

# Normalizzazione

---

PROF. DIOMAIUTA CRESCENZO

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA «*LUIGI VANVITELLI*»

# Forma normale (Normalizzazione sicuramente via forza)

Una **forma normale** è una proprietà di una base di dati relazionale che ne garantisce la “qualità”, cioè l'assenza di determinati difetti

Quando una relazione non è normalizzata ???

- Quando presenta ridondanze
- Quando si presta a comportamenti poco desiderabili durante gli aggiornamenti

Le forme normali sono di solito definite sul modello relazionale, ma hanno senso in altri contesti, ad esempio il modello E-R

# Normalizzazione

---

- L'attività che permette di trasformare schemi non normalizzati in schemi che soddisfano una forma normale è detta normalizzazione
- La normalizzazione va utilizzata come tecnica di verifica dei risultati della progettazione di una base di dati
- Non costituisce una metodologia di progettazione

# Una relazione con anomalie

- Lo stipendio per ciascun impiegato è unico ed è funzione del solo impiegato, indipendentemente dai progetti cui partecipa.
- Il bilancio per ciascun progetto è unico ed è funzione del solo progetto, indipendentemente dagli impiegati che vi partecipano.
- Ogni impiegato, in ciascun progetto cui partecipa, svolge una sola funzione, eventualmente diversa da progetto a progetto.

Stipendio è funzione solo di chi c'è nell'impiego

Impiegato	Stipendio	Progetto	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

# Una relazione con anomalie

- Lo stipendio di ciascun impiegato è ripetuto in tutte le tuple relative: **ridondanza**
- Se lo stipendio di un impiegato varia, è necessario modificare il valore in diverse tuple: **anomalia di aggiornamento** ⇒ es. Cambio stipendio di New
- Se un impiegato interrompe la partecipazione a tutti i progetti, senza lasciare l'azienda, dobbiamo cancellarlo: **anomalia di cancellazione**
- Un nuovo impiegato senza progetto non può essere inserito: **anomalia di inserimento**

Stipendio di New è sempre uguale.  
↑ Non aggiunge nuova informazione

Impiegato	Stipendio	Progetto	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

# Analisi della relazione

---

## Perché questi effetti indesiderati???

- Abbiamo usato un'unica relazione per rappresentare tutte queste informazioni eterogenee:
  - gli impiegati con i relativi stipendi
  - i progetti con i relativi bilanci
  - le partecipazioni degli impiegati ai progetti con le relative funzioni
- La fusione di concetti disomogenei in una unica relazione comporta:
  - Ridondanza
  - Anomalie di aggiornamento
  - Anomalie di cancellazione
  - Anomalie di inserimento

# Definizioni preliminari

---

- Sia R una relazione con chiave primaria X
- Ogni attributo A appartenente allo schema di R si dice
  - Primo se fa parte di X;
  - Non-primo se non appartiene a X.
- Persona(Nome,Cognome, dataNascita, via, CAP, comune, provincia)
  - nome, cognome, e dataNascita sono **primi**
  - via, CAP, comune e provincia sono **non-primi**

# Dipendenza funzionale

---

□ Per studiare in maniera sistematica questi aspetti, è necessario introdurre un vincolo di integrità: **la dipendenza funzionale**. Essa descrive legami di tipo funzionale tra gli attributi di una relazione

## PROPRIETA'

- Ogni impiegato ha un solo stipendio (anche se partecipa a più progetti)
- Ogni progetto ha un (solo) bilancio
- Ogni impiegato in ciascun progetto ha una sola funzione (anche se può avere funzioni diverse in progetti diversi)

# Una relazione con anomalie

Lo stipendio di ciascun impiegato è unico (anche se partecipa a più progetti).

Il valore dell'attributo **Impiegato** determina il valore dell'attributo **Stipendio**.

Impiegato	Stipendio	Progetto	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

Il bilancio di ciascun progetto è unico e dipende dal solo progetto, indipendentemente dagli impiegati.

Il valore dell'attributo **Progetto** determina il valore dell'attributo **Bilancio**.

# Dipendenza funzionale

---

Il concetto di dipendenza funzionale può essere formalizzato come segue:

- Data una relazione  $r$  su uno schema  $R(X)$
- e dati due sottoinsiemi (**non vuoti**) di attributi  $Y$  e  $Z$  di  $X$
- Diciamo che in  $r$  vale la dipendenza funzionale (**FD**)  $Y \rightarrow Z$  ( $Y$  determina funzionalmente  $Z$ ) se

per ogni coppia di ennuple  $t_1$  e  $t_2$  di  $r$  con gli stessi valori su  $Y$ ,  $t_1$  e  $t_2$  hanno gli stessi valori anche su  $Z$

# Notazione

---

$Y \rightarrow Z$

Nella relazione vista prima si hanno diverse FD tra cui:

**Impiegato → Stipendio** è funzione solo del **impiegato**

**Progetto → Bilancio**

**Impiegato, Progetto → Funzione** dipende funzionalmente da **imp e progetto**

# Dipendenza funzionale

Impiegato	Stipendio	Progetto	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

Ogni volta che nella relazione compare un certo Progetto, il valore del suo Bilancio rimane sempre lo stesso.

Impiegato → Stipendio

Progetto → Bilancio  
Impiegato, Progetto → Funzione

Si noti che le anomalie precedentemente identificate sono legate proprio alla presenza di DF.

Ogni volta che nella relazione compare un certo Impiegato, il valore del suo Stipendio rimane sempre lo stesso.

# Dipendenze funzionali banali

## **Impiegato, Progetto → Progetto**

Si tratta però di una FD “banale” (sempre soddisfatta)

$Y \rightarrow A$  è banale se  $A$  appartiene a  $Y$

$Y \rightarrow Z$  è non banale se nessun attributo in  $Z$  appartiene a  $Y$

in quanto due tuple con gli stessi valori sulla coppia di attributi **Impiegato** e **Progetto** hanno ovviamente lo stesso valore sull'attributo **Progetto**, che è uno dei due

## **gli impiegati hanno un unico stipendio**

$\text{Impiegato} \rightarrow \text{Stipendio}$

## **i progetti hanno un unico bilancio**

$\text{Progetto} \rightarrow \text{Bilancio}$

Faremo riferimento sempre a FD non banali

# Dipendenze funzionali

- $X \rightarrow Y$  non implica (ovviamente)  $Y \rightarrow X$  *No proprietà commutativa (ma può capitare che accada)*
- DF banale
  - una DF banalmente soddisfatta
    - Impiegato Progetto  $\rightarrow$  Progetto
    - $Y \subseteq X \Rightarrow$  banalmente  $X \rightarrow Y$
- DF piena:
  - $X \rightarrow Y$  è una dipendenza funzionale piena se: Rimuovendo un qualsiasi attributo da X essa non vale più.
- DF espressa in modo minimale:
  - $X \rightarrow A, B \Rightarrow \{X \rightarrow A, X \rightarrow B\}$
- Le DF che ci interessano sono quelle non banali, piene ed espresse in maniera minimale.
  - se e solo se!*

# Vincoli e dipendenze funzionali

---

- Una dipendenza funzionale è una caratteristica dello schema e non della particolare istanza dello schema.
- Una dipendenza funzionale è dettata dalla semantica degli attributi di una relazione e non può essere inferita da una particolare istanza dello schema
- Poiché la DF è un vincolo, una relazione è corretta quando soddisfa la DF

# Dipendenze funzionali e chiavi

---

- Se prendiamo una chiave  $K$  di una relazione  $R$  si può facilmente verificare che esiste una FD tra  $K$  e ogni altro attributo dello schema di  $R$ .
- Definizione di vincolo di chiave: non possono esistere due tuple con gli stessi valori su  $K$ . Quindi una DF che ha  $K$  al primo membro sarà sempre soddisfatta.  $\Rightarrow$  Garantisce di non avere valori doppioni
- Esisterà sempre una DF tra una chiave di una relazione e tutti gli attributi dello schema della relazione (esclusi quelli della chiave stessa).

# Dipendenze funzionali e chiavi

Impiegato	Stipendio	Progetto	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

Dato che gli attributi Impiegato e Progetto formano una chiave:

Impiegato, Progetto → Funzione

Analogamente, le seguenti DF non banali saranno automaticamente verificate:

Impiegato, Progetto → Stipendio; Impiegato, Progetto → Bilancio; Impiegato, Progetto → Stipendio, Funzione;

Impiegato, Progetto → Stipendio, Bilancio; Impiegato, Progetto → Bilancio, Funzione;

Impiegato, Progetto → Stipendio, Bilancio, Funzione;

# Anomalie e FD

- La terza FD **corrisponde ad una chiave e non causa anomalie**
    - **Impiegato, Progetto → Funzione** Perché al primo membro c'è una chiave primaria
  - Le prime due FD **non corrispondono a chiavi e causano anomalie**
    - **Impiegato → Stipendio**
    - **Progetto → Bilancio**
- X
- La differenza risiede nel fatto che Impiegato e Progetto formano **una superchiave** (nello specifico, la coppia di attributi è l'unica chiave della relazione), che – per definizione – non può contenere due tuple uguali. Perciò, la terza DF non genera mai ridondanze.
  - Le anomalie sono perciò causate dalle DF  $Y \rightarrow Z$  tali che Y non contiene una chiave.

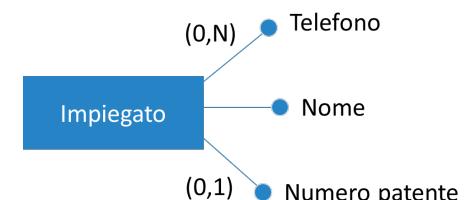
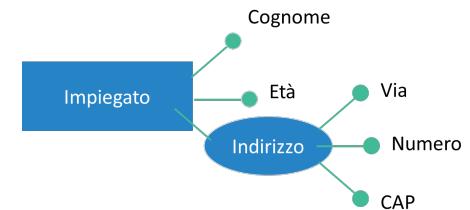
*Fisicamente sempre garantita. Dove comprendere solo valori atomici*

# Prima Forma Normale (1NF)

- Il dominio degli attributi deve comprendere solo valori atomici. Vanno cioè evitati:
  - attributi multi-valore;
  - attributi composti.
- Nel modello relazionale una relazione  $R(X)$  dovrebbe essere per definizione in 1NF

Impiegato	Progetto	Stipendio	Indirizzo	Funzione
Silvietti	Marte Saturno	50	via:Roma; città:Napoli; nc: 20	Direttore
Marcuccio	Marte Giove	30	Via: Claudio; città: Milano; nc=10	Progettista

- Progetto**
  - è un attributo multivalore
- Indirizzo**
  - è un attributo strutturato



# Riduzione in prima forma normale

---

- Una tabella può non rappresentare una relazione aderente al modello consentendo di esprimere attributi multi valore o strutturati
- Per ridurre una tabella in prima forma normale occorre:
  - Sviluppare gli attributi multi valore
  - Estrarre gli attributi strutturati

Che in realtà è garantita dalla 1<sup>a</sup> Forma Normale

Problema: attributi non primi che dipendono non dall'intera chiave

## Seconda Forma Normale (2NF)

- Uno schema di relazione  $R(X)$  è in 2NF se:
  - È in 1NF
  - Ogni attributo non primo dipende in maniera «piena» dalla chiave di  $R(X)$
- Per verificare il soddisfacimento della 2NF
  - Esaminare le parti sinistre delle DF che contengano attributi primi parte di chiavi.
- Se ogni chiave di  $R(X)$  ha un solo attributo  $\Rightarrow$  allora  $R(X)$  è già in 2NF.

<u>Articolo</u>	<u>Magazzino</u>	<u>Quantità</u>	<u>Indirizzo</u>
scarpe	NA1	2500	v. Leopardi 17, Napoli
scarpe	RM1	4500	v. S. Maria Maggiore 3, Roma
pantaloni	NA1	3000	v. Leopardi 17, Napoli

- Articolo, Magazzino  $\rightarrow$  *Indirizzo*
  - Magazzino  $\rightarrow$  Indirizzo
    - Indirizzo dipende anche da parte della chiave Articolo, Magazzino
- Indirizzo* è funzione solo del magazzino, che è parte della chiave

# Decomposizione

- ❑ Uno schema di relazione  $R(X)$  non in 2NF si può normalizzare **decomponendo** le relazioni di partenza in relazioni che soddisfano la 2NF.
- ❑ La relazione viene **decomposta** tramite gli "attributi comuni" in modo che:
  - ❑ Le relazioni ottenute soddisfano la 2NF
  - ❑ La decomposizione non alteri il contenuto informativo della base di dati

<u>Articolo</u>	<u>Magazzino</u>	<u>Quantità</u>	<u>Indirizzo</u>
scarpe	NA1	2500	v. Leopardi 17, Napoli
scarpe	RM1	4500	v. S. Maria Maggiore 3, Napoli
pantaloni	NA1	3000	v. Leopardi 17,Napoli

<u>Articolo</u>	<u>Magazzino</u>	<u>Quantità</u>
scarpe	NA1	2500
scarpe	RM1	2500
pantaloni	NA1	3000

<u>Magazzino</u>	<u>Indirizzo</u>
NA1	v. Leopardi 17, Napoli
RM1	v. S. Maria Maggiore 3,Napoli

↳ Se va solo a fare un form solo andare alla tabella oli partenza

# Proprietà della decomposizione

---

- ❑ Una decomposizione deve essere **senza perdita**:
  - ❑ Cioè devo essere in grado di ricostruire la relazione originaria
- ❑ Una decomposizione deve **mantenere** le **dipendenze funzionali**:
  - ❑ Ciascuna delle dipendenze funzionali dello schema originario coinvolge attributi che compaiono tutti insieme in uno degli schemi composti.

↓  
Mantengo le dipendenze funzionali

# Decomposizione con perdita

Consideriamo la seguente relazione

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

Si hanno le due FD:

$\text{Impiegato} \rightarrow \text{Sede}$  (un impiegato lavora presso un'unica sede)

$\text{Progetto} \rightarrow \text{Sede}$  (ogni progetto è sviluppato presso un'unica sede)

Decomponiamo la relazione in base alle dipendenze funzionali, secondo la tecnica vista precedentemente, proiettando la relazione originaria sugli attributi Impiegato e Sede e Progetto e Sede, rispettivamente.

Ml join rigiona dicendo: tutti quelli che lavorano alla sede A devono lavorare a tutti i progetti della sede A.

# Decomposizione con perdita

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

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

Proviamo a ricostruire le informazioni sulla base della partecipazione degli impiegati ai progetti attraverso un join naturale, utilizzando l'attributo Sede, che è l'unico attributo comune alle due relazioni.

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

La relazione ricostruita contiene tuple aggiuntive (dette spurie) rispetto alla relazione originaria. Ad esempio, risulta che Verdi lavora a Milano sul progetto Saturno, ma – guardando la relazione originaria – Verdi effettivamente non lavora a tale progetto.

# Decomposizione senza perdita

---

- Sia  $R$  una relazione su  $X$
- Con  $X_1 \subset X$  e  $X_2 \subset X$
- $R_1(X_1)$  proiezione di  $R$  sull'attributo  $X_1$
- $R_2(X_2)$  proiezione di  $R$  sull'attributo  $X_2$
- Inoltre sia  $X_0 = X_1 \cap X_2 \neq \emptyset$
- Si vuole essere in grado di ricostruire la relazione  $R$  nel seguente modo:
  - $R = R_1 \text{ Equi Join on } (X_0) R_2$
- Una decomposizione con perdita può generare *tuple spurious*.
- Una condizione sufficiente affinché la relazione  $R$  si decomponga senza perdite su  $R_1(X_1)$  e  $R_2(X_2)$  deve verificarsi:
  - $X_0$  sia chiave per  $R_1$  o per  $R_2$ . (ma non necessariamente)

# Decomposizione senza perdita

## Impiegato → Sede

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

## Progetto → Sede

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

Decomponiamo la relazione originaria sfruttando solamente la DF Impiegato→Sede, tenendo conto che nella relazione decomposta Impiegato è chiave per la proiezione di sugli attributi Impiegato e Sede.

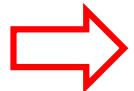
Proviamo a ricostruire le informazioni attraverso un join naturale, utilizzando l'attributo Impiegato, che è l'unico attributo comune alle due relazioni.

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

La relazione ricostruita è identica alla relazione originaria. E' importante notare che la tecnica appena analizzata è sufficiente ma non necessaria per la decomposizione senza perdita. Esistono infatti istanze di relazioni che non soddisfano nessuna delle due dipendenze, ma al tempo stesso si decompongono senza perdita.

# Conservazione delle FD

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



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

- Un impiegato deve operare su una sola sede
  - Impiegato → Sede**
- Un progetto deve insistere su una sola sede
  - Progetto → Sede**

Sfruttando la dipendenza  
**Impiegato → Sede**, otteniamo

Impiegato	Sede
Rossi	Roma
Verdi	Milano
Neri	Milano

Il join di queste due relazioni produce la tabella di partenza, per cui si può dire che la relazione si decomponе senza perdita su **Impiegato, Sede e Impiegato, Progetto**

# Conservazione delle dipendenze

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



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



Impiegato	Sede
Rossi	Roma
Verdi	Milano
Neri	Milano

**Impiegato** è chiave per la prima relazione, per cui la decomposizione senza perdita è garantita. Questa decomposizione però presenta un problema!

Problema:

# Conservazione delle dipendenze

- Supponiamo di voler inserire una nuova tupla che specifica la partecipazione dell'impiegato **Neri**, che opera a **Milano**, al progetto **Marte**.
- Sulla relazione originaria un tale aggiornamento verrebbe individuato come illecito, perché porterebbe ad una violazione della dipendenza **Progetto → Sede** (questa FD non è preservata).

<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
<b>Neri</b>	<b>Marte</b>

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

*Progetto Marte si svolge a Roma.*

# Conservazione delle dipendenze

- Sulle relazioni decomposte, invece, non è possibile rilevare alcuna violazione di dipendenze
- Sulla relazione decomposta **Impiegato,Progetto**, invece, non è possibile definire alcuna dipendenza funzionale e perciò rilevare alcuna violazione di dipendenze.
- Sulla relazione **Impiegato, Sede** la tupla con i valori **Neri** e **Milano** soddisfa la dipendenza funzionale **Impiegato → Sede**

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

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

Non è possibile effettuare alcuna verifica sulla dipendenza **Progetto → Sede**, perché i due attributi **Progetto** e **Sede** sono stati separati uno in una relazione e l'altro in un'altra.

# Conservazione delle dipendenze

- Una decomposizione conserva le dipendenze se ciascuna delle dipendenze funzionali dello schema originario coinvolge attributi che compaiono tutti insieme in uno degli schemi decomposti.
- In questo modo, è possibile garantire, sullo schema decomposto, il soddisfacimento degli stessi vincoli garantiti dallo schema originario.
- La DF Progetto → Sede non è conservata nella decomposizione dell'esempio precedente.
- La decomposizione sulla slide n.17 consente di conservare le dipendenze. Infatti, sia Progetto → Bilancio che Impiegato → Stipendio compaiono tutti insieme in uno degli schemi decomposti.

Impiegato → Stipendio

Impiegato	Stipendio
Rossi	20
Verdi	35
Neri	55
Mori	48
Bianchi	48

Progetto → Bilancio

Progetto	Bilancio
Marte	2
15	15
Venere	15

NOTA: Alcune forme normali non sono raggiungibili

# Forma Normale di Boyce e Codd (BCNF)

---

- Una relazione  $r$  è in forma normale di Boyce e Codd (BCNF) se,
  - È in 1FN
  - per ogni dipendenza funzionale (non banale)  $X \rightarrow Y$  definita su  $r$ ,  $X$  contiene una chiave  $K$  di  $r$ .
- La BCNF richiede che i concetti in una relazione siano omogenei (solo proprietà direttamente associate alla chiave).
- Se una relazione  $r$  non soddisfa la BCNF, la rimpiazziamo con altre relazioni che soddisfano la BCNF, **decomponendo**  $r$  sulla base delle dipendenze funzionali, al fine di separare i concetti.

# Procedura intuitiva di normalizzazione

---

- Non valida in generale, ma solo nei ‘casi semplici’
- Per ogni dipendenza  $X \rightarrow Y$  (con diverso primo membro) che viola la BCNF, definire una relazione su XY ed eliminare Y dalla relazione originaria.
- Purtroppo con questa tecnica non sempre le decomposizioni sono di buona qualità!

# Decomposizione in forma BCNF

Le tre relazioni, ottenute decomponendo la relazione originaria secondo le DF, sono in BCNF.

**Impiegato → Stipendio**

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

**Progetto → Bilancio**

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

<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

Si noti che abbiamo decomposto la relazione originaria in modo che a ciascuna dipendenza corrisponda una diversa relazione la cui chiave è proprio il primo membro della dipendenza stessa.

Ritrovazione della tabella di partenza è spesso possibile.

## Decomposizione in forma BCNF

**Impiegato, Progetto → Funzione**

<u>Impiegato</u>	<u>Progetto</u>	<u>Funzione</u>
Rossi	Marte	Tecnico
Verdi	Giove	progettista
Verdi	Venere	Progettista
Neri	Venere	Direttore
Neri	Giove	Consulente
Neri	Marte	Consulente
Mori	Marte	Direttore
Mori	Venere	Progettista
Bianchi	Venere	Progettista
Bianchi	Giove	Direttore

<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

Non tutte le decomposizioni sono desiderabili, individuando alcune proprietà essenziali che devono essere soddisfatte, riusciamo a tirar fuori una "buona" decomposizione

# Decomposizione in BCNF: Un altro esempio

<u>Numero</u>	<u>Località</u>	<u>Abbonato</u>	<u>Indirizzo</u>
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

- Decomponiamo lo schema in
  - Numero**(Numero,Località,Abbonato,Indirizzo)
  - Prefisso**(Località, Prefisso)
- La decomposizione è senza perdita poiché
  - (**Numero** *Equi Join* **Prefisso**) = **Telefono**
  - Località → Prefisso
  - Località, Numero → Abbonato, Indirizzo

# Qualità delle decomposizioni

---

- Una decomposizione dovrebbe sempre soddisfare:
  - la **decomposizione senza perdita**, che garantisce la ricostruzione delle informazioni originarie (cioè senza informazioni spurie) a partire da quelle rappresentate nelle relazioni decomposte;
  - la **conservazione delle dipendenze**, che garantisce il mantenimento dei vincoli di integrità originari, con le relazioni decomposte che hanno la stessa capacità della relazione originaria di rappresentare i vincoli di integrità e quindi di rilevare aggiornamenti illeciti.
- Quindi in definitiva, dato uno schema che viola una delle forme normali, l'attività di normalizzazione sulla base dati mira ad ottenere una decomposizione che sia senza perdita, che conservi le dipendenze e, infine, che contenga relazioni in forma normale.

# Limiti della BCNF

- A volte non è possibile raggiungere una buona decomposizione in BCNF (questo sempre raggiungibile)
- Ad esempio: Supponiamo di avere le seguenti DF:
  - Dirigente → Sede** (ogni dirigente opera presso una sede)
  - Progetto, Sede → Dirigente** (per ogni sede, un progetto ha un solo responsabile).

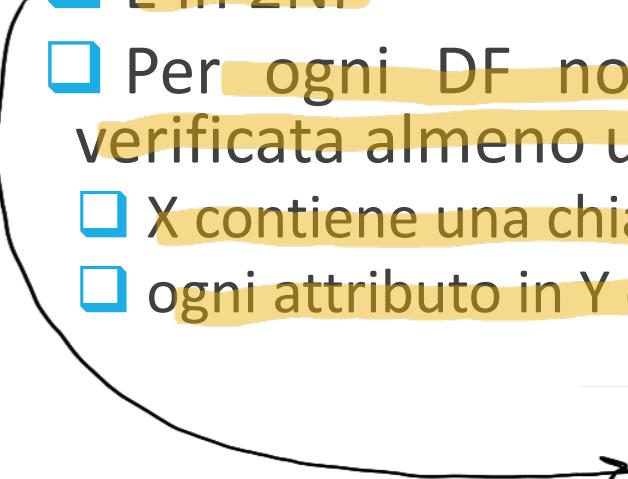
Dirigente	Progetto	Sede
Rossi	Marte	Roma
Verdi	Giove	Milano
Verdi	Marte	Milano
Neri	Saturno	Milano
Neri	Venere	Milano
Bianchi	Giove	Direttore

La relazione non è in BCNF, perché il primo membro della DF **Dirigente → Sede** non è superchiave.

Possiamo inoltre notare come non è possibile fare una buona decomposizione: infatti, la DF **Progetto, Sede → Dirigente** coinvolge tutti gli attributi e quindi nessuna decomposizione è in grado di conservarla.  
Quindi, talvolta la BCNF non è raggiungibile.

# Terza forma normale (3NF)

- Una relazione R è in Terza forma normale se e solo se,
  - È in 2NF
  - Per ogni DF non banale  $X \rightarrow Y$  definita su  $R(X)$  è verificata almeno una delle seguenti condizioni:
    - X contiene una chiave K di r
    - ogni attributo in Y è contenuto in almeno una chiave di r

- 
- Uno schema di relazione  $R(X)$  è in 2NF se:
    - È in 1NF
    - Ogni attributo non primo dipende in maniera «piena» dalla chiave di  $R(X)$

La BCNF ci garantisce assicura che tutte le anomalie  
La 3FN può tollerare presenza di ridondanze

# Limiti della BCNF

- Dirigente → Sede** (ogni dirigente opera presso una sede)
- Progetto, Sede → Dirigente** (per ogni sede, un progetto ha un solo responsabile).
- Questa relazione non soddisfa la BCNF, ma soddisfa la 3FN. Infatti, **Progetto Sede → Dirigente** ha come primo membro una chiave della relazione, mentre **Dirigente → Sede** anche se non contiene una chiave al primo membro, ha un unico attributo al secondo membro che fa parte della chiave **Progetto Sede**.

<b>Dirigente</b>	<b>Progetto</b>	<b>Sede</b>
Rossi	Marte	Roma
Verdi	Giove	Milano
Verdi	Marte	Milano
Neri	Saturno	Milano
Neri	Venere	Milano
Bianchi	Giove	Direttore

Si nota come la relazione presenta una forma di **ridondanza**. Ogni volta che un Dirigente compare in una tupla, viene ripetuta per esso la Sede in cui opera.

Questa ridondanza viene però tollerata dalla 3FN, perché non sarebbe possibile una decomposizione in BCNF che elimini tale ridondanza e al tempo stesso conservi tutte le dipendenze.

# Decomposizione in Terza forma normale (3NF)

---

- La 3FN è **meno restrittiva** della BCNF quindi non offre le stesse garanzie di qualità; però possiede il vantaggio di essere sempre ottenibile.
- Se una relazione ha una sola chiave, allora essa è in BCNF se e solo se è in 3NF
- Una relazione che non soddisfa la 3FN si decomponе in tante relazioni ottenute per proiezione sugli attributi che corrispondono alle dipendenze funzionali, basta far sì che almeno una delle relazioni decomposte contenga una chiave della relazione originaria.

↓  
elimina dipendenze transitivhe, cioè quelle che non sono legate alle chiavi delle relazioni al primo membro.

# Decomposizione in Terza forma normale (3NF)

**Dirigente → Stipendio**

Dirigente	Progetto	Stipendio
Rossi	Marte	30
Verdi	Giove	30
Verdi	Venere	30
Neri	Saturno	30
Neri	Venere	30

Una decomposizione in una relazione negli attributi **Dirigente Stipendio** e in un'altra sull'attributo **Progetto** violerebbe la proprietà di decomposizione senza perdita, in quanto nessuna delle due relazioni contiene una chiave.

La relazione non soddisfa la 3FN. Infatti, l'unica DF **Dirigente → Stipendio** ha come primo membro un attributo non chiave, e come secondo membro un attributo che non partecipa a nessuna chiave della relazione.

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

Bisogna invece definire la seconda relazione sugli attributi **Dirigente Progetto**, che formano una chiave della relazione originaria.

**Le due relazioni ottenute sono in 3FN!**

Dirigente	Stipendio
Rossi	30
Verdi	30
Neri	30

# Normalizzazione e scelta degli attributi

---

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

## □ Considerazione:

□ Esaminando le specifiche si poteva descrivere meglio l'applicazione di interesse, ad esempio introducendo un ulteriore attributo **Reparto**, che partiziona sulla base dei responsabili le singole sedi.

## □ Nuove Dipendenze funzionali

- **Dirigente -> Sede Reparto**
- **Sede Reparto -> Dirigente**
- **Progetto Sede -> Reparto**

**Progetto Sede -> Dirigente** (ricostruibile dalle due precedenti)

# Normalizzazione e scelta degli attributi

- Una buona decomposizione è la seguente:

Dirigente	Sede	Reparto
Rossi	Roma	1
Verdi	Milano	1
Neri	Milano	2

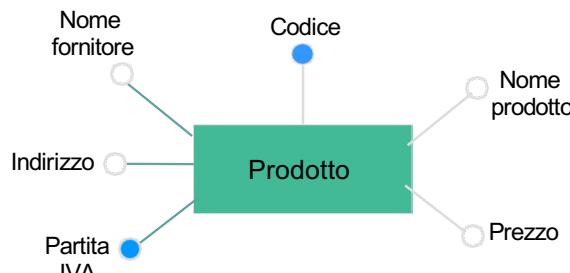
Progetto	Sede	Reparto
Marte	Roma	1
Giove	Milano	1
Marte	Milano	1
Saturno	Milano	2
Venere	Milano	2

- **La decomposizione è senza perdita:** gli attributi comuni Sede e Reparto formano una chiave per la prima relazione
- **Le dipendenze sono conservative:** per ogni dipendenza esiste una relazione decomposta che ne contiene tutti gli attributi
- **Entrambe le relazioni sono in forma normale di Boyce e Codd:** tutte le dipendenze hanno il primo membro costituito da una chiave

La non raggiungibilità della forma normale di Boyce e Codd può essere dovuta a un'analisi non sufficientemente accurata dell'applicazione

# Progettazione e normalizzazione

La teoria della normalizzazione può essere usata anche durante la progettazione concettuale per verificare la qualità dello schema concettuale stesso



ENTITA' NON NORMALIZZATA

Chiave: PartitaIVA, Codice

DF1: PartitaIVA → NomeFornitore, Indirizzo

DF2: Codice → NomeProdotto, Prezzo

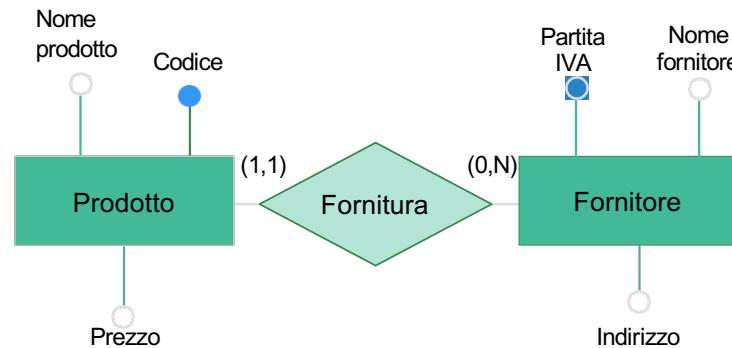
Rappresenta concetti elettronici tutti sulla stessa linea

# Progettazione e normalizzazione

L'entità viola la seconda forma normale a causa delle dipendenze parziali dalla chiave:

- PartitaVA → NomeFornitore Indirizzo
- Codice → NomeProdotto Prezzo

Possiamo decomporre sulla base di queste dipendenze



# Sintesi

---

## Prima forma normale (1NF)

Richiede che il dominio di un attributo comprenda solo valori atomici (semplici, indivisibili) e che il valore di qualsiasi attributo in una tupla sia un valore singolo del dominio.

- 1NF è già parte integrante della definizione formale di relazione nel modello relazionale.
  - tutte le righe della tabella contengono lo stesso numero di colonne
  - gli attributi rappresentano informazioni elementari
  - i valori che compaiono in una colonna appartengono allo stesso dominio
  - ogni riga è diversa da tutte le altre
  - l'ordine con il quale le righe compaiono nella tabella è irrilevante

# Sintesi

## Seconda forma normale (2NF)

- E' in prima forma normale
- tutti i suoi attributi non-chiave dipendono dall'intera chiave, cioè non possiede attributi che dipendono soltanto da una parte della chiave

La seconda forma normale elimina la dipendenza parziale degli attributi dalla chiave

La tabella T1 (A1,A2,A3,A4,A5) con  
 $(A1, A2) \rightarrow A3$   
 $A1 \rightarrow A4$   
 $A2 \rightarrow A5$

Non è in 2NF e può essere normalizzata in 2NF con le tabelle

T2(A1,A2,A3)  
T3(A1,A4)  
T4(A2,A5)

La seconda forma normale riguarda quindi le tabelle in cui la chiave primaria sia composta da più attributi e stabilisce che, in questo caso, tutte le colonne corrispondenti agli altri attributi dipendano dall'intera chiave primaria.

Le regole sono se nel fatto lo fatto su attributi non chiave.

Se avessi anche  $A_k \rightarrow A_s$  considerare ragionare un maniera differente. Fondamentalmente garantire decomposizione senza perdita e mantenere dipendenze funzionali. Non è sempre garantibile, quindi è possibile che dovrà lavorare con la 3FN, che magari non richiede di aggiungere qualche attributo e quindi garantire anche la 2FN.

# Sintesi

## Terza forma normale (3NF)

- E' in seconda forma normale
- tutti gli attributi non-chiave dipendono direttamente dalla chiave, cioè non possiede attributi che dipendono da altri attributi che non sono in chiave
- La terza forma normale elimina la dipendenza transitiva degli attributi dalla chiave
- La terza forma normale stabilisce che non esistano dipendenze tra le colonne di una tabella se non basate sulla chiave primaria.

La Tabella T1(A1,A2,A3,A4) con

A2 → A4 non è in 3FN.

Può essere normalizzata in 3FN con le tabelle:



T2(A1,A2,A3)

T3(A2,A4)

Applicabile se A2 può diventare una chiave primaria.

DT: dipendenza di A4 non primo che dipende da un altro attributo non primo.

A<sub>1</sub> → A<sub>2</sub>    A<sub>2</sub> → A<sub>4</sub>.    A<sub>4</sub> dipende in modo transitivo da A<sub>1</sub>.

NOTA: forma olí Boyce Codd é garantita per tabella fatta da 2 attributi.  
Anomalia sono obbligate a gestire. Ridondanza é più favorevole.

## Sintesi

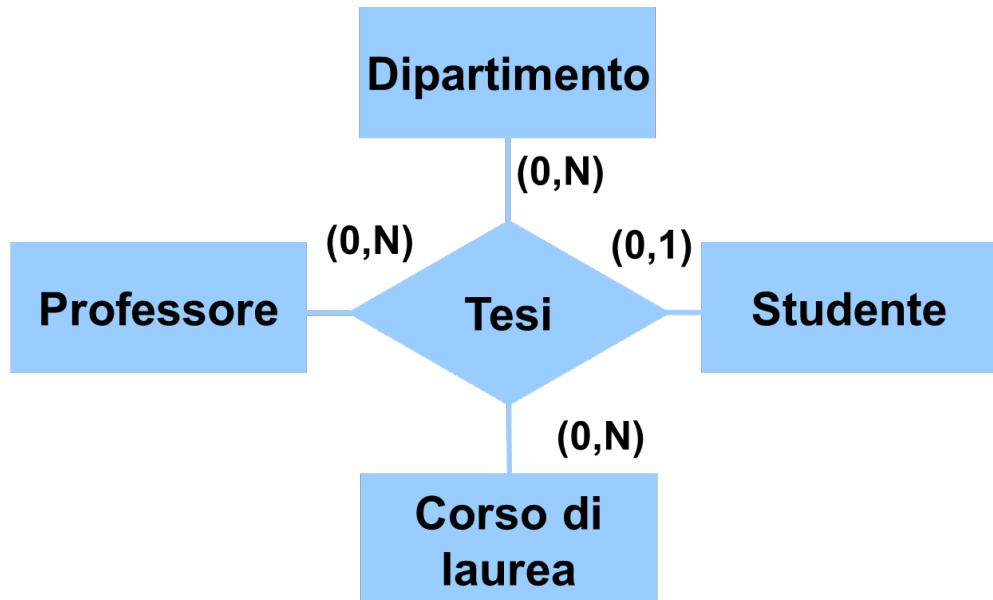
---

Con la **normalizzazione** si ha che

- la tabella iniziale viene **scomposta in più tabelle**
- complessivamente **forniscono le stesse informazioni di partenza**
- mantengono le dipendenze tra gli attributi**
- in **ciascuna di esse ogni attributo dipende direttamente dalla chiave**
- vengono evitati problemi di **ridondanza e di inconsistenza dei dati**
- non ci deve essere perdita complessiva delle informazioni**



# Esempi



Studente → Corso di laurea  
Studente → Professore  
Professore → Dipartimento

La relazione viola la terza forma normale a causa della dipendenza:  
 $\text{Professore} \rightarrow \text{Dipartimento}$   
Possiamo decomporre sulla base di questa dipendenza

## Esempi

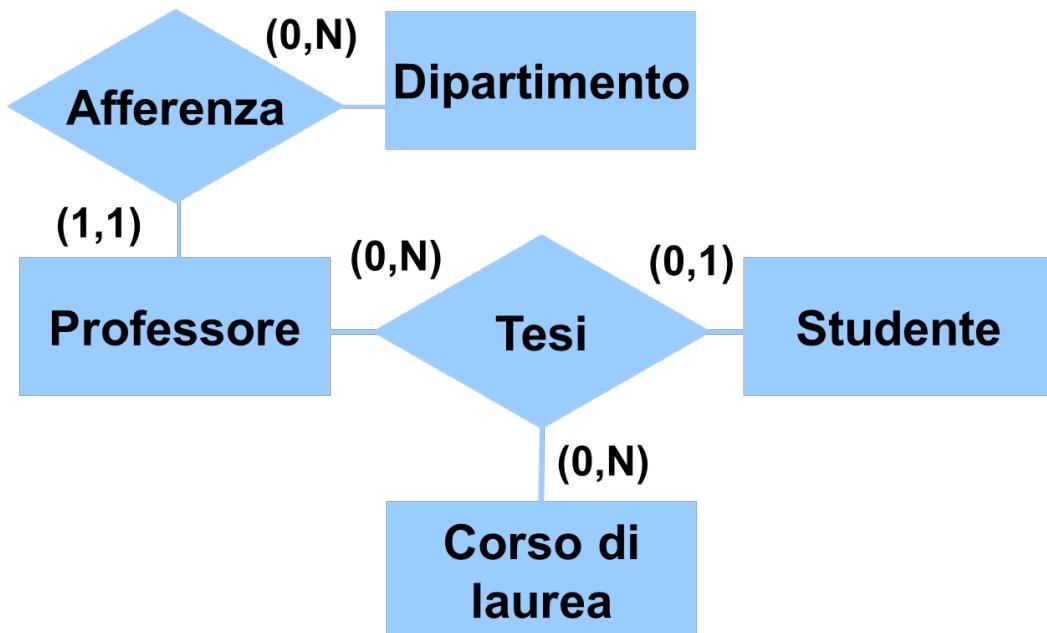
Interventi (Paziente, DataIntervento, OraIntervento, Chirurgo, Sala)

- Consideriamo una relazione che descrive l'allocazione delle sale operatorie di un ospedale
- Le sale operatorie sono prenotate, giorno per giorno, in orari previsti, per effettuare interventi su pazienti ad opera dei chirurghi dell'ospedale.
  - Nel corso di una giornata una sala operatoria è occupata sempre dal medesimo chirurgo che effettua più interventi, in ore diverse.
  - Noti i valori di Paziente e DataIntervento, sono noti anche: l' ora dell'intervento, chirurgo, e sala operatoria utilizzata.

Interventi

Paziente	DataIntervento	OraIntervento	Chirurgo	Sala
Bianchi	25/10/2005	8.00	De Bakey	Sala1
Rossi	25/10/2005	8.00	Romano	Sala2
Negri	26/10/2005	9.30	Veronesi	Sala1
Viola	25/10/2005	10.30	De Bakey	Sala1
Verdi	25/10/2005	11.30	Romano	Sala2

# Esempi



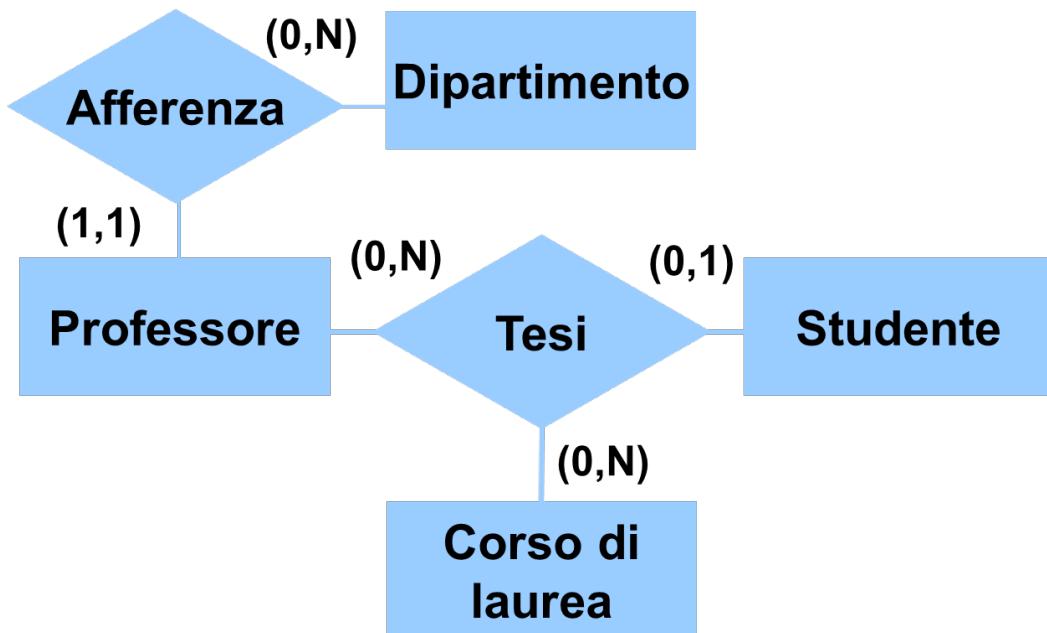
Studente → Corso di laurea  
Studente → Professore  
Professore → Dipartimento

La relazione viola la terza forma normale a causa della dipendenza:

Professore → Dipartimento

Possiamo decomporre sulla base di questa dipendenza

# Esempi



Studente → Corso di laurea  
Studente → Professore  
Professore → Dipartimento

La relazione **Tesi** è in BCNF sulla base delle dipendenze

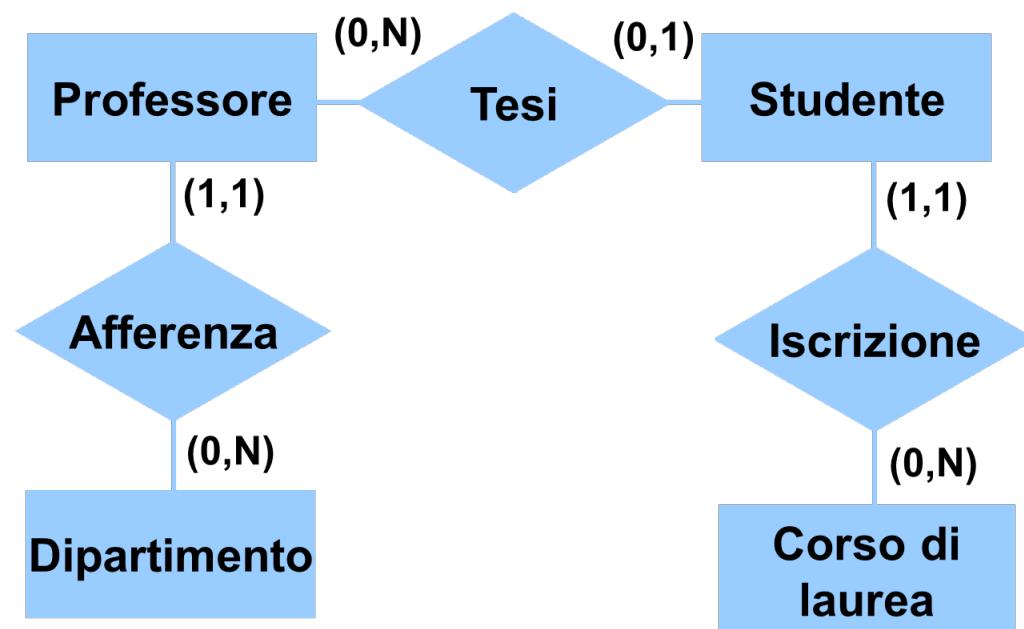
**Studente → CorsoDiLaurea**

**Studente → Professore**

le due proprietà sono indipendenti  
questo suggerisce una ulteriore decomposizione

# Esempi

---



# Esempi: 1FN

## Dipendenti

<b><u>Matricola</u></b>	<b>Nome</b>	<b>Indirizzo</b>	<b>Familiari a Carico</b>

## Dipendenti

<u>Matricola</u>	Nome	Indirizzo

Familiari

<b><u>CodiceFam</u></b>	<b>NomeFam</b>	<b>MatricolaDip</b>

# Esempi: 2FN

Merci

<u>Codice</u>	<u>Magazzino</u>	Quantità	LocalitàMagazzino

Merci

<u>Codice</u>	<u>Magazzino</u>	Quantità



Depositi

<u>Magazzino</u>	LocalitàMagazzino

# Esempi: 3FN

**Studenti**

<u>Nome</u>	<u>Scuola</u>	<u>Telefono Scuola</u>

**Studenti**

<u>Nome</u>	<u>Scuola</u>



**Istituti**

<u>Scuola</u>	<u>Telefono Scuola</u>

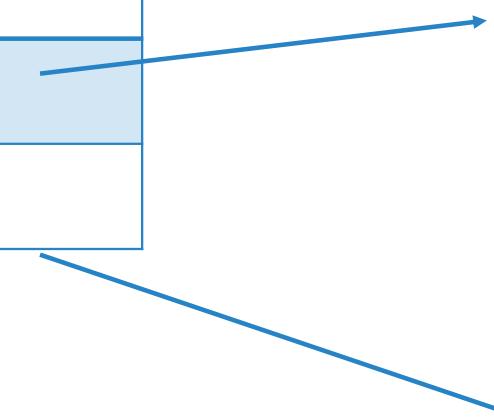
# Esempi: BCNF

Pagamenti

<u>Codice Esame</u>	Descrizione	Ticket

Esami

<u>Codice Esame</u>	Descrizione



Pagamenti

<u>Descrizione</u>	Ticket

# 3FN vs BCNF

**Studenti**

<u>Nome</u>	<u>Scuola</u>	<u>Telefono Scuola</u>
Rossi	Gallo	081/8111213
Verdi	Gallo	081/8986748
Bianchi	Siani	081/7656898

**Studenti**

<u>Nome</u>	<u>Scuola</u>
Rossi	Gallo
Verdi	Gallo
Bianchi	Siani

**Istituti**

<u>Scuola</u>	<u>Telefono Scuola</u>
Gallo	081/8111213
Gallo	081/8986748
Siani	081/7656898

# 3FN vs BCNF

Pagamenti

<u>Codice Esame</u>	<u>Descrizione</u>	<u>Ticket</u>
PS1	Prelievo Sangue	A001
UR1	Analisi Urine	A002
UR2	Urinocoltura	A002

In BCNF, ogni determinante deve essere una chiave candidata, cioè ogni determinante deve essere un insieme di attributi che identifica in modo univoco un record.

Esami

<u>Codice Esame</u>	<u>Descrizione</u>
PS1	Prelievo Sangue
UR1	Analisi Urine
UR2	Urinocoltura



Ticket

<u>Descrizione</u>	<u>Ticket</u>
Prelievo Sangue	A001
Analisi Urine	A002
Urinocoltura	A003



# 3FN vs BCNF

---

Uno schema di relazione  $R(X)$  è in forma normale di Boyce e Codd se:

- 2NF
- per ogni dipendenza funzionale  $Y \rightarrow Z$  (non banale) definita su  $R(X)$ 
  - $Y$  è chiave di  $R(X)$ , ovvero  $Y$  è una superchiave di  $R(X)$
  - **cioè ogni attributo  $Y$  dal quale dipendono altri attributi  $Z$  può svolgere la funzione di chiave.**

# 3FN vs BCNF

---

Uno schema di relazione  $R(X)$  è in 3NF se:

- 2NF
- per tutte le FD non banali  $Y \rightarrow Z$  definite su  $R(X)$ 
  - $Y$  è una chiave di  $R$  oppure  $Z$  è parte di una chiave di  $R$
- In conclusione se  $R$  è in 3NF qualche ridondanza è possibile

# Normalizzazione: Osservazioni

---

- La teoria della normalizzazione fornisce regole di progetto che cercano di prevenire anomalie di aggiornamento e inconsistenza dei dati
- Le relazioni in forme normali non sono le più utili ai fini del recupero delle informazioni, infatti, informazioni in tabelle non normalizzate si trovano in un'unica tupla, mentre in tabelle normalizzate sono necessarie una o più operazioni relazionali (join tipicamente)
- Nella pratica si cerca un trade-off tra normalizzazione ed efficienza nel recupero delle informazioni (in base alla frequenza delle operazioni sulle tabelle)

---

# ESERCIZI

# Esercizio 1

---

Considerare la relazione sotto e individuare le proprietà della corrispondente applicazione. Individuare inoltre eventuali ridondanze e anomalie nella relazione. Infine, individuare poi una decomposizione in forma normale di Boyce e Codd

Docente	Dipartimento	Facoltà	Preside	Corso
Verdi	Matematica	Ingegneria	Neri	Analisi
Verdi	Matematica	Ingegneria	Neri	Geometria
Rossi	Fisica	Ingegneria	Neri	Analisi
Rossi	Fisica	Scienze	Bruni	Analisi
Bruni	Fisica	Scienze	Bruni	Fisica

# Esercizio 1

---

Una chiave per questa relazione è **Dipartimento, Facoltà, Corso**; una dipendenza funzionale che non riguarda la chiave **Facoltà -> Preside**: questa dipendenza funzionale introduce una ridondanza nella relazione, perché per ogni corso nella stessa Facoltà, il Preside deve essere ripetuto.

La relazione ha un'anomalia di aggiornamento, perché se cambiamo il preside di una facoltà dobbiamo aggiornare tutte le tuple che contengono questa informazione, e non solamente una tupla. La relazione contiene anche un'anomalia di cancellazione, perché se cancelliamo il preside di una facoltà, perdiamo anche tutte le informazioni sui docenti di quel dipartimento.

# Esercizio 1

---

Anche gli attributi **Docente**, **Facoltà**, **Corso** sembrano formare una chiave in questa relazione, ma generalmente parlando questo non è corretto perché lo stesso docente può insegnare lo stesso corso in differenti dipartimenti di una Facoltà.

Docente	Dipartimento	Facoltà	Corso
Verdi	Matematica	Ingegneria	Analisi
Verdi	Matematica	Ingegneria	Geometria
Rossi	Fisica	Ingegneria	Analisi
Rossi	Fisica	Scienze	Analisi
Bruni	Fisica	Scienze	Fisica

Facoltà	Preside
Ingegneria	Neri
Scienze	Bruni

# Esercizio 2

---

Considerare lo schema della relazione sott: La chiave di questa relazione è costituita dagli attributi **Titolo** e **Copia**, e su di essa è definita la dipendenza **Titolo -> Autore, Genere**. Verificare se lo schema è o meno in **terza forma normale (3NF)** e, in caso negativo, dekomporlo opportunamente

<b>Titolo</b>	<b>Autore</b>	<b>Genere</b>	<b>Copia</b>	<b>Scaffale</b>
Decamerone	Boccaccio	Novelle	1	A75
Divina Commedia	Dante	Poema	1	A90
Divina Commedia	Dante	Poema	2	A90
I Malavoglia	Verga	Romanzo	1	A90
I Malavoglia	Verga	Romanzo	2	A75
I Promessi Sposi	Manzoni	Romanzo	1	B10
Adelchi	Manzoni	Tragedia	1	B20

# Esercizio 2

La relazione non è in terza forma normale perché il lato destro della dipendenza funzionale **Titolo -> Autore, Genere** non è parte della chiave. Una possibile decomposizione è:

R1

<b>Titolo</b>	<b>Copia</b>	<b>Scaffale</b>
Decamerone	1	A75
Divina Commedia	1	A90
Divina Commedia	2	A90
I Malavoglia	1	A90
I Malavoglia	2	A75
I Promessi Sposi	1	B10
Adelchi	1	B20

R2

<b>Titolo</b>	<b>Autore</b>	<b>Genere</b>
Decamerone	Boccaccio	Novelle
Divina Commedia	Dante	Poema
I Malavoglia	Verga	Romanzo
I Promessi Sposi	Manzoni	Romanzo
Adelchi	Manzoni	Tragedia

La relazione è in forma normale di Boyce-Codd, perché la chiave per R2 è **Titolo**, che è anche il lato sinistro della dipendenza funzionale.