

1 Capitolo 1: Introduzione

- **Sistema Informativo**: sistema per l'organizzazione e la gestione delle informazioni,
- **Sistema Informatico**: porzione automatizzata del sistema informativo
- **Informazione**: notizia o elemento che consente di avere **conoscenza** di fatti o situazioni
- **Dato**: ciò che è immediatamente presente alla conoscenza prima di ogni elaborazione (simboli da elaborare)
- **Base di dati**: collezione di dati (gestita da un **DBMS**)
- **DBMS**: Database Management System, software per la gestione dei DB, garantisce:
 - affidabilità
 - privacy
 - efficienza
 - efficacia
 - accesso simultaneoI DBMS estendono le funzionalità del file system.
- **Modello di dati**: rappresenta come i dati sono organizzati strutturalmente
 - **Modello relazionale**: basato sul concetto di relazioni, i dati sono rappresentati come tabelle chiamate relazioni
 - **Modello gerarchico**: basato su strutture ad albero
 - **Modello reticolare**: basato sull'uso di grafi
 - **Modello a oggetti**: evoluzione del modello relazionale, che estende alle basi di dati il paradigma OOP
 - **Modello XML**: rivisitazione del modello gerarchico
- **Schema**: descrive la struttura della base di dati ed invariante nel tempo. Nel modello relazionale lo schema di una relazione è costituito dal nome della relazione, seguito dai suoi attributi (`Docenza(Corso, Nome, Docente)`)
- **Istanza**: fa riferimento ai valori attualmente memorizzati nella base di dati in un determinato momento. L'istanza di una relazione è costituita dall'insieme variante nel tempo, delle sue righe.

2 Modello Relazionale

Modello logico per l'organizzazione dei dati, basato sul concetto matematico di **relazione** tra insiemi.

- **Relazione** (Tabella): una relazione può essere vista come una **tabella** (un insieme matematico) che contiene un insieme di **record omogenei**. Per definizione una relazione non può contenere tuple uguali tra loro. Una tabella rappresenta quindi una relazione solamente se le tuple sono **uniche**. L'ordine delle righe/colonne non è rilevante.
- **Tupla**: rappresenta un singolo record o riga della tabella.
- **Attributo** (Colonna): **intestazione** di una colonna nella tabella che descrive una proprietà della relazione. Ad ogni attributo è infatti associato un nome che ne descrive il ruolo e un dominio di possibili valori
- **Schema di Relazione**: schema (struttura) di una singola tabella
- **Istanza di relazione** (**relazione**): insieme delle tuple che compongono la tabella
- **Schema del database**: insieme degli schemi di relazione che compongono il database
- **Istanza del database** (**database**): insieme di tabelle che compongono il database
- **Valori nulli**: valori utilizzabili per rappresentare valori sconosciuti o non applicabili. Tuttavia, la loro presenza nelle chiavi può impedire l'identificazione univoca delle tuple e la realizzazione efficace dei riferimenti. Il loro numero deve essere ridotto!

2.1 Vincoli

- **Vincoli intra-relazionali**: regole che coinvolgono una **singola relazione**
 - **Vincoli di dominio**: riguardano i valori che un attributo può assumere
 - **Vincoli sui valori Nulli** (NOT NULL): impongono che un attributo non possa avere valori nulli
 - **Vincoli di unicità** (UNIQUE): assicurano che i valori in una colonna o un gruppo di colonne siano unici all'interno della tabella.
 - **Chiavi**:
 - **Superchiave**: insieme di attributi che identificano in **modo univoco una tupla**
 - **Chiave** (candidata): **set minimale** di attributi che identificano in modo univoco una tupla. È una **superchiave minimale**.
 - **Chiave Primaria**: chiave candidata scelta per essere l'**identificatore principale** di una relazione, sulla quale non sono ammessi valori nulli
 - **Chiave Esterna**: insieme di attributi in una relazione (**referenziante**) che fa riferimento alla chiave primaria di un'altra relazione (**referenziata**), garantendo l'**integrità referenziale**
- **Vincoli di Controllo** (CHECK): Permettono di specificare condizioni generiche che devono essere sempre verificate sui valori degli attributi, anche in relazione ad altri attributi, all'interno della stessa tabella.
- **Vincoli Inter-relazionali**: regole che coinvolgono relazioni diverse
 - **Vincoli di integrità Referenziale** (Chiave Esterna - FOREIGN KEY): meccanismo principale per collegare logicamente tuple di tabelle diverse. Una **chiave esterna** è un insieme di attributi in una relazione (detta **referenziante**) i cui valori devono corrispondere ai valori della **chiave primaria** in un'altra relazione (detta **referenziata**). La **chiave esterna** e la **chiave primaria** a cui fa riferimento devono avere lo stesso numero di colonne e tipi di dati compatibili. La presenza di un valore nullo in una chiave esterna non viola il vincolo.

Figura 2.19
Una base di dati
con vincoli di integrità
referenziale.

Infrazioni

Codice	Data	Agente	Articolo	Stato	Numero
143256	25/10/2021	567	44	I	AB 234 ZK
987554	26/10/2021	456	34	I	AB 234 ZK
987557	26/10/2021	456	34	I	CB 123 AA
630876	15/10/2021	456	53	F	CB 123 AA
539856	12/10/2021	567	44	F	CB 123 AA

Agenti

Matricola	CF	Cognome	Nome
567	RSSM ...	Rossi	Mario
456	NREL ...	Neri	Luigi
638	NREP ...	Neri	Piero

Auto

Stato	Numero	Proprietario	Indirizzo
I	CB 123 AA	Verdi Piero	Via Tigli ...
I	DE 834 ZZ	Verdi Piero	Via Tigli ...
I	AB 234 ZK	Bini Luca	Via Aceri ...
F	CB 123 AA	Beau Marcel	Rue Louis XIV ...

- **Un vincolo di integrità referenziale** ("**foreign key**" – **Chiave Esterna**) fra gli attributi X di una relazione R_1 (REFERENZIANTE) un'altra relazione R_2 (REFERENZIATA) impone ai valori su X in R_1 di comparire come valori della chiave primaria di R_2

3 SQL: Concetti Base

- **Catalogo Relazionale**: porzione del database che contiene una **descrizione centralizzata dei dati**. I DB contengono 2 diverse tabelle: quelle che contengono i dati e quelle che contengono i metadati, ovvero dati

che descrivono altri dati. L'insieme delle tabelle che contengono i metadati costituisce il **catalogo della basi di dati**.

- **Dizionario dei dati:** **descrizione centralizzata** delle strutture dei dati: entità, attributi, relazioni, tipi di dati, vincoli ecc. Viene utilizzato per documentare lo schema E-R


BASI DI DATI

Metodologie e Modelli per il Progetto

DIZIONARIO DEI DATI (ENTITÀ)

Baratterei tutta la mia tecnologia per una serata con Socrate (Steve Jobs)

Entità	Descrizione	Attributi	Identificatore
Impiegato	Dipendente dell'azienda	Codice, Cognome, Stipendio	Codice
Progetto	Progetti aziendali	Nome, Budget	Nome
Dipartimento	Struttura aziendale	Nome, Telefono	Nome, Sede
Sede	Sede dell'azienda	Città, Indirizzo	Città



4 Normalizzazione

La normalizzazione è una tecnica concepita per **trasformare** schemi non normalizzati in schemi di **alta qualità** (ovvero che soddisfano una forma normale). L'obiettivo primario è **evitare** problemi di **ridondanza** e inconsistenza dei dati che possono sorgere da informazioni eterogenee all'interno di una singola relazione. Questo processo si avvale delle **Forme Normali** (FN), che rappresentano delle proprietà che lo schema deve rispettare.

Terminologia:

1. **chiave o chiave primaria:** insieme di attributi che identificano univocamente una tupla;
2. **chiave candidata:** insieme di attributi che possono svolgere la funzione di chiave;
3. **attributo no chiave:** attributo che non fa parte della chiave primaria
4. **attributo primo:** attributo che fa parte di almeno una chiave della relazione

4.1 Dipendenze funzionali

Sono particolari **vincoli di integrità** che descrivono legami tra gli attributi di una relazione.

Una dipendenza funzionale $Y \rightarrow Z$ su uno schema di relazione $R(X)$ esiste se, per ogni coppia di tuple t_1 e t_2 nella relazione, che hanno gli stessi valori sull'insieme di attributi Y (detto **determinante**), avranno necessariamente anche gli stessi valori sull'insieme di attributi Z .

In altre parole, il valore di Y determina univocamente il valore di Z .

- **determinante:** quando tra due attributi uno determina il valore dell'altro si dice che A_1 è determinante per A_2

Sostanzialmente si ha una dipendenza funzionale quando il valore di un insieme di attributi A , determina il singolo valore dell'attributo B : $A \rightarrow B$. A è un determinante per B .

BASI DI DATI

La Normalizzazione

5

DISTINGUERE LE DIPENDENZE FUNZIONALI

Impiegato	Stipendio	Progetto	Bilancio	Funzione
Rossi	20	Marte	2	tecnico
Verdi	35	Giove	15	progettista
Verdi	35	Venere	15	progettista
Neri	55	Venere	15	direttore
Neri	55	Giove	15	consulente
Neri	55	Marte	2	consulente
Mori	48	Marte	2	direttore
Mori	48	Venere	15	progettista
Bianchi	48	Venere	15	progettista
Bianchi	48	Giove	15	direttore

Impiegato → Stipendio

Progetto → Bilancio

Impiegato Progetto → Funzione

- **DF banale**: una dipendenza $Y \rightarrow A$ si considera banale se l'attributo A è già incluso nell'insieme degli attributi Y . La dipendenza è sempre soddisfatta e non causa anomalie. (Eg. $Impiegato\ Progetto \rightarrow Progetto$).

Le chiavi delle relazioni identificano (sono **determinanti**) tutti gli altri attributi dello schema.

Il vincolo di dipendenza funzionale infatti **generalizza il vincolo di chiave**, in particolar modo: la dipendenza $Y \rightarrow Z$ degenera nel vincolo di chiave se l'unione di Y e Z è pari ad X (il quale rappresenta lo schema completo della relazione). In tal caso infatti Y è (super)chiave dello schema.

4.2 Forme Normali

Le forme normali sono delle proprietà che, se soddisfatte eliminano o limitano il numero di **anomalie** presenti.

1. 1NF:

- **dominio** di ogni attributo contiene solo valori **atomici** (indivisibili) e il valore di ogni **attributo** in una tupla è un valore **singolo** del dominio;
- È parte integrante della definizione formale di relazione nel modello relazionale.

2. 2NF:

- È in 1NF
 - Tutti gli **attributi non-chiave** dipendono dall'intera chiave primaria;
- Elimina la **dipendenza parziale** degli attributi dalla chiave.

3. 3NF:

- È in 2NF;
 - tutti gli attributi non-chiave dipendono (direttamente) solo dalla chiave primaria
- Elimina le **dipendenze transitive**. Formalmente, per ogni dipendenza funzionale non banale $X \rightarrow Y$ definita su una relazione r , deve essere verificata almeno una delle seguenti condizioni:
- X contiene una chiave K di r
 - ogni attributo in Y è contenuto in almeno una chiave di r , quindi è un attributo primo

In 3NF, sono ammesse dipendenze funzionali in cui il **determinante non** è una **superchiave solo se** il dipendente è un attributo **primo** (cioè parte di una chiave candidata);

4. BCNF:

- Per ogni dipendenza funzionale non banale $X \rightarrow Y$, X è (o contiene) una superchiave della relazione. È una forma di normalizzazione più stringente rispetto alla 3NF. In BCNF, **non sono ammesse eccezioni**: ogni determinante di una dipendenza funzionale non banale deve essere una superchiave.

4.2.1 3NF vs BCNF

- **3NF** è meno restrittiva: permette che un attributo non-primario dipenda da qualcosa che non è una superchiave, **a patto che il dipendente sia un attributo primario**.
- **BCNF** è più restrittiva: **non ammette alcuna dipendenza** in cui il determinante non sia una superchiave, indipendentemente dal fatto che il dipendente sia primario o meno.

4.3 Anomalie

Le anomalie sono prevalentemente causate dalla presenza di informazioni eterogenee all'interno della stessa relazione:

1. **Anomalia di aggiornamento**: a causa della ridondanza, la stessa informazione è registrata in più righe. Se si modifica il valore in una sola riga e non nelle altre, si crea **inconsistenza** tra i dati. Ad esempio, se l'indirizzo di un cliente è presente su più record (perché ripetuto ogni volta che acquista), aggiornare solo alcuni record rende il dato incoerente tra le righe.
2. **Anomalia di cancellazione**: cancellando un'informazione si perdono anche dati che si desiderava mantenere. Per esempio, se si elimina dalla tabella ordini un ordine che era l'unico associato a un certo cliente, si perde anche l'informazione sul cliente stesso. Questo accade perché le informazioni diverse sono **"accorpate"** in un'unica tabella.
3. **Anomalia di inserimento**: non è possibile aggiungere nuovi dati perché alcuni **campi chiave** non possono essere ancora inseriti. Ad esempio, se si vuole registrare un nuovo cliente che non ha ancora effettuato nessun ordine, ma lo schema obbliga ad associare ogni cliente a un ordine, non sarà possibile inserire il nuovo cliente, oppure si dovranno inserire dei valori nulli, con conseguente perdita di significato.

4.4 Processo di Normalizzazione

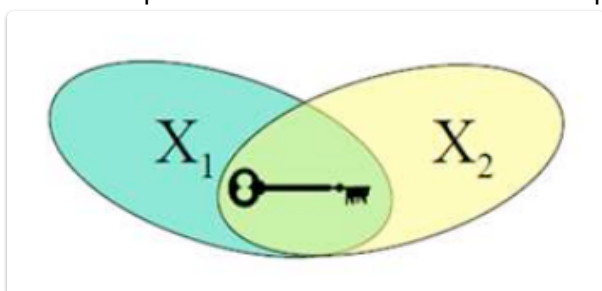
Data una relazione che non soddisfa la forma normale BCNF, è possibile **normalizzarla**, decomponendo la relazione originale in relazioni più piccole. Generalmente una buona decomposizione deve essere **senza perdita** e garantire la **conservazione della dipendenza**. Non è tuttavia sempre possibile raggiungere tali standard.

4.4.1 Decomposizione senza perdita

Definizione

Una relazione r si decompone senza perdita su X_1 e X_2 se il join delle due proiezioni è uguale ad r stessa (senza tuple spurie).

- **Condizione sufficiente**: la decomposizione senza perdita è garantita se gli **attributi comuni** contengono una **chiave** per almeno una delle relazioni decomposte.



🔗 Teorema di decomposizione senza perdita

sia:

- r una relazione su X
- siano X_1 ed X_2 sottoinsiemi di X tali che $X_1 \cup X_2 = X$ e $X_0 = X_1 \cap X_2$

allora:

r si decompone senza perdita su X_1 e X_2 se soddisfa la dipendenza funzionale

- $X_0 \rightarrow X_1$ e/o
- $X_0 \rightarrow X_2$
- In altre parole: **gli attributi comuni tra le due relazioni decomposte devono determinare tutti gli attributi almeno di una delle proiezioni**: l'intersezione degli attributi è una **superchiave** per almeno una delle due relazioni.

PROPRIETÀ DELLE DECOMPOSIZIONI

- **Teorema: La decomposizione senza perdita è garantita se, posto $X_0 = X_1 \cap X_2$ la relazione soddisfa la dipendenza funzionale $X_0 \rightarrow X_1$ oppure la dipendenza $X_0 \rightarrow X_2$ (oppure entrambe).**
- **Essa è garantita se gli attributi comuni alle relazioni decomposte contengono una chiave per almeno una di esse.**
- **Cio avviene se gli attributi comuni costituiscono il primo membro di almeno una delle dipendenze su cui si effettua la decomposizione**
- ❖ **Nell'esempio: l'intersezione degli insiemi di attributi su cui sono effettuate le due proiezioni è costituita da attributi che non è il primo membro di alcuna dipendenza funzionale.**

4.4.2 Conservazione della dipendenza

È fondamentale tuttavia che la decomposizione **conservi le dipendenze** dello schema originale ossia che tutti i **vincoli** di integrità originali della base di dati siano **mantenuti** e possano essere agevolmente controllati dal DBMS anche dopo la scomposizione dello schema.

🔗 Hint

La conservazione è garantita se **ciascuna dipendenza** dello schema originale coinvolge attributi che compaiono tutti insieme in uno degli schemi decomposti.

Vuol dire che **i due lati della freccia**, X e Y , devono stare **insieme** in almeno una delle tabelle dopo la decomposizione.

☰ Esempio Conservazione della dipendenza

Supponiamo di avere:

- $R(A, B, C)$

- $F = \{A \rightarrow B, B \rightarrow C\}$

Decomposte in:

- $R1(A, B)$
- $R2(B, C)$
- Gli attributi A e B compaiono **insieme** nella relazione $R1$
- Gli attributi B e C compaiono **insieme** nella relazione $R2$

☰ Controesempio Conservazione della dipendenza

- $R(A, B, C)$
- $F = \{A \rightarrow C\}$

Decomposte in:

- $R1(A, B)$
- $R2(B, C)$

L'attributo A è nella relazione $R1$ mentre l'attributo C è nella relazione $R2$ quindi non compaiono insieme in nessuna delle due relazioni per cui la dipendenza $A \rightarrow C$ **non è conservata**.

4.4.3 Decomposizione in BCNF

PROCEDURA INTUITIVA DI NORMALIZZAZIONE

❖ Scomposizione, con relazioni in BCNF

- **Non valida in generale, ma solo nei «casi semplici»**
- **Per ogni dipendenza $X \rightarrow Y$ che viola la BCNF, definire una relazione su XY ed eliminare Y dalla relazione originaria**
- **Nell'esempio precedente, la decomposizione è stata molto semplice: Ciascuna delle dipendenze è "migrata" in una delle tre relazioni; in pratica le dipendenze hanno suggerito la decomposizione.**
- **Non sempre le cose sono così immediate: pur in casi semplici si può avere una descrizione delle dipendenze non semplice.**

Tale processo di decomposizione può provocare perdita di informazioni qualora il caso in esame fosse semplice.

4.4.4 Decomposizione in 3NF

Non sempre la decomposizione in BCNF è possibile: se per esempio una dipendenza funzionale evidenziata coinvolge tutti gli attributi, nessuna decomposizione è in grado di conservarla. In tal caso una buona decomposizione in BCNF non è possibile.

In tal caso occorre decomporre in 3NF; tale **decomposizione è sempre possibile**.

 Hint

Si crea una relazione per ogni gruppo di attributi coinvolti in una dipendenza funzionale e si verifica che alla fine una dipendenza contenga una chiave della relazione originale.

4.4.5 Algoritmo di decomposizione

1. Identificare le **dipendenze funzionali**
2. Creare una **relazione per ogni gruppo di attributi coinvolti in una dipendenza funzionale**
3. Verificare che almeno una delle nuove relazioni contenga una **chiave della relazione originale**. Se le decomposizioni non contengono una chiave della tabella originale è necessario creare una nuova relazione che la contenga