

# Introduzione ai Modelli Logici

Basato sulle slide del Prof. Danilo Montesi

15 maggio 2025

## Indice

<b>1</b>	<b>Introduzione ai Modelli Logici</b>	<b>2</b>
1.1	Modelli Tradizionali . . . . .	2
1.2	Modelli Recenti . . . . .	2
1.3	Caratteristiche dei Modelli . . . . .	2
<b>2</b>	<b>Il Modello Relazionale</b>	<b>2</b>
2.1	Fondamenti . . . . .	2
2.2	Relazione Logica vs. Relazione (Modello Relazionale) . . . . .	2
<b>3</b>	<b>Informazione Parziale (Valori Mancanti)</b>	<b>2</b>
3.1	Il Problema . . . . .	2
3.2	Il Valore NULL . . . . .	3
<b>4</b>	<b>Vincoli di Integrità</b>	<b>3</b>
4.1	Scopo . . . . .	3
4.2	Tipi di Vincoli . . . . .	3
<b>5</b>	<b>Identificazione delle Tuple (Chiavi)</b>	<b>3</b>
5.1	Concetti di Base . . . . .	3
5.2	Primary Key (PK) . . . . .	3
5.3	Importanza delle Chiavi . . . . .	4
<b>6</b>	<b>Integrità Referenziale (Foreign Key)</b>	<b>4</b>
6.1	Concetto . . . . .	4
6.2	Foreign Key (FK) . . . . .	4
6.3	Gestione delle Violazioni (su DELETE/UPDATE) . . . . .	4

# 1 Introduzione ai Modelli Logici

## 1.1 Modelli Tradizionali

- Gerarchico
- Reticolare (Network)
- **Relazionale**

## 1.2 Modelli Recenti

- Orientato agli Oggetti (meno comune)
- Basato su XML (complementare al relazionale)

## 1.3 Caratteristiche dei Modelli

- Gerarchico e Reticolare: Usano **riferimenti espliciti** (puntatori) tra i record.
- Relazionale: È "**value-based**". I riferimenti tra i dati in relazioni diverse sono rappresentati tramite i **valori** stessi all'interno delle tuple.

Il modello value-based offre indipendenza dalla struttura fisica e facilita la condivisione dei dati.

# 2 Il Modello Relazionale

## 2.1 Fondamenti

- Definito da **E. F. Codd** nel 1970 per l'indipendenza dei dati.
- Implementato nei DBMS reali dal 1981.
- Basato sul concetto logico di "*relazione*" dalla teoria degli insiemi, ma con differenze pratiche.
- Le relazioni sono rappresentate come **tabelle**.

## 2.2 Relazione Logica vs. Relazione (Modello Relazionale)

- **Relazione Logica (Teoria Insiemi)**: Sottoinsieme del prodotto cartesiano di domini. È un **insieme di tuple ordinate**.
- **Relazione (Modello Relazionale)**: È un **insieme** con proprietà specifiche:
  - Non c'è **ordine tra le tuple** (righe).
  - Le **tuple sono tutte distinte**.
  - Gli attributi (colonne) sono identificati da nomi univoci e non dalla posizione (struttura **non posizionale**).
  - I valori all'interno di ogni colonna sono **omogenei** (dallo stesso dominio).

# 3 Informazione Parziale (Valori Mancanti)

## 3.1 Il Problema

- Il modello relazionale ha una struttura rigida.
- Non tutte le informazioni sono sempre complete o disponibili.
- Come rappresentare l'assenza di un valore?

### 3.2 Il Valore NULL

- Tecnica **rough but effective** nel modello relazionale.
- **NULL** indica l'**assenza di un valore** atteso.
- **NULL non è un valore del dominio**.
- Una tupla  $t$  per un attributo  $A$  può avere  $t[A]$  pari ad un valore del dominio  $\text{dom}(A)$  o NULL.
- Esistono concettualmente diversi tipi di NULL (unknown, inexistent, uninformative), ma i DBMS moderni spesso non li distinguono.
- Un eccessivo uso di NULL può complicare l'uso dei dati.

## 4 Vincoli di Integrità

### 4.1 Scopo

- Garantire che le istanze del database rappresentino **informazioni corrette** rispetto al mondo reale (anche se sintatticamente valide).
- Supportare la **qualità dei dati**.
- Utili nella **progettazione** del database e nell'ottimizzazione delle **query**.
- Un vincolo può essere visto come un predicato (funzione booleana) sull'istanza.

I DBMS supportano ed enforcement alcuni tipi di vincoli; altri devono essere gestiti a livello applicativo.

### 4.2 Tipi di Vincoli

- **Intra-relazionali**: Riguardano una singola relazione.
  - Su **valori** (o di dominio): Regole sui singoli attributi (es. età > 0).
  - Su **tuple**: Regole sui valori all'interno di una singola tupla (es. stipendio lordo = netto + deduzioni).
- **Inter-relazionali**: Riguardano più relazioni.

## 5 Identificazione delle Tuple (Chiavi)

### 5.1 Concetti di Base

- Ogni tupla in una relazione deve essere **identificabile univocamente**.
- **Superkey**: Un insieme di attributi che identifica univocamente le tuple in una relazione. Non esistono due tuple distinte che abbiano gli stessi valori per tutti gli attributi della Superkey.
- **Key (Chiave)**: Una Superkey **minimale**. Rimuovendo qualsiasi attributo dalla Key, l'insieme rimanente non è più una Superkey.
- Ogni relazione possiede **almeno una Key** (l'insieme di tutti gli attributi è sempre una Superkey).

### 5.2 Primary Key (PK)

- Una delle Key di una relazione scelta per essere l'**identificatore principale** delle tuple.
- Le **Primary Key non possono contenere valori NULL**.
- Spesso indicata sottolineando gli attributi.

### 5.3 Importanza delle Chiavi

- Garantiscono l'**accessibilità** di ogni singola tupla.
- Sono cruciali per **correlare** le tuple tra relazioni diverse, fondamento del modello value-based.
- La presenza di NULL nelle Keys compromette l'identificazione e la creazione di riferimenti.

## 6 Integrità Referenziale (Foreign Key)

### 6.1 Concetto

- Le correlazioni basate sui valori tra relazioni (tipicamente sulle PK) devono essere **coerenti**.
- Assicura che i riferimenti tra tabelle siano validi.

### 6.2 Foreign Key (FK)

- Un vincolo di integrità inter-relazionale.
- Un insieme di attributi **FK** in una relazione  $R_1$  **fa riferimento** alla **Primary Key PK** di un'altra relazione  $R_2$ .
- **Regole per una FK:**
  1. Gli attributi della FK devono avere lo stesso **dominio** degli attributi della PK a cui si riferiscono.
  2. Un valore negli attributi FK di una tupla in  $R_1$  deve essere:
    - **NULL**
    - **OPPURE** deve **esistere come valore nella PK** di **qualche tupla** nella relazione  $R_2$  (referenziata).

### 6.3 Gestione delle Violazioni (su DELETE/UPDATE)

Quando un'operazione sulla relazione referenziata  $R_2$  causerebbe una violazione (es. cancellazione di una tupla in  $R_2$  referenziata da  $R_1$ ), il DBMS può agire:

- **Restrict (Comportamento Standard):** Impedisce l'operazione su  $R_2$ .
- **Cascade:** Estende l'operazione alle tuple referenzianti in  $R_1$  (es. cancella le tuple in  $R_1$ ).
- **Set NULL:** Imposta i valori della FK a NULL nelle tuple referenzianti in  $R_1$ .
- **Set Default:** Imposta i valori della FK a un valore predefinito nelle tuple referenzianti in  $R_1$ .

Per le Foreign Key composte da più attributi, l'**ordine** degli attributi è rilevante per il mapping con la Primary Key referenziata.