



Politecnico
di Torino

DBG
MG

Normalizzazione

Progettazione delle basi di dati

Normalizzazione

- Introduzione
- Forma normale di Boyce Codd
- Decomposizione in forma normale
- Proprietà delle decomposizioni
- Decomposizione senza perdita
- Conservazione delle dipendenze

Introduzione

Normalizzazione

Normalizzazione

- La normalizzazione è un procedimento che, a partire da uno schema relazionale non normalizzato, permette di ottenere uno schema relazionale normalizzato
- La normalizzazione **non è una metodologia di progettazione**, bensì uno **strumento di verifica**
- La metodologia di progettazione basata su schemi ER produce normalmente schemi relazionali normalizzati
- Le verifiche di normalizzazione possono essere applicate anche agli schemi ER

Esempio

Esame Superato

<u>MatrStudente</u>	Residenza	<u>CodCorso</u>	NomeCorso	Voto
s94539	Milano	04FLYCY	Calcolatori elettronici	30
s94540	Torino	01FLTCY	Basi di dati	26
s94540	Torino	01KPNCY	Reti di calcolatori	28
s94541	Pescara	01KPNCY	Reti di calcolatori	29
s94542	Lecce	04FLYCY	Calcolatori elettronici	25

- **Vincoli**

- La chiave primaria è la coppia MatrStudente, CodCorso
- La residenza di ogni studente è unica ed è funzione solo dello studente, indipendentemente dagli esami che ha superato
- Il nome del corso è unico ed è funzione solo del corso, indipendentemente dagli studenti che superano il corrispondente esame

Esempio: Ridondanza

Esame Superato

<u>MatrStudente</u>	Residenza	<u>CodCorso</u>	NomeCorso	Voto
s94539	Milano	04FLYCY	Calcolatori elettronici	30
s94540	Torino	01FLTCY	Basi di dati	26
s94540	Torino	01KPNCY	Reti di calcolatori	28
s94541	Pescara	01KPNCY	Reti di calcolatori	29
s94542	Lecce	04FLYCY	Calcolatori elettronici	25

- **Ridondanza**

- In tutte le righe in cui compare uno stesso studente è ripetuta la sua residenza
- In tutte le righe in cui compare uno stesso corso è ripetuto il suo nome

Esempio: Anomalia

Esame Superato

<u>MatrStudente</u>	Residenza	<u>CodCorso</u>	NomeCorso	Voto
s94539	Milano	04FLYCY	Calcolatori elettronici	30
s94540	Torino	01FLTCY	Basi di dati	26
s94540	Torino	01KPNCY	Reti di calcolatori	28
s94541	Pescara	01KPNCY	Reti di calcolatori	29
s94542	Lecce	04FLYCY	Calcolatori elettronici	25

- **Anomalia di aggiornamento**

- Se la residenza di uno studente cambia, occorre modificare tutte le righe in cui compare contemporaneamente

- **Anomalia di inserimento**

- Se un nuovo studente si iscrive all'università, non può essere inserito nella base dati fino a quando non supera il primo esame

- **Anomalia di cancellazione**

- Se uno studente rinuncia agli studi, non è possibile tener traccia della sua residenza

Ridondanza

- Un'unica relazione è utilizzata per rappresentare informazioni eterogenee
 - alcuni dati sono ripetuti in tuple diverse senza aggiungere nuova informazione
 - dati ridondanti

Anomalia

- Le informazioni ridondanti devono essere aggiornate in modo atomico (tutte contemporaneamente)
- La cancellazione di una tupla comporta la cancellazione di tutti i concetti in essa rappresentati
 - inclusi quelli che potrebbero essere ancora validi
- L'inserimento di una nuova tupla è possibile solo se esiste almeno l'informazione completa relativa alla chiave primaria
 - non è possibile inserire la parte di tupla relativa ad un solo concetto

Forma normale di Boyce Codd

Normalizzazione

Dipendenza funzionale

- E' un tipo particolare di vincolo d'integrità
- Descrive legami di tipo funzionale tra gli attributi di una relazione
- Esempio: la residenza è unica per ogni studente
 - ogni volta che compare lo stesso studente, il valore è ripetuto
 - il valore di MatrStudente *determina* il valore di Residenza

Esame Superato

<u>MatrStudente</u>	Residenza	<u>CodCorso</u>	NomeCorso	Voto
s94539	Milano	04FLYCY	Calcolatori elettronici	30
s94540	Torino	01FLTCY	Basi di dati	26
s94540	Torino	01KPNCY	Reti di calcolatori	28
s94541	Pescara	01KPNCY	Reti di calcolatori	29
s94542	Lecce	04FLYCY	Calcolatori elettronici	25

Dipendenza funzionale

- Una relazione r soddisfa la dipendenza funzionale $X \rightarrow Y$ se, per ogni coppia t_1, t_2 di tuple di r , aventi gli stessi valori per gli attributi in X , t_1 e t_2 hanno gli stessi valori anche per gli attributi in Y
 - X determina Y (in r)
- Esempi

MatrStudente \rightarrow Residenza

CodCorso \rightarrow NomeCorso

MatrStudente CodCorso \rightarrow NomeCorso

Dipendenza non banale

- La dipendenza

MatrStudiante CodCorso \rightarrow CodCorso

è banale perché CodCorso fa parte di entrambi i lati

- Una dipendenza funzionale $X \rightarrow Y$ è non banale se nessun attributo in X compare tra gli attributi in Y

Dipendenze funzionali e chiavi

- Data una chiave K di una relazione r
 - $K \rightarrow$ qualsiasi altro attributo (o insieme di attributi) di r
- Esempi
 - MatrStudiante CodCorso \rightarrow Residenza
 - MatrStudiante CodCorso \rightarrow NomeCorso
 - MatrStudiante CodCorso \rightarrow Voto

Dipendenze funzionali e anomalie

- Le *anomalie* sono causate da proprietà degli attributi coinvolti in dipendenze funzionali
 - Esempi
 - MatrStudiante \rightarrow Residenza
 - CodCorso \rightarrow NomeCorso
- Le *dipendenze funzionali* dalle chiavi non originano anomalie
 - Esempio
 - MatrStudiante CodCorso \rightarrow Voto

Dipendenze funzionali e anomalie

- Le anomalie sono causate
 - dall'inclusione di concetti indipendenti tra loro nella stessa relazione
 - da dipendenze funzionali $X \rightarrow Y$ che permettono la presenza di più tuple con lo stesso valore di X
 - X non contiene una chiave

Forma normale di Boyce Codd (BCNF)

- BCNF = Boyce Codd Normal Form
- Una relazione r è in BCNF se, per ogni dipendenza funzionale (non banale) $X \rightarrow Y$ definita su di essa, X contiene una chiave di r (X è superchiave di r)
- Anomalie e ridondanze non sono presenti in relazioni in BCNF perché concetti indipendenti sono separati in relazioni diverse

Decomposizione in forma normale

Normalizzazione

Decomposizione in BCNF

- Normalizzazione
 - processo di sostituzione di una relazione non normalizzata con due o più relazioni in BCNF
- Criterio
 - una relazione che rappresenta più concetti indipendenti è decomposta in relazioni più piccole, una per ogni concetto, per mezzo delle dipendenze funzionali
- Le nuove relazioni sono ottenute mediante proiezioni sugli insiemi di attributi corrispondenti alle dipendenze funzionali
- Le chiavi delle nuove relazioni sono le parti sinistre delle dipendenze funzionali
 - le nuove relazioni sono in BCNF

Esempio

- Dipendenze funzionali nell'esempio
 - MatrStudente → Residenza
 - CodCorso → NomeCorso
 - MatrStudente CodCorso → Voto

Esame Superato

<u>MatrStudente</u>	Residenza	<u>CodCorso</u>	NomeCorso	Voto
s94539	Milano	04FLYCY	Calcolatori elettronici	30
s94540	Torino	01FLTCY	Basi di dati	26
s94540	Torino	01KPNCY	Reti di calcolatori	28
s94541	Pescara	01KPNCY	Reti di calcolatori	29
s94542	Lecce	04FLYCY	Calcolatori elettronici	25

Esempio

- Da

R (MatrStudente, Residenza, CodCorso, NomeCorso, Voto)

- Dipendenze funzionali nell'esempio

- MatrStudente \rightarrow Residenza
- CodCorso \rightarrow NomeCorso
- MatrStudente CodCorso \rightarrow Voto

- Le relazioni in BCNF sono

R1 (MatrStudente, Residenza) = $\pi_{\text{MatrStudente, Residenza}}$ R

R2 (CodCorso, NomeCorso) = $\pi_{\text{CodCorso, NomeCorso}}$ R

R3 (MatrStudente, CodCorso, Voto) = $\pi_{\text{MatrStudente, CodCorso, Voto}}$ R

Esempio

R₁

<u>MatrStudente</u>	Residenza
s94539	Milano
s94540	Torino
s94541	Pescara
s94542	Lecce

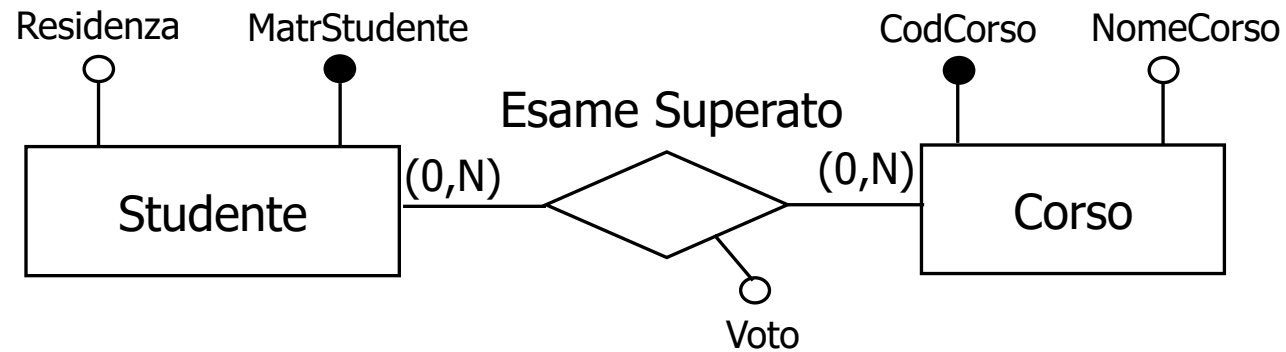
R₂

<u>CodCorso</u>	NomeCorso
04FLYCY	Calcolatori elettronici
01FLTCY	Basi di dati
01KPNCY	Reti di calcolatori

R₃

<u>MatrStudente</u>	<u>CodCorso</u>	Voto
s94539	04FLYCY	30
s94540	01FLTCY	26
s94540	01KPNCY	28
s94541	01KPNCY	29
s94542	04FLYCY	25

Esempio: schema ER corrispondente



Studente (MatrStudente, Residenza)

Corso (CodCorso, NomeCorso)

Esame Superato (MatrStudente, CodCorso, Voto)

Proprietà delle decomposizioni

Normalizzazione

Proprietà delle decomposizioni

- Sono accettabili tutte le decomposizioni?
 - proprietà essenziali per una “buona” decomposizione
- Problemi
 - perdita di informazione
 - perdita delle dipendenze

Esempio

<u>Impiegato</u>	Categoria	Stipendio
Rossi	2	1800
Verdi	3	1800
Bianchi	4	2500
Neri	5	2500
Bruni	6	3500

R (Impiegato, Categoria, Stipendio)

Impiegato → Categoria

Impiegato → Stipendio

Categoria → Stipendio

Decomposizione senza perdita

Normalizzazione

Esempio: decomposizione (n.1)

R (Impiegato, Categoria, Stipendio)

- Decomposizione basata sulle dipendenze funzionali

Impiegato \rightarrow Stipendio

Categoria \rightarrow Stipendio

Esempio: decomposizione (n.1)

R (Impiegato, Categoria, Stipendio)

- Decomponendo

R_1 (Impiegato, Stipendio) =

$\pi_{\text{Impiegato, Stipendio}} R$

<u>Impiegato</u>	Stipendio
Rossi	1800
Verdi	1800
Bianchi	2500
Neri	2500
Bruni	3500

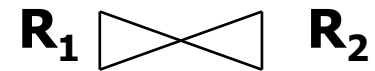
R_2 (Categoria, Stipendio) =

$\pi_{\text{Categoria, Stipendio}} R$

<u>Categoria</u>	Stipendio
2	1800
3	1800
4	2500
5	2500
6	3500

Esempio: decomposizione (n.1)

- Ricomponendo



Impiegato	Categoria	Stipendio
Rossi	2	1800
<i>Rossi</i>	<i>3</i>	<i>1800</i>
<i>Verdi</i>	<i>2</i>	<i>1800</i>
Verdi	3	1800
Bianchi	4	2500
...

← tuple
← "spurie"

- Ricostruzione *con perdita di informazione*

Decomposizione senza perdita

- La decomposizione di una relazione r su due insiemi di attributi X_1 e X_2 è *senza perdita* di informazione se il join delle proiezioni di r su X_1 e X_2 è uguale a r stessa (senza tuple “spurie”)
- Una decomposizione eseguita per normalizzare deve essere senza perdita

Decomposizione senza perdita

- Data la relazione $r(X)$ e gli insiemi di attributi X_1 e X_2 tali che

$$X = X_1 \cup X_2$$

$$X_0 = X_1 \cap X_2$$

se r soddisfa la dipendenza funzionale

$$X_0 \rightarrow X_1 \text{ o } X_0 \rightarrow X_2$$

la decomposizione di r su X_1 e X_2 è senza perdita

- Gli attributi comuni formano una chiave per almeno una delle relazioni decomposte

Esempio: perdita di informazione

R_1 (Impiegato, Stipendio)

R_2 (Categoria, Stipendio)

- Verifica della condizione per la decomposizione senza perdita

$X_1 =$ Impiegato, Stipendio

$X_2 =$ Categoria, Stipendio

$X_0 =$ Stipendio

- L'attributo Stipendio non soddisfa la condizione per la decomposizione senza perdita

Esempio: decomposizione (n.2)

R (Impiegato, Categoria, Stipendio)

- Decomposizione basata sulle dipendenze funzionali

Impiegato \rightarrow Categoria

Impiegato \rightarrow Stipendio

- Decomponendo

R_1 (Impiegato, Categoria) =

$\pi_{\text{Impiegato, Categoria}} R$

<u>Impiegato</u>	Categoria
Rossi	2
Verdi	3
Bianchi	4
Neri	4
Bruni	5

R_2 (Impiegato, Stipendio) =

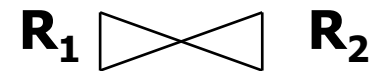
$\pi_{\text{Impiegato, Stipendio}} R$

<u>Impiegato</u>	Stipendio
Rossi	1800
Verdi	1800
Bianchi	2500
Neri	2500
Bruni	3500

Esempio: decomposizione senza perdita?

R_1 (Impiegato, Categoria)

R_2 (Impiegato, Stipendio)



- La decomposizione è *senza perdita*?
- Verifica della condizione per la decomposizione senza perdita
 - $X_1 =$ Impiegato, Categoria
 - $X_2 =$ Impiegato, Stipendio
 - $X_0 =$ Impiegato
- L'attributo Impiegato soddisfa la condizione per la decomposizione senza perdita

Conservazione delle dipendenze

Normalizzazione

Esempio: inserimento di una nuova tupla

R_1 (Impiegato, Categoria)

R_2 (Impiegato, Stipendio)

- Inserimento della tupla

- Impiegato: Gialli – Categoria: 3 – Stipendio: 3500

<u>Impiegato</u>	Categoria
Rossi	2
Verdi	3
Bianchi	4
Neri	4
Bruni	5



Gialli	3
--------	---

<u>Impiegato</u>	Stipendio
Rossi	1800
Verdi	1800
Bianchi	2500
Neri	2500
Bruni	3500



Gialli	3500
--------	------

Esempio: inserimento di una nuova tupla

- Cosa avviene se inserisco la tupla (Gialli,3500) in R_2 ?
 - nella relazione originaria l'inserimento è vietato perché causa la violazione della dipendenza Categoria \rightarrow Stipendio
 - nella decomposizione non è più possibile riconoscere alcuna violazione, poiché gli attributi Categoria e Stipendio sono in relazioni separate
- E' stata persa la dipendenza tra Categoria e Stipendio

Conservazione delle dipendenze

- Una decomposizione conserva le dipendenze se ciascuna delle dipendenze funzionali dello schema originario è presente in una delle relazioni decomposte
- E' opportuno che le dipendenze siano conservate, in modo da garantire che nello schema decomposto siano soddisfatti gli stessi vincoli dello schema originario

Esempio: decomposizione (n.3)

R (Impiegato, Categoria, Stipendio)

- Decomposizione basata sulle dipendenze funzionali

Impiegato \rightarrow Categoria

Categoria \rightarrow Stipendio

- Decomponendo

R_1 (Impiegato, Categoria) =

$\pi_{\text{Impiegato, Categoria}} R$

<u>Impiegato</u>	Categoria
Rossi	2
Verdi	3
Bianchi	4
Neri	4
Bruni	5

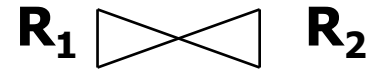
R_2 (Categoria, Stipendio) =

$\pi_{\text{Categoria, Stipendio}} R$

<u>Categoria</u>	Stipendio
2	1800
3	1800
4	2500
5	2500
6	3500

Esempio: decomposizione senza perdita

- Ricomponendo



- Verifica condizione per la decomposizione *senza perdita*

X_1 = Impiegato, Categoria

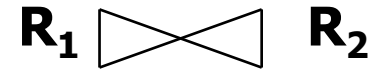
X_2 = Categoria, Stipendio

X_0 = Categoria

- L'attributo Categoria soddisfa la condizione per la decomposizione senza perdita

Esempio: conservazione delle dipendenze funzionali

- Ricomponendo



- Dipendenze funzionali conservate

Impiegato \rightarrow Categoria

Categoria \rightarrow Stipendio

- La dipendenze funzionale

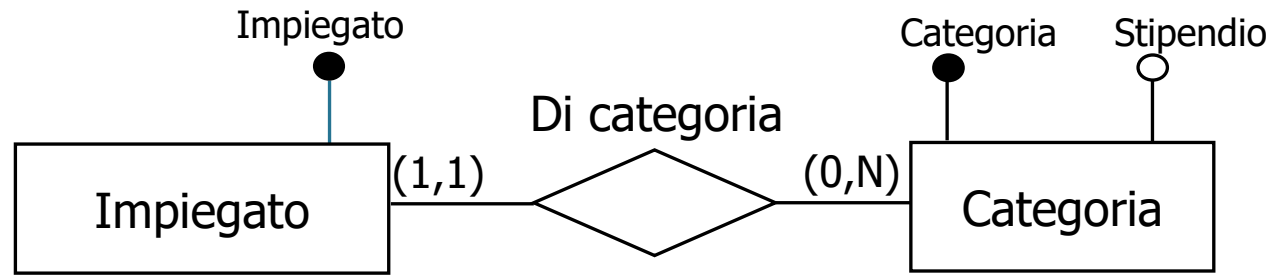
Impiegato \rightarrow Stipendio

può essere ricostruita da

Impiegato \rightarrow Categoria

Categoria \rightarrow Stipendio

Esempio: schema logico relazionale



Impiegato (Impiegato, Categoria)

Categoria (Categoria, Stipendio)

Qualità di una decomposizione

- Le decomposizioni devono sempre soddisfare le proprietà
 - decomposizione senza perdita
 - garantisce che le informazioni nella relazione originaria siano ricostruibili con precisione (senza tuple spurie) a partire da quelle nelle relazioni decomposte
 - conservazione delle dipendenze
 - garantisce che le relazioni decomposte abbiano la stessa capacità della relazione originaria di rappresentare i vincoli di integrità