Progettazione di basi di dati

Normalizzazione

Normalizzazione

- Introduzione
- Forma normale di Boyce Codd
- Decomposizione in forma normale
- Proprietà delle decomposizioni
  - decomposizione senza perdita di informazione
  - decomposizione con conservazione delle dipendenze

Normalizzazione

Introduzione

Normalizzazione

- Uno schema relazionale normalizzato è, *intuitivamente*, uno schema relazionale che permette la memorizzazione di dati senza ridondanze
- 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

Normalizzazione e modello ER

- 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

Relazione (tabella) **Esame Superato**

MatrStudente	Residenza	CodCorso	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: vincoli

MatrStudente	Residenza	CodCorso	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

- La chiave primaria è MatrStudente, CodCorso
- La residenza di ogni studente è unica ed è funzione solo dello studente (non dipende dagli esami che ha superato)
- Il nome di ogni corso *(NomeCorso)* è unico ed è funzione solo del corso (non dipende dagli studenti che superano l'esame di quel corso)

Ridondanza e Anomalie

- In tutte le righe in cui compare uno studente è ripetuta la sua residenza
  - ridondanza

MatrStudente	Residenza	CodCorso	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 e Anomalie

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

MatrStudente	Residenza	CodCorso	NomeCorso	Voto
s94539	Milano	04FLYCY	Calcolatori elettronici	30
s94540	Torino	01FLTCY	Basi di dati	26
s94540	Torino	01KPNCY	Reti di calcolatori	28

MatrStudente	Residenza	CodCorso	NomeCorso	Voto
s94539	Milano	04FLYCY	Calcolatori elettronici	30
s94540	Genova	01FLTCY	Basi di dati	26
s94540	Genova	01KPNCY	Reti di calcolatori	28

Ridondanza e Anomalie

- 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 inserimento

MatrStudente	Residenza	CodCorso	NomeCorso	Voto
s94539	Milano	04FLYCY	Calcolatori elettronici	30
s94541	Pescara	01KPNCY	Reti di calcolatori	29
s94542	Lecce	04FLYCY	Calcolatori elettronici	25
s99999	Venezia	**NULL**	NULL	NULL

Errore!
nessuna componente della chiave primaria può assumere il valore NULL

Ridondanza e Anomalie

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

MatrStudente	Residenza	CodCorso	NomeCorso	Voto
s94539	Milano	04FLYCY	Calcolatori elettronici	30
s94541	Pescara	01KPNCY	Reti di calcolatori	29
s94542	Lecce	04FLYCY	Calcolatori elettronici	25

MatrStudente	Residenza	CodCorso	NomeCorso	Voto
s94539	Milano	04FLYCY	Calcolatori elettronici	30
s94542	Lecce	04FLYCY	Calcolatori elettronici	25

Dove risiede lo studente s94541?

Informazioni eterogenee

- Un'unica relazione è utilizzata per rappresentare informazioni eterogenee

MatrStudente	Residenza	CodCorso	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

Anomalie in sintesi

- L'aggiornamento delle informazioni ridondanti deve essere effettuato contemporaneamente per tutte
- 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

Normalizzazione

Forma normale di Boyce Codd

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

MatrStudente	Residenza	CodCorso	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

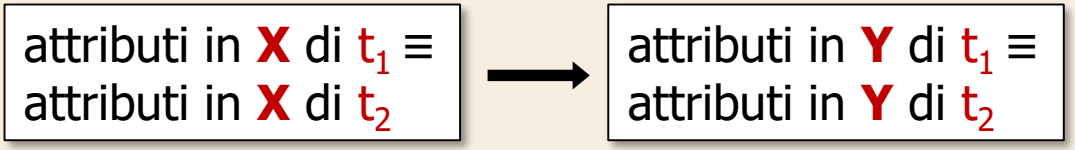
Dipendenza funzionale

- Se
  - **r** è una relazione
  - **X** e **Y** sono due insiemi di attributi di **r**

  si dice che **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 tutti gli attributi in **X**,

  t_1 e t_2 hanno gli stessi valori per gli attributi in **Y**



- In questo caso si dice anche che

  X determina **Y** (in **r**)

Dipendenza funzionale

- Esempio: dipendenza funzionale in **Esame Superato**

MatrStudente → Residenza

MatrStudente	Residenza	CodCorso	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
↑ **X**	↑ **Y**			

Dipendenza funzionale

- Esempio: dipendenza funzionale in **Esame Superato**

CodCorso → NomeCorso

MatrStudente	Residenza	CodCorso	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
		↑ X	↑ Y	

Dipendenza funzionale

- Esempio: dipendenza funzionale in **Esame Superato**

Residenza, CodCorso → NomeCorso

MatrStudente	Residenza	CodCorso	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
	↑ X		↑ Y	

Dipendenza non banale

- La dipendenza

  MatrStudente 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 → qualsiasi altro attributo (o insieme di attributi) di **r**

- Esempi:

  MatrStudente CodCorso
  è la chiave primaria di **Esame Superato**

  - MatrStudente CodCorso → Residenza
  - MatrStudente CodCorso → NomeCorso
  - MatrStudente CodCorso → Voto

Dipendenze funzionali e anomalie

- Le *anomalie* sono causate da proprietà degli attributi coinvolti in dipendenze funzionali
  - *Esempi*
    - MatrStudente → Residenza
    - CodCorso → NomeCorso
- Le *dipendenze funzionali* dalle chiavi non originano anomalie
  - *Esempio*
    - MatrStudente CodCorso → 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**

    ossia

    da dipendenze funzionali **X** $\rightarrow$ **Y** dove

    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

Normalizzazione

Decomposizione in forma normale

Decomposizione in BCNF

- Normalizzazione
  - processo di sostituzione di una relazione non normalizzata con due o più relazioni in BCNF
- Procedimento
  - una relazione che rappresenta più concetti indipendenti è decomposta in relazioni più piccole, una per ogni concetto, per mezzo delle dipendenze funzionali

Decomposizione in BCNF

- 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

Esame Superato

MatrStudente	Residenza	CodCorso	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

- Dipendenze funzionali (non banali) nell'esempio
  - MatrStudente $\rightarrow$ Residenza
  - CodCorso $\rightarrow$ NomeCorso
  - MatrStudente CodCorso $\rightarrow$ Voto

Esempio

- Da

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

  si ricavano le seguenti relazioni in BCNF:

  $\mathbf{R}_1$ (MatrStudente, Residenza) = $\pi_{\text{MatrStudente, Residenza}}$ **R**

  $\mathbf{R}_2$ (CodCorso, NomeCorso) = $\pi_{\text{CodCorso, NomeCorso}}$ **R**

  $\mathbf{R}_3$ (MatrStudente, CodCorso, Voto) = $\pi_{\text{MatrStudente, CodCorso, Voto}}$ **R**

Esempio

R

MatrStudente	Residenza	CodCorso	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

R_1

MatrStudente	Residenza
s94539	Milano
s94540	Torino
s94541	Pescara
s94542	Lecce

$+ R_2 + R_3$

Esempio

R

MatrStudente	Residenza	CodCorso	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

R_1 +

R_2

CodCorso	NomeCorso
04FLYCY	Calcolatori elettronici
01FLTCY	Basi di dati
01KPNCY	Reti di calcolatori

+ R_3

Esempio

R

MatrStudente	Residenza	CodCorso	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

$R_1 + R_2 +$

R_3

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

Esempio

R

MatrStudente	Residenza	CodCorso	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

R_1

MatrStudente	Residenza
s94539	Milano
s94540	Torino
s94541	Pescara
s94542	Lecce

R_2

CodCorso	NomeCorso
04FLYCY	Calcolatori elettronici
01FLTCY	Basi di dati
01KPNCY	Reti di calcolatori

R_3

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

Normalizzazione

Proprietà delle decomposizioni

Proprietà delle decomposizioni

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

Esempio

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

Relazione: (Impiegato, Categoria, Stipendio)

Dipendenze: Impiegato → Categoria
Impiegato → Stipendio
Categoria → Stipendio

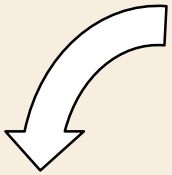
Normalizzazione

Decomposizione senza perdita di informazione

Esempio: decomposizione

Decomposizione secondo le dipendenze funzionali indicate:

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



Impiegato → Stipendio

Impiegato	Stipendio
Rossi	1800
Verdi	1800
Bianchi	2500
Neri	2500
Bruni	3500

Categoria → Stipendio

Categoria	Stipendio
2	1800
3	1800
4	2500
5	2500
6	3500

Esempio: ricomposizione

Ricomposizione per il recupero dell'informazione originale (join):

Impiegato	Stipendio
Rossi	1800
Verdi	1800
Bianchi	2500
Neri	2500
Bruni	3500

⋈

Categoria	Stipendio
2	1800
3	1800
4	2500
5	2500
6	3500

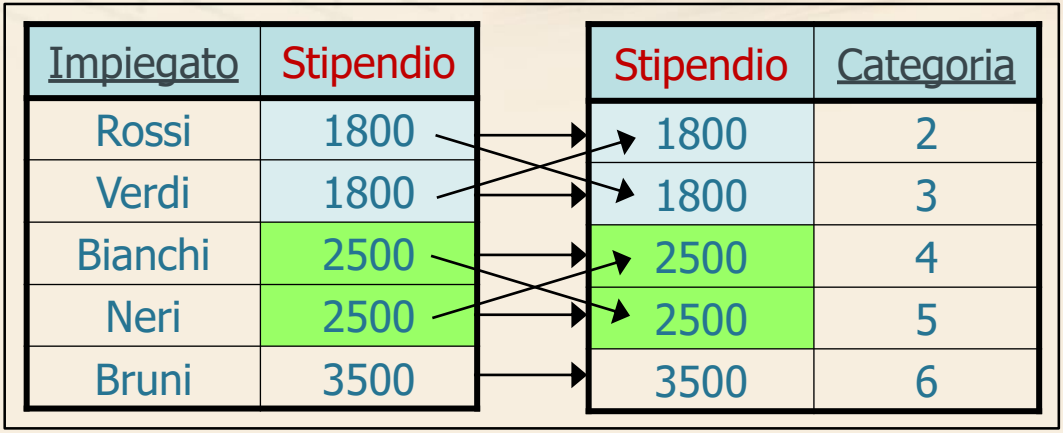
Il join viene eseguito sulla colonna comune (Stipendio):

Impiegato	Stipendio
Rossi	1800
Verdi	1800
Bianchi	2500
Neri	2500
Bruni	3500

Stipendio	Categoria
1800	2
1800	3
2500	4
2500	5
3500	6

Esempio: ricomposizione

Il join viene eseguito sulla colonna comune (Stipendio):



Impiegato	Stipendio
Rossi	1800
Verdi	1800
Bianchi	2500
Neri	2500
Bruni	3500

Stipendio	Categoria
1800	2
1800	3
2500	4
2500	5
3500	6

Risultato della ricomposizione (join):

Impiegato	Categoria	Stipendio
Rossi	2	1800
Rossi	**3**	1800
Verdi	3	1800
Verdi	**2**	1800
. . .	. . .	. . .
. . .	. . .	. . .

tuple *spurie*

Esempio: ricomposizione

Risultato completo della ricomposizione:

Impiegato	Categoria	Stipendio
Rossi	2	1800
~~Rossi~~	~~3~~	~~1800~~
~~Verdi~~	~~2~~	~~1800~~
Verdi	3	1800
Bianchi	4	2500
~~Bianchi~~	~~5~~	~~2500~~
~~Neri~~	~~4~~	~~2500~~
Neri	5	2500
Bruni	6	3500

- Ricostruzione *con perdita di informazione*

Decomposizione senza perdita

- La decomposizione di una relazione **r** su due insiemi di attributi $\mathbf{X_1}$ e $\mathbf{X_2}$ è *senza perdita di informazione* se il join delle proiezioni di **r** su $\mathbf{X_1}$ e $\mathbf{X_2}$ è uguale a **r** stessa (senza tuple *spurie*)

$$\pi_{\mathbf{X_1}}\mathbf{r} \bowtie \pi_{\mathbf{X_2}}\mathbf{r} \equiv \mathbf{r}$$

- Una decomposizione eseguita per normalizzare uno schema *non deve comportare la perdita di informazione*

Decomposizione senza perdita

- Data la relazione **r**(**X**)
  e gli insiemi di attributi $\mathbf{X_1}$ e $\mathbf{X_2}$ tali che

  $\mathbf{X} \equiv \mathbf{X_1} \cup \mathbf{X_2}$ (➔ **X** ≡ insieme degli attributi di **r**)

  $\mathbf{X_0} \equiv \mathbf{X_1} \cap \mathbf{X_2}$ (➔ $\mathbf{X_0}$ ≡ attributi comuni a $\mathbf{X_1}$, $\mathbf{X_2}$)

  se **r** soddisfa una delle due dipendenze funzionali

  $\mathbf{X_0} \rightarrow \mathbf{X_1}$ *oppure* $\mathbf{X_0} \rightarrow \mathbf{X_2}$

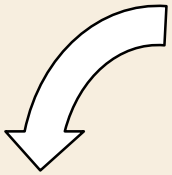
  la decomposizione di **r** su $\mathbf{X_1}$ e $\mathbf{X_2}$ è *senza perdita*

- In questo caso gli attributi comuni $\mathbf{X_0}$ formano una chiave per almeno una delle due relazioni risultato della decomposizione

Esempio: decomposizione con perdita

Decomposizione
secondo le dipendenze funzionali indicate:

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



Impiegato → Stipendio

Impiegato	Stipendio
Rossi	1800
Verdi	1800
Bianchi	2500
Neri	2500
Bruni	3500

Categoria → Stipendio

Categoria	Stipendio
2	1800
3	1800
4	2500
5	2500
6	3500

Esempio: decomposizione con perdita

- Verifica decomposizione senza/con perdita

Impiegato	Stipendio
Rossi	1800
Verdi	1800
Bianchi	2500
Neri	2500
Bruni	3500

Categoria	Stipendio
2	1800
3	1800
4	2500
5	2500
6	3500

$X_1 \equiv$ Impiegato, Stipendio $X_2 \equiv$ Categoria, Stipendio

$X_0 \equiv X_1 \cap X_2 \equiv$ Stipendio

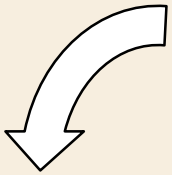
$X_0 \rightarrow X_1$ *oppure* $X_0 \rightarrow X_2$: **falso**

➔ decomposizione con perdita di informazione

Esempio: decomposizione senza perdita

Decomposizione secondo le dipendenze funzionali indicate:

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



Impiegato → Categoria

Impiegato	Categoria
Rossi	2
Verdi	3
Bianchi	4
Neri	5
Bruni	6

Impiegato → Stipendio

Impiegato	Stipendio
Rossi	1800
Verdi	1800
Bianchi	2500
Neri	2500
Bruni	3500

Esempio: decomposizione senza perdita

- Verifica decomposizione con/senza perdita

Impiegato	Categoria
Rossi	2
Verdi	3
Bianchi	4
Neri	5
Bruni	6

Impiegato	Stipendio
Rossi	1800
Verdi	1800
Bianchi	2500
Neri	2500
Bruni	3500

$X_1 \equiv$ Impiegato, Categoria $X_2 \equiv$ Impiegato, Stipendio

$X_0 \equiv X_1 \cap X_2 \equiv$ Impiegato

$X_0 \rightarrow X_1$ *oppure* $X_0 \rightarrow X_2$: **vero**

➔ decomposizione senza perdita di informazione

Normalizzazione

Decomposizione con conservazione delle dipendenze

Esempio senza conservazione delle dipendenze

Dipendenze:

Impiegato → Categoria
Impiegato → Stipendio
Categoria → Stipendio

Inserimento di una tupla

Impiegato	Categoria	Stipendio
Rossi	2	1800
Verdi	3	1800
Gialli	3	3500
Bianchi	4	2500
. . .	. . .	. . .

Impiegato → Categoria

Impiegato	Categoria
Rossi	2
Verdi	3
Gialli	3
Bianchi	4
. . .	. . .

Impiegato → Stipendio

Impiegato	Stipendio
Rossi	1800
Verdi	1800
Gialli	3500
Bianchi	2500
. . .	. . .

Esempio senza conservazione delle dipendenze

Dipendenze:

Impiegato → Categoria
Impiegato → Stipendio
Categoria → Stipendio

Inserimento di una tupla

Impiegato	Categoria	Stipendio
Rossi	2	1800
Verdi	3	1800
Gialli	~~3~~	~~3500~~
Bianchi	4	2500
. . .	. . .	. . .

- Nella relazione originale l'inserimento è vietato perché viola la dipendenza Categoria → Stipendio

Esempio senza conservazione delle dipendenze

Impiegato → Categoria

Impiegato	Categoria
Rossi	2
Verdi	3
Gialli	3
Bianchi	4
. . .	. . .

Impiegato → Stipendio

Impiegato	Stipendio
Rossi	1800
Verdi	1800
Gialli	3500
Bianchi	2500
. . .	. . .

- 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

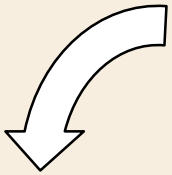
Conservazione delle dipendenze

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

Esempio corretto

Decomposizione secondo le dipendenze funzionali indicate:

Impiegato	Categoria	Stipendio
Rossi	2	1800
Verdi	3	1800
Bianchi	4	2500
Neri	5	2500
Bruni	5	2500



Impiegato → Categoria

Impiegato	Categoria
Rossi	2
Verdi	3
Bianchi	4
Neri	5
Bruni	5

Categoria → Stipendio

Categoria	Stipendio
2	1800
3	1800
4	2500
5	2500

Esempio corretto

- Verifica decomposizione con/senza perdita

Impiegato	Categoria
Rossi	2
Verdi	3
Bianchi	4
Neri	5
Bruni	5

Categoria	Stipendio
2	1800
3	1800
4	2500
5	2500

X_1 ≡ Impiegato, Categoria X_2 ≡ Categoria, Stipendio

$X_0 \equiv X_1 \cap X_2 \equiv$ Categoria

$X_0 \rightarrow X_1$ *oppure* $X_0 \rightarrow X_2$: **vero**

➔ decomposizione senza perdita di informazione

Esempio corretto

- Verifica conservazione delle dipendenze

Impiegato → Categoria

Impiegato	Categoria
Rossi	2
Verdi	3
Bianchi	4
Neri	5
Bruni	5

Categoria → Stipendio

Categoria	Stipendio
2	1800
3	1800
4	2500
5	2500

Impiegato → Stipendio è conservata in quanto implicata da

Impiegato → Categoria, Categoria → Stipendio

➔ decomposizione con conservazione delle dipendenze

Esempio corretto: schemi corrispondenti

Impiegato (Impiegato, Categoria)
Categoria (Categoria, Stipendio)

In sintesi: 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

Normalizzazione: è sempre possibile?

- Esempio di relazione non normalizzabile con il metodo di scomposizione

Venditore	Prodotto	Zona
Gatti	Scarpe	Valle d'Aosta
Gatti	Sandali	Valle d'Aosta
Lupi	Scarpe	Riviera ligure
Volpi	Scarponi	Valle d'Aosta
Volpi	Zoccoli	Valle d'Aosta

- Vincoli / dipendenze:
  - ciascun venditore opera in un'unica zona:
    Venditore → Zona
  - in ogni zona esiste solo un venditore che vende un dato prodotto:
    Prodotto Zona → Venditore

Relazione non decomponibile

R (Venditore, Prodotto, Zona)

con dipendenze

Venditore → Zona, Prodotto Zona → Venditore

⬇

R *non è* in forma normale poiché

Venditore non contiene la chiave Venditore Prodotto

⬇

R non è decomponibile poiché la dipendenza

Prodotto Zona → Venditore genera una nuova relazione

(Prodotto, Zona, Venditore) con gli stessi attributi di **R**

e quindi nuovamente *non in forma normale*