

La Normalizzazione delle Relazioni

Francesco Gobbi

I.I.S.S. Galileo Galilei, Ostiglia

23 gennaio 2025

La normalizzazione delle relazioni

Definizione: La normalizzazione delle relazioni è un processo fondamentale nella progettazione di basi di dati, che mira a:

- ▶ Ridurre la ridondanza dei dati.
- ▶ Evitare le anomalie durante le operazioni di trattamento dei dati (inserimento, aggiornamento, cancellazione).
- ▶ Garantire una struttura logica ottimale per le tabelle.

Problemi senza normalizzazione:

- ▶ Duplicazione inutile dei dati (ridondanza).
- ▶ Aumento dello spazio utilizzato su disco.
- ▶ Anomalie nelle operazioni sui dati (esempio: aggiornamenti incoerenti).

Esempio di anomalia senza normalizzazione

Scenario: Tabella OrdiniNonNormalizzata

<u>IDOrdine</u>	Cliente	Prodotto	Indirizzo
1	Mario Rossi	Laptop	Via Roma, 10
2	Mario Rossi	Smartphone	Via Roma, 10
3	Anna Bianchi	Tablet	Via Milano, 20

Tabella: Tabella con dati non normalizzati

Problemi:

- ▶ L'indirizzo di Mario Rossi viene ripetuto inutilmente.
- ▶ Aggiornare l'indirizzo per Mario Rossi richiede modifiche multiple.
- ▶ Possibilità di inconsistenze se non tutte le righe vengono aggiornate correttamente.

Come funziona la normalizzazione

La normalizzazione delle relazioni si basa su tre forme principali:

- ▶ **Prima Forma Normale (1NF):** Tutti gli attributi contengono valori atomici e non ci sono gruppi ripetuti.
- ▶ **Seconda Forma Normale (2NF):** Tutti gli attributi non chiave dipendono dall'intera chiave primaria (eliminazione di dipendenze parziali).
- ▶ **Terza Forma Normale (3NF):** Eliminazione delle dipendenze transitive tra gli attributi non chiave.

Benefici:

- ▶ Riduzione della ridondanza.
- ▶ Maggiore coerenza dei dati.
- ▶ Più facile gestione delle operazioni di aggiornamento e cancellazione.

Algoritmo di scomposizione dettagliato - Parte 1

Dipendenza funzionale:

- ▶ Una dipendenza funzionale $X \rightarrow Y$ significa che il valore di X determina univocamente il valore di Y .
- ▶ Esempio: Nella relazione `Studenti(IDStudente, Nome, Corso)`, $IDStudente \rightarrow Nome$ implica che ogni `IDStudente` identifica univocamente un `Nome`.

Algoritmo - Prima parte:

1. Identificare tutte le dipendenze funzionali e le chiavi candidate in T .
2. Selezionare una dipendenza funzionale $X \rightarrow Y$ che viola le regole di normalizzazione.

Nota: Questo passaggio permette di individuare gli attributi che causano ridondanza o anomalie nei dati.

Algoritmo di scomposizione dettagliato - Parte 2

Algoritmo - Seconda parte:

3. Creare una nuova relazione con gli attributi $X \cup Y$.
4. Rimuovere Y dalla relazione originale mantenendo X come chiave primaria.
5. Ripetere il processo finchè non ci sono più dipendenze funzionali che violano le regole.

Risultato: Ogni relazione sarà conforme alle forme normali richieste, garantendo una struttura logica ottimale e priva di ridondanza.

Esercizio 1

Scenario: Tabella non normalizzata

DipendentiNonNormalizzata

<u>ID</u>	Nome	Progetto	Dipartimento
1	Laura Bianchi	Progetto Alpha	Informatica
2	Marco Rossi	Progetto Beta	Fisica
3	Laura Bianchi	Progetto Gamma	Informatica

Domande:

- ▶ Identificare i problemi di ridondanza.
- ▶ Dividere la tabella in più relazioni normalizzate.

Soluzione Esercizio 1

Passaggi per la normalizzazione:

- ▶ Creazione delle seguenti tabelle:
 - ▶ Dipendenti(ID, Nome)
 - ▶ Progetti(IDProgetto, NomeProgetto, IDDipartimento)
 - ▶ Dipartimenti(IDDipartimento, NomeDipartimento)

Esercizio 2

Scenario: La tabella seguente rappresenta dati non normalizzati su studenti e corsi.

<u>IDStudente</u>	Nome	Corso	Docente
1	Maria Verdi	Matematica	Prof. Bianchi
2	Luca Neri	Fisica	Prof. Rossi
3	Maria Verdi	Fisica	Prof. Rossi

Richiesta: Dividere la tabella in più relazioni normalizzate rispettando la 3NF.

Soluzione Esercizio 2

Passaggi per la normalizzazione:

- ▶ Creazione delle seguenti tabelle:
 - ▶ Studenti(IDStudente, Nome)
 - ▶ Corsi(IDCorso, NomeCorso, IDDocente)
 - ▶ Docenti(IDDocente, NomeDocente)

Vantaggi della normalizzazione

- ▶ Riduzione dello spazio utilizzato grazie all'eliminazione della ridondanza.
- ▶ Maggiore coerenza e integrità dei dati.
- ▶ Operazioni di aggiornamento, cancellazione e inserimento più semplici e affidabili.
- ▶ Maggiore flessibilità nella gestione dei dati.

Prima Forma Normale (1NF)

Requisiti:

- ▶ Ogni colonna deve contenere valori atomici (non ulteriormente scomponibili).
- ▶ Non ci devono essere gruppi ripetuti di attributi.

Esempio: Una tabella `Clients` non normalizzata potrebbe contenere una colonna `Telefoni` con valori multipli separati da una virgola. Per rispettare la 1NF, ogni numero di telefono deve essere separato in righe distinte o in un'altra tabella.

Obiettivo: Garantire che i dati siano organizzati in modo tabellare e privo di valori multipli.

Seconda Forma Normale (2NF)

Requisiti:

- ▶ La tabella deve soddisfare la 1NF.
- ▶ Ogni attributo non chiave deve dipendere dall'intera chiave primaria (assenza di dipendenze parziali).

Esempio: Consideriamo una tabella Ordini (IDOrdine, IDProdotto, NomeProdotto). Qui NomeProdotto dipende solo da IDProdotto e non dall'intera chiave (IDOrdine, IDProdotto). Per soddisfare la 2NF, NomeProdotto deve essere spostato in una tabella Prodotti separata.

Obiettivo: Eliminare dipendenze parziali e separare le informazioni in tabelle logiche.

Terza Forma Normale (3NF)

Requisiti:

- ▶ La tabella deve soddisfare la 2NF.
- ▶ Non ci devono essere dipendenze transitive tra attributi non chiave.

Esempio: Consideriamo una tabella Dipendenti(ID, Nome, IDDipartimento, NomeDipartimento). Qui NomeDipartimento dipende transitivamente da IDDipartimento. Per rispettare la 3NF, NomeDipartimento deve essere spostato in una tabella Dipartimenti.

Obiettivo: Garantire una struttura logica ottimale e eliminare dipendenze transitive per una migliore gestione dei dati.