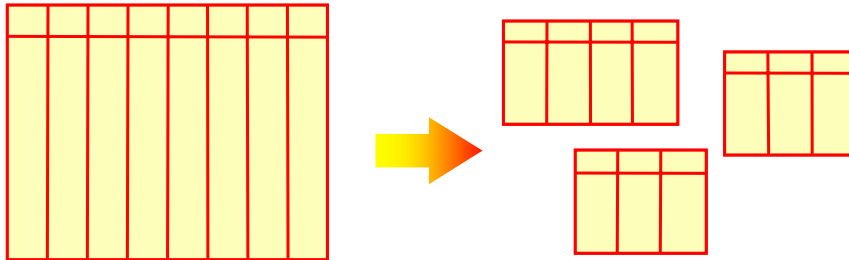


## Normalizzazione di schemi relazionali



*Dario Maio*

<http://www.csr.unibo.it/~maio/>  
[dmaio@deis.unibo.it](mailto:dmaio@deis.unibo.it)



## Forme normali

- Una forma normale è una proprietà di uno schema relazionale che ne garantisce la “**qualità**”, cioè **l’assenza di determinati difetti**
- Una relazione non normalizzata:
  - **presenta ridondanze**,
  - si presta a **comportamenti poco desiderabili durante gli aggiornamenti**
- Le forme normali sono di solito definite sul modello relazionale, ma hanno senso anche in altri contesti, ad esempio nel modello E/R
- 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 quindi una metodologia di progettazione



## *Una relazione con anomalie*

<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

In un'unica relazione sono rappresentati gli impiegati con i relativi stipendi, i progetti con i relativi bilanci e la partecipazione degli impiegati ai progetti

## *Analizziamo la relazione...*

- 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)
- Ma 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

## Ridondanze e anomalie

Impiegato	Stipendio	Progetto	Bilancio	Funzione
Rossi	20	Marte	2	tecnico
Verdi	35	Giove	15	progettista
Verdi	35	Venere	15	progettista
Neri	58	Venere	15	direttore
Neri	58	Giove	15	consulente
Neri	58	Marte	2	consulente
Mori	48	Marte	2	direttore
Mori	48	Venere	15	progettista
Bianchi	48	Venere	15	progettista
Bianchi	48	Giove	15	direttore

???



Gialli

- 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**
- Se un impiegato interrompe la partecipazione a tutti i progetti, dobbiamo cancellarlo: **anomalia di cancellazione**
- Un nuovo impiegato senza progetto non può essere inserito: **anomalia di inserimento**

Normalizzazione di schemi relazionali

5

## Ridondanze e anomalie

- **Ridondanza**: presenza di dati ripetuti in diverse tuple, senza aggiungere informazioni significative
- **Anomalia di aggiornamento**: necessità di estendere l'aggiornamento di un dato a tutte le tuple in cui esso compare
- **Anomalia di cancellazione**: l'eliminazione di una tupla motivata dal fatto che non è più valido l'insieme dei concetti in essa espressi, può comportare l'eliminazione di dati che conservano la loro validità
- **Anomalia di inserimento**: l'inserimento di informazioni relative a uno solo dei concetti di pertinenza di una relazione è impossibile se non esiste un intero insieme di concetti in grado di costituire una tupla completa.

Normalizzazione di schemi relazionali

6

## Ridondanze: una precisazione

- In una base dati l'informazione può essere duplicata in modo :

**NON RIDONDANTE** - la duplicazione dei dati è **necessaria**, l'eliminazione delle duplicazioni comporta **perdita di informazione**

STUDENTE

N-mat	Tutor
125233	Mario
127988	Carlo
150444	Carlo
190787	Mario

duplicazione di dati **non ridondante**

**RIDONDANTE** - la duplicazione dei dati **non è necessaria**, comporta spreco di memoria, è causa di possibili **anomalie e inconsistenze**

STUDENTE

N-mat	Tutor	Tel
125233	Mario	7575
127988	Carlo	5566
150444	Carlo	5566
190787	Mario	7575

duplicazione di dati **ridondante**

## Scomposizione di schemi

- Le ridondanze si possono eliminare mediante scomposizione degli schemi

STUDENTE

N-mat	Tutor	Tel
125233	Mario	7575
127988	Carlo	5566
150444	Carlo	5566
190787	Mario	7575

STUDENTE

N-mat	Tutor
125233	Mario
127988	Carlo
150444	Carlo
190787	Mario

TUTOR

Tutor	Tel
Mario	7575
Carlo	5566

## Dependenza funzionale

- Per formalizzare i problemi visti si introduce un nuovo tipo di vincolo, la **dependenza funzionale (FD)**

Consideriamo:

- Un'istanza  $r$  di uno schema  $R(X)$
- 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

$$\forall t1, t2 \in r : t1[Y] = t2[Y] \Rightarrow t1[Z] = t2[Z]$$

per ogni coppia di tuple  $t1$  e  $t2$  di  $r$  con gli stessi valori su  $Y$ ,  
 $t1$  e  $t2$  hanno gli stessi valori anche su  $Z$

## Esempi di FD

- Nella relazione 

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

 si hanno diverse FD, tra cui:

$Impiegato \rightarrow Stipendio$   
 $Progetto \rightarrow Bilancio$   
 $Impiegato, Progetto \rightarrow Funzione$

- Altre FD sono "meno interessanti" ("**banali**"), perché sempre soddisfatte, ad esempio:

$Impiegato, Progetto \rightarrow Progetto$

- Se  $Z \subseteq Y$  allora sicuramente  $Y \rightarrow Z$ .
- FD di questo tipo sono dette **FD banali**.
- $Y \rightarrow Z$  è non banale se nessun attributo in  $Z$  appartiene a  $Y$

## FD - Precisazioni

- Una dipendenza funzionale è una caratteristica dello schema, aspetto **intensionale**, e non della particolare istanza dello schema, aspetto estensionale
- Una dipendenza funzionale è **dettata dalla semantica** degli attributi di una relazione e non può essere inferita da una particolare istanza dello schema
- Una istanza di uno schema che rispetti una data dipendenza funzionale viene detta **istanza legale** dello schema rispetto alla data dipendenza funzionale
- Se  $X$  è una **chiave** in uno schema  $R$  allora ogni altro attributo di  $R$  dipende funzionalmente da  $X$
- Dire che  $X \rightarrow Y$  significa asserire che i valori della componente  $Y$  dipendono da (sono determinati da) i valori della componente  $X$
- Se  $X \rightarrow Y$  non necessariamente risulta anche  $Y \rightarrow X$

## FD e Superchiavi

- Il concetto di superchiave si esprime facendo uso di FD.

$K \subseteq T$  è superchiave di  $R(T) \Leftrightarrow K \rightarrow T$

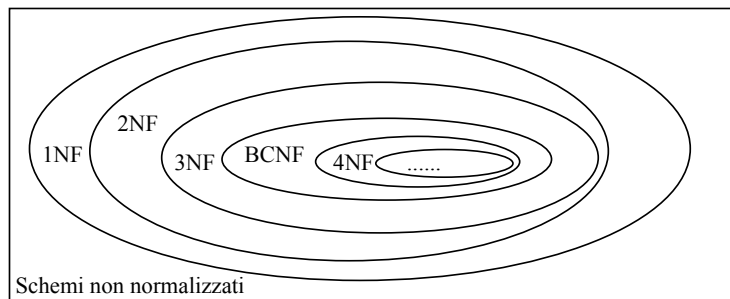
### Dimostrazione

- **(se)** Se  $K \rightarrow T$  allora per ogni istanza legale  $r$  si ha che  $\forall t_1, t_2 \in r : t_1[K] = t_2[K] \Rightarrow t_1[T] = t_2[T]$ , ovvero  $t_1 = t_2$ . Ciò equivale a dire che non possono esistere due tuple *distinte* con lo stesso valore di  $K$ .
- **(solo se)** Se  $K$  è superchiave di  $R(T)$ , dalla definizione di superchiave si ha che  $t_1[K] = t_2[K] \Rightarrow t_1 = t_2$ , e quindi  $t_1[T] = t_2[T]$ .

## **Anomalie e FD**

- Le anomalie viste si riconducono alla presenza delle FD:  
    Impiegato → Stipendio  
    Progetto → Bilancio
- Viceversa la FD  
    Impiegato, Progetto → Funzione  
non causa problemi
- Motivo:
  - La terza FD ha **sulla sinistra una chiave e non causa anomalie**
  - Le prime due FD **non hanno sulla sinistra una chiave e causano anomalie**
- La relazione contiene alcune informazioni legate alla chiave e altre ad attributi che non formano una chiave

## **Evitare le anomalie: schemi normalizzati**



- Una relazione è in **Prima Forma Normale (1NF)**, se e solo se ogni componente di ogni tupla ha un valore atomico, non ulteriormente decomponibile.

## 2<sup>a</sup> Forma Normale: un esempio

- Si consideri lo schema,  
MAGAZZINI(Articolo, Magazzino, Quantità, Indirizzo)
- i vincoli (FD)  
Articolo, Magazzino  $\rightarrow$  Quantità, Indirizzo (AM  $\rightarrow$  QI)  
Magazzino  $\rightarrow$  Indirizzo (M  $\rightarrow$  I)
- e l'istanza legale

Articolo	Magazzino	Quantità	Indirizzo
scarpe	VR1	25000	v. Albere 17 - Verona
pantaloni	VR1	18000	v. Albere 17 - Verona
scarpe	BO1	4500	v. Agucchi 3 - Bologna
camicie	VR2	7000	v. Monti 6 - Verona

I problemi sono dovuti a M  $\rightarrow$  I:

ogni tupla memorizza informazioni individuate da un valore della chiave AM, ma l'indirizzo **dipende solo parzialmente** dalla chiave (il valore di I non varia al variare del solo A).



## 2<sup>a</sup> Forma Normale: definizione

- **Attributo primo:** dato uno schema R(T), un attributo  $A \in T$  è **primo** se e solo se fa parte di almeno una chiave dello schema. In caso contrario A è detto **non-primo**.
- Nello schema MAGAZZINI(Articolo, Magazzino, Quantità, Indirizzo) Articolo e Magazzino sono primi, Quantità e Indirizzo sono non-primi.
- **Seconda Forma Normale - Definizione:**
  - Uno schema R(T) con vincoli F è in **2NF** se e solo se ogni **attributo non-primo** dipende **completamente (non parzialmente)** da ogni chiave candidata dello schema,
- **ovvero se**
  - non c'è **dipendenza parziale** di un attributo non-primo da una chiave

Uno schema in 1NF le cui chiavi siano tutte "semplici", ovvero formate da un singolo attributo, è anche in 2NF.





## Normalizzazione in 2NF

- La soluzione consiste nell'estrarre la FD che crea i problemi, generando gli schemi

MAG\_ART(Articolo,Magazzino,Quantità) (AM → Q)

MAG\_IND(Magazzino,Indirizzo) (M → I)

Articolo	Magazzino	Quantità
scarpe	VR1	25000
pantaloni	VR1	18000
scarpe	BO1	4500
camicie	VR2	7000

Magazzino	Indirizzo
VR1	v. Albere 17 - Verona
BO1	v. Agucchi 3 - Bologna
VR2	v. Monti 6 - Verona

- L'informazione originale si può ricostruire eseguendo un join tra le 2 tabelle

MAGAZZINO=MAG\_ART ▷◁ MAG\_IND



## Ancora anomalie

- Si consideri lo schema in 2NF,  
IMPIEGATI(Imp\_cod,Nome,Reparto,Capo\_reparto)
- i vincoli (FD)  
Imp\_cod → Nome,Reparto,Capo\_reparto (I → NRC)  
Reparto → Capo\_reparto (R → C)
- e l'istanza legale

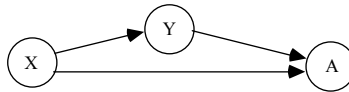
Imp_cod	Nome	Reparto	Caporeparto
#001	Rossi	Vendite	Marchi
#002	Verdi	Acquisti	Stefani
#003	Bianchi	Magazzino	Bielli
#004	Neri	Vendite	Marchi

I problemi sono dovuti a R → C:  
C dipende transitivamente dalla chiave I.



## 3<sup>a</sup> Forma Normale: definizione

- **Dipendenza transitiva:** dato uno schema  $R(T)$ ,  $X \subseteq T$ ,  $A \in T$ ,  $A$  **dipende transitivamente** da  $X$  se esiste  $Y \subset T$  tale che:
  - 1.  $X \rightarrow Y$             {X determina Y}
  - 2.  $\neg (Y \rightarrow X)$         {Y non determina X}
  - 3.  $Y \rightarrow A$             {Y determina A....}
  - 4.  $A \notin Y$             {...non banalmente}



- **Terza Forma Normale - Definizione:**
  - Uno schema  $R(T)$  con vincoli  $F$  è in **3NF** se e solo se ogni attributo non-primario **non dipende transitivamente** da nessuna chiave,
- **ovvero se**
  - non c'è **dipendenza transitiva** di un attributo non-primario da una chiave



## Normalizzazione in 3NF

- Anche in questo caso la soluzione consiste nell'**estrarre** la FD che crea i problemi, generando gli schemi

REP\_IMP(Imp\_cod, Nome, Reparto) (I → NR)

REP\_CAPO(Reparto, Capo\_reparto) (R → C)

Imp_cod	Nome	Reparto
#001	Rossi	Vendite
#002	Verdi	Acquisti
#003	Bianchi	Magazzino
#004	Neri	Vendite

Reparto	Caporeparto
Vendite	Marchi
Acquisti	Stefani
Magazzino	Bielli

- L'informazione originale si può ricostruire eseguendo un join tra le 2 tabelle

IMPIEGATI=REP\_IMP ▷◁ REP\_CAPO



## Esempio di normalizzazione

- Anche lo schema di riferimento non è normalizzato (non è in 3NF nè in 2NF), la soluzione è "decomporlo", sulla base delle FD

Impiegato   Stipendio   Progetto   Bilancio   Funzione

Impiegato → Stipendio

Impiegato, Progetto → Funzione

Progetto → Bilancio

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

Impiegato	Progetto	Funzione
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

Progetto	Bilancio
Marte	2
Giove	15
Venere	15

## Attenzione!

- La soluzione **non è sempre così semplice**, bisogna fare anche altre considerazioni; ad esempio, operando come prima:

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



Impiegato	Sede
Rossi	Roma
Verdi	Milano
Neri	Milano

Progetto	Sede
Marte	Roma
Giove	Milano
Saturno	Milano
Venere	Milano

Impiegato → Sede

Progetto → Sede



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

...se proviamo a tornare indietro  
(Join su Sede):

Diversa dalla relazione di partenza!

## Decomposizione senza perdita

- La decomposizione **non** deve assolutamente **alterare** il contenuto informativo del DB
- Si introduce pertanto il seguente requisito

### Decomposizione senza perdita (*lossless*)


Uno schema  $R(X)$  si decompone senza perdita negli schemi  $R1(X1)$  e  $R2(X2)$  se, per ogni istanza legale  $r$  su  $R(X)$ , il join naturale delle proiezioni di  $r$  su  $X1$  e  $X2$  è uguale a  $r$  stessa:

$$\pi_{X1}(r) \bowtie \pi_{X2}(r) = r$$

- Una decomposizione con perdita può generare **tuple spurie**
- Per decomporre senza perdita è necessario e sufficiente che il **join naturale sia eseguito su una superchiave di uno dei due sottoschemi**, ovvero che valga  $X1 \cap X2 \rightarrow X1$  oppure  $X1 \cap X2 \rightarrow X2$

## Esempio di decomposizione *lossless*

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



Impiegato	Sede
Rossi	Roma
Verdi	Milano
Neri	Milano

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

OK!

... ma i problemi non sono ancora finiti...

## Ancora anomalie

- Lo schema  $TEL(Pref, Num, Località, Abbonato, Via)$  con vincoli
  - $Pref, Num \rightarrow Località, Abbonato, Via$  (PN  $\rightarrow$  LAV)
  - $Località \rightarrow Pref$  (L  $\rightarrow$  P)
- Nella seguente istanza legale l'informazione sul prefisso viene replicata per ogni abbonato.

Pref	Numero	Località	Abbonato	Via
051	432175	Bologna	Rossi M.	Mazzini 124
059	272225	Modena	Bianchi G.	Emilia 233
051	227951	Bologna	Rossi M.	Amendola 14
051	314255	Castenaso	Neri E.	Mazzini 7
059	227951	Vignola	Verdi P.	Roma 14

Lo schema è in 3NF, in quanto **Pref** è primo (non c'è dipendenza transitiva)

## Forma normale di Boyce-Codd

- La 3NF mira a risolvere i problemi relativi agli attributi non-primi. La BCNF estende le considerazioni sinora svolte anche agli **attributi primi**
- **Forma Normale di Boyce-Codd (BCNF)** - Definizione:
  - Uno schema  $R(T)$  è in forma normale di Boyce e Codd se, per ogni dipendenza funzionale (non banale)  $X \rightarrow A$  definita su di esso,  $X$  è una superchiave di  $R(T)$
- **ovvero se**
  - non c'è **dipendenza transitiva** di un attributo a **sinistra** della dipendenza che **non sia superchiave** dello schema
- Lo schema  $TEL(Pref, Num, Località, Abbonato, Via)$  con vincoli
  - $Pref, Num \rightarrow Località, Abbonato, Via$  (PN  $\rightarrow$  LAV)
  - $Località \rightarrow Pref$  (L  $\rightarrow$  P)

**non è in BCNF** a causa della FD  $Località \rightarrow Pref$  in cui  $Località$  non è superchiave

## Una decomposizione non corretta

- La seguente decomposizione non va bene, perché non è lossless

NUM\_TEL(Pref, Num, Abbonato, Via)

PREF\_TEL(Località, Pref)

L'attributo importato in NUM\_TEL non è la chiave della relazione PREF\_TEL

- Non è possibile risalire univocamente all'indirizzo dell'abbonato (in presenza di più località con lo stesso prefisso)

Pref	Numero	Abbonato	Via
051	432175	Rossi M.	Mazzini 124
059	272225	Bianchi G.	Emilia 233
051	227951	Rossi M.	Amendola 14
051	314255	Neri E.	Mazzini 7
059	227951	Verdi P.	Roma 14

Pref	Località
051	Bologna
059	Modena
051	Castenaso
059	Vignola

Dove vive l'abbonato "Rossi M."? A Bologna o a Castenaso?

## Una soluzione corretta

- Una soluzione corretta consiste nel decomporre lo schema in:

NUM\_TEL(Num, Località, Abbonato, Via)

PREF\_TEL(Località, Pref)

Numero	Località	Abbonato	Via
432175	Bologna	Rossi M.	Mazzini 124
272225	Modena	Bianchi G.	Emilia 233
227951	Bologna	Rossi M.	Amendola 14
314255	Castenaso	Neri E.	Mazzini 7
227951	Vignola	Verdi P.	Roma 14

Pref	Località
051	Bologna
059	Modena
051	Castenaso
059	Vignola

La decomposizione è lossless perché

$(\text{NUM\_TEL} \bowtie \text{PREF\_TEL}) = \text{TEL}$

ma presenta ancora problemi...

## Modifichiamo il DB...

- Supponiamo di voler inserire un nuovo abbonato:

Numero	Località	Abbonato	Via
432175	Bologna	Rossi M.	Mazzini 124
272225	Modena	Bianchi G.	Emilia 233
227951	Bologna	Rossi M.	Amendola 14
314255	Castenaso	Neri E.	Mazzini 7
227951	Vignola	Verdi P.	Roma 14
227951	Modena	Gialli E.	Milano 4

Pref	Località
059	Modena
051	Bologna
051	Castenaso
059	Vignola

- Ricostruendo la relazione si ottengono 2 tuple con lo stesso numero di telefono:

Pref	Numero	Località	Abbonato	Via
051	432175	Bologna	Rossi M.	Mazzini 124
059	272225	Modena	Bianchi G.	Emilia 233
051	227951	Bologna	Rossi M.	Amendola 14
051	314255	Castenaso	Neri E.	Mazzini 7
059	227951	Vignola	Verdi P.	Roma 14
059	227951	Modena	Gialli E.	Milano 4

Normalizzazione di schemi relazionali

29

## Attenzione ai vincoli!

- Una istanza legale nello schema decomposto genera sullo schema ricostruito ( $\text{NUM\_TEL} \triangleright \triangleleft \text{PREF\_TEL}$ ) una soluzione **non ammissibile**
- Ogni singola istanza è (“localmente”) legale, ma il DB (“globalmente”) non lo è.  
Infatti esistono due abbonati (**Bianchi G. e Gialli E.**) con lo stesso numero di telefono (**059-227951**).
- 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 FD  $\text{Pref, Num} \rightarrow \text{Località}$  non è rispettata nel DB e nessuno dei due schemi include tutti e tre gli attributi.

Normalizzazione di schemi relazionali

30

## Preservazione delle dipendenze

- Si dice che 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  $\text{Pref,Num} \rightarrow \text{Località}$  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**
- In generale si devono prima eseguire **query SQL di verifica**



## Esempio di query di verifica

- Bisogna verificare che la DF  $\text{Pref,Num} \rightarrow \text{Località}$  sia conservata, a tal fine per inserire un nuovo abbonato occorre controllare che non esista nessun altro abbonato in una località con lo stesso prefisso di **Modena** che abbia lo stesso numero di telefono **227951**.

```
SELECT *          -- OK se non restituisce alcuna tupla
FROM   NUM_TEL N
WHERE  N.Numero = '227951'
      AND N.Località IN ( SELECT P1.Località
                        FROM   PREF_TEL P1, PREF_TEL P2
                        WHERE  P1.Pref = P2.Pref
                        AND    P1.Località = `Modena` )
```





## Osseervazioni

- Benché gli schemi in **3NF** non siano esenti da problemi, tale livello di normalizzazione è **comunemente accettato** nella pratica.
- Nel caso generale, problemi di complessità computazionale rendono imprononibile affrontare l'attività di normalizzazione mediante tecniche di "analisi". Tutti i seguenti problemi sono **NP-completi**:
  - Determinare se un attributo è primo
  - Verificare se esiste una chiave di grado minore di  $k$  ( $k$  costante)
  - Verificare se uno schema è in 3NF rispetto a un un insieme di FD
- L'approccio adottato è di tipo **costruttivo**, ovvero *anziché verificare se uno schema è al livello di normalizzazione richiesto, si progettano schemi che siano a tale livello di normalizzazione.*
- Qualità di una decomposizione (ottenibile con algoritmi di normalizzazione):
  - **deve** essere **senza perdita**, per garantire la ricostruzione delle informazioni originarie
  - **dovrebbe conservare le dipendenze**, per semplificare il mantenimento dei vincoli di integrità originari

## Decomposizione in 3NF

- L'idea alla base dell'algoritmo che produce una decomposizione in 3NF è creare **una relazione per ogni gruppo di FD che hanno lo stesso lato sinistro (determinante)** e inserire nello schema corrispondente gli attributi coinvolti in almeno una FD del gruppo

**Esempio:** Se le FD individuate sullo schema  $R(\underline{A}BCDEFG)$  sono:

$$AB \rightarrow CD, AB \rightarrow E, C \rightarrow F, F \rightarrow G$$

si generano gli schemi  $R_1(\underline{A}BCDE)$ ,  $R_2(\underline{C}F)$ ,  $R_3(\underline{F}G)$

- Se **2 o più determinanti si determinano reciprocamente**, si fondono gli schemi (più chiavi alternative per lo stesso schema)

**Esempio:** Se le FD su  $R(\underline{A}BCD)$  sono:  $A \rightarrow BC$ ,  $B \rightarrow A$ ,  $C \rightarrow D$

si generano gli schemi  $R_1(\underline{A}BC)$ ,  $R_2(\underline{C}D)$  con  $B$  chiave in  $R_1$

- Alla fine si verifica che **esista uno schema la cui chiave è anche chiave dello schema originario (se non esiste lo si crea)**

**Esempio:** Se le FD su  $R(\underline{A}BCD)$  sono:  $A \rightarrow C$ ,  $B \rightarrow D$

si generano gli schemi  $R_1(\underline{A}C)$ ,  $R_2(\underline{B}D)$ ,  $R_3(\underline{A}B)$

## Una limitazione non superabile

- In funzione del pattern di FD, può non essere possibile decomporre in BCNF e preservare le FD

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

Progetto, Sede → Dirigente  
Dirigente → Sede

- Progetto, Sede → Dirigente coinvolge tutti gli attributi e quindi nessuna decomposizione può preservare tale dipendenza!

## In pratica...

- Se la relazione non è normalizzata si decompone in terza forma normale
- Si verifica se lo schema ottenuto è anche in BCNF
  - Si noti che se una relazione ha una sola chiave allora le due forme normali coincidono
- Se uno schema non è in BCNF si hanno 3 alternative:
  - 1) Si lascia così com'è, gestendo le anomalie residue (se l'applicazione lo consente)
  - 2) Si decompone in BCNF, predisponendo opportune query di verifica
  - 3) Si cerca di rimodellare la situazione iniziale, al fine di permettere di ottenere schemi BCNF

## Decomposizione dello schema

- Decomposizione in BCNF per (Dirigente, Progetto, Sede) , con FD  
 $\text{Progetto, Sede} \rightarrow \text{Dirigente}$   
 $\text{Dirigente} \rightarrow \text{Sede}$
- È innanzitutto opportuno osservare che (Progetto, Dirigente) è una chiave
- La decomposizione:

non va bene, perché è con perdita!

ProgSedi

Progetto	Sede
Marte	Roma
Marte	Milano
Giove	Milano
Saturno	Milano
Venere	Milano

Dirigenti

Dirigente	Sede
Rossi	Roma
Verdi	Milano
Neri	Milano

- La decomposizione **corretta** è:  
 ma occorre una query di verifica per la FD  $\text{Progetto, Sede} \rightarrow \text{Dirigente}$

ProgDir

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

Dirigenti

Dirigente	Sede
Rossi	Roma
Verdi	Milano
Neri	Milano

## Ridefinizione dello schema

- Nell'esempio si può introdurre il concetto di **Reparto** per distinguere i dirigenti di una stessa sede (ogni dirigente opera in un reparto di una sede, e viceversa)

Dirigente	Progetto	Sede	Reparto
Rossi	Marte	Roma	1
Verdi	Giove	Milano	1
Verdi	Marte	Milano	1
Neri	Saturno	Milano	2
Neri	Venere	Milano	2

$\text{Dirigente} \rightarrow \text{Sede, Reparto}$   
 $\text{Sede, Reparto} \rightarrow \text{Dirigente}$   
 $\text{Progetto, Sede} \rightarrow \text{Reparto}$

- È ora possibile operare una decomposizione in **BCNF**

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

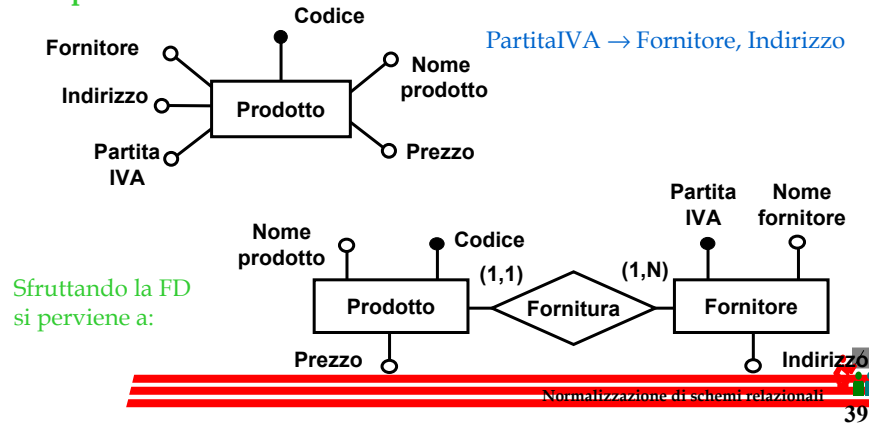
Progetto	Sede	Reparto
Marte	Roma	1
Giove	Milano	1
Marte	Milano	1
Saturno	Milano	2
Venere	Milano	2

La difficoltà a decomporre in BCNF era forse dovuta a un'analisi poco accurata

## Progettazione e normalizzazione

- La teoria della normalizzazione può essere usata nella progettazione logica per **verificare lo schema relazionale finale**
- Si può usare anche durante la progettazione concettuale per **verificare la qualità dello schema concettuale**

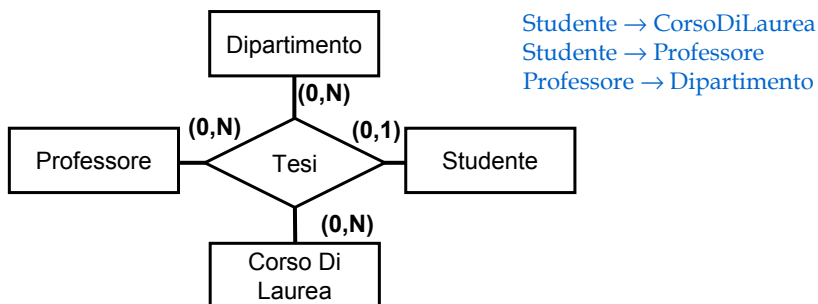
Esempio:



39

## Esempio: analisi di associazioni n-arie (1)

- Le associazioni n-arie spesso nascondono FD che possono dar luogo a schemi non normalizzati



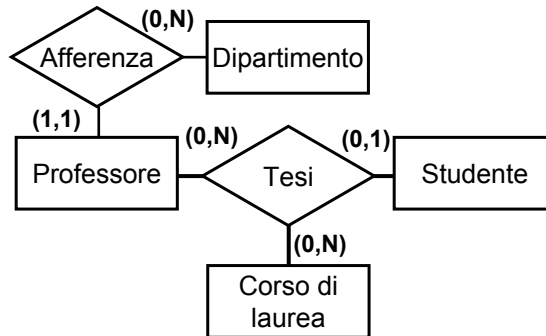
Tesi(Stu-dente, Professore, Dipartimento, CorsoDiLaurea)

non è in 3NF a causa di Professore → Dipartimento

40

## Esempio: analisi di associazioni n-arie (2)

- Si ristrutturata lo schema di conseguenza:



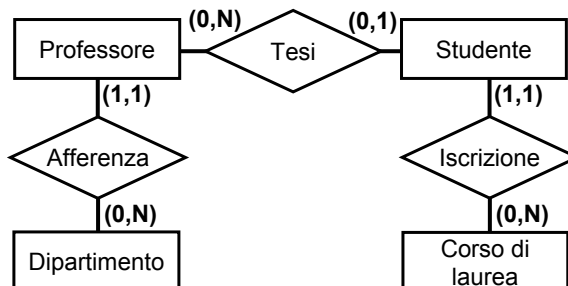
Tesi(Studente, Professore, CorsoDiLaurea) è ora in BCNF

## Esempio: analisi di associazioni n-arie (3)

- L'associazione Tesi in realtà include 2 FD, tra loro indipendenti:

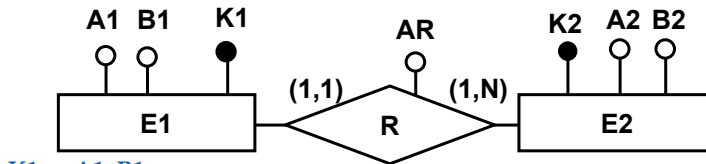
Studente → CorsoDiLaurea (iscrizione)  
 Studente → Professore (per chi ha un relatore)

- È quindi opportuno procedere a un'ulteriore ristrutturazione:



## FD e modello E/R

- È bene abituarsi a “leggere” uno schema E/R anche in termini di FD
- A tal fine si considerano le **cardinalità massime delle associazioni**:



$K1 \rightarrow A1, B1$

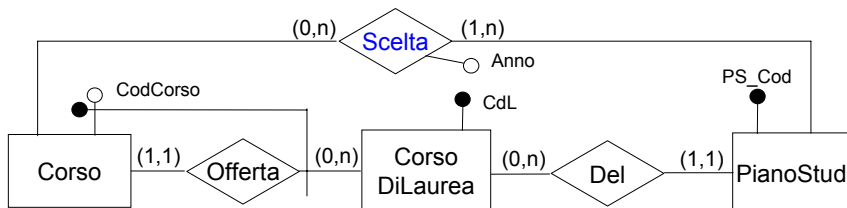
$K2 \rightarrow A2, B2$

$K1 \rightarrow K2, AR$  poiché  $\max\text{-card}(E1, R) = 1$

- Si suggerisce di rivedere le regole per la traduzione delle associazioni in termini di FD tra gli identificatori delle entità e di normalizzazione degli schemi...

## Possiamo fare a meno delle FD?

- Anche se in molti casi una buona progettazione concettuale rende superfluo ragionare in termini di FD, vi sono degli schemi E/R “corretti” che danno luogo a schemi relazionali non normalizzati



La traduzione dell'associazione **Scelta** genera lo schema

$Scelta(CdL, CodCorso, PS\_Cod, Anno)$

che **non è in 3NF** a causa della FD  $PS\_Cod \rightarrow CdL$

Inoltre quella individuata è solo una superchiave, ovvero

$Scelta(CdL, CodCorso, PS\_Cod, Anno)$

## Normalizzare o no?

- La normalizzazione non va intesa come un obbligo, in quanto in alcune situazioni le anomalie che si riscontrano in schemi non normalizzati sono un male minore rispetto alla situazione che si viene a creare normalizzando
- In particolare, le cose da considerare sono:
  - Normalizzare elimina le anomalie, ma può appesantire l'esecuzione di certe operazioni (join tra gli schemi normalizzati)
  - La frequenza con cui i dati vengono modificati incide su qual è la scelta più opportuna (relazioni "quasi statiche" danno meno problemi se non normalizzate)
  - La ridondanza presente in relazioni non normalizzate va quantificata, per capire quanto incida sull'occupazione di memoria, e sui costi da pagare quando le repliche di una stessa informazione devono essere aggiornate

## Esercizio

- È dato il seguente schema relazionale:  
R(Prodotto, Componente, TipoComponente, Quantità, PrezzoComponente, Fornitore, PrezzoProdotto)

Prodotto	Tipo Componente	Componente	Quantità	Prezzo Componente	Fornitore	Prezzo Prodotto
Libreria	Legno	Noce	50	10000	Forrest	400000
Libreria	Bulloni	B212	200	100	Bolt	400000
Libreria	Vetro	Cristal	3	5000	Clean	400000
Scaffale	Legno	Mogano	5	15000	Forrest	300000
Scaffale	Bulloni	B212	250	100	Bolt	300000
Scaffale	Bulloni	B412	150	300	Bolt	300000
Scrivania	Legno	Noce	10	8000	Wood	250000
Scrivania	Maniglie	H621	10	20000	Bolt	250000
Tavolo	Legno	Noce	4	10000	Forrest	200000

## Esercizio

- Sapendo che:

- L'attributo "PrezzoProdotto" rappresenta il costo totale di ciascun prodotto
- Ciascun fornitore fornisce diversi componenti e ciascun componente può essere fornito da fornitori diversi
- Un componente per un determinato prodotto viene fornito da un solo fornitore
- Gli attributi "Quantità" e "PrezzoComponente" indicano, per ciascun componente, la quantità necessaria per la costruzione di un determinato prodotto e il relativo costo totale

individuare le **dipendenze funzionali** e la **chiave** della relazione e proporre un **scomposizione in BCNF**



## Riassumiamo:

- Una forma normale è una proprietà di uno schema relazionale che ne garantisce la "qualità", cioè **l'assenza di determinati difetti**
- Una relazione non normalizzata **presenta ridondanze** e dà luogo a **comportamenti poco desiderabili** durante gli aggiornamenti
- La definizione delle **forme normali** (2NF, 3NF e BCNF) si basa sul vincolo di **dipendenza funzionale (FD)**
- Normalizzare uno schema significa **decomporlo in sottoschemi**
- **Ogni decomposizione deve essere senza perdita**, ovvero deve permettere di ricostruire esattamente la relazione originaria non decomposta
- È anche opportuno che la decomposizione **preservi le FD**, al fine di **evitare** (o ridurre la complessità di) **query di verifica** che garantiscano che i vincoli siano rispettati

