

Corso di Laurea in Ingegneria
Informatica
Fondamenti di Informatica II
Modulo "*Basi di dati*"
a.a. 2017-2018

Docente: Gigliola Vaglini
Docente di laboratorio: Francesco
Pistolesi

1

- Abbiamo messo delle ridondanze, ma cosa significa?
- Città con n.di abitanti ok
- Codicefattura - Lordo - iva - netto

2

Lezione 6

Dipendenze funzionali
Relazioni in forma normale

3

Forme normali

- La forma normale è una proprietà che garantisce la "qualità" di una base di dati relazionale, cioè l'assenza di determinati difetti
 - Ad esempio, una relazione non normalizzata presenta ridondanze e anomalie

4

Normalizzazione

- La normalizzazione è la procedura che permette di portare schemi relazionali in forma normale
- La normalizzazione è utilizzata come tecnica di verifica dei risultati della progettazione, non costituisce una metodologia di progetto

5

Una relazione con anomalie

<u>Impiegato</u>	Stipendio	<u>Progetto</u>	Bilancio	Funzione
Rossi	20	Marte	2	tecnico
Verdi	35	Giove	15	progettista
Verdi	35	Venere	15	progettista
Neri	55	Venere	15	direttore
Neri	55	Giove	15	consulente
Neri	55	Marte	2	consulente
Mori	48	Marte	2	direttore
Mori	48	Venere	15	progettista
Bianchi	48	Venere	15	progettista
Bianchi	48	Giove	15	direttore

6

Proprietà

- Ogni impiegato ha un solo stipendio (anche se partecipa a più progetti)
- Ogni progetto ha un bilancio
- Ogni impiegato in ciascun progetto ha una sola funzione (anche se può avere funzioni diverse in progetti diversi)

7

Anomalie

- Lo stipendio di ciascun impiegato è ripetuto in tutte le ennuple relative
 - ridondanza
- Se lo stipendio di un impiegato varia, è necessario andarne a modificare il valore in diverse ennuple
 - anomalia di aggiornamento
- Se un impiegato si licenzia, dobbiamo cancellarlo in diverse ennuple
 - anomalia di cancellazione

8

Perché questi fenomeni?

- Un'unica relazione per rappresentare informazioni eterogenee
 - gli impiegati con i relativi stipendi
 - i progetti con i relativi bilanci
 - le partecipazioni degli impiegati ai progetti con le relative funzioni

9

Per studiare in maniera
sistematica questi aspetti, è
necessario utilizzare il concetto di
dipendenza funzionale

10

9.1. Le dipendenze funzionali

11

Dipendenza funzionale

Si considerino

- la relazione r su $R(X)$
- due sottoinsiemi non vuoti Y e Z di X

esiste in r una dipendenza funzionale (FD) da Y a Z se, per ogni coppia di ennuple t_1 e t_2 di r con gli stessi valori su Y , risulta che t_1 e t_2 hanno gli stessi valori anche su Z

- ad ogni chiave K di R corrisponde una dipendenza funzionale in R da K verso tutti gli attributi della relazione

12

Notazione

$$X \rightarrow Y$$

- Esempi:

Impiegato \rightarrow Stipendio

Progetto \rightarrow Bilancio

Impiegato Progetto \rightarrow Funzione

13

Altre FD

- Impiegato Progetto \rightarrow Progetto
- Si tratta di una FD "banale" (sempre soddisfatta)
 - $Y \rightarrow A$ è non banale se A non appartiene a Y
 - $Y \rightarrow Z$ è non banale se nessun attributo in Z appartiene a Y

14

<u>Impiegato</u>	<u>Stipendio</u>	<u>Progetto</u>	<u>Bilancio</u>	<u>Funzione</u>
Rossi	20	Marte	2	tecnico
Verdi	35	Giove	15	progettista
Verdi	35	Venere	15	progettista
Neri	55	Venere	15	direttore
Neri	55	Giove	15	consulente
Neri	55	Marte	2	consulente
Mori	48	Marte	2	direttore
Mori	48	Venere	15	progettista
Bianchi	48	Venere	15	progettista
Bianchi	48	Giove	15	direttore

Impiegato → Stipendio

Progetto → Bilancio

Impiegato Progetto → Funzione

15

Alcune FD causano anomalie

- gli impiegati hanno un unico stipendio

Impiegato → Stipendio

- i progetti hanno un unico bilancio

Progetto → Bilancio

- Ma non tutte

Impiegato Progetto → Funzione

Come mai?

16

Legame tra FD e anomalie

Impiegato \rightarrow Stipendio

Progetto \rightarrow Bilancio

Impiegato Progetto \rightarrow Funzione

- Impiegato non e' una chiave
- Progetto non e' una chiave
- Impiegato Progetto è chiave
- La relazione contiene alcune informazioni legate alla chiave e altre ad attributi che non formano una chiave.

17

Ancora le FD

- *Implicazione*
- Sia F un insieme di dipendenze funzionali definite su $R(Z)$ e sia $X \rightarrow Y$ una delle dipendenze in F :
 - si dice che F implica $X \rightarrow Y$ ($F \Rightarrow X \rightarrow Y$) se, per ogni istanza r di R che verifica tutte le dipendenze in F , risulta verificata anche $X \rightarrow Y$
 - si dice anche che $X \rightarrow Y$ è implicata da F

18

FD (cont.)

- *Chiusura*
- Dato un insieme di dipendenze funzionali F definite su $R(Z)$, la chiusura di F è l'insieme di tutte le dipendenze funzionali implicate da F
- $F^+ = \{ X \rightarrow Y \mid F \Rightarrow X \rightarrow Y \}$
- Dato un insieme di dipendenze funzionali F definite su $R(Z)$, un'istanza r di R che soddisfa F , soddisfa anche F^+

19

FD (cont.)

- Dato $R(Z)$ ed un insieme F di FD, un insieme di attributi X appartenenti a Z si dice superchiave di R , se la dipendenza funzionale $X \rightarrow Z$ è logicamente implicata da F ($X \rightarrow Z$ è in F^+).
- Se nessun sottoinsieme proprio di X è superchiave di R , allora X si dice chiave di R

20

Calcolo di F^+

- La definizione di implicazione non è direttamente utilizzabile nella pratica, essa prevede, infatti, una quantificazione universale sulle istanze della base di dati ("per ogni istanza r "),
- Armstrong (1974) ha fornito delle regole di inferenza che permettono di derivare effettivamente tutte le dipendenze funzionali che sono implicate da un dato insieme iniziale
- tali regole sono *corrette e complete*, cioè permettono di ottenere tutte e sole le dipendenze in F^+

21

Regole di inferenza di Armstrong

1. Riflessività: Se $Y \subseteq X$, allora $X \rightarrow Y$
2. Additività (o espansione): Se $X \rightarrow Y$, allora $XZ \rightarrow YZ$, per qualunque Z
3. Transitività: Se $X \rightarrow Y$ e $Y \rightarrow Z$, allora $X \rightarrow Z$

22

Proprietà delle regole di Armstrong

- **Teorema** (correttezza): Le regole di inferenza di Armstrong sono corrette, cioè, applicandole ad un insieme F di dipendenze funzionali, si ottengono solo dipendenze logicamente implicate da F .
- **Teorema** (completezza): Le regole di inferenza di Armstrong sono complete, cioè, applicandole ad un insieme F di dipendenze funzionali, si ottengono tutte le dipendenze logicamente implicate da F .
- **Teorema** (minimalità): Le regole di inferenza di Armstrong sono minimali, cioè ignorando anche una sola di esse, l'insieme di regole che rimangono non è più completo.

23

Esempi di prove

- DIMOSTRARE che, \forall istanza di ogni relazione,

$$X \rightarrow Y \Rightarrow XZ \rightarrow YZ$$
- Supponiamo per assurdo che esista una istanza r di R in cui valga $X \rightarrow Y$ ma non $XZ \rightarrow YZ$,
 devono perciò esistere due tuple t_1 e t_2 di r tali che :
 (1) $t_1[X] = t_2[X]$, (2) $t_1[Y] = t_2[Y]$,
 (3) $t_1[XZ] = t_2[XZ]$, (4) $t_1[YZ] \neq t_2[YZ]$
 ma ciò è assurdo, poichè da (1) e (3) si deduce:
 (5) $t_1[Z] = t_2[Z]$,
 e da (2) e (5) si deduce :
 (6) $t_1[YZ] = t_2[YZ]$,
 in contraddizione con la (4)

24

(cont.)

- DIMOSTRARE che $X \rightarrow Y$ e $Y \rightarrow Z \Rightarrow X \rightarrow Z$
- Supponiamo per assurdo che esista una istanza r di R in cui valgano $X \rightarrow Y$ e $Y \rightarrow Z$, ma non $X \rightarrow Z$,
devono perciò esistere due tuple $t1$ e $t2$ in r tali che :
(1) $t1[X] = t2[X]$, (2) $t1[Y] = t2[Y]$,
(3) $t1[Z] = t2[Z]$, (4) $t1[Z] \neq t2[Z]$
ma ciò è assurdo

25

Regole derivate di Armstrong

4. Regola di unione
 $\{X \rightarrow Y, X \rightarrow Z\} \Rightarrow X \rightarrow YZ$
5. Regola di pseudotransitività (o aggiunta sinistra)
 $\{X \rightarrow Y, WY \rightarrow Z\} \Rightarrow XW \rightarrow Z$
6. Regola di decomposizione
Se $Z \subseteq Y, X \rightarrow Y \Rightarrow X \rightarrow Z$

26

Esempi di prove

- DIMOSTRARE che $X \rightarrow Y$ e $X \rightarrow Z \Rightarrow X \rightarrow YZ$
Per ipotesi valgono
 - a) $X \rightarrow Y$
 - b) $X \rightarrow Z$applicando la regola 2 ad (a) otteniamo
 - c) $XZ \rightarrow YZ$applicando la stessa regola a (b) otteniamo
 - $XX \rightarrow XZ$,che equivale a
 - d) $X \rightarrow XZ$per la regola 3 applicata a (d) e (c) otteniamo
 - $X \rightarrow YZ$

27

Esempio di calcolo di F^+

- Prendiamo le FD dell'esempio
 - i. $\text{Impiegato} \rightarrow \text{Stipendio}$
 - ii. $\text{Progetto} \rightarrow \text{Bilancio}$
 - iii. $\text{Impiegato Progetto} \rightarrow \text{Funzione}$
- E usiamo la regola 2 sulle dipendenze i e ii, otteniamo
 - iv. $\text{Impiegato Progetto} \rightarrow \text{Stipendio Progetto}$
 - v. $\text{Progetto Impiegato} \rightarrow \text{Bilancio Impiegato}$
- Di conseguenza
 - $\text{Impiegato Progetto} \rightarrow \text{Stipendio Progetto Impiegato Bilancio Funzione}$
- e quindi $\text{Impiegato Progetto}$ è chiave
- Ci sono altre FD in F^+ ?

28

Equivalenza

- Dato un insieme di dipendenze funzionali F è molto utile poter determinare un insieme di dipendenze funzionali G che sia equivalente ad F , ma sia anche strutturalmente più semplice
- F e G sono equivalenti se $F^+ = G^+$, ovvero, per ogni $X \rightarrow Y \in F$, deve essere $X \rightarrow Y \in G^+$ e, viceversa, per ogni $X \rightarrow Y \in G$, deve essere $X \rightarrow Y \in F^+$

29

Esempio 1

- $F = \{ A \rightarrow C, AC \rightarrow D, E \rightarrow AD, E \rightarrow H \}$
- $G = \{ A \rightarrow CD, E \rightarrow AH \}$

Verificare se F e G sono equivalenti

- Dimostro che le DF in F sono derivabili dalle DF in G , e viceversa
- $A \rightarrow CD \Rightarrow A \rightarrow C, A \rightarrow D$
- $A \rightarrow CD, CCD \rightarrow CD \Rightarrow AC \rightarrow CD \Rightarrow AC \rightarrow C, AC \rightarrow D$
- $E \rightarrow AH \Rightarrow E \rightarrow A, E \rightarrow H$
- $E \rightarrow A, A \rightarrow D \Rightarrow E \rightarrow D$
- $E \rightarrow A, E \rightarrow D \Rightarrow E \rightarrow AD$

30

(cont.)

- $A \rightarrow C, AC \rightarrow D \Rightarrow AA \rightarrow D \Rightarrow A \rightarrow D$
- $A \rightarrow C, A \rightarrow D \Rightarrow A \rightarrow CD$
- $E \rightarrow AD \Rightarrow E \rightarrow A, E \rightarrow D$
- $E \rightarrow A, E \rightarrow H \Rightarrow E \rightarrow AH$

31

Ovvero

- Il calcolo di F^+ è molto costoso (esponenziale nel numero di attributi dello schema nel caso peggiore),
- spesso quello che ci interessa è verificare se F^+ contiene una certa dipendenza
- alternativamente si può calcolare e utilizzare la chiusura transitiva di un insieme di attributi X (meno costoso ?), infatti
 - si può dimostrare che $X \rightarrow Y$ è in F^+ sse $Y \subseteq X^+$

32

Algoritmo per il calcolo di X^+

- Denotiamo con X^+ l'insieme degli attributi di $R(Z)$ che dipendono da X (chiusura di X) secondo F ;
- calcolare X^+ è semplice (complessità?)
 - $\text{CalcolaChiusura}(X, F) =$
 - { $X^+ = X$;
 - Ripeti:
 - Fine = true;
 - Per tutte le FD in $F = \{V_i \rightarrow W_i\}$:
 - Se $V_i \subseteq X^+$ e $W_i \not\subseteq X^+$ allora: $\{X^+ = X^+ \cup W_i$; Fine = false}
 - Fino a che Fine = true o $X^+ = Z$
 - }

33

Esempio 2

- Supponiamo di avere
 $F = \{A \rightarrow B, BC \rightarrow D, B \rightarrow E, E \rightarrow C\}$
e calcoliamo A^+ , ovvero l'insieme di attributi che dipendono da A
 - $A^+ = A$
 - $A^+ = AB$ poiché $A \rightarrow B$ e $A \subseteq A^+$
 - $A^+ = ABE$ poiché $B \rightarrow E$ e $B \subseteq A^+$
 - $A^+ = ABEC$ poiché $E \rightarrow C$ e $E \subseteq A^+$
 - $A^+ = ABECD$ poiché $BC \rightarrow D$ e $BC \subseteq A^+$
- Quindi da A dipendono tutti gli attributi dello schema, ovvero A è superchiave (e anche chiave)!

34

Esempio 1 (cont)

- $F = \{ A \rightarrow C, AC \rightarrow D, E \rightarrow AD, E \rightarrow H \}$
- $G = \{ A \rightarrow CD, E \rightarrow AH \}$

Verificare se F e G sono equivalenti

- Invece di verificare se $X \rightarrow Y$ in F è anche in G^+ , verifico se $Y \subseteq (X)^{+G}$ (chiusura di X rispetto a G), e viceversa per ogni df in G
- per $A \rightarrow C$ risulta $(A)^{+G} = ACD$; o.k. $C \subseteq (A)^{+G}$
- per $AC \rightarrow D$ risulta $(AC)^{+G} = ACD$; o.k. $D \subseteq (AC)^{+G}$
- per $E \rightarrow AD$ risulta $(E)^{+G} = EADCH$; o.k. $AD \subseteq (E)^{+G}$
- per $E \rightarrow H$ risulta $(E)^{+G} = EHADC$; o.k. $H \subseteq (E)^{+G}$

35

Importanza della chiusura di un insieme di attributi

- Dato $R(Z)$ con le sue dipendenze F :
- La chiusura di un insieme $X \subseteq Z$ di attributi è fondamentale per diversi scopi:
 - Si può utilizzare per verificare se una dipendenza funzionale è logicamente implicata da F
 - $X \rightarrow Y$ è in F^+ se e solo se $Y \subseteq X^+$
 - Si può utilizzare per verificare se un insieme di attributi è superchiave o chiave
 - X è superchiave di R se e solo se $X \rightarrow Z$ è in F^+ , cioè se e solo se $Z \subseteq X^+$
 - X è chiave di R se e solo se $X \rightarrow Z$ è in F^+ e non esiste alcun sottoinsieme Y di X ottenuto da X eliminando un solo elemento, tale che $Z \subseteq Y^+$

36

Ancora equivalenza

- F e G sono equivalenti se
- per ogni $X \rightarrow Y \in F$, $Y \in X^+$ secondo G , e,
- per ogni $Z \rightarrow W \in G$, $W \in Z^+$ secondo F

37

Copertura Minimale

- Alcuni attributi di una dipendenza funzionale possono essere ridondanti:
 - ridondanza a DESTRA: $\{ A \rightarrow B, B \rightarrow C, A \rightarrow CD \}$ può essere semplificata in $\{ A \rightarrow B, B \rightarrow C, A \rightarrow C, A \rightarrow D \}$ (FD semplici)
 - ridondanza a SINISTRA: $\{ A \rightarrow B, B \rightarrow C, AC \rightarrow D \}$ può essere semplificata in $\{ A \rightarrow B, B \rightarrow C, A \rightarrow D \}$ (senza attributi estranei)
- Un insieme F di dipendenze funzionali può contenere dipendenze ridondanti, ovvero ottenibili tramite le dipendenze di F
Esempio: $A \rightarrow C$ è ridondante in $\{ A \rightarrow B, B \rightarrow C, A \rightarrow C \}$
- Intuitivamente, una copertura minimale (canonica) di F è un insieme minimale di dipendenze funzionali equivalente a F e privo di dipendenze e attributi ridondanti

38

FD semplici

- Per minimizzare un insieme di FD è innanzitutto necessario scriverle tutte in una forma "standard" (forma canonica), in cui sulla destra c'è sempre un singolo attributo
- Supponiamo di avere
$$F = \{AB \rightarrow CD, AC \rightarrow DE\}$$
- Possiamo riscrivere F come
$$F = \{AB \rightarrow C, AB \rightarrow D, AC \rightarrow D, AC \rightarrow E\}$$

39

Attributi "estranei"

- In alcune FD è possibile che sul lato sinistro ci siano degli attributi inutili ("estranei"): come si identificano?
- Supponiamo di avere $F = \{AB \rightarrow C, A \rightarrow B\}$ e calcoliamo A^+ e B^+
$$A^+ = A \qquad B^+ = B$$
$$A^+ = AB \text{ poiché } A \rightarrow B \text{ e } A \subseteq A^+$$
$$A^+ = ABC \text{ poiché } AB \rightarrow C \text{ e } AB \subseteq A^+$$

C dipende solo da A, e in $AB \rightarrow C$ l'attributo B è estraneo (a sua volta dipende da A) e possiamo riscrivere l'insieme di FD più semplicemente come: $F' = \{A \rightarrow C, A \rightarrow B\}$
- Quindi in una FD del tipo $AX \rightarrow B$ l'attributo A è estraneo se X^+ include B (ovvero X da solo determina B)

40

FD ridondanti

- Dopo avere eliminato gli attributi estranei si deve verificare se vi sono intere FD inutili ("ridondanti"), ovvero FD che sono implicate da altre
- *Come facciamo a stabilire che una FD del tipo $X \rightarrow A$ è ridondante?*
 - La eliminiamo da F , calcoliamo X^+ e verifichiamo se include A , ovvero se con le FD che restano riusciamo ancora a dimostrare che X determina A

41

Copertura Minimale - Definizione

- Un insieme F di FD è minimale se
 1. la parte destra di ogni FD in F è formata da un solo attributo
 2. tutti gli attributi della parte sinistra di ogni FD in F sono necessari (se dato $X \rightarrow A$ si toglie un attributo in X l'insieme delle FD risultante non è più equivalente ad F)
 3. tutte le DF in F sono necessarie, nessuna è ridondante (non è possibile rimuovere una FD da F e avere un insieme equivalente a F)
 - Una copertura minimale di un insieme F è un insieme minimale equivalente a F (può essere usato al posto di F)
- N.B. In generale, la copertura minimale non è unica

42

Copertura Minimale - Algoritmo

- Calcolo di M minimale per un insieme di F :
 - $M = F$
 - ogni $X \rightarrow \{ A_1, A_2, \dots, A_n \}$ è sostituita da $X \rightarrow A_1, X \rightarrow A_2, \dots, X \rightarrow A_n$
 - ogni $X \rightarrow A$ è sostituita da $(X - \{ B \}) \rightarrow A$ se $A \subseteq (X - \{ B \})^+$
 - ogni rimanente $X \rightarrow A$ in M è rimossa se $A \subseteq X^+$ **anche in** $\{ F - \{ X \rightarrow A \} \}$

43

Esempio

- Sia $F = \{ AB \rightarrow C, B \rightarrow A, C \rightarrow A \}$, A è estraneo in $AB \rightarrow C$, quindi trasformiamo F in $F' = \{ B \rightarrow C, B \rightarrow A, C \rightarrow A \}$, dopo possiamo eliminare $B \rightarrow A$ trasformando F' in $F'' = \{ B \rightarrow C, C \rightarrow A \}$
- Se tentiamo di eliminare la FD ridondante prima di eliminare l'attributo estraneo non ci riusciamo
 - **NB: questa operazione è bene che segua l'eliminazione degli attributi estranei**

44

9.2 Relazioni in forma normale

45

Forma normale di Boyce-Codd (BCNF)

- Una relazione r è in forma normale di Boyce-Codd se, per ogni dipendenza funzionale (non banale) $X \rightarrow Y$ definita su di essa, X contiene una chiave K di r
- La forma normale richiede che i concetti in una relazione siano omogenei (tutte le proprietà sono associate alla chiave)

46

BCNF

Se un insieme F di dipendenze per R non è in BCNF, allora in F c'è almeno una dipendenza $X \rightarrow Y$ non banale con X non superchiave di R .

• **Teorema** Data R , se in F non c'è alcuna $X \rightarrow Y$ non banale con X non superchiave di R , allora non ce n'è nemmeno in F^+ .

47

BCNF

- Grazie al teorema, è sufficiente analizzare una ad una le dipendenze non banali in F per verificare se ognuna ha una superchiave come membro sinistro
- *occorre saper verificare se un insieme di attributi è superchiave di una relazione*
 - K è superchiave di $R(Z)$ con dipendenze F se $Z \subseteq K^+$

48

È sempre così facile?

Impiegato	Progetto	Sede
Rossi	Marte	Roma
Verdi	Giove	Milano
Verdi	Venere	Milano
Neri	Saturno	Milano
Neri	Venere	Milano

Impiegato → Sede
Progetto → Sede

51

Decomponiamo sulla base
delle dipendenze

Impiegato	Progetto	Sede
Rossi	Marte	Roma
Verdi	Giove	Milano
Verdi	Venere	Milano
Neri	Saturno	Milano
Neri	Venere	Milano

Impiegato	Sede
Rossi	Roma
Verdi	Milano
Neri	Milano

Progetto	Sede
Marte	Roma
Giove	Milano
Saturno	Milano
Venere	Milano

52

Proviamo a ricostruire

Impiegato	Sede
Rossi	Roma
Verdi	Milano
Neri	Milano

Progetto	Sede
Marte	Roma
Giove	Milano
Saturno	Milano
Venere	Milano

Impiegato	Progetto	Sede
Rossi	Marte	Roma
Verdi	Giove	Milano
Verdi	Venere	Milano
Neri	Saturno	Milano
Neri	Venere	Milano
Verdi	Saturno	Milano
Neri	Giove	Milano

Diversa dalla relazione di partenza!

53

Decomposizione senza perdita

- Una istanza r di una relazione R si decompone senza perdita su X_1 e X_2 se il join naturale delle proiezioni di r su X_1 e X_2 è uguale a r stessa (cioè non contiene ennuple spurie)

54

Algoritmo per la decomposizione in BCNF

- Assumiamo (senza perdita di generalità) che ogni volta che chiamiamo l'algoritmo descritto sotto, ogni dipendenza funzionale in F abbia un unico attributo come membro destro, e che U sia l'insieme di tutti gli attributi di R
- Decomponi(R, F):=


```

      { if esiste  $X \rightarrow A$  in  $F$  con  $X$  non superchiave di  $R$ 
      then      { sostituisci  $R$  con una relazione  $R_1$  con
                  attributi  $U-A$ , ed una relazione  $R_2$  con
                  attributi  $X \cup A$ ;
      Decomponi( $R_1, F_{U-A}$ );
      Decomponi( $R_2, F_{X \cup A}$ );
      }
      }
```

55

Relazione R

Impiegato	Progetto	Sede
Rossi	Marte	Roma
Verdi	Giove	Milano
Verdi	Venere	Milano
Neri	Saturno	Milano
Neri	Venere	Milano

$F = \{\text{Impiegato} \rightarrow \text{Sede}, \text{Progetto} \rightarrow \text{Sede}\}$

Decomponi(R, F):= {R1(Impiegato, Sede), R2(Impiegato, Progetto)}

56

Correttezza dell'algoritmo della decomposizione

- **Teorema** Qualunque sia l'input, l'esecuzione dell'algoritmo su tale input termina, e produce una decomposizione della relazione originaria tale che:
 - ogni relazione ottenuta è in BCNF
 - la decomposizione è senza perdita nel join

57

Proiezione delle FD di $R(U)$ su $X \subseteq U$

- La proiezione di F su X , denotata da F_X , è l'insieme di dipendenze funzionali $Z \rightarrow Y$ in F^+ che coinvolgono solo attributi in X , cioè tali che $Z \subseteq X$ e $Y \subseteq X$

58

Algoritmo

- Per calcolare F_X , cioè la proiezione di F su X , possiamo procedere per enumerazione (non si può fare meglio), evitando però di generare dipendenze funzionali "inutili"
- $\text{CalcolaProiezione}(F,X):=$
 $\{ \text{result} = \emptyset;$
 per ogni sottoinsieme proprio S di X , per ogni attributo A in X tale che A non è in S , e tale che non esiste alcun sottoinsieme S' di S tale che $S' \rightarrow A$ è in result ,
 if A è in $\text{CalcolaChiusura}(S,F)$ allora $\text{result} = \text{result} \cup \{ S \rightarrow A \};$
 $\}$

59

Dimensione della proiezione di F su X

- Ci sono casi in cui la proiezione di F su X ha dimensione esponenziale rispetto alla dimensione di F e X , come mostrato dal seguente esempio
- Consideriamo $R(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_n, C_1, C_2, \dots, C_n, D)$ e
 $F = \{ A_i \rightarrow C_i, B_i \rightarrow C_i \mid 1 \leq i \leq n \} \cup \{ C_1 C_2 \dots C_n \rightarrow D \}$
- La proiezione di F su $\{ A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_n, D \}$ è
 $P = \{ X_1 X_2 \dots X_n \rightarrow D \mid X_i = A_i \text{ oppure } X_i = B_i \text{ per } 1 \leq i \leq n \}$, la cui dimensione è ovviamente esponenziale rispetto alla dimensione dello schema R e delle dipendenze funzionali F .
- Si noti che si può dimostrare che nessun insieme equivalente a P ha cardinalità minore.

60

Proprietà dell'algoritmo di decomposizione

- N.B. A seconda dell'ordine con cui si considerano le dipendenze funzionali, il risultato della decomposizione può cambiare

61

Relazione R

Impiegato	Progetto	Sede
Rossi	Marte	Roma
Verdi	Giove	Milano
Verdi	Venere	Milano
Neri	Saturno	Milano
Neri	Venere	Milano

$F = \{\text{Impiegato} \rightarrow \text{Sede}, \text{Progetto} \rightarrow \text{Sede}\}$

$\text{Decomponi}(R, F) := \{R1(\text{Impiegato}, \text{Sede}), R2(\text{Impiegato}, \text{Progetto})\}$

$\text{Decomponi}(R, F) := \{R1(\text{Impiegato}, \text{Progetto}), R2(\text{Progetto}, \text{Sede})\}$

62

Consideriamo una di queste decomposizioni

Impiegato	Progetto	Sede
Rossi	Marte	Roma
Verdi	Giove	Milano
Verdi	Venere	Milano
Neri	Saturno	Milano
Neri	Venere	Milano

Impiegato	Sede
Rossi	Roma
Verdi	Milano
Neri	Milano

Impiegato	Progetto
Rossi	Marte
Verdi	Giove
Verdi	Venere
Neri	Saturno
Neri	Venere

Impiegato → Sede

Progetto → Sede

63

Osservazione

- La decomposizione è senza perdita sul join, però
 - La FD Progetto → Sede interessa attributi che non stanno nella stessa tabella.
 - E' un problema?

64

Problema

- Supponiamo di voler inserire una nuova ennuola che specifica la partecipazione dell'impiegato Neri, che opera a Milano, al progetto Marte

Impiegato	Sede
Rossi	Roma
Verdi	Milano
Neri	Milano

Impiegato	Progetto
Rossi	Marte
Verdi	Giove
Verdi	Venere
Neri	Saturno
Neri	Venere

Impiegato → Sede

Progetto → Sede

65

Impiegato	Sede
Rossi	Roma
Verdi	Milano
Neri	Milano

Impiegato	Progetto
Rossi	Marte
Verdi	Giove
Verdi	Venere
Neri	Saturno
Neri	Venere
Neri	Marte

66

Impiegato	Progetto	Sede
Rossi	Marte	Roma
Verdi	Giove	Milano
Verdi	Venere	Milano
Neri	Saturno	Milano
Neri	Venere	Milano
Neri	Marte	Milano

67

Conservazione delle dipendenze

- Una decomposizione conserva le dipendenze se ciascuna delle dipendenze funzionali dello schema originario coinvolge attributi che compaiono tutti insieme in uno degli schemi decomposti
- Progetto → Sede non è conservata

68

Decomposizione senza perdita di dipendenze

- Sia R uno schema di relazione con dipendenze funzionali F , e sia X un sottoinsieme di attributi di R
- La decomposizione di R in due relazioni con attributi X e Y è una decomposizione senza perdita di dipendenze se $(F_X \cup F_Y)$ è equivalente a F , cioè se $(F_X \cup F_Y)^+ = F^+$
- N.B. Non è assicurato che la decomposizione ottenuta dall'algoritmo per la decomposizione BCNF sia senza perdita di dipendenze

69

La verifica di decomposizione senza perdita di dipendenze

- La definizione di decomposizione senza perdita di dipendenze è basata sul verificare che $(F_X \cup F_Y)^+ = F^+$.
- Per applicare la definizione,
 - è necessario sapere calcolare se un insieme di dipendenze funzionali è equivalente ad un altro
 - è necessario saper calcolare la proiezione di un insieme di dipendenze funzionali su un insieme di attributi

70

- Per la verifica di equivalenza si può usare un metodo polinomiale e, per ogni $X \rightarrow Y \in F$, calcolare X^+ rispetto a G e verificare se $Y \in X^+$, idem per $X \rightarrow Y \in G$ e X^+ rispetto a F .
- Per calcolare la proiezione abbiamo invece un metodo esponenziale

71

Caso interessante

- La relazione $R(A,B,C)$, con $F = \{ A \rightarrow B, B \rightarrow C, A \rightarrow C \}$ non è in BCNF (qual è la chiave?, qual è la dipendenza sbagliata?).
- Se la decomponiamo in $R_1(A,B)$ e $R_2(B,C)$, partendo da $B \rightarrow C$, otteniamo due relazioni in BCNF, con la proprietà che la decomposizione è senza perdita nel join. La decomposizione è anche senza perdita di dipendenze, perchè la dipendenza funzionale $A \rightarrow C$ è logicamente implicata dalle due dipendenze funzionali che valgono in R_1 e R_2 .
- N.B. Se la definizione di conservazione delle dipendenze non considerasse $(F_X \cup F_Y)^+$ ma solo $(F_X \cup F_Y)$, allora la decomposizione sembrerebbe perdere la dipendenza $A \rightarrow C$, che non è esprimibile direttamente né in R_1 (cioè mediante F_{AB}) né in R_2 (cioè mediante F_{BC}).

72

Qualità delle decomposizioni

- Una decomposizione dovrebbe sempre garantire:
 - BCNF
 - l'assenza di perdite, in modo da poter ricostruire le informazioni originarie
 - la conservazione delle dipendenze, in modo da mantenere i vincoli di integrità originari

DB ben progettato

73

Relazione BCNF?

Proprietà: Ogni dirigente ha una sede;
un progetto può essere diretto da più
persone, ma in sedi diverse

<u>Dirigente</u>	<u>Progetto</u>	<u>Sede</u>
Rossi	Marte	Roma
Verdi	Giove	Milano
Verdi	Marte	Milano
Neri	Saturno	Milano
Neri	Venere	Milano

74

Verifica

<u>Dirigente</u>	<u>Progetto</u>	<u>Sede</u>
Rossi	Marte	Roma
Verdi	Giove	Milano
Verdi	Marte	Milano
Neri	Saturno	Milano
Neri	Venere	Milano

Progetto Sede → Dirigente **ok**
Dirigente → Sede **no**

75

Come decomporre?

- Progetto Sede → Dirigente coinvolge tutti gli attributi e quindi nessuna decomposizione può preservare tale dipendenza
- Si può trovare una BCNF, ma non potrà conservare le dipendenze

76

Approccio differente: una nuova forma normale

- Una relazione r è in terza forma normale se, per ogni FD (non banale) $X \rightarrow Y$ definita su r , è verificata almeno una delle seguenti condizioni:
 - X è superchiave di r
 - ogni attributo in Y è contenuto in almeno una chiave di r

77

Non BCNF, ma 3NF

Dirigente	<u>Progetto</u>	<u>Sede</u>
Rossi	Marte	Roma
Verdi	Giove	Milano
Verdi	Marte	Milano
Neri	Saturno	Milano
Neri	Venere	Milano

Progetto Sede \rightarrow Dirigente
Dirigente \rightarrow Sede
L'attributo Sede è contenuto nella chiave

78

Anomalie?

- C'è una ridondanza nella ripetizione della sede del dirigente per i vari progetti che dirige

79

Confronto

- 3NF è meno restrittiva di BCNF (e ammette relazioni con alcune anomalie e ridondanze)
- il problema di verificare se una relazione è in 3NF è NP-completo (il miglior algoritmo deterministico conosciuto ha complessità esponenziale nel caso peggiore), infatti:
 - Dati R ed F e un attributo A
 - si genera non deterministicamente un sottoinsieme S degli attributi di R che contiene A,
 - si controlla se S è una chiave (non una superchiave)
- ha il vantaggio però di essere sempre "raggiungibile", cioè si può sempre ottenere una decomposizione 3NF senza perdite e che conserva le dipendenze

80

Metodologia di decomposizione (1)

1. Data R ed F minimale, si usa $\text{Decomponi}(R, F)$ ottenendo gli schemi $R_1(X_1), R_2(X_2), \dots, R_n(X_n)$ in BCFN ciascuno con dipendenze F_{X_i}
2. Sia N l'insieme di dipendenze non preservate in R_1, R_2, \dots, R_n , cioè non incluse nella chiusura dell'unione dei vari F_{X_i}
 - Per ogni dipendenza $X \rightarrow A$ in N , aggiungiamo lo schema relazionale $X A$ con le dipendenze funzionali relative a XA

81

Altra metodologia (2)

- Si deriva la copertura minimale G di F .
- Si raggruppano le dipendenze in G in sottoinsiemi tali che ad ogni sottoinsieme G_i appartengono le dipendenze i cui membri sinistri hanno la stessa chiusura: i.e. $X \rightarrow A$ e $Y \rightarrow B$ appartengono a G_i se $X^+ = Y^+$ secondo G .
- Si partizionano gli attributi U nei sottoinsiemi U_i individuati dai sottoinsiemi G_i del passo precedente. Se un sottoinsieme è contenuto in un altro si elimina.
- Si crea una relazione $R_i(U_i)$ per ciascun sottoinsieme U_i , con associate le dipendenze G_i .
- Si aggiunge una relazione per gli attributi che non sono coinvolti in alcuna FD
- Se non c'è già una relazione che contenga una chiave della relazione originaria, si aggiunge

82

esempio (metodologia 2)

Se le FD individuate su $R(ABCDEFG)$ sono:

$AB \rightarrow CD$, $AB \rightarrow E$, $C \rightarrow F$, $F \rightarrow G$

si generano gli schemi

$R1(ABCDE)$, $R2(CF)$, $R3(FG)$

83

esempio (cont)

Se le FD su $R(ABCD)$ sono:

$A \rightarrow BC$, $B \rightarrow A$, $C \rightarrow D$

si generano gli schemi

$R1(ABC)$, $R2(CD)$ con A o B chiave in $R1$

84

esempio (cont)

Se le FD su $R(ABCD)$ sono:

$A \rightarrow C, B \rightarrow D$

si generano gli schemi

$R_1(AC), R_2(BD), R_3(AB)$

85

Confronto

- La prima metodologia garantisce come primo passo l'assenza di perdita sul join e poi conserva le dipendenze
- La seconda conserva le dipendenze e poi risolve l'eventuale perdita sul join

86

In generale

- Una volta effettuata la decomposizione in 3NF con la metodologia precedente si verifica se lo schema ottenuto è anche BCNF
- Se la relazione ha una sola chiave allora le due forme normali coincidono
- N.B. nel secondo esempio questo non succede

87

Qualità delle decomposizioni (2)

- Una decomposizione dovrebbe sempre garantire:
 - BCNF o 3NF
 - l'assenza di perdite, in modo da poter ricostruire le informazioni originarie
 - la conservazione delle dipendenze, in modo da mantenere i vincoli di integrità originari

88

- Quando una BCNF non è raggiungibile spesso è questione di cattiva progettazione

89

Progettazione e normalizzazione

- la teoria della normalizzazione può essere usata nella progettazione logica per verificare lo schema relazionale finale
- si può usare anche durante la progettazione concettuale per verificare la qualità dello schema concettuale

90