

Normalizzazione

Motivazione

- Lo schema logico di una base dati, rappresentato attraverso il modello relazionale, sia che sia stato derivato dalla traduzione di un precedente schema concettuale, sia che sia stato scritto ex-novo, può presentare talvolta delle anomalie o criticità in termini di prestazioni, integrità e manutenzione.
- Come caso limite si può considerare quello di una base dati rappresentata attraverso un'unica relazione (tabella).

Motivazione

- Tipicamente, questa rappresentazione è associata a:
 - Elevata ridondanza dei dati, con conseguente **inefficienza** dei tempi di risposta anche per interrogazioni semplici
 - **Complessità** delle operazioni di aggiornamento
 - Pericolo di **perdita** completa di informazioni in caso di cancellazioni di alcune righe della tabella
- Vediamo un esempio ...

3

Motivazione

- Si vogliono rappresentare le informazioni sugli ordini di alcuni prodotti (numero d'ordine, data dell'ordine, tipo di prodotto ordinato), nomi ed indirizzi degli acquirenti, crediti maturati dagli acquirenti, nomi dei venditori relativi a ciascun ordine effettuato.
- Tutte queste informazioni possono essere rappresentate in un'unica tabella...

4

Motivazione

Sales

product-name	order-no	cust-name	cust-addr	credit	date	sales-name
vacuum cleaner	1458	Dave Bachmann	Austin	6	5-5-92	Carl Bloch
computer	2730	Elena Huang	Mt.View	10	5-6-92	Ted Hanss
refrigerator	2460	Mike Stolarchuck	Ann Arbor	8	7-3-92	Dick Phillips
television	519	Peter Honeyman	Detroit	3	9-5-92	Fred Remley
radio	1986	Charles Antonelli	Chicago	7	9-18-92	R. Metz
CD player	1817	C.V. Ravishankar	Bombay	8	1-3-93	Paul Basile
vacuum cleaner	1865	Charles Antonelli	Chicago	7	4-18-93	Carl Bloch
vacuum cleaner	1885	Betsy Blower	Detroit	8	5-13-93	Carl Bloch
refrigerator	1943	Dave Bachmann	Austin	6	6-19-93	Dick Phillips
television	2315	Dave Bachmann	Austin	6	7-15-93	Fred Remley

Figure 5.1 Example single table database

- Tuttavia questa soluzione è caratterizzata da alcuni aspetti tutt'altro che vantaggiosi

5

Motivazione

- L'informazione su alcuni prodotti e compratori è memorizzata in forma ridondante.

Sales

product-name	order-no	cust-name	cust-addr	credit	date	sales-name
vacuum cleaner	1458	Dave Bachmann	Austin	6	5-5-92	Carl Bloch
computer	2730	Elena Huang	Mt.View	10	5-6-92	Ted Hanss
refrigerator	2460	Mike Stolarchuck	Ann Arbor	8	7-3-92	Dick Phillips
television	519	Peter Honeyman	Detroit	3	9-5-92	Fred Remley
radio	1986	Charles Antonelli	Chicago	7	9-18-92	R. Metz
CD player	1817	C.V. Ravishankar	Bombay	8	1-3-93	Paul Basile
vacuum cleaner	1865	Charles Antonelli	Chicago	7	4-18-93	Carl Bloch
vacuum cleaner	1885	Betsy Blower	Detroit	8	5-13-93	Carl Bloch
refrigerator	1943	Dave Bachmann	Austin	6	6-19-93	Dick Phillips
television	2315	Dave Bachmann	Austin	6	7-15-93	Fred Remley

Figure 5.1 Example single table database

Motivazione

- La ricerca dei compratori di un "CD player" richiede la scansione dell'intera tabella, dall'inizio alla fine.

Sales						
product-name	order-no	cust-name	cust-addr	credit	date	sales-name
vacuum cleaner	1458	Dave Bachmann	Austin	6	5-5-92	Carl Bloch
computer	2730	Elena Huang	Mt.View	10	5-6-92	Ted Hanss
refrigerator	2460	Mike Stolarchuck	Ann Arbor	8	7-3-92	Dick Phillips
television	519	Peter Honeyman	Detroit	3	9-5-92	Fred Remley
radio	1986	Charles Antonelli	Chicago	7	9-18-92	R. Metz
CD player	1817	C.V. Ravishankar	Bombay	8	1-3-93	Paul Basile
vacuum cleaner	1865	Charles Antonelli	Chicago	7	4-18-93	Carl Bloch
vacuum cleaner	1885	Betsy Blower	Detroit	8	5-13-93	Carl Bloch
refrigerator	1943	Dave Bachmann	Austin	6	6-19-93	Dick Phillips
television	2315	Dave Bachmann	Austin	6	7-15-93	Fred Remley

Figure 5.1 Example single table database

Motivazione

- L'aggiornamento dell'indirizzo di un compratore (o del numero di crediti che ha maturato) richiede una modifica di diverse righe della tabella.

Sales						
product-name	order-no	cust-name	cust-addr	credit	date	sales-name
vacuum cleaner	1458	Dave Bachmann	Austin	6	5-5-92	Carl Bloch
computer	2730	Elena Huang	Mt.View	10	5-6-92	Ted Hanss
refrigerator	2460	Mike Stolarchuck	Ann Arbor	8	7-3-92	Dick Phillips
television	519	Peter Honeyman	Detroit	3	9-5-92	Fred Remley
radio	1986	Charles Antonelli	Chicago	7	9-18-92	R. Metz
CD player	1817	C.V. Ravishankar	Bombay	8	1-3-93	Paul Basile
vacuum cleaner	1865	Charles Antonelli	Chicago	7	4-18-93	Carl Bloch
vacuum cleaner	1885	Betsy Blower	Detroit	8	5-13-93	Carl Bloch
refrigerator	1943	Dave Bachmann	Austin	6	6-19-93	Dick Phillips
television	2315	Dave Bachmann	Austin	6	7-15-93	Fred Remley

Figure 5.1 Example single table database

Motivazione

- Infine, la sola cancellazione di un ordine (per esempio l'acquisto di un computer da parte di *Elena Huang*) può portare alla perdita di tutte le informazioni sull'acquirente.

Sales						
product-name	order-no	cust-name	cust-addr	credit	date	sales-name
vacuum cleaner	1458	Dave Bachmann	Austin	6	5-5-92	Carl Bloch
computer	2730	Elena Huang	Mt.View	10	5-6-92	Ted Hanss
refrigerator	2460	Mike Stolarchuck	Ann Arbor	8	7-3-92	Dick Phillips
television	519	Peter Honeyman	Detroit	3	9-5-92	Fred Remley
radio	1986	Charles Antonelli	Chicago	7	9-18-92	R. Metz
CD player	1817	C.V. Ravishankar	Bombay	8	1-3-93	Paul Basile
vacuum cleaner	1865	Charles Antonelli	Chicago	7	4-18-93	Carl Bloch
vacuum cleaner	1885	Betsy Blower	Detroit	8	5-13-93	Carl Bloch
refrigerator	1943	Dave Bachmann	Austin	6	6-19-93	Dick Phillips
television	2315	Dave Bachmann	Austin	6	7-15-93	Fred Remley

Figure 5.1 Example single table database

Motivazione

- In tutti i casi come questo, per rendere la base dati sicura ed affidabile è necessario operare una suddivisione della tabella iniziale così da distribuire su più tabelle le informazioni in essa contenute.
- Per rappresentare diversi livelli di qualità di uno schema logico relativamente alle problematiche presentate, sono state introdotte le **forme normali**.

Normalizzazione

- Il processo attraverso il quale uno schema logico può essere trasformato per ottemperare ai requisiti delle forme normali è detto **normalizzazione**.
- Tipicamente, l'obiettivo minimo che ci si prefigge è quello di rendere una base dati in terza forma normale (3NF).
- Tuttavia, prima di arrivare alla terza forma normale è necessario passare per la prima e per la seconda...

11

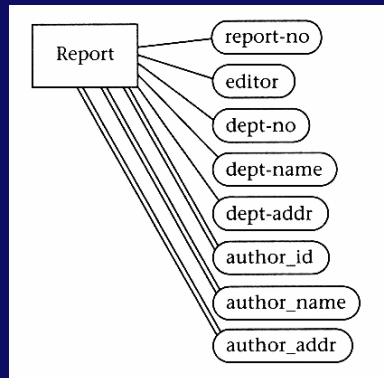
Prima forma normale

- **Def:** Una tabella si dice in **prima forma normale** (1NF) se e solo se ciascun attributo è definito su un dominio con valori atomici, vale a dire non c'è più di una colonna riferita ad uno stesso attributo (non ci sono attributi aggregati o attributi multivalore).
- Di fatto, la prima forma normale è considerata parte integrante della definizione formale di relazione nel modello relazionale

12

Prima forma normale

- Si consideri lo schema concettuale mostrato di seguito che vuole descrivere le informazioni associate ad un report creato da un numero di autori variabile da uno a tre.

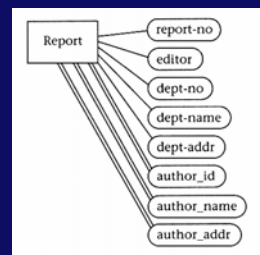


13

Prima forma normale

- Una possibile traduzione prevede l'uso di una tabella con il seguente schema:

REPORT (report-no, editor, dept-no, dept-name, dept-addr, author_id1, author_name1, author_addr1, author_id2, author_name2, author_addr2, author_id3, author_name3, author_addr3)



- In tale schema vi sono delle colonne che sono dedicate alla memorizzazione di un medesimo tipo di attributo: codice, nome ed indirizzo dell'autore.
- Pertanto, questa tabella non è in prima forma normale.

14

Prima forma normale

- Per portare la tabella in prima forma normale bisogna:
 - sostituire gli attributi aggregati con un unico attributo
 - riportare le informazioni sui diversi valori degli attributi in righe diverse della tabella.

Report

report_no	editor	dept_no	dept_name	dept_addr	author_id	author_name	author_addr
4216	woolf	15	design	argus 1	53	mantei	cs-tor
4216	woolf	15	design	argus 1	44	bolton	mathrev
4216	woolf	15	design	argus 1	71	koenig	mathrev
5789	koenig	27	analysis	argus 2	26	fry	folkstone
5789	koenig	27	analysis	argus 2	38	umar	prise
5789	koenig	27	analysis	argus 2	71	koenig	mathrev

Forme normali

- Le successive forme normali (seconda, terza, ...) mirano alla distribuzione su più relazioni delle informazioni da rappresentare nella base dati.
- Tale distribuzione ha come obiettivo quello di ridurre i legami funzionali presenti tra i diversi attributi di una relazione.
- Tali legami sono esplicitati attraverso il concetto di **dipendenza funzionale**

Dipendenze funzionali

- **Def:** Data una relazione $R(X)$ e due insiemi di attributi A e B non vuoti e sottoinsiemi di X , si dice che B è funzionalmente dipendente da A se il fatto che due generiche tuple abbiano valori uguali sull'insieme di attributi A implica che debbano avere valori uguali anche su B .
- Tale dipendenza viene rappresentata con la seguente simbologia: $A \rightarrow B$.

17

Dipendenze funzionali

- In altre parole, la dipendenza funzionale $A \rightarrow B$ richiede che se $t_1[A] = t_2[A]$ allora vale la relazione $t_1[B] = t_2[B]$.
- Da notare che può comunque succedere che due tuple abbiano valori diversi su A ed uguali su B

A	B	C
Gino	100	novembre
Gino	100	settembre
Carlo	150	giugno
Alfio	100	agosto
Gino	100	agosto
Manfredi	150	giugno

$A \rightarrow B$? **Sì**

$B \rightarrow A$? **No**

$A \rightarrow C$? **No**

$B \rightarrow C$? **No**

$C \rightarrow B$? **Sì**

18

Dipendenze funzionali

- Tuttavia, come nel caso del vincolo di chiave, la dipendenza funzionale esprime un **legame** logico che deve essere verificato **a livello di schema** piuttosto che di istanza
 - La dipendenza funzionale tra due attributi esprime un legame che è espressione di un vincolo valido nel contesto rappresentato dalla base dati.

19

Dipendenze funzionali

- Per la tabella sotto mostrata valgono le seguenti dipendenze funzionali:
 - report_no → editor, dept_no
 - dept_no → dept_name, dept_addr
 - author_id → author_name, author_addr

report_no	editor	dept_no	dept_name	dept_addr	author_id	author_name	author_addr
4216	woolf	15	design	argus 1	53	mantei	cs-tor
4216	woolf	15	design	argus 1	44	bolton	mathrev
4216	woolf	15	design	argus 1	71	koenig	mathrev
5789	koenig	27	analysis	argus 2	26	fry	folkstone
5789	koenig	27	analysis	argus 2	38	umar	prise
5789	koenig	27	analysis	argus 2	71	koenig	mathrev

Dipendenze funzionali

- Proprietà:
 - Riflessiva: se $X \supseteq Y \Rightarrow X \rightarrow Y$
 - Arricchimento: se $X \rightarrow Y \Rightarrow XZ \rightarrow YZ$
 - Transitiva: se $X \rightarrow Y$ e $Y \rightarrow Z \Rightarrow X \rightarrow Z$
 - Proiezione: se $X \rightarrow YZ \Rightarrow X \rightarrow Y$
 - Unione: se $X \rightarrow Y$ e $X \rightarrow Z \Rightarrow X \rightarrow YZ$
 - Pseudo-transitiva: se $X \rightarrow Y$ e $WY \rightarrow Z \Rightarrow WX \rightarrow Z$

21

Dipendenze funzionali

- **Def:** Data una relazione $R(X)$ e due insiemi di attributi A e B non vuoti e sottoinsiemi di X , la dipendenza funzionale $A \rightarrow B$ si dice **non banale** se $A \cap B = \emptyset$

22

Dipendenze funzionali

- Una dipendenza funzionale banale $A \rightarrow B$ può essere trasformata in una non banale se esiste un sottoinsieme non vuoto di attributi di B che non è incluso in A .
- Per esempio la dipendenza funzionale banale $AC \rightarrow BC$ può essere trasformata in $AC \rightarrow B$ che è non banale se $AC \cap B = \emptyset$
- **Attenzione:** sarebbe errato dire che se vale $AC \rightarrow BC$ allora vale $A \rightarrow B$

23

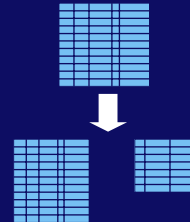
Dipendenze funzionali

- Per la definizione di chiave, data una relazione $R(X)$ con chiave K esiste una dipendenza funzionale tra K ed un qualunque insieme di attributi dello schema di R .
- In questo senso si è soliti dire che il **vincolo di dipendenza funzionale generalizza il vincolo di chiave**.
- In particolare, una dipendenza funzionale $A \rightarrow B$ su uno schema $R(X)$ degenera nel vincolo di chiave se $A \cup B = X$. In questo caso A è superchiave per $R(X)$ (si dimostra per assurdo).

24

Decomposizioni

- Il processo di normalizzazione di una base dati (e quindi di tutte le sue relazioni) si basa sulla **decomposizione delle relazioni** che non sono in forma normale.
- Il processo di decomposizione è un processo di distribuzione delle informazioni rappresentate in una relazione in due o più relazioni.
- Ai fini della normalizzazione si cerca di decomporre le relazioni in relazioni che soddisfano i vincoli delle forme normali.



25

Decomposizioni

- **Def:** Una **decomposizione** di una relazione $R(X)$ è un insieme di relazioni R_1, \dots, R_n ottenute proiettando $R(X)$ su un insieme di attributi A_1, \dots, A_n tali che $A_1 \cup \dots \cup A_n = X$.

26

Decomposizioni

- Data la relazione ...

Studente(matricola, nome, cognome, telefono)

- ...quali delle seguenti possono considerarsi delle decomposizioni di **Studente**?

R1(matricola, nome), R2(cognome, telefono) **Si**

R1(matricola, nome, cognome), R2(cognome, telefono) **Si**

R1(matricola), R2(cognome, telefono) **No**

27

Decomposizioni

- Tuttavia, non tutte le decomposizioni di una relazione $R(X)$ sono in grado di rappresentare in modo equivalente le informazioni contenute in $R(X)$:
 - Dipende dalla scelta degli attributi A_1, \dots, A_n su cui si effettua la decomposizione

28

Decomposizioni

- Per esempio, nella relazione sotto mostrata sono rappresentate le informazioni sui prestiti (loan-number, amount) concessi da alcune agenzie (branch-name, branch-city, assets) ad alcuni clienti (customer-name)

<i>branch-name</i>	<i>branch-city</i>	<i>assets</i>	<i>customer-name</i>	<i>loan-number</i>	<i>amount</i>
Downtown	Brooklyn	9000000	Jones	L-17	1000
Redwood	Palo Alto	2100000	Smith	L-23	2000
Perryridge	Brooklyn	1700000	Jackson	L-15	3000

29

Decomposizioni

- Immaginiamo di decomporre la relazione in due relazioni definite sullo schema R1(branch-name, branch-city, assets) ed R2(customer-name, loan-number, amount, branch-city)

<i>branch-name</i>	<i>branch-city</i>	<i>assets</i>	<i>customer-name</i>	<i>loan-number</i>	<i>amount</i>
Downtown	Brooklyn	9000000	Jones	L-17	1000
Redwood	Palo Alto	2100000	Smith	L-23	2000
Perryridge	Brooklyn	1700000	Jackson	L-15	3000

R1

<i>branch-name</i>	<i>branch-city</i>	<i>assets</i>
Downtown	Brooklyn	9000000
Redwood	Palo Alto	2100000
Perryridge	Brooklyn	1700000

R2

<i>customer-name</i>	<i>loan-number</i>	<i>amount</i>	<i>branch-city</i>
Jones	L-17	1000	Brooklyn
Smith	L-23	2000	Palo Alto
Jackson	L-15	3000	Brooklyn

Decomposizioni

- Si tratta di una decomposizione corretta dato che l'unione degli attributi delle tabelle decomposte è uguale all'insieme di attributi della tabella iniziale

<i>branch-name</i>	<i>branch-city</i>	<i>assets</i>	<i>customer-name</i>	<i>loan-number</i>	<i>amount</i>
Downtown	Brooklyn	9000000	Jones	L-17	1000
Redwood	Palo Alto	2100000	Smith	L-23	2000
Perryridge	Brooklyn	1700000	Jackson	L-15	3000

R1

<i>branch-name</i>	<i>branch-city</i>	<i>assets</i>
Downtown	Brooklyn	9000000
Redwood	Palo Alto	2100000
Perryridge	Brooklyn	1700000

R2

<i>customer-name</i>	<i>loan-number</i>	<i>amount</i>	<i>branch-city</i>
Jones	L-17	1000	Brooklyn
Smith	L-23	2000	Palo Alto
Jackson	L-15	3000	Brooklyn

Decomposizioni

- Tuttavia, è lecito chiedersi: L'informazione contenuta nelle tabelle decomposte è equivalente a quella contenuta nella tabella iniziale?

<i>branch-name</i>	<i>branch-city</i>	<i>assets</i>	<i>customer-name</i>	<i>loan-number</i>	<i>amount</i>
Downtown	Brooklyn	9000000	Jones	L-17	1000
Redwood	Palo Alto	2100000	Smith	L-23	2000
Perryridge	Brooklyn	1700000	Jackson	L-15	3000

R1

<i>branch-name</i>	<i>branch-city</i>	<i>assets</i>
Downtown	Brooklyn	9000000
Redwood	Palo Alto	2100000
Perryridge	Brooklyn	1700000

R2

<i>customer-name</i>	<i>loan-number</i>	<i>amount</i>	<i>branch-city</i>
Jones	L-17	1000	Brooklyn
Smith	L-23	2000	Palo Alto
Jackson	L-15	3000	Brooklyn

Decomposizioni

- Per ricostruire l'informazione contenuta nella tabella iniziale a partire dalle tabelle decomposte è necessario effettuare un JOIN tra le tabelle decomposte

<i>branch-name</i>	<i>branch-city</i>	<i>assets</i>	<i>customer-name</i>	<i>loan-number</i>	<i>amount</i>
Downtown	Brooklyn	9000000	Jones	L-17	1000
Redwood	Palo Alto	2100000	Smith	L-23	2000
Perryridge	Brooklyn	1700000	Jackson	L-15	3000

R1

<i>branch-name</i>	<i>branch-city</i>	<i>assets</i>
Downtown	Brooklyn	9000000
Redwood	Palo Alto	2100000
Perryridge	Brooklyn	1700000

R2

<i>customer-name</i>	<i>loan-number</i>	<i>amount</i>	<i>branch-city</i>
Jones	L-17	1000	Brooklyn
Smith	L-23	2000	Palo Alto
Jackson	L-15	3000	Brooklyn

Decomposizioni

- Tuttavia, la relazione che si ottiene effettuando il JOIN tra R1 ed R2 non è esattamente uguale alla relazione iniziale ma contiene delle tuple in più!

R1		
<i>branch-name</i>	<i>branch-city</i>	<i>assets</i>
Downtown	Brooklyn	9000000
Redwood	Palo Alto	2100000
Perryridge	Brooklyn	1700000

R2			
<i>customer-name</i>	<i>loan-number</i>	<i>amount</i>	<i>branch-city</i>
Jones	L-17	1000	Brooklyn
Smith	L-23	2000	Palo Alto
Jackson	L-15	3000	Brooklyn

R1 Join R2

<i>branch-name</i>	<i>branch-city</i>	<i>assets</i>	<i>customer-name</i>	<i>loan-number</i>	<i>amount</i>
Downtown	Brooklyn	9000000	Jones	L-17	1000
Redwood	Palo Alto	2100000	Smith	L-23	2000
Perryridge	Brooklyn	1700000	Jackson	L-15	3000
Downtown	Brooklyn	9000000	Jackson	L-15	3000
Perryridge	Brooklyn	1700000	Jones	L-17	1000

Decomposizioni

- In altri casi invece la relazione che si ottiene effettuando il Join è proprio la relazione originale.
- Per identificare queste ultime decomposizioni si introduce la definizione di **decomposizione lossless-join**.
- **Def:** una decomposizione di $R(X)$ in $R_1(A)$ ed $R_2(B)$ si dice lossless-join se $R(X) = R_1(A) \text{ JOIN } R_2(B)$
- Una decomposizione non lossless-join è detta lossy-join o semplicemente lossy (decomposizione con perdita).

35

Decomposizioni

- Le uniche decomposizioni utilizzabili per pervenire alla trasformazione di uno schema in una forma normale sono le decomposizioni senza perdita (lossless).
- E' quindi necessario un criterio per stabilire sulla base degli attributi su cui viene effettuata la decomposizione se questa sia o meno senza perdita.

36

Decomposizioni

- **Prop:** Data una relazione $R(X)$ siano A e B due sottoinsiemi di X tali che $A \cup B = X$ e sia $R_1(A)$ ed $R_2(B)$ una decomposizione di $R(X)$ su A, B . Sia inoltre $C = A \cap B$.
Condizione sufficiente affinché la decomposizione su A e B sia senza perdita è che C sia superchiave per almeno una delle due relazioni $R_1(A)$ ed $R_2(B)$.
- In realtà si può dimostrare che **condizione sufficiente** affinché la decomposizione sia lossless-join è che valga almeno una delle due seguenti dipendenze funzionali:
 $C \rightarrow A$
 $C \rightarrow B$

37

Decomposizioni

- La precedente proposizione descrive un **criterio di sufficienza** affinché la decomposizione sia senza perdita.
- Il criterio non è tuttavia necessario:
 - esistono decomposizioni senza perdita per le quali non è vera nessuna delle due dipendenze funzionali $C \rightarrow A$ e $C \rightarrow B$.

38

Conservazione delle dipendenze

- Un'ulteriore proprietà delle decomposizioni descrive come queste agiscono relativamente ai vincoli di dipendenza funzionale definiti sulla relazione di partenza.
- Intuitivamente, si può considerare che se su una relazione $R(X)$ vale la dipendenza funzionale $A \rightarrow B$ e la relazione viene decomposta su due insiemi tali che uno non contiene nessuno degli attributi presenti in A e l'altro non contiene nessuno degli attributi presenti in B , allora la decomposizione non consente di preservare la dipendenza funzionale.

39

Conservazione delle dipendenze

- **Def:** una decomposizione conserva le dipendenze dello schema originario se ciascuna delle dipendenze su questo definite coinvolgono attributi che compaiono tutti insieme in almeno uno degli schemi decomposti.

40

Seconda forma normale

- **Def:** Data una dipendenza funzionale $A \rightarrow B$, si dice che B **dipende completamente** da A se per ogni sottoinsieme $Y \subset A$ la dipendenza $\{A-Y\} \rightarrow B$ è falsa
- Data una dipendenza funzionale $A \rightarrow B$, si dice che B **dipende parzialmente** da A se non ne dipende completamente

41

Seconda forma normale

- **Def:** Data una relazione $R(X)$ ed A un generico sottoinsieme di X, A è detto **attributo primo** (prime attribute) se è sottoinsieme di una qualche chiave di $R(X)$

42

Seconda forma normale

- **Def:** Una tabella si dice in **seconda forma normale** (2NF) se e solo se è in 1NF e ciascun attributo non primo è completamente dipendente dalla chiave primaria.

43

Seconda forma normale

- Consideriamo la seguente tabella:
 - La chiave è (report_no, author_id)
 - Alcuni attributi dipendono solo da report_no:
report_no → editor, dept_no
dept_no → dept_name, dept_addr
 - alcuni attributi dipendono solo da author_id:
author_id → author_name, author_addr
- Quindi, la tabella non è in seconda forma normale.

report_no	editor	dept_no	dept_name	dept_addr	author_id	author_name	author_addr
4216	woolf	15	design	argus 1	53	mantei	cs-tor
4216	woolf	15	design	argus 1	44	bolton	mathrev
4216	woolf	15	design	argus 1	71	koenig	mathrev
5789	koenig	27	analysis	argus 2	26	fry	folkstone
5789	koenig	27	analysis	argus 2	38	umar	prise
5789	koenig	27	analysis	argus 2	71	koenig	mathrev

Seconda forma normale

- Se una relazione non è in seconda forma normale si possono presentare diversi inconvenienti.
- Con riferimento alla relazione sopra riportata, oltre alla duplicazione dei valori di report_no, editor, dept_no, dept_name, dept_addr per ogni autore del report, si possono avere le seguenti anomalie:
 - anomalia di aggiornamento (update anomaly).
 - anomalia di inserimento (insert anomaly).
 - anomalia di cancellazione (delete anomaly).

45

Seconda forma normale

- **Anomalia di aggiornamento** (update anomaly): Se cambia l'indirizzo di un autore è necessario aggiornare diverse righe della tabella.

report_no	editor	dept_no	dept_name	dept_addr	author_id	author_name	author_addr
4216	woolf	15	design	argus 1	53	mantei	cs-tor
4216	woolf	15	design	argus 1	44	bolton	mathrev
4216	woolf	15	design	argus 1	71	koenig	mathrev
5789	koenig	27	analysis	argus 2	26	fry	folkstone
5789	koenig	27	analysis	argus 2	38	umar	prise
5789	koenig	27	analysis	argus 2	71	koenig	mathrev

report_no→editor, dept_no
dept_no→dept_name, dept_addr
author_id→author_name, author_addr

46

Seconda forma normale

- **Anomalia di inserimento** (insert anomaly): Se si deve aggiungere un nuovo editor alla tabella, siccome il valore di report_no non può essere nullo (fa parte della chiave primaria), bisogna che questo abbia anche associato un report di cui è editore.
- Non è possibile aggiungere le informazioni di un editor che non sia ancora editore di un report.

report_no	editor	dept_no	dept_name	dept_addr	author_id	author_name	author_addr
4216	woolf	15	design	argus 1	53	mantei	cs-tor
4216	woolf	15	design	argus 1	44	bolton	mathrev
4216	woolf	15	design	argus 1	71	koenig	mathrev
5789	koenig	27	analysis	argus 2	26	fry	folkstone
5789	koenig	27	analysis	argus 2	38	umar	prise
5789	koenig	27	analysis	argus 2	71	koenig	mathrev

report_no→editor, dept_no
dept_no→dept_name, dept_addr
author_id→author_name, author_addr

47

Seconda forma normale

- **Anomalia di cancellazione** (delete anomaly): Se si vuole togliere l'informazione su un report (cancellando le righe ad esso relative) si ha l'effetto collaterale di rimuovere le informazioni sul nome ed indirizzo degli autori.
- Per gli autori di un solo report questo può portare alla totale e definitiva perdita delle informazioni ad essi relative.
- Delle tre anomalie questa è da considerarsi la peggiore in quanto la perdita di informazioni ad essa conseguente non è recuperabile.

report_no	editor	dept_no	dept_name	dept_addr	author_id	author_name	author_addr
4216	woolf	15	design	argus 1	53	mantei	cs-tor
4216	woolf	15	design	argus 1	44	bolton	mathrev
4216	woolf	15	design	argus 1	71	koenig	mathrev
5789	koenig	27	analysis	argus 2	26	fry	folkstone
5789	koenig	27	analysis	argus 2	38	umar	prise
5789	koenig	27	analysis	argus 2	71	koenig	mathrev

report_no→editor, dept_no
dept_no→dept_name, dept_addr
author_id→author_name, author_addr

48

Seconda forma normale

- Per superare questi inconvenienti bisogna decomporre la tabella in più tabelle che soddisfino le condizioni rappresentate nella seconda forma normale:
 - all'interno di ciascuna tabella la dipendenza dalla chiave primaria di ciascun attributo non chiave deve essere completa.
- La decomposizione dovrà essere senza perdita e dovrà preservare le dipendenze funzionali.

49

Seconda forma normale

- Nell'esempio considerato valgono le seguenti dipendenze funzionali:
 - report_no → editor, dept_no
 - dept_no → dept_name, dept_addr
 - author_id → author_name, author_addr
- I membri destri delle prime due dipendenze dipendono da report_no
- Il membro destro della terza dipendenza dipende da author_id

report_no	editor	dept_no	dept_name	dept_addr	author_id	author_name	author_addr
4216	woolf	15	design	argus 1	53	mantei	cs-tor
4216	woolf	15	design	argus 1	44	bolton	mathrev
4216	woolf	15	design	argus 1	71	koenig	mathrev
5789	koenig	27	analysis	argus 2	26	fry	folkstone
5789	koenig	27	analysis	argus 2	38	umar	prise
5789	koenig	27	analysis	argus 2	71	koenig	mathrev

Seconda forma normale

- Su quali insiemi di attributi conviene decomporre la tabella in modo che le tabelle finali rispettino la 2NF ?

Report							
report_no	editor	dept_no	dept_name	dept_addr	author_id	author_name	author_addr
4216	woolf	15	design	argus 1	53	mantei	cs-tor
4216	woolf	15	design	argus 1	44	bolton	mathrev
4216	woolf	15	design	argus 1	71	koenig	mathrev
5789	koenig	27	analysis	argus 2	26	fry	folkstone
5789	koenig	27	analysis	argus 2	38	umar	prise
5789	koenig	27	analysis	argus 2	71	koenig	mathrev

report_no → editor, dept_no
dept_no → dept_name, dept_addr
author_id → author_name, author_addr

51

Seconda forma normale

- Considerando solo i requisiti di 2NF si potrebbe decomporre su:
 1. report_no, editor, dept_no, dept_name, dept_addr
 2. author_id, author_name, author_addr
- Tuttavia,... in questo modo non c'è garanzia che la decomposizione sia senza perdita. Per esserlo bisogna che l'intersezione sia chiave ...

Report							
report_no	editor	dept_no	dept_name	dept_addr	author_id	author_name	author_addr
4216	woolf	15	design	argus 1	53	mantei	cs-tor
4216	woolf	15	design	argus 1	44	bolton	mathrev
4216	woolf	15	design	argus 1	71	koenig	mathrev
5789	koenig	27	analysis	argus 2	26	fry	folkstone
5789	koenig	27	analysis	argus 2	38	umar	prise
5789	koenig	27	analysis	argus 2	71	koenig	mathrev

report_no → editor, dept_no
dept_no → dept_name, dept_addr
author_id → author_name, author_addr

Seconda forma normale

- Aggiungiamo al secondo insieme di attributi la chiave della prima relazione:
 1. report_no, editor, dept_no, dept_name, dept_addr
 2. report_no, author_id, author_name, author_addr
- Tuttavia,...in questo modo il secondo insieme non rispetta la 2NF

report_no	editor	dept_no	dept_name	dept_addr	author_id	author_name	author_addr
4216	woolf	15	design	argus 1	53	mantei	cs-tor
4216	woolf	15	design	argus 1	44	bolton	mathrev
4216	woolf	15	design	argus 1	71	koenig	mathrev
5789	koenig	27	analysis	argus 2	26	fry	folkstone
5789	koenig	27	analysis	argus 2	38	umar	prise
5789	koenig	27	analysis	argus 2	71	koenig	mathrev

report_no→editor, dept_no
dept_no→dept_name, dept_addr
author_id→author_name, author_addr

Seconda forma normale

- Decomponendo il secondo insieme di attributi:
 1. report_no, editor, dept_no, dept_name, dept_addr
 2. author_id, author_name, author_addr
 3. report_no, author_id

report_no	editor	dept_no	dept_name	dept_addr	author_id	author_name	author_addr
4216	woolf	15	design	argus 1	53	mantei	cs-tor
4216	woolf	15	design	argus 1	44	bolton	mathrev
4216	woolf	15	design	argus 1	71	koenig	mathrev
5789	koenig	27	analysis	argus 2	26	fry	folkstone
5789	koenig	27	analysis	argus 2	38	umar	prise
5789	koenig	27	analysis	argus 2	71	koenig	mathrev

report_no→editor, dept_no
dept_no→dept_name, dept_addr
author_id→author_name, author_addr

Seconda forma normale

- La decomposizione della tabella originale in più tabelle che soddisfino la seconda forma normale necessita dell'introduzione di tre tabelle Report1 Report2 e Report3.

report_no	editor	dept_no	dept_name	dept_addr	author_id	author_name	author_addr
4216	woolf	15	design	argus 1	53	mantei	cs-tor
4216	woolf	15	design	argus 1	44	bolton	mathrev
4216	woolf	15	design	argus 1	71	koenig	mathrev
5789	koenig	27	analysis	argus 2	26	fry	folkstone
5789	koenig	27	analysis	argus 2	38	umar	prise
5789	koenig	27	analysis	argus 2	71	koenig	mathrev

Seconda forma normale

- La prima tabella (Report1):
 - Ha chiave primaria report_no
 - Conserva le informazioni su editor, dept_no, dept_name e dept_addr.
- La seconda tabella (Report2):
 - Ha chiave primaria author_id
 - Conserva le informazioni su author_name e author_addr.
- La terza tabella (Report3):
 - include gli attributi report_n e author_id in modo da memorizzare le associazioni tra ciascun report ed i relativi autori.

report_no → editor, dept_no
dept_no → dept_name, dept_addr
author_id → author_name,
author_addr

4216	woolf	15	design	argus 1	53	mantei	cs-tor
4216	woolf	15	design	argus 1	44	bolton	mathrev
4216	woolf	15	design	argus 1	71	koenig	mathrev
5789	koenig	27	analysis	argus 2	26	fry	folkstone
5789	koenig	27	analysis	argus 2	38	umar	prise
5789	koenig	27	analysis	argus 2	71	koenig	mathrev

Seconda forma normale

- Dipendenze funzionali:
- Report1:
report_no → editor, dept_no
dept_no → dept_name, dept_addr
- Report2:
author_id → author_name, author_addr
- Report3:
nessuna dipendenza funzionale:
l'insieme di tutti gli attributi costituisce
la chiave primaria

Report 1				
report_no	editor	dept_no	dept_name	dept_addr
4216	woolf	15	design	argus 1
5789	koenig	27	analysis	argus 2

Report 2		
author_id	author_name	author_addr
53	mantei	cs-tor
44	bolton	mathrev
71	koenig	mathrev
26	fry	folkstone
38	umar	prise
71	koenig	mathrev

Report 3	
report_no	author_id
4216	53
4216	44
4216	71
5789	26
5789	38
5789	71

Seconda forma normale

- Le tabelle in 2NF rappresentano un significativo miglioramento rispetto a quelle 1NF, tuttavia risentono ancora di alcune inefficienze.
- In particolare, la presenza di **dipendenze funzionali transitive** (ammesse dalla 2NF) è sintomatica del fatto che nella tabella vengono rappresentate due tipologie di informazioni diverse, una per ogni dipendenza funzionale.

Seconda forma normale

- Nella tabella Report1 è presente una dipendenza funzionale transitiva tra alcuni attributi e la chiave primaria: $report_no \rightarrow dept_no \rightarrow dept_name, dept_addr$
- A causa di questo, se si vuole rimuovere un report dall'archivio, bisogna rimuovere le informazioni ad esso associate dalle tabelle Report1 e Report3.
- Di conseguenza vengono rimosse anche le informazioni relative al dipartimento associato al report in questione.

report_no	editor	dept_no	dept_name	dept_addr
4216	woolf	15	design	argus 1
5789	koenig	27	analysis	argus 2

Terza forma normale

- **Def:** Una tabella si dice in **terza forma normale** (3NF) se e solo se è in 2NF e per ogni dipendenza funzionale non banale $X \rightarrow A$ tra due insiemi di attributi X ed A è vera una delle due condizioni seguenti:
 - X è una superchiave
 - A è prime attribute.

Terza forma normale

- Si può notare come la tabella Report1 precedentemente definita non rispetta la condizione di 3NF.
- Infatti, la dipendenza funzionale $dept_no \rightarrow dept_name, dept_addr$ non verifica nessuna delle due condizioni previste per la 3NF:
 - $dept_no$ NON E' una superchiave
 - $(dept_name, dept_addr)$ NON E' parte della chiave

report_no	editor	dept_no	dept_name	dept_addr
4216	woolf	15	design	argus 1
5789	koenig	27	analysis	argus 2

61

Terza forma normale

- Tale tabella può essere decomposta in più tabelle che soddisfino alla terza forma normale.
- La decomposizione prevede l'impiego di due tabelle atte a memorizzare separatamente da una parte le informazioni su report e codice dipartimento ($dept_no$), dall'altra le informazioni sul dipartimento ($dept_name, dept_addr$).

Report 11			Report 12		
report_no	editor	dept_no	dept_no	dept_name	dept_addr
4216	woolf	15	15	design	argus 1
5789	koenig	27	27	analysis	argus 2

Report 2			Report 3	
author_id	author_name	author_addr	report_no	author_id
53	mantei	cs-tor	4216	53
44	bolton	mathrev	4216	44
71	koenig	mathrev	4216	71
26	fry	folkstone	5789	26
38	umar	prise	5789	38
71	koenig	mathrev	5789	71

Terza forma normale

- La terza forma normale elimina la maggior parte delle anomalie oggi conosciute per le basi dati e rappresenta un livello qualitativo soddisfacente nella maggior parte dei casi.
- In ogni caso, le rimanenti anomalie possono essere rimosse attraverso la trasformazione delle tabelle nelle forme normali successive alla terza.
- La prima tra queste è la forma normale di Boyce-Codd che è considerata una forte variazione rispetto alla 3NF.

63

Forma normale di Boyce-Codd

- **Def:** Una tabella si dice in **forma normale di Boyce-Codd** (BCNF) se e solo se per ogni dipendenza funzionale non banale $X \rightarrow A$, X è una superchiave.

64

Forma normale di Boyce-Codd

- Questa forma normale rappresenta una ulteriore restrizione rispetto alla 3NF in quanto non prevede l'esistenza di dipendenze funzionali in cui il termine destro sia semplicemente membro di una chiave:
 - Tutte le dipendenze funzionali devono avere come membro sinistro una superchiave.
- Se una relazione è in BCNF ogni campo di ciascuna tupla registra un'informazione che **NON PUO'** essere dedotta dai valori degli altri campi non chiave delle tuple della relazione

65

Forma normale di Boyce-Codd

- Se una tabella verifica la forma normale di Boyce-Codd questa verifica anche le terza, seconda forma normale.
- Invece non è detto che una tabella che verifichi la 3NF debba anche rispettare la Boyce-Codd.
- In questi casi, si possono avere anomalie di cancellazione.
- Vediamo un esempio...

66

Forma normale di Boyce-Codd

- Rule1: In un gruppo ogni impiegato è diretto da un solo leader, anche se ci possono essere più di un leader per uno stesso gruppo:
 - NomeImpiegato, NomeGruppo → NomeLeader
- Rule2: Ogni leader dirige solo un gruppo:
 - NomeLeader → NomeGruppo

Gruppo		
NomeImpiegato	NomeGruppo	NomeLeader
Sutton	Hawks	Wei
Sutton	Condors	Bachmann
Niven	Hawks	Wei
Niven	Eagles	Makowski
Wilson	Eagles	DeSmith

67

Forma normale di Boyce-Codd

- Sia (NomeImpiegato, NomeGruppo) che (NomeImpiegato, NomeLeader) sono chiavi della relazione.
- A causa della regola [NomeLeader → NomeGruppo] la relazione:
 - È in terza forma normale: NomeGruppo è sottoinsieme di una chiave
 - Non è in forma normale di Boyce-Codd: NomeLeader non è una superchiave.

68

Forma normale di Boyce-Codd

- Inoltre, a causa della prima dipendenza funzionale [NomeImpiegato, NomeGruppo → NomeLeader], non è possibile definire nessuna decomposizione che conservi le dipendenze.
- Infatti tale dipendenza include tutti gli attributi della relazione iniziale e quindi non potrebbe essere conservata da alcuna decomposizione definita su un sottoinsieme proprio degli attributi della relazione iniziale.

69

Forma normale di Boyce-Codd

- Questo significa che la relazione di partenza non può essere trasformata in un insieme di relazioni che soddisfino la forma normale di Boyce-Codd.
- In generale:
 - è garantita la raggiungibilità della terza forma normale,
 - non è affatto garantita la raggiungibilità della forma normale di Boyce-Codd.

70

Forma normale di Boyce-Codd

- La non raggiungibilità di BoyceCodd è associata ad una anomalia di cancellazione: Se l'impiegato Sutton lascia il gruppo Condors, viene completamente persa traccia del fatto che Bachmann sia il leader del gruppo Condors.
- In questo caso, per eliminare l'anomalia bisogna creare due tabelle, una per ogni dipendenza funzionale.
- Questo equivale in realtà ad aggiungere un'altra tabella Leader(NomeLeader, NomeGruppo).

Gruppo		
NomeImpiegato	NomeGruppo	NomeLeader
Sutton	Hawks	Wei
Sutton	Condors	Bachmann
Niven	Hawks	Wei
Niven	Eagles	Makowski
Wilson	Eagles	DeSmith

71

Forma normale di Boyce-Codd

- Nel caso in questione la tabella Gruppo viene "decomposta" nella tabella stessa ed un'altra tabella (Leader).
- Questo porta all'introduzione di un'informazione ridondante ma con conseguente eliminazione dell'anomalia di cancellazione.
- Tale soluzione è in generale da preferirsi rispetto alla soluzione che non crea ridondanza ma lascia l'anomalia di cancellazione.

72

Forma normale di Boyce-Codd

- Ulteriore esempio: la seguente tabella...

Insegnamento(studente, corso, docente)

...memorizza le informazioni sui corsi seguiti dagli studenti e sui relativi docenti

- E' noto che:
 - Ciascun docente tiene un solo corso:
docente → corso
 - Un corso può essere tenuto da più docenti, ma per ogni studente, ciascun corso ha un solo docente:
studente, corso → docente
- Una chiave della relazione Insegnamento è (studente, corso)

73

Forma normale di Boyce-Codd

- Anche in questo caso, la tabella è in 3NF ma non in BCNF
- La sua decomposizione in BCNF richiede o di rinunciare a preservare la dipendenza funzionale $\text{studente, corso} \rightarrow \text{docente}$ o di introdurre una ridondanza

Insegnamento(studente, corso, docente)

docente → corso

studente, corso → docente

74

Forma normale di Boyce-Codd

- Nel caso si decida di procedere operando la decomposizione e non preservando la dipendenza funzionale si dovrebbe decomporre la relazione **Insegnamento**(studente, corso, docente) in

R1(docente, corso)

R2(studente, docente)

docente → corso

studente, corso → docente

75

Forme normali

- Esistono anche forme normali superiori alla BCNF, in particolare la 4NF e 5NF.

76

Verifiche di normalizzazione sul modello concettuale

- Le idee alla base della normalizzazione possono essere usate per effettuare verifiche di qualità di uno schema nel modello Entity-Relationship.
- In sostanza si procede considerando che ogni associazione o entità venga tradotta in una tabella e si verifica se ogni attributo dipenda funzionalmente dall'identificatore.