

Il modello relazionale

09:41 Tue 9 Jan Database

Liceo Database

Progettazione Logica: **MODELLO RELAZIONALE**

Prodotto Cartesiano → **RELAZIONE** → Tabella

Attributi

dominio

tupla

cardinalità

grado

DAL MODELLO E/R AL MODELLO RELAZIONALE

The diagram illustrates the transition from an Entity-Relationship (E/R) model to a Relational model. It features a table representing a relation with a pink header row for attributes, a green row for the domain, and a yellow row for a tuple. The table is annotated with 'Attributi' (Attributes), 'dominio' (domain), 'tupla' (tuple), 'cardinalità' (cardinality), and 'grado' (degree). Below this, the text 'DAL MODELLO E/R AL MODELLO RELAZIONALE' is written. The E/R diagram shows two entities connected by an association, with primary keys (PK) indicated. Arrows point from these entities to their respective relational tables, where the primary key is highlighted with red diagonal lines.



Il modello relazionale

- Il modello relazionale è stato introdotto nel 1970 da E.F. Codd.
- Soltanto a metà degli anni ottanta ha trovato una buona diffusione sul mercato, in quanto all'epoca della sua introduzione non esistevano soluzioni efficienti per implementare le strutture relazionali.
- Il modello relazionale si basa su due concetti chiave strettamente correlati:
 - La nozione matematica di relazione permette la formulazione e lo sviluppo di una teoria efficace a supporto del modello.
 - La nozione di tabella è semplice, supportando l'intuizione anche dei non "addetti ai lavori".
- Il successo del modello relazionale rispetto ad altri modelli (gerarchico, a grafo ecc.) è anche dovuto all'indipendenza dei dati e dell'interazione con essi dall'implementazione fisica delle relative strutture.



Relazioni

- Dati due insiemi D_1 e D_2 , il loro *prodotto cartesiano* si indica con $D_1 \times D_2$ ed è l'insieme di tutte le possibili *coppie* il cui primo elemento appartiene a D_1 ed il cui secondo elemento appartiene a D_2 .
- Una relazione su D_1 e D_2 (detti domini della relazione) è un sottoinsieme di $D_1 \times D_2$. Ad esempio se $D_1 = \{1, 2, 4\}$ e $D_2 = \{a, b\}$, allora $D_1 \times D_2 = \{(1, a), (1, b), (2, a), (2, b), (4, a), (4, b)\}$. In questo caso un esempio di relazione su D_1 e D_2 è $\{(1, a), (1, b), (4, b)\}$.
- Le nozioni di prodotto cartesiano e relazione possono essere generalizzati al caso di n insiemi D_1, D_2, \dots, D_n (non necessariamente distinti fra loro). In questo caso invece di coppie si avrà a che fare con n -tuple, i.e., elementi del tipo (v_1, v_2, \dots, v_n) dove v_1 appartiene a D_1 , v_2 a D_2 , \dots , v_n a D_n .
- In generale non vi sono vincoli sulla natura degli insiemi coinvolti. Tuttavia, siccome i computer possono memorizzare soltanto un numero finito di informazioni, si assume che le relazioni siano insiemi finiti. I domini invece possono essere infiniti in modo da poter assumere sempre l'esistenza di un valore non contenuto nel database.

Relazioni e tabelle

- Dati gli insiemi $D_1 = \{1, 2, 4\}$ e $D_2 = \{a, b\}$, il prodotto cartesiano $D_1 \times D_2 = \{(1, a), (1, b), (2, a), (2, b), (4, a), (4, b)\}$ e la relazione $\{(1, a), (1, b), (4, b)\}$ ammettono la seguente rappresentazione sotto forma di tabelle:

1	a
1	b
2	a
2	b
4	a
4	b

1	a
1	b
4	b

- Dato il prodotto cartesiano $D_1 \times D_2 \times \dots \times D_n$ il numero n è il grado del prodotto e delle relazioni sugli n domini. Il numero di n -tuple di una relazione costituisce invece la cardinalità di quest'ultima.



Relazioni con attributi

- Data una relazione, non ha importanza l'ordine in cui sono elencate le n-tuple che la compongono non è importante (l'ordine delle righe nelle tabelle è soltanto ai fini della presentazione). Si noti inoltre che ogni n-tupla è *unica*.
- Una n-tupla (v_1, v_2, \dots, v_n) stabilisce delle connessioni fra i dati che contiene; tali legami sono intrinsecamente dipendenti dalla posizione dei dati all'interno della n-tupla (notazione *posizionale*).
- Infatti, cambiando l'ordine dei componenti di una n-tupla, cambia anche l'interpretazione delle connessioni fra i dati (ogni posizione corrisponde ad un certo dominio).
- Per evitare i problemi connessi all'ordinamento dei componenti delle n-tuple, si definisce una notazione non posizionale che distingue i vari domini (e quindi i componenti) assegnando loro dei nomi univoci.



Relazioni con attributi

- Nella visualizzazione per mezzo di tabelle gli attributi diventano le intestazioni delle colonne:

HomeTeam	VisitorsTeam	HomeGoals	VisitorsGoals
Real Madrid	Liverpool	3	1
Liverpool	Milan	2	0
Real Madrid	Roma	1	2

- Formalmente esiste una funzione $DOM: X \rightarrow \mathbf{D}$ che associa ad ogni attributo $A \in X$ un dominio $DOM(A) \in \mathbf{D}$. Quindi abbiamo quanto segue:
 - Una n-tupla su un insieme di attributi X è una funzione t che associa ad ogni attributo $A \in X$ un valore del dominio $DOM(A)$. Ad esempio per la seconda tupla della tabella precedente vale $t[\text{HomeTeam}] = \text{Liverpool}$.
 - Una relazione su X è un insieme di tuple su X .



Relazioni e database

- Solitamente un database è composto da più relazioni con valori comuni laddove vi siano delle corrispondenze fra di esse.
- Esempio:

Studenti

Matricola	Cognome	Nome	DataDiNascita
276545	Rossi	Mario	25/11/1980
485745	Verdi	Luisa	23/04/1981
200768	Gialli	Luigi	12/02/1981
587614	Rosa	Maria	10/10/1980

Esami

Studente	Voto	Corso
276545	24	01
276545	27	04
587614	30	01
200768	28	04

Codice	Titolo	Docente
01	Fisica	Tizio
03	Chimica	Caio
04	Algebra	Sempronio

Corsi



Relazioni e database

- Nell'esempio precedente la prima relazione contiene dei dati su un insieme di studenti (n. di matricola, cognome, nome, data di nascita).
- La terza relazione contiene dei dati su un insieme di corsi (codice, titolo e docente).
- La seconda relazione contiene dei dati (n. di matricola, voto e codice del corso) relativi agli esami sostenuti dagli studenti. Tramite il primo ed il terzo attributo vengono stabilite due corrispondenze con la prima e la terza relazione rispettivamente.
- Siccome le corrispondenze tra n-tuple appartenenti a relazioni diverse sono fondate sui valori di certi attributi, si dice che il modello relazionale è value-based, ossia, basato sui valori.

Informazioni incomplete e valori NULL



- Il modello relazionale impone una certa rigidità in quanto l'informazione deve essere sempre rappresentata da tuple di valori omogenei.
- Spesso tuttavia capita che i dati disponibili non corrispondano esattamente allo schema di relazione in quanti manca un dato.
- L'utilizzo di particolari valori del dominio associato ad un attributo per denotare che il valore corrispondente manca non è una buona idea in quanto può generare confusione.
- Il concetto di relazione viene quindi esteso in modo da permettere ad ogni tupla di assumere per ogni attributo o un valore del dominio corrispondente oppure un valore speciale detto *null value* (indicato dalla stringa NULL).
- Il significato del null value è quello dell'assenza di informazione, i.e., il valore potrebbe non esistere affatto od essere semplicemente sconosciuto.



Notazione

- Uno schema di relazione consiste in un simbolo R (nome della relazione) ed in un insieme di nomi di attributi $X = \{A_1, A_2, \dots, A_n\}$. Complessivamente uno schema di relazione è denotato dall'espressione $R(X)$.
- Uno schema di database consiste in un insieme di schemi di relazioni con nomi distinti e si denota con l'espressione $\mathbf{R} = \{R_1(X_1), R_2(X_2), \dots, R_m(X_m)\}$.
- Un'istanza di relazione (o semplicemente una relazione) sullo schema $R(X)$ è un insieme r di tuple su X .
- Un'istanza di database (o semplicemente un database) sullo schema $\mathbf{R} = \{R_1(X_1), R_2(X_2), \dots, R_m(X_m)\}$ è un insieme di relazioni $\mathbf{r} = \{r_1, r_2, \dots, r_m\}$ tale che ogni r_i è una relazione sullo schema $R_i(X_i)$ per $1 \leq i \leq m$.
- Esempio:
 $\mathbf{R} = \{$ Studenti(Matricola, Cognome, Nome, DataDiNascita),
 Esami(Studente, Voto, Corso), Corsi(Codice, Titolo, Docente)
 }

Correttezza dell'informazione

- Nonostante il modello relazionale permetta di rappresentare ed organizzare l'informazione in modo conveniente per le applicazioni, non è detto che ogni insieme di tuple rappresenti dell'informazione *corretta* rispetto ad un'applicazione.
- Esempio:

Studenti

Matricola	Cognome	Nome	DataDiNascita
276545	Rossi	Mario	25/11/1980
276545	Verdi	Luisa	23/04/1981
200768	Gialli	Luigi	12/02/1981
587614	Rosa	Maria	10/10/1980

Codice	Titolo	Docente
01	Fisica	Tizio
03	Chimica	Caio
04	Chimica	Sempronio

Corsi

Esami

Studente	Voto	Lode	Corso
276545	35	No	07
276545	27	Sì	04
587614	30	No	01
300000	28	No	04



Vincoli di integrità

- Per evitare situazioni in cui le tuple delle relazioni di un database non rappresentano informazioni corrette, viene introdotto il concetto di *vincolo di integrità*.
- Ogni vincolo è un predicato che associa il valore vero o il valore falso ad ogni istanza di database a seconda che quest'ultimo soddisfi o meno il vincolo stesso.
- I vincoli di integrità sono classificabili come segue:
 - *Vincoli intra-relazionali* (ogni vincolo di questo tipo coinvolge una sola relazione del database); si possono suddividere ulteriormente in:
 - *Vincoli di tupla* (ogni vincolo va valutato su ogni tupla presa singolarmente).
 - *Vincoli di valore o di dominio* (impongono delle restrizioni sui domini degli attributi): coinvolgono un singolo attributo.
 - *Vincoli inter-relazionali* (ogni vincolo di questo tipo coinvolge più relazioni del database).



Vincoli di tupla

- I vincoli sui valori che una singola tupla può assumere vengono indicati con una sintassi che contempla gli operatori logici AND, OR, NOT e degli atomi che specificano dei confronti fra espressioni costruite a partire dai valori degli attributi.
- Ad esempio i vincoli sulla relazione Corsi possono essere specificati come segue:
(Voto \geq 0) AND (Voto \leq 30)
(NOT(Lode='Sì')) OR (Voto=30)
Il primo vincolo è di fatto un vincolo di dominio in quanto coinvolge soltanto l'attributo Voto.
- Si possono specificare anche vincoli più complessi.
Ad esempio dato lo schema di relazione
Pagamenti(Data,Imponibile,Tasse,ValoreNetto)
si può specificare il vincolo di integrità
ValoreNetto=Imponibile-Tasse



Chiavi

- Intuitivamente una chiave è un insieme di attributi che consente di individuare in modo univoco le tuple di una relazione.
- Formalmente si hanno le seguenti definizioni:
 - Un insieme di attributi K è una *superchiave* per la relazione r se quest'ultima non contiene due tuple distinte t_1 e t_2 tali che $t_1[K]=t_2[K]$.
 - Un insieme di attributi K è una *chiave* per la relazione r se K è la superchiave minimale per r (i.e., se non esiste un'altra superchiave K' per r contenuta in K).

- Il vincolo intra-relazionale più importante è quello di chiave che si specifica elencando gli attributi che costituiscono la chiave stessa. Ad esempio per la relazione

Studenti(Matricola,Cognome,Nome,DataDiNascita)

i vincoli di chiave possibili sono i seguenti (assumendo che non possano esistere due persone con lo stesso nome, lo stesso cognome e la stessa data di nascita):

Matricola

Cognome, Nome, DataDiNascita



Chiavi

- Ogni relazione ha sempre una chiave; infatti ogni tupla di una relazione $r(X)$ è unica, quindi l'insieme di attributi X è una superchiave per essa. A questo punto si hanno due casi:
 - La superchiave è anche la chiave per la relazione $r(X)$.
 - Esiste un'altra superchiave K' contenuta in K : si ripete il ragionamento rispetto a K' fino ad individuare una superchiave minimale. Tale procedimento termina sicuramente visto che gli attributi di una relazione sono in numero finito.
- L'esistenza di una chiave per ogni relazione ha le seguenti importanti conseguenze:
 - È garantito l'accesso in modo non ambiguo a tutti i valori del database.
 - È possibile avere delle connessioni fra dati contenuti in relazioni distinte.



Chiave primaria

- Nel caso in cui i valori di una chiave siano nulli (NULL), non è più possibile riferirsi in modo univoco alla tupla corrispondente. Inoltre non è più possibile stabilire delle corrispondenze con tuple di altre relazioni.
- Per evitare questi effetti indesiderati si stabilisce che i valori NULL non possano occorrere in una delle chiavi di una relazione, detta *chiave primaria* (mentre possono eventualmente occorrere nelle rimanenti chiavi).
- Negli schemi di relazione e nelle rappresentazioni sotto forma di tabella le chiavi primarie vengono evidenziate sottolineando gli attributi che le compongono (ad esempio Studenti(Matricola,Cognome,Nome,DataDiNascita)).
- Le connessioni fra tuple di relazioni distinte sono quindi solitamente realizzate per mezzo delle chiavi primarie.
- Se non è possibile trovare un insieme di attributi per la chiave primaria, si introduce un attributo apposito con la funzione di codice che viene generato in modo univoco al momento dell'inserzione delle tuple.



Vincoli referenziali

- I vincoli referenziali (foreign key) sono il tipo di vincolo inter-relazionale più importante.

<u>Codice</u>	<u>Data</u>	<u>Agente</u>	<u>Dipartimento</u>	<u>Registrazione</u>
143256	25/10/92	567	75	5694 FR
987554	26/10/92	456	75	5694 FR
987557	26/10/92	456	75	6544 XY
630876	15/10/92	456	47	6544 XY
539856	12/10/92	567	47	6544 XY

Agenti

<u>NumReg</u>	<u>Cognome</u>	<u>Nome</u>
567	Brun	Jean
456	Larue	Henri
638	Larue	Jacques

Contravvenzioni

<u>Registrazione</u>	<u>Dipartimento</u>	<u>Proprietario</u>	<u>Indirizzo</u>
6544 XY	75	Cordon Eduard	Rue du Pont
7122 HT	75	Cordon Eduard	Rue du Pont
5694 FR	75	Latour Hortense	Avenue Foch
6544 XY	47	Mimault Bernard	Avenue FDR

Veicoli



Vincoli referenziali

- Un vincolo referenziale permette di stabilire delle corrispondenze fra tuple appartenenti a due relazioni distinte specificando un insieme di attributi X della prima in modo tale che
 - i valori in X di ogni tupla della prima relazione appaiano come valori della chiave primaria di una tupla della seconda relazione.
- Nell'esempio precedente vi sono due vincoli referenziali:
 - Il primo vincolo è fra l'attributo Agente della relazione Contravvenzioni e l'attributo NumReg della relazione Agenti.
 - Il secondo vincolo è fra gli attributi Registrazione e Dipartimento della relazione Contravvenzioni e gli attributi con gli stessi nomi della relazione Veicoli.
- I vincoli referenziali sono diretti verso la chiave primaria della seconda relazione



Vincoli referenziali

- Indicando con $X=A_1A_2...A_p$ un insieme di attributi di R_1 , con $K=B_1B_2...B_p$ gli attributi che compongono la chiave primaria di R_2 si ha che un vincolo referenziale fra gli attributi X di R_1 e R_2 è soddisfatto se
 - per ogni tupla t_1 di R_1 che non contenga valori nulli (NULL) in corrispondenza degli attributi X , esiste una tupla t_2 di R_2 tale che $t_1[A_i]=t_2[B_i]$ per $1 \leq i \leq p$.
- L'ordine in cui vengono specificati gli attributi di X e K è importante. Infatti supponendo di avere una relazione con schema

Incidenti(Codice, Dipartimento1, Registrazione1, Dipartimento2, Registrazione2)
per stabilire un vincolo referenziale con la relazione Veicoli
bisogna necessariamente usare l'ordinamento (dato che i nomi degli attributi in gioco non coincidono con quelli della chiave primaria di Veicoli).

