

Modello Relazionale

Modello logico dei dati utilizzato nell'ambito delle basi di dati, introdotto da Codd nel 1970.

◇ **modello dei dati**

insieme di strutture dati e regole

logico \Rightarrow strutture disponibili su un DBMS

Modello relazionale : *relazione*

PERCORSO

PARTENZA	ARRIVO	ORAPARTENZA
Milano	Modena	1310
Milano	Roma	1550
Roma	Napoli	0400

◇ **schema**

rappresentazione di uno specifico dominio

◇ **istanza**

informazioni effettivamente memorizzate

◇ Altri modelli logici

- Modello Gerarchico (strutture ad *albero*)
- Modello Reticolare (strutture a *grafi*)

Architettura a tre livelli di un DBMS

Livello fisico (interno):

- organizzazione dei dati nelle memorie permanenti
- strutture dati ausiliarie per accedere ai dati

Livello logico (interno):

- struttura dei dati e relazioni fra essi
- nessun riferimento alla organizzazione fisica

Livello esterno (delle viste):

- struttura vista da un utente/applicazione

◇ **Indipendenza dei dati**

- indipendenza fisica
- indipendenza logica

Modello Relazionale: Caratteristiche

◇ Il concetto di relazione

- consente una descrizione astratta dei dati
⇒ indipendenza dei dati
- consente una formalizzazione semplice e precisa
⇒ Teoria relazionale dei dati
- è riconducibile al concetto semplice di *tabella*
⇒ facilmente comprensibile dall'utente finale

◇ linguaggio di interrogazione (SQL) *dichiarativo*

Modello Relazionale: Formalizzazione

Dominio: insieme di valori $D = \{v_1, v_2, \dots, v_k\}$

Tupla: Dati n domini D_1, D_2, \dots, D_n , una *ennupla* (tupla) ordinata t su D_1, D_2, \dots, D_n è definita come

$$t = (v_1, v_2, \dots, v_n), v_i \in D_i, \forall 1 \leq i \leq n$$

Prodotto Cartesiano: Il *prodotto cartesiano* di n domini D_1, D_2, \dots, D_n , indicato con $D_1 \times D_2 \times \dots \times D_n$, è l'insieme di tutte le tuple t su D_1, D_2, \dots, D_n .

Relazione: Una *relazione* r su n domini D_1, D_2, \dots, D_n , è un sottoinsieme di $D_1 \times D_2 \times \dots \times D_n$:

$$r \subseteq D_1 \times D_2 \times \dots \times D_n$$

- Il valore di n è il **Grado** della relazione
- Il numero di tuple è la **Cardinalità** della relazione

Modello Relazionale : Esempi

$$D_1 = \{a, b\}$$

$$D_2 = \{1, 2, 3\}$$

$$D_1 \times D_2 = \{(a, 1), (a, 2), (a, 3), (b, 1), (b, 2), (b, 3)\}$$

$$\text{Relazione su } D_1 \times D_2: r = \{(a, 1), (a, 2), (b, 1), (b, 2)\}$$

◇ Rappresentazione tabellare : r

a	1
a	2
b	1
b	2

◇ Un esempio concreto : *Orario di partenza dei treni*

Stringa \times Stringa \times Intero

r

Milano	Modena	1310
Milano	Roma	1550
Roma	Napoli	0400

Relazioni con attributi

◇ La relazione è un *insieme*, quindi

- le tuple sono distinte l'una dall'altra
⇒ differenza con la *tabella (multiset)*
- non è definito alcun ordinamento tra le tuple

◇ Ciascuna tupla è, al proprio interno, *ordinata*.

Attributo : Nome dato ad un dominio in una relazione

- **indipendenza** dall'ordinamento dei domini
- **significato** ai valori del dominio

◇ Un esempio concreto : *Orario di partenza dei treni*

	Stringa ×	Stringa ×	Intero
	↕	↕	↕
r	PARTENZA	ARRIVO	ORAPARTENZA
	Milano	Modena	1310
	Milano	Roma	1550
	Roma	Napoli	0400

Schema ed Istanza

Schema di relazione: $R(X)$

R : nome della relazione

$X = (A_1, A_2, \dots, A_n)$: insieme di nomi di attributi

Istanza di relazione: di uno schema $R(A_1, A_2, \dots, A_n)$
è una relazione r su (A_1, A_2, \dots, A_n) .

Schema di base di dati relazionale:

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

Istanza di base di dati relazionale:

Dato $\mathbf{R} = \{R_1(X_1), R_2(X_2), \dots, R_n(X_n)\}$,
una *istanza* su \mathbf{R} è un insieme di relazioni

$\mathbf{r} = \{r_1, r_2, \dots, r_n\}$, con r_i relazione su $R_i, \forall 1 \leq i \leq n$

Notazione: Sia $R(X)$ uno schema e t tupla su $R(X)$
Siano $Y, Z \subseteq X$ e $A \in X$

- l'insieme Y è denotato anche con $R.Y$
- l'insieme $\{A\}$ è denotato con A
- l'insieme $\{Y\} \cup \{Z\}$ è denotato con XY denota ,
- $t[A]$, oppure $t.A$, denota il valore di t su A
- $t[Y]$ denota la sottotupla di t ottenuta considerando i valori degli attributi in Y

Esempio

Schema di base di dati: DB_UNIVERSITÀ = {

STUDENTE (MATR, NOME, CITTÀ, ACORSO)

CORSO (CODCOR, NOME, CODDOC)

DOCENTE (CODDOC, CF, CITTÀ) }

Istanza di base di dati:

STUDENTE

MATR	NOME	CITTÀ	ACORSO
M1	Marco Quaranta	SA	1
M2	Giacomo Tedesco	PA	2
M3	Maria Mei	BO	1
M4	Ugo Rossi	MO	2
M6	Agata Verdi	MI	1

CORSO

CODCOR	NOME	CODDOC
C1	Fisica 1	D1
C2	Analisi Matematica 1	D2
C3	Fisica 2	D1
C4	Analisi Matematica 2	D2

DOCENTE

CODDOC	CF	CITTÀ
D1	CF1	MO
D2	CF2	BO
D3	CF3	MO
D4	CF4	FI

Chiavi

- ◇ Informalmente, per *chiave* di una relazione si intende un sottoinsieme dei suoi attributi che identifica univocamente ogni tupla della relazione stessa.

- ◇ Dato $R(X)$, $K \subseteq X$ è detto *chiave* di $R(X)$ se e solo se per ogni relazione r su $R(X)$ valgono le proprietà:
 1. **(Univocità)** $\forall t_1, t_2 \in r, t_1[K] = t_2[K] \Rightarrow t_1 \equiv t_2$
 2. **(Minimalità)** $\forall A_i \in K, K - A_i$ non verifica 1.

- ◇ Dato $R(X)$ ed una chiave K di $R(X)$, $Y \subseteq X$ tale che $Y \supset K$ è detto **superchiave** di $R(X)$.

- ◇ Per ogni $R(X)$, X è una superchiave.
Quindi, ogni $R(X)$ ha almeno una chiave.

Chiavi candidate, primaria, alternative

◇ $R(X)$ può avere più chiavi dette **candidate**

Tra le chiavi candidate una a scelta è detta **primaria**

Le rimanenti chiavi vengono dette **alternative**

◇ **Notazione**

chiave primaria: $R(\underline{K_1}, \underline{K_2}, \dots, \underline{K_m}, A_2, \dots, A_n)$

chiave alternativa: $R(X)$

AK: K_1, \dots, K_m

Esempio:

DOCENTE (CODDOC, CF, CITTÀ)

AK: CF

Valori nulli

Informazione incompleta : tupla il cui valore di uno o più attributi non è disponibile.

- ◇ Ad esempio, in `DOCENTE (CODDOC , CF , NOME , CITTÀ)`
 - la `CITTÀ` del docente è sconosciuta
 - il `CF` non è previsto per docenti di determinati paesi
 - al docente non è ancora stato assegnato un `CODICE`
 - ...

- ◇ Ogni dominio di relazione viene esteso con un particolare valore, detto *valore nullo* e denotato con `null`, che rappresenta assenza di informazione.

- ◇ Nei sistemi relazionali
 - ⇒ è possibile specificare se un attributo può o meno assumere il valore `null`.

 - ⇒ Logica a tre valori in SQL

Vincoli di Integrità

I *vincoli di integrità* stabiliscono condizioni di correttezza delle informazioni nella base di dati.

Vincolo intrarelazionale :

coinvolge singole relazioni della base di dati

◇ Vincolo di chiave

Una chiave K è un vincolo di integrità: essa stabilisce l'univocità dei valori assunti dagli attributi di K .

– *vincolo di tupla*

coinvolge una singola tupla di una relazione

Vincolo interrelazionale :

coinvolge più relazioni della base di dati

◇ Vincolo di Integrità Referenziale

Vincolo di Entity Integrity

- ◇ In presenza di valori nulli, non sarebbe quindi possibile controllare l'univocità dei valori assunti dagli attributi di una chiave. Per questo motivo viene imposto il seguente vincolo.

Tale vincolo stabilisce che gli attributi che costituiscono la chiave primaria di una relazione non possono assumere valore nullo.

- ◇ Formalmente, un'istanza r di uno schema di relazione R con chiave primaria K_1, K_2, \dots, K_m , soddisfa il vincolo di Entity integrity se e solo se

$$\forall t \in r, t[K_i] \neq \text{null}, \forall 1 \leq i \leq m$$

Vincolo di Integrità Referenziale

- ◇ I riferimenti tra le tuple delle relazioni vengono stabiliti tramite i valori assunti dagli attributi nelle tuple.

L'integrità referenziale assicura che quando in una tupla si utilizza il valore di un attributo per riferirsi ad un'altra tupla, quest'ultima sia una tupla esistente.

CORSO

CODCOR	NOME	CODDOC
C1	Fisica 1	D1
C2	Analisi Matematica 1	D2
C3	Fisica 2	D1
C4	Analisi Matematica 2	D2
C5	Meccanica Razionale	null

DOCENTE

CODDOC	CF	CITTÀ
D1	CF1	MO
D2	CF2	BO
D3	CF3	MO
D4	CF4	FI

Vincolo di Integrità Referenziale

◇ Dichiarazione *vincolo di integrità referenziale* in **R**:

Foreign Key o Chiave Esterna: insieme di attributi $FK = \{FK_1, FK_2, \dots, FK_n\}$ di uno schema $R_1 \in \mathbf{R}$

Chiave della Relazione riferita: schema di relazione $R_2 \in \mathbf{R}$, non necessariamente distinto da R_1 , con una chiave $K = \{K_1, K_2, \dots, K_m\}$, con $m = n$.

◇ Un'istanza $r = \{r_1, r_2, \dots\}$ su **R** soddisfa il vincolo se e solo se i valori su FK di ciascuna tupla di r_1 sono valori su K di una r_2 , o sono valori nulli:

$$\forall 1 \leq i \leq n, \forall t_1 \in r_1 \quad (\exists t_2 \in r_2 : t_1[FK_i] = t_2[K_i]) \vee (t_1[FK_i] = \text{null})$$

Notazione:

$R_1(X)$

FK: FK_1, \dots, FK_n REFERENCES $R_2(K_1, \dots, K_n)$

Esempio:

CORSO (CODCOR, NOME, CODDOC)

FK: CODDOC REFERENCES DOCENTE

Istanza Legale di Base di Dati

◇ Dato uno schema di basi di dati

$\mathbf{R} = \{R_1(X_1), R_2(X_2), \dots, R_n(X_n)\}$, un'istanza $\mathbf{r} = \{r_1, r_2, \dots, r_n\}$ su \mathbf{R} tale che

- ciascuna relazione r_i soddisfa il vincolo di entity integrity
- \mathbf{r} soddisfa tutti i vincoli di integrità referenziale imposti su \mathbf{R}

verrà detta istanza **legale** della base di dati \mathbf{R} .

Il linguaggio SQL

- ◇ Il linguaggio SQL (*Structured Query Language*) è il linguaggio standard per la definizione e manipolazione delle basi di dati relazionali.

Basi di dati relazionali

Schema : Corso(CC , CNome , CD)
 Docente(CD , Cognome , Città)

Istanza :

Corso		
CC	CNome	CD
C1	Fisical1	D1
C2	Analisi1	D2
C3	Fisica2	D1
C4	Analisi2	D3

Docente		
CD	Cognome	Città
D1	Rossi	MO
D2	Pastore	BO
D3	Carboni	FI

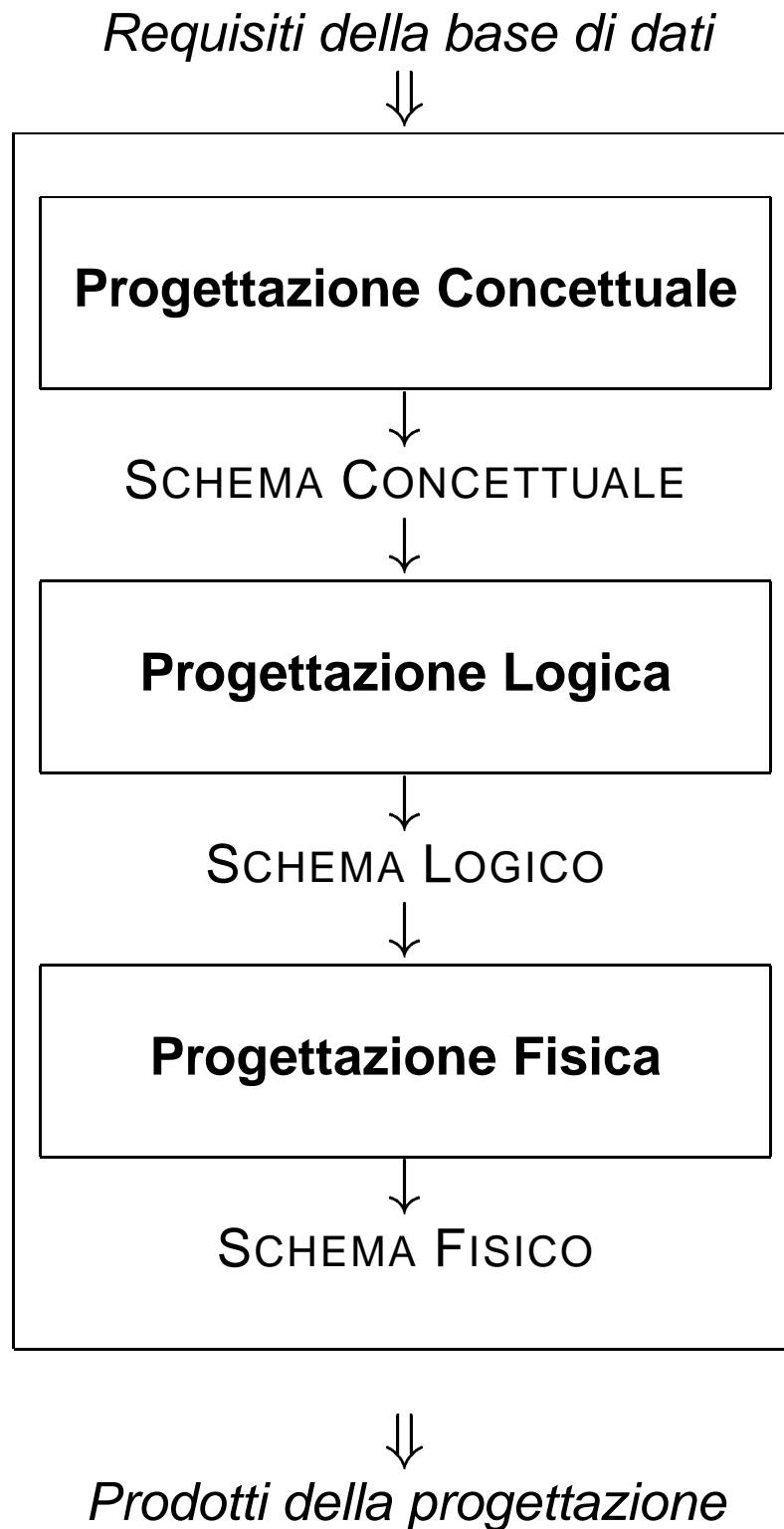
linguaggio di manipolazione dei dati - *DML*

interrogazione e aggiornamento dell'istanza

linguaggio di definizione dei dati - *DDL*

definizione schema, autorizzazioni per l'accesso, ...

Progettazione di una base di dati



Modello Relazionale: Cronologia

- 1970** “A Relational Model of Data for Large Shared Data Banks”, di E.F. Codd
- 1974** prima versione del linguaggio SQL
- 1975-79** sviluppo del prototipo System R
- 1980** versione commerciale di SystemR, e Oracle
- 1982** IBM Database2 (DB2)
- 1986-89** standardizzazione ISO-ANSI SQL-89
- 1992** ISO-ANSI SQL2
- 1999** ISO-ANSI SQL3