La Normalizzazione

Prof. Viglietti Francesco
www.in4matika.altervista.org

Definizione

La Normalizzazione è quel processo che serve a rimuovere gli errori di progettazione dalle basi di dati.
Nella normalizzazione viene presentato un insieme di forme normali, cioè insiemi di regole che descrivono cosa è permesso e cosa è vietato nella struttura delle tabelle.
Il processo di normalizzazione porta di solito a spezzare le tabelle in altre più piccole che costituiscono il progetto migliore.
Durante il processo di normalizzazione, il progetto passa da una forma all'altra, seguendo un ordine ben preciso. Generalmente, ciascuna forma implica la forma precedente.
Ad es. uno schema in terza forma normale deve anzitutto essere in seconda forma normale e così via. Ad ogni passo vengono aggiunte nuove regole che lo schema deve soddisfare.

Definizioni

Chiave (primaria) → insieme di attributi che identificano in modo univoco una tupla

Chiave candidata → insieme minimale di attributi che possono svolgere il ruolo di chiave. Possono esserci molte chiavi candidate, ma una solo chiave primaria.

Attributo non-chiave → campo che non fa parte della chiave primaria

Dipendenza funzionale → quando il valore di un insieme di attributi A determina un singolo valore dell'attributo B. Si dice che A è determinante per B. (A → B)

Dipendenza transitiva → quando un attributo A determina B e B determina C; si dice allora che C dipende transitivamente da A. (A → C transitivamente)

Forme normali

1) Criteri che definiscono le condizioni che devono essere soddisfatte per evitare situazioni anomale: **forme normali**

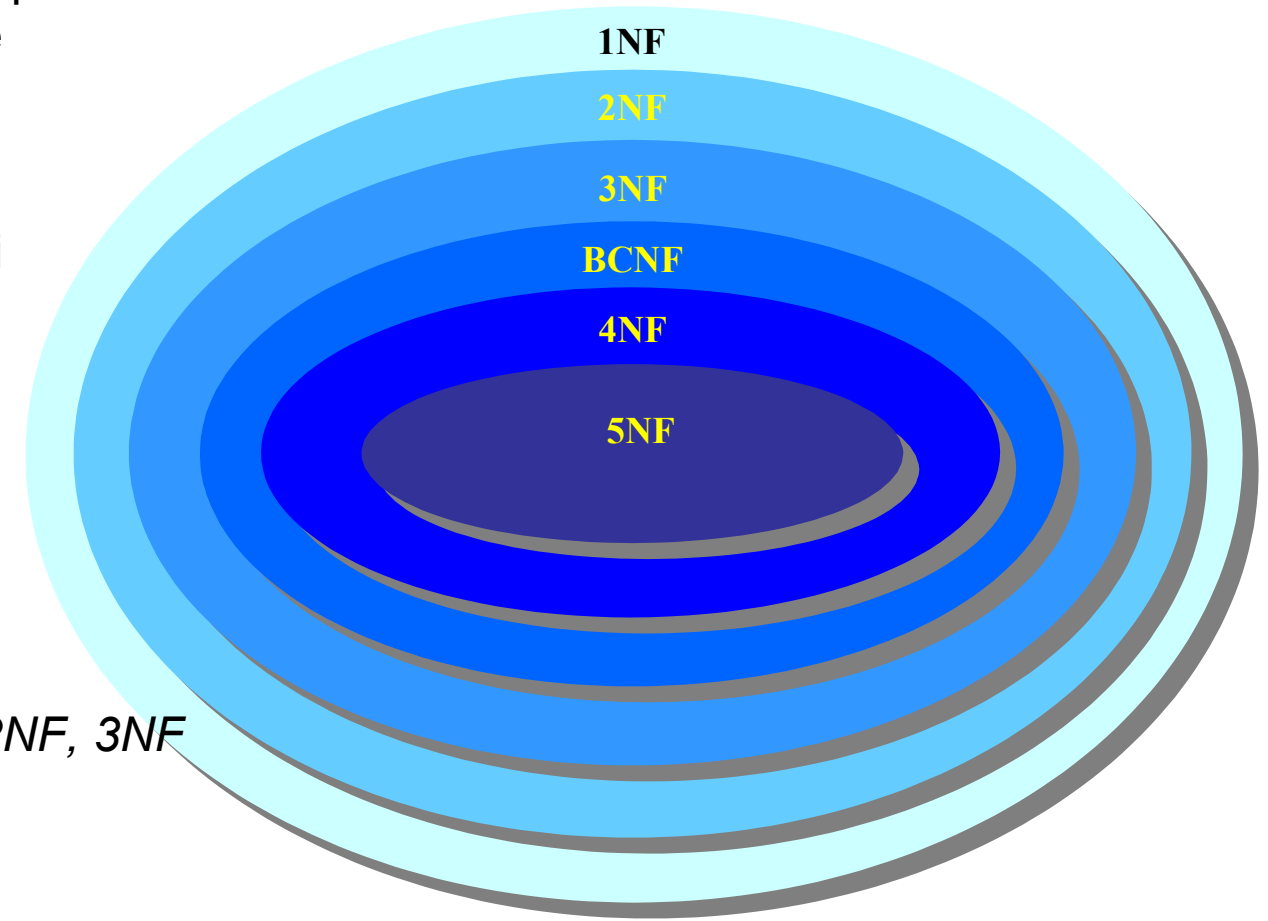
2) **1NF** richieste di base per il modello relazionale

3) **2NF**, **3NF, BCNF** anomalie da dipendenze funzionali

4) **4NF** anomalie per dipendenze multivalore

5) …

*** Ci interessano: 1NF, 2NF, 3NF*



Prima Forma Normale

La prima forma normale, detta anche 1NF, dice semplicemente che ciascuna colonna o attributo deve contenere soltanto valori atomici. Questo vuol dire che un attributo può contenere solo singoli valori (per esempio un numero o una stringa), non insiemi di valori o un'intera riga di una tabella.

Considerate ad esempio la tabella seguente (tab.1)

impiegato

IDimpiegato	nome	mansione	IDdipartimento	competenze
7513	Nora Edwards	Programmatore	128	C, Perl, Java
9842	Ben Smith	DBA	42	DB2
6651	Ajay Patel	Programmatore	128	VB, Java
9006	Candy Burnett	Amministratore di sistema	128	NT, Linux

Questo schema non è in prima forma normale perché la tabella contiene valori non atomici nella colonna delle competenze.

Prima Forma Normale

Nella tabella seguente (tab.2) tutti i valori sono atomici. Abbiamo creato una riga per ogni competenza, e lo schema risulta in prima forma normale

impiegato

IDimpiegato	nome	mansione	IDdipartimento	competenze
7513	Nora Edwards	Programmatore	128	C
7513	Nora Edwards	Programmatore	128	Perl
7513	Nora Edwards	Programmatore	128	Java
9842	Ben Smith	DBA	42	DB2
6651	Ajay Patel	Programmatore	128	VB
6651	Ajay Patel	Programmatore	128	Java
9006	Candy Burnett	Amministratore di sistema	128	NT
9006	Candy Burnett	Amministratore di sistema	128	Linux

Ovviamente, questa soluzione non è ottimale in quanto porta ad avere una grande ridondanza di dati visto che per ogni competenza di un impiegato occorre ripetere tutti i dati dell'impiegato stesso.

Prima Forma Normale

Una soluzione migliore, che rappresenta il modo più corretto per portare uno schema in prima forma normale, è la seguente (tab.3):

Impiegato

IDimpiegato	nome	mansione	IDdipartimento
7513	Nora Edwards	Programmatore	128
9842	Ben Smith	DBA	42
6651	Ajay Patel	Programmatore	128
9006	Candy Burnett	Amministratore di sistema	128

impiegComp

IDimpiegato	competenze
7513	C
7513	Perl
7513	Java
9842	DB2
6651	VB
6651	Java
9006	NT
9006	Linux

Abbiamo correttamente risolto il problema creando una seconda tabella.

Prima Forma Normale

In questo esempio sono state separate le competenze dalla tabella impiegato creando una nuova tabella che collega soltanto gli IDimpiegato alle relative competenze. Questo risolve i problemi di ridondanza. Ci si potrebbe chiedere come si arriva a questa soluzione. Esistono due possibili risposte. La prima è attraverso l'esperienza, la seconda è che se si prendesse la tabella rappresentata dalla tab.2 e si continuasse col processo di normalizzazione si arriverebbe comunque alle tabelle tab.3.

L'aiuto dell'esperienza permette di guardare avanti e arrivare subito a questo tipo di progettazione, ma anche procedere più sistematicamente nel processo di normalizzazione è un buon sistema.

Seconda Forma Normale

Dopo aver ottenuto un progetto in prima forma normale si passa alle forme superiori che sono un po' più complesse da comprendere. Principalmente sono introdotte per eliminare problemi durante le operazioni di aggiornamento o cancellazione.

Si dice che uno schema si trova in seconda forma normale (detta anche 2NF) se tutti gli attributi non chiave sono funzionalmente del tutto dipendenti dalla chiave primaria e lo schema si trova inoltre già in prima forma normale.

Che cosa significa? Vuol dire che ciascun attributo non chiave deve dipendere funzionalmente da tutta la chiave primaria, cioè se la chiave primaria è formata da più colonne, ogni altro attributo della tabella deve dipendere dalla combinazione di queste colonne.

Seconda Forma Normale

Chiariamoci le idee con un esempio.Consideriamo la tab.2 (con una riga per ciascuna competenza). Essa si trova in prima forma normale ma non in seconda. Perché no?

impiegato

IDimpiegato	nome	mansione	IDdipartimento	competenze
7513	Nora Edwards	Programmatore	128	C
7513	Nora Edwards	Programmatore	128	Perl
7513	Nora Edwards	Programmatore	128	Java
9842	Ben Smith	DBA	42	DB2
6651	Ajay Patel	Programmatore	128	VB
6651	Ajay Patel	Programmatore	128	Java
9006	Candy Burnett	Amministratore di sistema	128	NT
9006	Candy Burnett	Amministratore di sistema	128	Linux

Qual è la chiave primaria della tabella? La chiave primaria identifica in modo univoco la riga nella tabella. In questo caso l'unico modo di identificare una riga è quello di usare la combinazione di IDimpiegato e competenza.

Seconda Forma Normale

Con questo modo di inserire le competenze IDimpiegato non è più sufficiente per identificare la riga, infatti IDimpiegato, 7513, ad esempio, identifica tre righe. Comunque la combinazione di IDimpiegato e competenza è sufficiente per identificare una riga, quindi si prendono entrambi come chiave primaria ottenendo il seguente schema:

impiegato(IDimpiegato{PPK},nome,mansione, IDdipartimento, competenza{PPK})

Seconda Forma Normale

A questo punto dovete chiedervi: “quali dipendenze funzionali ci sono?”

IDimpiegato, competenza → nome, mansione, IDdipartimento

Ma anche

IDimpiegato → nome, mansione, IDdipartimento

In altre parole possiamo determinare il valore di nome, mansione e IDdipartimento dal solo valore di IDimpiegato.

Ciò significa che questi attributi dipendono funzionalmente dalla chiave primaria in modo parziale e non totale, cioè dipendono da un solo attributo della chiave primaria e non dalla chiave primaria intera. Di conseguenza questo schema non è in seconda forma normale.

Seconda Forma Normale

La domanda successiva è “come si arriva in seconda forma normale?”
Bisogna spezzare la tabella in più tabelle in modo che tutti gli attributi non chiave siano totalmente dipendenti dalla chiave. E abbastanza ovvio che nell’esempio si risolve suddividendo la tabella in due, e cioè:
impiegato (IDimpiegato{PK}, nome, mansione, Iddipartimento)
impiegatoCompetenze(IDimpiegato{PPK}, competenza{PPK})

Questo è lo schema delle tabelle tab.3
Come già detto, è in prima forma normale perché tutti i valori sono atomici; si trova inoltre in seconda forma normale perché tutti gli attributi non chiave di una tabella dipendono funzionalmente da tutta la chiave primaria.

Terza Forma Normale

Ogni tanto potrete sentire il detto “la normalizzazione riguarda la chiave, tutta la chiave, nient’altro che la chiave”.

La seconda forma normale ci dice che gli attributi devono dipendere da tutta la chiave. *La terza forma normale ci dice che gli attributi non-chiave devono dipendere direttamente dalla chiave.* Cioè non ci sono attributi non-chiave che dipendono da attributi non-chiave.

Da un punto di vista formale, per avere uno schema in terza forma normale (3NF) occorre rimuovere tutte le dipendenze transitive ed essere già in seconda forma normale. Che cos’è una dipendenza transitiva?

Riferiamoci alla tabella **impiegDip** seguente

IDimpiegato	nome	mansione	IDdip	nomeDipartimento
7513	Nora Edwards	programmatore	128	ricerca e sviluppo
9842	Ben Smith	DBA	42	finanza
6651	Ajay Pattel	programmatore	128	ricerca e sviluppo
9006	Candy Burrell	Sistemista	128	ricerca e sviluppo

Lo schema è:

impiegDip(IDimpiegato{PK}, nome, mansione, IDdip, nomeDipartimento)

Terza Forma Normale

.

Lo schema contiene le seguenti dipendenze funzionali: ***IDimpiegato → nome, mansione, IDdip, nomeDipartimento; IDdip → nomeDipartimento***
La chiave primaria è IDimpiegato, e tutti gli attributi dipendono funzionalmente da lui: facile da verificare visto che la chiave è formata da un solo attributo!

Comunque si ha anche:

IDimpiegato → nomeDipartimento

IDimpiegato → IDdip

e

IDdip → nomeDipartimento

Notate anche che l'attributo IDdip non fa parte della chiave.

Questa relazione significa che la dipendenza funzionale

IDimpiegato → IDdip è una dipendenza transitiva. Essa infatti possiede uno stadio intermedio

(IDdip → nomeDipartimento).

Terza Forma Normale

Per arrivare alla 3FN si deve eliminare la dipendenza transitiva. Esattamente come fatto per le due forme precedenti, anche qui si deve dividere la tabella in più tabelle. Quindi nell'esempio, si divide lo schema in due tabelle, impiegato e dipartimento, in questo modo:

Impiegato (IDimpiegato{PK}, nome, mansione, IDdip{FK})

dipartimento (IDdip{PK}, nomeDipartimento) Adesso è in terza forma normale.

Un altro modo per descrivere la 3FN consiste nel dire che formalmente, se uno schema è in 3FN, per ogni dipendenza funzionale in ogni tabella dello schema si hanno due possibili situazioni:

1. ciò che si trova a sinistra della dipendenza è una superchiave (cioè una chiave non necessariamente minima)
2. ciò che si trova a destra della dipendenza (→) è parte di una chiave (primaria o superchiave) della tabella.

Il secondo caso non si verifica poi così spesso! La maggior parte delle volte le dipendenze funzionali si comportano come nel primo caso.

Riepilogando

1FN.

Una relazione si dice in prima forma normale quando rispetta i requisiti fondamentali del modello relazionale, in particolare ogni attributo è atomico, non ci sono righe uguali e non ci sono attributi ripetitivi.

2FN.

Una relazione si dice in seconda forma normale quando è in prima forma normale e non ci sono attributi non chiave che dipendono parzialmente dalla chiave.

3FN.

Una relazione si dice in terza forma normale quando è in seconda forma normale e non ci sono attributi non chiave che dipendono transitivamente dalla chiave.

Forma Normale di Boyce-Codd e superiori

L'ultima forma normale che qui viene esaminata brevemente è quella di Boyce-Codd, detta a volte BCNF. Si tratta di una variazione della terza forma normale.

Abbiamo appena visto i due possibili casi che si verificano nella terza forma normale. Una tabella in BCNF deve essere in terza forma normale e ricadere nel primo caso trattato; cioè quello che si trova a sinistra della dipendenza deve essere una superchiave.

Questo si verifica frequentemente, come nell'esempio. Qualora ci fosse ancora una dipendenza che viola questa regola bisognerebbe spezzare di nuovo la tabella che la contiene, come già fatto per la 1NF, la 2NF e la 3NF

Esistono forme normali di grado superiore (quarto, quinto, ...) ma sono più utili in applicazioni di tipo squisitamente accademico che in pratica. La terza forma normale (o quella di Boyce Codd) è più che sufficiente per evitare i problemi di ridondanza che si incontrano nella pratica.

Esercizio 1

La tabella **anagrafica** contiene informazioni sui dipendenti di un'azienda e ha il seguente schema:

anagrafica (matricola{PPK}, nome, cognome, nascita, indirizzo, codice_Dipartimento{PPK}, nome_Dipartimento, indirizzo_Dipartimento, stipendio)

Definire le dipendenze funzionali che sussistono sullo schema di anagrafica, individuare le eventuali violazioni alla seconda e terza forma normale e scomporre la tabella in due o più tabelle per eliminare le violazioni trovate.

Soluzione esercizio 1

***1FN** deve avere campi atomici → ok*

2FN tutti gli attributi devono dipendere dalla chiave primaria

3FN tutti gli attributi dipendono esclusivamente dalla chiave primaria

Come si può notare la tabella non è in **2FN**, in quanto vi sono attributi che dipendono parzialmente dalla chiave primaria. Infatti *nome_Dipartimento* e *indirizzo_Dipartimento* dipendono da *codice_Dipartimento che è PPK.*

Allora la soluzione per portare la tabella in 2FN è quella di suddividerla nelle seguenti due tabelle:

anagrafica (matricola{PK}, nome, cognome, nascita, indirizzo, *codice_Dipartimento{FK}*, stipendio)

dipartimenti (codice_Dipartimento{PK}, nome_Dipartimento, indirizzo_Dipartimento)

A questo punto le tabelle sono anche in 3FN!

Esercizio 2

La tabella **materiali** contiene informazioni sui materialimetallici di un magazzino e ha il seguente schema:

materiali (codice{PK}, descrizione, fornitore, prezzo, indirizzo_fornitore, quantità, posizione_magazzino, acciaio, nome_acciaio, resistenza_acciaio, nome_fornitore)

Definire le dipendenze funzionali che sussistono sullo schema di materiali, individuare le eventuali violazioni alla seconda e terza forma normale e scomporre la tabella in due o più tabelle per eliminare le violazioni trovate.

Soluzione esercizio 2

1FN *deve avere campi atomici → ok*

2FN tutti gli attributi devono dipendere dalla chiave primaria → *ok*

3FN tutti gli attributi dipendono esclusivamente dalla chiave primaria

Come si può notare la tabella non è in **3FN**, in quanto vi sono attributi che dipendono da attributi non-chiave. Infatti *indirizzo_fornitore nome_fornitore,* dipendono da *fornitore* e *nome_acciaio, resistenza_acciaio* dipendono da *acciaio.*

Allora la soluzione per portare la tabella in 3FN è quella di suddividerla nelle seguenti tre tabelle:

materiali (codice{PK}, descrizione, *ID_fornitore{FK}*, prezzo, quantità, posizione_magazzino, *ID_acciaio{FK}*)

fornitori (ID_fornitore{PK}, nome_fornitore, indirizzo_fornitore)

acciai (ID_acciaio{PK}, nome_acciaio, resistenza_acciaio)

L'integrità referenziale

È un insieme di regole del modello relazionale che garantiscono l'integrità dei dati quando si hanno relazioni associate tra loro attraverso la FK: queste regole validano le associazioni tra tabelle e ne eliminano gli errori.
L'integrità referenziale viene rispettata quando per ogni valore non nullo della chiave esterna, esiste un valore corrispondente della chiave primaria nella tabella associata.
Vediamo ad esempio le seguenti tabelle:

L'integrità referenziale

Studenti

Matricola	Cognome	Nome	DataNasc	*Codice*
545	Rossi	Maria	NULL	125
653	Neri	Anna	20-set-1994	125
768	Verdi	Giuseppe	30-ott-1996	**NULL**
654	Rossi	Franco	**31-ott-1994**	180
314	Bruni	Enrico	27-ott-1995	**185**

Scuole

Codice	NomeScuola
125	ITC Manzoni
180	Liceo Dante
190	Liceo Fermi
185	ITIS Galvani

1) Non si deve poter **inserire** una riga in **Studenti** con valore di *Codice* che non compare fra quelli di *Codice* in **Scuole**
2) Non è possibile **cancellare** una scuola dalla tabella **Scuole** se ci sono righe nella tabella **Studenti** che si riferiscono ad essa
3) Non si possono **modificare** i valori di *Codice di Scuole* o di *Codice di Studenti* se sono violate le regole di integrità referenziale