

# **La normalizzazione**

di Roberta Molinari

# La normalizzazione

## Introduzione

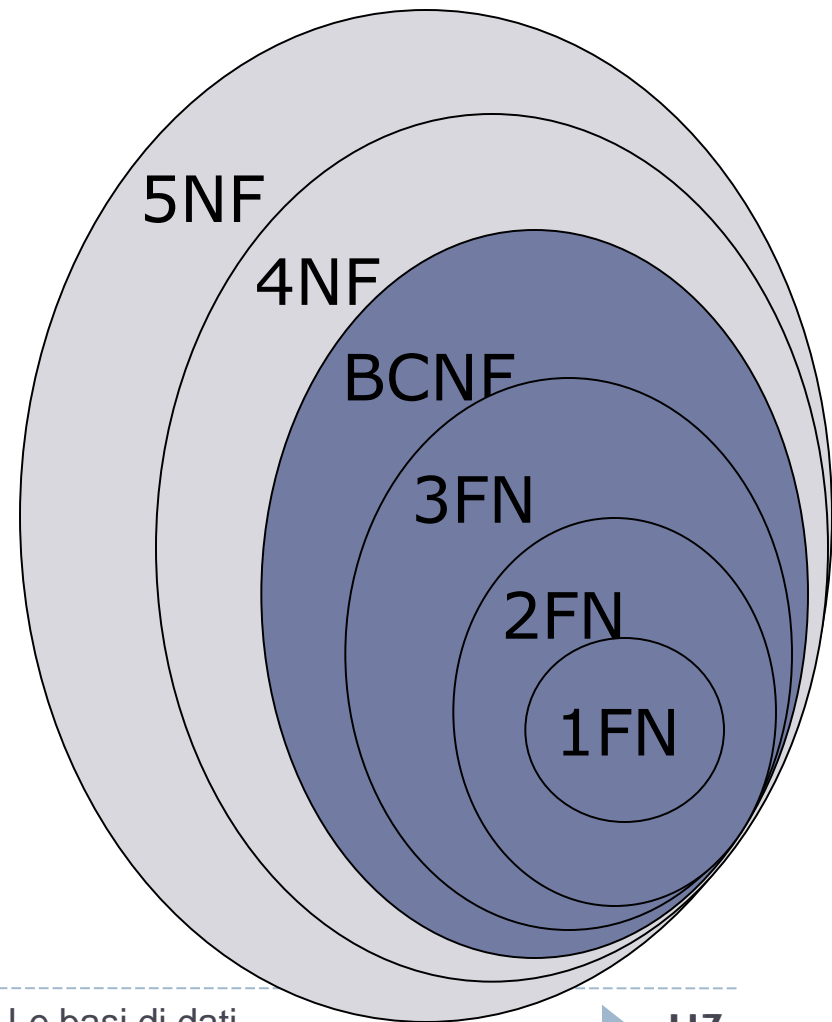
---

- ▶ È un procedimento di analisi e scomposizione di tabelle in più tabelle al fine di eliminare la ripetizione o ridondanza di dati, senza perdita di informazioni.
- ▶ A partire da delle relazioni definite a livello logico, ne crea delle altre corrispondenti a un livello di **forma normale** via via crescente (dalla prima si passa alla seconda e così via).
- ▶ L'aumento del numero delle tabelle a livello fisico rallenta l'aggiornamento e il reperimento dei dati, ma garantisce l'integrità degli stessi

# La normalizzazione

## Introduzione

- ▶ Le FN di ordine superiore contengono la stessa quantità di informazioni di quelle inferiori
- ▶ Solo la 1NF è richiesta dal modello relazionale ed è sufficiente la 3NF
- ▶ La 4FN e la 5FN servono a risolvere problemi legati alla presenza di attributi multivalore e a rendere minimo il numero degli attributi delle chiavi composte



# La normalizzazione

## Concetti base

**Principio di minimalità:** si sceglie quella con il minor numero di campi o che occupi meno spazio in memoria

- ▶ **Chiave candidata:** insieme minimo (non si considerano i sovrainsiemi, vedi superchiave) di attributi che identificano univocamente una tupla (ce ne possono essere molte)
- ▶ **Chiave primaria:** chiave candidata eletta a primaria secondo un principio di minimalità (ce n'è una sola)
- ▶ **Attributo non-chiave o non-primo:** campo che non fa parte di nessuna chiave primaria o candidata
- ▶ **Chiave :** chiave primaria o candidata
- ▶ **Superchiave o Sovrachiave:** chiave o soprainsieme di chiave

# La normalizzazione

## Dipendenza

---

- ▶ Esiste la **dipendenza funzionale FD** tra A e B se il valore di B dipende dal valore di A (che è il suo **determinante**). Ovvero se ad ogni valore della colonna A corrisponde un solo valore nella colonna B.
- ▶ Si indica così  $A \rightarrow B$  (Es.  $CAP \rightarrow Città$ )
- ▶ Ovviamente tutti gli attributi sono funzionalmente dipendenti dalla chiave primaria o dalle chiavi candidate o dalle superchiavi
- ▶ Se il Y è composto e se  $Z \subseteq Y$  allora è sempre vero  $Y \rightarrow Z$ . Se  $A \rightarrow B$  e  $A \subseteq C$  allora  $C \rightarrow B$  (dipendenze banali)
- ▶ Se  $A \rightarrow B$  e  $B \rightarrow C$  allora  $A \rightarrow C$ , cioè C **dipende transitivamente** da A, con B non chiave

# La normalizzazione

## Quali problemi risolve

---

La ripetizione dei dati e la dipendenza creano:

- ▶ spreco di spazio
- ▶ **anomalie di aggiornamento**
  - ▶ **di modifica:** se modifico un valore ripetuto o determinante, lo devo modificare in tutte le occorrenze
  - ▶ **di cancellazione:** se cancello un valore ripetuto o determinante, lo devo cancellare in tutte le occorrenze. Inoltre potrei perdere informazioni
  - ▶ **di inserimento:** potrei non poter aggiungere delle nuove informazioni di campi dipendenti

# La normalizzazione

## Esempio

---

### Esempio di relazione non normalizzata

Iscritti	CodiceCorso	Corso
Rossi Mario	1111	matematica
Verdi Lucia	1111	matematica
Bianchi Ugo	1212	logica

- ▶ Se cambiasse il codice dei corsi devo cambiarlo in tutte le tuple in cui compare
- ▶ Se elimino Bianchi Ugo perdo le informazioni sul corso di logica
- ▶ Per aggiungere un nuovo corso devo aver almeno un iscritto

# La normalizzazione

---

- ▶ Si vuole fare in modo che all'interno delle tabelle non ci siano dipendenze, se non quelle con la PK.
- ▶ Per far ciò (a partire dalla 2NF) si decompongono le tabelle iniziali in tabelle più piccole attraverso proiezioni.
- ▶ La decomposizione deve essere senza perdita
  - ▶ di **informazioni** (si deve poter riottenere le tabelle iniziali attraverso dei join naturali)
  - ▶ di **dipendenze** (gli attributi coinvolti nella dipendenza iniziale devono comparire tutti insieme in uno degli schemi decomposti)ovvero deve permettere di ricostruire esattamente la relazione originaria



# La normalizzazione

## Problemi di perdita

<u>Impiegato</u>	<u>Progetto</u>	Sede
Rossi	Marte	Roma
Verdi	Giove	Milano
Verdi	Venere	Milano
Neri	Saturno	Milano
Neri	Venere	Milano

Un impiegato deve operare su una sola sede e anche i progetti devono insistere su una sola sede

*Impiegato → Sede*

*Progetto → Sede*



# La normalizzazione

## Problemi di perdita

<u>Impiegato</u>	Sede		<u>Progetto</u>	Sede
Rossi	Roma		Marte	Roma
Verdi	Milano		Giove	Milano
Neri	Milano		Saturno	Milano
			Venere	Milano

<u>Impiegato</u>	<u>Progetto</u>	Sede
Rossi	Marte	Roma
Verdi	Giove	Milano
	Verdi	Saturno
	Verdi	Venere
	Neri	Giove
	Neri	Saturno
	Neri	Venere

**NON riottengo  
la relazione di  
partenza!!**

# La normalizzazione

## Problemi di perdita

<u>Impiegato</u>	<u>Progetto</u>	<u>Sede</u>
Rossi	Marte	Roma
Verdi	Giove	Milano
Verdi	Venere	Milano
Neri	Saturno	Milano
Neri	Venere	Milano

**Decomposizione  
senza perdita di  
informazioni**

<u>Impiegato</u>	<u>Sede</u>
Rossi	Roma
Verdi	Milano
Neri	Milano

<u>Impiegato</u>	<u>Progetto</u>
Rossi	Marte
Verdi	Giove
Verdi	Venere
Neri	Saturno
Neri	Venere

# La normalizzazione

## Problemi di perdita

---

- ▶ Supponiamo di voler inserire una nuova ennupla che specifica la partecipazione dell'impiegato Neri, che opera a Milano, al progetto Marte
- ▶ Una istanza legale nello schema decomposto genera sullo schema ricostruito una soluzione non ammissibile
- ▶ Ogni singola istanza è (“localmente”) legale, ma il DB (“globalmente”) non lo è
  - ▶ Infatti il progetto “Marte” risulta essere assegnato a due sedi, in violazione del vincolo *Progetto* → *Sede*
- ▶ Problemi di consistenza dei dati si hanno quando la decomposizione “separa” gli attributi di una FD. Per verificare che la FD sia rispettata si rende necessario far riferimento a entrambe le relazioni.

# La normalizzazione

## Problemi di perdita

---

- ▶ 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 la dipendenza *Progetto* → *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

# La normalizzazione

## Prima Forma Normale 1NF

---

Una relazione è in **1NF** se rispetta i requisiti del modello relazionale:

1. Tutte le righe hanno lo stesso n° di colonne
2. Gli attributi sono atomici, né multivalore, né composti (un dato può considerarsi indivisibile se le eventuali sottoparti non hanno significato particolare nel contesto di interesse)
3. I valori di una colonna appartengono allo stesso dominio
4. Ogni tupla differenzia dalle altre per almeno un campo (esiste una PK)
5. L'ordine delle colonne è irrilevante

# La normalizzazione

## Seconda Forma Normale 2NF

---

Una relazione è in **2NF** se è in 1NF e:

- ▶ nessun campo non-chiave dipende funzionalmente da un sottoinsieme degli attributi di una chiave primaria composta

Es. Per una classe esiste la relazione

VotiMaterie(data, materia, voto, insegnante)

Insegnante dipende solo dalla materia, per cui si creano  
2 relazioni

VotiMaterie(data, materia, voto)

Insegnanti\_Materie(materia, insegnante)

# La normalizzazione

## Seconda Forma Normale 2NF

---

**Procedimento per trasformare la tabella in 2NF**  
(si fa solo per tabelle con chiavi primaria composte):

1. si individuano gli attributi dipendenti da sottoinsiemi della chiave primaria composta
2. si crea una nuova tabella per ogni dipendenza individuata e si copiano le colonne determinante e dipendente
3. le colonne determinanti saranno le nuove PK
4. si cancellano dalla tabella di partenza le colonne dipendenti
5. le colonne determinanti nella tabella di partenza diventano chiavi esterne sulle nuove tabelle



# La normalizzazione

## Terza Forma Normale 3NF

---

Una relazione è in **3NF** se è in 2NF e:

- ▶ nessun campo non-chiave dipende funzionalmente da altri campi non-chiave, non ci deve essere dipendenza transitiva di un attributo non primo dalla chiave.
- ▶ TEOREMA ogni relazione può essere portata in 3NF
- ▶ Se c'è un solo attributo non primo automaticamente è in 3NF

Studente (matricola, cognome, nome, Via, CAP, Città)

La città dipende dal CAP, per cui si creano le relazioni

Studente (matricola, cognome, nome, Via, CAP)

CAPCittà (CAP, Città)

Una **relazione** è in **3NF** se per ogni FD non banale  $X \rightarrow Y$  è vera una delle seguenti condizioni:  
X è una superchiave della relazione  
Y è un attributo primo

# La normalizzazione

## Terza Forma Normale 3NF

---

### Procedimento per trasformare la tabella in 3NF:

1. si individuano gli attributi dipendenti da un attributo o combinazione di attributi non chiave
2. si crea una nuova tabella per ogni dipendenza individuata e si copiano le colonne determinante e dipendente (se uno o più determinanti si determinano reciprocamente  $A \rightarrow B$   $B \rightarrow A$  non si divide lo schema)
3. le colonne determinanti saranno le nuove PK
4. si cancellano dalla tabella di partenza le colonne dipendenti
5. le colonne determinanti nella tabella di partenza diventano chiavi esterne sulle nuove tabelle

# La normalizzazione

## Forma Normale di Boyce-Codd

---

Una relazione è in **Forma normale di Boyce-Codd BCNF** se è in 1NF e:

- ▶ ogni determinante è una chiave candidata o superchiave. Non è possibile garantire sempre il raggiungimento della BCNF senza perdite

Esempio

data la relazione ABCD(A,B,C,D) se esiste la dipendenza non banale

$AD \rightarrow B$  è in 2NF e in 3NF (B non è un campo non chiave), ma non è in BCNF perché AD non è né chiave, né superchiave.

Analogamente per i casi

$AB \rightarrow C$  e  $D \rightarrow A$

# La normalizzazione

## Forma Normale di Boyce-Codd

---

Lo schema

TEL(Prefisso,Numero,Località,Abbonato,Indirizzo)

ha i seguenti vincoli

- ▶ *Località,Numero* → *Prefisso,Abbonato, Indirizzo*
- ▶ *Prefisso,Numero* → *Località,Abbonato, Indirizzo* la scelgo come PK per principio di minimalità
- ▶ *Località* → *Prefisso*
- ▶ È in 2NF e in 3NF, in quanto *Prefisso* è primo, ma non è in BCNF

<u>Prefisso</u>	<u>Numero</u>	Località	Abbonato	Indirizzo
051	457856	Bologna	Rossi	Via Roma 8
059	452332	Modena	Verdi	Via Bari 16
051	987856	Bologna	Bianchi	Via Napoli 77
051	552346	Castenaso	Neri	Piazza Borsa 12
059	387654	Vignola	Mori	Via Piave 65

# La normalizzazione

## Forma Normale di Boyce-Codd

---

- ▶ Una soluzione consiste nel decomporre lo schema in
  - ▶ NUM\_TEL(Numero, Località, Abbonato, Indirizzo)
  - ▶ PREF\_TEL(Località, Prefisso)

<u>Numero</u>	<u>Località</u>	Abbonato	Indirizzo		<u>Località</u>	Prefisso
457856	Bologna	Rossi	Via Roma 8		Bologna	051
452332	Modena	Verdi	Via Bari 16		Modena	059
987856	Bologna	Bianchi	Via Napoli 77		Castenaso	051
552346	Castenaso	Neri	Piazza Borsa 12		Vignola	059
387654	Vignola	Mori	Via Piave 65			

# La normalizzazione

## Forma Normale di Boyce-Codd

---

- ▶ Se una relazione è in BCNF è anche in 2 e 3NF in quanto esclude che un determinante possa essere composto solo da una parte della chiave, come avviene per le violazioni alla 2NF, o che possa essere esterno alla chiave, come avviene per le violazioni alla 3NF. Non è vero il contrario.
- ▶ La 3NF garantisce di non perdere informazioni e dipendenze, non la BCNF, ovvero possono esserci relazioni che non possono essere normalizzate nella forma Boyce-Codd senza perdita di dipendenze funzionali (la BCNF non è senza perdita di dipendenze)

[Esempi](#)

# La normalizzazione

## Problema non superabile

---

- Ogni dirigente si trova in una sola sede. Un progetto può svilupparsi su più sedi, ma in ogni sede ha un solo dirigente. Pertanto le FD sono:

1. *Progetto, Sede* → *Dirigente*
2. *Dirigente* → *Sede*

Dirigente	<u>Progetto</u>	<u>Sede</u>
Rossi	Marte	Roma
Verdi	Giove	Milano
Verdi	Marte	Milano
Neri	Saturno	Milano
Neri	Venere	Milano

# La normalizzazione

## Problema non superabile

---

- ▶ La 2<sup>a</sup> FD rispetta la 3NF perchè *Sede* è un attributo chiave, ma non è in BCNF perché *Dirigente* non è chiave candidata
- ▶ Però la 1<sup>a</sup> FD coinvolge tutti gli attributi e quindi nessuna decomposizione può preservare tale dipendenza
- ▶ Quindi potrebbe non essere possibile decomporre in BCNF e preservare le FD
- ▶ Potrei pensare di cambiare  $T1(\underline{\text{Progetto}}, \underline{\text{Dirigente}})$  e  $T2(\underline{\text{Dirigente}}, \underline{\text{Sede}})$ . Ma così potrei aggiungere *Verdi-Saturno*, ma violerebbe il vincolo che ogni progetto in una particolare sede ha un solo dirigente (in Milano risulterebbero Verdi e Neri sullo stesso progetto)



# La normalizzazione

## In pratica

---

- ▶ Se la relazione non è normalizzata si decompone in terza forma normale
- ▶ Si verifica se lo schema ottenuto è anche in BCNF
- ▶ Se uno schema non è in BCNF si hanno 3 alternative:
  - ▶ Si lascia così com'è, gestendo le anomalie residue (se l'applicazione lo consente)
  - ▶ Si decompone in BCNF, predisponendo opportune query di verifica (per verificare le dipendenze originarie vengano violate)
  - ▶ Si cerca di rimodellare la situazione iniziale, al fine di permettere di ottenere schemi BCNF

# La normalizzazione

## Procedimento

---

- ▶ Aggiungere ipotesi
- ▶ 1FN scrivere le considerazioni che portano a scegliere o meno al suddivisione o la non suddivisione. Scegliere la PK nei passi successivi
- ▶ Elencare le DF partendo dai campi singoli, aumentando la complessità
- ▶ Individuare chiavi candidate (saranno in ordine crescente di dimensione e tenendo conto dell'eventuale ordine logico per fare gli indici) e poi eleggere la PK
- ▶ 2FN giustificare se lo è, altrimenti indicare la definizione e indicare le DF che non la soddisfano e creare le nuove tabelle
- ▶ 3FN come sopra
- ▶ BCNF come sopra, valgono solo più le DF che hanno tutti i campi determinanti in una tabella con almeno un campo dipendente nella stessa tabella
- ▶ Sottolineare le tabelle risultato