Database

Modulo 5

C. Marrocco

Università degli Studi di Cassino

Database: definizione

(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.

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.

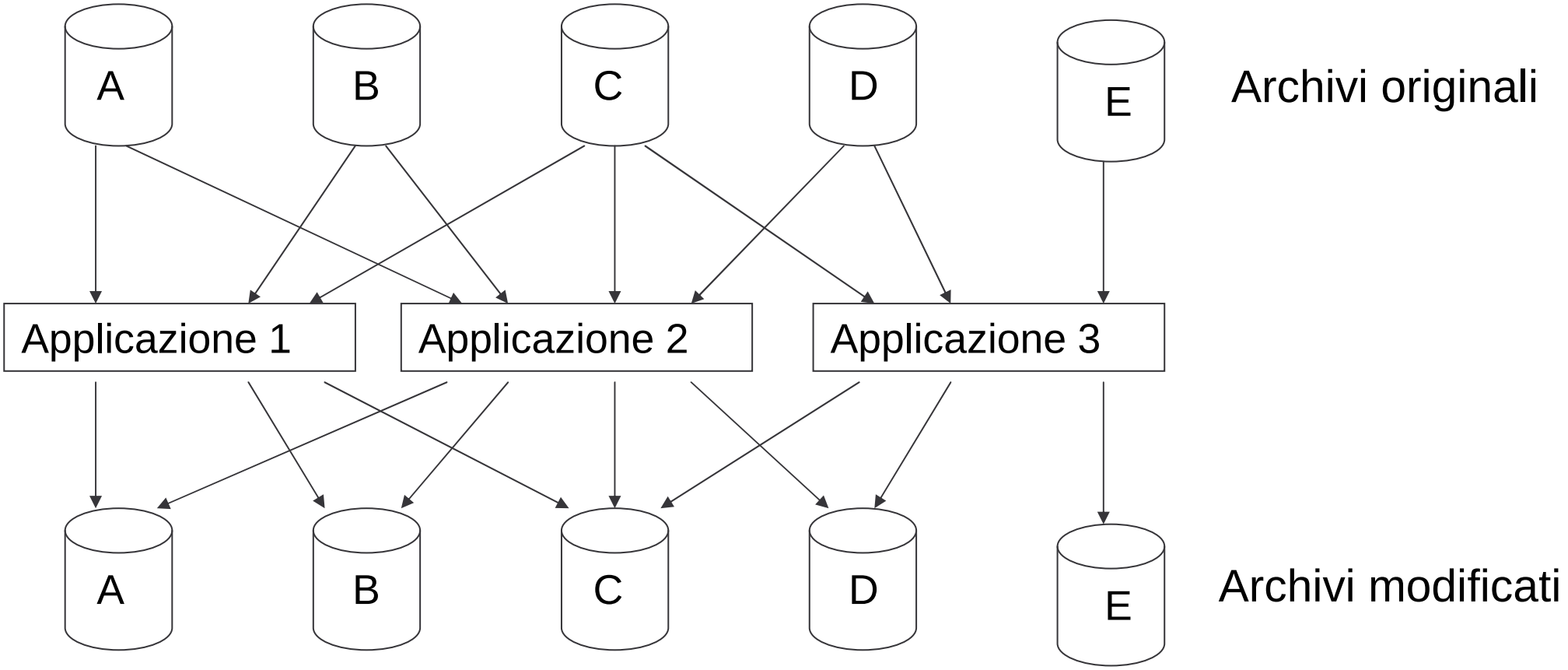
DBMS vs. file system

- La gestione di insiemi di dati grandi e persistenti è possibile anche attraverso i file system dei sistemi operativi, che permettono di realizzare anche rudimentali forme di condivisione.
- 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 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.

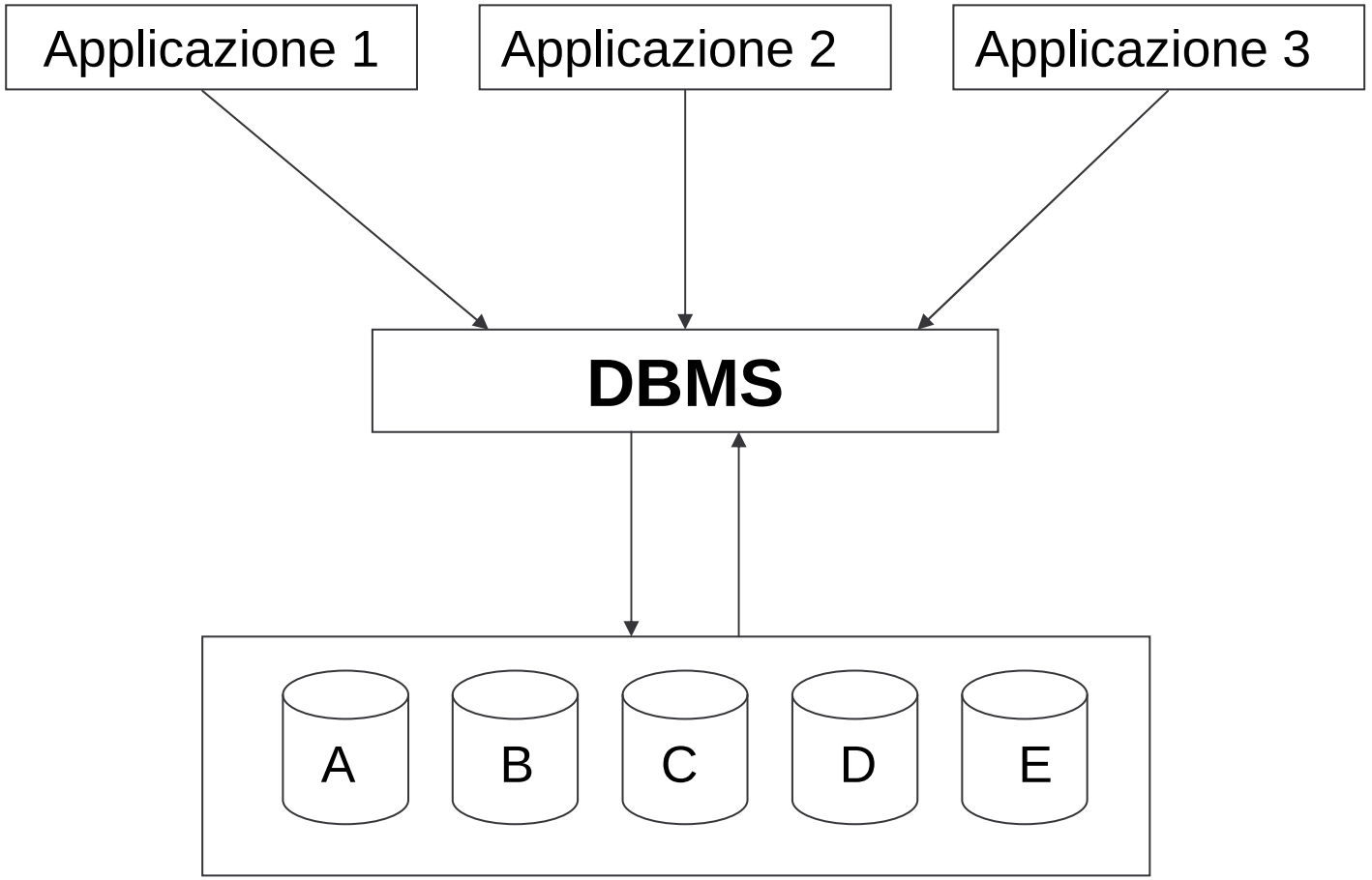
DBMS vs. file system

- I DBMS estendono le funzionalità dei file system, fornendo più servizi in maniera integrata (maggiore efficacia).
- Nei DBMS c'è maggiore flessibilità: si può accedere contemporaneamente a record diversi di uno stesso file o addirittura allo stesso record (in lettura).
- 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.

DBMS vs. file system



DBMS vs. file system



Vantaggi e svantaggi del DBSM

Pro

- Dati come risorsa comune, database 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 scorporabilità 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.

- Possibile 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 insieme di record omogenei.

I modelli logici dei file

- 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 ma reso disponibile come modello logico in DBMS reali solo nel 1981 (non è semplice implementare l'indipendenza con efficienza e affidabilità).
- Si basa sul concetto matematico di relazione.
- Le relazioni hanno una rappresentazione naturale per mezzo di tabelle.

Relazioni nel modello relazionale

- 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 tra 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.
- I riferimenti fra dati in relazioni diverse sono rappresentati per mezzo di valori dei domini che compaiono nelle tuple.

Esempio

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

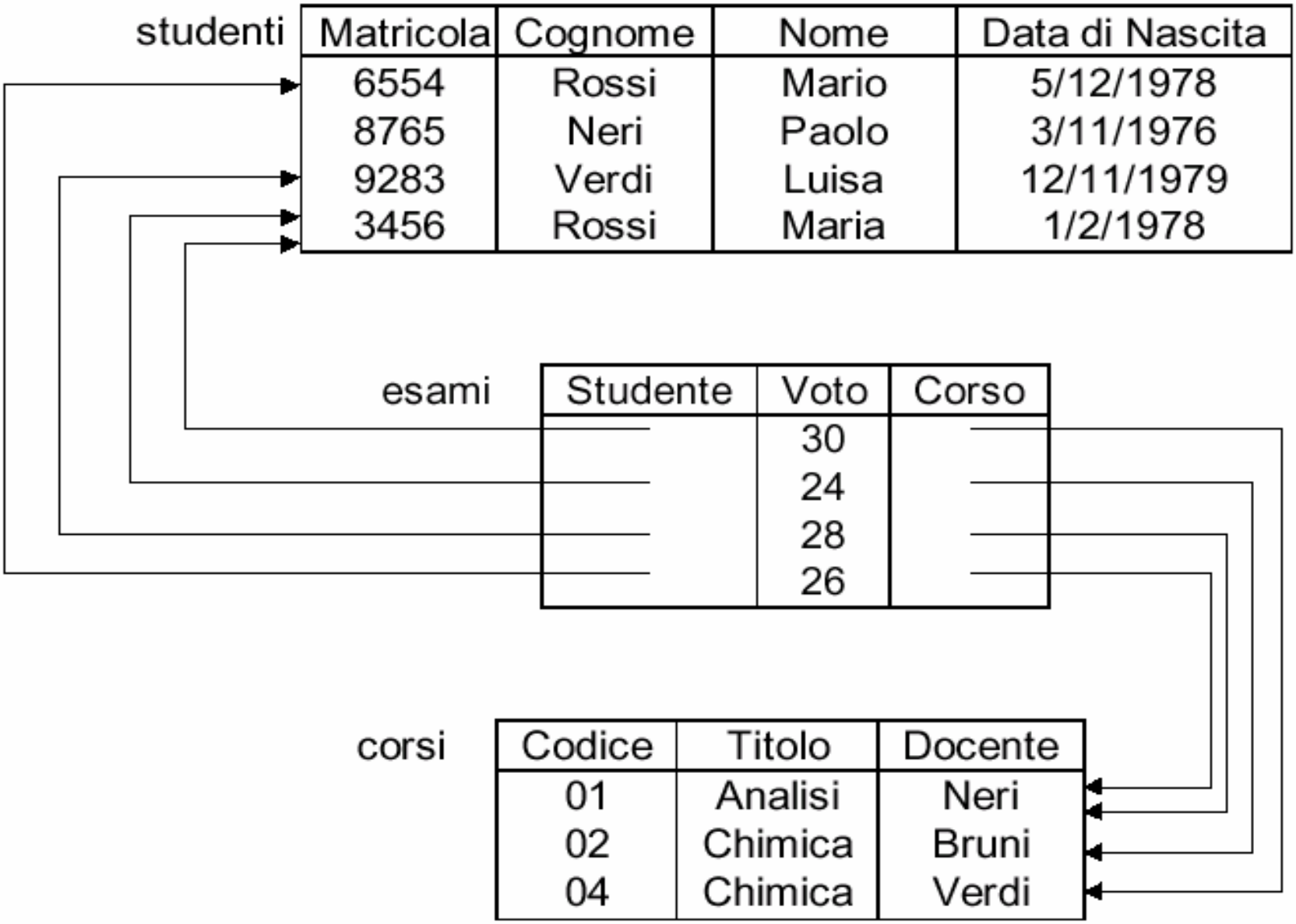
esami

Studente	Voto	Corso
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

Esempio



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

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 mentre senza puntatori l'utente finale vede gli stessi dati dei programmatori.
- I puntatori sono direzionali mentre senza puntatori i dati sono portabili.
- Indipendenza dalle strutture fisiche che possono cambiare anche dinamicamente.

Relazioni tra attributi di tabelle diverse

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

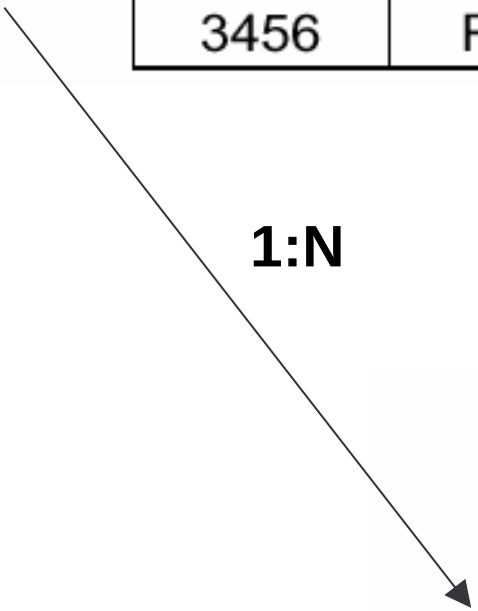
Queste possono essere di diverso tipo:

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

Esempio

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



1:N

esami

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

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

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

Vincoli 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 che associa ad ogni istanza il valore vero o falso.

Tipi di vincoli:

- Vincoli intrarelazionali: casi particolari sono i vincoli su valori (o di dominio) e i vincoli di tupla.
- Vincoli interrelazionali.

Vincoli di integrità

- Sono 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.
- 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 tupla

- Esprimono condizioni sui valori di ciascuna tupla, indipendentemente dalle altre tuple.
- 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 tupla è un vincolo di dominio se coinvolge un solo attributo.
- **Esempi**:

(Voto ≥18) AND (Voto ≤30)

(Voto =30) OR NOT (Lode = “e lode”)

Lordo = (Ritenute - Netto)

Identificazione delle tuple

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 tuple con lo stesso valore sull'attributo Matricola.

I dati anagrafici identificano gli studenti:

- Non ci sono due tuple 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.

Esempio

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, Data di Nascita} è un'altra chiave:

- L'insieme Cognome, Nome, Data di Nascita è superchiave.
- Nessuno dei suoi sottoinsiemi è superchiave.

Esempio

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 tuple fra loro uguali su Cognome e Corso:

- In ogni corso di laurea gli studenti hanno cognomi diversi.
- L'insieme {Cognome, CorsoDiStudio} 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.
- Interessano a livello di schema (con riferimento cioè a tutte le istanze):
  - Ad uno schema associamo un insieme di vincoli e consideriamo corrette solo le istanze che soddisfano tutti i vincoli.
  - Singole istanze possono soddisfare ulteriori vincoli.

Individuazione delle chiavi

- Definendo uno schema di relazione, associamo ad esso i vincoli di chiave che vogliamo siano soddisfatti dalle sue istanze.
- Li individuiamo:
  - Considerando le proprietà che i dati soddisfano nell'applicazione.
  - Notando quali insiemi di attributi permettono di identificare univocamente le tuple.
  - Individuando i sottoinsiemi minimali di tali insiemi che conservano la capacità di identificare le tuple

Esempio

- Allo schema di relazione

  STUDENTI (Matricola, Cognome, Nome, Data di Nascita, CorsoDiStudio)

  associamo i vincoli che indicano come chiavi gli insiemi di attributi {Matricola} e {Cognome, Nome, Data di 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, CorsoDiStudio} è chiave per essa.

Esistenza delle chiavi

- Poiché le relazioni sono insiemi, ogni relazione non può contenere tuple distinte ma uguali tra 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 del database.
- Ogni singolo valore è univocamente accessibile tramite:
  - Nome della relazione.
  - Valore della chiave.
  - Nome dell’attributo.
- Le chiavi sono lo strumento principale attraverso cui vengono correlati i dati in relazioni diverse

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 nelle tabelle: gli attributi che la compongono sono sottolineati

Matricola	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

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).

Esempio

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**

Esempio

- Il numero di telefono dipende 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.

Esempio

Impiegato	$\rightarrow$	Telefono
Funzione	$\rightarrow$	Stipendio
Progetto	$\rightarrow$	Descrizione progetto
Impiegato Progetto	$\rightarrow$	Funzione
Impiegato Progetto	$\rightarrow$	Telefono, Stipendio, Funzione, Descrizione progetto

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.

Esempio

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

Esempio

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

Esempio

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: la Chiave Esterna

- Informazioni in relazioni diverse sono correlate attraverso valori comuni (in particolare, attraverso valori delle chiavi primarie)
- 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 tupla dell'istanza di R_1 di comparire come valori della chiave (primaria) dell'istanza di R_2. Si parla in questo caso di "foreign key" (chiave esterna).

Esempio

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.

Esempio

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

Viola i vincoli di integrità referenziale

vigili

Matricola	Cognome	Nome
343	Rossi	Luca
548	Nicolosi	Gino

automobili

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

Vincoli di integrità referenziale

- I vincoli di integrità referenziale giocano un ruolo fondamentale nel concetto di modello relazionale.
- 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.

Esempio: con vincoli di integrità referenziale

incidenti

Codice	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

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