



# Capitolo 2 (Modello relazionale)

## Modelli logici dei dati

Un **modello dei dati** è un insieme di concetti utilizzati per organizzare i dati di interesse e descriverne la struttura in modo che essa risulti comprensibile a un elaboratore

Fornisce **meccanismi di strutturazione** che permettono di definire nuovi tipi sulla base di tipi predefiniti

I modelli dei dati sono detti **logici**, per sottolineare il fatto che le strutture utilizzate da questi modelli riflettono una particolare organizzazione

Sono stati introdotti altri modelli dei dati, detti **concettuali**, utilizzati per determinare i dati in maniera completamente indipendente dalla scelta del modello logico

Tre modelli logici tradizionali:

- Gerarchico
- Reticolare
- Relazionale
- A oggetti

Il modello gerarchico e il modello reticolare utilizzano riferimenti espliciti (puntatori) fra record

Il modello relazionale “è basato su valori”

- Anche i riferimenti fra dati in strutture (relazioni) diverse sono rappresentati per mezzo di valori

## Il modello relazionale

Proposto da E.F. Codd nel 1970 per favorire l'indipendenza dei dati

Disponibile in DBMS reali nel 1981 (non era facile implementare l'indipendenza con efficienza e affidabilità)

Il modello relazionale si basa su due concetti:

- **Relazione** che proviene dalla matematica, cioè dalla teoria degli insiemi
- **Tabella** che è un concetto semplice ed intuitivo

Inoltre esso risponde al requisito dell'indipendenza dei dati che prevede una distinzione, nella descrizione dei dati, che prevede una distinzione fra il livello *fisico* e il livello *logico*

## Relazioni



Siano  $D_1$  e  $D_2$  due insiemi, il *prodotto cartesiano* tra i due è definito come l'insieme delle coppie ordinate  $(v_1, v_2)$  tali che  $v_1$  è un elemento di  $D_1$  e  $v_2$  è un elemento di  $D_2$

- $D_1 \times D_2 = \{(v_1, v_2) : v_1 \in D_1 \wedge v_2 \in D_2\}$



Una *relazione matematica* sugli insiemi  $D_1$  e  $D_2$  (detti **domini** della relazione) è un sottoinsieme di  $D_1 \times D_2$

Le relazioni possono essere rappresentate graficamente tramite tabelle

Possiamo generalizzare queste definizioni per un numero  $n$  di insiemi:

Dati  $n > 0$  insiemi  $D_1, D_2, \dots, D_n$ , non necessariamente distinti, il **prodotto cartesiano**  $D_1 \times D_2 \times \dots \times D_n$  è costituito dalle  $n$ -uple  $(v_1, v_2, \dots, v_n)$  tali che  $v_i \in D_i$ , con  $1 \leq i \leq n$

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 viene detto **grado** del prodotto cartesiano e della relazione

Il numero di elementi (cioè di  $n$ -uple) della relazione viene chiamato **cardinalità** della relazione

@rosacarota e @redyz13

Esempio:

- $D_1 = \{a, b\}$
- $D_2 = \{x, y, z\}$

Prodotto cartesiano  $D_1 \times D_2$

a	x
a	y
a	z
b	x
b	y
b	z

Una relazione  $r \subseteq D_1 \times D_2$

a	x
a	z
b	y

## Relazioni con attributi

Una relazione matematica è un insieme di n-uple **ordinate**  $(v_1, v_2, \dots, v_n)$  con  $v_1 \in D_1, v_2 \in D_2, \dots, v_n \in D_n$

Essendo la relazione matematica un insieme, esso gode delle seguenti proprietà

- Non c'è ordinamento fra le n-uple; nelle tabelle c'è un ordine "occasionale". Due tabelle con le stesse righe, ma in ordine diverso, rappresentano la stessa relazione
- Le n-uple sono distinte l'una dall'altra: tra gli elementi di un insieme non ne possono essere presenti due uguali. una tabella rappresenta una relazione solo se le sue righe sono l'una diversa dall'altra
- Ciascuna n-upla è ordinata al proprio interno: l' $i$ -esimo valore proviene dall' $i$ -esimo dominio (è definito un ordinamento fra i domini che è significativo ai fini dell'interpretazione dei dati nelle relazioni)

Ciascun dominio ha ruoli diversi, distinguibili tramite la posizione: si ha quindi una **struttura posizionale**

## Struttura non posizionale

Le relazioni nelle basi di dati hanno una struttura che è facilmente riconducibile a quella dei record: esse, infatti, sono insieme di record omogenei, cioè definiti sugli stessi campi

Nel caso dei record, a ogni campo è associato un nome: associamo a ciascun occorrenza di dominio nella relazione un nome, detto **attributo**, che descrive il “ruolo” che gioca il dominio.

Nella rappresentazione tabellare utilizziamo gli attributi come intestazioni delle colonne delle tabelle; dovendo identificare univocamente le componenti, gli attributi di una relazione devono essere diversi l'uno dall'altro

Con l'introduzione degli attributi, vediamo che l'ordinamento di questi risulta irrilevante: non parliamo più, quindi, di primo dominio, secondo dominio e così via, è necessario solo far riferimento agli attributi (abbiamo una **struttura non posizionale**)

### Formalizziamo i concetti

Indichiamo con  $\mathcal{D}$  l'insieme dei domini e specifichiamo la corrispondenza tra attributi e domini per mezzo di una funzione  $dom : X \rightarrow \mathcal{D}$  che associa a ciascun attributo  $A \in X$  un dominio  $dom(A) \in \mathcal{D}$

Poi, diciamo che una *tupla* su un insieme di attributi  $X$  è una funzione  $t$  che associa a ciascun attributo  $A \in X$  un valore del dominio  $dom(A)$

Possiamo dare una nuova definizione di relazione:



Una **relazione** su  $X$  è un insieme di tuple su  $X$

La differenza di questa definizione e quella tradizionale di relazione matematica è nella definizione di tupla:

- Nella *relazione matematica* abbiamo  $n$ -uple i cui elementi sono individuati per posizione
- Nelle *tuple* della nuova definizione gli elementi sono individuati per mezzo degli attributi, cioè con una tecnica non posizionale

Notazione:

Se  $t$  è una tupla su  $X$  e  $A \in X$ , allora  $t[A]$  indica il valore di  $t$  su  $A$

Esempio:

$t[\text{Fuori}] = \text{Lazio}$

Casa	Fuori	RetiCasa	RetiFuori
Juve	Lazio	3	1
Lazio	Milan	2	0
Juve	Roma	0	2
Roma	Milan	0	1

$t[\text{Casa}, \text{Fuori}] \rightarrow$  tupla su due attributi

## Tabelle e Relazioni

Una tabella rappresenta una relazione se:

- I valori di ogni colonna sono fra loro omogenei (dello stesso tipo)
- Le righe sono diverse fra loro (altrimenti non è un insieme)
- Le intestazioni delle colonne sono diverse tra loro

In una tabella che rappresenta una relazione:

- L'ordinamento tra le righe è irrilevante
- L'ordinamento tra le colonne è irrilevante

Le righe della tabelle sono le ennuple mentre le colonne i domini

Una delle caratteristiche fondamentali è che il modello relazionale è **basato su valori**

- I riferimenti fra dati in relazioni diverse (es. i dati di uno studente e quello dei suoi esami) sono rappresentati per mezzo di valori dei domini che compaiono nelle ennuple (tuple)

Altri modelli logici, come il reticolare o il gerarchico, realizzano le corrispondenze tramite puntatori

Vantaggi del modello relazionale:

- Richiede di rappresentare solo ciò che è utile dal punto di vista dell'applicazione: i puntatori sono legati all'aspetto realizzativo
- La rappresentazione logica non fa alcun riferimento alla rappresentazione fisica: si ha, quindi, **l'indipendenza dalle strutture fisiche** (le strutture fisiche possono cambiare nel tempo)
  - I dati potrebbero essere rappresentati tramite puntatori a livello fisico, ma essi non sono visibili a livello logico
  - I puntatori sono direzionali, i valori no
- Risulta semplice trasferire i dati da un contesto all'altro (per esempio da un calcolatore ad un altro)
- L'utente finale vede gli stessi dati dei programmatori

## Definizioni relative al modello relazionale

### Schema di Relazione e di Base di Dati

Nel modello relazionale i concetti di schema (intensione) ed istanza (estensione) di base di dati si traducono in:

- **Schema di relazione:**

- Costituito da un simbolo  $R$ , detto **nome della relazione**, e da un **insieme di (nomi di) attributi**  $X = \{A_1, A_2, \dots, A_n\}$  di solito indicato con  $R(X)$ . Ad ogni attributo è associato un dominio (visto prima)

- **Schema di base di dati:**

- Un insieme di schemi di relazione con nomi diversi:

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

Dove  $X_1, \dots, X_n$  sono insiemi di attributi

I nomi di relazioni hanno come scopo principale quello di distinguere le varie relazioni nella base di dati

### Ennuple (tuple) di relazioni

Una **ennupla** (su un insieme di attributi  $X$ ) è una funzione che associa a ciascun attributo  $A$  in  $X$  un valore del dominio di  $A$

$t[A]$  denota il valore della ennupla  $t$  sull'attributo  $A$

Le ennuple sono anche dette **tuple**

(come sopra)

### Istanze di Relazione e di BD

- **Istanza di relazione** (o semplicemente *relazione*) su uno schema  $R(X)$ 
  - Insieme  $r$  di ennuple su  $X$



- Talvolta si usa la notazione  $r(X)$  per indicare una relazione sull'insieme di attributi  $X$  (descrivendo contemporaneamente schema e istanza)
- **Istanza di base di dati** (o semplicemente *base di dati*) su uno schema  $\mathbf{R} = \{R_1(X_1), \dots, R_n(X_n)\}$ :
  - Insieme di relazioni  $\mathbf{r} = \{r_1, \dots, r_n\}$  dove ogni  $r_i$ , per  $1 \leq i \leq n$ , è una relazione sullo schema  $R_i(X_i)$

## Informazione incompleta

Il modello relazionale impone ai dati una struttura rigida: le informazioni devono essere rappresentate per mezzo di tuple di dati omogenee

In ogni relazione solo alcuni formati di ennuple sono ammessi: quelli che corrispondono agli schemi di relazione

Per la rappresentazione della “non disponibilità” di valori si potrebbero usare valori del dominio stesso

Questa scelta richiede:

- L'esistenza di un valore del dominio mai utilizzato per valori significativi
- Potrebbero non esistere valori disponibili per questo scopo

Una tecnica rudimentale, ma efficace è l'introduzione della possibilità che una tupla possa assumere un valore speciale:



**Valore nullo:** Denota l'assenza di un valore del dominio (e non è un valore del dominio)

$t[A]$ , per ogni attributo  $A$ , diventa un valore del dominio  $dom(A)$  oppure il valore nullo **NULL**

Si possono (e si devono) imporre restrizioni sulla presenza di valori nulli (possono essere ammessi in alcuni attributi e non su altri)

## Tipi di valore nullo

Esistono varie interpretazioni di un valore nullo, ad esempio:

- Valore **sconosciuto**
- Valore **inesistente**

I DBMS non distinguono i diversi tipi di valore nullo

## Vincoli di integrità

Esistono istanze di basi di dati che, pur se sintatticamente corrette, non rappresentano informazioni possibili per l'applicazione di interesse



**Vincolo di integrità:** Una proprietà che deve essere soddisfatta dalle istanze che rappresentano informazioni corrette per l'applicazione

Un vincolo è una funzione booleana (un **predicato**): associa ad ogni istanza il valore **vero** o **falso**

Se il predicato ha valore vero allora diciamo che l'istanza **soddisfa** il vincolo

A uno schema di base di dati associamo un insieme di vincoli e consideriamo **corrette** (o **lecite**, **ammissibili**) le istanze che soddisfano tutti i vincoli

Garantiscono:

- Una descrizione più accurata della realtà
- Un contributo alla “qualità dei dati”
- Utilità nella progettazione
- Utilizzo degli stessi dai DBMS nell’esecuzione delle interrogazioni

Non tutte le proprietà di interesse sono tuttavia rappresentabili per mezzo di vincoli formulabili in modo esplicito

## Tipi di vincoli

Vincoli **intrarelazionali** (valgono per una singola relazione):

- Il suo soddisfacimento è definito rispetto a singole relazioni della base di dati

Essi si dividono in:

- **Vincoli di tupla:** un vincolo che può essere valutato su ciascuna tupla indipendentemente dalle altre
- **Vincoli su valori o domini:** è un caso più specifico del vincolo di enupla, è definito con riferimento a singoli valori . Impone una restrizione sul dominio dell’attributo

Vincoli **interrelazionali** (riguardano più relazioni contemporaneamente):

- Quelli di nostro interesse sono i **vincoli di integrità referenziale**

## Vincoli di ennupla



**Vincoli di ennupla:** Esprimono condizioni sui valori di ciascuna ennupla, in modo indipendente dalle altre ennuple

La sintassi permette di definire espressioni booleane (con AND, OR e NOT) di atomi che confrontano (con operatori di disuguaglianza, uguaglianza e ordinamento) valori di attributo o espressioni aritmetiche su di essi

Es:

- $(\text{Voto} \geq 18) \text{ AND } (\text{Voto} \leq 30)$
- $(\text{Voto} = 30) \text{ OR NOT } (\text{Lode} = \text{"e lode"})$

Caso particolare:

- Vincoli di dominio che coinvolgono un solo attributo

## Chiave

I vincoli a chiave sono i più importanti del modello relazionale



Una **chiave** è un insieme di attributi che identificano univocamente le ennuple di una relazione

Formalmente:

- Un insieme  $K$  di attributi è **superchiave** per una relazione  $r$  se  $r$  non contiene due ennuple distinte  $t_1$  e  $t_2$  con  $t_1[K] = t_2[K]$  (stessa combinazione di valori)

- $K$  è **chiave candidata** (sul libro solo **chiave**) per  $r$  se è una **superchiave minimale** per  $r$  (cioè non esiste un'altra superchiave  $K'$  di  $r$  che sia contenuta in  $K$  come sottoinsieme proprio (praticamente se  $K$  non contiene un'altra superchiave))

Esempio:

Matricola	Cognome	Nome	Corso	Nascita
27655	Rossi	Mario	Ing Inf	5/12/78
78763	Rossi	Mario	Ing Inf	3/11/76
65432	Neri	Piero	Ing Mecc	10/7/79
87654	Neri	Mario	Ing Inf	3/11/76
67653	Rossi	Piero	Ing Mecc	5/12/78

l'insieme {Matricola} è una superchiave; è anche superchiave minimale in quanto contiene un solo attributo (più l'insieme vuoto, ma esso non è in grado di identificare tuple), quindi non ha sottoinsiemi di conseguenza non ha altre superchiavi. Di conseguenza, {Matricola} è una chiave candidata

Matricola	Cognome	Nome	Corso	Nascita
27655	Rossi	Mario	Ing Inf	5/12/78
78763	Rossi	Mario	Ing Inf	3/11/76
65432	Neri	Piero	Ing Mecc	10/7/79
87654	Neri	Mario	Ing Inf	3/11/76
67653	Rossi	Piero	Ing Mecc	5/12/78

Cognome, Nome, Nascita è un'altra chiave:

- Anche superchiave
- Anche minimale

Matricola	Cognome	Nome	Corso	Nascita
27655	Rossi	Mario	Ing Inf	5/12/78
78763	Rossi	Mario	Ing Civile	3/11/76
65432	Neri	Piero	Ing Mecc	10/7/79
87654	Neri	Mario	Ing Inf	3/11/76
67653	Rossi	Piero	Ing Mecc	5/12/78

Non ci sono ennuple uguali su Cognome e Corso:

- Cognome e Corso formano una chiave?
  - Dipende se è sempre vero (spoiler non lo è, lo è solo per questo mini schema)

@rosacarota e @redyz13

## Vincoli, schemi e istanze

I vincoli corrispondono a proprietà del mondo reale modellato dalla base di dati

Interessano a livello di schema (con riferimento cioè a tutte le istanze)

Ad uno schema associamo un insieme di vincoli e consideriamo **corrette** (valide, ammissibili) solo le istanze che soddisfano tutti i vincoli

## Chiavi

### Esistenza delle chiavi

Una relazione non può contenere ennuple distinte ma uguali (ennuple diverse ma con valori di attributi uguali)

#### Esistenza e ricerca delle chiavi

Una relazione (si intende istanza di relazione) è un insieme, quindi, ha elementi diversi tra loro; di conseguenza, per ogni relazione  $r(X)$ , l'insieme  $X$  e di tutti gli attributi su cui è definita la relazione  $r$  è sicuramente una superchiave

In questo modo, abbiamo due casi:

- Questo insieme è anche chiave, quindi è confermato che l'esistenza della chiave stessa
- Questo insieme non è chiave, perché contiene un'altra chiave al suo interno

Procediamo quindi ricorsivamente : il processo terminerà in un numero finito di passi (poiché l'insieme degli attributi su cui è definita  $r$  è un insieme finito)

Possiamo quindi affermare che ogni relazione ha una chiave

Lo stesso ragionamento ricorsivo può essere svolto a livello dello schema di relazione: l'insieme di tutti gli attributi è superchiave per ciascuna relazione, quindi lo è per ciascuna relazione lecita; la ricerca prosegue, poi, ricorsivamente

Può capitare che l'insieme di tutti gli attributi è anche una (immagino ci vada un (super))chiave minimale (quindi una chiave candidata)

### Importanza delle chiavi

L'esistenza delle chiavi garantisce l'accessibilità a ciascun dato della base di dati

Le chiavi permettono di correlare i dati in relazioni diverse (modello basato su valori, tramite i valori delle chiavi realizziamo i riferimenti)

## Chiave primaria

In presenza di valori nulli i valori della chiave non permettono:

- Di identificare le ennuple
- Di realizzare facilmente i riferimenti da altre relazioni

Bisogna, quindi, porre dei limiti alla presenza di valori nulli: su una delle chiavi si vieta la presenza di valori nulli



**Chiave primaria:** Chiave su cui non sono assolutamente ammessi valori nulli

Sulle altre chiavi, in genere, i valori nulli sono ammessi (salvo necessità)

Gli attributi che costituiscono la chiave primaria vengono spesso evidenziati attraverso sottolineatura

Il DBMS in automatico non permette che vengano inseriti valori nulli nelle chiavi e inoltre non permette modifiche che portano ad un valore nullo nelle chiavi

## Integrità referenziale (vincolo interrelazionale)



I vincoli di integrità referenziale sono la classe più importante di vincoli interrelazionali

In sostanza, informazioni in relazioni diverse sono correlate attraverso valori comuni

Le correlazioni devono essere “coerenti”

L'integrità referenziale fa sì che il DBMS traduca la specifica, la salvi nel catalogo come vincolo e se si rende conto che porta ad un riferimento che non funziona impedirà che venga tranciato il vincolo

## Vincolo di integrità referenziale



**Vincolo di integrità referenziale (foreign key o referential integrity constraint)** fra un insieme di attributi  $X$  di una relazione  $R_1$  e un'altra relazione  $R_2$  è soddisfatto se i valori su  $X$  di ciascuna tupla dell'istanza di  $R_1$  compaiono come valori della chiave primaria dell'istanza di  $R_2$

@rosacarota e @redyz13

Insieme di attributi di una relazione che coincide con la chiave primaria di un'altra relazione

Primo caso: la chiave di  $R_2$  è unica e composta di un solo attributo  $B$  (e quindi l'insieme  $X$  è costituito da un solo attributo  $A$ )

Il vincolo di integrità referenziale 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  $t_1[A] = t_2[B]$

Caso più generale: dobbiamo prestare attenzione al fatto che ciascuno degli attributi in  $X$  deve corrispondere a un preciso attributo della chiave primaria  $K$  di  $R_2$

E' necessario specificare un ordinamento sia nell'insieme  $X$  sia in  $K$ :  $X = A_1A_2...A_p$  e  $K = B_1B_2...B_p$ .

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 tra 1 e  $p$

Se sono presenti più chiavi, ovviamente il riferimento va fatto sulla chiave primaria

## Chiave esterna

In altre parole una chiave esterna è un gruppo di attributi di una relazione che costituisce anche la chiave primaria di un'altra relazione dello schema di database

## Considerazioni

I vincoli di integrità referenziale giocano, quindi, un ruolo fondamentale nel concetto di "modello basato su valori"

In presenza di valori nulli i vincoli possono essere resi meno restrittivi

Esempio: si potrebbe mettere NULL alla matricola di un vigile (poiché andato in pensione), ma non ad una targa di un'auto

Sono possibili meccanismi per il supporto alla loro gestione ("azioni" compensative a seguito di violazioni) (consenti l'operazione ma effettua questi cambiamenti nel caso)

Attenzione ai vincoli su più attributi

## Azioni compensative del DBMS

Esempio:

- Viene eliminata una ennupla causando una violazione

Comportamento “standard” del DBMS:

- Rifiuto dell’operazione

Azioni compensative del DBMS:

- Eliminazione in cascata
- Introduzione di valori nulli

@rosacarota e @redyz13