

Progettazione dei dati nel modello relazionale



ALBERTO BELUSSI

ANNO ACCADEMICO 2018-'19

Esempio di progetto dati



Il progetto dei dati nel modello relazionale potrebbe seguire il seguente approccio:

- Identificare gli attributi elementari da rappresentare
- Creare inizialmente un'unica relazione (tabella) che li contiene tutti
- Analizzare il risultato
- Rivedere e decomporre la tabella in caso di problemi

Esempio di progetto dati



Quali problemi genera la rappresentazione
di tutti gli attributi in un'unica
relazione (tabella)?

Esempio



Caso pratico:

- Informazioni sui proprietari di appartamenti:

- codice fiscale, cognome, nome, data di nascita, codice catastale, via, numero civico, subalterno, tipo

- Tabella unica:

PROPRIETA' (CodiceFiscale, Cognome, Nome, DataNas,
CodiceCatasto, Via, NumeroCivico,
Subalterno, Tipo)

Problemi della tabella unica



- Se in una relazione (tabella) si uniscono concetti disomogenei e con esistenza autonoma si presenta la seguente situazione:
 - Ogni tupla rappresenta solitamente un'istanza della associazione che lega i concetti autonomi
 - Per le istanze di informazione che sono coinvolte in più associazioni si produce una ripetizione inutile di valori in diverse tuple (RIDONDANZA)

Problemi legati alla presenza di ridondanza



- La presenza di ridondanza produce le seguenti anomalie:
 - ANOMALIA di **aggiornamento**: per aggiornare il valore di un attributo si è obbligati a modificare tale valore su più tuple della base di dati
 - ANOMALIA di **inserimento**: per inserire una nuova istanza di un concetto è necessario inserire valori al momento sconosciuti (sostituibili con valori NULLI) per gli attributi non disponibili
 - ANOMALIA di **cancellazione**: per cancellare un'istanza di un concetto è necessario cancellare valori ancora validi oppure inserire valori NULLI per gli attributi da cancellare

Progettazione dei dati



- Un buon progetto dei dati a livello logico non contiene ridondanza inutile
- Un metodo per eliminare la ridondanza è quello di decomporre la relazione (tabella) unica in più tabelle
- Tuttavia anche nella decomposizione è necessario seguire delle regole per non perdere informazione.

Progetto dei dati



Ad esempio, nel caso di tabella unica mostrata in precedenza quale decomposizione possiamo applicare per eliminare la ridondanza e non perdere informazione?

PROPRIETA'(CodiceFiscale, Cognome, Nome, DataNas,
CodiceCatasto, Via, NumeroCivico, Subalterno,
Tipo)

si può decomporre in

PROPRIETARIO (CodiceFiscale, Cognome, Nome, DataNas)

UNITA'_IMMOBILIARE (CodiceCatasto, Via, NumeroCivico,
Subalterno, Tipo)

Progetto dei dati



PROPRIETARIO (CodiceFiscale, Cognome, Nome, DataNas)

UNITA'_IMMOBILIARE (CodiceCatasto, Via, NumeroCivico,
Subalterno, Tipo)

Si noti che questa decomposizione:

- elimina qualsiasi ridondanza, ma
- non conserva l'informazione che descrive l'associazione tra proprietario e unità immobiliare.

Come è possibile conservare tale informazione senza ritornare nella condizione di ridondanza che si produce nella tabella unica?

Progetto dei dati



- Per conservare il legame logico è necessario replicare parte degli attributi, scegliendo tra questi quelli che hanno la proprietà di identificare il concetto verso il quale si vuole generare il legame.
- Nell'esempio precedente possiamo scegliere il codice fiscale come identificatore del proprietario e il codice catastale come identificatore dell'unità immobiliare.
- Quindi applichiamo una decomposizione dove replichiamo solo questi due attributi per rappresentare le istanze di associazione tra proprietario e unità immobiliare.

Progetto dei dati



Quindi otteniamo la seguente decomposizione della tabella unica:

PROPRIETA' (CodiceFiscale, Cognome, Nome, DataNas,
CodiceCatasto, Via, NumeroCivico, Subalterno, Tipo)

PROPRIETARIO (CodiceFiscale, Cognome, Nome, DataNas)

UNITA'_IMMOBILIARE (CodiceCatasto, Via, NumeroCivico,
Subalterno, Tipo)

PROPRIETA' (CodiceFisale, CodiceCatasto)

Questa decomposizione elimina tutta la ridondanza inutile e conserva tutta l'informazione rappresentata nella tabella unica.

Progetto dei dati nel modello relazionale



Si noti che il modello relazionale è VALUE-BASED, e ciò significa che:

- È totalmente indipendente dalla rappresentazione fisica (tutta l'informazione è nei valori) e non ci sono meccanismi per gestire riferimenti o puntatori tra istanze di informazione.
- I legami logici tra tuple diverse si realizzano attraverso la replicazione di alcuni attributi (che hanno la proprietà di identificare il concetto e quindi di rappresentarlo): il legame tra due tuple si intende stabilito quando esse presentano gli stessi valori negli attributi replicati.
- È facile trasferire i dati da una base di dati all'altra.
- È rappresentato solo ciò che è rilevante per l'applicazione.

Terminologia



- Schema di una relazione: è costituito dal nome della relazione e da un insieme di nomi per i suoi attributi:

$R(A_1, \dots, A_n)$ oppure $R(A_1:D_1, \dots, A_n:D_n)$

dove $DOM(A_i) = D_i$

- Schema di una base di dati: è un insieme di schemi di relazione:

$S = \{R_1(A_{1,1}, \dots, A_{1,n_1}), \dots, R_m(A_{m,1}, \dots, A_{m,n_m})\}$

dove $R_1 \neq \dots \neq R_m$

Terminologia



- Istanza di una relazione di schema $R(A_1, \dots, A_n)$ con $X = \{A_1, \dots, A_n\}$:

è un insieme r di tuple su X .
- Istanza di una base di dati di schema $S = \{R_1(A_{1,1}, \dots, A_{1,n_1}), \dots, R_m(A_{m,1}, \dots, A_{m,n_m})\}$:

è un insieme di istanze di relazioni $db = \{r_1, \dots, r_m\}$ dove ogni r_i è un'istanza della relazione di schema $R_i(A_{i,1}, \dots, A_{i,n_i})$.

Esercizio



Rappresentare attraverso un insieme di relazioni le informazioni contenute in un orario ferroviario dove si riportino per ogni treno in partenza dalla stazione di Verona PN:

il numero, l'orario di partenza, la destinazione finale, la categoria, le fermate intermedie (con orario di fermata) e l'orario di arrivo alla destinazione finale.