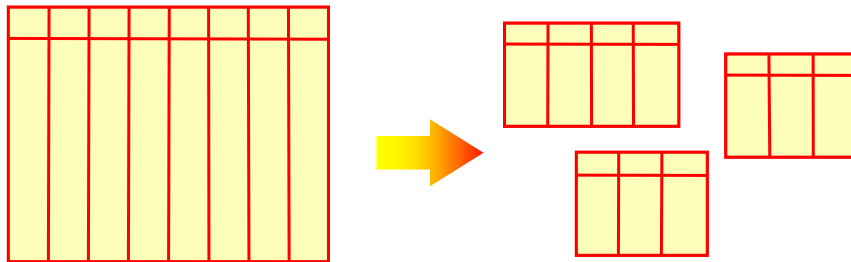


Normalizzazione di schemi relazionali



Dario Maio

<http://bias.csr.unibo.it/maio/>

Normalizzazione di schemi relazionali



1

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 deve essere utilizzata come tecnica di verifica dei risultati della progettazione di una base di dati.

Normalizzazione di schemi relazionali



2

Una relazione con anomalie

matricola cognome nome



Matricola	Cognome	Nome	nomeFac	indFac
29321	Bianchi	Giorgio	Ingegneria	Risorgimento 2
24467	Rossi	Lucia	Medicina	Massarenti 9
26654	Verdi	Marco	Ingegneria	Risorgimento 2
41132	Neri	Anna	Agraria	Fanin 50
11375	Viola	Carlo	Agraria	Fanin 50

- ✚ L'indirizzo di una facoltà è ripetuto in tutte le tuple dei suoi studenti:
ridondanza
- ✚ Se l'indirizzo di una facoltà cambia, è necessario modificare il valore in diverse tuple:
anomalia di aggiornamento
- ✚ Una nuova facoltà senza studenti non può essere inserita:
anomalia di inserimento
- ✚ ...

Normalizzazione di schemi relazionali



3

Un altro esempio di relazione con anomalie

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

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.

Normalizzazione di schemi relazionali



4



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.

Normalizzazione di schemi relazionali



5



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



6

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



7

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

Matr	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

Matr	Tutor	Tel
125233	Mario	7575
127988	Carlo	5566
150444	Carlo	5566
190787	Mario	7575

duplicazione di dati **ridondante**

Normalizzazione di schemi relazionali



8

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

Normalizzazione di schemi relazionali



9

Dipendenza funzionale

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

Si considerino:

- un'istanza r di uno schema $R(X)$;
- due sottoinsiemi (non vuoti) di attributi Y e Z di X .
- Si dice che in r vale la dipendenza funzionale (FD) $Y \rightarrow Z$ (Y determina funzionalmente Z) se

$$\forall t_1, t_2 \in r : t_1[Y] = t_2[Y] \Rightarrow t_1[Z] = t_2[Z]$$

per ogni coppia di tuple 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

Normalizzazione di schemi relazionali



10

Esempi di FD

- Nella relazione

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

 si hanno diverse FD, tra cui:

$\text{Impiegato} \rightarrow \text{Stipendio}$
 $\text{Progetto} \rightarrow \text{Bilancio}$
 $\text{Impiegato, Progetto} \rightarrow \text{Funzione}$

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

$\text{Impiegato, Progetto} \rightarrow \text{Progetto}$

- 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 .

Normalizzazione di schemi relazionali



11

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 è 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 (sono determinati) dai valori della componente X .
- Se $X \rightarrow Y$ non necessariamente risulta anche $Y \rightarrow X$

Normalizzazione di schemi relazionali



12

FD e Superchiavi

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

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

Dimostrazione

- **(se)** Se $K \rightarrow T$ allora per ogni istanza legale r si ha che $\forall t1, t2 \in r : t1[K] = t2[K] \Rightarrow t1[T] = t2[T]$, ovvero $t1 = t2$. 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 $t1[K] = t2[K] \Rightarrow t1 = t2$, e quindi $t1[T] = t2[T]$.



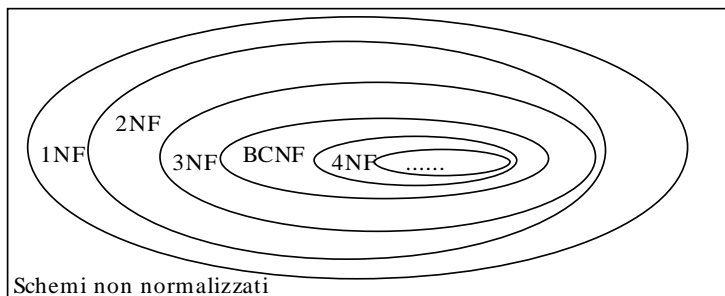
Anomalie e FD

- Le anomalie viste si riconducono alla presenza delle FD:
 $\text{Impiegato} \rightarrow \text{Stipendio}$
 $\text{Progetto} \rightarrow \text{Bilancio}$
- viceversa la FD
 $\text{Impiegato, Progetto} \rightarrow \text{Funzione}$
non causa problemi.
- Motivazioni:
 - 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



- Il processo di normalizzazione fu inizialmente introdotto da Codd (1972) con la definizione delle prime tre forme normali (**1NF**, **2NF**, **3NF**). In seguito Boyce e Codd definirono una forma più restrittiva di 3NF denominata **BCNF**. Tutte queste forme normali si basano sulle dipendenze funzionali tra gli attributi di una relazione.
- Più tardi furono definite altre forme normali (**4NF**, **5NF**) basate sulle dipendenze multivalore e sulle dipendenze di join.



1ª Forma Normale

- Una relazione è in **Prima Forma Normale (1NF)** se e solo se il dominio di ciascun attributo comprende solo **valori atomici** (semplici, indivisibili) e il valore di ciascun attributo in una tupla è un **valore singolo** del dominio di quell'attributo.
- Dunque 1NF non permette "relazioni dentro relazioni" e "relazioni come attributi di tuple". I soli valori di attributi ammissibili sono i singoli valori atomici (non ulteriormente scomponibili a parte funzioni speciali) **rispetto al RDMBS (definizione di atomicità secondo Codd)**.
- Oggi 1NF è considerata parte integrante della definizione formale di relazione del **modello relazionale di base**. Nel modello relazionale nidificato (**nested relational model**) e nel modello relazionale a oggetti (**object relational model**) sono invece consentite relazioni UNF (Unnormalized Form).
- Con **non-scomponibilità** di un attributo non dobbiamo intendere che il valore dell'attributo non possa essere suddiviso in sotto-parti (ad esempio, l'indirizzo può essere scomposto in "Via", "Risorgimento", "22", o addirittura in tutti i caratteri che lo compongono). Quello che importa è che ogni valore dell'attributo sia dal punto di vista semantico un **informazione unica**: ad esempio, non si possono inserire due sedi nel campo "Sede" in quanto si tratta di due informazioni semanticamente distinte. In pratica, riprendendo il concetto di insieme, un attributo può assumere uno e un solo valore, preso fra gli elementi del suo dominio.





1ª Forma Normale - note

- What First Normal Form really means and means not, Chris Date, 2003: a table is in 1NF **if and only if** it is "isomorphic to some relation", which means, specifically, that it satisfies the following five conditions:

1. There's no top-to-bottom ordering to the rows.
2. There's no left-to-right ordering to the columns.
3. There are no duplicate rows.
4. Every row-and-column intersection contains exactly one value from the applicable domain (and nothing else).
5. All columns are regular [i.e. rows have no hidden components such as row IDs, object IDs, or hidden timestamps].

Esempi di violazione: una tabella che non ha una chiave primaria; una vista che obbliga la presentazione dei risultati in un ordine particolare che dà significato alla vista stessa; una tabella che presenta attributi con valori nulli.

■ **Un attributo che ammette valori nulli viola la condizione 4** (NULL non fa parte del dominio). Questo aspetto molto controverso ha dato luogo a un dibattito ampio e nella pratica si tende a tollerare la presenza di NULL in 1NF. Comunque è bene fare attenzione a limitare l'uso dei NULL per le molteplici implicazioni logiche che comporta.

Normalizzazione di schemi relazionali



17



1ª Forma Normale: esempio A (1)

- Si consideri lo schema:

DIPARTIMENTO(CodDip, Nome, CodDir, SediDip)

e l'istanza

<u>CodDip</u>	Nome	CodDir	SediDip
D0001	Amministrazione	33301	(Milano, Napoli, Roma)
D0005	Produzione	18007	Aprilia
D0003	Ricerca	33010	Napoli

La relazione non è in 1NF a causa dell'attributo SediDip. Sono possibili due interpretazioni:

1. il dominio di SediDip contiene valori atomici ma alcune tuple hanno un insieme di questi valori, in questo caso SediDip non dipende funzionalmente da CodDip;
2. Il dominio di SediDip contiene insiemi di valori e perciò non è atomico; in questo caso $\text{CodDip} \rightarrow \text{SediDip}$ poiché ogni insieme è considerato un unico membro del dominio dell'attributo, ovvero il dominio di SediDip è l'insieme potenza dell'insieme delle singole sedi.

Normalizzazione di schemi relazionali



18

1ª Forma Normale: esempio A (2)

- **Soluzione 1:** si espande la chiave in modo da avere tuple separate per ogni sede differente di un dipartimento.

DIPARTIMENTO(CodDip, Nome, CodDir, SedeDip)

<u>CodDip</u>	Nome	<u>CodDir</u>	<u>SedeDip</u>
D0001	Amministrazione	33301	Milano
D0001	Amministrazione	33301	Napoli
D0001	Amministrazione	33301	Roma
D0005	Produzione	18007	Aprilia
D0003	Ricerca	33010	Napoli

Questa soluzione ha lo svantaggio di inserire ridondanza d'informazione.



1ª Forma Normale: esempio A (3)

- **Soluzione 2:** se è noto a priori il numero massimo N di sedi che può avere un dipartimento si può sostituire l'attributo SedeDip con N attributi separati, ad esempio nel caso di N=3

DIPARTIMENTO(CodDip, Nome, CodDir, Sede1, Sede2, Sede3)

<u>CodDip</u>	Nome	<u>CodDir</u>	<u>Sede1</u>	<u>Sede2</u>	<u>Sede3</u>
D0001	Amministrazione	33301	Milano	Napoli	Roma
D0005	Produzione	18007	Aprilia	Null	Null
D0003	Ricerca	33010	Napoli	Null	Null

Questa soluzione ha lo svantaggio di introdurre valori nulli.



1ª Forma Normale: esempio A (4)

- **Soluzione 3:** si rimuove l'attributo SediDip e lo si pone in un'altra relazione separata con chiave combinazione di CodDip e SedeDip.

DIPARTIMENTO(CodDip, Nome, CodDir)

CodDip	Nome	CodDir
D0001	Amministrazione	33301
D0005	Produzione	18007
D0003	Ricerca	33010

SEDIDIP(CodDip, SedeDip)

CodDip	SedeDip
D0001	Milano
D0001	Napoli
D0001	Roma
D0005	Aprilia
D0003	Napoli

Questa soluzione non presenta ridondanze ed è completamente generale, non presentando limiti sul massimo numero di sedi per un dipartimento.

Normalizzazione di schemi relazionali



21

1ª Forma Normale: esempio B (1)

- Se fosse concesso di avere relazioni nidificate si potrebbe definire lo schema **RIEPILOGO_ORELAVORATE**:

<u>CodImp</u>	Nome	<u>CodProg</u>	<u>Data</u>	OreLav
---------------	------	----------------	-------------	--------

Possibile istanza

0012	Rossi	1023	05/07/2010	5
		1225	15/07/2010	6
		1225	15/10/2010	8
0115	Bianchi	1023	08/07/2010	7
		1128	17/09/2010	3
0085	Verdi	1023	05/07/2010	4
		1023	06/07/2010	6

Una tupla rappresenta un impiegato e una relazione (che riepiloga le ore lavorate dall'impiegato nei vari progetti in varie date).

{CodProg, Data} è la chiave primaria parziale della relazione nidificata, ovvero {CodProg, Data} deve esibire valori unici all'interno di ogni tupla della relazione nidificata.

Normalizzazione di schemi relazionali



22

1ª Forma Normale: esempio B (2)

- La normalizzazione in 1NF porta a progettare gli schemi:

IMPIEGATI

<u>CodImp</u>	Nome
---------------	------

IMP_PROG_ORE

<u>CodImp</u>	<u>CodProg</u>	<u>Data</u>	OreLav
---------------	----------------	-------------	--------

A livello di istanze:

0012	Rossi
0015	Bianchi
0085	Verdi

0012	1023	05/07/2010	5
0012	1225	15/07/2010	6
0012	1225	15/10/2010	8
0015	1023	08/07/2010	7
0015	1128	17/09/2010	3
0085	1023	05/07/2010	4
0085	1023	06/07/2010	6

Si spostano gli attributi della relazione nidificata in una nuova relazione e si propaga la chiave primaria della relazione originaria. La nuova relazione ha come chiave primaria la combinazione della chiave parziale e della chiave primaria della relazione originaria.

Normalizzazione di schemi relazionali



23

2ª Forma Normale: un esempio

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

<u>Articolo</u>	<u>Magazzino</u>	<u>Quantità</u>	<u>Indirizzo</u>
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**).

Normalizzazione di schemi relazionali



24

2ª 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 $MAGAZZINI(\text{Articolo}, \text{Magazzino}, \text{Quantità}, \text{Indirizzo})$ 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 di schemi relazionali



25

Normalizzazione in 2NF

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

$MAG_ART(\text{Articolo}, \text{Magazzino}, \text{Quantità}) \quad (AM \rightarrow Q)$
 $MAG_IND(\text{Magazzino}, \text{Indirizzo}) \quad (M \rightarrow I)$

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 due tabelle:

$MAGAZZINO = MAG_ART \bowtie MAG_IND$

Normalizzazione di schemi relazionali



26

2NF e chiavi candidate

- Una relazione in cui non vi sono dipendenze funzionali parziali dalla chiave primaria è tipicamente in 2NF ma non sempre. Si consideri ad esempio la relazione:

Produttore	Modello	NomeModelloCompleto	Stato
Forte	X-Prime	F X-Prime	Italia
Forte	Ultraclean	F Ultraclean	Italia
Dent-o-Fresh	EZbrush	DoF EZBrush	USA
Kobayashi	ST-60	K ST-60	Giappone
Hoch	Toothmaster	H Toothmaster	Germania
Hoch	X-Prime	H X-Prime	Germania

- La relazione non è in 2NF anche se il progettista ha scelto come chiave primaria {NomeModelloCompleto}. Una chiave candidata è anche {Produttore, Modello} ma Produttore → Stato.
- La trasformazione in 2NF prevede due relazioni:

PRODUTTORE_SPAZZOLINO(Produttore,Stato)

MODELLO_SPAZZOLINO(Produttore,Modello,NomeModelloCompleto)

Normalizzazione di schemi relazionali



27

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.

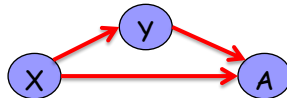
Normalizzazione di schemi relazionali



28

3^a 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:

$REP_IMP(Imp_cod, Nome, Reparto) \quad (I \rightarrow NR)$
 $REP_CAPO(Reparto, Capo_reparto) \quad (R \rightarrow C)$

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 due tabelle:

$IMPIEGATI = REP_IMP \bowtie REP_CAPO$

Esempio di normalizzazione

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

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

Impiegato → Stipendio Impiegato, Progetto → Funzione Progetto → Bilancio

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

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

Normalizzazione di schemi relazionali



31

Esercizio riepilogativo (1)

- Si consideri il seguente schema relazionale:

TEST_LAB (matrStudiante, nomeStudiante, codCorso, nomeCorso, codTitolare, nomeTitolare, codEsaminatore, nomeEsaminatore, dataProva, voto)

Sono registrate anche eventuali prove non superate. Un corso ha un solo professore titolare, che non coincide necessariamente con il professore esaminatore. **Si evidenzino tutte le dipendenze funzionali non banali e problematiche presenti nello schema.** Se lo schema risulta non normalizzato, si determini un insieme di schemi che siano in **terza forma normale** e risultino equivalenti, dal punto di vista informativo, allo schema dato.

- | | |
|---|-------------------------|
| D1) <u>matrStudiante</u> → nomeStudiante | } Dipendenze parziali |
| D2) <u>codCorso</u> → nomeCorso | |
| D3) <u>codCorso</u> → codTitolare | |
| D4) <u>codTitolare</u> → nomeTitolare | } Dipendenze transitive |
| D5) <u>codEsaminatore</u> → nomeEsaminatore | |

A causa della presenza di dipendenze funzionali parziali la relazione non è in 2NF. Per ottenere una relazione in 2NF si devono spezzare le dipendenze parziali. Per ottenere una relazione in 3NF si devono poi risolvere le dipendenze funzionali transitive.

Normalizzazione di schemi relazionali



32



Esercizio riepilogativo (2)

TEST_LAB (matrStudente, nomeStudente, codCorso, nomeCorso, codTitolare, nomeTitolare, codEsaminatore, nomeEsaminatore, dataProva, voto)

D1) matrStudente → nomeStudente

D2) codCorso → nomeCorso

D3) codCorso → codTitolare

} Dipendenze parziali

■ Spezzando le dipendenze parziali si ottengono gli schemi in 2NF:

PROVE_LAB (matrStudente, codCorso, codEsaminatore, nomeEsaminatore, dataProva, voto)

STUDENTI (matrStudente, nomeStudente)

CORSI (codCorso, nomeCorso, codTitolare, nomeTitolare)

Normalizzazione di schemi relazionali



33



Esercizio riepilogativo (3)

PROVE_LAB (matrStudente, codCorso, codEsaminatore, nomeEsaminatore, dataProva, voto)

STUDENTI (matrStudente, nomeStudente)

CORSI (codCorso, nomeCorso, codTitolare, nomeTitolare)

D4) codTitolare → nomeTitolare

D5) codEsaminatore → nomeEsaminatore

} Dipendenze transitive

■ Spezzando le dipendenze transitive si ottengono gli schemi in 3NF:

PROVE_LAB (codStudente, codCorso, codEsaminatore, dataProva, voto)

PROFESSORI (codProfessore, nomeProfessore)


STUDENTI (codStudente, nomeStudente)

CORSI (codCorso, nomeCorso, codTitolare)

Normalizzazione di schemi relazionali



34



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


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

diversa dalla relazione di partenza!

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



Normalizzazione di schemi relazionali



35



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 $R_1(X_1)$ e $R_2(X_2)$ se, per ogni istanza legale r su $R(X)$, il join naturale delle proiezioni di r su X_1 e X_2 è uguale a r stessa:

$$\pi_{X_1}(r) \bowtie \pi_{X_2}(r) = r$$

- 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:

$$X_1 \cap X_2 \rightarrow X_1 \text{ oppure } X_1 \cap X_2 \rightarrow X_2$$



Normalizzazione di schemi relazionali



36

Esempio di decomposizione lossless

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



OK!

Impiegato	Sede
Rossi	Roma
Verdi	Milano
Neri	Milano

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



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

Normalizzazione di schemi relazionali



37

Modifica con violazione di una FD

... supponiamo di voler effettuare una modifica:

Neri assegnato anche al progetto **Marte**.

Impiegato	Sede
Rossi	Roma
Verdi	Milano
Neri	Milano

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

... ricostruendo la relazione otteniamo:

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

che viola la FD **Progetto → Sede**

Normalizzazione di schemi relazionali



38

Ancora anomalie

- Si consideri lo schema $TEL(\underline{Pref}, \underline{Num}, Località, Abbonato, Via)$ 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 è 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 $R(T)$,
- Lo schema $TEL(\underline{Pref}, \underline{Num}, Località, Abbonato, Via)$ 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 è corretta, **poiché 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?

Normalizzazione di schemi relazionali



41

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** infatti: $(\text{NUM_TEL} \bowtie \text{PREF_TEL}) = \text{TEL}$

ma presenta ancora problemi...

Normalizzazione di schemi relazionali



42

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



43

Attenzione ai vincoli!

- Una istanza legale nello schema decomposto genera sullo schema ricostruito ($NUM_TEL \bowtie PREF_TEL$) una soluzione **non ammissibile**.
- Ogni singola istanza è ("localmente") legale, ma il DB ("globalmente") non lo è, infatti esistono in questo caso due abbonati (Verdi P. e Gialli E.) che hanno 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 $Pref, Num \rightarrow Località$ non è rispettata nel DB e nessuno dei due schemi include tutti e tre gli attributi.

Normalizzazione di schemi relazionali



44

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 $Pref, Num \rightarrow Località$ 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, prima di effettuare una modifica, si devono eseguire **query SQL di verifica.**

Esempio di query di verifica

- Bisogna verificare che la FD $Pref, Num \rightarrow Località$ 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 P2.Località
                       FROM   PREF_TEL P1, PREF_TEL P2
                       WHERE  P1.Pref = P2.Pref
                       AND    P1.Località = `Modena` )
```

Altro esempio di query di verifica

- Con riferimento all'esempio precedente per evitare che l'inserimento del fatto 'Neri assegnato al progetto Marte provochi la violazione della FD **Progetto → Sede**, si deve verificare il progetto (**Marte**) sia presso la stessa sede dell'impiegato (**Neri**). A tal fine si deve trovare un impiegato che lavora al progetto Marte.

```
SELECT *      -- OK se restituisce una tupla
FROM Impiegati I
WHERE I.Impiegato = 'Neri'
AND I.Sede IN ( SELECT I1.Sede
                  FROM Impiegati I1, ImpProg IP
                  WHERE I1.Impiegato = IP.Impiegato
                  AND IP.Progetto = `Marte` )
```

Qualità delle decomposizioni

Una decomposizione:

- **deve** essere senza perdita, per garantire la ricostruzione delle informazioni originarie
- **dovrebbe** preservare le dipendenze, per semplificare il mantenimento dei vincoli di integrità originari

Nell'esempio, questo suggerisce di inserire anche la relazione:

Progetto	Sede
Marte	Roma
Giove	Milano
Saturno	Milano
Venere	Milano

La query di verifica è ora più semplice



Osservazioni

- 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 improponibile affrontare l'attività di normalizzazione mediante tecniche di "analisi". 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 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 $R1(\underline{A}BCDE)$, $R2(\underline{C}F)$, $R3(\underline{F}G)$.

- 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 $R1(\underline{A}BC)$, $R2(\underline{C}D)$ con B chiave in $R1$.

- 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 $R1(\underline{A}C)$, $R2(\underline{B}D)$, $R3(\underline{A}B)$.



Una limitazione non superabile

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

<u>Dirigente</u>	<u>Progetto</u>	<u>Sede</u>
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

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

Dirigenti

<u>Dirigente</u>	<u>Sede</u>
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

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

Dirigenti

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

Normalizzazione di schemi relazionali



53



Ridefinizione dello schema

- Nell'esempio precedente 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).

<u>Dirigente</u>	<u>Progetto</u>	<u>Sede</u>	<u>Reparto</u>
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:

<u>Progetto</u>	<u>Sede</u>	<u>Reparto</u>
Marte	Roma	1
Giove	Milano	1
Marte	Milano	1
Saturno	Milano	2
Venere	Milano	2

<u>Dirigente</u>	<u>Sede</u>	<u>Reparto</u>
Rossi	Roma	1
Verdi	Milano	1
Neri	Milano	2

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

Normalizzazione di schemi relazionali

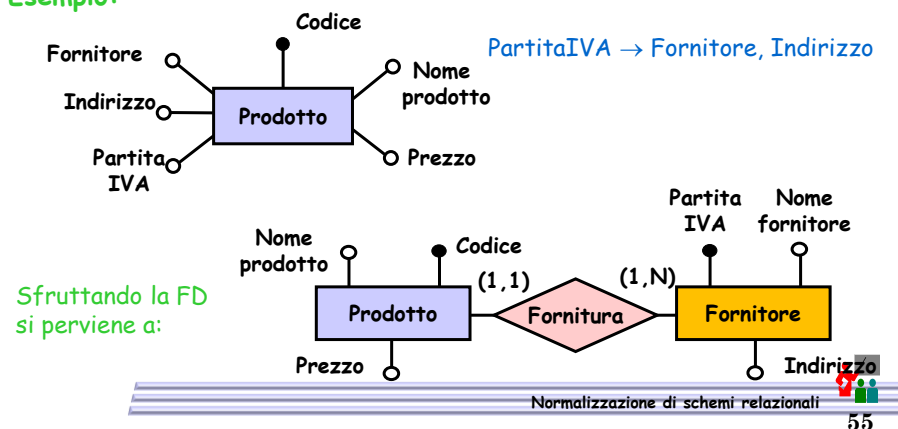


54

Progettazione e normalizzazione

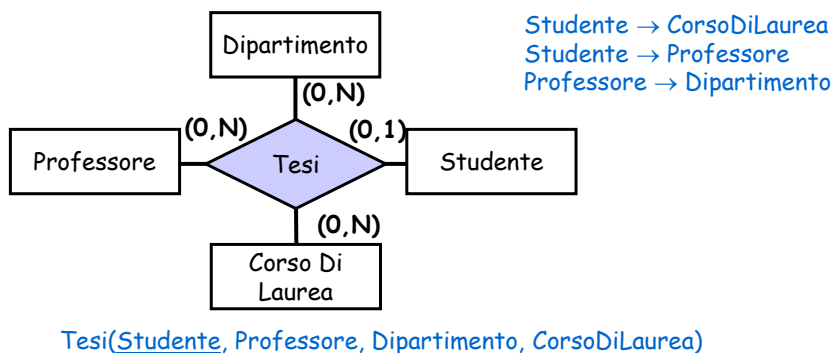
- La teoria della normalizzazione può essere usata nella progettazione logica per **verificare lo schema relazionale finale**.
- Si può usare anche durante la progettazione concettuale per **verificare la qualità dello schema concettuale**.

Esempio:



Esempio: analisi di associazioni n-arie (1)

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



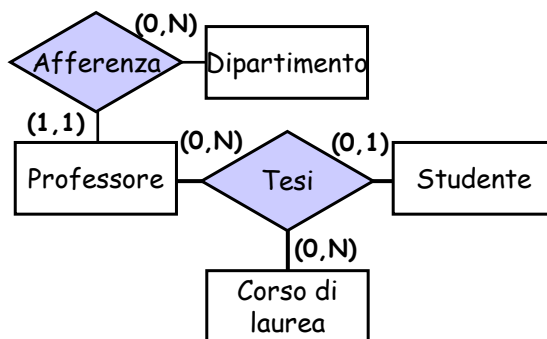
non è in 3NF a causa di Professore → Dipartimento

Normalizzazione di schemi relazionali



Esempio: analisi di associazioni n-arie (2)

- Si ristruttura lo schema di conseguenza:

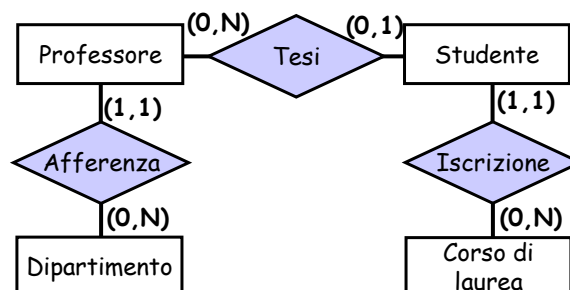


Tesi(Studente, Professore, CorsoDiLaurea) è ora in BCNF.



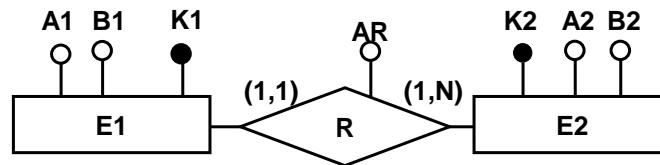
Esempio: analisi di associazioni n-arie (3)

- L'associazione Tesi in realtà include 2 FD, tra loro indipendenti:
 $\text{Studente} \rightarrow \text{CorsoDiLaurea}$ (iscrizione)
 $\text{Studente} \rightarrow \text{Professore}$ (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...

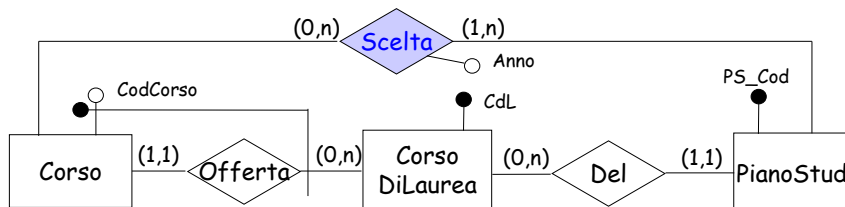
Normalizzazione di schemi relazionali



59

Possiamo fare a meno delle FD?

- Anche se in molti casi una buona progettazione concettuale rende superfluo ragionare in termini di FD, **vi sono 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:

$Scelta(CdL, CodCorso, PS_Cod, Anno)$

Normalizzazione di schemi relazionali



60

Normalizzare o no?

- La normalizzazione non va intesa come un obbligo; infatti in alcune situazioni le anomalie che si riscontrano in schemi non normalizzati sono un male minore rispetto alla situazione che si verrebbe a creare normalizzando.
- In particolare, gli aspetti 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 sono soggetti a modifica incide su qual è la scelta più opportuna (relazioni "quasi statiche" danno un minor numero di problemi se non sono normalizzate);
 - la ridondanza presente in relazioni non normalizzate va quantificata al fine di comprendere quanto possa incidere sull'occupazione di memoria e sui costi da pagare quando le repliche di una stessa informazione devono essere aggiornate.



Esercizio (1)

- È dato il seguente schema relazionale:

R(Prodotto, Componente, TipoComponente, Quantità, CostoComponente, Fornitore, PrezzoProdotto)

Prodotto	Tipo Componente	Componente	Quantità	Costo Componente	Fornitore	Prezzo Prodotto
Libreria	Legno	Noce	5	100	Forrest	2500
Libreria	Bulloni	B212	200	0.10	Bolt	2500
Libreria	Vetro	Fumé	3	150	Clean	2500
Scaffale	Legno	Mogano	5	80	Forrest	1250
Scaffale	Bulloni	B212	250	0.10	Bolt	1250
Scaffale	Bulloni	B412	150	0.20	Bolt	1250
Scrivania	Legno	Noce	10	120	Wood	3000
Scrivania	Maniglie	H621	10	2	Bolt	3000
Tavolo	Legno	Noce	4	100	Forrest	1100



Esercizio (2)

- Sapendo che:

- l'attributo "PrezzoProdotto" rappresenta il prezzo di vendita di ciascun prodotto;
- ciascun fornitore fornisce diversi componenti e ciascun componente può essere fornito da fornitori diversi;
- un componente per un determinato prodotto è fornito da un solo fornitore;
- gli attributi "Quantità" e "CostoComponente" indicano, per ciascun componente, la quantità necessaria per la costruzione di un determinato prodotto e il relativo costo pagato al fornitore.

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



Sommario:

- 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.



Appendice: Algoritmo di normalizzazione 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.
- Per far questo è tuttavia necessario minimizzare l'insieme di FD individuate, altrimenti non è garantito un risultato corretto.

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

$$AB \rightarrow CDEF, C \rightarrow F, F \rightarrow G$$

si genererebbero gli schemi $R1(\underline{A}BCDEF)$, $R2(\underline{C}F)$, $R3(\underline{F}G)$

Ma $R1$ non risulta essere in 3NF a causa della FD $C \rightarrow F$!

- Dunque è opportuno prima ragionare più precisamente con le FD e su come possono essere minimizzate.

Chiusura di X

- Come prima cosa ci chiediamo:
Se F è un insieme di FD su $R(U)$ e X un insieme di attributi, quali attributi di U dipendono funzionalmente da X ?
- Ad es., se F include $A \rightarrow B$ e $B \rightarrow C$, allora è anche vero che $A \rightarrow C$.
Infatti C dipende da B , che a sua volta dipende da A
- Denotiamo con X^+ l'insieme degli attributi di $R(U)$ che dipendono da X
- Calcolare X^+ è semplice:

```
X+ = X;  
repeat  
  termination = true;  
  for each FD in  $F = \{V_i \rightarrow W_i, i=1, \dots, n\}$   
    if  $V_i \subseteq X^+$  and  $W_i \not\subseteq X^+$  then  $\{X^+ = X^+ \cup W_i; \text{termination} = \text{false}\}$   
until termination = true or  $X^+ = U$ 
```

Chiusura di X - esempio

- Supponiamo di avere $F = \{A \rightarrow B, BC \rightarrow D, B \rightarrow E, E \rightarrow C\}$ e calcoliamo A^+ , ovvero l'insieme di attributi che dipendono da A

$A^+ = A$

$A^+ = AB$ usando $A \rightarrow B$

$A^+ = ABE$ usando $B \rightarrow E$

$A^+ = ABEC$ usando $E \rightarrow C$

$A^+ = ABECD$ usando $BC \rightarrow D$

- Quindi da A dipendono tutti gli attributi dello schema, ovvero **A è superchiave (e anche chiave)**

Passo 1: FD "semplici"

- Per minimizzare un insieme di FD è in primo luogo necessario riscrivere tutte le FD in una forma "standard", in cui **sulla destra compare sempre un singolo attributo**
- Supponiamo di avere $F = \{AB \rightarrow CD, AC \rightarrow DE\}$
- Allora si riscrive l'insieme F come:

$F = \{AB \rightarrow C, AB \rightarrow D, AC \rightarrow D, AC \rightarrow E\}$

Passo 2: attributi "estranei"

- In alcune FD è possibile che sul lato sinistro ci siano degli attributi inutili ("estranei"): **devono essere eliminati!**
- Supponiamo di avere $F = \{AB \rightarrow C, A \rightarrow B\}$ e calcoliamo A^+

$$A^+ = A$$

$$A^+ = AB \quad \text{poiché } A \rightarrow B \text{ e } A \subseteq A^+$$

$$A^+ = ABC \quad \text{poiché } AB \rightarrow C \text{ e } AB \subseteq A^+$$

- Quindi **C dipende solo da A**, ovvero in $AB \rightarrow C$ l'attributo **B è estraneo** (perché a sua volta dipendente da A) e possiamo riscrivere l'insieme di FD più semplicemente come: $F' = \{A \rightarrow C, A \rightarrow B\}$

Come facciamo a stabilire che in una FD del tipo $AX \rightarrow B$ l'attributo A è estraneo?

- Calcoliamo X^+ e verifichiamo se include B, ovvero se basta X a determinare B!

Normalizzazione di schemi relazionali



69

Passo 3: FD ridondanti

- Dopo avere eliminato gli attributi estranei si deve verificare se vi sono intere FD inutili ("ridondanti"), ovvero FD che sono implicate da altre
- Supponiamo di avere $F = \{B \rightarrow C, B \rightarrow A, C \rightarrow A\}$
- Si vede che **$B \rightarrow A$ è ridondante** in quanto bastano le altre due per stabilire che A dipende da B, e quindi possiamo riscrivere l'insieme di FD più semplicemente come: $F' = \{B \rightarrow C, C \rightarrow A\}$

Come facciamo a stabilire che una FD del tipo $X \rightarrow A$ è ridondante?

- La eliminiamo da F, calcoliamo X^+ e verifichiamo se include A, ovvero se con le FD che restano riusciamo ancora a dimostrare che X determina A!

NB: I PASSI 2 e 3 non possono essere invertiti!

- Sia $F = \{AB \rightarrow C, B \rightarrow A, C \rightarrow A\}$. Con il passo 2 scopriamo che A è estraneo in $AB \rightarrow C$, quindi otteniamo $F' = \{B \rightarrow C, B \rightarrow A, C \rightarrow A\}$ e dopo possiamo eliminare $B \rightarrow A$, restando con $F'' = \{B \rightarrow C, C \rightarrow A\}$
- Se invertiamo i passi 2 e 3 non riusciamo più a eliminare $B \rightarrow A$!

Normalizzazione di schemi relazionali



70

Creazione degli schemi in 3NF

- Avendo minimizzato l'insieme iniziale di FD si può procedere con la creazione degli schemi in 3NF

Passo 4:

Si raggruppano tutte le FD che hanno lo stesso **lato sinistro (determinante)** X e si crea uno schema che ha X come chiave.

Passo 5:

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

Passo 6:

Alla fine si verifica che esista uno schema la cui chiave è anche chiave dello schema originario. Se non esiste lo si crea, usando gli attributi che compaiono in tutti i determinanti ed eliminando quelli determinati da altri.



Esempio 1

Sia $F = \{A \rightarrow BC, B \rightarrow C, ABE \rightarrow D\}$ e $R(ABCDE)$

Passo 1: $F = \{A \rightarrow B, A \rightarrow C, B \rightarrow C, ABE \rightarrow D\}$

Passo 2: $F' = \{A \rightarrow B, A \rightarrow C, B \rightarrow C, AE \rightarrow D\}$ poiché B dipende da A

Passo 3: $F'' = \{A \rightarrow B, B \rightarrow C, AE \rightarrow D\}$ poiché $A \rightarrow C$ è ridondante

Passo 4: Si generano gli schemi $R1(\underline{AB})$, $R2(\underline{BC})$ e $R3(\underline{AED})$

Passi 5 e 6: Le chiusure delle chiavi sono:

$A^+ = ABC$

$B^+ = BC$

$AE^+ = ABCED$ che è quindi anche chiave dello schema non decomposto



Esempio 2

Sia $F = \{A \rightarrow BC, B \rightarrow AG, BE \rightarrow D\}$ e $R(ABCDEFG)$

Passo 1: $F = \{A \rightarrow B, A \rightarrow C, B \rightarrow A, B \rightarrow G, BE \rightarrow D\}$

Passo 2: $F' = F$

Passo 3: $F'' = F$

Passo 4: Si generano gli schemi $R1(\underline{ABC})$, $R2(\underline{BAG})$ e $R3(\underline{BED})$

Passi 5 e 6: Le chiusure delle chiavi sono:

$A^+ = ABCG$

$B^+ = BAGC$

$BE^+ = ABCEDG$ che è quindi anche chiave dello schema non decomposto

Tuttavia $A \rightarrow B$ e $B \rightarrow A$, quindi si fondono $R1$ e $R2$ in $R12(\underline{ABCG})$, con, ad es., A chiave primaria e B chiave alternata.

Esempio 3

Sia $F = \{A \rightarrow B, B \rightarrow C, D \rightarrow E\}$ e $R(ABCDE)$

Passo 1: $F = \{A \rightarrow B, B \rightarrow C, D \rightarrow E\}$

Passo 2: $F' = F$

Passo 3: $F'' = F$

Passo 4: Si generano gli schemi $R1(\underline{AB})$, $R2(\underline{BC})$ e $R3(\underline{DE})$

Passo 5: Non si fonde nulla

Passo 6: Nessun determinante è chiave dello schema non decomposto; si crea quindi lo schema

$R0(\underline{ABD})$

Poiché $A \rightarrow B$, si elimina B restando con $R0(\underline{AD})$.

Esempio 4

Sia $R(ABCDE)$ e $F = \{C \rightarrow AB, BC \rightarrow DE, D \rightarrow B\}$

Passo 1: $F = \{C \rightarrow A, C \rightarrow B, BC \rightarrow D, BC \rightarrow E, D \rightarrow B\}$

Passo 2: $F' = \{C \rightarrow A, C \rightarrow B, C \rightarrow D, C \rightarrow E, D \rightarrow B\}$ poiché B dipende da C

Passo 3: $F'' = \{C \rightarrow A, C \rightarrow D, C \rightarrow E, D \rightarrow B\}$ poiché $C \rightarrow B$ è ridondante

Passo 4: $R1(\underline{C}ADE), R2(\underline{D}B)$

Passo 5: nessuna fusione

Passo 6: $C^+ = ABCDE$

$D^+ = BD$

C è quindi anche chiave dello schema non decomposto.

Esempio 5

Sia $R(ABCDEFGH)$ e $F = \{ABC \rightarrow DEG, BD \rightarrow ACE, C \rightarrow BH, H \rightarrow BDE\}$

Passo 1: $F = \{ABC \rightarrow D, ABC \rightarrow E, ABC \rightarrow G, BD \rightarrow A, BD \rightarrow C, BD \rightarrow E, C \rightarrow B, C \rightarrow H, H \rightarrow B, H \rightarrow D, H \rightarrow E\}$

Passo 2: Conviene innanzitutto calcolare le chiusure dei lati sinistri delle FD:

$ABC^+ = ABCDEGH$

$BD^+ = BDACEHG$

$C^+ = CBHDEAG$

$H^+ = HBDEACG$

quindi:

$F' = \{C \rightarrow D, C \rightarrow E, C \rightarrow G, BD \rightarrow A, BD \rightarrow C, BD \rightarrow E, C \rightarrow B, C \rightarrow H,$

$H \rightarrow B, H \rightarrow D, H \rightarrow E\}$

Passo 3: $F'' = \{C \rightarrow G, BD \rightarrow A, BD \rightarrow C, C \rightarrow H, H \rightarrow B, H \rightarrow D, H \rightarrow E\}$

Passi 4,5,6: a questo punto si può osservare che C , BD e H sono tutte chiavi di R , quindi lo schema è in 3NF (e anche in BCNF!) e non deve essere decomposto.

Esempio 6

Sia $R(ABCDEFGH)$ e $F = \{AB \rightarrow CDE, CE \rightarrow AB, A \rightarrow G, G \rightarrow BD\}$

Passo 1: $F = \{AB \rightarrow C, AB \rightarrow D, AB \rightarrow E, CE \rightarrow A, CE \rightarrow B, A \rightarrow G, G \rightarrow B, G \rightarrow D\}$

Passo 2: Chiusure $AB^+ = ABCDEG$ $CE^+ = CEABGD$
 $A^+ = AGBDCE$ $G^+ = GBD$

quindi:

$F' = \{A \rightarrow C, A \rightarrow D, A \rightarrow E, CE \rightarrow A, CE \rightarrow B, A \rightarrow G, G \rightarrow B, G \rightarrow D\}$

Passo 3: $F'' = \{A \rightarrow C, A \rightarrow E, CE \rightarrow A, A \rightarrow G, G \rightarrow B, G \rightarrow D\}$

Passo 4: $R1(\underline{ACEG}), R2(\underline{CEA}), R3(\underline{GBD})$

Passo 5: si fondono $R1$ e $R2$ ottenendo $R12(\underline{ACEG})$, con CE chiave alternata

Passo 6: non esiste nessuno schema che include una chiave di R , in quanto in R c'è anche H ; si crea quindi lo schema $R0(\underline{AH})$: $AH^+ = ABCDEGH$.

Esempio 7

Sia $F = \{\text{Imp} \rightarrow \text{Stip}; \text{Prog} \rightarrow \text{Bilancio}\}$

Passo 1: $F = \{\text{Imp} \rightarrow \text{Stip}; \text{Prog} \rightarrow \text{Bilancio}\}$

Passo 2: $F' = F$

Passo 3: $F'' = F$

Passo 4: Si generano gli schemi:

$\text{IMP}(\underline{\text{Imp}}, \text{Stip})$ e $\text{PROG}(\underline{\text{Prog}}, \text{Bilancio})$.

Passo 5: Non si fonde nulla.

Passo 6: Né Imp né Prog sono chiavi dello schema non decomposto; si crea quindi lo schema:

$\text{PARTECIPAZIONI}(\underline{\text{Imp}}, \underline{\text{Prog}})$.