Refinamiento del Esquema y Formas Normales

Tema 16

Problemas de la Redundancia

- En un esquemas la *redundancia* pueden causar los siguientes problemas:
  - ***Almacenamiento innecesario de la misma información en distintos lugares de la base de datos.***
  - ***Anomalías en las operaciones insert/delete/update***
- Relación Hourly_emps (<u>ssn</u>, name, lot, rating, hourly_wages, hourly_worked)
- La llave es ssn. hourly_wages esta determinado por rating (para cada valor de rating, hay un solo valor permitido para hourly_wages) Esta restricción de integridad es un ejemplo de dependencia funcional.

Continuación

- Las *dependencias funcionales*, son un tipo de restricción de integridad que permite identificar problemas en los esquemas y sugerir su refinamiento.
- La técnica principal de refinamiento es: *descomposición* que consiste en reemplazar una relación con una colección de relaciones más pequeñas, y cada una contiene un subconjunto de atributos de la relación original. Ej.
- Hourly_emps2(ssn,name, lot, rating, hours_worked)
  Wages(rating, hourly_wages)
- La descomposición se debe usar con criterio:
  - ¿Existe razón para descomponer una relación?
  - ¿Qué problemas (si hay) causa la descomposición?

Dependencias Funcionales (FDs)

- Hay una <u>dependencia funcional</u> X → Y sobre la relación R si, por cada instancia legal *r* de R:
  - $t1 \in r, t2 \in r, \pi_x(t1) = \pi_x(t2) \Rightarrow \pi_y(t1) = \pi_y(t2)$
  - Ej., dadas dos tuplas en *r*, si el valor X coincide, entonces el valor Y también coincide. (X y Y son *conjuntos* de atributos)
- Una instancia legal debe satisfacer *todas* las restricciones de integridad, incluso las dependencias funcionales.
  - Deben ser identificadas basado en la semántica de la aplicación
  - Dada una instancia *r1* de R es posible verificar si ésta no cumple con una FD *f*, pero no es posible decir si *f* se preserva sobre R!
- K es una llave candidata de R significa que K → R
- Una restricción de llave primaria es un caso especial de FD. Los atributos en la llave son X, el conjunto de todos es Y.

Ejemplo: Restricción sobre conjunto de entidades

- Considere la relación que se obtiene de Hourly_Emps:
  - Hourly_Emps (<u>*ssn*,</u> *name, lot, rating, hrly_wages, hrs_worked*)
- <u>*Notación*</u>: Denotaremos el esquema de la relación por la lista de atributos: SNLRWH
  - Que es realmente *el conjunto* de atributos {S,N,L,R,W,H}.
  - Algunas veces, haremos referencia a todos los atributos usando el nombre de la relación. (Ejemplo, Hourly_Emps por SNLRWH)
- Algunos FDs sobre Hourly_Emps:
  - *ssn* es la llave: $S \rightarrow SNLRWH$
  - *rating* determina *hrly_wages*: $R \rightarrow W$

Ejemplo (Cont.)

- Problemas debidos a R → W :
  - <u>*Anomalias al modificar*</u>: ¿Es posible modificar W solo en la 1a tupla de SNLRWH?
  - <u>*Anomalias al insertar*</u>: ¿Qué pasa si queremos agregar un empleado el valor de hourly wage asociado a su rating?
  - <u>*Aomalias al eliminar*</u>: Si se desea eliminar todos los empleados con rating 5, perderemos la información de wage para rating 5!

S	N	L	R	W	H
123-22-3666	Attishoo	48	8	10	40
231-31-5368	Smiley	22	8	10	30
131-24-3650	Smethurst	35	5	7	30
434-26-3751	Guldu	35	5	7	32
612-67-4134	Madayan	35	8	10	40

S	N	L	R	H
123-22-3666	Attishoo	48	8	40
231-31-5368	Smiley	22	8	30
131-24-3650	Smethurst	35	5	30
434-26-3751	Guldu	35	5	32
612-67-4134	Madayan	35	8	40

Hourly_Emps2

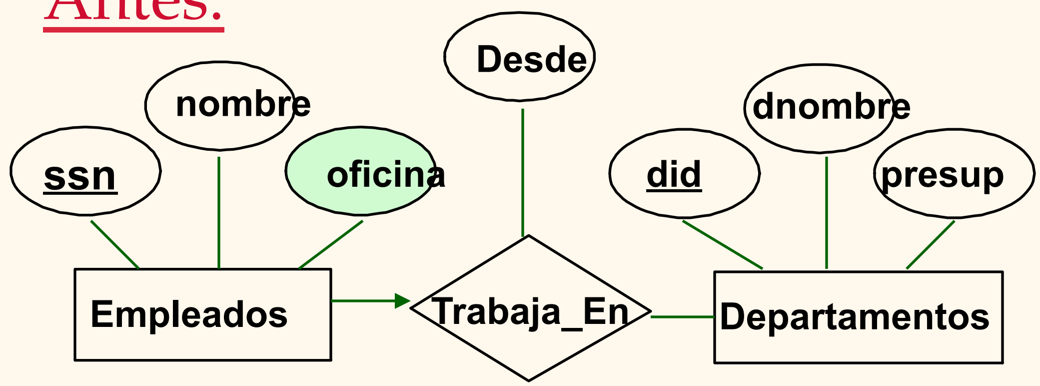
R	W
8	10
5	7

Wages

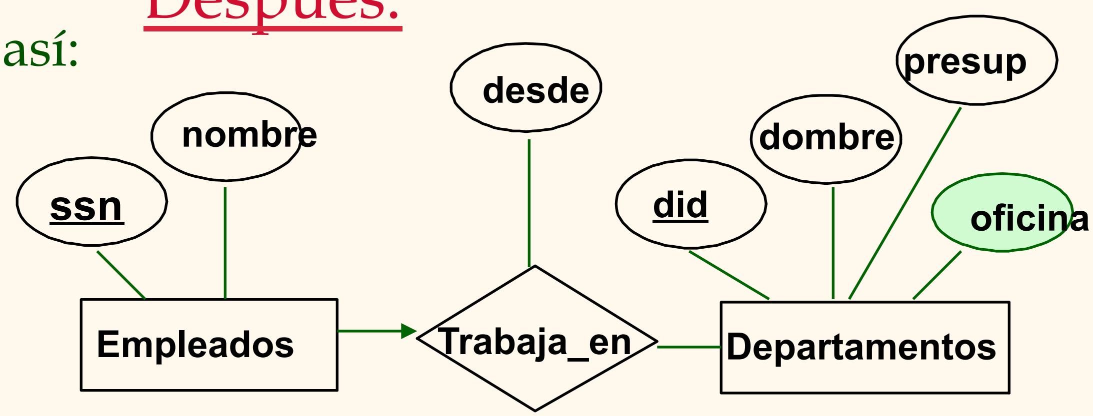
Refinar un diagrama ER

- 1er diagrama traducido:
  Trabajadores(S,N,O,D,S)
  Departamentos(D,M,B)
  - Oficina asociado con Trabajadores.
- Supongamos que todos los Trabajadores de un departamento están asignados a la misma oficina: D → O
- Hay redundancia; se corrige así:
  Trabajadores2(S,N,D,S)
  Depto_Oficina(D,O)
- Podemos refinar:
  Trabajadores(S,N,D,S)
  Departmentos(D,M,B,O)

Antes:



Después:



Aspectos sobre FDs

- Dado algún FDs, es posible deducir FDs adicionales:
  - $ssn \rightarrow did$, $did \rightarrow lot$ implica $ssn \rightarrow lot$
- Un FD f es *implicada por* un conjunto de FDs F si f se preservan cuando todos los FDs en F se preservan.
  - F^+ = *cerradura de F* es el conjunto de FDs implicados por F.
- Axiomas de Armstrong (X, Y, Z conjuntos de atributos):
  - *Reflexividad*: Si $Y \subseteq X$, entonces $X \rightarrow Y$
  - *Aumento*: Si $X \rightarrow Y$, entonces $XZ \rightarrow YZ$ para cualquier Z
  - *Transitividad*: Si $X \rightarrow Y$ y $Y \rightarrow Z$, entonces $X \rightarrow Z$

Aspectos sobre FDs (Cont)

- Reglas adicionales (que siguen de los axiomas):
  - *Unión*: Si $X \to Y$ y $X \to Z$, entonces $X \to YZ$
  - *Descomposición*: Si $X \to YZ$, entonces $X \to Y$ y $X \to Z$
- Ejemplo: Contratos(*cid,sid,jid,did,pid,qty,value*), y:
  - C es la llave: $C \to CSJDPQV$
  - Proyecto compra una parte usando un solo contrato: $JP \to C$
  - Departamento compra a lo mas una parte de proveedor: $SD \to P$
- **$JP \to C$, $C \to CSJDPQV$ implica $JP \to CSJDPQV$**
- **$SD \to P$ implica $SDJ \to JP$**
- **$SDJ \to JP$, $JP \to CSJDPQV$ implica $SDJ \to CSJDPQV$**

Aspectos sobre FDs (Cont)

- Calcular toda la cerradura del un conjunto de FDs puede ser muy costoso.
- Lo que se desea es saber si un FD $X \rightarrow Y$ esta en la cerradura del F. Una verificación eficiente:
  - Calcular *atributos de la cerradura* de X (denotado X^+)
    - Conjunto de los atributos A tal que $X \rightarrow A$ que esté en F^+
    - Hay un algoritmo para calcularlo.
  - Verificar si Y esta en X^+
- ¿Si $F = \{A \rightarrow B,\ B \rightarrow C,\ C\,D \rightarrow E\}$ implica $A \rightarrow E$?
  - Ej, ¿Esta $A \rightarrow E$ in F^+? Equivalentemente, ¿Está E in A^+?

Cerradura de conjuntos de atributos

- El siguiente algoritmo calcula : [illegible]$^{+}$

result := [illegible]

while (changes to *result*) **do**

  for each dependencia funcional $\beta \rightarrow \gamma$ **in** *F* **do**

  begin

  if $\beta \subset$ *result*

    then *result* := *result* $\cup\ \gamma$;

  end

Formas Normales

- Regresando a la aplicación del refinamiento del esquema, la primera pregunta es si es necesario algún refinamiento!
- Si una relación esta en cierta *forma normal* (BCNF, 3NF etc.), se sabe que hay problemas que se han evitado/ minimizado. Esto permite decidir si hay beneficio al descomponer la relación..
- El rol de los FDs al detectar redundancia:
  - Si R es una relación con 3 atributos, ABC.
    - Si no hay FDs: No hay redundancia.
    - Dado $A \rightarrow B$:   Muchas tuplas podrian tener el mismo valor de A y entonces tendrían todas el mismo valor B!

Formas Normales

- Una relación esta en **primera forma normal** si cada campo posee únicamente valores atómicos, es decir, no listas o conjuntos.
- **2NF** es de interés histórico, y dice que no es permitido una dependencia parcial..
- **3NF y BCNF** son importantes desde el punto de vista de diseño del diseño de la base de datos.

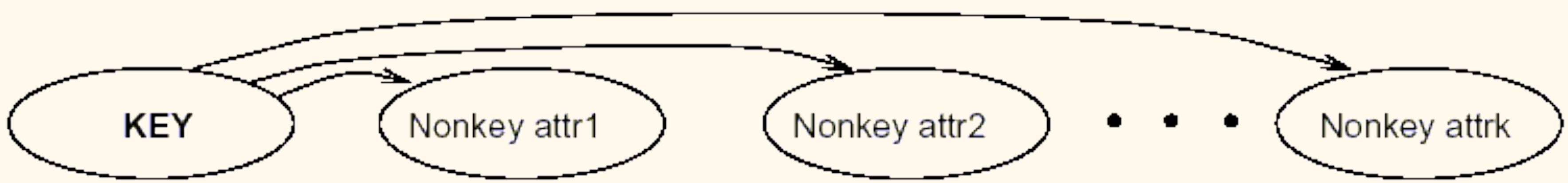
Forma Normal Boyce-Codd (BCNF)

- Si R es el esquema de una relación, X un subconjunto de atributos de R, y si A es un atributo de R. R esta en BCNF si para cada FD X → A in F^+
  - A ∈ X (llamada FD *trivial*) o
  - X contiene una llave para R. (X es una súper llave)
- En otras palabras, R está en BCNF si las únicas FDs no triviales que se mantienen sobre R son restricciones de llave. (Es suficiente verificar si el lado izquierdo de cada dependencia en F es una súper llave (Calcular $Atributo^+$ y observar si incluye todos los atributos de R))
  - Si X → A y se tienen dos tuplas que coinciden en el valor X es posible deducir el valor A en una de las tuplas a partir del valor A de la otra.
  - Si la relación de ejemplo esta en BCNF, las 2 tuplas deben ser idénticas. (puesto que X es una llave).

X	Y	A
x	y1	a
x	y2	?

BCNF

Si se usan óvalos para denotar los atributos o conjuntos de atributos y se dibujan arcos para indicar FDs, una relación en BCNF posee la siguiente estructura, considerando solo una llave. (si hay varias llaves candidatas, cualquiera puede ser llave)



Si una relación está en BCNF, cada campo de cada tupla posee información que no se puede deducir (usando solamente FD's) de los valores de los otros campos (de todas las tuplas) en la instancia de la relación.

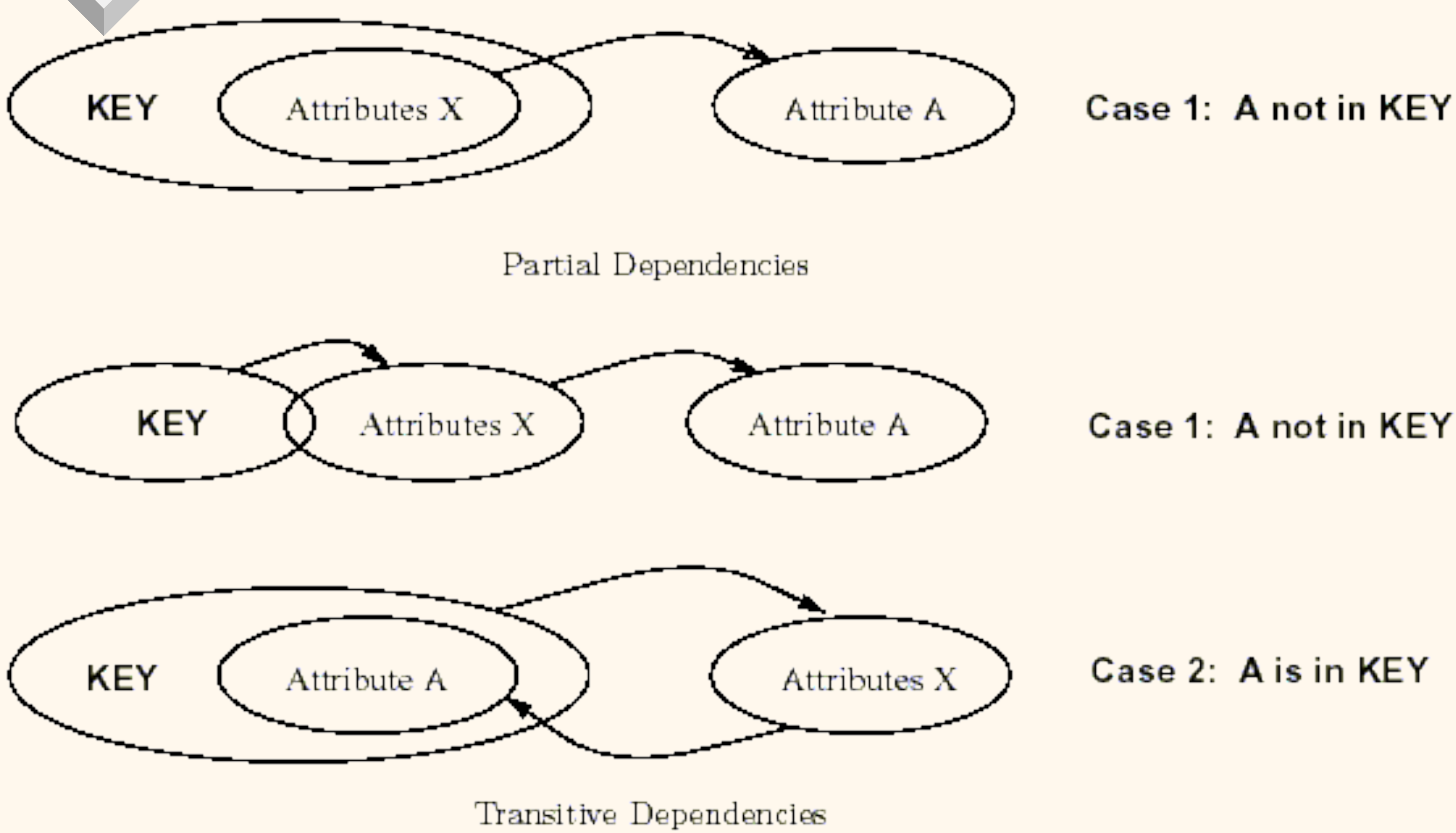
Tercera Forma Normal (3NF)

- Si R es el esquema de una relación, X un subconjunto de atributos de R, y si A es un atributo de R. R esta en 3NF si para cada FD $X \rightarrow A$ in F^+
  - $A \in X$ (llamada FD *trivial*) o
  - X contiene una llave para R, (X es una súper llave) o
  - A es parte de alguna llave para R
- *Mínimo* de una llave es crucial en la tercera condición!
- Si R esta en BCNF, obviamente en 3NF.
- Si R esta en 3NF, es posible alguna redundancia.

¿Qué alcance hay con 3NF?

- Si X $\rightarrow$ A causa una violación de 3NF, se tiene lo siguiente:
  - X es un subconjunto de una llave K (Dependencia Parcial)
    - Se almacenan pares (X, A) redundantemente. (Ej. Reserves (SBDC))
  - X no es un subconjunto propio para llave (Dependencia transitiva)
    - Hay una cadena de FDs K $\rightarrow$ X $\rightarrow$ A, que significa que no es posible asociar un valor X con una valor K a menos que se asocie un valor A con un valor X. (Ej. hourlyEmp (SNLRWH))
- Pero: aunque la relación este en 3NF, estos problemas podrían presentarse.
  - Ej., Reserves SBDC, S $\rightarrow$ C, C $\rightarrow$ S esta en 3NF, por cada reservación del marinero S, se almacena algun par (S, C).

3NF


KEY
Attributes X
Attribute A
Case 1: A not in KEY
Partial Dependencies
KEY
Attributes X
Attribute A
Case 1: A not in KEY
KEY
Attribute A
Attributes X
Case 2: A is in KEY
Transitive Dependencies

Descomposición del esquema de una relación R

- Si R tiene atributos *A1 … An.* Una *descomposición* de R es reemplazar R por dos o mas relaciones tales que:
  - El esquema de cada nueva relación posea un subconjunto de atributos de R (y no posea atributos que no son de R) y
  - Cada atributo de R aparezca como atributo de una de las nuevas relaciones (Entre todas incluyan el total de atributos de R)
- Descomponer R significa que se almacenaran instancias de los nuevos esquemas, en vez de instancias de R.
- Ej. SNLRWH Se puede descomponer en SNLRH y RW.

Ejemplo de descomposición

- La descomposición se usa solo cuando es necesaria
  - SNLRWH tiene FDs $S \rightarrow SNLRWH$ y $R \rightarrow W$
  - Segundo FD viola la 3NF; los valores de W están varias veces asociados con los valores de R. La manera más de corregirlo es creando una relación RW para almacenar las asociaciones y, y quitar W del esquema original:
    - Ej, Descomponer SNLRWH en SNLRH y RW

Problemas con Descomposición

- Hay tres problemas que considerar:
  - ❶ Algunas consultas tendrán un mayor costo.
    - Ej, ¿Cuánto recibirá el empleado Joe? (salario = W*H)
  - ❷ No es posible reconstruir la instancia original de la relación a partir de las instancias de las relaciones de la descomposición!
    - Afortunadamente no sucede en el ejemplo SNLRWH.
  - ❸ Para verificar algunas dependencias es necesario hacer join de las instancias de las relaciones de la descomposición.
    - Afortunadamente no sucede en el ejemplo SNLRWH.
- *Considerar*: Estos problemas vs. redundancia.

Descomposición Lossless-Join

- La descomposición del esquema en 2 esquemas con conjuntos de atributos X y Y es *lossless-join* con respecto a un conjunto de FDs F si, para cada instancia *r* de R que satisface F:
  - $\pi_X(r) \bowtie \pi_Y(r) = r$
- Siempre es verdadero que $r \subseteq \pi_X(r) \bowtie \pi_Y(r)$
  - En general, no se da en la otra dirección! Si esto se da, la descomposición es lossless-join.
- La definición se extiende a la descomposición en 3 o más relaciones en forma directa.
- *Es importante que las descomposiciones usadas para repartir redundancias sean lossless-join! (Evita el problema 2.)*

Mas a cerca de Lossless-Join

- La descomposición de R en X y Y es lossless-join con respecto a F si y solo si la cerradura de F posee:
  - $X \cap Y \rightarrow X$, o
  - $X \cap Y \rightarrow Y$
- En particular, la descomposición de R en UV y R - V es lossless-join si se mantiene $U \rightarrow V$ sobre R y $U \cap V$ es vacio.

A	B	C
1	2	3
4	5	6
7	2	8

A	B
1	2
4	5
7	2

B	C
2	3
5	6
2	8

A	B	C
1	2	3
4	5	6
7	2	8
1	2	8
7	2	3

Descomposiciones que preserva las dependencias

- Dado CSJDPQV, C es llave, JP $\rightarrow$ C y SD $\rightarrow$ P.
  - BCNF descomposición: CSJDQV y SDP
  - Problema: verificar JP $\rightarrow$ C requiere hacer join!
- Descomposición que preserva dependencias:
  - Si R se descompone en X, Y y Z, y se cumplen las FDs sobre X, Y y Z, entonces todas FDs dadas sobre R se deben cumplir (Se pueden cumplir todas las FDs examinando una sola instancia de la relación en cada inserción o modificación de la tupla). *(Evitando el problema 3)*
- *Proyección de conjuntos de FDs F*: Si R se descompone en X, ... La proyección F en X (F_X) es el conjunto de FDs U $\rightarrow$ V en F^+ tales que U, V esten en X.

Descomposiciones que preservan las dependencias (Cont)

- La descomposicion de R en X y Y *preserva las dependencias* si $(F_X \text{ unión } F_Y)^+ = F^+$
  - Ej., al tomar las dependencias Fx y Fy y calcular la cerradura de la unión de ellas, se obtienen todas las dependencias de la cerradura de F. Por lo tanto es necesario que se cumplan solamente las dependencias en Fx y Fy.
- Es importante considerar F^+, no F, en la definición:
  - ABC, $A \rightarrow B$, $B \rightarrow C$, $C \rightarrow A$, descompuesta en AB y BC.
  - Se preservan las dependencias? ¿Se preserva $C \rightarrow A$?
- Que se mantengan las dependencias no implica lossless join:
  - ABC, $A \rightarrow B$, descompuesto AB y BC.

Descomposición en BCNF

- Consideremos la relación R con FDs F. X ⊂ R, Y debe ser un solo atributo. Si X → Y viola BCNF, descomponer R en R - Y y XY.
  - Repetir esta idea da una colección de relaciones que están en BCNF; descomposición lossless-join, y garantiza terminar.
  - Ej, CSJDPQV, Llave C, JP → C, SD → P, J → S
  - Al tratar con SD → P, descomponer en SDP, CSJDQV.
  - Al tratar con J → S, descomponer CSJDQV en JS y CJDQV
- En general, varias dependencias pueden causar violación de la BCNF. El orden en que se traten nos conduce a diferentes conjuntos de relaciones!

BCNF y preservación de dependencias

- En general, podría no haber preservación de dependencias en descomposición en BCNF.
  - Ej, CSZ, $CS \rightarrow Z$, $Z \rightarrow C$
  - No se puede descomponer mientras se preserve la primera FD; no en BCNF.
- De forma similar la descomposición de CSJDQV en SDP, JS y CJDQV no preserva las dependencias (con respecto a las FDs $JP \rightarrow C$, $SD \rightarrow P$ y $J \rightarrow S$).
  - Sin embargo, es una descomposición lossless-join.
  - En esta caso al agregar JPC a la colección de dependencias se logra preservar la dependencia.
    - Las tuplas JPC permiten verificar FD (*Redundancia*)

Descomposición en 3NF

- El algoritmo para descomposición lossless-join en BCNF puede ser usado para descomposición lossless-join en 3NF.
- Para asegurar la preservación de la dependencia:
  - Si $X \rightarrow Y$ no se preserva, agregar la relación XY.
  - El problema es que XY puede violar 3NF! Ej., Si se agrega CJP para preservar $JP \rightarrow C$ y tambien se tiene $J \rightarrow C$?
- Refinamiento: En vez del conjunto de F de FDs, usar una *cubierta mínima de F*.

Cubierta Mínima para un conjunto de FDs

- *Cubierta mínima* G para un conjunto F de FDs:
  - Cerradura de F = Cerradura de G.
  - El lado derecho de cada FD en G es un solo atributo.
  - Si se modifica G eliminando un FD eliminando atributos de un FD en G, la cerradura cambia.
- Cada FD en G es necesaria, y "*tan pequeña como sea posible*" para conseguir la misma cerradura que F.
- Ej, $A \rightarrow B$, $ABCD \rightarrow E$, $EF \rightarrow GH$, $ACDF \rightarrow EG$ Tiene la siguiente cubierta mínima:
  - $A \rightarrow B$, $ACD \rightarrow E$, $EF \rightarrow G$ y $EF \rightarrow H$

Cont

1. **Put the FDs in a standard form:** Obtain a collection G of equivalent FDs with a single attribute on the right side (using the decomposition axiom).
2. **Minimize the left side of each FD:** For each FD in G, check each attribute in the left side to see if it can be deleted while preserving equivalence to F^+.
3. **Delete redundant FDs:** Check each remaining FD in G to see if it can be deleted while preserving equivalence to F^+.

Dependencias Multivaluadas

course	*teacher*	*book*
Physics101	Green	Mechanics
Physics101	Green	Optics
Physics101	Brown	Mechanics
Physics101	Brown	Optics
Math301	Green	Mechanics
Math301	Green	Vectors
Math301	Green	Geometry

- Si R es el esquema de una relación donde X y Y son subconjuntos de atributos de R. La **dependencia multivaluada X $\rightarrow\rightarrow$ Y** se mantiene sobre R si, en cada instancia legal r of R, cada valor de X esta asociado con un conjunto de valores de Y y este cojunto es independiente de los valores en los otros atributos.

Continuación.

- Formalmente, si se mantiene X →→ Y sobre R y Z=R – XY, lo siguiente es verdadero para cada instancia legal r de R:

  Si t1 ∈ r, t2 ∈ r y t1.X=t2.X. entonces debe haber un t3 ∈ r tal que t1.XY=t3.XY y t2.Z=t3.Z

X	Y	Z	
a	b_1	c_1	— tuple t_1
a	b_2	c_2	— tuple t_2
a	b_1	c_2	— tuple t_3
a	b_2	c_1	— tuple t_4

Continuación

- Dado un conjunto de FDs y MVDs, en general se puede deducir que que FDs y MVDs adicionales se mantienen. Un conjunto de A reglas consiste de los tres axiomas de Armstrongv mas las 5 reglas adicionales. 3 se refieren únicamente a MVDs:

- **MVD Complementation:** If $X \rightarrow\rightarrow Y$, then $X \rightarrow\rightarrow R - XY$.
- **MVD Augmentation:** If $X \rightarrow\rightarrow Y$ and $W \supseteq Z$, then $WX \rightarrow\rightarrow YZ$.
- **MVD Transitivity:** If $X \rightarrow\rightarrow Y$ and $Y \rightarrow\rightarrow Z$, then $X \rightarrow\rightarrow (Z - Y)$.

- **Replication:** If $X \rightarrow Y$, then $X \rightarrow\rightarrow Y$.
- **Coalescence:** If $X \rightarrow\rightarrow Y$ and there is a W such that $W \cap Y$ is empty, $W \rightarrow Z$, and $Y \supseteq Z$, then $X \rightarrow Z$.

Cuarta Forma Normal

- Si R es el esquema de una relación, X y Y son subconjuntos no vacíos de atributos de R, y F es un conjuntos de dependencias (FDs y MVDs). Se dice que R esta en 4NF si por cada MVD $X \rightarrow\rightarrow Y$ que hay sobre R, alguna de las siguientes es verdad:
  - $Y \subset X$ o $XY = R$ o
  - X es una superllave
- Si el esquema de una relación esta en BCNF y por lo menos una de sus llaves posee un solo atributo, también esta en 4NF

Join de dependencias

- Se dice que sobre una relación R se mantiene un **Join de dependencias (JD) >< {R1,…,Rn}** si R1,...,Rn es una descomposición lossless-join de R.
- Un MVD $X \rightarrow\rightarrow Y$ sobre la relación R puede ser expresada como el join de dependencias >< {XY,X(R-Y)}. Ejemplo, en la relación CTB, la MVD $C \rightarrow\rightarrow Y$ puede ser expresada como el join de dependencias >< {CT,CB}

Quinta Forma Normal

- El esquema de una relación R se dice que esta en 5NF si para cada JD >< {R1,…,Rn} sobre R, es verdad una de las siguientes:
  - Ri = R para algún i o
  - El JD es implicado por el conjunto de las FDs sobre R donde el lado izquierdo es una llave de R
- Si el esquema de una relación esta en 3NF y cada una de sus llaves es un solo atributo, entonces también esta en 5NF.

Continuación

- **repeat**

  Usar la regla de la unión para reemplazar las dependencias con la forma $\alpha_1 \rightarrow \beta_1$ y $\alpha_1 \rightarrow \beta_2$ con $\alpha_1 \rightarrow \beta_1\beta_2$.

  Buscar una dependencia funcional $\alpha \rightarrow \beta$ with an atributo extraño en α o en β.

  Si se encuentra un atributo extraño borrarlo de $\alpha \rightarrow \beta$

- **until** *F no cambie.*