

Modello relazionale

Index

- [Introduzione](#)
 - [Definizioni](#)
 - [Relazioni e tabelle](#)
 - [Esempio](#)
 - [Valori nulli](#)
 - [Vincoli di integrità](#)
 - [Vincoli intrarelazionali](#)
 - [Esempio](#)
 - [Vincoli interrelazionali](#)
 - [Esempio](#)
 - [Chiavi](#)
 - [Dipendenza funzionale](#)
 - [Esempio](#)
-

Introduzione

Il modello relazione venne proposto per la prima volta da Codd nel 1970 per favorire l'indipendenza dei dati, ma fu disponibile in DBMS reali solo a partire dal 1981.

Questo modello è basato sulla nozione matematica di **relazione** le quasi si traducono in maniera naturale in **tabelle** (infatti useremo sempre il termine relazione invece di tabella). Dati e relazioni (riferimenti) tra dati di insiemi (tabelle) diversi sono rappresentati come **valori**

Definizioni

Il **dominio** un insieme possibilmente infinito di valori (es. insieme dei numeri interi, insieme delle stringhe di caratteri di lunghezza 20 ecc.). Siano D_1, D_2, \dots, D_k domini, non necessariamente distinti. Il prodotto cartesiano di tali domini è denotato da:

$$D1 \times D2 \times \dots \times Dk$$

è l'insieme

$$\{(v1, v2, \dots vk) | v1 \in D1, v2 \in D2, \dots V_k \in Dk\}$$

Una **relazione matematica** è un qualsiasi sottoinsieme del prodotto cartesiano di uno o più domini

Una relazione che è sottoinsieme del prodotto cartesiano di k domini si dice di **grado k**

Gli elementi di una relazione sono detti **tuple**. Il numero di tuple in una relazione è la sua **cardinalità**. Ogni tupla di una relazione di grado k ha k componenti ordinate ma non c'è ordinamento tra le tuple

Esempio

- supponiamo $k = 2$
- $D1 = \{\text{bianco}, \text{nero}\}, D2 = \{0, 1, 2\}$
- $D1 \times D2 = \{(\text{bianco}, 0), (\text{bianco}, 1), (\text{bianco}, 2), (\text{nero}, 0), (\text{nero}, 1), (\text{nero}, 2)\}$

$\{(\text{bianco}, 0), (\text{nero}, 0), (\text{nero}, 2)\}$ sono una relazione di grado 2, cardinalità 3 e con tuple $(\text{bianco}, 0), (\text{nero}, 0), (\text{nero}, 2)$

$\{(\text{nero}, 0), (\text{nero}, 2)\}$ è una relazione di grado 2, cardinalità 2 e con tuple $(\text{nero}, 0), (\text{nero}, 2)$

Relazioni e tabelle

Però a questo punto in che modo posso interpretare i dati nella tabella?

Utilizzo una notazione anche per la tabella e le colonne

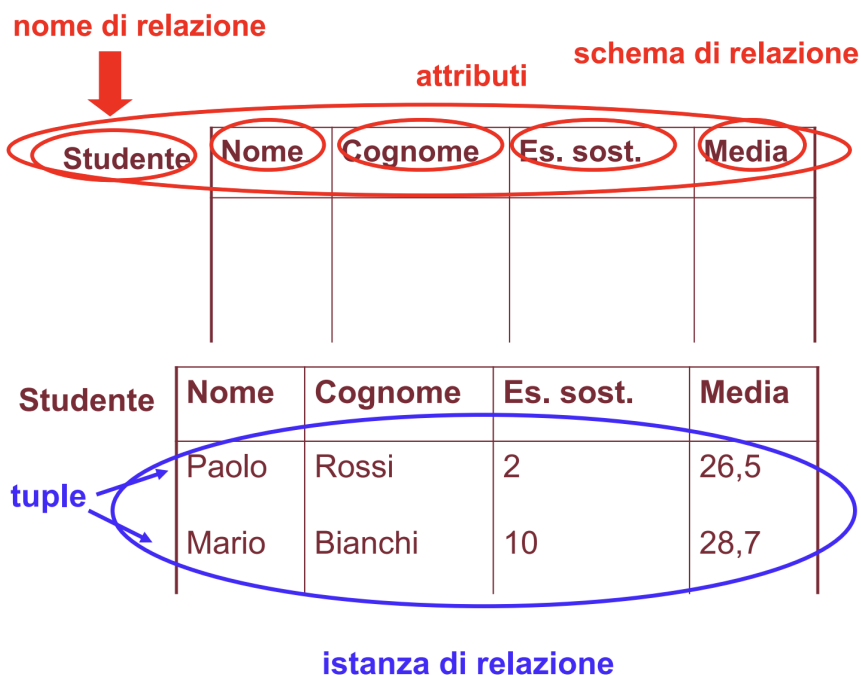
Infatti uso:

- un **attributo** è definito da un nome A (che ne descrive il ruolo) e dal **dominio** dell'attributo a che indichiamo con $\text{dom}(A)$ (la coppia dominio, nome di attributo è definita attributo)
- sia R un insieme di attributi. Un'ennupla su R è una **funzione** definita su R che associa ad ogni attributo A in R un elemento di $\text{dom}(A)$
- se t è un'ennupla su R ed A è un attributo in R , allora con $t(A)$ indicheremo il valore assunto dalla funzione t in corrispondenza dell'attributo A
- con **schema di relazione** l'rappresento l'insieme degli attributi di una relazione $R(A1, A2, \dots, Ak)$ (es. Info_Città(Città, Regione, Popolazione)). Questo rimane invariato nel tempo e descrive la struttura stessa della relazione. Per

schema di base di dati si intende un insieme di schemi di relazione con nomi differenti

- per **istanza** di una relazione con schema $R(X)$ si indica l'insieme R di tuple su X . Questa contiene i valori attuali, che possono cambiare molto rapidamente nel tempo (corpo della relazione)
- con $t[A_i]$ si indica il **valore dell'attributo** con nome A_i della tupla t (nell'esempio sotto se t è la seconda tupla, allora $t[\text{Cognome}] = \text{Bianchi}$)
- se Y è un sottoinsieme di attributi dello schema X di una relazione allora $t[Y]$ è il sottoinsieme di valori nella tupla t che corrispondono ad attributi contenuti in Y . Questo è chiamato **restrizione** di t

Esempio



Valori nulli

I valori **NULL** rappresentano la mancanza di informazione o il fatto che l'informazione non è applicabile. Questo valore può essere assegnato a un qualunque dominio, indipendentemente da come è definito.

Tutti i valori NULL sono considerati diversi tra di loro (un valore NULL nel campo di una tupla è diverso dal valore NULL di un altro campo di una stessa tupla oppure dello stesso campo di un'altra tupla)

Vincoli di integrità

I **vincoli di integrità** sono delle proprietà che devono essere soddisfatte da ogni istanza della base di dati (sono **legate allo schema**). Questi descrivono proprietà specifiche del campo di applicazione, e quindi delle informazioni ad esso relative modellate attraverso la base di dati.

Una istanza di base di dati è corretta se soddisfa tutti i vincoli di integrità associati al suo schema

Esistono due tipi di vincoli:

- **Vincoli intrarelazionali** → definiti sui valori di singoli attributi (di dominio) o tra valori di attributi di una stessa tupla o tra tuple della stessa relazione
- **Vincoli interrelazionali** → definiti tra più relazioni

Vincoli intrarelazionali

Questi possono essere:

- Vincolo di chiave primaria (**primary key**) → unica e mai nulla
- Vincoli di dominio (es. ASSUNZIONE > 1980)
- Vincoli di unicità (**unique**)
- Vincoli di esistenza del valore per un certo attributo (**not null**)
- Espressioni sul valore di attributi della stessa tupla (es. data_arrivo < data_partenza)

Esempio

IMPIEGATO CODICE	COGNOME	NOME	RUOLO	ASSUNZIONE	DIP
COD1	Rossi	Mario	Analista	1795	01
COD2	Bianchi	Pietro	Analista	1990	05
COD2	Neri	Paolo	Amministratore	1985	01

DIPARTIMENTO
NUMERO NOME

01	Progettazione
02	Amministrazione

(ASSUNZIONE > 1980)

COD2 UNIQUE

DIP REFERENCES
DIPARTIMENTO.NUMERO

Esami	Studente	Voto	Lode	Corso	
	276545	32		01	(Voto \geq 18) AND (Voto \leq 30)
	276545	30	si	02	
	787643	27	si	03	(Voto = 30) OR NOT (Lode = „si“)
	739430	24		04	vincoli intra-relazionali

Studente references Studenti.Matricola vincoli inter-relazionali

Studenti	Matricola	Cognome	Nome	
	276545	Rossi	Mario	
	787643	Neri	Piero	Matricola unique
	787643	Bianchi	Luca	

Vincoli di dominio

- ASSUNZIONE > 1980
- (Voto \geq 18) AND (Voto \leq 30)

Vincoli di tupla

- (Voto = 30) OR NOT (Lode = “si“)

Vincoli tra valori in tuple di relazioni diverse

- DIP REFERENCES DIPARTIMENTO.NUMERO
- Studente REFERENCES Studenti.Matricola

Vincoli interrelazionali

- Vincolo di integrità referenziale (**foreign key**) → quando porzioni di informazione in relazioni diverse sono correlate attraverso valori di chiave

Esempio

l'attributo Vigile della relazione INFRAZIONI e l'attributo Matricola (chiave) della relazione VIGILI

Infrazioni

<u>Codice</u>	<u>Data</u>	<u>Vigile</u>	<u>Prov</u>	<u>Numero</u>
34321	1/2/95	3987	MI	39548K
53524	4/3/95	3295	TO	E39548
64521	5/4/96	3295	PR	839548
73321	5/2/98	9345	PR	839548

Vigili

<u>Matricola</u>	<u>Cognome</u>	<u>Nome</u>
3987	Rossi	Luca
3295	Neri	Piero
9345	Neri	Mario
7543	Mori	Gino

gli attributi Prov e Numero di INFRAZIONI e gli attributi Prov e Numero (chiave) della relazione AUTO

Infrazioni

<u>Codice</u>	<u>Data</u>	<u>Vigile</u>	<u>Prov</u>	<u>Numero</u>
34321	1/2/95	3987	MI	39548K
53524	4/3/95	3295	TO	E39548
64521	5/4/96	3295	PR	839548
73321	5/2/98	9345	PR	839548

Auto

<u>Prov</u>	<u>Numero</u>	<u>Cognome</u>	<u>Nome</u>
MI	39548K	Rossi	Mario
TO	E39548	Rossi	Mario
PR	839548	Neri	Luca

Chiavi

Una **chiave** di una relazione (non necessariamente unica) è un attributo o insieme di attributi (chiave **composta**) che identifica univocamente una tupla

Un attributo per essere considerato una chiave deve rispettare queste condizioni:

1. per ogni istanza di una relazione R, non esistono due tuple distinte t1 e t2 che hanno gli stessi valori per gli attributi in un insieme X (chiavi), tali cioè che

$$t1[X] = t2[X]$$

- nessun sottoinsieme proprio di X soddisfa la prima condizione

Una relazione potrebbe avere inoltre più chiavi alternative. Quella più usata o quella composta da numero minore di attributi viene scelta come chiave **primaria**. La chiave primaria non ammette valori nulli e ne deve esistere almeno una all'interno di ogni relazione (sono infatti le chiavi a consentire di mettere in relazione dati in tabelle diverse)

Si parla di chiave **minimale** quando una chiave non contiene un sottoinsieme di attributi che a sua volta è una chiave (si applica ai sottoinsiemi di super-chiave). Si parla di **super-chiave** quando un insieme di attributi contiene una chiave (una chiave è in senso improprio una super-chiave ma non il contrario)

Dipendenza funzionale

Una **dipendenza funzionale** stabilisce un particolare legame semantico tra due insiemi non-vuoti di attributi X e Y appartenenti ad uno schema R

Tale vincolo si scrive $X \rightarrow Y$ e si legge X determina Y

diremo che una relazione r con schema R **soddisfa** la dipendenza funzionale $X \rightarrow Y$ se:

- la dipendenza funzionale $X \rightarrow Y$ è applicabile ad R , nel senso che sia X sia Y sono sottoinsiemi di R
- le ennuple in r che concordano su X concordano anche su Y , cioè per ogni coppia di ennuple $t1$ e $t2$ in r (se hanno la stessa X devono avere la stessa Y)

$$t1[X] = t2[X] \rightarrow t1[Y] = t2[Y]$$

Esempio

Supponiamo di avere uno schema di relazione

VOLI(CodiceVolo, Giorno, Pilota, Ora)

Con i vincoli:

- un volo con un certo codice parte sempre alla stessa ora
- esiste un solo volo con un dato pilota, in un dato giorno, ad una data ora

I vincoli corrispondono alle dipendenze funzionali:

- $\text{CodiceVolo} \rightarrow \text{Ora}$
- $\{\text{Giorno}, \text{Pilota}, \text{Ora}\} \rightarrow \text{CodiceVolo}$
- $\{\text{CodiceVolo}, \text{Giorno}\} \rightarrow \text{Pilota}$