-sucursal)
UNIQUE (ciudad-sucursal))                    - Restricción de unicidad
```

5.2 Integridad referencial

La *integridad referencial* permite asegurar que un valor que aparece en una relación para un conjunto de atributos determinado aparezca también en otra relación para un cierto conjunto de atributos.

5.2.1 Conceptos básicos

Sean un par de relaciones $r(R)$ y $s(S)$ y la reunión natural $r \; |\mathrm{x}| \; s$. Puede haber una tupla t_r de r que no se reúna con ninguna tupla de s.
Es decir, no hay ningún t_s en s tal que $t_r[R \cap S] = t_s[R \cap S]$. Estas tuplas se denominan *colgantes*.
Las tuplas colgantes pueden ser aceptables en función del conjunto de entidades o de relaciones que se esté modelando.
La reunión externa permite operar con relaciones que contengan tuplas colgantes.

Sean $r_1(R_1)$ y $r_2(R_2)$ dos relaciones con las claves primarias K_1 y K_2, respectivamente. Se dice que un subconjunto α de R_2 es una *clave externa* que hace referencia a K_1 de la relación r_1 si se exige que para cada t_2 de r_2 haya una tupla t_1 de r_1 tal que $t_1[K_1] = t_2[\alpha]$. Las exigencias de este tipo se denominan *restricciones de integridad referencial* o *dependencia de subconjunto*.

La denominación *dependencia de subconjunto* se debe a que esta última restricción de integridad referencial puede escribirse como $\Pi_{\alpha}(r_2) \subseteq \Pi_{K_1}(r_1)$. Obsérvese que, para que una restricción de integridad referencial tenga sentido, α debe ser igual a K_1, o bien α y K_1 deben ser conjuntos compatibles de atributos.

5.2.2 Integridad referencial en el modelo E-R

Las restricciones de integridad referencial aparecen con frecuencia. Si se obtiene el esquema de la base de datos relacional creando tablas a partir de los diagramas E-R, cada relación que proceda de un conjunto de relaciones tendrá restricciones de integridad referencial.

Sea un conjunto N-ario de relaciones R, que relaciona los conjuntos de entidades E_1, E_2, ..., E_n. K_i denota la clave primaria de E_i. Los atributos del esquema de la relación del conjunto de relaciones R incluyen $K_1 \cup K_2 \cup ... \cup K_n$. Cada K_i del esquema de R es una clave externa que lleva a una restricción de integridad referencial.

Otra fuente de restricciones de integridad referencial son los conjuntos de entidades débiles. Un conjunto de entidades débiles debe incluir la clave primaria del conjunto de entidades del que éste depende. Por tanto, el esquema de la relación de cada conjunto de entidades débiles incluye una clave externa que lleva a una restricción de integridad referencial.

5.2.3 Modificación de la base de datos

La modificación de la base de datos puede ocasionar violaciones de la integridad referencial.

Para la restricción de integridad referencial $\Pi_{\alpha}(r_2) \subseteq \Pi_K(r_1)$ se describe la comprobación que debe hacerse para cada tipo de modificación de la base de datos:

- Insertar. Si se inserta una tupla t_2 en r_2, el sistema debe asegurar que hay una tupla t_1 de r_1 tal que $t_1[K] = t_2[\alpha]$. Es decir,

$$t_2[\alpha] \in \Pi_K(r_1)$$

- Borrar. Si se borra una tupla t_1 de r_1 el sistema debe calcular el conjunto de tuplas de r_2 que hacen referencia a r_1:

$$\sigma_{\alpha=t_1[K]}(r_2)$$

Si este conjunto no es el conjunto vacío, o bien se rechaza la orden borrar como error, o bien se deben borrar las tuplas que hacen referencia a t_1. La última solución puede llevar a borrados en cascada, dado que las tuplas pueden hacer referencia a tuplas que hagan referencia a t_1, etcétera.

- Actualizar. Hay que considerar dos casos: las actualizaciones de la relación que realiza la referencia (r_2) y las actualizaciones de la relación a la que se hace referencia (r_1).

- Si se actualiza la tupla t_2 de la relación r_2 y esta actualización modifica valores de la clave externa α, se realiza una comprobación parecida al del caso de la inserción. t'_2 denota el nuevo valor de la tupla t_2. El sistema debe asegurar que

$$t'_2[\alpha] \in \Pi_K(r_1)$$

- Si se actualiza la tupla t_1 de la relación r_1 y esta actualización modifica valores de la clave primaria (K), se realiza una comprobación parecida al del caso del borrado. El sistema debe asegurar que

$$\sigma_{\alpha=t_1[K]}(r_2)$$

utilizando el valor anterior de t_1 (el valor antes de que se lleve a cabo la actualización). Si este conjunto no es el conjunto vacío, la actualización se rechaza como error o se ejecuta en cascada de manera parecida al borrado.

5.3 Dependencias funcionales

Una dependencia funcional (DF) es una propiedad semántica de un esquema de relación que presentan las tuplas válidas de la relación que determina para cada valor de un conjunto de atributos X el valor de otro conjunto de atributos Y. Es decir, dada una tupla t_1 de la relación con un valor para X y otro para Y, si aparece otra tupla t_2 con el mismo valor para X, entonces esta tupla debe tener el mismo valor en Y que t_1.

Ejemplo 1.
En la siguiente relación se combinan los datos de los empleados, como su código de identificación y nombre, y de los centros a los que están adscritos, como la dirección y el teléfono.

Empleados_Centros							
Id_empleado	**NombreE**	**DirecciónE**	**Puesto**	**Salario**	**Centro**	**DirecciónC**	**TeléfonoC**
123A	Ana Almansa	c/ Argentales	Profesor	20.000	Informática	c/ Complutense	123
012D	David Díaz	c/ Daroca	Ayudante	10.000	Informática	c/ Complutense	123
789C	Carlos Crespo	c/ Cruz	Catedrático	30.000	Empresariales	c/ Coruña	789

En este ejemplo se muestra gráficamente que el valor del conjunto de campos DirecciónC y TeléfonoC depende del valor del campo Centro. En concreto, a un centro en particular le corresponden unívocamente una dirección y un teléfono. Es decir, cada vez que aparezca una fila con el valor Informática para Centro, siempre le corresponderá los mismos valores para los campos DirecciónC y TeléfonoC.
Se dice entonces que tanto DirecciónC como TeléfonoC son dependientes funcionalmente de Centro. Por cada fila con un mismo valor de Centro se repiten los valores DirecciónC y TeléfonoC, lo que implica una redundancia de valores no deseable que se estudiará más adelante en la normalización de relaciones.
La validez de una relación con respecto a las DF se interpreta desde el significado que el diseñador asocia a la relación. Por tanto, una DF no se puede inferir de una relación, sino que se debe definir explícitamente sobre los atributos de la relación conociendo

perfectamente su semántica. Una DF define los estados consistentes de una relación en función de las dependencias entre los valores de los atributos.
A continuación se proporciona una definición más formal de las dependencias funcionales.

Definición 1. Dependencias funcionales
Sea $R = \{A_1, \ldots, A_n\}$ el esquema universal de la base de datos relacional, es decir, el conjunto de todos los atributos que pueden definirla y r una instancia del esquema R. Una dependencia funcional $X \rightarrow Y$ (los valores de X determinan unívocamente (o funcionalmente) los valores de Y) entre dos conjuntos de atributos X e Y, tales que $X, Y \subseteq R$ especifica la siguiente restricción: $\forall t_1, t_2 \in r$ tales que $t_1[X] = t_2[X]$, entonces $t_1[Y] = t_2[Y]$. X se denomina antecedente e Y consecuente.
En otras palabras, quiere decir que los componentes Y de cada tupla de r están determinados unívocamente por los valores de X.

Observaciones:

- $X \rightarrow Y$ no implica necesariamente $Y \rightarrow X$.

  Ejemplo 2.
  $\{NIF\} \rightarrow \{Nombre\}$.
  Sin embargo, no es cierto $\{Nombre\} \rightarrow \{NIF\}$ puesto que se pueden repetir nombres para diferentes personas. No se debe confiar en general en la dependencia funcional $\{NIF\} \rightarrow \{Nombre\}$ porque en la práctica también hay NIF repetidos. Por ello, en las bases de datos generalmente se usa un identificador propio que identifica unívocamente cada tupla asociada a una persona.
- Una dependencia funcional determina una relación uno a varios entre dos conjuntos de atributos:
  $X \xrightarrow{1:N} Y$, es decir:
  Para un valor de X sólo puede haber un valor de Y, pero para un valor de Y habrá en general varios de X. Por lo tanto, una dependencia funcional se puede observar como una restricción de cardinalidad entre conjuntos de atributos de una *misma* relación.

  Ejemplo 3.

Empleados_Centros							
<u>Id_empleado</u>	**NombreE**	**DirecciónE**	**Puesto**	**Salario**	**Centro**	**DirecciónC**	**TeléfonoC**
123A	Ana Almansa	c/ Argentales	Profesor	20.000	Informática	c/ Complutense	123
012D	David Díaz	c/ Daroca	Ayudante	10.000	Informática	c/ Complutense	123
789C	Carlos Crespo	c/ Cruz	Catedrático	30.000	Empresariales	c/ Coruña	789

$X \longrightarrow Y$

Corolario 1.

- Una restricción de cardinalidad de uno a varios entre dos esquemas de relación R_1 y R_2 con superclaves $X \subseteq R_1$ e $Y \subseteq R_2$ se especifica con la dependencia funcional $X \rightarrow Y$ en una nueva relación R_3.

Ejemplo 4.

Empleados_Centros							
Id_empleado	**NombreE**	**DirecciónE**	**Puesto**	**Salario**	**Centro**	**DirecciónC**	**TeléfonoC**
123A	Ana Almansa	c/ Argentales	Profesor	20.000	Informática	c/ Complutense	123
012D	David Díaz	c/ Daroca	Ayudante	10.000	Informática	c/ Complutense	123
789C	Carlos Crespo	c/ Cruz	Catedrático	30.000	Empresariales	c/ Coruña	789

$X \longrightarrow Y$

- Una restricción de cardinalidad de uno a uno entre dos esquemas de relación R_1 y R_2 con superclaves $X \subseteq R_1$ e $Y \subseteq R_2$ se especifica con las dependencias funcionales $X \to Y$ e $Y \to X$ en un nuevo esquema de relación R_3. No obstante, las relaciones uno a uno se implementan generalmente como un atributo de la relación.
- Una superclave se puede definir en términos de dependencias funcionales. S es superclave del esquema de relación R si $S \to R$ y $S \subseteq R$. Es decir, si $\forall t_1, t_2 \in r$ tales que $t_1[S] = t_2[S]$, entonces $t_1[R] = t_2[R]$, lo cual implica $t_1 = t_2$ porque deben coincidir en todos sus atributos, por lo que se está hablando de la misma tupla.

Ejemplo 5.
Las siguientes son dependencias funcionales de la tabla anterior:

$$\{Id\_empleado\} \to \{NombreE, DirecciónE, Puesto, Salario, Centro, DirecciónC, TeléfonoC\}$$
$$\{Centro\} \to \{DirecciónC, TeléfonoC\}$$
$$\{DirecciónC\} \to \{Centro, TeléfonoC\}$$
$$\{TeléfonoC\} \to \{Centro, DirecciónC\}$$

A partir de la definición de DF se puede definir el concepto de satisfacción de DF.

Definición 2. Satisfacción de dependencias funcionales
Una relación r con esquema R satisface una dependencia funcional $X \to Y$, con $X, Y \subseteq R$, si todas las tuplas de r satisfacen $\forall t_1, t_2 \in r$ tales que $t_1[X] = t_2[X]$, entonces $t_1[Y] = t_2[Y]$.
La comprobación de la satisfacción de dependencias funcionales es necesaria en casos como la migración de datos o la actualización de sistemas heredados.
Bajo esta definición se pueden proponer algoritmos sencillos para comprobar la satisfacción de un conjunto de dependencias funcionales de una relación r, o para comprobar la validez de la inserción de una tupla.

Algoritmos para la comprobación de integridad definida por un conjunto de dependencias funcionales
Algoritmo 1.
a) Algoritmo para comprobar la integridad de una relación con respecto a un conjunto de dependencias funcionales.
Entrada: relación r con un conjunto enumerable de tuplas $t_i \in r, i = 1 \ldots n$ y conjunto enumerable D de dependencias funcionales $d_i = X_i \to Y_i, d_i \in D, i = 1 \ldots m$.

```
for i:=1 to n-1
  for j:=i+1 to n
    for k:=1 to  m
```

```
    if  t_i[X_k] = t_j[X_k]  and  t_i[Y_k] ≠ t_j[Y_k]  then
       Valores inconsistentes de  t_i  y t_j  debido a DF  d_k
```

Algoritmo 2.
b) Algoritmo para comprobar la integridad de la inserción de una tupla en una relación con respecto a un conjunto de dependencias funcionales.
Entrada: relación r con un conjunto enumerable de tuplas $t_i \in r, i = 1 \ldots n$, una tupla t para insertar en r y un conjunto enumerable D de dependencias funcionales $d_i = X_i \to Y_i, d_i \in D, i = 1 \ldots m$.

```
for i:=1 to n
  for j:=1 to  m
    if  t[X_j] = t_i[X_j]  and  t[Y_j] ≠ t_i[Y_j]  then
      Valores inconsistentes de  t_i  y t debido a la DF  d_j
```

Las dependencias funcionales representan restricciones de integridad que el sistema de gestión de bases de datos debe asegurar. Así que, dado un cierto conjunto D de dependencias funcionales, es deseable encontrar otro conjunto E que sea lo menor posible que D de manera que cada $d \in D$ se deduzca de E, con el objetivo de que el coste de mantener la integridad definida en D se reduzca con E. Éste es un objetivo de gran interés práctico al que se dedica el resto del apartado.

Una de las maneras de reducir el coste del aseguramiento de la consistencia mediante dependencias funcionales es eliminar las que no aportan nada semánticamente, es decir, son dependencias funcionales que cumple cualquier tupla.

Definición 3. Dependencias funcionales triviales
Una dependencia funcional $X \to Y$ es trivial si y sólo si $Y \subseteq X$.
Esto sólo dice que si dos tuplas coinciden en una serie de atributos, entonces coinciden (obviamente) en un subconjunto de esos mismos atributos. Se denomina trivial porque no aporta ninguna restricción al esquema de relación.

En general interesará encontrar el conjunto mínimo de dependencias funcionales que sea semánticamente equivalente (asegure el mismo nivel de integridad) a un conjunto dado de dependencias funcionales aportadas por el diseñador de la base de datos.

5.3.1 Cierre de un conjunto de dependencias funcionales

Definición 4. Cierre de un conjunto de dependencias funcionales
El cierre de un conjunto de dependencias funcionales S, denotado S^+, es el conjunto de todas las dependencias definidas intensionalmente por S.

En otras palabras, es el conjunto de todas las dependencias funcionales que se pueden deducir de S. Este concepto es importante para poder determinar la equivalencia semántica de dos conjuntos de dependencias y poder elegir el menor de forma que la comprobación de su satisfacción sea más rápida. Por otra parte, permite razonar sobre la descomposición de relaciones que se estudia en el tema Normalización.

Ejemplo 6.

Es fácil ver que $\{Centro\} \to \{DirecciónC, Teléfono\}$ implica $\{Centro\} \to \{DirecciónC\}$ y $\{Centro\} \to \{Teléfono\}$. Si a un centro le corresponden una dirección y un teléfono determinados, en particular también es cierto que a ese centro le corresponde una dirección, y que a ese centro le corresponde un teléfono.

Notación: Si X e Y son conjuntos de atributos, $XY = X \cup Y$

Para calcular el cierre de un conjunto de dependencias funcionales se dispone de un conjunto de axiomas de producción denominados Axiomas de Armstrong en honor a la persona que los propuso.

1. Reflexividad: Si $Y \subseteq X$, entonces $X \to Y$.

2. Aumentatividad: Si $X \to Y$, entonces $XZ \to YZ$.

3. Transitividad: Si $X \to Y$ e $Y \to Z$, entonces $X \to Z$.

Estos axiomas son correctos en cuanto que derivan información consistente con la definición de dependencia funcional. Además son completos porque permiten deducir todas las consecuencias de un conjunto de dependencias funcionales, es decir, su cierre.

Demostración de la corrección:

1. De la definición de dependencia funcional trivial.
2. Supongamos una relación r que satisfaga $X \to Y$, y dos tuplas de r, t_1 y t_2, que coinciden en XZ pero no coinciden en YZ. Sólo es posible que difieran en X e Y, pero esto contradice $X \to Y$. Por lo tanto, de la definición de dependencia funcional, se debe cumplir $XZ \to YZ$.
3. Por la definición de dependencia funcional: de $X \to Y$ se tiene que $t_1[X] = t_2[X]$ implica $t_1[Y] = t_2[Y]$, y de $Y \to Z$ se tiene que $t_1[Y] = t_2[Y]$ implica $t_1[Z] = t_2[Z]$. Por lo tanto, si se tiene que si $t_1[X] = t_2[X]$ implica $t_1[Z] = t_2[Z]$, entonces, por la definición de dependencia funcional, se cumple $X \to Z$.

Hay otras reglas de inferencia que se deducen de los axiomas de Armstrong y que permiten calcular más rápidamente el cierre de un conjunto de dependencias funcionales.

4. Autodeterminación: $X \to X$.

5. Unión: Si $X \to Y$ y $X \to Z$, entonces $X \to YZ$.

6. Descomposición: Si $X \to YZ$, entonces $X \to Y$ y $X \to Z$.

7. Composición: Si $X \to Y$ y $Z \to W$, entonces $XZ \to YW$.

8. Pseudotransitividad: Si $X \to Y$ e $YZ \to W$, entonces $XZ \to W$.

La autodeterminación se sigue directamente de la reflexividad. La unión y la descomposición son duales.

Ejemplo 7.

Dado el conjunto S de dependencias funcionales:

$\{A\} \to \{B, C\}$

$\{C, D\} \to \{E, F\}$

Se puede demostrar que $\{A, D\} \to \{F\}$ está en S^+:

$\{A\} \to \{B, C\}$, dada.

$\{A\} \to \{C\}$, descomposición.

$\{A, D\} \to \{C, D\}$, aumentatividad.

$\{C, D\} \rightarrow \{E, F\}$, dada.
$\{A, D\} \rightarrow \{E, F\}$, transitividad de las dos anteriores.
$\{A, D\} \rightarrow \{F\}$, descomposición.

Se puede desarrollar un algoritmo que calcule el cierre del conjunto de dependencias funcionales a partir de sólo las tres primeras reglas de inferencia aplicándolas repetidamente hasta que no se produzcan más dependencias funcionales (se alcance el punto fijo). Este algoritmo es seguro con respecto a la completitud de los axiomas. La demostración de completitud necesita la noción de cierre de un conjunto de atributos. Sin embargo, también es un algoritmo muy ineficiente por la cantidad de dependencias funcionales que se generan.

Ejemplo 8.
Dado el conjunto de dependencias funcionales:
$S = \{X \rightarrow \{B_1\}, \ldots, X \rightarrow \{B_n\}\}$
El cierre de S incluye todas las dependencias funcionales $X \rightarrow Y_i$ tales que $Y_i \subseteq \{B_1, \ldots, B_n\}$, es decir, $2^n - 1$, demasiado grande aunque S sea pequeño.

5.3.2 Cierre de un conjunto de atributos

En la práctica no es necesario en general calcular todo el cierre de un conjunto de dependencias. Es más interesante calcular el conjunto de las dependencias que tienen en su parte izquierda un conjunto especificado de atributos.
El cálculo del cierre de un conjunto de atributos permite:

1. Comprobar si una dependencia funcional se deduce de un conjunto de dependencias funcionales sin necesidad de calcular su cierre. Se puede determinar si su comprobación es redundante para la integridad de los datos.
2. Comprobar si un conjunto de atributos es superclave. Asegura que el conjunto de atributos elegido por el diseñador es adecuado para determinar unívocamente cada tupla de una relación. Permite determinar superclaves que se pueden usar como índice sin repetidos (algoritmo de indexación más eficiente) para el acceso a los datos mediante consultas.
3. Calcular un conjunto mínimo de dependencias funcionales. Útil para mantener la comprobación de integridad menos costosa.

Definición 5. Cierre de un conjunto de atributos
El cierre de un conjunto de atributos X con respecto a un conjunto de dependencias funcionales S, denotado X_S^+, es el conjunto de atributos Y tales que $X \rightarrow Y$ se puede deducir de S.
En otras palabras, el cierre de un conjunto de atributos X es el conjunto de atributos Y determinados funcionalmente por X.
Ahora se puede usar el siguiente lema para asegurar el primer punto anterior.

Lema 1:
$X \rightarrow Y$ se deduce de un conjunto de dependencias funcionales $S \Leftrightarrow Y \subseteq X_S^+$.

Demostración:
Sea $Y = \{B_1, \ldots, B_n\}$:

$\Leftarrow$) Supongamos $Y \subseteq X_S^+$. Por la definición de X_S^+ se deduce en particular el conjunto de todas las dependencias funcionales $X \to \{B_i\}$. Por la regla de la unión para todas las $X \to \{B_i\}$ ($X \to \{B_1\}, \ldots, X \to \{B_n\}$) se deduce $X \to Y$.
$\Rightarrow$) Supongamos $X \to Y$. Por la regla de descomposición se deducen todos los $X \to \{B_i\}$ y, por tanto, $Y \subseteq X_S^+$.

Algoritmo 3.
Algoritmo simple para calcular el cierre de un conjunto de atributos
Entrada: Conjunto de atributos X y un conjunto de dependencias funcionales S.
Salida: X_S^+

```
Resultado := X
while cambios en Resultado do
  for each Y → Z ∈ S do
    if Y ⊆ Resultado then Resultado := Resultado ∪ Z
```

Se puede demostrar que el algoritmo es correcto y completo. El algoritmo tiene una complejidad cuadrática con la cardinalidad de S. Existen otros algoritmos de complejidad lineal. Estos aspectos se tratan en los ejercicios.

Corolario 2.
Se puede determinar si una dependencia funcional $X \to Y$ se deduce de un conjunto S de dependencias funcionales si $Y \subseteq X_S^+$. Se puede determinar, por tanto, en tiempo lineal, si una dependencia funcional está en S^+.

Corolario 3.
Se puede determinar si un conjunto de atributos C es superclave de una relación r bajo un conjunto de dependencias funcionales S si todos los atributos de r pertenecen al cierre de C, es decir, si todos los atributos de la relación están determinados funcionalmente por C. Además, será clave candidata si el conjunto de atributos C es irreducible (no hay ningún conjunto de cardinalidad menor que C tal que determine funcionalmente todos los atributos de r).

Ejercicio.
Proponer un algoritmo para determinar el conjunto de claves candidatas de una relación.

A continuación ya es posible definir lo que es un recubrimiento mínimo de dependencias o conjunto irreducible equivalente, que va a permitir mantener la integridad definida por un conjunto de dependencias funcionales a coste mínimo.

5.3.3 Recubrimientos mínimos de dependencias funcionales

Para definir un recubrimiento mínimo hay que definir dos conceptos: el recubrimiento de un conjunto de dependencias funcionales y la equivalencia entre conjuntos de dependencias funcionales.

Definición 6. Recubrimiento de un conjunto de dependencias funcionales

Dados dos conjuntos de dependencias funcionales S_1 y S_2, se dice que S_2 es un recubrimiento de S_1 si cada dependencia de S_1 se deduce de S_2 (es decir, se puede demostrar que cada dependencia de S_1 está en el cierre de S_2).

Definición 7. Equivalencia entre conjuntos de dependencias funcionales
Dos conjuntos de dependencias funcionales S_1 y S_2 son equivalentes si $S_1^+ = S_2^+$.
De forma alternativa se define como: Dos conjuntos de dependencias funcionales S_1 y S_2 son equivalentes si S_1 es un recubrimiento de S_2 y S_2 es un recubrimiento de S_1.
La comprobación de la equivalencia se hace usando el Corolario 2. Así, para cada dependencia funcional de S_1 se comprueba si ésta pertenece al cierre de S_2. De igual forma se procede con cada dependencia de S_2 comprobando que pertenezca al cierre de S_1.

Definición 8. Conjunto mínimo (o irreducible) de dependencias funcionales
Un conjunto S de dependencias funcionales es irreducible si y solamente si cumple las siguientes propiedades:

1. La parte derecha de cada dependencia funcional de S tiene sólo un atributo.
2. La parte izquierda de cada dependencia funcional de S es irreducible en el sentido en que si se elimina algún atributo, necesariamente cambia el cierre de S.
3. No se puede eliminar ninguna dependencia funcional de S sin cambiar su cierre.

Definición 9. Recubrimiento mínimo de un conjunto de dependencias funcionales
Al conjunto mínimo de dependencias funcionales S_1 equivalente a S_2 se le denomina recubrimiento mínimo de S_2.

Se puede demostrar que todo conjunto de dependencias funcionales tiene al menos un recubrimiento mínimo, por lo que se plantea el siguiente lema.

Lema 2.
Todo conjunto S de dependencias funcionales tiene un conjunto de dependencias funcionales equivalente en el que el lado derecho de cada dependencia funcional tiene un único atributo.
Demostración:
La regla de descomposición asegura que una dependencia funcional $X \to \{B_1, \dots, B_n\}$ se puede dividir en n dependencias funcionales $X \to \{B_1\}, \dots, X \to \{B_n\}$.

Teorema 1.
Todo conjunto S de dependencias funcionales tiene al menos un recubrimiento mínimo.
Demostración:
1. Por la aplicación del lema 2 se obtiene un conjunto T equivalente a S de manera que todas las partes derecha de sus dependencias funcionales tienen un solo atributo.
2. Para cada dependencia funcional $X \to \{B\} \in T, X = \{A_1, \dots, A_n\}$ se examina cada atributo A_i: si la eliminación de A_i de X no tiene efecto sobre S^+, se elimina A_i de X.

3. Para cada dependencia funcional restante, si su eliminación no tiene efecto sobre S^+, se elimina de T.

Como la aplicación de estos pasos siempre elimina un atributo o una dependencia funcional, se alcanzará un conjunto de dependencias funcionales que se ajuste a la definición de recubrimiento mínimo.

Ejemplo 9.

$\{A\} \to \{B, C\}$
$\{B\} \to \{C\}$
$\{A\} \to \{B\}$
$\{A, B\} \to \{C\}$
$\{A, C\} \to \{D\}$

1. Primer paso:
   $\{A\} \to \{B\}$
   $\{A\} \to \{C\}$
   $\{B\} \to \{C\}$
   $\{A\} \to \{B\}$
   $\{A, B\} \to \{C\}$
   $\{A, C\} \to \{D\}$

Antes de aplicar el segundo paso, vemos que se repite $\{A\} \to \{B\}$, por lo que en el conjunto resultado no se repite.

2. Segundo paso:
   - Se puede eliminar C de $\{A, C\} \to \{D\}$ porque se tiene $\{A\} \to \{C\}$, y por la regla de aumentatividad se obtiene $\{A\} \to \{A, C\}$, como se tiene también $\{A, C\} \to \{D\}$, se deduce por transitividad $\{A\} \to \{D\}$.
3. Tercer paso:
   - Se puede eliminar $\{A, B\} \to \{C\}$ porque se tiene $\{A\} \to \{C\}$, y por la regla de aumentatividad se obtiene $\{A, B\} \to \{B, C\}$, y $\{A, B\} \to \{C\}$ por descomposición.
   - Se puede eliminar $\{A\} \to \{C\}$ porque se tiene $\{A\} \to \{B\}$, $\{B\} \to \{C\}$ y por transitividad se obtiene $\{A\} \to \{C\}$.

Se llega al recubrimiento mínimo T de S en el que las partes derecha de las dependencias funcionales tienen un solo atributo, las partes izquierda no son reducibles (ya que son unitarias) y no se puede eliminar ninguna dependencia funcional sin alterar S^+:

$T = \{\{A\} \to \{B\}, \{B\} \to \{C\}, \{A\} \to \{D\}\}$

Para comprobar que no se puede eliminar ninguna dependencia funcional se usa el corolario 2:

- Se comprueba si $\{A\} \to \{B\}$ se puede deducir de $U = \{\{B\} \to \{C\}, \{A\} \to \{D\}\}$ comprobando si $\{B\} \subseteq \{A\}_U^+ = \{A, D\}$, que no se cumple.
- Se comprueba si $\{B\} \to \{C\}$ se puede deducir de $U = \{\{A\} \to \{B\}, \{A\} \to \{D\}\}$ comprobando si $\{C\} \subseteq \{B\}_U^+ = \{B\}$, que no se cumple.
- Se comprueba si $\{A\} \to \{D\}$ se puede deducir de $U = \{\{A\} \to \{B\}, \{B\} \to \{C\}\}$ comprobando si $\{D\} \subseteq \{A\}_U^+ = \{A, B, C\}$, que no se cumple.

Por lo tanto, es irreducible.

5.4 Dependencias multivaloradas

Las dependencias multivaloradas son restricciones de integridad que expresan relaciones entre los atributos de un esquema que no pueden ser expresables con las dependencias funcionales.

Ejemplo 10.

En la siguiente relación se representan los empleados, sus domicilios y teléfonos, asumiendo que pueden tener más de una vivienda y teléfono, y que no se dispone información acerca del tipo de teléfono, fijo o móvil, por lo que no se puede relacionar con un domicilio. Estos atributos son independientes entre sí. Para mantener la relación consistente es necesario expresar todas las combinaciones de los atributos.

Empleados		
Nombre	**Dirección**	**Teléfono**
Ana Almansa	c/ Argentales	1
Ana Almansa	c/ Argentales	2
Ana Almansa	c/ Argentales	3
Ana Almansa	c/ Amaniel	1
Ana Almansa	c/ Amaniel	2
Ana Almansa	c/ Amaniel	3

Mientras que las dependencias funcionales impiden que aparezcan ciertas tuplas en las relaciones, las dependencias multivaloradas obligan a ello.

Las dependencias multivaloradas aparecen cuando en un esquema de relación hay varias relaciones 1:N independientes entre sí.

Definición 10. Dependencias multivaloradas

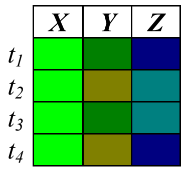
Dados dos subconjuntos de atributos X e Y y un esquema R, la dependencia multivalorada $X \rightarrow\rightarrow Y$ (X multidetermina a Y) especifica la siguiente restricción sobre la relación r del esquema R: si existen en r dos tuplas t_1 y t_2 tales que $t_1[X] = t_2[X]$, entonces deben existir dos tuplas, t_3 y t_4, tales que:

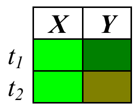
$t_1[X] = t_2[X] = t_3[X] = t_4[X]$

$t_1[Y] = t_3[Y]$ y $t_2[Y] = t_4[Y]$

$t_2[Z] = t_3[Z]$ y $t_1[Z] = t_4[Z]$, donde $Z = R - (X \cup Y)$

Esta definición es más sencilla de lo que parece si se observa el siguiente gráfico. En definitiva se imponen todas las combinaciones de los valores de los atributos Y y Z. Si Z es vacío o sus valores son únicos, necesariamente $t_1 = t_3$ y $t_2 = t_4$, es decir, estamos hablando sólo de dos tuplas.





Informalmente se dice que siempre que existan dos tuplas con valores iguales de X pero distintos de Y, los valores de Y se deben repetir en tuplas separadas por cada

valor distinto de Z. En definitiva, con esta restricción se dice que la relación entre X e Y es independiente de la relación entre X y Z.
Ejemplo 11.
En el ejemplo anterior se observan las restricciones multivaloradas
$\{Nombre\} \rightarrow\rightarrow \{Dirección\}$ y $\{Nombre\} \rightarrow\rightarrow \{Teléfono\}$.
Debido a la simetría de la definición (se pueden intercambiar los papeles de Y y Z) se deduce que si se cumple $X \rightarrow\rightarrow Y$, entonces también se cumple $X \rightarrow\rightarrow Z$, que se representa de forma compacta como $X \rightarrow\rightarrow Y \mid Z$.

Definición 11. Dependencias multivaloradas triviales
Una dependencia multivalorada $X \rightarrow\rightarrow Y$ se denomina trivial si $Y \subseteq X$ o $X \cup Y = R$. Se denomina trivial porque no aporta ninguna restricción relevante al esquema. En el primer caso, $Y \subseteq X$, sólo se impone que un subconjunto de los valores de X esté asociado siempre a los valores de X, lo cual es trivialmente cierto. El segundo caso se vio en la definición de dependencia multivalorada.

5.4.1 Reglas de inferencia para las dependencias multivaloradas

Para las dependencias multivaloradas también se proponen axiomas de producción que permiten calcular el cierre de un conjunto de ellas.

1. **Reflexividad para dependencias funcionales**:
   Si $Y \subseteq X$, entonces $X \rightarrow Y$.
2. **Aumentatividad para dependencias funcionales**:
   Si $X \rightarrow Y$, entonces $XZ \rightarrow YZ$.
3. **Transitividad para dependencias funcionales**:
   Si $X \rightarrow Y$ e $Y \rightarrow Z$, entonces $X \rightarrow Z$.
4. **Complemento para dependencias multivaloradas**:
   Si $X \rightarrow\rightarrow Y$, entonces $X \rightarrow\rightarrow R - (X \cup Y)$
5. **Aumentatividad para dependencias multivaloradas:**
   Si $X \rightarrow\rightarrow Y$ y $V \subseteq W$, entonces $WX \rightarrow\rightarrow VY$.
6. **Transitividad para dependencias multivaloradas**:
   Si $X \rightarrow\rightarrow Y$ e $Y \rightarrow\rightarrow Z$, entonces $X \rightarrow\rightarrow (Z - Y)$
7. Si $X \rightarrow Y$, entonces $X \rightarrow\rightarrow Y$.
8. Si $X \rightarrow Y$, $Z \subseteq Y$ y existe un W disjunto de Y, entonces $W \rightarrow Z$ y $X \rightarrow Z$.

Las tres primeras son las mismas reglas que los axiomas de Armstrong, las tres siguientes se refieren a dependencias multivaloradas y las dos últimas se refieren tanto a dependencias multivaloradas como funcionales. En concreto, la séptima se refiere a que una dependencia funcional es un caso particular de una dependencia multivalorada.
Se puede demostrar que este conjunto de reglas de inferencia es correcto y completo para calcular el cierre de un conjunto de dependencias, denotado por S^+.

5.5 Resumen

- Las dependencias funcionales permiten imponer restricciones de integridad que no son posibles de expresar con claves.
- Una dependencia funcional es una generalización del concepto de superclave.
- Se ha mostrado un procedimiento que permite determinar el mínimo número de dependencias funcionales necesario para la comprobación de la consistencia de una relación a partir de un conjunto inicial de dependencias funcionales. Con este

procedimiento el diseñador de la base de datos tiene una herramienta muy útil para mejorar el rendimiento de sus diseños.

- Las dependencias multivaloradas permiten asegurar la consistencia cuando se expresan atributos multivalorados independientes en un único esquema de relación.
- Las dependencias multivaloradas permiten establecer relaciones uno a varios y uno a uno entre los atributos de ***un*** esquema de relación, mientras que las dependencias multivaloradas permiten expresar relaciones varios a varios.
- Las dependencias funcionales y multivaloradas se usarán como herramienta fundamental en el proceso de normalización de esquemas. Además son útiles en la comprobación de consistencia de relaciones resultado de migraciones y de sistemas heredados.

Bibliografía

[ACPT00] P. Atzeni, S. Ceri, S. Paraboschi y R. Torlone, "Database Systems. Concepts, Languages and Architectures", McGraw-Hill, 2000.

[EN00] R. Elmasri y S.B. Navathe, "Fundamentals of Data Base Systems", Addison-Wesley, 2000.

[SKS98] A. Silberstachatz, Korth y Sudarshan, "Database System Concepts", McGraw-Hill, 2001
Apartados 3.1-3.4

[Ull98] J.D. Ullman, "Principles of Database and Knowledge Base Systems", Vol. I y II, Computer Science Press, 1998.