

Capitolo terzo

Lo schema logico secondo il modello relazionale.

3.1. Introduzione

La fase della progettazione concettuale dei dati porta a definire lo Schema Concettuale.

Il modello concettuale cerca di dare una struttura ai dati evitando il più possibile i dettagli realizzativi, per cui con lo Schema Concettuale si otterrà un documento di base che descrive la struttura del sistema informatico in maniera astratta, vale a dire astratta dal modello concettuale prescelto. Quindi, è un documento che guiderà i progettisti, nella fase della progettazione logica, alla realizzazione dello Schema logico secondo un particolare modello scelto. Un modello dei dati può essere definito come *un insieme di concetti, descritti tramite un preciso formalismo, il cui scopo è quello di permettere una rappresentazione ed una manipolazione delle informazioni che costituiscono il mondo della realtà*¹. Date afferma che il modello dei dati è sempre costituito da tre componenti fondamentali :

- A. un insieme di oggetti
- B. un insieme di operatori
- C. un insieme di regole di integrità

L'insieme degli **oggetti** è costituito dallo schema logico della base dei dati secondo le regole del modello. Gli **operatori** permettono di realizzare sul database funzioni come ricerca, aggiornamento, interrogazioni, ecc., mentre le **regole di integrità** definiscono gli stati di congruenza del modello. Un modello può essere definito (Codd) come uno strumento che allo stesso tempo realizza una rappresentazione logica della realtà e una manipolazione di questa rappresentazione.

Il modello scelto per realizzare lo schema concettuale è quello relazionale elaborato da Codd nel 1970. Si tratta di un modello matematico per la descrizione dello schema logico in maniera indipendente dalla realizzazione fisica il quale trae le sue origini dalla teoria degli insiemi. Molti autori, in

seguito, hanno approfondito l'argomento cercando di formalizzare i diversi problemi che l'utilizzo del modello dei dati ha suscitato, volendo rappresentare in un sistema informatico un frammento del mondo reale.

Questo modello si è sempre più affermato nel mondo dei database divenendo, ormai, il modello standard adottato per la descrizione dello schema logico dei database, anche se nessun gestore di database può dirsi, attualmente, completamente relazionale.

3.2. Il modello relazionale dei dati

Innanzitutto vengono date alcune definizioni :

Coppia ordinata : Dati due insiemi $D1$ e $D2$, due elementi :

$$d1 \in D1 \quad d2 \in D2$$

considerati in un dato ordine costituiscono una coppia ordinata :

$$(d1, d2)$$

Tuple : Una tuple è una generalizzazione del concetto di coppia ordinata.

Dati n insiemi :

$$D1, D2, \dots, Dn$$

una tuple è una sequenza di n elementi ordinati :

$$(d1, d2, \dots, dn)$$

con :

$$d_i \in D_i \quad i = 1, 2, \dots, n$$

Relazione : un insieme di tuples costituisce una relazione definendola in questo modo (Nazzini) :

Dati n insiemi (o DOMINI), non necessariamente distinti :

$$D1, D2, \dots, Dn$$

una relazione R sui domini $D1, D2, \dots, Dn$ è un insieme di tuples :

$$d1, d2, \dots, dn$$

¹ Schiavetti : Database, Jackson 1985

con :

$$d_i \in D_i \quad i = 1, 2, \dots, n$$

A1	A2	An
d1	d2	dn
d'1	d'2	d'n
:	:	:	:
:	:	:	:
d(m)1	d(m)2	d(m)n

Alla base del modello relazionale si trova la relazione detta anche tabella o schema poiché questa è costituita su di una struttura di righe dette tuple e colonne dette domini della tabella.

Le testate della tabella A_i si dicono **attributi**, mentre il numero n dei domini viene detto **grado o arietà** della tabella. Le tabelle che hanno un solo attributo si dicono unarie, quelle con due attributi binarie, con tre ternarie e via dicendo. Il numero delle tuple si dice **cardinalità** ($m+1$).

Le proprietà delle relazioni possono essere così riassunte² :

- ? *I valori di ogni colonna sono fra loro omogenei.* I valori di un attributo appartengono allo stesso dominio (interi, stringhe di caratteri, ecc..). Quindi, si può affermare che gli attributi rappresentano l'uso dei domini in una determinata tabella
- ? *L'ordinamento delle colonne è irrilevante.* Poiché sono sempre identificate per nome e non per posizione
- ? *L'ordinamento delle righe è irrilevante.* Poiché queste sono identificate per contenuto e non per posizione
- ? *Lo schema di una tabella $R(A)$ è un'espressione che contiene il nome della tabella seguito dall'elenco di tutti gli attributi della tabella stessa. Può essere sia l'intestazione della tabella sia questa espressione : Prodotto (codice, descrizione, giacenza)*

² Atzeni, Batini, De Antonellis :Introduzione alla teoria relazionale, Masson

3.2.1. Chiavi ed attributi di una relazione

Il concetto di chiave viene elaborato nella teoria relazionale poiché le tabelle devono essere costruite in modo tale che ogni tupla deve essere distinta per mezzo di uno o più attributi. Quindi, scopo delle chiavi è identificare univocamente queste tuple. Si distingue fra superchiave, chiave candidate e chiave primaria. La **superchiave** è l'insieme degli attributi che identificano univocamente una tuple. La **chiave candidata** è in numero minimo di attributi necessari per identificare univocamente una tuple. Può essere definita come *una superchiave dalla quale possono essere eliminati attributi senza distruggere la proprietà della identificazione univoca della tuple. Per cui una chiave candidata è una superchiave con la proprietà di non ridondanza*³.

Fra le chiavi candidate, che possono essere più di una, se ne sceglierà una che ha il minor numero di attributi e che, a parità di attributi, ha il minor numero di caratteri. Questa è la **chiave primaria** e sarà la chiave scelta fra le chiavi candidate a rappresentare univocamente una tuple.

Rispetto alla chiave della tabella si possono distinguere gli **attributi primi**, cioè quelli che fanno parte di almeno una chiave candidata, e i restanti **attributi non primi**. Infine, quegli attributi che possono costituire la chiave di altre tabelle sono detti **chiave esterna** e permettono di stabilire associazioni fra le tabelle che rappresentano diverse entità.

3.3. Traduzione dei diagrammi E|R di uno schema concettuale in un modello relazionale

Si può tradurre abbastanza agevolmente gli schemi E\R secondo le strutture delle basi di dati relazionali seguendo, però, alcune regole.

3.3.1 Entità e gerarchie IS-A

³ Nazzini, Sanges, Vaccaro :Introduzione ai Data Base relazionali

Le **entità** degli schemi E|R vengono tradotte in altrettante tabelle. Gli attributi delle entità vengono riportati nelle colonne mentre le occorrenze delle entità saranno tradotte in tuples.

Per tradurre, invece, le strutture **IS-A** nel modello relazionale si possono adottare diverse soluzioni.

Una può essere quella di inserire tutto in una tabella che contenga gli attributi comuni e gli attributi caratteristici di ciascuna classe. Per questo in ogni tupla saranno vuote alcune colonne relative ad alcuni attributi a seconda che, come nell'esempio del capitolo 2, il dipendente appartenga alla categoria impiegati o dipendenti. In questo caso ci sarà uno spreco di spazio. Un'altra soluzione, che può porre rimedio a questo inconveniente, è quello di creare una tabella che contenga tutti gli attributi comuni e inserire gli attributi specifici dei sottoinsiemi definiti dalla IS-A in altre tabelle legate a questa da associazioni 1 ad 1. Quindi, nell'esempio della gerarchia IS-A individuata dal diagramma dipendenti si può inserire in una tabella gli attributi comuni e realizzare due tabelle, operai ed impiegati, in cui inserire gli attributi specifici delle due categorie.

3.3.2. Associazioni 1 a 1 e 1 a n

Per tradurre il legame tra due entità che viene realizzato da un'associazione 1 a 1, si deve inserire tra gli attributi di almeno una delle entità la chiave esterna, cioè quell'attributo o insieme di attributi che identificano l'entità ad essa collegata. Le associazioni 1 a n sono tradotte in modo analogo. Occorre però prestare attenzione a non inserire però **attributi ripetuti**, siano chiavi esterne o attributi normali, perché possono essere causa di problemi nella gestione delle tabelle. Ad esempio, in una relazione 1 a n fra due entità, *cliente* e *numero d'ordine*, inserisco la chiave primaria della tabella cliente nella tabella ordini divenendo chiave esterna. Se facessi l'inverso, ovvero se inserissi come chiave esterna nella relazione clienti la chiave della tabella ordini, dovrei inserire per ogni cliente tutti gli ordini creando molte tuples.