

# Forme normali & Normalizzazione

# Forme normali

- Le **forme normali** sono condizioni che garantiscono che certe anomalie non possano emergere in una relazione
- Il processo mediante cui una relazione viene scomposta in due o più relazioni una relazione che non soddisfa una forma normale si chiama **normalizzazione**
- Nel seguito analizzeremo le più importanti forme normali, facendo vedere come normalizzare le relazioni che non sono in forma normale

# Normalizzazione via scomposizione

- L'intuizione di base è che le anomalie delle relazioni non normalizzate possano essere risolte via scomposizione (**decomposition**) della relazione originale in relazioni «più piccole»
- Due relazioni **R1(Y)** e **R2(Z)** sono una **scomposizione** di una relazione **R(X)** se:
  - **Y** e **Z** sono sottoinsiemi di **X**
  - **Y U Z = X**
- Y e Z sono quindi due **proiezioni** di R tali che la loro unione contenga tutti gli attributi in X

# Proprietà desiderabili in una scomposizione

- Nella scomposizione di una relazione, è molto importante che la scomposizione soddisfi le due seguenti proprietà:
  - **Scomposizione senza perdita di informazione** (*lossless join*): nel ricomporre mediante join la relazione di partenza a partire dalla sua scomposizione, non devono essere incluse **tuple spurie** (ovvero non appartenenti alla relazione iniziale)
  - **Conservazione delle DF**: la scomposizione deve conservare le DF della relazione originale, onde preservare i vincoli di integrità
- La prima proprietà DEVE essere SEMPRE garantita, la seconda in certi casi può essere «sacrificata»

# Perdita di informazione - Esempio

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

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

scomposizione

Impiegato → Sede

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

Progetto → Sede

<u>Progetto</u>	Sede
Marte	Roma
Giove	Milano
Venere	Milano
Saturno	Milano

Natural JOIN su Sede

**Tuple SPURIE!**

# Perdita di informazione - Esempio

Intuitivamente, perché si generano tuple spurie nell'esempio appena visto?

- «Sede» è l'unico attributo comune tra le due relazioni, quindi il JOIN può essere fatto solo su «Sede»
- Tuttavia, «Sede» non è (e non può essere) chiave di alcuna delle due relazioni, dato che alla stessa sede possono corrispondere più impiegati e più progetti
- Questo fa sì che ci siano triple in cui c'è il match di sede, ma viene persa l'associazione tra impiegato, progetto e sede (questa è la «perdita di informazione» vera e propria)
- In altre parole, ogni impiegato sarà associato a tutti progetti della sua sede, anche quando l'associazione con il progetto non è corretta

Come si può essere sicuri che ciò non possa accadere?

# Scomposizione senza perdita di informazione– Condizioni formali per garantirla

- E' possibile individuare una condizione che garantisce la scomposizione senza perdita di una relazione:
  - Sia  $R$  una relazione su un insieme di attributi  $X$  e siano  $X_1$  e  $X_2$  due sottoinsiemi di  $X$  tali che  $X = X_1 \cup X_2$
  - Inoltre, sia  $X_0 = X_1 \cap X_2$ ;
  - $R$  si scompone senza perdita di informazione su  $X_1$  e  $X_2$  se soddisfa la dipendenza funzionale  $X_0 \rightarrow (X_1 - X_0)$  oppure  $X_0 \rightarrow (X_2 - X_0)$
- Quindi la scomposizione senza perdita è **garantita** se gli attributi comuni alle due relazioni ottenute dalla scomposizione **contengono una chiave** (ovvero sono **superchiave**) di almeno una delle due relazioni scomposte.

# Scomposizione senza perdita – Torniamo all'esempio

- Sia **R** la relazione iniziale (**Impiegato, Progetto, Sede**)
- Prendiamo i due sottoinsiemi di attributi  $R_1 = (\text{Impiegato, Sede})$  e  $R_2 = (\text{Progetto, Sede})$
- $R = R_1 \cup R_2$
- $\text{Sede} = R_1 \cap R_2$ ;
- **R** si scompone senza perdita di informazione su  $R_1$  e  $R_2$  se soddisfa la dipendenza funzionale  $\text{Sede} \rightarrow \text{Impiegato}$  oppure  $\text{Sede} \rightarrow \text{Progetto}$

Siccome nessuna delle due condizioni vale (**Sede non contiene una chiave** né per  $R_1$  né per  $R_2$ ), la scomposizione non è senza perdita di informazione.



# Conservazione delle DF

- Le dipendenze sono conservate se è verificata la seguente condizione:
  - Ognuna delle dipendenze della relazione originale deve essere ottenibile per proiezione da almeno una delle relazioni ottenute dalla scomposizione
- Come accennato, non sempre è possibile garantire questa condizione

# Conservazione delle DF – Esempio

- Immaginiamo di dover scomporre la relazione:

R(Impiegato, Progetto, Sede)

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

- Le dipendenze funzionali (DF) della relazione originale R(Impiegato, Progetto, Sede) sono:
  - Impiegato → Sede
  - Impiegato → Progetto
  - Progetto → Sede

} dalla chiave

# Conservazione delle DF – Esempio

- Proviamo a scomporre la relazione come segue (utilizzando le prime due DF):

Impiegato	Sede
Rossi	Roma
Verdi	Milano
Neri	Milano

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

- Questa scomposizione però non conserva la DF **Progetto → Sede**
- Quale problema può essere causato da questa situazione?

# Conservazione delle DF - Esempio

- Se ora aggiungiamo la tupla <Neri, Marte> a R2, non violiamo alcuna DF di R2

Impiegato	Sede
Rossi	Roma
Verdi	Milano
Neri	Milano

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

- In questo modo però inseriamo un'informazione che nella relazione di partenza sarebbe stata illecita (ovvero che Neri lavora sul progetto Marte con sedi diverse a Roma e a Milano), dato che viola la DF **Progetto → Sede** che nella scomposizione si è persa

# La Prima forma normale (1NF)

- La Prima Forma Normale prescrive che una relazione non debba avere:
  - tuple ripetute (ci deve essere una chiave)
  - attributi composti e/o multi-valore
- Molti DBMS non accettano neanche relazioni che non siano in 1NF!

# Esempio 1NF e normalizzazione

DEPARTMENT

Dname	<u>Dnumber</u>	Dmgr_ssn	Dlocations
Research	5	333445555	{Bellaire, Sugarland, Houston}
Administration	4	987654321	{Stafford}
Headquarters	1	888665555	{Houston}

- La relazione non è in 1NF perché **Dlocations** ha molteplici valori per lo stesso attributo
- Per portarla in 1NF basta modificare la relazione DEPARTMENT come segue:

DEPARTMENT

Dname	<u>Dnumber</u>	Dmgr_ssn	<u>Dlocation</u>
Research	5	333445555	Bellaire
Research	5	333445555	Sugarland
Research	5	333445555	Houston
Administration	4	987654321	Stafford
Headquarters	1	888665555	Houston

# Terminologia (ripasso)

- **Superchiave** di una relazione R:
  - È un insieme di attributi  $S_K$  di R tali che non esistono due tuple di  $r(R)$  in cui gli attributi in  $S_K$  hanno lo stesso valore (ovvero, se  $t_1$  e  $t_2$  sono tuple distinte di  $r(R)$ ,  $t_1[S_K] \neq t_2[S_K]$ )
- **Chiave** di una relazione R:
  - Una chiave è una superchiave **minimale**, ovvero una superchiave  $K$  tale che la rimozione di qualsiasi attributo da  $S_K$  produrrebbe un insieme di attributi che non è più una superchiave di R
  - Una Chiave è sempre una Superchiave, ma non viceversa
- **Chiavi candidate**: tutte le superchiavi minimali di R
- **Chiave primaria**: la chiave candidata scelta dal DBA per identificare le tuple di R

# Attributi primi e non primi

- Definiamo inoltre:

- un **attributo primo** (*prime attribute*) è un attributo che appartiene ad almeno una chiave candidata
- un **attributo non primo** (*non-prime attribute*) è un attributo che non è primo, ovvero non appartiene ad alcuna chiave candidata



# Le forme normali basate sulle chiavi

- Le più importanti forme normali sono quelle legate alle chiavi della relazione:
  - **seconda forma normale (2NF)**: è in prima forma normale **e** ogni attributo *non primo* dipende dall'intera chiave primaria
  - **terza forma normale (3NF)**: è in seconda forma normale **e** non ci sono attributi *non primi* che dipendono transitivamente dalla chiave
  - **forma normale di Boyce-Codd (BCNF)**: ogni insieme di attributi di **R** dal quale dipendono altri attributi è una superchiave di **R**

## Seconda forma normale (2NF)

<u>Impiegato</u>	<u>Stipendio</u>	<u>Progetto</u>	<u>Bilancio</u>	<u>Funzione</u>
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

- La Seconda Forma Normale (2NF) richiede **ogni attributo non primo dipenda funzionalmente dall'intera chiave primaria** della relazione
- Nella tabella esempio, l'attributo **Stipendio** dipende da **Impiegato** (ma non da **Progetto**), mentre **Bilancio** dipende da **Progetto** (ma non da **Impiegato**)
- Quindi la tabella non è in 2NF

# Normalizzazione via scomposizione

<u>Impiegato</u>	Stipendio	<u>Progetto</u>	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

## Progetto → Bilancio

<u>Progetto</u>	Bilancio
Marte	2
Giove	15
Venere	15

## Impiegato → Stipendio

<u>Impiegato</u>	Stipendio
Rossi	20
Verdi	35
Neri	55
Mori	48
Bianchi	48

Gli attributi non primi dipendono tutti dall'intera chiave delle relazioni dove compaiono

# Seconda forma normale (2NF) - Osservazioni

- Può una relazione in 1NF con chiave primaria non composta violare la 2NF?
  - Ovviamente no, perché non possono esserci dipendenze di attributi non-primi solo da parte della chiave ☺
- Quali problemi genera la violazione della 2NF?
  - Principalmente, ridondanza dei dati (nell'esempio lo stipendio di un impiegato è ripetuto ogni volta che un impiegato è associato a un progetto), con conseguenti possibili anomalie di inserimento, cancellazione o aggiornamento

# Perché la 2NF non basta?

<u>CodImpiegato</u>	Nome	Reparto	TelReparto
123	Mario	IT	98765432
...	...	...	...
...	...	...	...

Chiave primaria: CodImpiegato

Dipendenze Funzionali: CodImpiegato → Reparto  
Reparto → TelReparto

Osservazione:

- In realtà, il numero di telefono del reparto non dipende tanto dal CodImpiegato, ma dal fatto che quell'impiegato lavora in quel reparto e che al reparto è associato un certo numero
- Quindi la dipendenza di **TelReparto** dalla chiave **CodImpiegato** è ottenuta per transitività attraverso l'attributo **Reparto** (che invece dipende direttamente da **CodImpiegato**)

■ ■ ■

<u>CodImpiegato</u>	Nome	Reparto	TelReparto
123	Mario	IT	98765432
...	...	...	...
...	...	...	...

Chiave primaria: CodImpiegato

Dipendenze Funzionali: CodImpiegato → Reparto

Reparto → TelReparto

Questo è all'origine di possibili anomalie che già conosciamo:

- il telefono del reparto è ripetuto per ogni impiegato di quel reparto (ridondanza)
- se il telefono del reparto cambia, occorre modificare molte righe (e con errori di aggiornamento, si avrebbero telefoni differenti per lo stesso reparto!)
- se un Reparto non ha impiegati, non si può conoscere il suo telefono!

# Terza forma normale (3NF)

- Una relazione **R** è in 3NF se:
  - è in Seconda Forma Normale (2NF)
  - nessun attributo *non primo* di **R** dipende in modo transitivo da una chiave candidata
    - Non deve succedere che valga  $X \rightarrow Y$ ,  $Y \rightarrow Z$ ,  $X \rightarrow Z$ , dove **X** è una chiave candidata [a meno che **Y** non sia esso stesso una chiave candidata – vedi sotto!]
- Si può dimostrare che questi requisiti sono soddisfatti se per ogni DF (non banale)  $X \rightarrow A$  vale almeno una delle seguenti condizioni:
  - **X** è una superchiave di **R**, oppure
  - **A** è un attributo primo di **R**

# Terza forma normale

- Perché una relazione che soddisfa i due requisiti appena elencati è in 3NF?
  - Se per ogni DF non banale  $X \rightarrow A$  vale che **X** è una superchiave K di **R**, allora l'eventuale dipendenza transitiva dalla chiave è solo apparente (nel senso che A dipende effettivamente da X e non indirettamente eventuali chiavi candidate da cui potrebbe a sua volta dipendere X)
  - Se **A** è un attributo primo di **R**, allora i suoi valori contribuiscono in modo essenziale alla composizione di una qualche chiave candidata di **R**. Quindi non possono essere replicati a piacere senza «rompere» l'integrità della relazione



# Normalizzazione in 3NF

<u>CodImpiegato</u>	Nome	Reparto
123	Mario	IT
...	...	...
...	...	...

<u>Reparto</u>	TelReparto
IT	98765432
...	...
...	...

Le due nuove relazioni:

- Sono in 3NF
- Sono conservate le dipendente funzionali:
  - CodImpiegato → Reparto
  - Reparto → TelReparto
- NB: non c'è perdita di informazione (l'attributo Reparto – comune alle due relazioni – implica funzionalmente gli attributi della seconda relazione, ovvero TelReparto)

# 3NF e (ancora) anomalie

<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

## ■ DF:

- **Prefisso, Numero** → **Località, Abbonato, Indirizzo** [NB: (**Prefisso, Numero**) è una chiave]
- **Località** → **Prefisso** [NB: **Località** non è una chiave ma **Prefisso** è *primo*]
- La relazione quindi è 3NF
- Tuttavia, il prefisso è ripetuto per ogni abbonato! [ridondanza]

# 3NF e (ancora) anomalie

<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

- Per escludere questi casi di ridondanza, potrebbe quindi aver senso voler escludere situazioni in cui ci sono DF  $Y \rightarrow Z$  tali che:
  - $Y$  non è una superchiave e
  - $Z$  primo
- A questo fine è stata definita la cosiddetta Forma Normale di Boyce-Codd (BCNF)

# Forma normale di Boyce-Codd (BCNF)

- Una relazione **R** è in forma normale di Boyce - Codd (**BCNF**) se è in 2NF e per ogni dipendenza funzionale (non banale)  $X \rightarrow Y$  definita su di essa, **X** è una superchiave di **R**
  - Intuitivamente, la BCNF richiede che i concetti in una relazione siano omogenei (solo proprietà direttamente associate alla chiave)
- Procedura «semplice» di normalizzazione in BCNF:
  - Data una relazione  $R(U)$  e una dipendenza  $X \rightarrow A$  che viola la BCNF, scomporre  $R$  in:
    - $R_1(U-A)$
    - $R_2(XA)$
  - Ripetere la procedura se ci sono DF tali che  $R_1$  o  $R_2$  non sono ancora in BCNF

# Forma normale di Boyce-Codd (BCNF)

- È una forma normale più restrittiva della 3NF
- È certamente una proprietà desiderabile, ma non sempre la si può ottenere!
- Infatti, talvolta non esiste una scomposizione che preservi tutte le dipendenze funzionali della relazione originale e ci si deve «accontentare» della 3NF (che invece è sempre ottenibile), nonostante i suoi limiti

# Esempio di scomposizione 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 DF **Località** → **Prefisso** viola la BCNF (**Località** non è una chiave candidata)
- La seguente scomposizione è in BCNF:

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

<u>Prefisso</u>	<u>Località</u>
051	Bologna
059	Modena
051	Castenaso
059	Vignola

## Esempio (2)

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

- Purtroppo non è sempre possibile usare l'algoritmo di scomposizione per ottenere una scomposizione BCNF della relazione iniziale che preservi le dipendenze e sia *lossless*
- Assumiamo che valgano le seguenti DF:
  - Ogni dirigente opera in una singola sede:  
**Dirigente → Sede**
  - Per ogni sede, ogni progetto ha un solo dirigente:  
**Progetto, Sede → Dirigente**
- La relazione non è in BCNF (nella prima DF, Dirigente non è una superchiave)
- Nessuna scomposizione – anche se le singole relazioni ottenute sono in BCNF – potrà mai preservare le DF originali (banalmente, perché la seconda DF coinvolge tutti gli attributi) e quindi può essere opportuno limitarsi a una scomposizione in 3NF

# Esempio (3)

- Talvolta le relazione ottenute da una relazione non in BCNF sono in BCNF, ma non sono «fedeli» alla relazione originale
  - Si consideri la relazione  
INSEGNA(STUDENTE, INSEGNAMENTO, DOCENTE)  
con le seguenti dipendenze funzionali:  
 $\{STUDENTE, INSEGNAMENTO\} \rightarrow DOCENTE$   
 $DOCENTE \rightarrow INSEGNAMENTO$ 
    - Si consideri la seguente decomposizione di INSEGNA:  
INSEGNA1(DOCENTE, INSEGNAMENTO)  
INSEGNA2(DOCENTE, STUDENTE)
    - INSEGNA1 e INSEGNA2 sono entrambe in BCNF (la prima per motivi ovvi, la seconda perché di fatto non ha dipendenze funzionali)
- Tuttavia, questa decomposizione non preserva tutte le dipendenze funzionali (abbiamo perso  $\{STUDENTE, INSEGNAMENTO\} \rightarrow DOCENTE$ ) e quindi in generale non è opportuno usarla



# Considerazioni su BCNF

- La BCNF è la forma teoricamente migliore che può assumere una relazione:
  - Ogni attributo descrive un'entità «identificata da una chiave, dall'intera chiave, da nient'altro che dalla chiave» [W. Kent, *Data and Reality*, 1978]
  - Ogni tupla registra un «pezzo» di informazione su un'entità (o su una relazione, nel senso di ER) che non è derivabile dai valori di altre tuple usando solo le dipendenze funzionali
  - Questo elimina ridondanze o confusione tra attributi di diverse entità
  - Purtroppo – come abbiamo visto – non sempre è possibile scomporre una relazione in BCNF, ed è per questo che si ricorre spesso alla 3NF

# Alcuni esercizi dal libro di testo

3. Give a set of FDs for the relation schema  $R(A,B,C,D)$  with primary key  $AB$  under which  $R$  is in 1NF but not in 2NF.
4. Give a set of FDs for the relation schema  $R(A,B,C,D)$  with primary key  $AB$  under which  $R$  is in 2NF but not in 3NF.
5. Consider the relation schema  $R(A,B,C)$ , which has the FD  $B \rightarrow C$ . If  $A$  is a candidate key for  $R$ , is it possible for  $R$  to be in BCNF? If so, under what conditions? If not, explain why not.
6. Suppose we have a relation schema  $R(A,B,C)$  representing a relationship between two entity sets with keys  $A$  and  $B$ , respectively, and suppose that  $R$  has (among others) the FDs  $A \rightarrow B$  and  $B \rightarrow A$ . Explain what such a pair of dependencies means (i.e., what they imply about the relationship that the relation models).

# Alcuni esercizi dal libro di testo

**Exercise 19.2** Consider a relation  $R$  with five attributes  $ABCDE$ . You are given the following dependencies:  $A \rightarrow B$ ,  $BC \rightarrow E$ , and  $ED \rightarrow A$ .

1. List all keys for  $R$ .
2. Is  $R$  in 3NF?
3. Is  $R$  in BCNF?

**Exercise 19.4** Assume that you are given a relation with attributes  $ABCD$ .

1. Assume that no record has NULL values. Write an SQL query that checks whether the functional dependency  $A \rightarrow B$  holds.
2. Assume again that no record has NULL values. Write an SQL assertion that enforces the functional dependency  $A \rightarrow B$ .
3. Let us now assume that records could have NULL values. Repeat the previous two questions under this assumption.

# Alcuni esercizi dal libro di testo

**Exercise 19.5** Consider the following collection of relations and dependencies. Assume that each relation is obtained through decomposition from a relation with attributes *ABCDEFGHI* and that all the known dependencies over relation *ABCDEFGHI* are listed for each question. (The questions are independent of each other, obviously, since the given dependencies over *ABCDEFGHI* are different.) For each (sub)relation: (a) State the strongest normal form that the relation is in. (b) If it is not in BCNF, decompose it into a collection of BCNF relations.

1.  $R1(A,C,B,D,E)$ ,  $A \rightarrow B$ ,  $C \rightarrow D$
2.  $R2(A,B,F)$ ,  $AC \rightarrow E$ ,  $B \rightarrow F$
3.  $R3(A,D)$ ,  $D \rightarrow G$ ,  $G \rightarrow H$
4.  $R4(D,C,H,G)$ ,  $A \rightarrow I$ ,  $I \rightarrow A$
5.  $R5(A,I,C,E)$