LA NORMALIZZAZIONE

Introduzione

La normalizzazione e' una tecnica di progettazione dei database, mediante la quale si elimina la rindondanza dei dati al fine di evitare anomalie nella loro consistenza in seguito a operazioni di inserimento, cancellazione o modifica.
La normalizzazione viene eseguita in varie fasi.
Al termine di ciascuna fase il database si trova in uno degli stati di normalizzazione, o come si dice di solito, e' in una delle "forme normali".

Quindi una forma normale è una proprietà di uno schema relazionale che ne garantisce la "qualità".

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 ER

L’attività che permette di trasformare schemi non normalizzati in schemi che soddisfano una forma normale è detta normalizzazione.

La normalizzazione va utilizzata come tecnica di verifica dei risultati della progettazione di una base di dati, non costituisce quindi una metodologia di progettazione.

Ridondanza e coerenza dei dati

Es: Sia la tabella Software

Matricola	Prodotto	Costo
2346	Gestione Clienti	1.000.000
2145	Gestione Bilancio	5.000.000
2128	Fax	1.000.000
2367	Inventario	3.500.000

dove il campo Matricola è chiave per la tabella.

I campi chiave permettono non solo di ritrovare facilmente dati all'interno delle tabelle, ma soprattutto di collegare tra loro tabelle diverse.
La necessità di scrivere informazioni su più tabelle collegate nasce da due esigenze fondamentali: la prima riguarda lo spreco di spazio all'interno di una tabella (**ridondanza delle informazioni**), la seconda riguarda la **coerenza** tra le informazioni ridondanti.

Es: Sia la tabella Elezioni:

Lista	Candidato	Preferenze
Tutti per uno e uno per tutti	Aramis	5
Siamo forti	Superman	7
Tutti per uno e uno per tutti	Athos	8
Tutti per uno e uno per tutti	Portos	2
Ce la possiamo fare	Paperino	1
Siamo forti	Mazinga Z	7

Si vede subito che la prima colonna contiene informazioni ripetute, ed anche se questo spreco ci sembra di poca importanza, consideriamo che la colonna Lista contiene 129 caratteri, mentre la seconda 42 e la terza 6. Quindi su un totale di 129+42+6=177 caratteri circa il 73% è occupato dalla prima colonna (questi calcoli non hanno nessuno scopo formale, ma servono solo per spiegare il concetto che è alla base dell'esempio).
Se invece avessimo due tabelle:

1 - *Lista* in cui il campo *Num_lista* è chiave per la tabella

Num_lista	Nome Lista
1	Tutti per uno e uno per tutti
2	Siamo forti
3	Ce la possiamo fare

2 - *Preferenze* in cui il campo *Candidato* è chiave per la tabella

Num_lista	Candidato	Preferenze
1	Aramis	5
2	Superman	7
1	Athos	8
1	Portos	2
2	Paperino	1
2	Mazinga Z	7

Le informazioni occupate dalla prima colonna scenderebbero a 6 caratteri, quindi sul totale di 6+42+6=54 caratteri il campo Num_lista occupa l' 11% dello spazio occupato dalla tabella. (In queste righe non si è tenuto conto della nuova tabella Lista come influisce lo spazio occupato da questa tabella? E' sempre conveniente spezzare?)
Ovviamente abbiamo fatto un esempio in cui la tabella Preferenze è molto piccola, e quindi correggendo alcuni conti il risparmio potrebbe non essere evidente, ma si pensi che una base dati di un certo interesse ha almeno alcune migliaia di righe per ogni tabella. Il Campo *Num_lista* della tabella *Preferenze* viene detto *Chiave Esterna* per tale tabella, perché è il campo che permette di *collegare* le informazioni qui contenute con quelle contenute nella tabella Liste.

Un'ultima considerazione: supponiamo che il nome della lista numero 2 invece di "Siamo forti" sia "Siamo molto forti", e che questa correzione debba essere fatta a tabella Preferenze già riempita: nel caso a tabella unica le righe da correggere sono due, mentre nel caso a tabelle separate bisogna solo correggere la tabella Liste, senza dover toccare la tabella Preferenze, cosa che ha un numero ampio di aspetti postivi, tra cui uno dei più importanti è che non si possono creare situazioni "ibride" in cui qualche record è stato aggiornato e altri no.

- Se un attributo ridondante varia, è necessario modificare il valore in diverse tuple: **anomalia di aggiornamento**
- Un attributo ridondante deve essere cancellato da tutte le tuple: **anomalia di cancellazione**
- **anomalie di inserimento** potrebbero verificarsi nell'inserimento di dati in tuple con attributi ridondanti.

Relazioni con anomalie

- 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**

Impiegato	**Stipendio**	**Progetto**	**Bilancio**	**Funzione**
Rossi	20	Marte	2	Tecnico
Verdi	35	Giove	15	Progettista
Verdi	35	Venere	15	Direttore
Neri	55	Venere	15	Consulente
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

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

Dipendenza funzionale

Per formalizzare i concetti che verranno trattati viene introdotto un nuovo tipo di vincolo,
la **dipendenza funzionale**

Consideriamo:

Un'istanza r di uno schema R(X), due sottoinsiemi (non vuoti) di attributi Y e Z di X
diciamo che in r vale la dipendenza funzionale (Fd) Y -> Z (Y determina funzionalmente Z) se:

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

Y -> A è non banale se A non appartiene a Y;
Y -> Z è non banale se nessun attributo in Z appartiene a Y.

1ª Forma Normale (1FN)

Una tabella si dice posta in prima forma normale se tutti gli attributi che vi compaiono sono semplici.

Esempio: *non è posta in 1NF una relazione contenente un attributo nel quale vengono memorizzati gli indirizzi composti da: via città e provincia.*

Una tabella inoltre è in prima forma normale se non contiene colonne ripetute, che vengono usate per descrivere proprietà dell'entità della tabella che possono comparire in numero variabile.

Esempio: *una tabella Persona, che contiene le colonne Figlio1, Figlio2 e Figlio3, le quali descrivono i figli della persona.*

E' evidente che, nel caso in cui la persona abbia meno di tre figli, alcune colonne non saranno utilizzate.

Inoltre sarà impossibile rappresentare una persona con piu' di tre figli.

Per portare tale tabella in forma normale occorre scomporla in due tabelle: una per la persona, senza le colonne dei figli; l'altra per i figli, contenente una chiave esterna verso la tabella delle persone, indicante la riga che rappresenta il genitore.

2ª Forma Normale (2FN)

Un tabella e' in seconda forma normale se e' in prima forma normale e se tutti i suoi attributi sono funzionalmente dipendenti dall'intera chiave primaria e non solo da una parte di essa.

Per portare una tabella in seconda forma normale bisogna:

- individuare per ogni attributo Y che dipende parzialmente dalla chiave il sottoinsieme degli attributi X della chiave da cui dipende;
- costruire una nuova tabella avente X come chiave primaria e Y come attributo;
- togliere Y dalla tabella originaria.

Esempio: nella seguente relazione

Codice	**Titolo**	**Voto**	**Matricola**
INF1	Informatica	7	0988
SIS1	Sistemi	7	0988
INF1	Informatica	8	0325
ITA1	Italiano	8	0546
SIS1	Sistemi	6	0325

l'attributo **Titolo** dipende solo da **codice** e non da **Matricola**
mentre **Voto** dipende da entrambi.

La normalizzazione 2NF darà origine alle seguenti due tabelle.

Codice	**Voto**	**Matricola**
INF1	7	0988
SIS1	7	0988
INF1	8	0325
ITA1	8	0546
SIS1	6	0325

Codice	**Titolo**
INF1	Informatica
SIS1	Sistemi
INF1	Informatica
ITA1	Italiano
SIS1	Sistemi

3ª Forma Normale (3FN)

Una tabella e' in terza forma normale se e' in seconda forma normale e tutti gli attributi che non fanno parte della chiave primaria dipendono esclusivamente dalla chiave stessa.

Ovvero nessuno dei suoi attributi e' transitivamente dipendente dalla chiave primaria.

Avere una dipendenza transitiva significa che una attributo dipende da un'altro attributo descrittore, il quale a sua volta dipende dalla chiave primaria.

Per portare uno schema in terza forma normale bisogna procedere in questo modo:

- individuare tutti gli attributi Z che dipendono funzionalmente da un sottoinsieme di attributi Y diversi dalla chiave primaria;
- costruire una nuova tabella, avente Y per chiave primaria e Z come attributi;
- togliere Z dalla tabella originaria.

Esempio: nella relazione seguente

Matricola	**Nome**	**Comune**	**CODICE ISTAT**
0988	Gianni	Palermo	G273
0989	Franco	Palermo	G273
0325	Isabella	Trapani	H456
0546	Nicoletta	Milano	I123
0328	Maurizio	Palermo	G273

*l'attributo **comune** dipende dall'attributo **codice istat** che non appartiene alla chiave primaria (**matricola**)*
lo schema normalizzato sarà:

Matricola	**Nome**	**CODICE ISTAT**
0988	Gianni	G273
0989	Franco	G273
0325	Isabella	H456
0546	Nicoletta	I123
0328	Maurizio	G273

Comune	**CODICE ISTAT**
Palermo	G273
Palermo	G273
Trapani	H456
Milano	I123
Palermo	G273

B.C.F.N (Forma Normale Boyce-Codd)

Per evitare le anomalie viste si introduce la:

Forma Normale di Boyce-Codd (BCNF)

Uno schema R(X) è in forma normale di Boyce e Codd se, per ogni dipendenza funzionale (non banale) Y ->Z definita su di esso, Y è una superchiave di R(X).

- Si noti che, come al solito, il vincolo si riferisce allo schema, in quanto dipende dalla semantica degli attributi
- Un'istanza può pertanto soddisfare "per caso" il vincolo, ma ciò non garantisce che lo schema sia normalizzato
- In altri termini, le FD non si "ricavano" dall'analisi dei dati, ma ragionando sugli attributi dello schema.