

BASE DI DATI

(accezione generica)

collezione di dati, utilizzati per rappresentare le informazioni di interesse per una o più applicazioni di una organizzazione.

(accezione specifica)

collezione di dati gestita da un DBMS

SISTEMA DI GESTIONE DI BASI DI DATI DATABASE MANAGEMENT SYSTEM - DBMS

Sistema informativo in grado di gestire **collezioni di dati** che siano (anche):

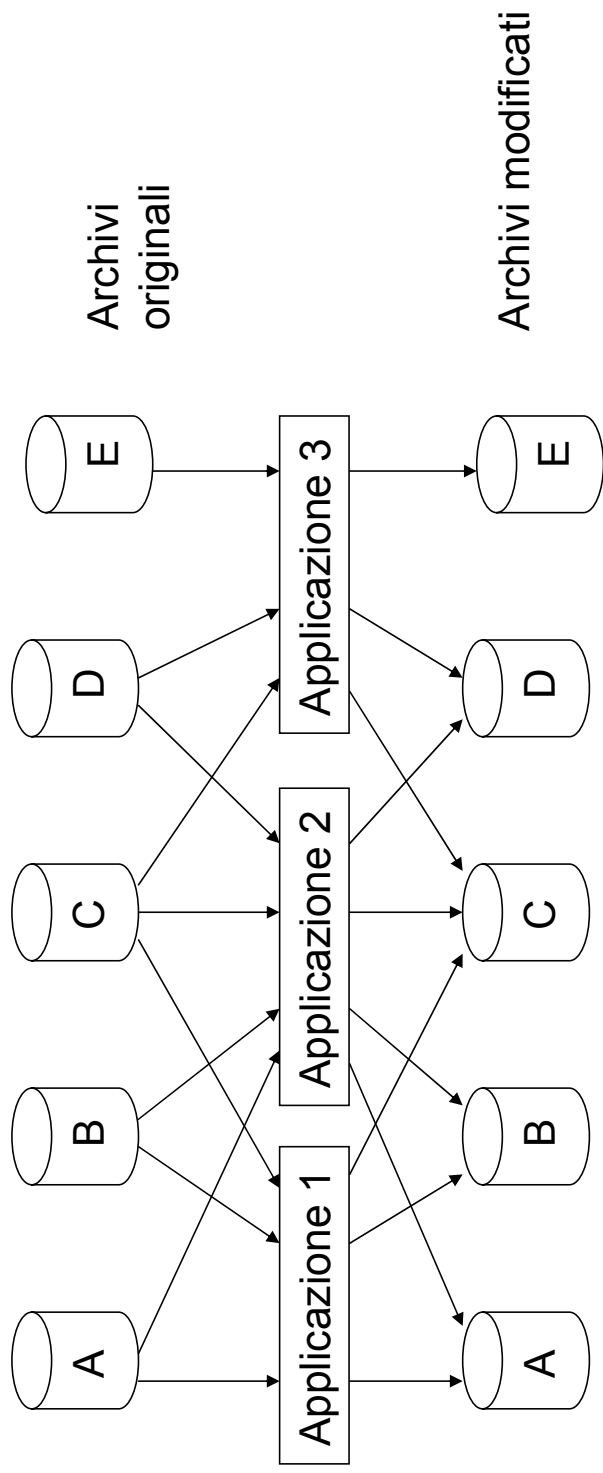
- **grandi** (di dimensioni (molto) maggiori della memoria centrale dei sistemi di calcolo utilizzati)
- **persistenti** (con un periodo di vita indipendente dalle singole esecuzioni dei programmi che le utilizzano)
- **condivise** (utilizzate da applicazioni diverse)

garantendo **affidabilità** (resistenza a malfunzionamenti hardware e software) e **privatezza** (con una disciplina e un controllo degli accessi).

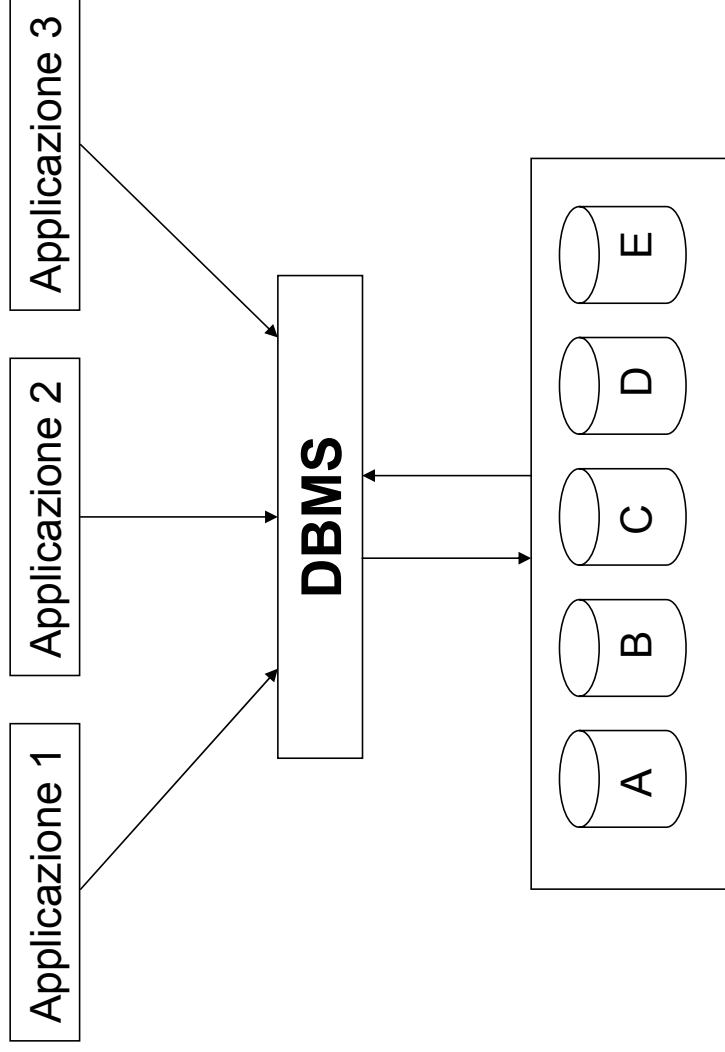
Condivisione

- Una base di dati è una risorsa **integrata, condivisa** fra vari settori.
- L'integrazione e la condivisione permettono di ridurre la **ridondanza** (evitando ripetizioni) e, di conseguenza, le possibilità di incoerenza (o **inconsistenza**) fra i dati.
- Poiché la condivisione non è mai completa (o comunque non opportuna) i DBMS prevedono meccanismi di definizione della **privatezza** dei dati e di limitazioni all'accesso (**autorizzazioni**).
- La condivisione richiede un opportuno coordinamento degli accessi: **controllo della concorrenza**.

DBMS vs. file system



DBMS vs. file system



DBMS vs. file system

- La gestione di insiemi di dati grandi e persistenti è possibile anche attraverso sistemi più semplici - gli ordinari **file system** dei sistemi operativi, che permettono di realizzare anche rudimentali forme di condivisione.
- I DBMS estendono le funzionalità dei file system, fornendo più servizi ed in maniera integrata (**maggiore efficacia**).
- I file system prevedono forme di condivisione, permettendo accessi contemporanei in lettura ed esclusivi in scrittura: se è in corso un'operazione di scrittura su un file, altri utenti non possono accedere affatto al file. Nei DBMS, c'è maggiore flessibilità: si può accedere contemporaneamente a record diversi di uno stesso file o addirittura allo stesso record (in lettura).

DBMS vs. file system (2)

- Nei programmi tradizionali che accedono a file, ogni programma contiene una descrizione della struttura del file stesso, con i conseguenti rischi di incoerenza fra le descrizioni (ripetute in ciascun programma) e i file stessi.
- Nei DBMS, esiste una porzione della base di dati (il **catalogo** o **dizionario**) che contiene una descrizione centralizzata dei dati, che può essere utilizzata dai vari programmi.

Vantaggi e svantaggi dei DBMS

Pro

- dati come risorsa comune, base di dati come modello della realtà
- gestione centralizzata con possibilità di standardizzazione ed “economia di scala”
- disponibilità di servizi integrati
- riduzione di ridondanze e inconsistenze
- indipendenza dei dati (favorisce lo sviluppo e la manutenzione delle applicazioni)

Contro

- costo dei prodotti e della transizione verso di essi
- non scorponabilità delle funzionalità (con riduzione di efficienza)

Descrizioni dei dati nei DBMS

- Esistono descrizioni e rappresentazioni dei dati a livelli diversi, che permettono **l'indipendenza dei dati** dalla rappresentazione fisica: i programmi fanno riferimento alla struttura a livello più alto, e le rappresentazioni sottostanti possono essere modificate senza necessità di modifica dei programmi.
- Precisiamo attraverso il concetto di modello dei dati.

Modello dei dati

- insieme di costrutti utilizzati per organizzare i dati di interesse e descriverne la dinamica
- componente fondamentale: **meccanismi di strutturazione (o costruttori di tipo)**
- come nei linguaggi di programmazione esistono meccanismi che permettono di definire nuovi tipi, così ogni modello dei dati prevede alcuni costruttori
- ad esempio, il **modello relazionale** prevede il costruttore **relazione**, che permette di definire insiemi di record omogenei

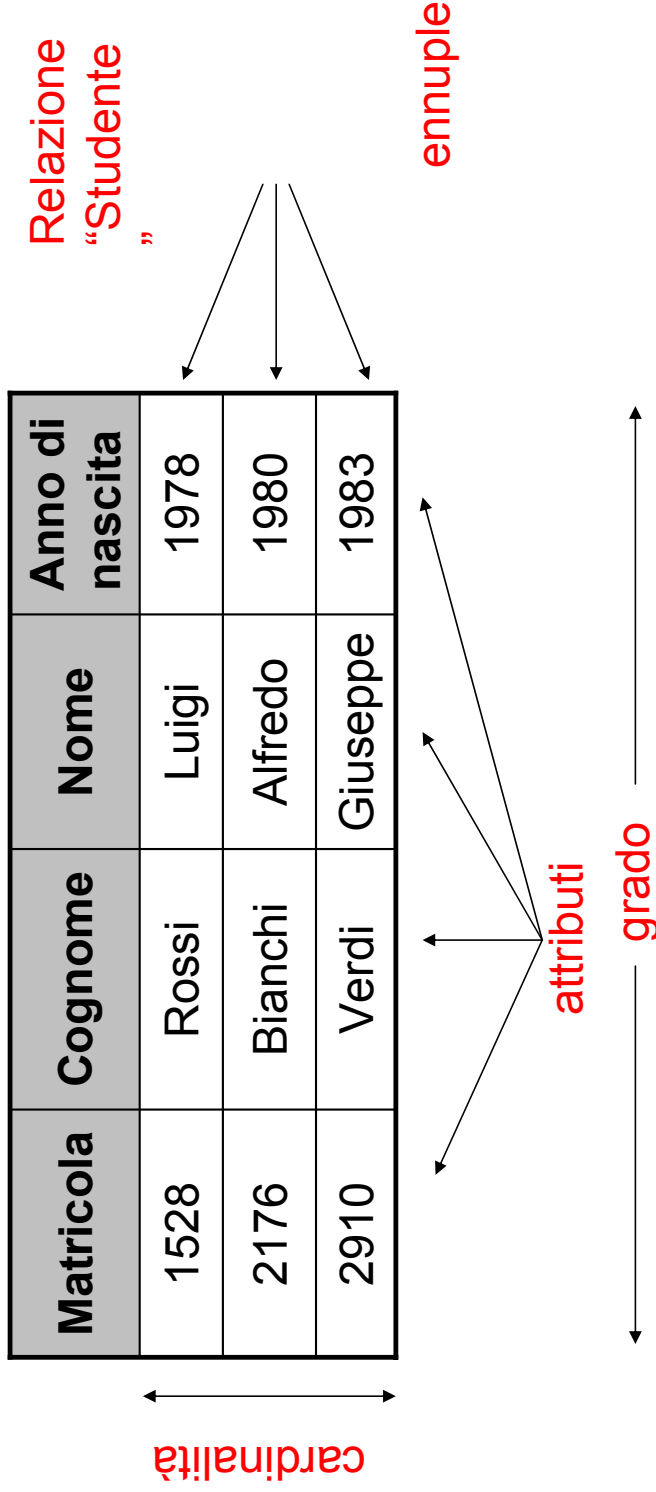
I modelli logici dei dati

- Tradizionalmente, esistono tre modelli logici:
 - **gerarchico**
 - **reticolare**
 - **relazionale**
- I modelli gerarchico e reticolare sono più vicini alle strutture fisiche di memorizzazione, mentre il modello relazionale è più astratto:
 - nel modello relazionale si rappresentano solo valori — anche i riferimenti fra dati in strutture (relazioni) diverse sono rappresentati per mezzo dei valori stessi;
 - nei modelli gerarchico e reticolare si utilizzano riferimenti espliciti (puntatori) fra record.
- Più recentemente, è stato introdotto il modello **a oggetti**

Il modello relazionale

- Proposto da E. F. Codd nel 1970 per favorire l'indipendenza dei dati e reso disponibile come modello logico in DBMS reali nel 1981 (non è facile implementare l'indipendenza con efficienza e affidabilità!).
- Si basa sul concetto matematico di **relazione** (con una variante).
- Le relazioni hanno una rappresentazione naturale per mezzo di tabelle.

Una relazione è costituita su un insieme di domini, non necessariamente distinti.



Il modello relazionale è basato su valori: i riferimenti fra dati in relazioni diverse sono rappresentati tramite valori dei domini presenti nelle ennuple.

Relazioni nel modello relazionale dei dati

- A ciascun dominio associamo un nome (**attributo**), unico nella relazione, che “descrive” il ruolo del dominio.
- Nella rappresentazione tabellare, gli attributi possono essere usati come intestazioni delle colonne.

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

- L'ordinamento fra gli attributi è irrilevante: la struttura è **non posizionale**

Tabelle e relazioni

- Una tabella rappresenta una relazione se
 - i valori di ciascuna colonna sono fra loro omogenei (dallo stesso dominio)
 - le righe sono diverse fra loro
 - le intestazioni delle colonne sono diverse tra loro
- Inoltre, in una tabella che rappresenta una relazione
 - l'ordinamento tra le righe è irrilevante
 - l'ordinamento tra le colonne è irrilevante

studenti

Matricola	Cognome	Nome	Data di Nascita
6554	Rossi	Mario	5/12/1978
8765	Neri	Paolo	3/11/1976
9283	Verdi	Luisa	12/11/1979
3456	Rossi	Maria	1/2/1978

corsi

Codice	Titolo	Docente
01	Analisi	Neri
02	Chimica	Bruni
04	Chimica	Verdi

Vantaggi della struttura basata su valori

- si rappresenta solo ciò che è rilevante dal punto di vista dell'applicazione (dell'utente)
- i puntatori sono meno comprensibili per l'utente finale (senza, l'utente finale vede gli stessi dati dei programmatori)
- i puntatori sono direzionali
- senza puntatori, i dati sono portabili
- indipendenza dalle strutture fisiche (si potrebbe avere anche con puntatori di alto livello) che possono cambiare anche dinamicamente
- nota: i puntatori possono esistere a livello fisico

Esempio

- sono possibili relazioni su un solo attributo

studenti

Matricola	Cognome	Nome	Data di Nascita
6554	Rossi	Mario	5/12/1978
8765	Neri	Paolo	3/11/1976
9283	Verdi	Luisa	12/11/1979
3456	Rossi	Maria	1/2/1978

Studenti lavoratori

Matricola
6554
8765

Esempio

Vincoli di integrità

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

Matricola	Cognome	Corso	Voto	Lode
6554	Rossi	B01	28	
8765	Neri	B03	32	
9283	Bruni	B04	28	e lode
3456	Verdi	B03	30	e lode

Codice	Titolo
B01	Fisica
B02	Analisi
B03	Chimica

Vincolo di integrità

- Definizione
 - proprietà che deve essere soddisfatta dalle istanze che rappresentano informazioni corrette per l'applicazione ogni vincolo può essere visto come una funzione booleana (o un predicato) che associa ad ogni istanza il valore **vero** o **falso**.
- Tipi di vincoli:
 - vincoli intrarelazionali; casi particolari:
 - vincoli su valori (o di dominio)
 - vincoli di ennuola
 - vincoli interrelazionali

Vincoli di integrità, motivazioni

- risultano utili al fine di descrivere la realtà di interesse in modo più accurato di quanto le strutture permettano;
- forniscono un contributo verso la “qualità dei dati”
- costituiscono uno strumento di ausilio alla progettazione (vedremo la “teoria delle forme normali”)
- sono utilizzati dal sistema nella scelta della strategia di esecuzione delle interrogazioni

Nota:

- non tutte le proprietà di interesse sono rappresentabili per mezzo di vincoli esprimibili direttamente

Vincoli di ennupla

- Esprimono condizioni sui valori di ciascuna ennupla, indipendentemente dalle altre ennuple.
- Una possibile sintassi: espressione booleana (con AND, OR e NOT) di atomi che confrontano valori di attributo o espressioni aritmetiche su di essi.
- Un vincolo di ennupla è un **vincolo di dominio** se coinvolge un solo attributo
- Esempi:

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

$(\text{Voto} = 30) \text{ OR NOT } (\text{Lode} = \text{"e lode"})$

Esempio

Lordo = (Ritenute - Netto)

Identificazione delle ennuple

Matricola	Cognome	Nome	CorsoDiStudio	Data di Nascita
6554	Rossi	Mario	Informatica	5/12/1978
8765	Rossi	Mario	Informatica	3/11/1976
4723	Verdi	Laura	Meccanica	10/7/1979
9283	Verdi	Mario	Informatica	3/11/1976
3456	Rossi	Laura	Meccanica	5/12/1978

- il numero di matricola identifica gli studenti:
 - non ci sono due ennuple con lo stesso valore sull'attributo Matricola
- i dati anagrafici identificano gli studenti:
 - non ci sono due ennuple uguali su tutti e tre gli attributi Cognome, Nome e Data di Nascita

Chiave di una relazione

E' un sottoinsieme K degli attributi che soddisfa le proprietà:

unicità

in qualunque istanza di R , non possono esistere due tuple distinte la cui restrizione su K sia uguale

minimalità

non è possibile sottrarre a K un attributo senza violare la condizione di unicITÀ

In generale una relazione può avere più di una chiave

Matricola	Cognome	Nome	CorsoDiStudio	Data di Nascita
6554	Rossi	Mario	Informatica	5/12/1978
8765	Rossi	Mario	Informatica	3/11/1976
4723	Verdi	Laura	Meccanica	10/7/1979
9283	Verdi	Mario	Informatica	3/11/1976
3456	Rossi	Laura	Meccanica	5/12/1978

- Matricola è una chiave:
 - Matricola è superchiave
 - contiene un solo attributo e quindi è minimale
- Cognome, Nome, Nascita è un'altra chiave:
 - l'insieme Cognome, Nome, Nascita è superchiave
 - nessuno dei suoi sottoinsiemi è superchiave

Matricola	Cognome	Nome	CorsoDiStudio	Data di Nascita
6554	Rossi	Mario	Informatica	5/12/1978
8765	Rossi	Mario	Elettronica	3/11/1976
4723	Verdi	Laura	Meccanica	10/7/1979
9283	Verdi	Mario	Informatica	3/11/1976
3456	Rossi	Laura	Meccanica	5/12/1978

- la relazione non contiene ennuple fra loro uguali su Cognome e Corso:
 - in ogni corso di laurea gli studenti hanno cognomi diversi;
 - l'insieme { Cognome, Corso } è superchiave minimale e quindi chiave
- possiamo dire che questa proprietà è sempre soddisfatta?
 - No! In generale ci possono essere in un corso di laurea studenti con lo stesso cognome

Chiavi, schemi e istanze

- i vincoli corrispondono a proprietà del mondo reale modellato dalla base di dati;
- quindi interessano a livello di schema (con riferimento cioè a tutte le istanze):
 - ad uno schema associamo un insieme di vincoli e consideriamo **corrette** (lecite, valide, ammissibili) solo le istanze che soddisfano tutti i vincoli;
 - singole istanze possono soddisfare ulteriori vincoli (“per pura coincidenza”)

Individuazione delle chiavi

- definendo uno schema di relazione, associamo ad esso i vincoli di chiave che vogliamo siano soddisfatti dalle sue istanze (corrette)
- li individuiamo
 - considerando le proprietà che i dati soddisfano nell'applicazione (il “frammento di mondo reale di interesse”);
 - notando quali insiemi di attributi permettono di identificare univocamente le ennuple;
 - e individuando i sottoinsiemi minimali di tali insiemi che conservano la capacità di identificare le ennuple.

Individuazione delle chiavi, esempio

- Allo schema di relazione
STUDENTI(Matricola, Cognome, Nome, Nascita, Corso)
associamo i vincoli che indicano come chiavi gli insiemi di attributi Matricola e Cognome, Nome, Nascita
- La relazione

Matricola	Cognome	Nome	CorsoDiStudio	Data di Nascita
6554	Rossi	Mario	Informatica	5/12/1978
8765	Rossi	Mario	Elettronica	3/11/1976
4723	Verdi	Laura	Meccanica	10/7/1979
9283	Verdi	Mario	Informatica	3/11/1976
3456	Rossi	Laura	Meccanica	5/12/1978

- è corretta, perché soddisfa i vincoli associati allo schema.
- Ne soddisfa anche altri. Ad esempio, Cognome, Corso è chiave per essa.

Esistenza delle chiavi

- poiché le relazioni sono insiemi, ogni relazione non può contenere ennuple distinte ma uguali fra loro:
 - ogni relazione ha come superchiave l'insieme degli attributi su cui è definita;
- poiché l'insieme di tutti gli attributi è una superchiave per ogni relazione, ogni schema di relazione ha tale insieme come superchiave;
- poiché l'insieme di attributi è finito, ogni schema di relazione ha (almeno) una chiave

Importanza delle chiavi

- l'esistenza delle chiavi garantisce l'accessibilità a ciascun dato della base di dati
- ogni singolo valore è univocamente accessibile tramite:
 - nome della relazione
 - valore della chiave
 - nome dell'attributo
- le chiavi sono lo strumento principale attraverso il quale vengono correlati i dati in relazioni diverse ("il modello relazionale è basato su valori")

Chiave primaria

- La presenza di valori nulli nelle chiavi deve essere limitata
- Soluzione pratica: per ogni relazione scegliamo una chiave (la chiave primaria) su cui non ammettiamo valori nulli.
- Notazione per la chiave primaria: gli attributi che la compongono sono sottolineati

<u>Matricola</u>	Codice Fiscale	Cognome	Nome	CorsoDiStudio
6554	NULL	Rossi	NULL	Informatica
8765	NULL	Rossi	Mario	Elettronica
3562	VRDLRA76C45H501J	Verdi	Laura	Meccanica
7654	RSSMRA78B23H501K	Rossi	Mario	NULL
4443	NULL	Bruni	Laura	Meccanica

Relazioni tra attributi di tabelle diverse

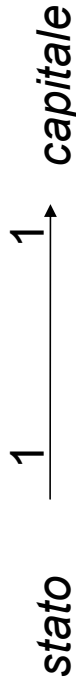
Nella definizione di un insieme di tabelle, sono riconoscibili delle relazioni esistenti tra attributi di tabelle diverse.

Queste possono essere di diverso tipo:

- **uno a molti (1:N)**



- **uno a uno (1:1)**



- **molti a molti (N:N)**



Impiegato

Matricola	Cognome	Nome	Cod. reparto
1528	Rossi	Luigi	015
2176	Bianchi	Alfredo	021
1571	Gialli	Giuseppe	003
1803	Neri	Aldo	015
2004	Rossi	Roberto	021
2910	Verdi	Giuseppe	015



Reparto

Codice	Funzione
003	Progettazione
005	Forniture
010	Produzione
015	Marketing
021	Distribuzione
025	Contr. Qualità

↑ **Chiave esterna**

Per le chiavi esterne della relazione Impiegato sono ammissibili solo valori che appaiono come chiavi primarie nella relazione Reparto.



Stato	Capitale
Francia	Parigi
Italia	Roma
Portogallo	Lisbona

Città	Milioni abitanti
Lione	0.5
Roma	2.8
Barcellona	3.1
Milano	3.0
Lisbona	1.5
Parigi	9.3

studenti

Matricola	Cognome	Nome	Data di Nascita
6554	Rossi	Mario	5/12/1978
8765	Neri	Paolo	3/11/1976
9283	Verdi	Luisa	12/11/1979
3456	Rossi	Maria	1/2/1978

N



N

esami

Studente	Voto	Corso
3456	30	04
3456	24	02
9283	28	01
6554	26	01

corsi

Codice	Titolo	Docente
01	Analisi	Neri
02	Chimica	Bruni
04	Chimica	Verdi

studenti

Matricola	Cognome	Nome	Data di Nascita
6554	Rossi	Mario	5/12/1978
8765	Neri	Paolo	3/11/1976
9283	Verdi	Luisa	12/11/1979
3456	Rossi	Maria	1/2/1978

esami

Studente	Voto	Corso
_____	30	_____
_____	24	_____
_____	28	_____
_____	26	_____

corsi

Codice	Titolo	Docente
01	Analisi	Neri
02	Chimica	Bruni
04	Chimica	Verdi

Esempi

1:N

1:1

N:N

Normalizzazione

- Non sempre la progettazione di una relazione porta ad una forma efficiente
- Ci possono essere anomalie che non sono sempre evidenti ad un primo esame, ma che pregiudicano il buon uso del database
- E' quindi necessaria una fase di analisi che porti ad una forma della relazione che risolva tali problemi (**normalizzazione**)

impiegato	telefono	stipendio	funzione	progetto	descrizione progetto
rossi	814	1800000	produzione	spazio-1	realizzazione componenti per la stazione spaziale
giordano	978	1900000	progettazione	spazio-2	progettazione componenti per la stazione
neri	312	2000000	marketing	spazio-3	analisi marketing
franco	223	1800000	produzione	spazio-1	realizzazione componenti per la stazione spaziale
franco	223	1800000	produzione	giardini spa	realizzazione zappe per giardini
barbareschi	370	1900000	progettazione	spazio-2	progettazione componenti per la stazione
milo	899	1900000	progettazione	spazio-1	realizzazione componenti per la stazione spaziale
milo	899	1800000	produzione	giardini spa	realizzazione zappe per giardini

il valore dello stipendio di ciascun impiegato è ripetuto in tutte le tuple in cui questo è presente => **ridondanza**

se lo stipendio di un impiegato varia, è necessario aggiornarlo in tutte le tuple => **anomalia di aggiornamento**

se un impiegato interrompe la partecipazione a tutti i progetti senza lasciare l'azienda, sarebbero cancellate tutte le tuple a lui relative, senza lasciare traccia dei suoi dati => **anomalia di cancellazione**

se si hanno informazioni su un nuovo impiegato, non potranno essere inserite finchè non lo si inserisce in un progetto => **anomalia di inserimento**

Problemi simili si hanno anche con la gestione dei progetti

Alcune considerazioni...

- il numero di telefono dipenda unicamente dall'impiegato
- lo stipendio di ogni impiegato dipende unicamente dalla funzione svolta
- la descrizione di un progetto dipende unicamente dal nome del progetto
- il dipartimento di appartenenza dipende dal solo impiegato
- la funzione di un impiegato all'interno di un progetto dipende dall'impiegato e dal nome del progetto

Queste proprietà si definiscono dipendenze funzionali

Impiegato	→	Telefono
Funzione	→	Stipendio
Progetto	→	Descrizione progetto
Impiegato Progetto	→	Funzione
Impiegato Progetto progetto	→	Telefono Stipendio Funzione Descrizione

Dove si presentano le anomalie ?

In corrispondenza delle relazioni che non dipendono dalla chiave

Quali sono le cause ?

Inclusione di concetti eterogenei in un'unica relazione

Possibili soluzioni ?

Le anomalie sarebbero evitate se tutti i concetti fossero omogenei, cioè se tutte le dipendenze derivassero da una chiave. In questo caso la relazione sarebbe in forma normale

La relazione non è attualmente in forma normale perché ci sono delle dipendenze che non derivano dalla chiave

Soluzione: separare la relazione

impiegato	telefono
rossi	814
giordano	978
neri	312
franco	223
barbareschi	370
milo	899

funzione	stipendio
produzione	1800000
progettazione	1900000
marketing	2000000

progetto	descrizione progetto
spazio-1	realizzazione componenti per la stazione spaziale
spazio-2	progettazione componenti per la stazione
spazio-3	analisi marketing
giardini spa	realizzazione zappe per giardini

impiegato	progetto	funzione
rossi	spazio-1	produzione
giordano	spazio-2	progettazione
neri	spazio-3	marketing
franco	spazio-1	produzione
franco	giardini	produzione
barbareschi	spazio-2	progettazione
milo	spazio-1	progettazione
milo	giardini	produzione

Nell'esempio, la decomposizione è stata molto semplice: sono state le dipendenze a suggerire la decomposizione. Purtroppo non è sempre così facile.

Tutte le decomposizioni sono ammissibili?

Solo quelle che permettono la completa ricostruzione delle informazioni originarie

Vincoli di integrità referenziale (“foreign key”)

- informazioni in relazioni diverse sono correlate attraverso valori comuni
- in particolare, valori delle chiavi (primarie, di solito)
- un **vincolo di integrità referenziale** 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

Base di dati con vincoli di integrità referenziale

infrazioni

Codice	Data	Vigile	Prov	Numero
65524	3/9/1997	343	MI	3K9886
87635	4/12/1997	476	MI	6D5563
82236	4/12/1997	343	RM	7C5567
35632	6/1/1998	476	RM	7C5567
76543	5/3/1998	548	MI	6D5563

vigili

Matricola	Cognome	Nome
343	Rossi	Luca
476	Neri	Pino
548	Nicolosi	Gino

automobili

Prov	Numero	Proprietario	...
MI	3K9886	Nestore	...
MI	6D5563	Nestore	...
RM	7C5567	Menconi	...
RM	1A6673	Mussone	...
MI	5E7653	Marchi	...

- nell'esempio, esistono vincoli di integrità referenziale fra:
 - l'attributo Vigile della relazione INFRAZIONI e la relazione VIGILI
 - gli attributi Prov e Numero di INFRAZIONI e la relazione AUTO

Base di dati che viola vincoli di integrità referenziale

infrazioni

<u>Codice</u>	Data	Vigile	Prov	Numero
65524	3/9/1997	343	MI	3K9886
87635	4/12/1997	476	MI	6D5563
82236	4/12/1997	343	RM	7C5567
35632	6/1/1998	476	RM	7C5567
76543	5/3/1998	548	MI	6D5563

vigili

<u>Matricola</u>	Cognome	Nome
343	Rossi	Luca
548	Nicolosi	Gino

automobili

<u>Prov</u>	<u>Numero</u>	Proprietario	...
MI	3K9886	Nestore	...
RM	6D5563	Nestore	...
MI	7C5567	Menconi	...
RM	1A6673	Mussone	...
MI	5E7653	Marchi	...

Vincoli di integrità referenziale: commenti

- I vincoli di integrità referenziale giocano un ruolo fondamentale nel concetto “modello relazionale basato su valori.”
- Sono possibili meccanismi per il supporto alla gestione dei vincoli di integrità referenziale (“azioni” da svolgere in corrispondenza a violazioni).
- In presenza di valori nulli i vincoli possono essere resi meno restrittivi
- Attenzione ai vincoli su più attributi

Base di dati con vincoli di integrità referenziale

incidenti

<u>Codice</u>	Data	ProvA	NumeroA	ProvB	NumeroB
65524	3/9/1997	MI	3K9886	RM	7C5567
87635	4/12/1997	RM	6D5563	RM	1A6673
82236	6/12/1997	MI	7C5567	RM	6D5563

automobili

<u>Prov</u>	<u>Numero</u>	Proprietario	...
MI	3K9886	Nestore	...
RM	6D5563	Nestore	...
MI	7C5567	Menconi	...
RM	1A6673	Mussone	...
MI	5E7653	Marchi	...

Esempio

Gestione Integrità referenziale

Operazioni sulle relazioni

Per estrarre informazioni dal database, è possibile effettuare particolari operazioni. Le principali sono:

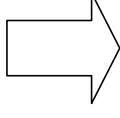
- **selezione**
- **proiezione**
- **join**

Selezione

Applicata ad una certa relazione, costruisce una nuova relazione formata dalle tuple che soddisfano una particolare condizione

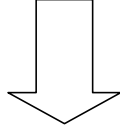
impiegato	progetto	funzione
rossi	spazio-1	produzione
giordano	spazio-2	progettazione
neri	spazio-3	marketing
franco	spazio-1	produzione
franco	giardini	produzione
barbareschi	spazio-2	progettazione
milo	spazio-1	progettazione
milo	giardini	produzione

“Fornire gli impiegati ed i progetti in cui gli impiegati hanno funzione di progettazione”



impiegato	progetto	funzione
giordano	spazio-2	progettazione
barbareschi	spazio-2	progettazione
milo	spazio-1	progettazione

Seleziona le tuple aventi attributo funzione con valore progettazione



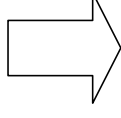
Esempio

Proiezione

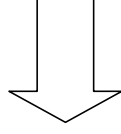
Applicata ad una certa relazione, costruisce una nuova relazione formata da un sottinsieme definito di attributi

impiegato	progetto	funzione
rossi	spazio-1	produzione
giordano	spazio-2	progettazione
neri	spazio-3	marketing
franco	spazio-1	produzione
franco	giardini	produzione
barbareschi	spazio-2	progettazione
milo	spazio-1	progettazione
milo	giardini	produzione

“Fornire gli impiegati ed i progetti in cui sono impegnati”



impiegato	progetto
rossi	spazio-1
giordano	spazio-2
neri	spazio-3
franco	spazio-1
franco	giardini
barbareschi	spazio-2
milo	spazio-1
milo	giardini



Proietta gli attributi **impiegato** e **progetto**

[Esempio](#)

Join

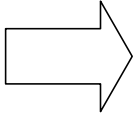
È un operatore binario che permette di correlare dati contenuti in relazioni diverse, confrontando i valori contenuti in esse. Ci sono due tipi di join: **join naturale** e **theta-join**.

Join naturale

Applicato a due relazioni, costruisce la relazione formata dalle tuple ottenute combinando le tuple degli operandi con valori uguali sugli attributi comuni.

Esempio

“Fornire l’elenco degli impiegati, i progetti su cui sono impegnati, la funzione svolta e lo stipendio percepito”

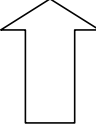


impiegato	progetto	funzione
rossi	spazio-1	produzione
giordano	spazio-2	progettazione
neri	spazio-3	marketing
franco	spazio-1	produzione
franco	giardini	produzione
barbareschi	spazio-2	progettazione
milo	spazio-1	progettazione
milo	giardini	produzione

funzione	stipendio
produzione	1800000
progettazione	1900000
marketing	2000000

Join naturale delle relazioni
**(impiegato progetto
funzione) e (funzione
stipendio)**

Esempio



impiegato	progetto	funzione	stipendio
rossi	spazio-1	produzione	1800000
giordano	spazio-2	progettazione	1900000
neri	spazio-3	marketing	2000000
franco	spazio-1	produzione	1800000
franco	giardini	produzione	1800000
barbareschi	spazio-2	progettazione	1900000
milo	spazio-1	progettazione	1900000
milo	giardini	produzione	1800000

Che cosa succede se il database avesse avuto un'organizzazione leggermente diversa ?

impiegato	progetto	funzione
rossi	spazio-1	produzione
giordano	spazio-2	progettazione
neri	spazio-3	marketing
franco	spazio-1	produzione
franco	giardini	produzione
barbareschi	spazio-2	progettazione
milo	spazio-1	progettazione
milo	giardini	produzione

ruolo	stipendio
produzione	1800000
progettazione	1900000
marketing	2000000

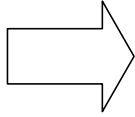
Non è possibile applicare il join naturale perché gli attributi hanno un nome diverso, anche se hanno lo stesso significato.

In questo caso si impiega il **theta-join**.

Applicato a due relazioni, il **theta-join** costruisce la relazione formata dalle tuple ottenute combinando le tuple degli operandi per cui è soddisfatta una condizione definita sui valori dei rispettivi attributi.

Esempio

“Fornire l’elenco degli impiegati, i progetti su cui sono impegnati, la funzione svolta e lo stipendio percepito”



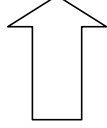
impiegato	progetto	funzione
rossi	spazio-1	produzione
giordano	spazio-2	progettazione
neri	spazio-3	marketing
franco	spazio-1	produzione
franco	giardini	produzione
barbareschi	spazio-2	progettazione
milo	spazio-1	progettazione
milo	giardini	produzione

ruolo	stipendio
produzione	1800000
progettazione	1900000
marketing	2000000

Theta-join delle relazioni
**(impiegato progetto
funzione) e (ruolo stipendio)**
con i valori degli attributi
funzione e ruolo uguali

Esempio

impiegato	progetto	funzione	ruolo	stipendio
rossi	spazio-1	produzione	produzione	1800000
giordano	spazio-2	progettazione	progettazione	1900000
neri	spazio-3	marketing	marketing	2000000
franco	spazio-1	produzione	produzione	1800000
franco	giardini	produzione	produzione	1800000
barbareschi	spazio-2	progettazione	progettazione	1900000
milo	spazio-1	progettazione	progettazione	1900000
milo	giardini	produzione	produzione	1800000



Interrogazioni sui database

Avendo a disposizione gli operatori visti, è possibile trasformare una qualunque interrogazione sul database in una combinazione di operatori applicati sulle relazioni definite nel database.

Esempio

“Trovare nomi e telefoni degli impiegati impegnati nel progetto spazio-1”

impiegato	telefono
rossi	814
giordano	978
neri	312
franco	223
barbareschi	370
milo	899

funzione	stipendio
produzione	1800000
progettazione	1900000
marketing	2000000

progetto	descrizione progetto
spazio-1	realizzazione componenti per la stazione spaziale
spazio-2	progettazione componenti per la stazione
spazio-3	analisi marketing
giardini spa	realizzazione zappe per giardini

impiegato	progetto	funzione
rossi	spazio-1	produzione
giordano	spazio-2	progettazione
neri	spazio-3	marketing
franco	spazio-1	produzione
franco	giardini	produzione
barbareschi	spazio-2	progettazione
milo	spazio-1	progettazione
milo	giardini	produzione

Join naturale delle relazioni
**(impiegato progetto
funzione)**
e **(impiegato telefono)**

impiegato	progetto	funzione	telefono
rossi	spazio-1	produzione	814
giordano	spazio-2	progettazione	978
neri	spazio-3	marketing	312
franco	spazio-1	produzione	223
franco	giardini	produzione	223
barbareschi	spazio-2	progettazione	370
milo	spazio-1	progettazione	899
milo	giardini	produzione	899

Seleziona le tuple aventi
attributo **progetto** con valore
spazio-1

impiegato	progetto	funzione	telefono
rossi	spazio-1	produzione	814
franco	spazio-1	produzione	223
milo	spazio-1	progettazione	899

Proietta gli attributi **impiegato**
e **telefono**

impiegato	telefono
rossi	814
franco	223
milo	899