Database
Il modello relazionale

Persistenza

- Quando un programma termina le informazioni contenute nelle variabili in memoria vengono perse
- Molto spesso però le applicazioni hanno la necessità di elaborare e utilizzare informazioni che devono essere conservate alla fine dell'esecuzione per essere recuperate in una sessione successiva
- Hanno cioè bisogno di dati persistenti
- Un semplice strumento per realizzare la persistenza è costituito dai file, binari o di testo
- Abbiamo infatti visto che un'applicazione Java, utilizzando gli stream può leggere e scrivere informazioni su i file

Database

- Tuttavia se le informazioni sono molte o se hanno una struttura e un'organizzazione complessa i file di testo si rivelano insufficienti
- E' necessario quindi utilizzare strumenti più potenti, come i database
- Che cos'è un database?
- Una possibile definizione è la seguente:

  Un database è una raccolta di informazioni organizzata in modo tale da poter essere facilmente accessibile per consultazione, modifiche e aggiornamenti.

Modelli

- Esistono diversi modelli di database:
  - Gerarchico
  - Reticolare
  - Relazionale
  - Ad oggetti
- Il modello relazionale è quello a più larga diffusione: praticamente tutti database comunemente usati sono relazionali
- 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, che però non è ancora molto diffuso

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
- Si basa su due concetti fondamentali:
- Relazione: concetto matematico derivato dalla teoria degli insiemi => Potenza
- Tabella: concetto semplice ed intuitivo => Semplicità
- I due concetti sono riconducibili l'uno all'altro e questo ha decretato il successo del modello: anche gli utenti finali ne comprendono facilmente il significato
- Indipendenza dei dati: gli utenti e i programmatori vedono solo il livello logico (ciò non era possibile con gli altri modelli)

Il concetto di relazione

- Nella teoria degli insiemi viene definita un'operazione chiamata prodotto cartesiano
- Dati due insiemi, A e B, il prodotto cartesiano A x B è l'insieme di tutte le possibili coppie ordinate che si possono costruire con gli elementi dei due insiemi
- Per esempio:

```
Se A = {a1,a2,a3} e B = {b1,b2}
A x B = {(a1,b1),(a1,b2),(a2,b1),
         (a2,b2),(a2,b1),(a3,b2)}
```

- Si definisce relazione (in senso matematico) un qualunque sottoinsieme di un prodotto cartesiano
- Per esempio:

```
R1 = {(a1,b2),(a2,b2),(a3,b1)}
R2 = {(a1,b1),(a3,b1)}
```

Relazioni e tabelle

- Una relazione può essere rappresentata sotto forma di tabella

```
R1 = {(a1,b2),(a2,b2),(a3,b1)}
```

a1	b2
a2	b2
a3	b1

```
R2 = {(a1,b1),(a3,b1)}
```

a1	b1
a3	b1

Domini

- Gli insiemi A e B prendono il nome di domini della relazione
- I domini possono essere finiti (per esempio i codici postali italiani) oppure infiniti (tutti i numeri interi)
- Se i domini sono infiniti il loro prodotto cartesiano è infinito
- Dal momento che i computer possono gestire solo quantità finite di informazioni le relazioni devono essere sottoinsiemi finiti del prodotto cartesiano
- E' comunque utile ammettere domini infiniti per permettere ad ogni istante di assumere che possa esistere un valore non presente nel DB

N-uple

- Il prodotto cartesiano può essere esteso ad un numero qualunque di insdiemi: A x B x C x D x …
  - Con 2 insiemi il risultato è un insieme di coppie
  - Con 3 insiemi il risultato è un insieme di terne
  - Con n insiemi il risultato è un insieme di n-uple
- Una n-upla (si legge ennupla) è quindi una combinazione di n valori e si rappresenta con una tabella formata da n colonne
- Gli elementi che compongono una n-upla vengono chiamati componenti dell'n-upla
- Il numero dei componenti di ogni n-upla (numero delle colonne della tabella) viene chiamato grado
- Il numero di n-uple presenti nella relazione (numero di righe della tabella) prende il nome di cardinalità

Esempio di relazione

- I risultati delle partite di un campionato di calcio costituiscono delle n-uple (a 4 valori)
- Un qualunque gruppo di risultati è una relazione
- I domini sono l'insieme delle squadre (S) e l'insieme dei numeri interi G (gol segnati)
- Dal momento che nell'n-upla abbiamo sia il nome che i gol di due squadre ognuno dei domini apparirà due volte nel prodotto cartesiano: S x S x G x G
- Una possibile relazione, di grado 4 e cardinalità 4, rappresentata sotto forma di tabella è:

Juventus	Lazio	3	2
Lazio	Milan	2	0
Juventus	Roma	2	1
Roma	Milan	1	2

Caratteristiche delle relazioni

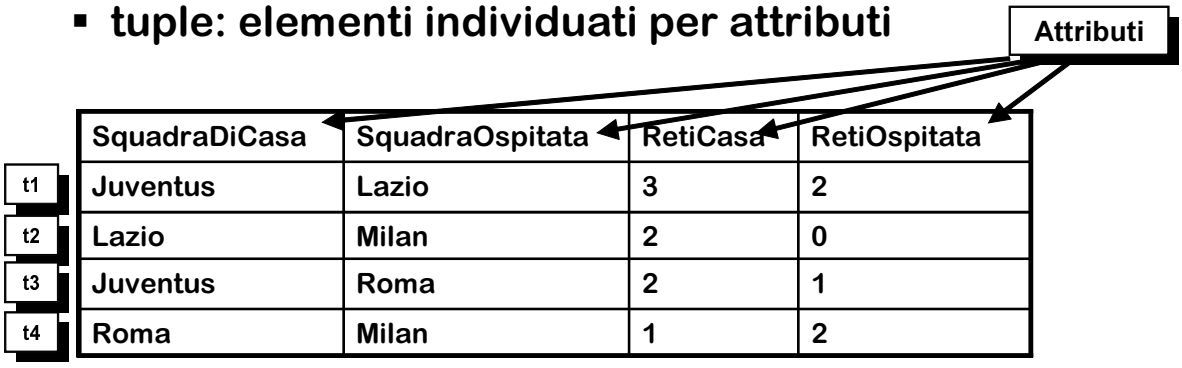
- Le n-uple sono ordinate: l' ordine dei loro elementi è significativo
- Per esempio: (Juventus,Lazio,3,2) significa che il risultato della partita Juventus-Lazio, giocata in casa dalla Juventus, è 3 a 2
- Non è la stessa cosa che (Juventus,Lazio,2,3)!
- Inoltre una relazione è un insieme, quindi:
  - Le n-uple della relazione devono essere distinte (non è possibile avere due volte la stessa riga)
  - Le n-uple non sono tra loro ordinate (tabelle con stesse righe ordinate in modo diverso rappresentano la stessa relazione)
- L'ordine delle colonne è importante, l'ordine delle righe no

Relazioni con attributi

- I modello relazionale utilizza in realtà una variante del concetto di relazione matematica
- L'ordinamento dei domini di una relazione impone ordinamento posizionale degli elementi di n-uple
- Nella gestione di dati si preferisce utilizzare ordinamenti non posizionali in cui si può far riferimento alle componenti delle n-uple in modo non ambiguo
- Si associa a ciascun insieme del prodotto cartesiano un nome, chiamato attributo, che descrive il ruolo giocato
- Nel nostro esempio: SquadraDiCasa, SquadraOspitata, RetiCasa, RetiOspitata
- Ogni colonna della tabella ha quindi un'intestaziione

N-uple e tuple

- In una relazione con attributi gli elementi vengono chiamati tuple
  - n-uple: elementi individuati per posizione
  - tuple: elementi individuati per attributi



	SquadraDiCasa	SquadraOspitata	RetiCasa	RetiOspitata
t1	Juventus	Lazio	3	2
t2	Lazio	Milan	2	0
t3	Juventus	Roma	2	1
t4	Roma	Milan	1	2

t1[SquadraDiCasa]=Juventus t1[SquadraOspitata]=Lazio t1[RetiCasa]=3 t1[RetiOspitata]=2	t2[SquadraDiCasa]=Lazio t2[SquadraOspitata]=Milan t2[RetiCasa]=2 t2[RetiOspitata]=0

Relazioni e Basi di Dati

- Un database (DB) è solitamente costituito da più relazioni (tabelle) le cui tuple contengono valori comuni (usati per stabilire corrispondenza tra tuple)
- Il modello relazionale è infatti basato esclusivamente sui valori
- I riferimenti fra dati in relazioni diverse sono rappresentati per mezzo di valori che compaiono nelle tuple
- Non esistono puntatori o altri meccanismi nascosti
- Abbiamo quindi un'indipendenza die dati dalle strutture fisiche

Esempio di database

- Il database Carriere gestisce le carriere degli studenti
- E' costituito da 3 tabelle: studenti, esami, corsi:

Studenti

Matricola	Cognome	Nome	DataNascita
276545	Rossi	Maria	25/11/1981
485745	Neri	Anna	23/04/1982
200768	Verdi	Fabio	12/02/1982
587614	Rossi	Luca	10/10/1981
937653	Bruni	Mario	01/12/1981

Valori comuni

Esami

Studente	Voto	Corso
276545	28	01
485745	27	04
200768	25	01
587614	24	04

Valori comuni

Corsi

Codice	Titolo	Docente
01	Analisi	Giani
03	Chimica	Melli
04	Chimica	Belli

Schemi di relazione e di DB

- Per ogni relazione (tabella) abbiamo uno schema
- Lo schema è costituito dal nome della relazione e da insieme di nomi di attributi
- Per esempio: Esami(Studente,Voto,Corso)
- L'insieme degli schemi di tutte le relazioni di un DB è lo schema del database:
- Nel nostro esempio:

  Carriere =
  {Studenti(Matricola,Cognome,Nome,DataNascita),
  Esami(Studente,Voto,Corso),
  Corso(Codice,Titolo,Docente)}

Terminologia

- Nel parlare di database si una una terminologia formale, derivata dal modello matematico, e una informale, basata sull'uso pratico

Terminologia formale	Terminologia informale
Relazione	Tabella
Attributo	Colonna
Tupla / n-upla	Riga
Dominio	Tipo di dato
Cardinalità	Numero di righe
Grado	Numero di colonne

Esempio: Pizzeria Vesuvio - 1

- La pizzeria Vesuvio vuole memorizzare le ricevute fiscali emesse
- Le ricevute sono fatte in questo modo:

Pizzeria Vesuvio		
Ricevuta n. 1357		
Del 5/2/04		
3	coperti	3,00
2	antipasti	6,00
3	primi	12,00
2	bistecche	18,00
Totale		39,00

Pizzeria Vesuvio		
Ricevuta n. 2334		
Del 7/2/04		
2	coperti	2,00
1	antipasti	3,00
2	primi	8,00
2	orate	14,00
2	caffè	2,00
Totale		29,00

Pizzeria Vesuvio		
Ricevuta n. 3002		
Del 13/2/04		
3	coperti	3,00
2	antipasti	6,00
3	primi	14,00
1	Orate	18,00
1	Caprese	2,00
2	Caffè	2,00
Totale		45,00

Esempio: Pizzeria Vesuvio - 2

- Le ricevute hanno una struttura che prevede
  - Alcune informazioni fisse: numero, data e totale
  - Un numero variabile di righe
- Non è possibile rappresentare correttamente l'insieme delle ricevute con un'unica relazione
- Infatti non si riuscirebbe a gestire un numero non predeterminato di righe
- Creeremo quindi un database composo da due relazioni: Ricevute e Righe
- Lo schema sarà:

  Vesuvio = {Ricevute(Num,Data,Totale), Righe(Num,Quant,Descr,Importo)}

Esempio: Pizzeria Vesuvio - 3

- Ecco qui le tabelle:

Righe

Num	Quant	Descr	Importo
1357	3	Coperti	3,00
1357	2	Antipasti	6,00
1357	3	Primi	12,00
1357	2	Bistecche	18,00
2334	2	Coperti	2,00
2334	1	Antipasti	3,00
2334	2	Primi	8,00
2334	2	Orate	14,00
2334	2	Caffè	2,00
3002	3	Coperti	3,00
3002	2	Antipasti	6,00
3002	3	Primi	14,00
3002	1	Orate	18,00
3002	1	Caprese	2,00
3002	2	Caffè	2,00

Ricevute

Num	Data	Totale
1357	5/2/04	39,00
2334	7/2/04	29,00
3002	13/2/04	45,00

Esempio: Pizzeria Vesuvio - 4

- La base di dati nella slide precedente rappresenta correttamente le ricevute solo a due condizioni:
  - Non interessa mantenere traccia dell'ordine con cui le righe compaiono in ciascuna ricevuta
  - In un ricevuta non compaiono due righe uguali
- In entrambi i casi, si può risolvere il problema aggiungendo un attributo, che indica la posizione della riga sulla ricevuta:

Num	Riga	Quant	Descr	Importo
1357	1	3	Coperti	3,00
1357	2	2	Antipasti	6,00
1357	3	3	Primi	12,00
1357	4	2	Bistecche	18,00
2334	1	2	Coperti	2,00

Informazione incompleta e valori nulli

- In una tupla di una relazione un attributo può non avere valore:
- Per esempio: Mario Rossi non ha telefono in Persone(Cognome,Nome,Indirizzo,Telefono)
- Oppure il valore di un attributo potrebbe esistere ma essere sconosciuto a chi inserisce i dati nel DB
- Il modello relazionale prevede un valore speciale per questi casi: NULL (valore nullo)
- Viene assegnato agli elementi di tuple che risultano inesistenti o sconosciuti
- NULL è valore aggiuntivo rispetto al dominio di un attributo:

💣 Attenzione: non equivale a 0 per i numeri o "" per le stringhe è diverso da qualunque altro valore

Identificazione di una tupla

- Abbiamo visto che una tupla è unica
- Ma non può essere identificata dalla sua posizione che non è rilevante in una relazione
- Come facciamo quindi ad identificare in modo preciso una tupla?
- Si ricorre al concetto di chiave
- Una chiave è un sottoinsieme degli attributi dello schema che ha la proprietà di unicità e minimalità
  - Unicità: non esistono due tuple con chiave uguale
  - Minimalità: sottraendo un qualunque attributo alla chiave si perde la proprietà di unicità
- Se si ha solo l'unicità ma non la minimalità si ha una superchiave

Chiavi: esempio - 1

- Riprendiamo l'esempio degli studenti

Matricola	Cognome	Nome	Corso	Nascita
27655	Rossi	Mario	INF	5/12/78
78763	Rossi	Mario	INF	3/11/76
65432	Neri	Piero	ELE	10/7/79
87654	Neri	Mario	TLC	3/11/76
67653	Rossi	Piero	ELE	5/12/78

- Non ci sono due ennuple con lo stesso valore sull'attributo Matricola
- Non ci sono due ennuple uguali su tutti e tre gli attributi Cognome, Nome e Data di Nascita

Chiavi: esempio - 2

- {Matricola} è una chiave:
  - Garantisce l'unicità: non esistono due studenti con la stessa matricola
  - Contiene un solo attributo e quindi è minimale
- {Cognome, Nome, Nascita} è un'altra chiave:
  - Garantisce l'unicità: non ci sono due studenti con lo stesso nome, cognome e data di nascita
  - E' minimale:
    - Se togliamo la data di nascita abbiamo due Rossi Mario
    - Se togliamo il cognome abbiamo due Mario nati nello stesso giorno
    - Se togliamo il nome abbiamo due Rossi nati nello stesso giorno

Chiave primaria

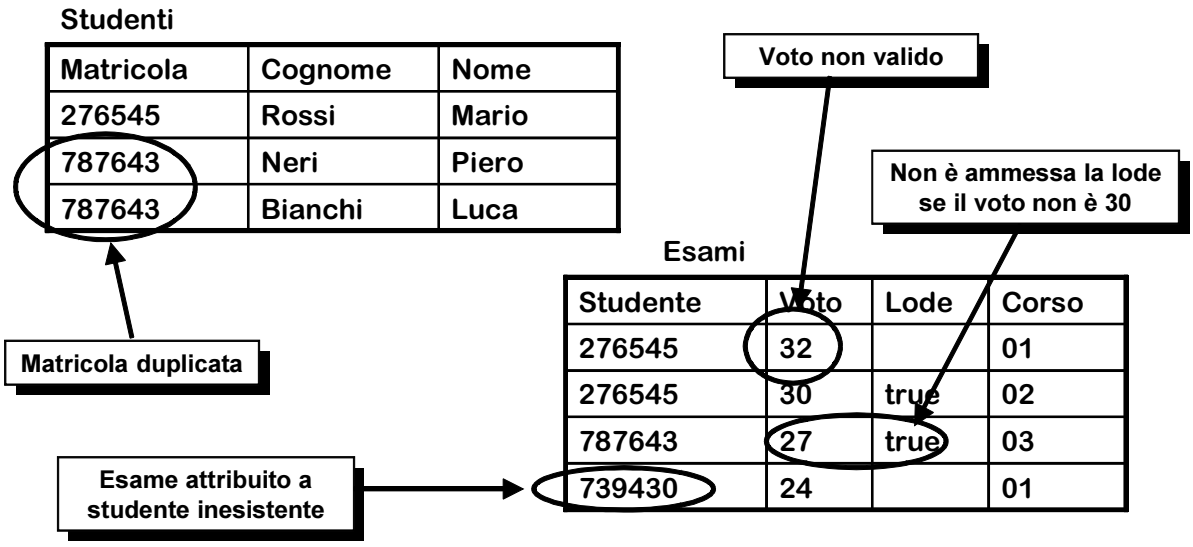
- Si definisce come chiave primaria una chiave per cui non sono ammessi valori nulli in nessuno dei suoi componenti
- Se modifichiamo la tabella dell'esempio in questo modo:

<u>Matricola</u>	Cognome	Nome	Corso	Nascita
27655	Rossi	Mario	INF	5/12/78
78763	Rossi	Mario	INF	3/11/76
65432	Neri	Piero	ELE	null
87654	Neri	Mario	TLC	3/11/76
67653	Rossi	Piero	ELE	5/12/78

- {Nome, Cognome, Nascita} non può essere una chiave primaria mentre {Matricola} lo è
- Si evidenzia sottolineando gli attributi che la compongono
- Le chiavi non primarie vengono anche dette secondarie

Integrità

- Un database, pur sintatticamente corretto, può non esserlo semanticamente se non è garantita l'integrità
- Per esempio questo database contiene diverse violazioni di integrità:



Vincoli di integrità

- Le violazioni di integrità possono rendere inutilizzabile un database
- Fortunatamente il modello relazionale prevede la possibilità di definire vincoli di integrità il cui rispetto viene garantito dal database stesso
- Vincolo di integrità: proprietà che deve essere soddisfatta da tutte le entità che rappresentano informazioni corrette per una data applicazione
- Si dividono in;
  - Vincoli intrarelazionali: il loro soddisfacimento è definito all'interno di una singola relazione
  - Vincoli interrelazionali: coinvolgono più relazioni

Vincoli intrarelazionali e chiavi

- Sono sostanzialmente di due tipi:
  - Chiavi
  - Vincoli di tupla
- Delle chiavi abbiamo già parlato: il loro compito è garantire l'unicità di una tupla in una relazione
- Nel nostro esempio se dichiariamo che la Matricola è la chiave primaria due studenti non potranno avere la stessa matricola

Vincoli di tupla

- I vincoli di tupla riguardano i valori che possono assumere i campi di una tupla
- Nel caso più semplice possiamo definire un vincolo relativo ad un solo attributo: vincolo di dominio
- In generale possiamo esprimere vincoli relativi a più attributi
- Nella relazione Esami i vincoli di tupla saranno:
  - Voto compreso fra 18 e 30 (vincolo di dominio):
    `(Voto>=18) AND (Voto<=30)`
  - Lode ammessa solo se il voto è 30:
    `(NOT(Lode=true)) OR (Voto=30)`
- Un caso particolare di vincolo di domino è costituito dalla clausola `NOT NULL` che vincola un campo ad avere un valore definito

Integrità referenziale - 1

- informazioni in relazioni diverse sono correlate attraverso valori comuni e in particolare i valori delle chiavi (primarie o secondarie)
- Queste correlazioni debbono essere "coerenti"
- Lo strumento per esprimere i vincoli interrelazionali e garantire questa coerenza è l'integrità referenziale (foreign key)
- Un vincolo di integrità referenziale fra gli attributi X di una relazione R1 e un'altra relazione R2 impone ai valori su X in R1 di comparire come valori della chiave primaria di R2
- Nel nostro esempio un vincolo di integrità referenziale potrebbe imporre all'attributo Studente della relazione Esami di comparire come chiave primaria in una tupla della relazione Studenti

Integrità referenziale - 2

- I vincoli di integrità referenziale giocano un ruolo fondamentale nel concetto di modello basato su valori che non prevede altri meccanismi per correlare i dati
- In presenza di valori nulli i vincoli possono essere resi meno restrittivi
- E' possibile anche gestire automaticamente le situazioni in cui si ha una violazione di integrità referenziale in diversi modi:
  - Impedendo l'azione che genera la violazione: non posso cancellare uno studente che ha esami
  - Agendo su tutte le righe correlate (cascading): se cancello uno studente cancello tutti i suoi esami
  - Inserendo valori nulli: gli esami di uno studente cancellato avranno null nel campo studente

Conclusioni sull'integrità

- **Nell'esempio che abbiamo preso in considerazione all'inizio ogni violazione di integrità può essere evitata con uno degli strumenti descritti:**