

2 - Modello relazionale

Il modello relazionale si basa su due concetti:

- **Relazione** (formalmente in matematica)
- **Tabella** (intuitivo per tutti gli utenti)

Ricordiamo in matematica che:

Dati due insiemi A e B (detti **domini** della relazione), dicesi relazione tra A e B un qualsiasi sottoinsieme del loro prodotto cartesiano ($A \times B$).

Una **relazione matematica** sui domini D_1, D_2, \dots, D_n è un sottoinsieme del prodotto cartesiano $D_1 \times D_2 \times \dots \times D_n$.

Il numero n delle componenti del prodotto cartesiano rappresenta il **grado** della relazione, il numero di elementi (cioè n -uple) della relazione rappresenta la **cardinalità** della relazione.

È bene, comunque, sottolineare le differenze che ci sono tra il concetto di relazione ed il concetto di tabella;

Infatti, poiché una relazione è un insieme di n -uple, si ha che:

- fra le n -uple non è definito alcun ordinamento, quindi l'ordine presente in una tabella è occasionale: due tabelle con le stesse righe ma in ordine diverso rappresentano la stessa relazione;
- le n -uple di una relazione sono distinte l'una dall'altra, in quanto tra gli elementi di un insieme non sono ammessi duplicati, quindi una tabella rappresenta una relazione solo se le sue righe sono l'una diversa dall'altra.

Al tempo stesso però, ciascuna n -upla è internamente ordinata: l' i -esimo valore di ciascuna proviene dall' i -esimo dominio:

$$(v_1, v_2, \dots, v_n) \quad \text{con} \quad v_1 \in D_1, v_2 \in D_2, \dots, v_n \in D_n$$

Questo implica che ci sia un ordinamento tra i domini (tra le colonne della tabella), che è significativo ai fini dell'interpretazione dei dati nelle relazioni.

Risulta evidente come le informazioni che siamo interessati ad organizzare nelle relazioni dei DB abbiano una struttura riconducibile a quella dei record: una relazione è sostanzialmente un insieme di record omogenei, cioè definiti sugli stessi campi, di conseguenza attribuendo ad ogni dominio un nome identificativo, detto **attributo**, l'ordinamento dei domini (quindi la posizione delle colonne nella tabella) diventa irrilevante.

| | | | | | | |
|--|--|--|------------------|----------------|-------------|------------------------|
| | | | Studenti | | | |
| | | | Matricola | Cognome | Nome | Data di Nascita |
| | | | 276545 | Rossi | Maria | 25/11/1994 |
| | | | 485745 | Neri | Anna | 23/04/1997 |
| | | | 200768 | Verdi | Fabio | 12/02/2002 |
| | | | 587614 | Rossi | Luca | 10/10/2004 |
| | | | 937653 | Bruni | Mario | 01/12/2003 |

| | | | | | |
|-----------------|-------------|--------------|--------------|--|--|
| | | | Esami | | |
| Studente | Voto | Corso | | | |
| 276545 | 28 | 01 | | | |
| 276545 | 27 | 04 | | | |
| 200768 | 24 | 04 | | | |

| | | | | | |
|---------------|----------------------------|----------------|--------------|--|--|
| | | | Corsi | | |
| Codice | Titolo | Docente | | | |
| 01 | Algoritmi e strutture dati | G. Pio | | | |
| 03 | Architettura elaboratori | G.Castellano | | | |
| 04 | Basi di dati | P. Buono | | | |

Rispetto al modello basato su record e puntatori (gerarchico e reticolare), il modello relazionale, basato su valori, presenta diversi vantaggi:

1. **Astrazione:** Richiede di rappresentare solo ciò che è rilevante dal punto di vista di applicazione/utente.
2. **Portabilità:** essendo tutta l'informazione contenuta nei valori, è relativamente semplice trasferire i dati da un contesto ad un altro.
Al contrario, i puntatori hanno un significato locale al singolo sistema, che non sempre è immediato esportare
3. **Indipendenza fisica dei dati:** la rappresentazione logica dei dati non fa alcun riferimento a quella fisica, che può anche cambiare nel tempo;

Possiamo a questo punto riassumere le definizioni relative al modello relazionale distinguendo il livello degli schemi da quello delle istanze:

- Uno **schema di relazione** è costituito da un simbolo R , detto nome della relazione e da un insieme di attributi $X = \{A_1, A_2, \dots, A_n\}$, il tutto di solito indicato con $R(X)$.
A ciascun attributo è associato un dominio, come già visto.
- Uno **schema di basi** di dati è un insieme di schemi di relazione con nomi diversi:

$$R = \{R_1(X_1), R_2(X_2), \dots, R_n(X_n)\}$$

- Una **istanza di relazione (o relazione)** su uno schema $R(X)$ è un insieme di r tuple su X .
- Una **istanza di base di dati (o base di dati)** su uno schema $R = \{R_1(X_1), R_2(X_2), \dots, R_n(X_n)\}$ è un insieme di relazioni $r = \{r_1, r_2, \dots, r_n\}$, dove ogni r_i , per $1 \leq i \leq n$, è una relazione sullo schema $R_i(X_i)$.

Lo **schema** quindi è la parte invariante (struttura, intenzione), mentre l'**istanza** è la parte variabile (i dati effettivi, estensione) che cambia nel **tempo**

Tabella 2.1: Tabella delle convenzioni

| SIGNIFICATO | NOTAZIONE |
|----------------------|--------------------|
| Attributi | A, B, C, A', A_1 |
| Insiemi di attributi | X, Y, Z, X', X_1 |
| $X = \{A, B, C\}$ | $X = ABC$ |
| $X \cup Y$ | XY |
| $X \cup A$ | XA |
| Nomi di relazioni | R, S, R', R_1 |
| Istanze (relazioni) | r, s, r', r_1 |

Valori nulli o incompleti

Quando si aggiunge una tupla ad una relazione può non essere possibile specificare il valore di un attributo per alcune ragioni come:

- Valore non applicabile
- Valore sconosciuto
- Valore senza informazioni

Questa assenza di valore si potrebbe ovviare usando un valore del dominio per rappresentare l'assenza (0 per esempio), ma non va bene per 2 ragioni specifiche:

- **Si richiede un valore mai utilizzato per valori significativi** (generalmente impossibile a meno di condizioni specifiche)
- **L'uso di valori del dominio genera confusione**: la distinzione tra valori veri e fittizi è nascosta, costringendo i programmi che accedono alla base di dati di tenerne conto, distinguendo opportunamente e tenendo conto di quali sono in ciascun caso

Per poter rappresentare in modo semplice e comodo si estende la possibilità di inserire un **valore nullo**, ben distinto dai valori del dominio e aggiunto proprio per lo scopo

Qualche esempio:

STUDENTI

| Matricola | Cognome | Nome | Data di nascita |
|-----------|---------|-------|-----------------|
| 276545 | Rossi | Maria | NULL |
| NULL | Neri | Anna | 23/04/1992 |
| NULL | Verdi | Fabio | 12/02/1992 |

ESAMI

| Studente | Voto | Corso |
|----------|------|-------|
| 276545 | 28 | 01 |
| NULL | 27 | NULL |
| 200768 | 24 | NULL |

CORSI

| Codice | Titolo | Docente |
|--------|---------|---------|
| 01 | Analisi | Giani |
| 03 | Chimica | NULL |
| NULL | Chimica | Belli |

Alcune considerazioni da fare sono:

- Il valore nullo sulla data di nascita è ammissibile, poiché si può pensare che non sia un dato essenziale in questo contesto
- Il valore nullo sul numero di matricola impedisce di stabilire correlazioni fra tuple di relazioni diverse.
- La presenza di più valori nulli in una tupla può rendere inutilizzabili le altre informazioni nella tupla.
- La presenza di più valori nulli in una relazione può causare problemi sull'identità delle tuple

I sistemi relazionali permettono di specificare per ciascun attributo di una relazione se esso può assumere il valore nullo oppure se per esso vale il vincolo not-null e quindi non può assumere il valore nullo.

Vincoli di integrità

Non è corretto dire che qualsiasi insieme di tuple (insieme di dati) sullo schema rappresenti informazioni corrette per l'applicazione, considerando l'esempio precedente:

STUDENTI

| Matricola | Cognome | Nome | Data di nascita |
|-----------|---------|-------|-----------------|
| 276545 | Rossi | Maria | NULL |
| NULL | Neri | Anna | 23/04/1992 |
| NULL | Verdi | Fabio | 12/02/1992 |

ESAMI

| Studente | Voto | Corso |
|----------|------|-------|
| 276545 | 28 | 01 |
| NULL | 27 | NULL |
| 200768 | 24 | NULL |

CORSI

| Codice | Titolo | Docente |
|--------|---------|---------|
| 01 | Analisi | Giani |
| 03 | Chimica | NULL |
| NULL | Chimica | Belli |

- Nella tupla ESAMI non si potrebbe avere un voto pari a 36
- Nella tupla STUDENTI non si potrebbero avere due studenti con stessa matricola ed etc.

Per poter evitare situazioni come queste viene introdotto il concetto di **vincolo di integrità** come proprietà che deve essere soddisfatta dalle istanze che rappresentano informazioni corrette per l'applicazione.

Ogni vincolo può essere visto come un predicato che assegna valori vero o falso se queste soddisfano le condizioni o no.

È possibile classificare i vincoli a seconda degli elementi di una base di dati che ne sono coinvolti, in particolare ci sono due categorie:

- **Vincolo intrarelazionale:** se il suo soddisfacimento è definito rispetto alle singole relazioni della base di dati (per esempio i casi descritti sopra), talvolta il coinvolgimento riprende direttamente le tuple separatamente chiamato vincolo di **tupla**
- **Vincolo interrelazionale:** se coinvolge più relazioni

Chiavi

Una **chiave** è un insieme di attributi utilizzato per identificare univocamente le tuple di una relazione, formalmente:

Def.: Data una relazione $R(X)$ con $K \subseteq X$, l'insieme K di attributi è superchiave per R se r non contiene due tuple distinte t_1 e t_2 con $t_1[K] = t_2[K]$.

In altre parole l'insieme di istanze di attributi utilizzato come superchiave non deve contenere elementi uguali in più di una tupla.

Un insieme K di attributi è chiave per r se è una superchiave minimale di r (cioè non esiste un'altra superchiave K' di r che sia contenuta in K come sottoinsieme proprio).

Poiché una relazione è un insieme di elementi distinti, si può concludere che per ogni relazione $r(X)$, l'insieme X di tutti gli attributi su cui è definita è chiaramente una superchiave per essa.

Ora i casi sono due:

- La **superchiave è anche chiave**, quindi si conferma l'esistenza della chiave stessa
- **Non è chiave**, perché contiene un'altra superchiave, quindi applicando ricorsivamente questo ragionamento si giunge, in un numero finito di passi (poiché l'insieme degli attributi è finito), ad una superchiave minimale.

Il fatto che su ogni schema di relazione si possa definire almeno una chiave garantisce l'**accessibilità** a tutti i valori di una base di dati e la loro univoca identificabilità e inoltre permette di stabilire delle corrispondenze "basate su valori" fra dati contenuti in relazioni diverse, che caratterizzano il modello relazionale.

Su una delle chiavi della relazione, detta **chiave primaria**, si vieta la presenza di valori nulli, per sceglierla si preferisce poi quella con il minor valore di attributi.

Per convenzione gli attributi della chiave primaria sono evidenziati tramite sottolineatura.

In quasi tutti i casi è possibile trovare una chiave fra gli attributi, quando ciò non accade si ricorre a un codice come attributo aggiuntivo per l'identificazione di una specifica tupla.

Vincoli di integrità referenziale

Un vincolo di integrità referenziale o chiave esterna fra un insieme di attributi $Y \subseteq X$ di una relazione $R_1(X)$ e un'altra relazione R_2 è soddisfatto se i valori su Y di ciascuna tupla dell'istanza su R_1 compaiono come valori della chiave primaria dell'istanza su R_2 .

Nel caso in cui la chiave di R_2 è unica e composta di un solo attributo B (e quindi l'insieme Y è a sua volta costituito da un solo attributo A) allora il vincolo di integrità referenziale fra l'attributo A di R_1 e la relazione R_2 è soddisfatto se, per ogni tupla t_1 in R_1 per cui $t_1[A]$ non è nullo, esiste una tupla t_2 in R_2 tale che $t_1[A] = t_2[B]$.

Nel caso più generale, ciascuno degli attributi in X deve corrispondere a un preciso attributo della chiave primaria K di R_2 .

Allo scopo è necessario specificare un ordinamento sia nell'insieme X sia in K , indicando gli attributi in ordine, $X = A_1, A_2 \dots A_p$ e $K = B_1, B_2 \dots B_p$, il vincolo è soddisfatto se per ogni tupla t_1 in R_1 senza nulli su X esiste una tupla t_2 in R_2 con $t_1[A_i] = t_2[B_i]$, per ogni i compreso fra 1 e p .

N.B: Per i DBMS che non consentono di indicare esplicitamente una chiave primaria, ma solo più chiavi, il vincolo di integrità referenziale deve indicare esplicitamente gli attributi che compongono la chiave cui si fa riferimento

Esempio:

| INFRAZIONI | Codice | Data | Agente | Articolo | Prov | Numero |
|------------|--------|------------|--------|----------|------|--------|
| | 143256 | 25/10/2012 | 567 | 44 | RM | 4E5432 |
| | 987554 | 26/10/2012 | 456 | 34 | RM | 4E5432 |
| | 987557 | 26/10/2012 | 456 | 34 | RM | 2F7643 |
| | 630876 | 15/10/2012 | 456 | 53 | MI | 2F7643 |
| | 539856 | 12/10/2012 | 567 | 44 | MI | 2F7643 |
| | | | | | | |

| AGENTI | Matricola | CF | Cognome | Nome |
|--------|-----------|-----------|---------|-------|
| | 567 | RSSM. . . | Rossi | Mario |
| | 456 | NREL. . . | Neri | Luigi |
| | 638 | NREP. . . | Neri | Piero |

| AUTO | Prov | Numero | Proprietario | Indirizzo |
|------|------|--------|--------------|-----------|
| | RM | 2F7643 | Verdi Piero | Via Tigli |
| | RM | 1A2396 | Verdi Piero | Via Tigli |
| | RM | 4E5432 | Bini Luca | Via Aceri |
| | MI | 2F7643 | Luci Gino | Via Aceri |

Figura 2.19 Una base di dati con vincoli di integrità referenziale.

- fra l'attributo **Agente** della relazione **INFRAZIONI** e la relazione **AGENTI**;
- fra gli attributi **Prov** e **Numero** di **INFRAZIONI** e la relazione **AUTO**, in cui l'ordine degli attributi nella chiave preveda prima **Prov** e poi **Numero**.

Esercizi sul Modello relazionale

Presenti nel PDF [2.1 - Esercizi - Modello Relazionale.pdf](#)