



# Basi di dati



## Il Modello Relazionale dei Dati

# Il Modello Relazionale dei Dati

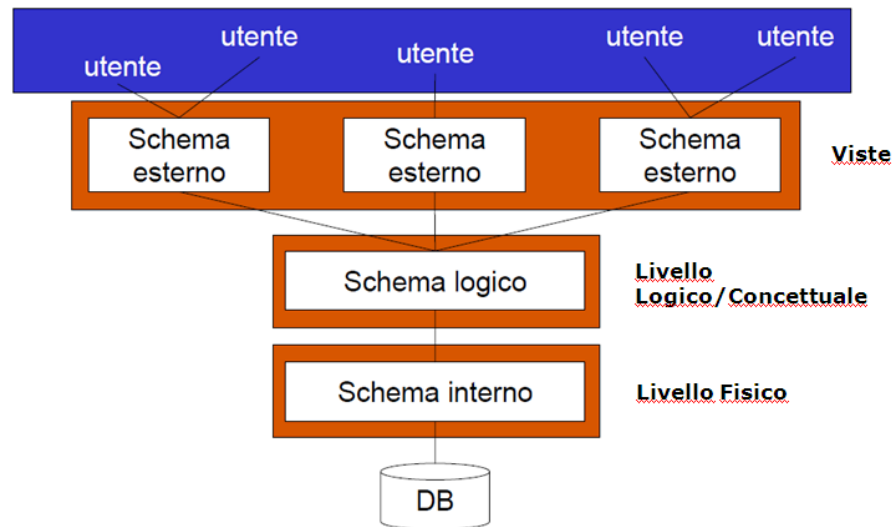
---

- Proposto da **E. Codd** nel 1970 per favorire l'indipendenza dei dati
- Disponibile come modello logico in DBMS reali nel 1981 (non è facile realizzare l'indipendenza con efficienza e affidabilità)
- Si basa sul concetto matematico di **Relazione**
- è dotato di un'elegante caratterizzazione formale e strutture dati (a livello logico) semplici e uniformi.
- Linguaggi di definizione, di aggiornamento e di interrogazione dei dati.

# Il Modello Relazionale dei Dati

Il Modello Relazionale riguarda la struttura logica di una base di dati, non lo schema concettuale

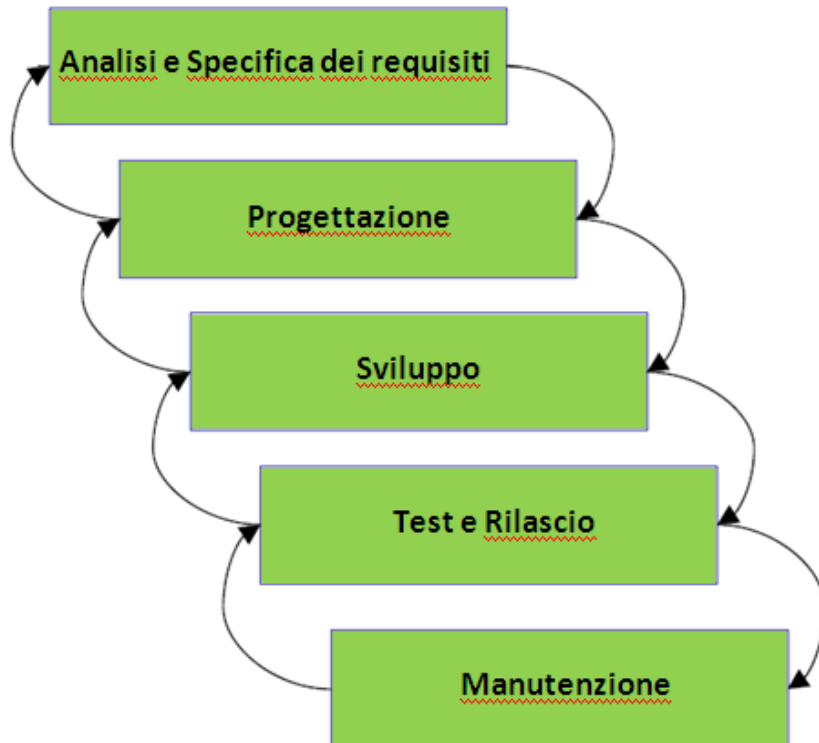
- non siamo più a livello di “progettazione”(concettuale), ma siamo a livello delle astrazioni che fornisce il DBMS adottato (Livello Concettuale di figura o Logico)



# Il Modello Relazionale dei Dati

---

Nella fase di progettazione concettuale, ossia nella fase di specifica, si può utilizzare un qualunque linguaggio di specifica (l'ER, ma anche altri)



➤ Ora siamo nella fase di progettazione Logica (effettiva)

➤ Siamo condizionati dal modello logico supportato dal DB scelto

# Il Modello Relazionale dei Dati

---

## Obiettivo della progettazione logica:

Tradurre lo schema concettuale (espresso nel modello ER) in uno schema logico che rappresenti gli stessi dati:

- utilizzando il modello logico del DBMS scelto
  - nel nostro caso, il **modello relazionale**
- in maniera completa ed efficiente

# Il Modello Relazionale dei Dati

---

- ▶ Il modello relazionale non è stato il primo modello ad essere adottato dai DB commerciali
  - ▶ agli albori dei DBMS, i modelli reticolare e gerarchico erano più usati
- ▶ Lo sviluppo della teoria relazionale dei dati per la manipolazione dei dati e lo sviluppo di opportune tecniche di memorizzazione hanno reso i DB relazionali lo standard dei DB commerciali attuali
  - ▶ Oracle, DB2, Informix, Access, MySQL sono tutti DB relazionali
  - ▶ Oracle supporta anche il modello relazionale ad oggetti
  - ▶ PostgreSQL DB relazionale ad oggetti

# Il Modello Relazionale dei Dati

---

- Si basa sul concetto matematico di **Relazione**
  - va intesa in senso **algebrico**
  - da non confondere con le relazioni tra i dati del modello concettuale
- Un database relazionale consiste di una collezione di tabelle, a ciascuna delle quali è assegnato un nome unico
- Poiché una tabella rappresenta una collezione di tali associazioni, c'è corrispondenza fra il concetto di tabella ed il concetto matematico di relazione, da cui il modello prende il nome.
- Costrutti fondamentali
  - ▶ **dominio**
  - ▶ **attributo/colonna**
  - ▶ **relazione/tabella**
  - ▶ **tupla/riga**

# Il Modello Relazionale dei Dati

---

## Relazione matematica

Siano  $D_1, D_2, \dots, D_n$  ( $n$  insiemi anche non distinti)

**Prodotto cartesiano**  $D_1 \times D_2 \times \dots \times D_n$ :

- l'insieme di tutte le tuple ordinate  $(d_1, d_2, \dots, d_n)$  tali che  $d_1 \in D_1, d_2 \in D_2, \dots, d_n \in D_n$

**Relazione matematica**  $R$  su  $D_1, D_2, \dots, D_n$

- è un sottoinsieme del prodotto cartesiano  $D_1 \times D_2 \times \dots \times D_n$

$D_1, D_2, \dots, D_n$  sono i **domini** della relazione.

Una relazione su  $n$  domini ha **grado** (o **arietà**)  $n$ .

il numero di tuple è la **cardinalità** della relazione.





# Il Modello Relazionale dei Dati

---

$D1 = \{0,1\}$

$D2 = \{2,3\}$

**Prodotto Cartesiano**  $D1 \times D2$

$D1 \times D2 =$

0	2
0	3
1	2
1	3

**Relazione R**  $\subseteq D1 \times D2$

$R =$

0	2
1	3

- I valori delle colonne di una tabella appartengono a dei **domini**
- Una tabella è un sottoinsieme del prodotto cartesiano fra i domini delle sue colonne

# Il Modello Relazionale dei Dati

---

Il costrutto principale è la **Relazione**

- ❑ Rappresenta un insieme di oggetti della realtà di interesse
- ❑ Consiste in uno Schema ed in una Istanza
  
- lo “scheletro” della relazione è lo **Schema**
  - Specifica il nome della Relazione
  - Descrive le intestazioni delle colonne della relazione (tabella)
  - Specifica il nome di ogni campo o colonna o attributo
  - Il Dominio di ciascun campo (a cui è associato un insieme di valori)
- la tabella con le varie tuple riempite è una **istanza** della relazione
- le righe di una tabella si chiamano **tuple o record**
- le colonne di una tabella sono gli **attributi**
- Il contenuto di una relazione **non dipende dall'ordinamento delle tuple**

# Rappresentazione di uno schema

---

Per rappresentare in **forma compatta** uno schema si utilizza la seguente notazione:

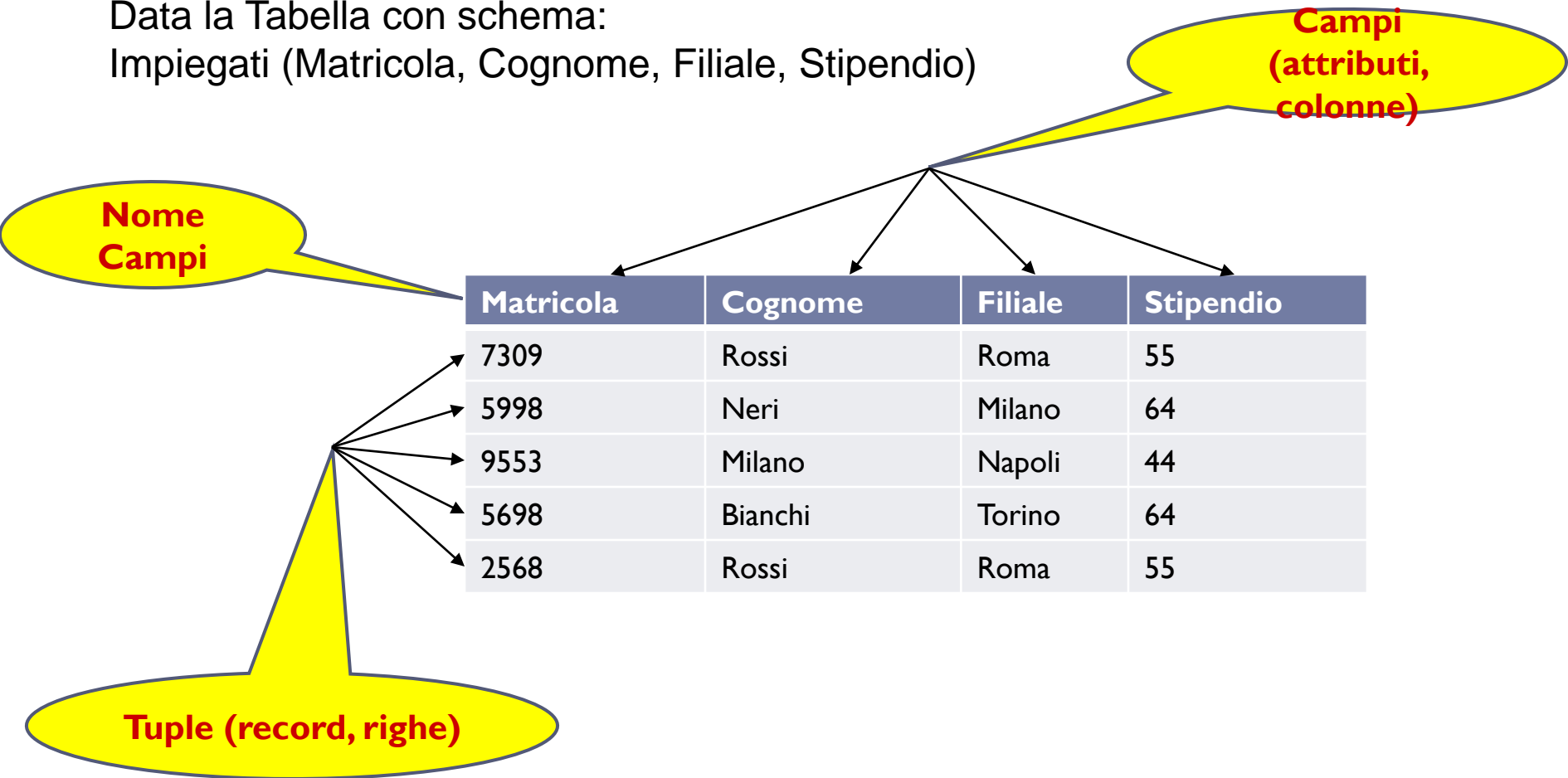
Studenti(Matricola,Cognome,Nome,DataNascita,Email)

Clienti(NomeCliente, Indirizzo)

# Rappresentazione di uno schema

Data la Tabella con schema:

Impiegati (Matricola, Cognome, Filiale, Stipendio)



# Vincoli di Dominio

---

E' il vincolo di base:

- va specificato per ogni attributo
- riguarda il suo dominio, ovvero che tipo di dati “ha senso” per quell'attributo
- i domini che si possono usare dipendono dal DBMS (Domini predefiniti)
- è possibile definire nuovi Domini

# Vincoli di integrità intra-relazionali e inter-relazionali

---

- ▶ **Schema** di una base di dati relazionale = insieme di schemi di relazione  $\{R_1; \dots; R_n\}$  + insieme di vincoli di integrità.
- ▶ **Istanza** di una base di dati relazionale = insieme di istanze di relazione  $\{R_1; \dots; R_n\}$  che soddisfano i vincoli di integrità.

I **vincoli intra-relazionali** coinvolgono una o più tuple di una singola relazione

I **vincoli inter-relazionali** coinvolgono tuple appartenenti a relazioni diverse

Sono i vincoli che le nostre istanze devono soddisfare affinché si possano considerare valide.

# Vincoli di integrità intra-relazionali

---

- ▶ vincoli relativi ad un singolo valore di una singola tupla
  - vincolo di dominio
  - vincolo NOT NULL
- ▶ vincoli che coinvolgono più valori di una singola tupla  
Esempio:  
nella tabella ESAMI(STUDENTE,MATERIA,DATA,VOTO,LODE)  
la LODE può essere assegnata solo se il VOTO è 30
- ▶ vincoli che coinvolgono più tuple di una singola tabella
  - vincolo di chiave: specifica delle chiavi candidate (le nozioni di superchiave, di chiave candidata e di chiave primaria)

Al vincolo di chiave viene affiancato il vincolo di integrità dell'entità:

- ▶ il valore della chiave primaria non può essere NULL

# Vincoli di integrità inter-relazionali

---

Informazioni in relazioni diverse sono correlate attraverso valori comuni, in particolare, attraverso valori delle chiavi (primarie, di solito)

Un vincolo di **integrità referenziale** (detto anche vincolo di (“**foreign key**”)) fra un insieme di attributi  $X$  di una relazione  $R_1$  e un'altra relazione  $R_2$  impone ai valori su  $X$  di ciascuna ennupla dell'istanza di  $R_1$  di comparire come valori della chiave (primaria) dell'istanza di  $R_2$ .



# Vincoli di tupla

---

I **vincoli di tupla** esprimono condizioni che devono essere soddisfatte da ciascuna tupla.

**Esempio:**

Esami(Matricola, Corso, Voto, Lode)

Il Voto deve essere compreso tra 18 e 30:

$(\text{Voto} \geq 18) \text{ AND } (\text{Voto} \leq 30)$



# Il Modello Relazionale dei Dati

---

## Esempio:

Supponiamo di avere i seguenti schemi:

- ▶ Depositi(NomeFiliale, NumConto, NomeCliente, Saldo)
- ▶ Clienti(NomeCliente, Indirizzo)

o in alternativa il seguente schema:

- ▶ Informazioni(NomeFiliale, NumConto, NomeCliente, Saldo, Indirizzo)
- lo schema Informazioni contiene tutte le informazioni
  - se un cliente ha più conti vengono ripetute in tutte le tuple le informazioni sull'indirizzo che sono inutili
  - la suddivisione in tabelle non è banale: posso avere problemi di ripetizione dell'informazione (ed altri)

# Il Modello Relazionale dei Dati

---

## Riassumiamo:

- ▶ Il modello relazionale è basato sul concetto di **relazione**
- ▶ Lo **schema** di una relazione consiste di un nome e di un insieme di attributi
- ▶ L' **istanza** di una relazione è un insieme di tuple
- ▶ In assenza di informazioni si fa uso di un particolare valore, detto valore nullo (**NULL**), che non appartiene a nessun dominio

# Il Modello Relazionale dei Dati

---

- ▶ Per garantire l' **integrità dei dati** si possono specificare diversi tipi di **vincoli**, che definiscono quali sono le istanze legali (ammissibili)
- ▶ I principali tipi di vincoli sono:
  - vincoli di dominio: limitano l'insieme dei valori ammissibili per un attributo
  - vincoli di assenza di valori nulli
  - vincoli di chiave e di chiave primaria
  - vincoli di integrità referenziale
  - vincoli di tupla