

# Normalizzazione (Codd, 1972)

- La normalizzazione non è una tecnica, nè una metodologia di progettazione
- Le forme normali costituiscono uno dei criteri per ottenere basi di dati relazionali “ben progettate”
- Non necessariamente uno schema normalizzato è “ben progettato”
- Intuitivamente, cosa significa “ben progettato?”
  - eliminazione o limitazione delle ridondanze
  - ortogonalità e indipendenza
  - assenza di anomalie

## Esempio: biblioteca

- Biblioteca{Utente, Indirizzo, Telefono, Codice Materiale, Autore, Titolo, DataPrestito}

Utente	Indirizzo	Tel.	Codice Mat.	Autore	Titolo	Data Prestito
Rossi	Venezia	75444	XY188A	Boccaccio	Decameron	07/07
Paolicchi	Pisa	25346	XX256B	Verdi	Traviata	07/08
Poli	Udine	59729	XY090C	Verga	Novelle	17/09
Paolicchi	Pisa	25346	KK101A	Petrarca	Canzoniere	20/09
Paolicchi	Pisa	25346	HJ700A	Dante	Vita Nova	21/09
Bianchi	Udine	66123	XY008B	Manzoni	Adelchi	03/10
				Moravia	La Noia	

- Ripetizioni
  - L'indirizzo viene ripetuto
  - Ridondanza
  - Probabile inconsistenza
- Anomalie di inserzione
  - Indirizzo e n. di telefono non possono essere inseriti se l'utente non ha prestiti
- Anomalie di eliminazione
  - Indirizzo e n. di telefono spariscono se si eliminano i prestiti

# Cosa non deve succedere

- **Ridondanza**
  - non dovrebbero esistere informazioni ripetute
- **Anomalie di aggiornamento**
  - se varia un dato non deve essere necessario andare a cercare tutte le ennuple che devono essere aggiornate
- **Anomalie di cancellazione**
  - la cancellazione di una ennupla non deve lasciare altre ennuple non integre, né eliminare qualunque traccia di sé
- **Anomalie di inserimento**
  - l'inserimento di una nuova ennupla non deve causare problemi di relazione

## Altro esempio

- Lo stipendio dipenderebbe solo dall'impiegato...
- Il bilancio dipenderebbe solo dal progetto...

<b>Impiegato</b>	<b>Stipendio</b>	<b>Progetto</b>	<b>Bilancio</b>	<b>Funzione</b>
Rossi	20000000	Marte	2000	tecnico
Verdi	35000000	Giove	15000	progettista
Verdi	35000000	Venere	15000	progettista
Neri	55000000	Venere	15000	direttore
Neri	55000000	Giove	15000	consulente
Neri	55000000	Marte	2000	consulente
Mori	48000000	Marte	2000	direttore
Mori	48000000	Venere	15000	progettista
Bianchi	48000000	Venere	15000	progettista
Bianchi	48000000	Giove	15000	direttore

# Cosa non deve succedere

- **Ridondanza**
  - lo stipendio è inutilmente ripetuto
- **Anomalie di aggiornamento**
  - una variazioni di stipendio comporta una variazione di numerose ennuple
- **Anomalie di cancellazione**
  - se un impiegato non partecipa ad alcun progetto cosa succede?
- **Anomalie di inserimento**
  - non è possibile gestire informazioni relative ad un impiegato se esso non è assegnato ad almeno un progetto

# Dipendenza funzionale

- si parla di dipendenza funzionale  $X \rightarrow Y$  se
$$\forall t_1, t_2 / t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y]$$
- se  $X \rightarrow Y$  è soddisfatta, si può dire che  $X$  *determina*  $Y$
- ad ogni chiave  $K$ , corrisponde una dipendenza funzionale da  $K$  verso altri attributi della relazione di cui  $K$  è chiave

## Dipendenza e anomalie

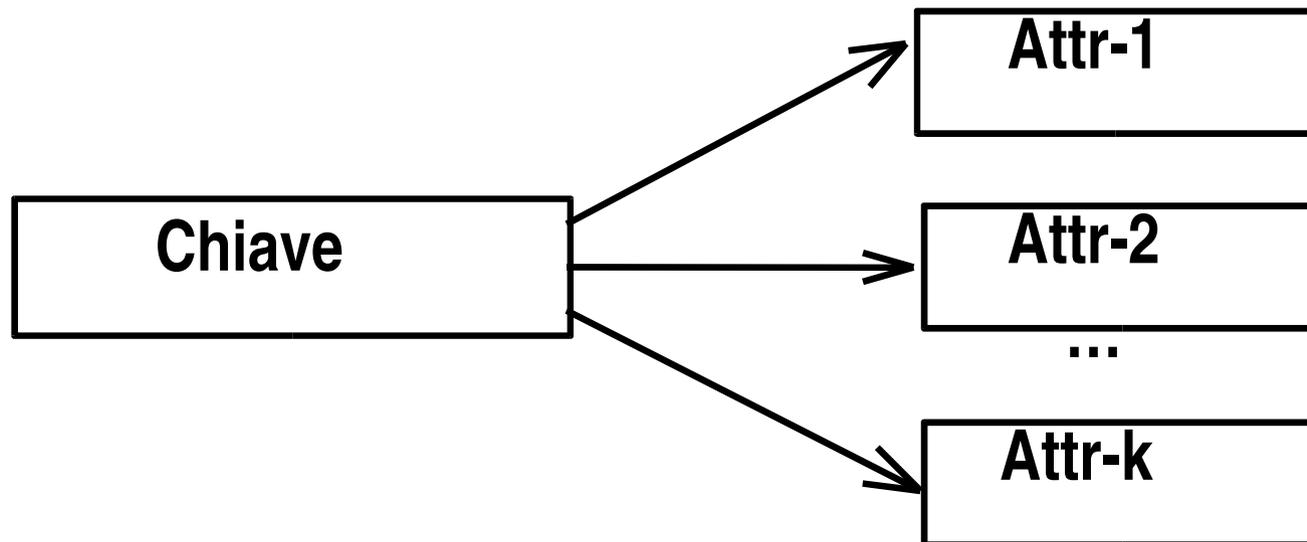
- le anomalie si verificano di solito in corrispondenza di dipendenze funzionali dipendenti da chiavi
- le anomalie si verificano in relazioni che considerano collezioni eterogenee di oggetti

## Esempio "Biblioteca"

- Codice  $\rightarrow$  Autore, Titolo
  - Il codice "determina" l'autore e il titolo
  - L'autore e il titolo "dipendono funzionalmente" dal codice
- $X \rightarrow Y$  quando per ogni valore di  $X$  c'è un solo valore possibile di  $Y$ 
  - $Y$  è *chiave primaria*
  - se  $X$  è chiave determina tutti gli altri attributi nella relazione

- Chiave candidata
  - insieme di attributi che identifica univocamente una ennupla in una relazione
- Chiave primaria
  - una delle chiavi candidate, generalmente quella con meno attributi (idealmente un solo attributo)
- Chiave esterna
  - attributo che è chiave primaria in un'altra relazione

# L'obiettivo



# Decomposizione

Comunemente, una relazione non è in forma normale quando comprende concetti eterogenei

La soluzione consiste nella separazione di tali concetti in due (o più) nuove relazioni, decomponendo la relazione originale in nuove relazioni che soddisfino i criteri di dipendenza funzionale

Sono accettabili solo e soltanto le decomposizioni senza perdita

## Decomposizione senza perdita

- si ha decomposizione senza perdita su due insiemi di attributi  $A_1$  e  $A_2$  qualora un'operazione di join delle proiezioni  $A_1$   $A_2$  della relazione è identico alla relazione originaria (cioè, ricomponendo non si generano ennuple spurie)
- sia  $A_0 = A_1 \cap A_2$ . Si ha decomposizione senza perdita se la relazione soddisfa  $A_0 \rightarrow A_1$  o  $A_0 \rightarrow A_2$  o entrambe
- si ha decomposizione senza perdita se gli attributi in comune tra  $A_1$  e  $A_2$  costituiscono una chiave per almeno una delle due nuove relazioni
- una decomposizione conserva le dipendenze se ciascuna delle dipendenze originarie è presente nelle nuove relazioni

# Uso della decomposizione

A partire da uno schema denormalizzato, lo si porta in forma normale applicando decomposizioni successive tali che

- non vi sia perdita d'informazione
- le dipendenze funzionali siano preservate
- il nuovo schema sia in forma normale

**Prima forma normale (1NF)**

**Seconda forma normale (2NF)**

**Terza forma normale (3NF)**

**Forma normale di  
Boyce-Codd (BCNF)**

# Prima forma normale

- Sono ammissibili come attributi solo e soltanto valori atomici
  - non è possibile definire attributi che abbiano più di un valore
  - un attributo deve essere semplice e indivisibile
  - un attributo deve consistere di un solo valore, tra tutti i valori possibili definiti a livello di dominio per quell'attributo
  - non sono possibili relazioni entro relazioni
  - non sono possibili ennuple di ennuple

## Per portare in 1NF

- Decomporre l'attributo che viola 1NF in una nuova relazione comprendente anche la chiave primaria
  - $t_0[K, A_1, A_2, A_z, \dots, A_n] \Rightarrow t_1[K, A_1, A_2, \dots, A_n] ; t_2[K, A_z]$
  - si decompone una relazione non-1NF in due relazioni 1NF
- Espandere la chiave in modo da avere ennuple separate
  - si genera ridondanza
- Se è noto *a priori* il numero massimo di ripetizioni possibili per un attributo A, decomporre A in più attributi
  - si genera ridondanza
  - si generano numerose istanze di schema con valori nulli

# Esempio

Genere	Specie	Sito
Pimpliduchia	fulvoralis	A
Nasta	foetentissima	A, C

Genere	Specie	Sito
Pimpliduchia	fulvoralis	A
Nasta	foetentissima	A
Nasta	foetentissima	C

Genere	Specie	ID	ID Specie	Sito
Pimpliduchia	fulvoralis	1	1	A
Nasta	foetentissima	2	2	A
			2	C

Sito contiene un attributo non atomico

- La seconda ennupla viene replicata, rendendo atomico l'attributo
- La relazione viene decomposta

La migliore soluzione è la seconda

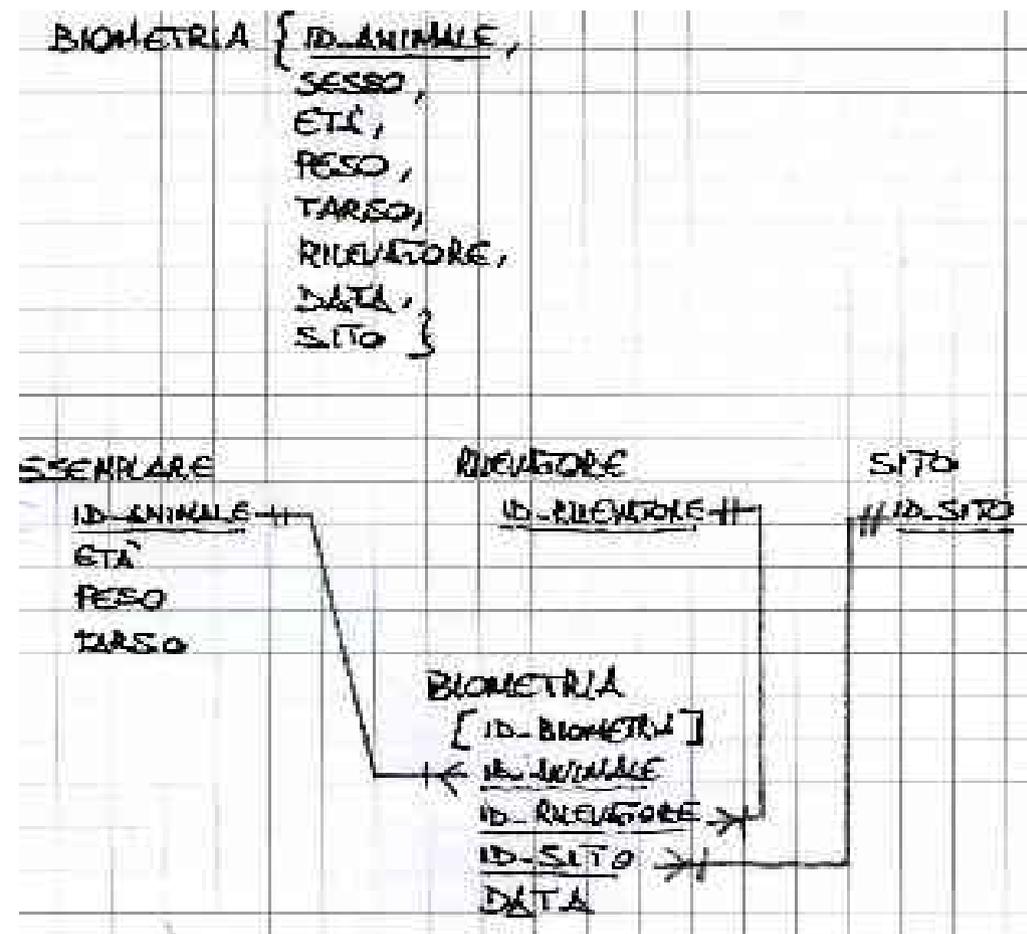
## Seconda forma normale

- Gli attributi soddisfano 1NF
- Ogni attributo che non è chiave, dipende in maniera completa dalla chiave (FFD: *Full Functional Dependency*)
  - ciascun attributo non-chiave deve essere funzionalmente dipendente dalla chiave primaria e solo da essa
  - test: verificare l'esistenza di dipendenze funzionali il cui termine di sinistra sia parte della chiave primaria
    - se la chiave primaria è atomica il test non è necessario

## Per portare in 2NF

- Decomporre la relazione che viola 2NF in n nuove relazioni comprendenti sottoinsiemi di attributi omogenei per dipendenza funzionale dalla chiave primaria originale o da una parte di essa (se la chiave primaria è anche superchiave)
  - $t_0[K, A_1, A_2, A_x, A_y, A_z, \dots, A_n] \Rightarrow$   
 $t_1[K, A_1, A_2, \dots, A_n]; t_2[K, A_x]; t_3[K, A_y]; t_2[K, A_z]$
  - si decompone una relazione non-2NF con n diverse dipendenze funzionali dalla chiave primaria K in n relazioni 2NF

# Esempio



Sesso → ID\_animale

Età → ID\_animale

Peso → ID\_animale

Tarso → ID\_animale

rilevatore, data e sito non dipendono da ID\_animale

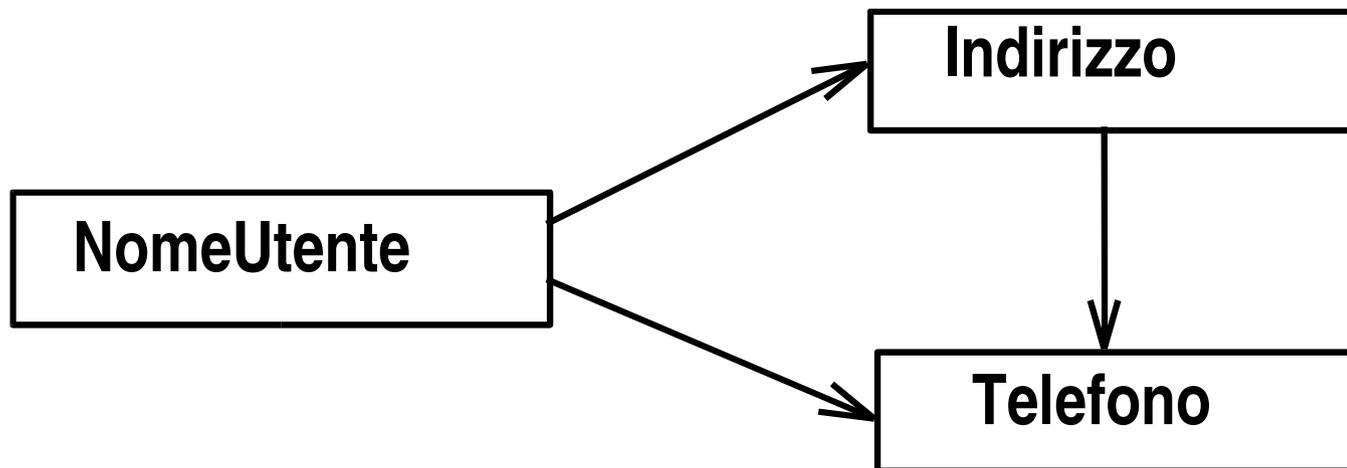
gli attributi relativi a rilevatore, data e sito vengono decomposti

# Eseminare le dipendenze

- Analizzando le dipendenze funzionali è possibile capire come decomporre la relazione
  - NomeRilevatore → Indirizzo, Telefono
  - CodiceMateriale → NomeMateriale, Descrizione
  - NomeRilevatore, DataPrestito → CodiceMateriale
- Schemi risultanti
  - Rilevatore{NomeRilevatore, Indirizzo, Telefono}
  - Materiale{CodiceMateriale, NomeMateriale, Descrizione}
  - Prestito{NomeRilevatore, DataPrestito, CodiceMateriale}

# Terza forma normale

- Gli attributi soddisfano 2NF
- Ogni attributo che non è chiave, non ha dipendenze funzionali da altri attributi non chiave
- Assenza di dipendenze transitive
  - $X \rightarrow Z$  in maniera transitiva se  $X \rightarrow Y$  e  $Y \rightarrow Z$



## 3NF - Esempio

Schema:

Utenti{NomeUtente, Indirizzo, Telefono}

3NF:

Utenti{NomeUtente, Indirizzo}

Telefoni{Indirizzo, Telefono}

1NF: attributi atomici,  
relazioni non nidificate

2NF: in caso di chiave  
primaria multipla, nessun  
attributo non-chiave deve  
dipendere da una parte  
della chiave

3NF: non devono esistere  
attributi non-chiave  
determinati da altri  
attributi non-chiave

Formare nuove relazioni per  
ogni attributo non  
atomico o nidificato

Decomporre per ogni chiave  
parziale, mantenere le  
relazioni con la chiave  
primaria originale

Decomporre creando  
relazioni che includono  
come chiave gli attributi  
non-chiave determinanti

# Forma normale di Boyce-Codd

- La relazione soddisfa 1NF
- Ogni determinante è chiave candidata
  - un determinante è un attributo che determina altri attributi
- 3NF, e tutti i determinanti sono chiavi
  - determinante: attributo che determina altri attributi

CIC{Capoluogo, Indirizzo, CAP}

Capoluogo, Indirizzo  $\rightarrow$  CAP

CAP  $\rightarrow$  Capoluogo

- CAP non è chiave, ma determina Capoluogo

È una condizione più restrittiva rispetto a 3NF

Non sempre è possibile arrivare a una relazione in BCNF

# Forma normale di Boyce-Codd - Esempio

IC(Indirizzo, CAP)

CC(CAP, Capoluogo)

Esistono altre forme normali...

Elmasri & Navathe (2004): 1NF, 2NF, 3NF, BCNF, 4NF,  
5NF, DKNF