

Lezione 13 - Decomposizioni che preservano le dipendenze

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



SAPIENZA
UNIVERSITÀ DI ROMA

Cosa significa preservare le dipendenze



- Quando uno schema di relazione viene di solito decomposto:
- **quando non è in 3NF**
- per motivi di efficienza degli accessi
 - più piccola è la taglia delle tuple maggiore è il numero che riusciamo a caricare in memoria nella stessa operazione di lettura
 - se le informazioni della tupla non vengono utilizzate dallo stesso tipo di operazioni nella base di dati meglio decomporre lo schema
 - Esempio: lo schema *Studiante* potrebbe essere decomposto separando le informazioni anagrafiche (CF, Nome, Cognome, DataNascita, LuogoNascita, ecc.) da quelle accademiche (Matricola, CorsoLaurea, AnnoCorso, ecc.)

Cosa significa preservare le dipendenze



SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

- Abbiamo visto che quando uno schema viene decomposto, **non basta** che i sottoschemi siano in 3NF
- Rivediamo i due esempi (uno più «astratto», l'altro più «concreto»)

Cosa vogliamo ottenere – La 3NF non basta



- Uno schema che non è in 3NF può essere **decomposto** in **più modi** in un insieme di schemi in 3NF. Ad esempio lo schema $R=ABC$ con l'insieme di dipendenze funzionali $F=\{A \rightarrow B, B \rightarrow C\}$ non è in 3NF per la presenza in F^+ della dipendenza transitiva $B \rightarrow C$, dato che la chiave è evidentemente A .
- R può essere decomposto in:
 - $R1=AB$ con $\{A \rightarrow B\}$ e
 - $R2=BC$ con $\{B \rightarrow C\}$
- Oppure
 - $R1=AB$ con $\{A \rightarrow B\}$ e
 - $R2=AC$ con $\{A \rightarrow C\}$
- Entrambi gli schemi **sono** in 3NF, tuttavia la seconda soluzione **non è soddisfacente**.

Cosa vogliamo ottenere – La 3NF non basta



- Consideriamo le istanze **legali** degli schemi ottenuti

R1	A	B
	a1	b1
	a2	b1

R2	A	C
	a1	c1
	a2	c2

- L'istanza dello schema originario R che posso ricostruire da questa (l'unico modo è di ricostruirla facendo un join naturale!) è la seguente

R	A	B	C
	a1	b1	c1
	a2	b1	c2

- MA** non è un'istanza legale di R , in quanto **non soddisfa** la dipendenza funzionale $B \rightarrow C$ (che sarà pure transitiva **ma va soddisfatta comunque!**)
- Occorre preservare TUTTE le dipendenze in F^+**

- Consideriamo lo schema $R=(Matricola, Comune, Provincia)$ con l'insieme di dipendenze funzionali

$$F=\{Matricola \rightarrow Comune, Comune \rightarrow Provincia\}$$

- Lo schema non è in 3NF per la presenza in F^+ della dipendenza transitiva $Comune \rightarrow Provincia$, dato che la chiave è evidentemente $Matricola$ (*Provincia dipende transitivamente da Matricola*).
- R può essere decomposto in:

$$R1=(Matricola, Comune) \text{ con } \{Matricola \rightarrow Comune\}$$

$$R2=(Comune, Provincia) \text{ con } \{Comune \rightarrow Provincia\}$$

- Oppure**

$$R1=(Matricola, Comune) \text{ con } \{Matricola \rightarrow Comune\}$$

$$R2=(Matricola, Provincia) \text{ con } \{Matricola \rightarrow Provincia\}$$

- Entrambi gli schemi **sono** in 3NF, tuttavia la seconda soluzione **non è soddisfacente**.

Cosa vogliamo ottenere – La 3NF non basta



- Consideriamo le istanze **legali** degli schemi ottenuti

R1	Matricola	Comune	R2	Matricola	Provincia
	501	Tivoli		501	Roma
	502	Tivoli		502	Rieti

- L'istanza dello schema originario R che posso ricostruire da questa (l'unico modo è di ricostruirla facendo un join naturale!) è la seguente

R	Matricola	Comune	Provincia
	501	Tivoli	Roma
	502	Tivoli	Rieti

- MA** non è un'istanza legale di R , in quanto **non soddisfa** la dipendenza funzionale $Comune \rightarrow Provincia$ (che sarà pure transitiva **ma possiamo rivelare praticamente che va soddisfatta comunque!**)
- E' evidente che c'è stato un errore di inserimento, ma non abbiamo potuto rilevarlo**

- Per formalizzare il concetto di decomposizione che “preserva un insieme di dipendenze funzionali”, cominciamo con il riprendere la definizione di **decomposizione di uno schema di relazione**

Definizione Sia R uno schema di relazione. Una decomposizione di R è una famiglia $\rho = \{R_1, R_2, \dots, R_k\}$ di sottoinsiemi di R che ricopre R ($\bigcup_{i=1}^k R_i = R$) (i sottoinsiemi possono avere intersezione non vuota)

- **In altre parole:** se lo schema R è composto da un certo insieme di attributi, decomporlo significa definire dei **sottoschemi** che contengono **ognuno un sottoinsieme** degli attributi di R . I sottoschemi **possono** avere attributi **in comune**, e la loro **unione** deve **necessariamente** contenere **tutti** gli attributi di R .
- **Quindi:** R è un insieme di attributi, una decomposizione di R è una famiglia di insiemi di attributi

- Per lo schema $R=(CF, Nome, Cognome, Matricola, Datan, Luogon, CorsoLaurea, Anno)$ possiamo avere le decomposizioni
 - $\rho=\{ R1=(CF, Matricola, Nome, Cognome, Datan, Luogon), R2=(Matricola, CF, Nome, Cognome, CorsoLaurea, Anno) \}$
 - oppure in
 - $\rho=\{ R1=(CF, Matricola, Nome, Cognome, Datan, Luogon), R2=(Matricola, CorsoLaurea, Anno) \}$
- In entrambi i casi le famiglie di sottoinsiemi di R (sottoschemi) ricoprono lo schema ... ma poi occorre verificare se entrambe le decomposizioni sono «buone», cioè se effettivamente lo schema può essere decomposto in entrambi i modi preservando la 3NF dei sottoschemi, le dipendenze funzionali e fornendo un join senza perdita



- «Decomporre» una **istanza** di una relazione **con un certo schema**, in **base alla decomposizione dello schema stesso**, significa proiettare ogni tupla dell'istanza originaria sugli attributi dei singoli sottoschemi ...
- ...**eliminando i duplicati** che potrebbero essere generati dal fatto che due tuple sono distinte **ma hanno una porzione comune** che ricade nello stesso sottoschema

Esempio



R

CF	Nome	Cognome	Matricola	Datan	Luogon	CorsoLaurea	Anno
DDD	Davide	Bigi	1111	12-12-90	Bari	Lettere	3
FFF	Gianni	Neri	1212	15-09-91	Milano	Legge	2
AAA	Antonio	Rossi	1313	09-08-92	Napoli	Matematica	1

R₁

CF	Nome	Cognome	Matricola	Datan	Luogon
DDD	Davide	Bigi	1111	12-12-90	Bari
FFF	Gianni	Neri	1212	15-09-91	Milano
AAA	Antonio	Rossi	1313	09-08-92	Napoli

sottoschema della
decomposizione di R

proiezione dell'istanza sul
sottoschema

R₂

CF	Nome	Cognome	Matricola	CorsoLaurea	Anno
DDD	Davide	Bigi	1111	Lettere	3
FFF	Gianni	Neri	1212	Legge	2
AAA	Antonio	Rossi	1313	Matematica	1

sottoschema della
decomposizione di R

proiezione dell'istanza sul
sottoschema

- Continuiamo con la definizione di **equivalenza tra due insiemi di dipendenze funzionali**.
- **Definizione** Siano F e G due insiemi di dipendenze funzionali. F e G sono *equivalenti* ($F \equiv G$) se $F^+ = G^+$. (**F e G NON contengono le stesse dipendenze, ma le loro chiusure SI**)

- **Verificare l'equivalenza** di due insiemi F e G di dipendenze funzionali richiede dunque che venga verificata **l'uguaglianza di F^+ e G^+** , cioè che $F^+ \subseteq G^+$ e che $F^+ \supseteq G^+$.
- Ricordiamo infatti che sia F e G che F^+ e G^+ sono **insiemi di dipendenze** \rightarrow uguaglianza \Leftrightarrow contenimento nei due versi
- Come detto in precedenza, calcolare la chiusura di un insieme di dipendenze funzionali richiede **tempo esponenziale**. Il seguente **lemma** ci permette tuttavia di verificare l'equivalenza dei due insiemi di dipendenze funzionali **in tempo polinomiale**.

- **Lemma** Siano F e G due insiemi di dipendenze funzionali. Se $F \subseteq G^+$ allora $F^+ \subseteq G^+$.
NOTA: Vale per qualunque coppia di insiemi di dipendenze!
- **Dim.** Sia $f \in F^+ - F$ (è una dipendenza in F^+ che non compare in F).
- **ogni dipendenza funzionale in F è derivabile da G** mediante gli assiomi di Armstrong (per l'ipotesi si trova in G^+ e il Teorema che abbiamo dimostrato afferma che $F^+ = F^A$)
- sempre per il Teorema che dimostra che $F^+ = F^A$, f in F^+ è **derivabile dalle dipendenze in F mediante gli assiomi di Armstrong**
OVVIAMENTE il fatto che l'insieme abbia un nome diverso non ne cambia le proprietà ... visto che sia F che G sono insiemi di dipendenze funzionali!
- f è derivabile da G mediante gli assiomi di Armstrong.

$$G \xrightarrow{A} F \xrightarrow{A} F^+$$

con \xrightarrow{A} denotiamo la derivazione
tramite gli assiomi di Armstrong

Finalmente la definizione formale che ci interessa.

- **Definizione** Sia R uno schema di relazione, F un insieme di dipendenze funzionali su R e $\rho = \{R_1, R_2, \dots, R_k\}$ una decomposizione di R .
- Diciamo che ρ **preserva** F se $F \equiv \bigcup_{i=1}^k \pi_{R_i}(F)$,
dove $\pi_{R_i}(F) = \{X \rightarrow Y / X \rightarrow Y \in F^+ \wedge XY \subseteq R_i\}$.
- **ATTENZIONE!**
- ovviamente $\bigcup_{i=1}^k \pi_{R_i}(F)$ è un insieme di dipendenze funzionali
- ogni $\pi_{R_i}(F)$ è un insieme di dipendenze funzionali dato dalla proiezione dell'insieme di dipendenze funzionali F sul sottoschema R_i
- proiettare un insieme di dipendenze F su un sottoschema R_i **non significa** banalmente prendere le dipendenze dell'insieme F ed **eliminare** da queste dipendenze gli attributi che **non sono** in R_i ...
- ... ma **prendere tutte e sole** le dipendenze **derivabili da F** tramite gli assiomi di Armstrong (quindi quelle in F^+) che hanno **tutti gli attributi** (dipendenti e determinanti) **in R_i**

- Supponiamo di **avere già** una **decomposizione** e di voler **verificare se preserva le dipendenze funzionali**.
- Verificare **se** una decomposizione preserva un insieme di dipendenze funzionali F richiede che venga verificata l'**equivalenza** dei due insiemi di dipendenze funzionali F e $G = \bigcup_{i=1}^k \pi_{R_i}(F)$ e quindi la doppia inclusione $F^+ \subseteq G^+$ e che $F^+ \supseteq G^+$.
- **Nota: per come è stato definito G in questo caso**, sarà sicuramente $F^+ \supseteq G$
- Infatti $G = \bigcup_{i=1}^k \pi_{R_i}(F)$, dove $\pi_{R_i}(F) = \{X \rightarrow Y / X \rightarrow Y \in F^+ \wedge XY \subseteq R_i\}$
- Ogni **proiezione di F** che viene **inclusa per definizione in G** è un **sottoinsieme di F^+** , quindi **F^+ contiene G** (che **ovviamente** può anche essere scritto come $G \subseteq F^+$) e per il lemma sulle chiusure questo implica che $G^+ \subseteq F^+$ (che **ovviamente** può anche essere scritto come $F^+ \supseteq G^+$)
- Per il Lemma sulle chiusure è sufficiente quindi verificare che $F \subseteq G^+$ (che poi implica $F^+ \subseteq G^+$)

La verifica che $F \subseteq G^+$ (che poi implica che $F^+ \subseteq G^+$) può essere fatta con l'algoritmo che segue (la cui correttezza è una banale conseguenza del Lemma sulla chiusura di un insieme di attributi e del Teorema sull'uguaglianza $F^+ = F^A$).

Algoritmo - contenimento di F in G^+

Input due insiemi F e G di dipendenze funzionali su R ;

Output la variabile *successo* (al termine avrà valore **true** se $F \subseteq G^+$, **false** altrimenti)

begin

successo := true;

 for every $X \rightarrow Y \in F$

 do begin

 calcola X_G^+ ;

 if $Y \notin X_G^+$ then *successo* := false

 end

end

Lemma Siano R uno schema di relazione ed F un insieme di dipendenze funzionali su R . Si ha che: $X \rightarrow Y \in F^A$ se e solo se $Y \subseteq X^+$.

Teorema Siano R uno schema di relazione ed F un insieme di dipendenze funzionali su R . Si ha $F^+ = F^A$.

- **Se** $Y \notin X_G^+$ allora $X \rightarrow Y \notin G^A$ per il lemma e quindi $X \rightarrow Y \notin G^+$ per il Teorema
- Basta verificare che **anche una sola dipendenza** non appartiene alla chiusura di G per poter affermare che l'**equivalenza non sussiste**

- Come calcoliamo X_G^+ ?
- Se volessimo utilizzare l' Algoritmo già visto per il calcolo della chiusura di un insieme di attributi, dovremmo **prima calcolare G ...**
- ... ma, **per la definizione di G , ciò richiede il calcolo di F^+ che richiede tempo esponenziale.**
- Presentiamo un algoritmo che permette di calcolare X_G^+ a **partire da F .**

Algoritmo calcolo X_G^+ a partire da F .



SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Algoritmo - X_G^+ a partire da F .

Input uno schema R , un insieme F di dipendenze funzionali su R , una decomposizione $\rho = \{R_1, R_2, \dots, R_k\}$ di R , un sottoinsieme X di R ;

Output la chiusura di X rispetto a $G = \bigcup_{i=1}^k \pi_{R_i}(F)$, (nella variabile Z);

begin

$Z := X$;

$S := \emptyset$;

for $i := 1$ to k

do $S := S \cup (Z \cap R_i)^+_{F \cap R_i}$

while $S \not\subseteq Z$

do

begin

$Z := Z \cup S$;

for $i := 1$ to k

do $S := S \cup (Z \cap R_i)^+_{F \cap R_i}$

end

end

Attenzione!!!

L'intersezione ha **priorità maggiore** dell'unione, quindi $Z \cap R_i)^+_{F \cap R_i}$ va calcolato **prima dell'unione con S**. Se si **inverte l'ordine delle operazioni** da S potremmo eliminare ciò che non rientra in R_i (perché è stato inserito in un passaggio precedente grazie alla proiezione su un sottoschema senza intersezioni con R_i) ma questo non avrebbe senso perché in S stiamo accumulando gli attributi che sono determinati funzionalmente da X anche se appartengono a sottoschemi diversi

Notiamo la relazione tra la definizione delle proiezioni di F

$$\pi_{R_i}(F) = \{X \rightarrow Y / X \rightarrow Y \in F^+ \wedge XY \subseteq R_i\}$$

e il passo di aggiornamento di S dove vengono aggiunti gli attributi che appartengono a

$$(Z \cap R_i)^+ \cap R_i$$

Stiamo raccogliendo gli attributi determinati funzionalmente **dalla parte di Z** che interseca R_i , (per il lemma, $A \in X^+$ se e solo se $X \rightarrow A \in F^+$, **e in questo caso $X = Z \cap R_i$**) in base alle dipendenze in F^+ , e da questi selezioniamo quelli contenuti comunque in R_i (con l'intersezione $\cap R_i$)

Partendo da un insieme X , stiamo «ricostruendo» la sua **chiusura** dalle proiezioni di F che **compongono G** , e da lì iterando (ciclo while) dall'insieme G^+ (tramite la transitività). Nel fare questo attraverso l'intersezione con R_i ci assicuriamo che la dipendenza sia valida nel singolo sottoschema.

- Notiamo la differenza col passo di aggiornamento della S nel calcolo della chiusura di X rispetto ad F e ad un singolo schema

$$S := \{A/Y \rightarrow V \in F \wedge A \in V \wedge Y \subseteq Z\}$$

$$S := S \cup (Z \cap R_i)^+_{F \cap R_i}$$

- qui l'unione è necessaria perché **stiamo considerando a turno i diversi sottoschemi**
- inoltre prendiamo le dipendenze nella chiusura F^+ ($Z \cap R_i \rightarrow A \in F^+$ e ... ricordiamo il lemma) perché **le dipendenze nell'insieme G che ci interessa** sono incluse nelle proiezioni di F sui sottoschemi, che includono però dipendenze in F^+
- In conclusione avremo gli attributi che dipendono funzionalmente da X ... anche se appartengono a sottoschemi in cui X non è incluso ... perché dipendono da attributi che sono nello stesso sottoschema di X e dipendono da X , ma si trovano anche in altri sottoschemi!

- L' algoritmo (per definizione di algoritmo ...) **termina sempre!**
- Il fatto che l'algoritmo termini **non indica** che una dipendenza $X \rightarrow Y$ è **preservata!**
- Per **verificare se** $X \rightarrow Y$ è preservata, in base al Lemma sulla chiusura di un insieme di attributi e in base al Teorema sull'uguaglianza $F^+ = F^A$, **dobbiamo controllare SE Y è contenuto nella copia finale della variabile Z (che conterrà la chiusura di X rispetto a G , X_G^+)**

- **Teorema** Sia R uno schema di relazione, F un insieme di dipendenze funzionali su R , $\rho = \{R_1, R_2, \dots, R_k\}$ una decomposizione di R e X un sottoinsieme di R .
L' Algoritmo dato calcola correttamente X^+_G , dove $G = \bigcup_{i=1}^k \pi_{R_i}(F)$.
- **Dim.** Indichiamo con $Z^{(0)}$ il valore iniziale di Z ($Z^{(0)} = X$) e con $Z^{(i)}$, $i \geq 1$, il valore di Z **dopo** l' i -esima esecuzione dell' assegnazione $Z := Z \cup S$; è facile vedere che $Z^{(i)} \subseteq Z^{(i+1)}$, per ogni i . Sia $Z^{(f)}$ il valore di Z quando l' algoritmo termina; proveremo che:

$$A \in Z^{(f)} \text{ se e solo se } A \in X^+_G.$$

- **Parte solo se.** Mostriamo per induzione su i che $Z^{(i)} \subseteq X_G^+$, per **ogni** i (e in particolare per $i=f$)

qualunque sia l'insieme di dipendenze su R
- *Base dell'induzione:* $i=0$. Poiché $Z^{(0)}=X$ e $X \subseteq X_G^+$, si ha $Z^{(0)} \subseteq X_G^+$.
- *Induzione:* $i>0$.

È stato aggiunto **proprio** durante la i -esima iterazione perché **non era** in $Z^{(i-1)}$
- Per l'ipotesi induttiva $Z^{(i-1)} \subseteq X_G^+$.
- Sia A un attributo in $Z^{(i)} - Z^{(i-1)}$

A è nell'intersezione, cioè appartiene sia a R_j che a $(Z^{(i-1)} \cap R_j)^+_F$
- in tal caso **deve** esistere un indice j (della decomposizione) tale che $A \in (Z^{(i-1)} \cap R_j)^+_F \cap R_j$. Poiché $A \in (Z^{(i-1)} \cap R_j)^+_F$ si ha $(Z^{(i-1)} \cap R_j) \rightarrow A \in F^+$ (per il lemma sulla chiusura di un insieme di attributi e il Teorema $F^+ = F^A$). Poiché $(Z^{(i-1)} \cap R_j) \rightarrow A \in F^+$, $A \in R_j$ e $Z^{(i-1)} \cap R_j \subseteq R_j$ si ha, per la definizione di G , che $(Z^{(i-1)} \cap R_j) \rightarrow A \in G$. Poiché per l'ipotesi induttiva si ha che $X \rightarrow Z^{(i-1)} \in G^+$, per la regola di decomposizione si ha anche che $X \rightarrow (Z^{(i-1)} \cap R_j) \in G^+$ e, quindi, per l'assioma della transitività, che $X \rightarrow A \in G^+$, cioè $A \in X_G^+$. Quindi $Z^{(i)} \subseteq X_G^+$.

Gli attributi in $Z^{(i-1)}$ ci sono per ipotesi induttiva e abbiamo mostrato che ci vanno anche quelli inseriti in Z all' i -esima iterazione del ciclo



- **Parte se.** Vedi dispensa associata al corso.