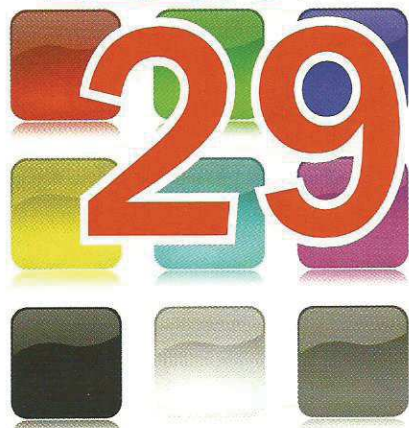




# La seconda forma normale (2FN)

## LEZIONE



Posto che i dati devono essere tutti semplici, in questo secondo livello normativo si va ad affrontare il vero e proprio aspetto logico della struttura. Partiamo da qualche concetto.

Sia R una relazione che contiene almeno due attributi X e Y. Se il valore di Y varia al variare del valore di X, si dice che Y ha una **dipendenza funzionale** da X e si indica con  $X \rightarrow Y$  (si dice anche che X determina Y, o che X è un determinante per Y). Lo stesso vale se, invece di singoli attributi, si considerano insiemi di attributi.

Ciò significa che, per ogni insieme di ennuple che possono esistere in R, non possono esistere due ennuple che hanno lo stesso valore di X e valori diversi di Y. Questo concetto è molto importante. Se il valore di un campo può variare al variare di qualsiasi parte della chiave primaria, possiamo dire che quel campo *dipende dall'intera chiave*, mentre se può variare alla variazione di *una sola parte della chiave*, allora quel campo non dipende dalla chiave intera, ma solo da una parte di essa.



Dalla definizione si evidenzia chiaramente che una chiave primaria è determinante per ogni attributo della relazione.

Diciamo, dunque, che:

Una relazione R è in **seconda forma normale (2FN)** se è in prima forma normale e ogni attributo non chiave dipende funzionalmente e completamente (cioè non parzialmente) dalla chiave primaria.

Tutti i campi diversi dalla chiave primaria, pertanto, devono dipendere dall'intera chiave primaria e non da una sua parte.

Prendiamo in considerazione il seguente schema di relazione:

**ORDINI**(CodOrdine, CodCliente, CodProdotto, DataOrdine, Quantità, PrezzoUnit, Descrizione)

Questa relazione non è in seconda forma normale perché al suo interno sono memorizzate informazioni riguardanti più di un oggetto, cioè più elementi distinti: gli ordini, i prodotti, i prodotti ordinati in ciascun ordine. Supponiamo che le informazioni sugli ordini e sui prodotti non siano presenti in altri schemi.

CodOrdine	CodCliente	CodProdotto	DataOrdine	Quantità	PrezzoUnit	Descrizione
1	C1	A023	15/12/04	25	€ 50,25	Cintura
1	C1	A789	15/12/04	120	€ 25,80	PortachiaviA
1	C1	A976	15/12/04	100	€ 22,14	PortachiaviB
2	C24	G324	15/12/04	45	€ 30,25	Portapatente
3	C56	A023	29/12/04	120	€ 50,25	Cintura

Per questi motivi lo schema presenta delle anomalie:

- **anomalia in inserimento**: non è possibile inserire un nuovo articolo in magazzino sino a quando non è ordinato (ciò perché la chiave primaria della relazione è *CodOrdine* e *CodProdotto* e non può mai essere nulla);
- **anomalia in cancellazione**: se si cancellano la seconda o la terza o la quarta tupla, si perdono informazioni sugli articoli che compaiono in esse;
- **anomalia in aggiornamento**: se varia il prezzo unitario di un articolo, occorre aggiornare tutte le t-uple in cui compare, con i relativi totali.

Verifichiamo le dipendenze funzionali dalla chiave primaria, composta dagli attributi *CodOrdine* e *CodProdotto*:

*CodOrdine* → *CodCliente*, *DataOrdine*  
*CodOrdine*, *CodProdotto* → *DataOrdine*, *Quantità*  
*CodProdotto* → *PrezzoUnit*, *Descrizione*

L'attributo *CodCliente* dipende solo da *CodOrdine* e non dal prodotto ordinato, così come *DataOrdine*. *Quantità*, invece, dipende sia dall'ordine che dal prodotto, infatti, nello stesso ordine ci sono articoli con quantità diverse; lo stesso articolo compare in ordini diversi con quantità differenti. Questo significa che, se si indica l'ordine di codice 1, non si ottiene una sola quantità, ma più d'una (25, 120, 100); analogamente, se si indica l'articolo A023 si ottengono le quantità 25 e 120. Se, invece, si indicano l'ordine 1 e l'articolo A023, si ottiene solo la quantità 25. *PrezzoUnit* e *Descrizione* dipendono solo da *CodProdotto*.

Per risolvere questi problemi, si deve scomporre la relazione in relazioni più semplici, ciascuna relativa a una data categoria: gli ordini, i prodotti e i prodotti ordinati. Le relazioni devono essere collegate tramite le chiavi primarie.

Il nuovo schema relazionale che soddisfa la seconda forma normale è, pertanto, il seguente:

**PRODOTTIORDINATI**(*CodOrdine*, *CodProdotto*, *Quantità*)

**PRODOTTI**(*CodProdotto*, *PrezzoUnit*, *Descrizione*)

**ORDINI**(*CodOrdine*, *DataOrdine*, *CodCliente*)

È facile verificare che ora le anomalie non sono più presenti e che, attraverso le chiavi primarie è possibile reperire tutte le informazioni.

In generale, quindi, si deve fare in modo che tutti gli attributi non chiave in uno schema di relazione dipendano funzionalmente dall'intera chiave primaria. Per normalizzare si procede in questo modo:

- nello schema originario rimangono la chiave primaria e tutti gli attributi non chiave, se ci sono, che dipendono completamente da essa;
- si crea un nuovo schema di relazione per ogni parte di chiave primaria da cui dipendono completamente altri attributi non chiave.

Per consolidare le conoscenze e applicare i precedenti punti, facciamo un esempio generico. Consideriamo lo schema di relazione:

**R**(A, B, C, D, E) con le seguenti dipendenze funzionali:

$A, B \rightarrow C$        $A \rightarrow D$        $B \rightarrow E$

Gli attributi D ed E non dipendono dall'intera chiave, pertanto la relazione non è in 2FN. Decomponiamo cominciando dalla dipendenza  $A \rightarrow D$ . Dobbiamo costruire due nuove relazioni:

- una prima relazione R1 che comprende tutti gli attributi presenti nella dipendenza funzionale che viola la 2FN. In questa relazione il determinante della dipendenza funzionale diventa, così, la chiave della nuova relazione;
- una seconda relazione R2 composta dagli attributi di R1 privati dell'attributo che dipende parzialmente dalla chiave.

Otteniamo, quindi:

R1(A, D)      R2(A, B, C, E)

Il processo non può terminare poiché è presente l'altra dipendenza funzionale  $B \rightarrow E$ . Applicando ancora il procedimento di decomposizione otteniamo:

R3(B, E)      R4(A, B, C)

Le relazioni R1, R3 e R4 rappresentano la decomposizione di R(A, B, C, D, E) in 2FN.



# La terza forma normale (3FN) e la forma normale di Boyce-Codd

## LEZIONE

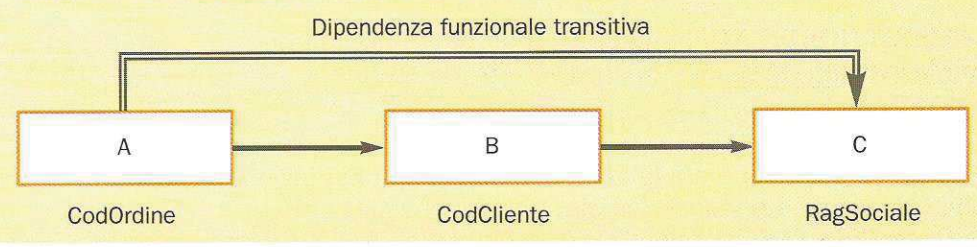
# 30

Una relazione R è in **terza forma normale (3FN)** se è in 2FN e ogni attributo non chiave dipende direttamente dalla chiave. In altri termini, la relazione R non deve possedere attributi non chiave che dipendano funzionalmente da altri attributi non chiave.

La 3FN elimina la *dipendenza transitiva* degli attributi dalla chiave. Diamo una definizione di dipendenza transitiva in modo da ben comprendere questa nuova forma normale:

Data una relazione con attributi A, B e C e con chiave primaria A:

- se C dipende funzionalmente dal determinante B, a sua volta funzionalmente dipendente da A;
  - se il determinante B non è una chiave candidata della relazione, alternativa ad A;
- allora C ha una **dipendenza funzionale transitiva** da A.



Riprendiamo lo schema di relazione visto nella lezione precedente e aggiungiamo alla relazione *ORDINI* la ragione sociale del cliente:

**ORDINI**(CodOrdine, CodCliente, RagSociale)

<u>CodOrdine</u>	CodCliente	RagSociale
1	C1	Rossi Accessori
2	C2	Verdi e figli
3	C1	Rossi Accessori
4	C3	Accessori per tutti
5	C2	Verdi e figli

Possiamo osservare che anche in questo caso ci sono delle anomalie:

- **anomalia in inserimento:** non è possibile inserire la ragione sociale relativa a un cliente sino a quando quest'ultimo non compare in un ordine (ciò perché la chiave primaria della relazione è *CodOrdine* e non può mai essere nulla);
- **anomalia in cancellazione:** se si cancella, ad esempio, la quarta t-upla si perde l'informazione che il cliente di codice C3 ha ragione sociale *Accessori per tutti*;
- **anomalia in aggiornamento:** se varia la ragione sociale di un certo cliente, occorre aggiornare tutte le t-uple interessate.

Questi problemi sono dovuti al fatto che la ragione sociale, in effetti, è indipendente dal codice dell'ordine e dipende solo dal codice del cliente:

CodCliente → RagSociale

Siamo quindi di fronte al caso in cui un attributo non chiave (*RagSociale*) dipende da un altro attributo non chiave (*CodCliente*). Lo schema deve essere, pertanto, trasformato nel seguente:

**ORDINI**(CodOrdine, CodCliente)  
**CLIENTI**(CodCliente, RagSociale)

In generale si deve fare in modo che, in uno schema di relazione, tutti gli attributi non chiave dipendano direttamente (non transitivamente) dall'intera chiave primaria. Per normalizzare si procede in questo modo:

- nello schema originario rimangono la chiave primaria e tutti gli attributi non chiave che dipendono direttamente da essa;
- si crea un nuovo schema di relazione per ogni attributo da cui dipendono altri attributi non chiave.

Per consolidare le conoscenze e applicare i precedenti punti, facciamo un esempio generico. Consideriamo il seguente schema di relazione:

$R(\underline{A}, B, C, D, E)$

con le seguenti dipendenze funzionali:

$A \rightarrow B$        $A \rightarrow E$        $B \rightarrow C$        $B \rightarrow D$

Si evince facilmente che  $A \rightarrow C$  e  $A \rightarrow D$  transitivamente. Questa relazione, quindi, non è in 3FN: procediamo alla decomposizione. Consideriamo la dipendenza funzionale  $B \rightarrow C$ ; otteniamo le due seguenti relazioni:

$R1(\underline{B}, C)$        $R2(\underline{A}, B, D, E)$

Il processo non può terminare poiché è presente l'altra dipendenza funzionale  $B \rightarrow D$ . Applicando ancora il procedimento di decomposizione otteniamo:

$R3(\underline{B}, D)$        $R4(\underline{A}, B, E)$

Le relazioni  $R1$ ,  $R3$  e  $R4$  rappresentano la decomposizione di  $R(\underline{A}, B, C, D, E)$  in 3FN.



È ovvio che uno schema in seconda forma normale in cui c'è un solo attributo non chiave è sicuramente in terza forma normale. Un caso particolare è dato dagli attributi il cui valore è calcolato a partire dal valore di altri attributi. Ad esempio:

**PRODOTTIORDINATI**(CodOrdine, CodProdotto, Quantità, PrezzoTot)  
**PRODOTTI**(CodProdotto, Descrizione, PrezzoUnit)

L'attributo PrezzoTot dipende dal campo Quantità della relazione ProdottiOrdinati e dal campo PrezzoUnit della relazione Prodotti. Se venisse modificato il valore del campo Quantità o del campo PrezzoUnit, dovrebbe essere modificato anche il valore del rispettivo campo PrezzoTot. Il campo PrezzoTot, pertanto, dovrebbe essere eliminato dalla tabella e il totale calcolato in fase di esecuzione con opportune interrogazioni (che impareremo a conoscere con il termine query). A volte, però, prima di procedere in automatico alla cancellazione, è bene valutare la convenienza di tale operazione. In questo caso, ad esempio, una volta che l'ordine è stato fatto, la quantità non può più variare e il prezzo unitario deve essere fissato alla data dell'ordine. È utile e più efficiente, quindi, conservare il campo PrezzoTot, che viene calcolato solo al momento dell'emissione dell'ordine.

## La forma normale di Boyce-Codd

Generalmente, uno schema relazionale è normalizzato, quindi si arresta la decomposizione, una volta che le varie relazioni sono in 3FN. Si può tuttavia continuare a decomporre per portare lo schema nella forma normale di Boyce-Codd.

Una relazione  $R$  è in **forma normale di Boyce-Codd** (BCNF, Boyce-Codd Normal Form) quando rispetta le caratteristiche fondamentali del modello relazionale (1FN) e se, per ogni dipendenza funzionale  $X \rightarrow Y$  definita su di essa,  $X$  (cioè il determinante) è una chiave candidata. Dalla definizione emerge che, se in una relazione è definita la dipendenza funzionale  $X \rightarrow Y$ , allora l'insieme di attributi  $X$  deve contenere una chiave candidata (quindi può svolgere la funzione di chiave). La conseguenza è che una relazione che soddisfa la BCNF è anche in 2FN e in 3FN, poiché la BCNF non ammette che un determinante possa essere formato solo da una parte della chiave, come avviene per le violazioni alla 2FN, o che possa essere esterno alla chiave, come avviene per le violazioni alla 3FN. Una relazione che rispetta la forma normale di Boyce-Codd è, quindi, anche in 3FN, ma non è vero il contrario. Inoltre, se la chiave primaria della relazione è formata da un unico attributo ed è in 3FN, allora è anche in BCNF.