

Modellazione Logica

Modello Relazionale

di Roberta Molinari

Modello relazioni

Definizioni

Integrità sull'entità: non possono esserci record duplicati, quindi deve esistere una chiave primaria il cui valore è univoco e non NULL

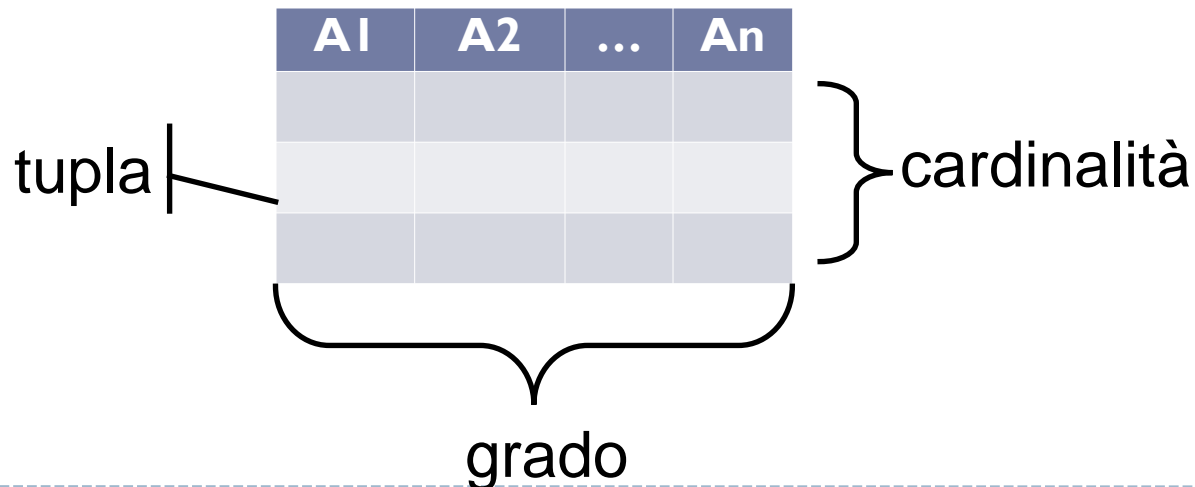
- ▶ Dati gli N insiemi A_1, A_2, \dots, A_N , si chiama **relazione** un sottoinsieme di tutte le N -ple (a_1, a_2, \dots, a_N) del prodotto cartesiano degli N insiemi. $R \subseteq A_1 \times A_2 \times \dots \times A_N$. Si rappresenta con una tabella.
- ▶ $N \geq 1$ è il **grado** della relazione. (n° di colonne)
- ▶ Gli insiemi A_i sono i **domini**, ad ogni dominio è associato un nome detto **attributo**.
- ▶ Gli elementi di R sono dette **tuple**, sono tutte diverse fra loro. Il loro insieme è variabile nel tempo: si possono aggiungere, modificare, cancellare.
- ▶ In ogni istante l'insieme delle tuple di R è detto **istanza della relazione**.
- ▶ Il numero di tuple presente in quel momento è la **cardinalità** della relazione. (n° di righe)
- ▶ L'insieme minimo di attributi che identificano univocamente le tuple si chiama **chiave primaria (pk)**, deve esistere e non può essere NULL (*integrità sull'entità*)

Modello relazionale

Rappresentazione

Una relazione viene a coincidere con una tabella, in cui le intestazioni di colonna sono gli attributi e le righe le tuple.

Relazione	Tabella o Tabella relazionale
Attributi	Colonne o Campi
Tupla	Riga o Record



Modello relazionale

Proprietà relazioni

Le tabelle relazionali hanno le seguenti proprietà:

1. I valori sono atomici, non ulteriormente scomponibili
2. Tutti i valori di una colonna appartengono al medesimo dominio
3. Ogni riga è univoca (differisce almeno per la pk)
4. La sequenza delle colonne non è significativa
5. La sequenza delle righe non è significativa
6. Ogni colonna deve avere un nome univoco

Modello relazionale

Derivazione del modello logico

- ▶ Il modello relazionale, su cui si basano i DB relazionali, serve per definire la struttura dei dati per poter definire le tabelle nel DBMS scelto.
- ▶ La struttura si deriva dal modello ER seguendo delle semplici regole
- ▶ La derivazione dal modello ER al modello logico si chiama **mapping**

Modello relazionale

Mapping

- ▶ Ogni *entità* diventa una relazione (tabella)
- ▶ Ogni *attributo* dell'entità diventa un attributo della relazione e ne eredita le caratteristiche (tipo, dim, obbligatorietà). Gli attributi *composti* vengono suddivisi in elementari. Quelli *multipli* sono suddivisi in campi se i valori contemporanei sono limitati (es Telefoni) o in una associazione 1:N o N:N con una nuova entità se sono molti (es. Sports)
- ▶ La *chiave primaria* nell'entità diventa chiave primaria della relazione

Modello relazionale

Mapping

- ▶ L'associazione $1:1$ diventa una sola relazione fusione delle due entità con l'unione degli attributi e l'eventuale integrazione con l'attributo della associazione (oppure si tratta come una $1:N$)
- ▶ L'associazione $1:N$ fa aggiungere un attributo nella entità di arrivo (con la N) con valore la chiave dell'entità di partenza (diventa **chiave esterna**). Si aggiungono anche gli eventuali attributi della relazione

Modello relazionale

Mapping

- ▶ L'associazione $N:N$ diventa una nuova relazione composta dalle chiavi delle due entità che diventano esterne e dagli eventuali attributi della relazione.

Bisogna definire la chiave primaria:

- se non possono esistere due record con la stessa combinazione di valori per le chiavi esterne la chiave primaria sarà l'unione delle stesse chiavi,
- altrimenti si crea un campo apposta.

Modello relazionale

Chiave esterna

Una **chiave esterna** o **foreign key (fk)** crea una gerarchia tra le istanze delle tabelle associate:

- ▶ l'istanza che contiene la chiave esterna si chiama **figlio**,
- ▶ quella che contiene la chiave primaria corrispondente si chiama **padre**.

Modello relazionale

Integrità referenziale

Nel modello relazionale deve essere garantita l'**integrità referenziale** il cui scopo è di impedire la presenza di record orfani e di mantenere sincronizzati i riferimenti, in modo che non vi siano record che facciano riferimento a record non più esistenti: deve esistere coerenza tra le tabelle associate, cioè ad ogni chiave esterna non NULL deve corrispondere una chiave primaria nella tabella associata.

Per garantire l'integrità referenziale il DBMS deve seguire alcune regole a seconda dell'operazione effettuata:

- ▶ **Regole di inserimento o inserzione**
- ▶ **Regole di cancellazione**
- ▶ **Regole di modifica**

Modello relazionale

Integrità referenziale

Inserzione :

- ▶ **dipendente:** si può inserire un'istanza figlio solo se l'istanza padre esiste
- ▶ **automatica:** se si inserisce un figlio prima del padre, si crea anche il padre con gli altri campi *null*
- ▶ **nulla:** se si inserisce un figlio prima del padre, si assegna *null* alla fk
- ▶ **di default:** se si inserisce un figlio prima del padre, si imposta la fk al valore predefinito

Modello relazionale

Integrità referenziale

Cancellazione:

- ▶ **con restrizione:** si può cancellare un'istanza del padre solo se non ha figli (NO ACTION, RESTRICT)
- ▶ **a cascata:** quando si elimina un padre, si eliminano anche tutti i figli (CASCADE)
- ▶ **nulla:** quando si elimina un padre, la fk nelle istanze dei figli viene impostata a *null* (SET NULL)
- ▶ **di default:** quando si elimina un padre, la fk nelle istanze dei figli viene impostata al valore predefinito (SET DEFAULT)

Modello relazionale

Integrità referenziale

Modifica della fk di un figlio senza avere una corrispondente pk in un padre:

- ▶ **dipendente:** non lo permette
- ▶ **automatica:** si crea un padre con quella pk e gli altri campi *null*
- ▶ **nulla:** si assegna *null* alla fk
- ▶ **di default:** si imposta la fk al valore predefinito

Modifica della pk di un padre:

- ▶ **con restrizione:** non lo permette (NO ACTION, RESTRICT)
- ▶ **a cascata:** si aggiornano tutte le fk di tutti gli eventuali figli (CASCADE)
- ▶ **nulla:** si assegna *null* alla fk (SET NULL)
- ▶ **di default:** si imposta la fk al valore predefinito (SET DEFAULT)

Modello relazionale

Mapping

- ▶ L'associazione *ricorsiva* 1:N o N:N si traduce normalmente, ma si tiene conto che le chiavi esterne sono tutte riferite alla stessa tabella

Persone (cod, Cognome, Nome, NatoIl)

èGenitoreDi (Genitore, Figlio)

- ▶ Le *associazioni multiple* (non dovrebbero esserci, ma solo binarie) generano una nuova tabella per l'associazione e si individuano le chiavi esterne nelle varie entità coinvolte.

Commessi (...)

Prodotti (...)

Clienti (...)

Vendite (codCommesso, codProd, codCli, Data, Qt)

Modello relazionale

Mapping

Le *associazioni IsA* possono venir tradotte in diversi modi a seconda dello schema ER di partenza:

1. **Accorpamento delle figlie nel padre (se non sovrapposta)**

Persona (CF, Cognome, Nome, NatoIl, maggiorenne▼, patente-)

Per cui sex specifica la sottoclasse e i campi delle figlie sono opzionali

2. **Accorpamento del padre nelle figlie se totale**

Maggiorenni (CF, Cognome, Nome, NatoIl, patente)

Minorenni (CF, Cognome, Nome, NatoIl)

3. **Sostituendo la generalizzazione con associazioni 1:1 se esclusiva o 1:N se sovrapposta**

Persona (CF, Cognome, Nome, NatoIl)

Studenti (CF, classe)

Lavoratori (CF, stipendio)

Modello relazionale

Mapping

- ▶ Ad ogni entità coinvolta in una *associazione HasA* corrisponde una nuova tabella. Se il numero di elementi è fisso, nella tabella dell'entità contenitore ci saranno delle chiavi esterne associate ad ogni entità contenuto, se no si seguono le regole come per le associazioni 1:N o N:N (solo per lasche)
- ▶ Le associazioni di aggregazione *lasca* avranno vincoli di integrità referenziale con chiavi esterna opzionali

Museo(codM, Nome, Indirizzo, Città, Paese)

Quadri(codQ, titolo, autore, Museo-)

Statue(codS, titolo, autore, Museo-)

- ▶ Le associazioni di aggregazione *stretta* avranno vincoli di integrità referenziale che richiedono l'obbligatorietà delle chiavi esterne (le modifiche o cancellazioni si ripercuotono sulle tabelle associate)

Auto(targa, Marca, Prezzo)

Bolli(cod, Importo, Validità, targa)

Assicurazioni(cod, Compagnia, Premio, Massimale,
Scadenza, targa)

Modello relazionale

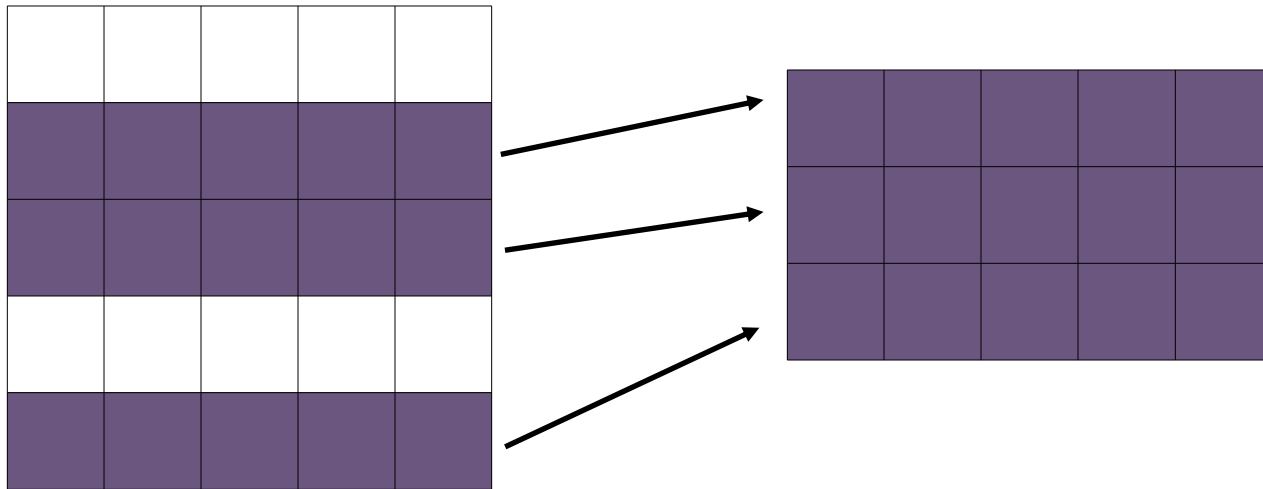
Operazioni relazionali

Selezione (select): genera una nuova relazione composta dalle sole tuple che soddisfano certe condizioni. Avrà lo stesso grado, ma cardinalità \leq

`SELECT * FROM R WHERE cond`

SELEZIONE di R PER cond

$\sigma_{\text{cond}} R$



Modello relazionale

Operazioni relazionali

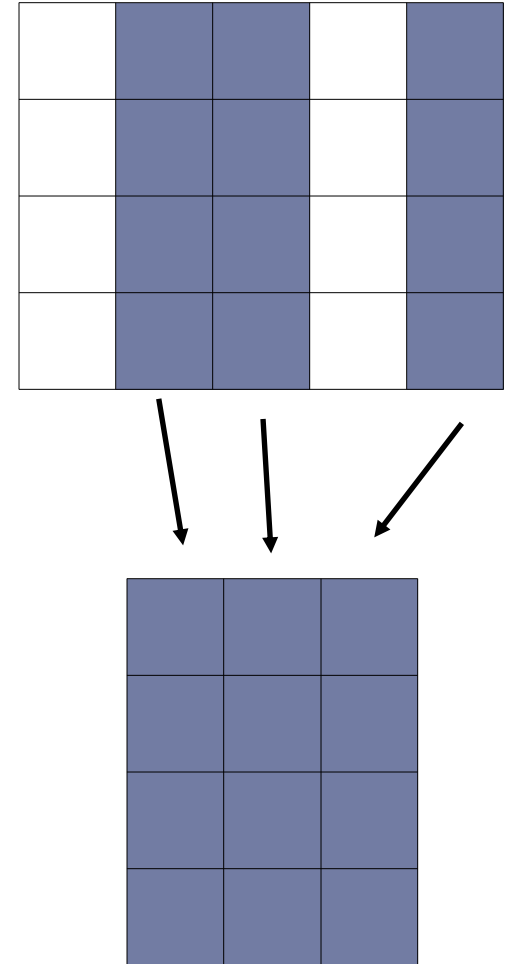
Proiezione (project): genera una nuova relazione composta dalle sole colonne relative ai campi di interesse. Se si generano 2 tuple uguali, ne rimane una sola. Avrà grado \leq , e cardinalità \leq (minore se tuple ripetute)

SELECT A_x, A_y, \dots, A_z FROM R

PROJECT R ON A_x, A_y, \dots, A_z

PROIEZIONE di R SU A_x, A_y, \dots, A_z

$\pi_{A_x, A_y, \dots, A_z} R$



Modello relazionale

Operazioni insiemistiche

Prodotto cartesiano (cross join) $R1 \times R2$: produce una relazione formata da tutte le tuple che è possibile ottenere combinando tutte le tuple di $R1$ con tutte le tuple di $R2$. La cardinalità è uguale al prodotto delle cardinalità, il grado dalla somma dei gradi

a_1	b_1	c_1
a_2	b_2	c_2
a_3	b_3	c_3

a_2	d_2
a_5	d_3



a_1	b_1	c_1	a_2	d_2
a_1	b_1	c_1	a_5	d_3
a_2	b_2	c_2	a_2	d_2
a_2	b_2	c_2	a_5	d_3
a_3	b_3	c_3	a_2	d_2
a_3	b_3	c_3	a_5	d_3

Modello relazionale

Operazioni relazionali

Congiunzione (inner o theta join): genera una nuova relazione a partire da 2 relazioni aventi un attributo comune, composta dalla fusione delle tuple con valori che soddisfano un criterio tra attributi.

Se il criterio è l'uguaglianza si parla di **equijoin**: in questo caso due colonne coincidono. Se si eliminano le colonne ridondanti si parla di **equi-join naturale**.

Se l'operatore di confronto tra gli attributi non è per forza l'=' si parla di **inner join**.

Prima è eseguito un prodotto cartesiano tra le tabelle e poi una selezione.

Modello relazionale

Operazioni relazionali

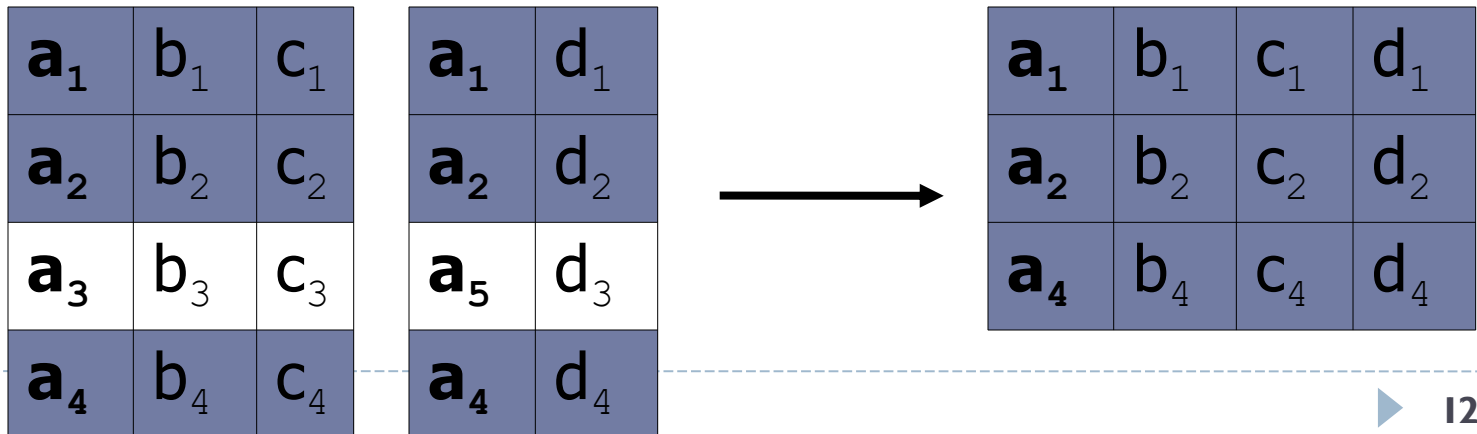
Congiunzione (join naturale):

$R.A_x \text{ JOIN } S.A_y$

CONGIUNZIONE di R SU A_x E di S SU A_y

$R_{A_x} \bowtie S_{A_y}$

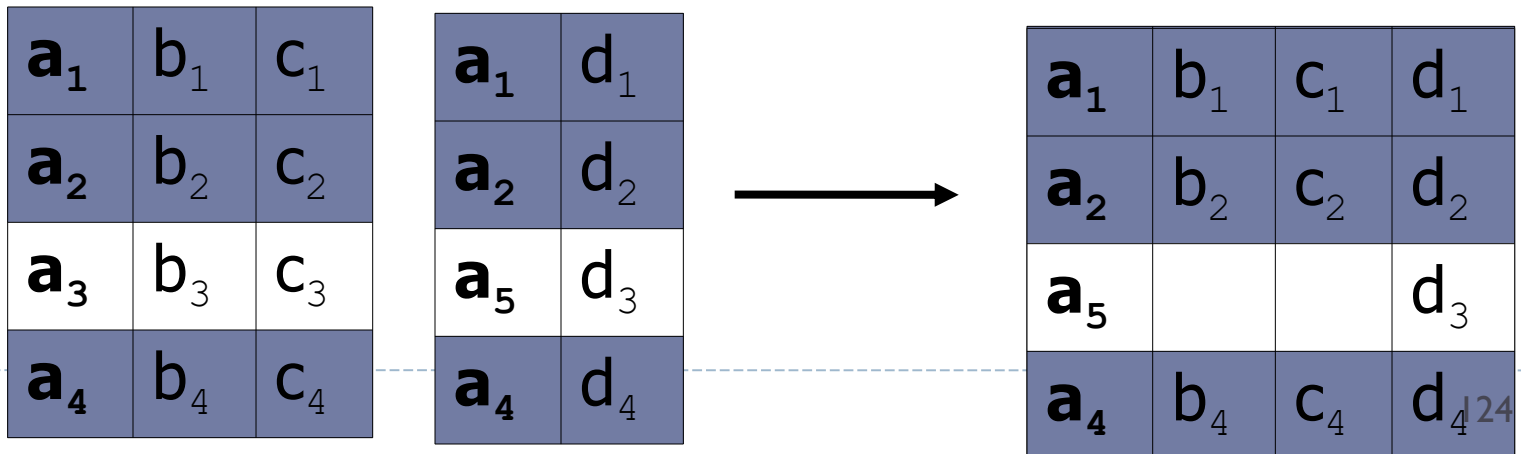
Avrà $\text{grado} = N1 + N2 - 1$, mentre la cardinalità non è prevedibile



Modello relazionale

Operazioni relazionali

Join esterno (outer join): genera una nuova relazione a partire da 2 relazioni aventi un attributo comune, composta da tutte le tuple della prima relazione più quelle ottenute dalla fusione delle tuple con valori uguali per quell'attributo (**left join**).
Oppure è formata da tutte le tuple della 2^a più quelle con i valori comuni (**right join**) o dall'unione di entrambi (**full join**)




Modello relazionale

Operazioni relazionali

Self Join (autocongiunzione): se esiste una associazione ricorsiva 1:N in una tabella, ogni tupla può essere in relazione con una tupla della tabella stessa. In questo caso ci sono due colonne con gli stessi domini (es. tabella persone, campo FiglioDi)

a₁	b ₁	c ₁	a₂
a₂	b ₂	c ₂	a₃
a₃	b ₃	c ₃	
a₄	b ₄	c ₄	a₂



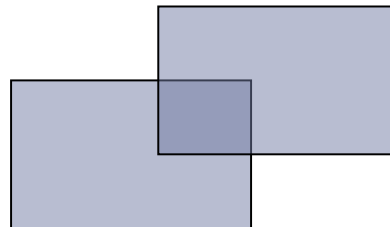
a₁	b ₁	c ₁	a₂	b ₂	c ₂
a₂	b ₂	c ₂	a₃	b ₃	c ₃
a₃	b ₃	c ₃			
a₄	b ₄	c ₄	a₂	b ₂	c ₂

Modello relazionale

Operazioni insiemistiche

Se le relazioni hanno lo stesso grado e gli stessi domini si possono usare i seguenti operatori insiemistici per ottenere nuove relazioni, che avranno sempre lo stesso grado e gli stessi domini delle relazioni di partenza.

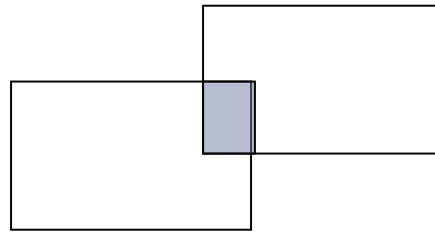
Unione $R \cup R2$: produce una relazione che contiene tutti gli elementi che appartengono all'una e/o all'altra (ci sono tutti gli elementi di entrambe, con quelli in comune ripetuti una sola volta)



Modello relazionale

Operazioni insiemistiche

Intersezione $R \cap R2$: produce una relazione che contiene le sole tuple che appartengono ad entrambe le relazioni



Differenza $R1 - R2$: produce una relazione che contiene le tuple che appartengono alla prima relazione, ma non alla seconda. Non è simmetrica $R1 - R2 \neq R2 - R1$

