

Lezione 19 – Algoritmo di decomposizione

Prof.ssa Maria De Marsico
demarsico@di.uniroma1.it



SAPIENZA
UNIVERSITÀ DI ROMA

- In questa lezione mostreremo che dato uno schema di relazione R e un insieme di dipendenze funzionali F su R esiste **sempre** una decomposizione $\rho = \{R_1, R_2, \dots, R_k\}$ di R tale che:
 - per ogni $i, i=1, \dots, k$, R_i è in 3NF
 - ρ preserva F
 - ρ ha un join senza perdita
- ... e che una tale decomposizione può essere calcolata in tempo polinomiale.

- Il seguente algoritmo, dato uno schema di relazione R e un insieme di dipendenze funzionali F su R , che è una **copertura minimale**, permette di **calcolare** in tempo polinomiale una decomposizione $\rho = \{R_1, R_2, \dots, R_k\}$ di R tale che:
 - per ogni $i, i=1, \dots, k$, R_i è in 3NF
 - ρ preserva F
- Ci interessa una **qualunque** copertura minimale dell'insieme di dipendenze funzionali definite sullo schema R .
- Se ce ne fosse più di una, con eventualmente cardinalità diversa, potremmo scegliere ad esempio quella **con meno dipendenze**, ma questo non è tra i nostri scopi.
- Quindi per fornire l'input all'algoritmo di decomposizione è **sufficiente trovarne una tra quelle possibili**.
- Poi vedremo **perché** ci occorre che sia una copertura minimale.

Algoritmo per la decomposizione di uno schema



SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Algoritmo – decomposizione di uno schema di relazione

Input uno schema di relazione R e un insieme F di dipendenze funzionali su R , che è una copertura minimale;

Output una decomposizione ρ di R che preserva F e tale che per ogni schema di relazione in ρ è in 3NF;

begin

$S := \emptyset$;

for every $A \in R$ tale che A non è coinvolto in nessuna dipendenza funzionale in F do

$S := S \cup \{A\}$;

if $S \neq \emptyset$ then

 begin

$R := R - S$;

$\rho := \rho \cup \{S\}$

 end

if esiste una dipendenza funzionale in F che coinvolge tutti gli attributi in R

 then $\rho := \rho \cup \{R\}$

else for every $X \rightarrow A \in F$ do $\rho := \rho \cup \{XA\}$

end

R residuo dopo aver eventualmente eliminato gli attributi inseriti prima in S

in questo caso ci fermiamo anche se la copertura minimale contiene anche altre dipendenze; in altre parole la copertura minimale potrebbe contenere anche altre dipendenze

Teorema Sia R uno schema di relazione ed F un insieme di dipendenze funzionali su R , che è una copertura minimale. L'Algoritmo di decomposizione permette di calcolare in tempo polinomiale una decomposizione ρ di R tale che:

- ogni schema di relazione in ρ è in 3NF
- ρ preserva F .

Dim.

- Dimostriamo separatamente le due proprietà della decomposizione
- ρ preserva F .

Sia $G = \bigcup_{i=1}^k \pi_{R_i}(F)$. Poiché per ogni dipendenza funzionale $X \rightarrow A \in F$ (proprio per tutte!) si ha che $XA \in \rho$ (è proprio uno dei sottoschemi), si ha che questa dipendenza di F sarà sicuramente in G , quindi $G \supseteq F$ e, quindi $G^+ \supseteq F^+$.

L'inclusione $G^+ \subseteq F^+$ è banalmente verificata in quanto, per definizione, $G \subseteq F^+$.



$F^+ = G^+ \quad G = \text{unione } P_Ri(F)$

$F \text{ in } G^+ \text{ AND } G \text{ in } F^+$

$P_Ri(F) = \{X \rightarrow Y \mid X \rightarrow Y \text{ in } F^+ \text{ AND } XY \text{ in } Ri\}$

2 $R-A \rightarrow A$ porta a sottoschema R

3 OGNI $X \rightarrow A$ in F porta a sottoschema XA

Teorema

ricordiamo che **gli attributi in S** sono quelli che non sono coinvolti nelle dipendenze, e siccome la chiave deve determinare **tutto** lo schema, **dovranno essere necessariamente nella chiave che li determinerà per riflessività**



SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

- Ogni schema di relazione in ρ è in 3NF.

Analizziamo i diversi casi che si possono presentare

1. Se $S \in \rho$, ogni attributo in S fa parte della chiave e quindi, banalmente, S è in 3NF.

o in quello che ne rimane dopo aver tolto S

2. Se $R \in \rho$ **esiste** una dipendenza funzionale in F che coinvolge tutti gli attributi in R . Poiché F è una copertura minimale tale dipendenza avrà la forma $R-A \rightarrow A$; poiché F è una copertura minimale, non ci può essere una dipendenza funzionale $X \rightarrow A$ in F^+ tale che $X \subset R-A$ e, quindi, $R-A$ è chiave in R . Sia $Y \rightarrow B$ una qualsiasi dipendenza in F ; se $B=A$ allora, poiché F è una copertura minimale, $Y=R-A$ (cioè, Y è una superchiave); se $B \neq A$ allora $B \in R-A$ e quindi B è primo.

$R-A$ è una chiave perché determina **TUTTO** lo schema $((R-A) \cup A)$ e **nessun** sottoinsieme ha la stessa proprietà; ricordiamo che siccome abbiamo trovato $R-A \rightarrow A$ l'algoritmo si è fermato, ma nella copertura minimale potrebbero esserci altre dipendenze

3. Se $XA \in \rho$, poiché F è una copertura minimale, non ci può essere una dipendenza funzionale $X' \rightarrow A$ in F^+ tale che $X' \subset X$ e, quindi, X è chiave in XA . Sia $Y \rightarrow B$ una qualsiasi dipendenza in F tale che $YB \subseteq XA$; se $B=A$ allora, poiché F è una copertura minimale, $Y=X$ (cioè, Y è una superchiave); se $B \neq A$ allora $B \in X$ e quindi B è primo.

- Nota: Possiamo avere 1+2 (R residuo), oppure 1+3, oppure solo 3

le parti
destra
sono
singleton



1. $S \neq \text{vuoto}$ implica tutti attrib S primi

2. $R-A \rightarrow A$ sottoschema R con superchiave R-A

R-A è anche chiave? Esiste X in R-A tale che $X \rightarrow R$?

F cop minimale NON ESISTE $X \rightarrow A$ in F^+ con X in R-A

Quindi R-A chiave!

$Y \rightarrow B$ in F rispetta 3NF? Con YB in R

$B=A$ Y non può essere diverso da R-A oppure

$B \neq A$ allora B in R-A (chiave) allora B primo!



3. $X \rightarrow A$ sottoschema XA con superchiave X

X chiave? Esiste X' in X tale che $X' \rightarrow XA$?

Se esistesse $X' \rightarrow A$ in F^+ allora F non sarebbe minimale

X' non determina A e quindi non può determinare XA

X chiave per XA

$Y \rightarrow B$ in F con YB in XA

Se $B=A$ Y deve essere X (F cop copertura minimale)

Se $B \neq A$ allora B deve essere in X e quindi primo



- E per avere anche un join senza perdita?
- **Basta aggiungere un sottoschema contenente una chiave** al risultato dell'algoritmo di decomposizione

Teorema Sia R uno schema di relazione, F un insieme di dipendenze funzionali su R , che è una copertura minimale e ρ la decomposizione di R prodotta dall'Algoritmo di decomposizione. La decomposizione $\sigma = \rho \cup \{K\}$, dove K è una chiave per R , è tale che:

- ogni schema di relazione in σ è in 3NF
- σ preserva F
- σ ha un join senza perdita.

Dim.

- **σ preserva F .** Poiché ρ preserva F anche σ preserva F .

Stiamo aggiungendo un nuovo sottoschema, quindi alla nuova G' dobbiamo aggiungere una proiezione di F , cioè $G' = G \cup \pi_K(F)$ quindi $G' \supseteq G \supseteq F$ e quindi $G'^+ \supseteq G^+ \supseteq F^+$. L'inclusione $G'^+ \subseteq F^+$ è di nuovo banalmente verificata in quanto, **per definizione**, $G \subseteq F^+$.

- Ogni schema di relazione in σ è in 3NF.
- Poiché $\sigma = \rho \cup \{K\}$, è **sufficiente** verificare che **anche** lo schema di relazione K è in 3NF. Mostriamo che K è chiave **anche** per lo schema K . Supponiamo per **assurdo** che K **non sia** chiave per lo schema K ; allora esiste un **sottoinsieme** proprio K' di K che determina tutto lo schema K , cioè tale che $K' \rightarrow K \in F^+$ (più precisamente alla chiusura di $\pi_K(F)$, ma poiché $\pi_K(F) \subset F^+$ allora $(\pi_K(F))^+ \subset F^+$); poiché K è chiave per lo schema R , $K \rightarrow R \in F^+$; pertanto per transitività $K' \rightarrow R \in F^+$, che contraddice il fatto che **K è chiave per lo schema R** . (verrebbe violato il requisito di **minimalità**) Pertanto, K è chiave per lo schema K e quindi per ogni dipendenza funzionale $X \rightarrow A$ in F^+ con $XA \subseteq K$, A è primo.

- σ ha un join senza perdita.

Supponiamo che l'ordine in cui gli attributi in $R-K$ vengono aggiunti a Z dall'Algoritmo che calcola la chiusura di un insieme di attributi (in questo caso K^+) sia A_1, A_2, \dots, A_n , e supponiamo che per ogni $i, i=1, \dots, n$, l'attributo A_i venga aggiunto a Z a causa della presenza in F (**copertura minimale!**) della dipendenza $Y_i \rightarrow A_i$ (quindi avremo anche un sottoschema $Y_i \rightarrow A_i$) dove:

$$Y_i \subseteq Z^{(i-1)} = KA_1A_2 \dots A_{i-1} \subseteq K^+.$$

ricordiamo che in $Z^{(i-1)}$ ci sono gli attributi aggiunti a Z fino all'iterazione i

- Per dimostrare che σ ha un join senza perdita mostreremo che quando l'Algoritmo per la verifica del join senza perdita è applicato a σ viene prodotta una tabella che ha una riga con tutte 'a'.
- Senza perdita di generalità, supponiamo che l'Algoritmo che verifica il join senza perdita **esamini** le dipendenze funzionali $Y_1 \rightarrow A_1, Y_2 \rightarrow A_2, \dots, Y_n \rightarrow A_n$ **in questo ordine**. Dimostreremo per induzione su i che **dopo che è stata considerata la dipendenza funzionale $Y_i \rightarrow A_i$** nella **riga** che corrisponde allo schema di relazione K c'è una 'a' in **ogni colonna j con $j \leq i$** .

- *Base dell'induzione: $i=1$.* Poiché $Y_1 \subseteq Z^{(0)} = K$, sia nella riga che corrisponde allo schema di relazione $Y_1 A_1$ (per costruzione) che in quella che corrisponde allo schema di relazione K (perché $Y_1 \subseteq K$) ci sono tutte 'a' in corrispondenza degli attributi in Y_1 ; inoltre nella riga che corrisponde allo schema di relazione $Y_1 A_1$ c'è **una 'a' in corrispondenza ad A_1** . (è un attributo dello schema) Pertanto l'Algoritmo pone una 'a' in corrispondenza ad A_1 nella riga che corrisponde allo schema di relazione K per fare in modo che venga soddisfatta $Y_1 \rightarrow A_1$
- *Induzione: $i > 1$.* Per l'ipotesi induttiva, nella riga che corrisponde allo schema di relazione K c'è una 'a' in corrispondenza di **ogni attributo A_j con $j \leq i-1$** .
- Poiché $Y_i \subseteq K A_1 A_2 \dots A_{i-1}$, sia nella riga che corrisponde **allo schema di relazione $Y_i A_i$** , che in quella che corrisponde **allo schema di relazione K** ci sono tutte 'a' in corrispondenza agli attributi in Y_i ; (per la costruzione iniziale della riga corrispondente a K , per l'ipotesi induttiva e perché Y_i è contenuto in questo insieme) inoltre nella riga che corrisponde allo schema di relazione $Y_i A_i$ **c'è una 'a' in corrispondenza ad A_i** (fa parte dello schema). Pertanto l'Algoritmo pone una 'a' in corrispondenza ad A_i nella riga che corrisponde allo schema di relazione K .

Ci sono schemi di relazione che non sono “buoni”

Sono quelli in cui sono rappresentati più concetti come lo schema **Curriculum** (Matr, CF, Cogn, Nome, DataN, Com, Prov, C#, Tit, Doc, DataE, Voto)

in quanto presentano ridondanza e anomalie di aggiornamento, inserimento e cancellazione.

DOMANDA 1:

È possibile formalizzare il concetto di schema relazionale “buono”?

DOMANDA 2:

È sempre possibile rappresentare la realtà di interesse con uno schema di BD in cui ogni schema di relazione sia “buono”?



Si!

Uno schema è buono se è in Terza
Forma Normale (3NF)



È sempre possibile rappresentare la realtà di interesse con uno schema di BD in cui ogni schema di relazione sia in 3NF?

Lo schema di BD {**Studente, Esame, Corso, Comune**} in cui ogni schema di relazione è in 3NF, **può essere ottenuto dallo schema di relazione** (non in 3NF) **Curriculum mediante** un procedimento di **decomposizione**

DOMANDA 3:

Qualsiasi schema di BD tale che:

- è ottenuto mediante decomposizione
- ogni schema di relazione è in 3NF

rappresenta adeguatamente la realtà di interesse?



NO!

La decomposizione potrebbe **non** permettere di rappresentare:

- tutte le dipendenze funzionali definite sullo schema di relazione originario
- l'informazione rappresentabile mediante lo schema di relazione originario

Esempio di decomposizione che non preserva le dipendenze



SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Studente = Matr Comune Provincia

$F = \{ \text{Matr} \rightarrow \text{Comune}, \text{Matr} \rightarrow \text{Provincia}, \text{Comune} \rightarrow \text{Provincia} \}$

$\rho = \{R1, R2\}$

$R1 = \text{Matr Comune}$

$R2 = \text{Matr Provincia}$

$F1 = \{ \text{Matr} \rightarrow \text{Comune} \}$

$F2 = \{ \text{Matr} \rightarrow \text{Provincia} \}$

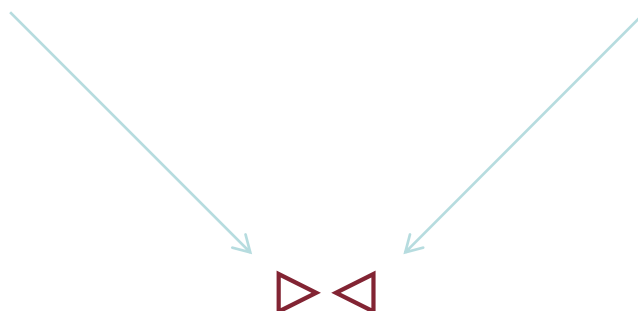


Istanza
legale

Matr	Comune
01	Marino
02	Marino

Matr	Provincia
01	Roma
02	Latina

Istanza
legale



Matr	Comune	Provincia
01	Marino	Roma
02	Marino	Latina

Esempio di decomposizione che perde informazione



SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Ordine = Cliente Articolo Data

$F = \emptyset$

$\rho = \{R1, R2\}$

$R1 = \text{Cliente Articolo}$

$F1 = \emptyset$

$R2 = \text{Articolo Data}$

$F2 = \emptyset$

Cliente	Articolo	Data
C1	A1	01/03/2013
C2	A1	12/03/2013

π Cliente Articolo

π Articolo Data

Client e	Articol o
C1	A1
C2	A1

Articolo	Data
A1	01/03/2013
A1	12/11/2013



Cliente	Articolo	Data
C1	A1	01/03/2013
C1	A1	12/11/2013
C2	A1	01/03/2013
C2	A1	12/11/2013



È sempre possibile rappresentare la realtà di interesse con uno schema di BD, ottenuto per **decomposizione** di uno schema di relazione, in cui:

- **ogni schema di relazione sia in 3NF**
- tutte le dipendenze funzionali definite sullo schema di relazione originario siano preservate
(la decomposizione preserva F)
- l'informazione rappresentabile mediante lo schema di relazione originario non venga persa
(la decomposizione ha un join senza perdita) ?



SI!

Esiste un algoritmo polinomiale che, dati uno schema di relazione R e un insieme di dipendenze funzionali F su R , fornisce una decomposizione di R tale che:

- ogni schema di relazione nella decomposizione è in 3NF
- la decomposizione preserva F
- la decomposizione ha un join senza perdita.