INTRODUZIONE ALLE BASI DI DATI

Agenda

- Motivazioni
- Basi di Dati
- DB Management Systems
- Relational DBMS
- Structured Query Language
- Esempio

Motivazioni

- La gestione dei dati è un'attività cruciale per molti dei sistemi informatici commerciali e non
- Gran parte dei servizi commerciali su larga scala si basa sulla gestione dei dati
  - Servizi bancari, servizi informativi, facebook, ...
- E' necessario disporre di sistemi adeguati al trattamento di tali dati

Basi di Dati (DB)

- Collezioni di dati che devono essere memorizzati, acceduti e gestiti
- Generalmente di grandi dimensioni
  - es. il DB dell'anagrafe
- La loro gestione pone problemi di natura pratica
  - grande spazio di memoria => ricerche difficili
  - molti accessi simultanei => rallentamento
  - modifiche concorrenti => coerenza dei dati
  - ...

Basi di Dati (DB)

- In genere i dati da gestire non sono “grezzi”
  - “21”, “10 aprile 2005”, “rosso”, “3.14”, ...
- Spesso sono accompagnati da informazioni aggiuntive (meta-dati)
  - “Mario”, “Rossi”, “20/05/77”, “Sposato”, ...
- I meta-dati forniscono una struttura con cui i dati possono essere organizzati

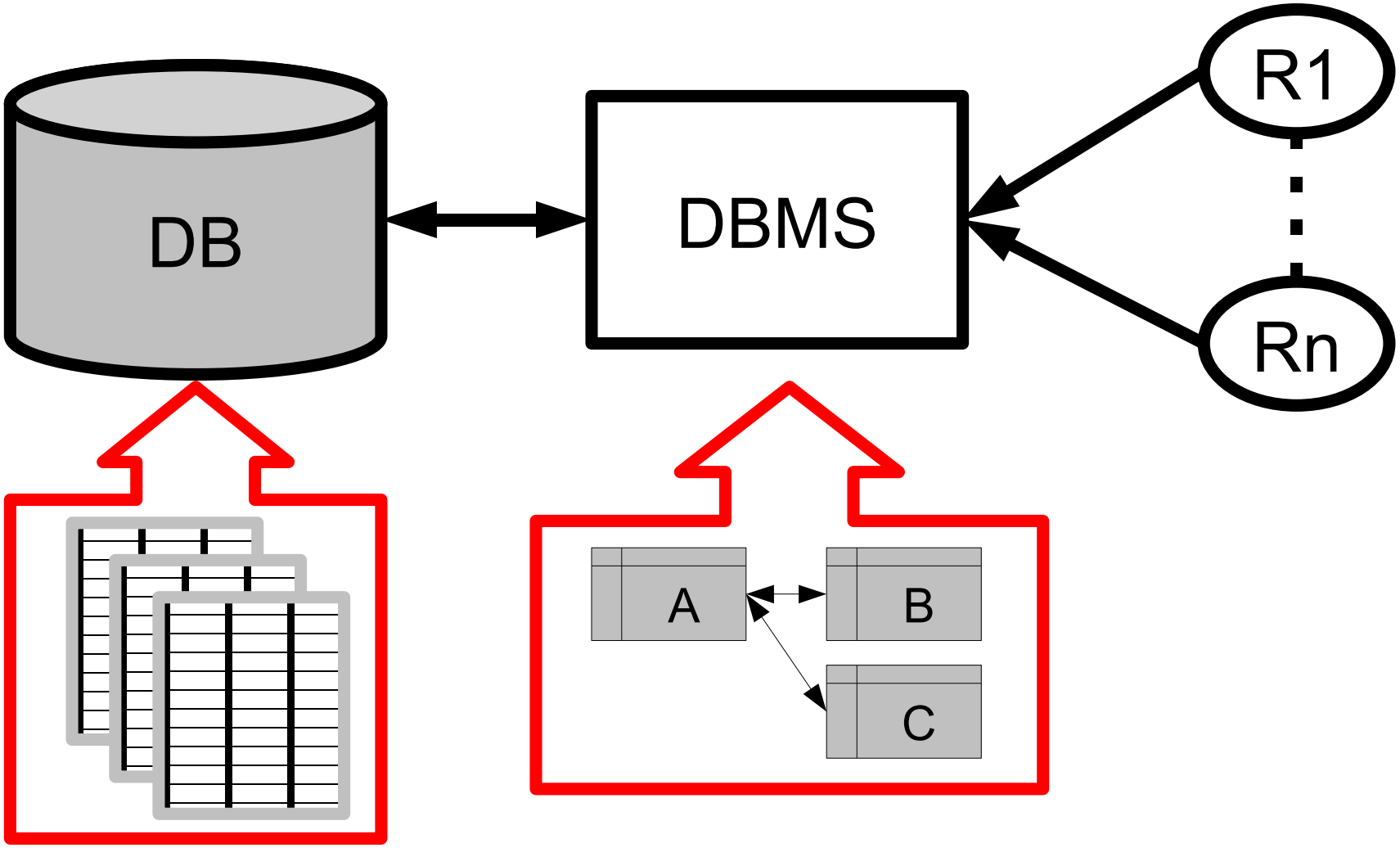
Basi di Dati (DB)

- Chiaramente la struttura rappresenta una parte dei dati da memorizzare
  - Più struttura
    - Migliore organizzazione ma DB più grande
  - Meno struttura
    - DB più snello ma difficile da gestire
- Es
  - Un libro senza indice
    - nessuna struttura => difficile da usare
  - Una voce di indice per ogni parola del libro
    - molta struttura => spazio sprecato

Basi di Dati (DB)

- Importante trovare un compromesso nella gestione dei dati
- La logica del DB viene divisa dalla gestione fisica dell'informazione
- Due componenti distinti
  - DB si occupa di gestire fisicamente i dati nella maniera più opportuna
  - DBMS comanda il DB decidendo quali dati e meta-dati devono essere memorizzati e come essi devono essere gestiti

Basi di Dati (DB)



DB Management System (DBMS)

- Agente SW che si occupa di gestire l'archiviazione e l'accesso ai dati del DB
- Principalmente deve offrire tre servizi
  1. Creare nuovi record
  2. Modificare la struttura del DB
  3. Consultare i dati
- In generale esistono più tecniche diverse per implementare queste funzionalità (diversi tipi di DBMS)

DB Management System (DBMS)

- DBMS gerarchico
  - Storicamente uno dei primi approcci alla gestione delle basi di dati
  - Usa una struttura gerarchica ad albero
    - Es: si immagini la struttura di un filesystem
  - Molto vicino alla reale gestione dei dati ma molto rigido
    - permette una sola classe di strutture

Relational DBMS (RDBMS)

- Approccio basato sul meccanismo delle relazioni
- Una relazione è un insieme di n-uple (o tuple) di elementi di determinati tipi (domini)
  - $R \subseteq D_1 \times D_2 \times ... \times D_n$
  - $<d_1 , d_2 , ... , d_n> \in R$
- Naturale rappresentazione delle relazioni tramite tabelle
  - Ogni riga della tabella/relazione corrisponde ad una n-upla

Relational DBMS (RDBMS)

- Esempio
  - Immaginiamo di voler memorizzare i dati relativi a degli studenti
    - Nome, Cognome, Matricola, Data di nascita
  - R contiene 4-uple
    - Es <“Mario”, “Rossi”, 111111, “1/1/1985”>
- Esistono diversi RDBMS
  - MySQL (open source)
  - SQLite (open source)
  - Oracle (proprietario)
  - ...

Structured Query Language (SQL)

- Linguaggio universale di interrogazione per i RDBMS
- Permette di estrarre informazioni dalle tabelle usando una sintassi semplice e intuitiva
- Nella maggior parte dei casi il passaggio dalla specifica dell'interrogazione alla query è immediato

Structured Query Language (SQL)

- Sintassi di base di una query
  - SELECT C_1 , ... , C_n
    FROM T_1 , ... , T_k
    WHERE Cond
  - Dove
    - C_1 , ... , C_n sono i campi desiderati (“Nome”, ...)
    - T_1 , ... , T_k sono le tabelle interessate (“Studenti”, ...)
    - Cond è una condizione che deve essere soddisfatta dai risultati dell'interrogazione (“Nome == Mario”, ...)

Structured Query Language (SQL)

- Selezionare i cognomi e i numeri di matricola di tutti gli studenti di nome “Mario”
- SELECT “Cognome”, “Matricola”
  FROM “Studenti”
  WHERE “Nome” = “Mario”

Nome	Cognome	Matricola	Nascita
Mario	Rossi	111111	1/1/84
Luigi	Bianchi	222222	2/2/85
Mario	Verdi	333333	3/3/86

Structured Query Language (SQL)

- Selezionare i cognomi e i numeri di matricola di tutti gli studenti di nome “Mario”
- SELECT “Cognome”, “Matricola”
  FROM “Studenti”
  WHERE “Nome” = “Mario”

Nome	Cognome	Matricola	Nascita
Mario	Rossi	111111	1/1/84
Luigi	Bianchi	222222	2/2/85
Mario	Verdi	333333	3/3/86

Structured Query Language (SQL)

- Selezionare i cognomi e i numeri di matricola di tutti gli studenti di nome “Mario”
- SELECT “Cognome”, “Matricola”
  FROM “Studenti”
  WHERE “Nome” = “Mario”

Nome	Cognome	Matricola	Nascita
Mario	Rossi	111111	1/1/84
Luigi	Bianchi	222222	2/2/85
Mario	Verdi	333333	3/3/86

Structured Query Language (SQL)

- Selezionare i cognomi e i numeri di matricola di tutti gli studenti di nome “Mario”
- SELECT “Cognome”, “Matricola”
  FROM “Studenti”
  WHERE “Nome” = “Mario”

Cognome	Matricola
Rossi	111111
Verdi	333333

Esempio