Normalizzazione di Schemi Relazionali

Forme Normali

- Una forma normale è una proprietà di uno schema relazionale che ne garantisce la "qualità", cioè l'assenza di determinati difetti
- Una relazione non normalizzata:
  - presenta ridondanze,
  - si presta a comportamenti poco desiderabili durante gli aggiornamenti
- Le forme normali sono di solito definite sul modello relazionale, ma hanno senso anche in altri contesti, ad esempio nel modello E/R

- L'attività che permette di trasformare schemi non normalizzati in schemi che soddisfano una forma normale è detta normalizzazione
- La normalizzazione va utilizzata come tecnica di verifica dei risultati della progettazione di una base di dati
- Non costituisce quindi una metodologia di progettazione

Una relazione con “anomalie”

- Lo stipendio di ciascun impiegato è ripetuto in tutte le tuple relative: ridondanza
- Se lo stipendio di un impiegato varia, è necessario modificare il valore in diverse tuple: anomalia di aggiornamento
- Se un impiegato interrompe la partecipazione a tutti i progetti, dobbiamo cancellarlo: anomalia di cancellazione
- Un nuovo impiegato senza progetto non può essere inserito: anomalia di inserimento

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

Analizziamo la relazione…

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

- Ma abbiamo usato un'unica relazione per rappresentare tutte queste informazioni eterogenee:
  - gli impiegati con i relativi stipendi
  - i progetti con i relativi bilanci
  - le partecipazioni degli impiegati ai progetti con le relative funzioni

Dipendenze Funzionali

- Per formalizzare i problemi visti si introduce un nuovo tipo di vincolo, la dipendenza funzionale

Consideriamo:

- Un'istanza r di uno schema R(X)
- Due sottoinsiemi (non vuoti) di attributi Y e Z di X

- Diciamo che in r vale la dipendenza funzionale (FD) $Y \rightarrow Z$
  (Y determina funzionalmente Z) se

per ogni coppia di tuple t1 e t2 di r con gli stessi valori su Y,
t1 e t2 hanno gli stessi valori anche su Z

Esempi di FD

- Nella relazione vista si hanno diverse FD, tra cui:

  Impiegato $\rightarrow$ Stipendio

  Progetto $\rightarrow$ Bilancio

  Impiegato, Progetto $\rightarrow$ Funzione

- Altre FD sono meno “interessanti” (“banali”), perché sempre soddisfatte, ad esempio:

  Impiegato, Progetto $\rightarrow$ Progetto

  - $Y \rightarrow A$ è non banale se A non appartiene a Y
  - $Y \rightarrow Z$ è non banale se nessun attributo in Z appartiene a Y

Anomalie e FD

- Le anomalie viste si riconducono alla presenza delle FD:

  Impiegato $\rightarrow$ Stipendio

  Progetto $\rightarrow$ Bilancio

- Viceversa la FD

  Impiegato, Progetto $\rightarrow$ Funzione

  non causa problemi
- Motivo:
  - La terza FD ha sulla sinistra una chiave e non causa anomalie
  - Le prime due FD non hanno sulla sinistra una chiave e causano anomalie

- La relazione contiene alcune informazioni legate alla chiave e altre ad attributi che non formano una chiave

Seconda Forma Normale (2NF)

- Per evitare le anomalie viste si può introdurre la

Seconda Forma Normale (2NF)

Uno schema R(X) è in seconda forma normale se e solo se
ogni attributo non-primo (ovvero non appartenente a nessuna chiave)
dipende **completamente** da ogni chiave
(ovvero non dipende solamente da una parte di chiave)

- Esiste in realtà anche una 1NF

Prima Forma Normale (1NF)

Richiede semplicemente che tutti gli attributi dello schema abbiano domini "atomici" (ovvero non siano composti o multivalore)

Forma Normale di Boyce e Codd (BCNF)

- Ma per evitare anche altre anomalie è meglio ricorrere alla:

Forma Normale di Boyce-Codd (BCNF)

Uno schema R(X) è in forma normale di Boyce e Codd se e solo se, per ogni dipendenza funzionale (non banale) $Y \rightarrow Z$ definita su di esso, Y è una **superchiave** di R(X)

- Si noti che, come al solito, i vincoli si riferiscono allo schema, in quanto dipendeno dalla semantica degli attributi
- Un'istanza può pertanto soddisfare "per caso" il vincolo, ma ciò non garantisce che lo schema sia normalizzato

- In altri termini, le FD non si "ricavano" dall'analisi dei dati, ma ragionando sugli attributi dello schema

Normalizzazione in BCNF

- Se uno schema non è in BCNF, la soluzione è “decomporlo”, sulla base delle FD

Impiegato → Stipendio

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

Impiegato, Progetto → Funzione

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

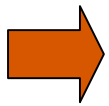
Progetto → Bilancio

Progetto	Bilancio
Marte	2
Giove	15
Venere	15

Attenzione!

- La soluzione non è sempre così semplice, bisogna fare anche altre considerazioni; ad esempio, operando come prima:

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 → Sede

Progetto	Sede
Marte	Roma
Giove	Milano
Saturno	Milano
Venere	Milano

Progetto → Sede

…se proviamo a tornare indietro (Join su Sede):

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!

Decomposizione Senza Perdita

- La decomposizione non deve assolutamente alterare il contenuto informativo del DB
- Si introduce pertanto il seguente requisito

Decomposizione senza perdita *(lossless)*

Uno schema R(X) si decompone senza perdita negli schemi R1(X1) e R2(X2) se, per ogni istanza legale r su R(X), il join naturale delle proiezioni di r su X1 e X2 è uguale a r stessa:

$$\pi_{X1}(r) \bowtie \pi_{X2}(r) = r$$

- Una decomposizione con perdita può generare tuple spurie
- Per decomporre senza perdita è necessario e sufficiente che il join naturale sia eseguito su una superchiave di uno dei due sottoschemi, ovvero che valga $X1 \cap X2 \rightarrow X1$ oppure $X1 \cap X2 \rightarrow X2$

Esempio di decomposizione lossless

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

… ma i problemi non sono ancora finiti…

Modifichiamo il DB…

- Supponiamo di voler inserire l'informazione che Neri lavora al progetto Marte:

Impiegato	Sede
Rossi	Roma
Verdi	Milano
Neri	Milano

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

- Ricostruendo la relazione otteniamo:

che però viola la FD Progetto → Sede!

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

Preservazione delle Dipendenze

- Diciamo che una decomposizione preserva le dipendenze se ciascuna delle dipendenze funzionali dello schema originario coinvolge attributi che compaiono tutti insieme in uno degli schemi decomposti
  - Nell'esempio Progetto $\rightarrow$ Sede non è conservata
- Se una FD non si preserva diventa più complicato capire quali sono le modifiche del DB che non violano la FD stessa
- In generale si devono prima eseguire query SQL di verifica

Esempio di query di verifica

- Bisogna verificare che il progetto (Marte) sia presso la stessa sede dell'impiegato (Neri). A tal fine bisogna trovare un impiegato che lavora al progetto Marte

Impiegati

Impiegato	Sede
Rossi	Roma
Verdi	Milano
Neri	Milano

ImpProg

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

```
SELECT *      -- OK se restituisce una tupla
FROM   Impiegati I
WHERE  I.Impiegato = 'Neri'
  AND  Sede IN ( SELECT I1.Sede
                 FROM   Impiegati I, ImpProg IP
                 WHERE  I1.Impiegato = IP.Impiegato
                   AND  IP.Progetto = `Marte')
```

Qualità delle Decomposizioni

- Una decomposizione:
  - deve essere senza perdita, per garantire la ricostruzione delle informazioni originarie
  - dovrebbe conservare le dipendenze, per semplificare il mantenimento dei vincoli di integrità originari
- Nel nostro esempio, questo suggerisce di inserire anche lo schema:
  - Va sempre eseguita una query, ma più "semplice":

```
SELECT *  -- OK se restituisce una tupla
FROM   Impiegati I, Progetti P
WHERE  I.Impiegato = 'Neri'
  AND  P.Progetto = `Marte'
  AND  I.Sede = P.Sede
```

Progetti

Progetto	Sede
Marte	Roma
Giove	Milano
Venere	Milano
Saturno	Milano

Una limitazione non superabile…

- In funzione del pattern di FD, può non essere possibile decomporre in BCNF e preservare le FD

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

Progetto, Sede → Dirigente
Dirigente → Sede

- Progetto, Sede → Dirigente coinvolge tutti gli attributi e quindi nessuna decomposizione può preservare tale dipendenza!

La Terza Forma Normale (3NF)

- Una forma normale meno restrittiva della BCNF si definisce come segue:

Terza Forma Normale (3NF)

Uno schema R(X) è in terza forma normale se e solo se,
per ogni dipendenza funzionale (non banale) $Y \rightarrow Z$ definita su di esso,
Y è una superchiave di R(X) oppure
ogni attributo in Z è primo (cioè contenuto in almeno una chiave di R(X)

Una relazione in 3NF può ancora presentare anomalie

- Tuttavia il vantaggio è che è sempre possibile ottenere schemi in 3NF preservando tutte le dipendenze

La definizione è anche equivalente alla seguente:

Uno schema R(X) è in terza forma normale se e solo se ogni attributo non-primo non dipende **transitivamente** da nessuna chiave

Decomposizione in 3NF

- L'idea alla base dell'algoritmo che produce una decomposizione in 3NF è creare una relazione per ogni gruppo di FD che hanno lo stesso lato sinistro (determinante) e inserire nello schema corrispondente gli attributi coinvolti in almeno una FD del gruppo

  Esempio: Se le FD individuate sullo schema R(ABCDEFG) sono:

  AB → CD, AB → E, C → F, F → G

  si generano gli schemi **R1(ABCDE), R2(CF), R3(FG)**

- Se 2 o più determinanti si determinano reciprocamente, si fondono gli schemi (più chiavi alternate per lo stesso schema)

  Esempio: 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

- Alla fine si verifica che esista uno schema la cui chiave è anche chiave dello schema originario (se non esiste lo si crea)

  Esempio: Se le FD su R(ABCD) sono: **A → C, B → D**

  si generano gli schemi **R1(AC), R2(BD), R3(AB)**

In pratica…

- Se la relazione non è normalizzata si decompone in terza forma normale
- Si verifica se lo schema ottenuto è anche in BCNF
  - Si noti che se una relazione ha una sola chiave allora le due forme normali coincidono

- Se uno schema non è in BCNF si hanno 3 alternative:

1) Si lascia così com'è, gestendo le anomalie residue (se l'applicazione lo consente)
2) Si decompone in BCNF, predisponendo opportune query di verifica
3) Si cerca di rimodellare la situazione iniziale, al fine di permettere di ottenere schemi BCNF

Decomposizione dello schema

- È innanzitutto opportuno osservare che {Progetto, Dirigente} è una chiave
- La decomposizione:

non va bene, perché è con perdita!

ProgSedi

Progetto	Sede
Marte	Roma
Marte	Milano
Giove	Milano
Saturno	Milano
Venere	Milano

Dirigenti

Dirigente	Sede
Rossi	Roma
Verdi	Milano
Neri	Milano

- La decomposizione corretta è:

ProgDir

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

Dirigenti

Dirigente	Sede
Rossi	Roma
Verdi	Milano
Neri	Milano

Ridefinizione dello schema

- Nell'esempio, introduciamo il concetto di Reparto per distinguere i dirigenti di una stessa sede (ogni dirigente opera in un reparto di una sede, e viceversa)

Dirigente	Progetto	Sede	Reparto
Rossi	Marte	Roma	1
Verdi	Giove	Milano	1
Verdi	Marte	Milano	1
Neri	Saturno	Milano	2
Neri	Venere	Milano	2

Dirigente → Sede, Reparto
Sede, Reparto → Dirigente
Progetto, Sede → Reparto

- È ora possibile operare una decomposizione in BCNF

Dirigente	Sede	Reparto
Rossi	Roma	1
Verdi	Milano	1
Neri	Milano	2

Progetto	Sede	Reparto
Marte	Roma	1
Giove	Milano	1
Marte	Milano	1
Saturno	Milano	2
Venere	Milano	2

Normalizzare o no?

- La normalizzazione non va intesa come un obbligo, in quanto in alcune situazioni le anomalie che si riscontrano in schemi non normalizzati sono un male minore rispetto alla situazione che si viene a creare normalizzando
- In particolare, le cose da considerare sono:
  - Normalizzare elimina le anomalie, ma può appesantire l'esecuzione di certe operazioni (join tra gli schemi normalizzati)
  - La frequenza con cui i dati vengono modificati incide su qual è la scelta più opportuna (relazioni "quasi statiche" danno meno problemi se non normalizzate)
  - La ridondanza presente in relazioni non normalizzate va quantificata, per capire quanto incida sull'occupazione di memoria, e sui costi da pagare quando le repliche di una stessa informazione devono essere aggiornate

Riassumiamo:

- Una forma normale è una proprietà di uno schema relazionale che ne garantisce la "qualità", cioè l'assenza di determinati difetti
- Una relazione non normalizzata presenta ridondanze e dà luogo a comportamenti poco desiderabili durante gli aggiornamenti
- La definizione delle forme normali (3NF e BCNF) si basa sul vincolo di dipendenza funzionale (FD)
- Normalizzare uno schema significa decomporlo in sottoschemi
- Ogni decomposizione deve essere senza perdita, ovvero deve permettere di ricostruire esattamente la relazione originaria non decomposta
- È anche opportuno che la decomposizione preservi le FD, al fine di evitare (o ridurre la complessità di) query di verifica che garantiscano che i vincoli siano rispettati