Corso di Laurea
in
Ingegneria Informatica
"*Basi di dati*"
a.a. 2019-2020

Docente: Gigliola Vaglini
Docente di laboratorio SQL: Francesco Pistolesi

Lezione 7

Qualità delle relazioni

Come abbiamo progettato

- Abbiamo ipotizzato che gli attributi vengano raggruppati per formare uno schema di relazione usando il buon senso del progettista di basi di dati o traducendo uno schema di base di dati da un modello dei dati concettuale (ER).
- Ma abbiamo bisogno di misurare formalmente perché un raggruppamento di attributi in uno schema di relazione possa essere migliore di un altro.

- Obiettivo: valutare la qualità della progettazione di schemi relazionali

Che approccio abbiamo usato

- Top-down: abbiamo iniziato individuando un certo numero di raggruppamenti di attributi per formare relazioni che sussistono come tali nel mondo reale, ad esempio una fattura, un form o un report. Queste relazioni sono poi state analizzate portando eventualmente a decomposizioni successive.

Obiettivi impliciti

1. La conservazione dell'informazione, cioè il mantenimento di tutti i concetti espressi precedentemente mediante il modello concettuale, inclusi tipi di attributi, tipi di entità e tipi di associazioni.
2. La minimizzazione della ridondanza, cioè l'evitare la memorizzazione ripetuta della stessa informazione, e quindi la necessità di effettuare molteplici aggiornamenti al fine di mantenere la consistenza tra le diverse copie della medesima informazione.

Possiamo derivare da questi obiettivi alcune linee guida per il progetto

5

5

LG1: semplice è bello

- Uno schema di relazione deve essere progettato in modo che sia semplice spiegarne il significato. Non si devono raggruppare attributi provenienti da più tipi di entità e tipi di relazione in un'unica relazione.
- Intuitivamente: se uno schema di relazione corrisponde a un solo tipo di entità o a un solo tipo di relazione, risulta semplice spiegarne il significato; in caso contrario, nascerà un'ambiguità semantica e quindi lo schema non potrà essere spiegato con facilità

6

6

LG2: no alle anomalie

- Gli schemi vanno progettati in modo che non possano presentarsi anomalie di inserimento, cancellazione o modifica.
- Se possono presentarsi anomalie, vanno chiaramente rilevate e si deve assicurare che i programmi che aggiornano la base di dati operano correttamente.

7

7

Esempio 1

- Fattura(CodF, CodProd, TotDaPagare, CostoNettoProd, IVA)
- Semantica attributi:
  - CodF determina CodProd e TotDaPagare
  - CodProd determina CostoNettoProd e IVA
  - CostoNettoProd e IVA determinano TotDaPagare

8

8

Esempio 1 cont.

- Ovviamente TotDaPagare deve essere consistente con la regola che lo lega al CostoNettoProd e all'IVA
- Per evitare anomalie di inserimento o modifica conviene che TotDaPagare non ci sia nella tabella Fattura

9

9

Esempio 2

- Anagrafe(CF, NomeP, Indirizzo, NomeC, NumAb)
- Semantica attributi:
  - CF determina NomeP, Indirizzo e NomeC
  - NomeC determina NumAb

10

10

Esempio2 cont.

- NumAb è ripetuto per lo stesso NomeC per quanti sono i residenti
- Il valore deve essere mantenuto consistente (uguale) per ogni persona di una stessa città

11

11

Esempio 2 cont.

- Come si può evitare il problema
- Trasformando Anagrafe in due schemi separati
  - Persona(CF, NomeP, Indirizzo, NomeC)
  - Residenza(NomeC, NumAb)
  - Con vincolo di integrità referenziale su NomeC e un vincolo aggiuntivo su NumAb…

12

12

LG3: evitare frequenti valori nulli

- Si eviti di porre in una relazione attributi i cui valori possono essere frequentemente nulli.
- Se i valori nulli sono inevitabili, ci si assicuri che si presentino solo in casi eccezionali rispetto al numero di tuple di una relazione.

13

13

Verso un approccio formale

- Dipendenze funzionali (FD)

- Una dipendenza funzionale esprime un legame semantico tra due gruppi di attributi di uno schema di relazione R
- Una FD è una proprietà di R, non di un particolare stato valido r di R.
- Una FD non può essere dedotta a partire da uno stato valido r, ma deve essere definita esplicitamente da qualcuno che conosce la semantica degli attributi di R.

14

14

Cont.

- Forme normali

- Esistono diversi tipi di forma normale, ognuna garantisce l'assenza di determinati difetti in uno schema di relazione R e quindi definisce un determinato livello di qualità di R
- E' possibile eseguire una serie di test per certificare che R soddisfa una data forma normale

15

15

Normalizzazione

- La normalizzazione è la procedura che permette di portare uno schema relazionale in una determinata forma normale

- La normalizzazione è utilizzata come tecnica di verifica dei risultati della progettazione, non costituisce una metodologia di progetto

16

16

Esempio 3

<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

17

17

Semantica

- Ogni impiegato può partecipare a più progetti, sempre con lo stesso stipendio, e con una sola funzione per progetto.
- Ogni progetto ha un bilancio indipendentemente da quanti dipendenti ci lavorano

18

18

Possibili anomalie

- 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
- Se un progetto diminuisce/aumenta il suo bilancio, devo cambiare l'attributo relativo per tutti gli impiegati che ci lavorano
  - anomalia di aggiornamento

19

19

Causa dei problemi

- Abbiamo ripetizione dello stipendio di un impiegato e del bilamcio di un progetto
- Errore di progetto: Abbiamo usato un'unica relazione per rappresentare gruppi di informazioni eterogenee

20

20

Vediamo di usare il concetto di

dipendenza funzionale

per studiare meglio questi problemi

21

21

7.1. Le dipendenze funzionali

22

22

Dipendenza funzionale: definizione

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

Notazione: $Y \rightarrow Z$

- Ad es., ad ogni chiave K di R corrisponde una dipendenza funzionale $K \rightarrow X$
- N.B. non è detto che esista $Z \rightarrow Y$

23

23

Esempio 3 cont.

- Caratterizziamo in termini di dipendenze le informazioni semantiche che abbiamo
- Ogni impiegato hanno un solo stipendio

Impiegato → Stipendio

- Ogni progetto hanno un solo bilancio

Progetto → Bilancio

- Ogni impiegato ha una sola funzione per progetto

Impiegato Progetto → Funzione

24

24

Impiegato	Stipendio	Progetto	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

Impiegato → Stipendio
Progetto → Bilancio
Impiegato Progetto → Funzione

25

25

Esistono altre FD?

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

26

26

Legame tra FD e anomalie

Impiegato → Stipendio ci sono ripetizioni

Progetto → Bilancio ci sono ripetizioni

Impiegato Progetto → Funzione non ci sono ripetizioni

- 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 lo sono.

27

27

7.2. Un po' di teoria delle dipendenze

28

28

Implicazione

- Sia F un insieme di dipendenze funzionali definite su R(Z) e sia $X \rightarrow Y$:
  - 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

29

29

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^+

30

30

Superchiave

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

31

31

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 *costruttivamente* 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^+

32

32

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 $X \rightarrow Z$

33

33

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.

34

34

Esempi di prove

- DIMOSTRARE che, $\forall$ istanza di relazione,
  $X \rightarrow Y \Rightarrow X Z \rightarrow Y Z$
- Supponiamo per assurdo che esista una istanza r di R in cui valga $X \rightarrow Y$ ma non $X Z \rightarrow Y Z$,
  devono perciò esistere due tuple t1 e t2 di r tali che :
  (1) t1[X] = t2[X], (2) t1[Y] = t2[Y],
  (3) t1[XZ] = t2[XZ], (4) t1[YZ] ≠ t2[YZ]
  ma ciò è assurdo, poichè da (1) e (3) si deduce:
  (5) t1[Z] = t2[Z],
  e da (2) e (5) si deduce :
  (6) t1[YZ] = t2[YZ],
  in contraddizione con la (4)

35

35

(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] ≠ t2[Z]
  ma ciò è assurdo

36

36

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 $Z \subseteq Y$, $X \rightarrow Y \Rightarrow X \rightarrow Z$

37

37

Esempi di prove

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

38

38

Esempio di calcolo di F^+ con le regole

- Prendiamo le FD dell'esempio
  i. **Impiegato → Stipendio**
  ii. **Progetto → Bilancio**
  iii. **Impiegato Progetto → Funzione**
- E usiamo la regola 2 sulle dipendenze i e ii, otteniamo
  iv. **Impiegato Progetto → Stipendio Progetto**
  v. **Progetto Impiegato → Bilancio Impiegato**
- Di conseguenza otteniamo con la regola 4
  – **Impiegato Progetto → Stipendio Progetto Impiegato Bilancio Funzione**
- e quindi Impiegato Progetto è chiave
- Ci sono altre FD in F^+?

39

39

Equivalenza

- Dato un insieme di FD F, è molto utile poter determinare se un insieme di FD G sia equivalente ad F
- 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 $X \rightarrow Y \in F^+$

40

40

Esempio 1

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

Verificare se F e G sono equivalenti

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

41

41

(cont.)

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

42

42

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
- Come alternativa si può calcolare e utilizzare la chiusura transitiva di un insieme di attributi X, infatti
  - si può dimostrare che $X \rightarrow Y$ è in F^+ sse $Y \subseteq X^+$

43

43

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?)
  - CalcolaChiusura(X,F)=

```
{ X+ = X;
  Ripeti:
    – Fine = true;
    – Per tutte le FD in F = {Vi → Wi}:
    – Se Vi ⊆ X+ e Wi ⊄ X+ allora: {X+ = X+ ∪ Wi; Fine = false}
  Fino a che Fine = true
}
```

44

44

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)!

45

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 FD 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. $H \subseteq (E)^{+G}$

E viceversa per ogni FD in G

46

quindi

- 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

47

47

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 ottenuto da X eliminando almeno un elemento, tale che $Z \subseteq Y^+$

48

48

Ridondanze di un insieme di FD

- 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 altre dipendenze di F

  Esempio: $A \rightarrow C$ è ridondante in $\{A \rightarrow B, B \rightarrow C, A \rightarrow C\}$

49

49

FD semplici

- Possiamo portare un insieme di FD F in forma "standard", quella in cui sulla destra c'è un singolo attributo
- Supponiamo di avere

  $F = \{AB \rightarrow CD, AC \rightarrow DE\}$
- Possiamo riscrivere F come

  $F = \{AB \rightarrow C, AB \rightarrow D, AC \rightarrow D, AC \rightarrow E\}$

50

50

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^+
  $A^+ = A$ $B^+ = B$
  $A^+ = AB$ poiché $A \rightarrow B$ e $A \subseteq A^+$
  $A^+ = ABC$ poiché $AB \rightarrow C$ 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)*

51

51

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

52

52

Copertura Minimale – Definizione

- Un insieme F di FD è minimale se nella parte destra di ogni FD c'è un solo attributo e non si possono togliere attributi nella parte sinistra di qualche FD né alcuna FD senza perdere l'equivalenza dell'insieme ottenuto con F.
- Una copertura minimale di un insieme F è quindi un insieme equivalente a F (può essere usato al posto di F), ma di complessità minore.

N.B. In generale, la copertura minimale non è unica

Copertura Minimale - Algoritmo

- Calcolo di M minimale per un insieme di F :
  - M = F
  - ogni $X \rightarrow \{ A1, A2, \ldots, An \}$ è sostituita da $X \rightarrow A1$, $X \rightarrow A2$, ..., $X \rightarrow An$
  - 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** $\{ \{ F - \{ X \rightarrow A \} \}$

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**

55

55

7.3. Forme normali e normalizzazione

56

56

Legame tra FD e anomalie

Impiegato $\rightarrow$ Stipendio ci sono ripetizioni

Progetto $\rightarrow$ Bilancio ci sono ripetizioni

Impiegato Progetto $\rightarrow$ Funzione non ci sono ripetizioni

- 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 lo sono.

57

57

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 è superchiave di r

- Questa forma normale richiede che i concetti in una relazione siano omogenei (cioè che tutte le proprietà siano direttamente associate alla chiave)

58

58

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** Dato uno schema R e un insieme F di FD, se F non contiene alcuna $X \rightarrow Y$ non banale con X non superchiave di R, allora neanche F^+ la contiene.

59

59

Verifica BCNF

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

60

60

Normalizzazione

- Se una relazione non è in BCNF la rimpiazziamo con altre relazioni che siano BCNF

Come?

- Decomponendo sulla base delle dipendenze funzionali, al fine di separare i concetti

61

61

Impiegato	Stipendio	Progetto	Bilancio	Funzione
Rossi				tecnico
				progettista
				consulente
Mori				direttore
Mori				progettista
Bianchi				progettista
Bianchi	48	Giove	15	direttore

Impiegato	Stipendio
Rossi	20
Verdi	35
Neri	55
Mori	48
Bianchi	48

Impiegato	Progetto	Funzione
Rossi	Marte	tecnico
Verdi	Giove	progettista
Verdi	Venere	progettista
Neri	Venere	direttore
Neri	Giove	consulente
Neri	Marte	consulente
Mori	Marte	direttore
Mori	Venere	progettista
Bianchi	Venere	progettista
Bianchi	Giove	direttore

Progetto	Bilancio
Marte	2
Giove	15
Venere	15

62

62

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

63

63

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

64

64

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!

65

65

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)

66

66

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 R1 con attributi U-A, ed una relazione R2 con attributi $X \cup A$;
  Decomponi(R1,F_{U-A});
  Decomponi(R2,$F_{X \cup A}$)
  }
  }

67

67

Relazione R

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

F={Impiegato → Sede, Progetto → Sede}

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

68

68

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

69

69

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 $Y \subseteq X$

70

70

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"

- CalcolaProiezione(F,X):=
  { *result* = +∅;
  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' → A è in *result,*
  if $A \subseteq S^+$ then result = *result* ∪ { S → A };
  }

71

71

Dimensione della proiezione di F su X

- Ci sono casi un cui la proiezione di F su X ha dimensione esponenziale rispetto alla dimensione di F e X, come mostrato dal seguente esempio
- Consideriamo R(A1,A2,…,An,B1,B2,….,Bn,C1,C2,…,Cn,D) e
  F = { Ai → Ci, Bi → Ci | 1 ≤ i ≤ n } ∪ { C1C2…Cn → D }
- La proiezione di F su { A1,A2,…,An,B1,B2,….,Bn,D } è
  P = { X1X2…Xn → D | Xi = Ai oppure Xi = Bi per 1 ≤ i ≤ 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.

72

72

Proprietà dell'algoritmo di decomposizione

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

Relazione R

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

F={**Impiegato** → **Sede, Progetto** → **Sede**}

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

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

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

75

75

Osservazione

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

76

76

Problema

- Supponiamo di voler inserire una nuova ennupla 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

77

77

Impiegato	Sede
Rossi	Roma
Verdi	Milano
Neri	Milano

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

78

78

Impiegato	Progetto	Sede
Rossi	$\boxed{\text{Marte}}$	$\boxed{\text{Roma}}$
Verdi	Giove	Milano
Verdi	Venere	Milano
Neri	Saturno	Milano
Neri	Venere	Milano
Neri	Marte	Milano

79

79

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 $\rightarrow$ Sede non è conservata

80

80

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

81

81

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

82

82

- 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

83

83

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?).
- Se la decomponiamo in R1(A,B) e R2(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 R1 e R2.
- 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 R1 (cioè mediante F_{AB}) nè in R2 (cioè mediante F_{BC}).

84

84

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

85

85

Relazione BCNF?

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

Dirigente	**Progetto**	**Sede**
Rossi	Marte	Roma
Verdi	Giove	Milano
Verdi	Marte	Milano
Neri	Saturno	Milano
Neri	Venere	Milano

86

86

Verifica

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

Progetto Sede → Dirigente ok
Dirigente → Sede no

87

87

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

88

88

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

89

89

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

90

90

Anomalie?

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

91

91

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

92

92

Metodologia di decomposizione (1)

1. Data R ed F minimale, si usa Decomponi(R,F) ottenendo gli schemi $R_1(X_1)$, $R_2(X_2)$,…, $R_n(X_n)$ in BCFN ciascuno con dipendenze F_{X_i}
2. Sia N l'insieme di dipendenze non preservate in $R_1,R_2,…,R_n$, 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

93

93

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 Ui, 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

94

94

esempio metodologia 2

Se le FD individuate su R(ABCDEFG) sono:

AB→CD, AB→E, C→F, F→G

si generano gli schemi

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

95

95

esempio (cont)

Se le FD su R(ABCD) sono:

A→BC, B→A, C→D

si generano gli schemi

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

96

96

esempio (cont)

Se le FD su R(ABCD) sono:

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

si generano gli schemi

R1(AC), R2(BD), R3(AB)

97

97

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

98

98

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

99

99

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

100

100

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

101

101

Progettazione e normalizzazione

- la teoria della normalizzazione serve per verificare la qualità dello schema logico
- Ma si può usare anche durante la progettazione concettuale per ottenere uno schema di buona qualità (verifica ridondanze, partizionamento di entità/relazioni)

102

102