Corso di Laurea in
Ingegneria Informatica

Basi di dati
a.a. 2017-2018

Docente: Gigliola Vaglini
Docente laboratorio: Francesco Pistolesi

1

Principali Obiettivi del corso

- Imparare a portare a termine un buon progetto di base di dati, sia concettuale che logico.
- Imparare ad analizzare un progetto di base di dati, sia concettuale che logico, per verificarne la consistenza.
- Imparare ad impostare query per una base di dati relazionale, sia utilizzando un linguaggio di interrogazione reale che gli operatori utilizzati nell'esecuzione della query dal gestore della base di dati.

2

Bibliografia e strumenti

- Lucidi lezioni ed esercitazioni del modulo sul sito del corso
  - http://elearn.ing.unipi.it/course/view.php?id=1099
- Atzeni, Ceri, Fraternali, Paraboschi, Torlone

  Basi di Dati. Quinta Edizione.McGraw-Hill Italia, 2018
- Martorini, Vaglini

  Progettare una base di dati: dalla specifiche informali alle tabelle, Esculapio, 2011.

3

Comunicazione docente

- g.vaglini@iet.unipi.it
- Ricevimento lunedi mattina o su appuntamento

- f.pistolesi@iet.unipi.it
- Ricevimento su appuntamento http://www.iet.unipi.it/f.pistolesi

4

Esame

- Progetto concettuale e logico per un database relazionale
- Prova pratica di scrittura di query e stored programs in linguaggio MySQL su database già costituito
- Test scritto.

5

Laboratorio

- Prima lezione del laboratorio 8 marzo in F9
- Entro l' 8 dovreste iscrivervi sulla pagina web di Francesco Pistolesi

  http://www.iet.unipi.it/f.pistolesi

  Nel menu in alto a destra seguire "Teaching".

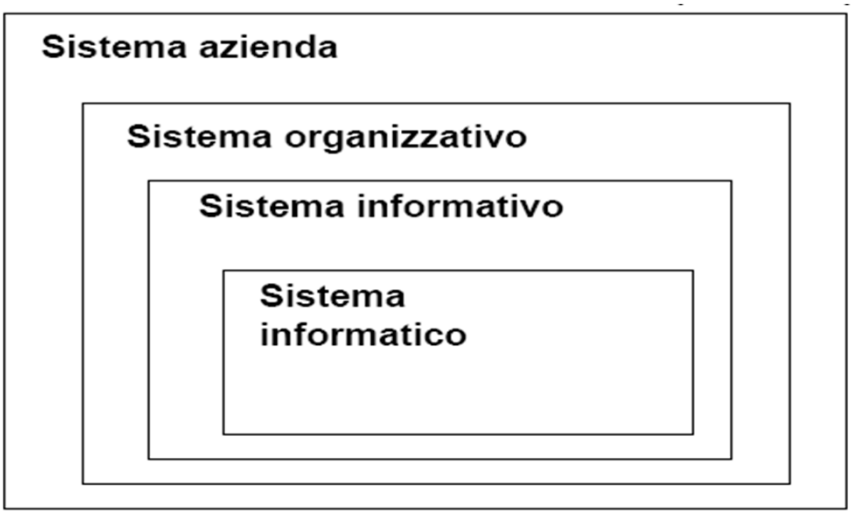
6

Lezione 1
Introduzione

Queste diapositive e le successive sono state rielaborate da G. Vaglini a partire dal materiale del sito http://www.ateneonline.it/atzeni/homeA.asp relativo al libro "Basi di dati"- Paolo Atzeni, et al.

7

Gerarchia di sistemi



8

Sistema informativo

- Il sistema informativo è la componente del sistema organizzativo che acquisisce, elabora, conserva, produce le informazioni di interesse (cioè utili al perseguimento degli scopi); inoltre esegue/gestisce i processi informativi (cioè i processi che coinvolgono informazioni)
- Il sistema organizzativo è costituito da risorse e regole per lo svolgimento coordinato di attività (processi) per perseguire gli scopi propri di un 'organizzazione (azienda o ente);
  - le risorse possono essere
    - persone, denaro, materiali, informazioni.

9

Sistemi informativi e automazione

- Il concetto di "sistema informativo" è indipendente da qualsiasi automatizzazione:
  - esistono organizzazioni la cui ragion d' essere è la gestione di informazioni (p. es. servizi anagrafici e banche) e che operano da secoli senza impiegare automatizzazioni.
- La parte del sistema informativo che gestisce informazioni con tecnologia informatica è il sistema informativo automatizzato (o sistema informatico)

10

Le Basi di Dati

- Il cuore di un sistema informativo automatizzato è la BD, cioè un insieme organizzato di dati utilizzati per rappresentare le informazioni di interesse
- Le Basi di Dati
  - hanno dimensioni (molto) maggiori della memoria centrale dei sistemi di calcolo utilizzati
  - hanno un tempo di vita indipendente dalle singole esecuzioni dei programmi che le utilizzano (persistenza dei dati)

11

Informazioni e dati

- Un' informazione è una notizia, dato o elemento che consente di avere conoscenza più o meno esatta di fatti o situazioni.
- Nelle attività standardizzate dei sistemi informativi complessi, sono state introdotte col tempo forme di organizzazione e *codifica* delle informazioni
- Nei sistemi informatici, le informazioni vengono rappresentate attraverso i dati: ovvero simboli che debbono essere elaborati
- Tali simboli hanno bisogno di un' interpretazione per rappresentare l' informazione originaria

12

Perché

- La rappresentazione precisa di forme di informazione più ricche è difficile.
- La rappresentazione tramite semplici simboli ha una struttura stabile nel tempo, in generale più delle procedure che vi operano; anzi nuove procedure di elaborazione ereditano i dati delle vecchie.

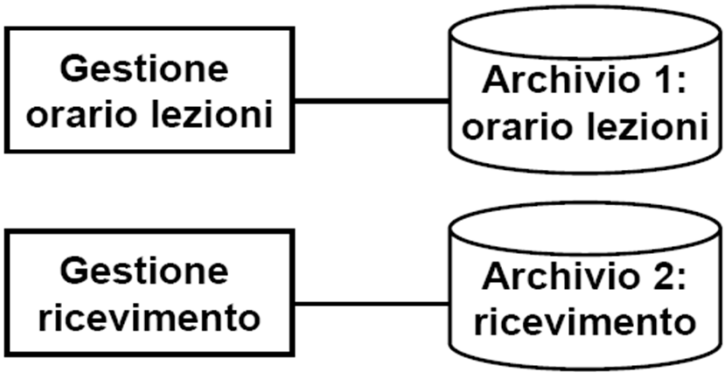
13

Le basi di dati sono „condivise“

- Ciascun settore/attività di un'organizzazione ha un (sotto)sistema informativo
- Una base di dati è una risorsa integrata e condivisa (i vari sottosistemi informativi non sono disgiunti) fra applicazioni (al contrario dei dati privati di singoli programmi)

14

Archivi separati



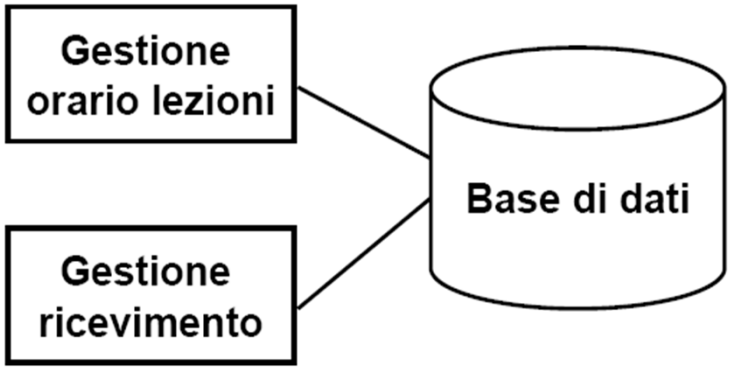
15

Problemi

- Informazioni ripetute
  - Rischio di incoerenza
    - Le versioni possono non coincidere (assegnazione docenti ai corsi)
  - Nessun controllo di consistenza possibile

16

Base di dati integrata e condivisa



17

Vantaggi

- l' integrazione e la condivisione permettono di evitare le inconsistenze
  - Potrebbe essere anche un vantaggio per l'occupazione di memoria
- Ma chi fa i controlli??

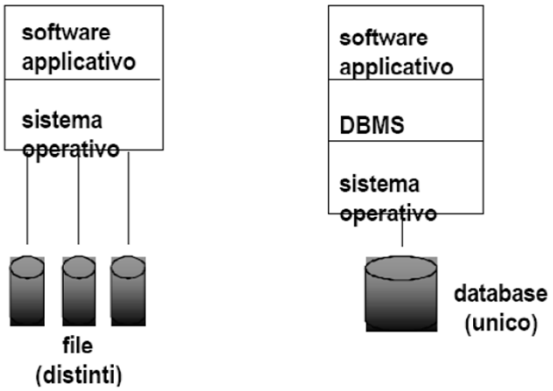
18

Base di dati (accezione tecnologica)

- *La base di dati è un insieme organizzato di dati grandi, persistenti e condivisi gestito da un DataBase Management System (DBMS)*
- *Cosa è e dove sta?*

19

File system e DBMS



20

File system e DBMS

- Nei DBMS, esiste una porzione della base di dati (il catalogo o dizionario) che contiene la descrizione centralizzata (unica) dei dati, e che può essere utilizzata dai vari programmi
- I DBMS usano comunque i file per la memorizzazione dei dati, ma estendono le funzionalità dei file system, fornendo più servizi ed in maniera integrata

21

Vantaggi e svantaggi dei DBMS

- Pro
  - gestione centralizzata con possibilità di "economia di scala"
  - disponibilità di servizi integrati
  - indipendenza dei dati (favorisce lo sviluppo e la manutenzione delle applicazioni)
- Contro
  - costo dei prodotti e della transizione verso di essi
  - non scorporabilità (spesso) delle funzionalità (con riduzione di efficienza)

22

DataBase Management System: quali servizi?

- Il DBMS gestisce insiemi di dati (grandi, persistenti, condivisi) garantendo
  - privatezza
  - efficienza
  - efficacia
  - affidabilità

23

I DBMS garantiscono la privatezza dei dati

- Esistono meccanismi di autorizzazione differenti per utenti differenti
  - A è autorizzato a leggere e a modificare tutti i dati
  - B è autorizzato solo a leggere il dato X mentre può leggere e/o modificare il dato Y

24

I DBMS cercano di essere efficienti

- Cercano di utilizzare al meglio le risorse di memoria (principale e secondaria) e di ridurre il tempo di esecuzione e di risposta
- I DBMS però forniscono tante funzioni e sono quindi sempre a rischio di diventare inefficienti e per questo ci sono grandi investimenti e competizione

25

I DBMS cercano di essere efficaci

- Forniscono agli utilizzatori funzionalità articolate, potenti e flessibili in modo da facilitare la loro produttività
- Il sistema informatico deve essere adeguatamente dimensionato e la base di dati ben progettata (e realizzata)

26

I DBMS garantiscono l'affidabilità dei dati

- Affidabilità significa resistenza a malfunzionamenti hardware e software
- Il contenuto di una base di dati deve essere conservato intatto a lungo termine o almeno deve essere ricostruibile:
  - funzioni di backup (salvataggio) e
  - recovery (recupero)

27

Come si garantisce l'affidabilità

- Fondamentale per garantire l' affidabilità è il concetto di transazione.
- La **transazione** è l'unità di lavoro elementare
- Tutti i DBMS (relazionali) sono **sistemi transazionali, cioè** mettono a disposizione un meccanismo per la definizione e l'esecuzione di transazioni

28

Transazione: definizione

- Insieme di operazioni elementari sulla base di dati da considerare indivisibile (atomicità), corretto anche in presenza di concorrenza (controllo della concorrenza) e con effetti definitivi (permanenza)

29

Atomicità

- La sequenza di operazioni sulla base di dati viene eseguita per intero o per niente:
  - Transazione bancaria
    - trasferimento di fondi da un conto A ad un conto B: o si fanno il prelevamento da A e il versamento su B o nessuno dei due

30

Serializzabilità

- L'effetto dell' esecuzione di transazioni concorrenti deve essere coerente (ad esempio, "equivalente" alla loro esecuzione in sequenza)

31

Permanenza (dei risultati)

- La conclusione positiva di una transazione corrisponde ad un impegno (in inglese commit) a mantenere traccia del suo risultato in modo definitivo, anche in presenza di guasti e di concorrenza

32

Prodotti commerciali

- Prodotti software (complessi) disponibili sul mercato sono:
  - Access
  - DB2
  - Oracle
  - Informix
  - Sybase
  - SQLServer

33

Descrizioni dei dati nei DBMS

- I programmi fanno riferimento ai dati, ma la loro struttura in memoria deve poter essere modificata senza dover modificare i programmi
- Viene introdotto il concetto di
  - modello dei dati : insieme di costrutti utilizzati per organizzare i dati di interesse e descriverne la dinamica
  - il modello dei dati fornisce ai programmi applicativi una vista astratta dei dati

34

Modelli logici

- Gerarchico e reticolare
  - utilizzano riferimenti espliciti (puntatori) fra record di un file per tenere conto della strutturazione dei dati ad albero o a grafo
- Relazionale
  - i riferimenti fra dati, anche in strutture (dette relazioni) diverse, sono ottenuti per mezzo dei valori stessi;
  - il costruttore di tipo in questo modello è la relazione, che permette di definire insiemi di record omogenei a struttura fissa, tale struttura è equivalente ad una tabella

35

Schema e istanza

- Ogni tipo di dato ha
  - Uno schema, sostanzialmente invariante nel tempo, che ne descrive la struttura (aspetto intensionale)
    - Nel modello relazionale le intestazioni delle tabelle
  - un' istanza, i valori attuali, che possono cambiare anche molto rapidamente (aspetto estensionale)
    - Nel modello relazionale il "corpo" di ciascuna tabella

36

Linguaggi per basi di dati

- La disponibilità di vari linguaggi e interfacce per la definizione di schemi e per la lettura/modifica di istanze contribuisce all'efficacia del DBMS
- Una distinzione terminologica
  - data definition language (DDL) per la definizione di schemi (logici, esterni, fisici)
  - data manipulation language (DML) per l'interrogazione e l'aggiornamento di (istanze di) basi di dati

37

Tipi di Linguaggi per basi di dati

- linguaggi testuali interattivi (**SQL**)
- comandi (SQL) immersi in un linguaggio ospite (Pascal, Java, C ...)
- con interfacce amichevoli (senza linguaggio testuale come Access)

38

Architettura del DBMS

- Abbiamo parlato del modello dei dati usato dai DBMS
- Parliamo adesso del modello di esecuzione, usato da un sistema in cui esiste un DBMS.

39

Architettura e modello di esecuzione

single tier

Mainframe + terminali intelligenti

- *Pro:* Facilmente gestibile da un amministratore centrale
- *Contro:* Interfaccia grafica richiede potenza di calcolo, sottratta alla gestione del BD

40

Architetture distribuite

- Quasi tutte le realizzazioni di DBMS considerano un'architettura distribuita, caratterizzata dalla presenza di una rete che collega almeno due macchine che lavorano autonomamente, ma sono anche in grado di interagire
- Architettura client-server: è la più semplice e diffusa

41

Architettura client-server

- Modello di interazione in cui i processi software si dividono in Client e Server.
  - Client
    - Richiedono i servizi
    - Dedicati a interagire con l' utente finale
    - Ruolo attivo: genera richieste
  - Server
    - Offrono i servizi
    - Ruolo reattivo: si limita a rispondere alle richieste dei diversi client
  - L' interazione fra client e server richiede una interfaccia che è l' elenco dei servizi messi a disposizione dal server

42

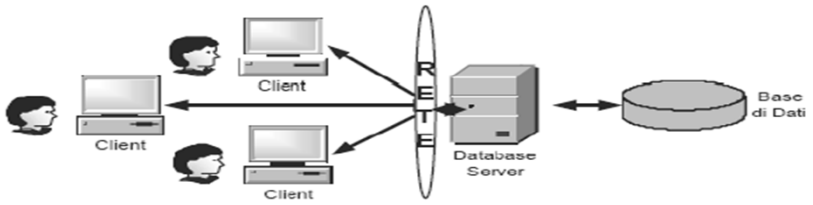
Architettura client-server

- Processi client e server potrebbero girare sulla stessa macchina, ma
- in genere ogni processo risiede su una macchina diversa, collegata alle altre via rete
- Il server può essere unico, ma se ne esiste più di uno per realizzare una funzionalità si ha una architettura completamente distribuita

43

Architettura client-server: esigenze HW diverse

- Un processo client può richiedere servizi a vari processi server. Il client è un elaboratore adatto alla interazione con l' utente
  - Strumenti di produttività
  - Applicazioni "amichevoli" che accedono alla base dati
- Ogni processo server risponde a (molte) richieste da parte di molti processi client gestendo in modo opportuno le relative transazioni. Il server è dimensionato in base ai servizi che deve offrire e al carico transazionale (grande memoria centrale, grande memoria di massa, ...)



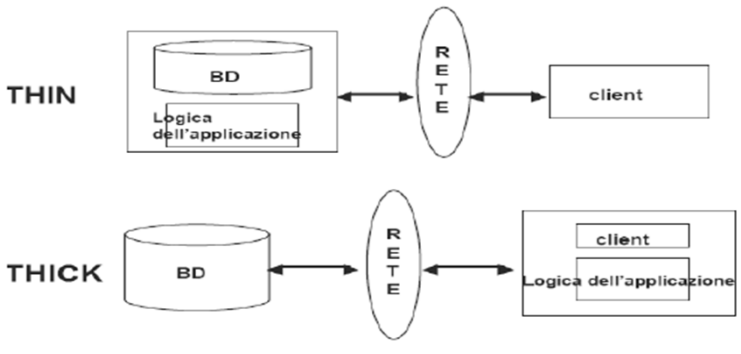
44

Architettura two tier

- L'architettura client-server è detta anche a due livelli (*two tier*)
- Le macchine del livello Client non sono necessariamente poco complesse

45

Struttura del client



Più diffusa l'architettura thin client

46

Svantaggi thick client

- È necessaria fiducia tra il server e i client (i dati vengono trasmessi al client)
- Non scalabile (non più di poche centinaia di client):
  - richiede anche una altissima capacità di elaborazione da parte del server per trasmettere i dati ai client

47

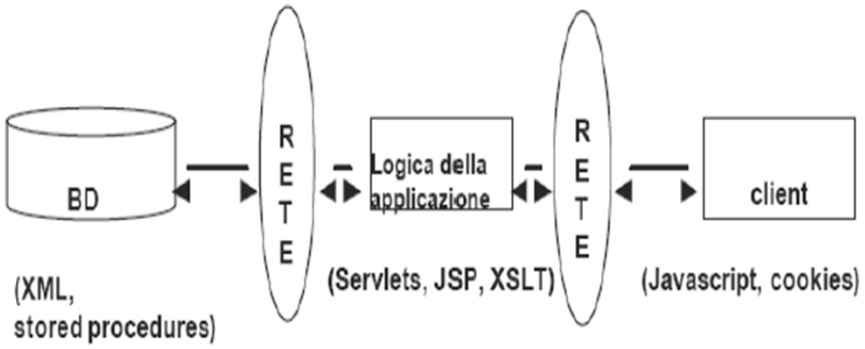
Architettura a 3 livelli

- Nell'architettura three-tier è presente un secondo server, il server applicativo, responsabile di gestire la logica applicativa comune a più client
- Il client è più semplice e si occupa solo dell'interfacciamento con l'utente finale (thin client)
  - Il client invia le richieste al server applicativo
  - Il server applicativo dialoga con il server per la gestione dei dati
- Questa architettura è predisposta per Internet

CLIENT = BROWSER WEB

48

Architetture three tier



49

Vantaggi three tier

- Si possono interconnettere sistemi eterogenei
- I client sono thin (browser web)
- Scalabilità sul numero di client: si aggiunge un server (si possono appropriatamente moltiplicare anche le macchine a livello intermedio)

50

Big Data

- Già nel 1975 si cominciò a parlare di *Very Large Data Bases,* recentemente è stato introdotto nell'uso comune il termine *big data.*
- Da un lato la produzione di dati ha raggiunto livelli straordinari (grazie alle più svariate fonti, ad esempio: la telefonia, i sensori, le reti sociali, i dati sperimentali della fisica e della medicina), dall'altro sono stati introdotti strumenti informatici sempre più evoluti in grado di memorizzare, interrogare ed analizzare questi dati.

51

I Big Data e le quattro V

- **Volume.** È la principale caratteristica dei *big data*. Lo stesso quantitativo di dati prodotto dall'inizio dell'umanità ad oggi sarà a breve generato in un minuto. In un minuto, ad esempio, si generano 2,3 milioni di interrogazioni a Google, 3 milioni di like e 3 milioni di share su Facebook; 2,7 milioni di video vengo scaricati da Youtube (e 139 mila nuovi video vengono caricati); 44 milioni di messaggi vengono processati, di cui 486 mila contenenti foto e 70 mila contenenti video (dati al 25/7/2017).

52

Le quattro V (2)

- **Velocità.** Ossia la velocità con cui i dati vengono generati e scambiati.
- Tecnologie specifiche di analisi di dati in memoria e tecniche di *data streaming* consentono di analizzare i dati mentre fluiscono verso il sistema di gestione, spesso evitando di memorizzare i dati nel database dove vengono memorizzati solo i risultati dell'analisi.

53

Le quattro V (3)

- **Varietà.** In passato l'attenzione principale era rivolta ai dati strutturati, tipicamente memorizzati in tabelle, oggi la maggior parte dei dati sono non strutturati (testi, immagini, voci, video).
- Tecnica fondamentale per gestire al meglio la varietà dei dati è la *data integration* .

54

Le quattro V (4)

- **Veridicità.** E' possibile estrarre informazioni vere dai dati, nonostante essi contengano gravi errori, imprecisioni e incompletezze.
- Si parla di *data quality* come tecnica per gestire la (mancanza di) veridicità che caratterizza molte raccolte di dati.

55

La scienza dei dati

- La *Data Science* è tipicamente associata ai *Big Data* ed è una materia interdisciplinare, tra la statistica e l'informatica,
- Una definizione abbastanza completa di data science include almeno i seguenti aspetti.

56

Data science (1)

- *data cleaning*: ha per obiettivo la costruzione di una raccolta dati che abbia un sufficiente livello di qualità. Numerose tecniche possono esser usate per esempio metodi che utilizzano regole di trasformazione.
- *data integration*: ha per obiettivo la costruzione di una raccolta dati integrata a partire da differenti sorgenti dati. Questo è un problema che spesso richiede soluzioni ad hoc.

57

Data science (2)

- *data mining*, ovvero l'estrazione di informazione utile dai dati. Nel data mining sono molto utilizzate le regole di associazione, che consentono di dire quante volte, nel contesto di una specifica operazione (ad esempio, una transazione di acquisto) sono coinvolte le stesse istanze di dati (ad esempio, gli stessi oggetti).

58

Data Science (3)

- *Metodi predittivi*. Ovvero metodi che consentono di prevedere, a partire da osservazioni nel passato, i dati caratterizzanti uno scenario futuro oppure alternativo. Tecniche statistiche consentono di selezionare i dati più utili (*feature selection*). Le predizioni vengono associate ad una probabilità.

59

Data science (4)

- Metodi di apprendimento automatico o *machine learning*. Sono particolari metodi predittivi per classificare i dati a partire da esempi di classificazione. Alcuni dati (*training set*) sono classificati a priori (ad esempio, da un esperto) e associati ad una etichetta (*label*) che ne indica la classe.
- Il sistema di machine learning apprende come classificare gli altri dati a partire dal training set.

60

Data science (4) cont.

- Tra i metodi di machine learning assume sempre maggior rilevanza il cosiddetto *deep learning*, che consiste in un'imitazione del comportamento del cervello umano, in quanto dotato di moduli software che simulano i neuroni e le connessioni fra di essi.

61

Tecnologie per i Big data

- Alcune tecnologie associate ai big data sono state introdotte ad hoc, ad esempio i sistemi di *cloud computing* sono pensati soprattutto per l'elaborazione batch di enormi moli di dati in parallelo, e sono scarsamente utilizzabili per rispondere ad interrogazioni online.

62

Tecnologie per i Big data (2)

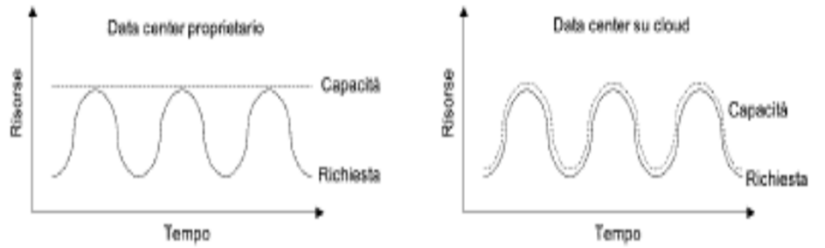
- I cosiddetti sistemi *NoSQL* rinunciano a gestire l'intera complessità del linguaggio SQL o le caratteristiche di piena transazionalità pur di gestire efficientemente alcune tipologie di dati con una semplice struttura.

63

Cloud Computing

- La maggior parte delle nuove applicazioni usa l'architettura three-tier di Internet, e le applicazioni sono eseguite sfruttando un cloud (privato o pubblico) e servono un grande numero di utenti.
- Infatti c'è grande necessità di spazio e potenza di calcolo per l'analisi dei dati ma non continuativa

64



65

E i database?

- Poichè aumentano sia i Big Users che i Big Data, le applicazioni e anche i sottostanti database devono scalare; le scelte possibili sono due: scalare verticalmente (up) o orizzontalmente (out).
  - Scalare verticalmente implica un approccio centralizzato che richiede server sempre più potenti.
  - Scalare orizzontalmente implica invece un approccio distribuito che sfrutta standard server.

66

Scalare i Database

- I database relazionali hanno un approccio centralizzato (dicevamo che il catalogo è unico) e quindi possono scalare verticalmente ma non orizzontalmente.
  - Significa che, se aumentano utenti o dati, necessitano server più potenti: più CPU, più memoria, più memoria su disco. I server molto potenti sono molto complessi e costano molto, inoltre vanno comprati così.

67

Google, Facebook ….

- I giganti del web, che si trovano a dover gestire database di dimensioni veramente imponenti, hanno sviluppato (o contribuito allo sviluppo di) vari **NRDBMS** (database non relazionali) con approci leggermente diversi, ma tutti con gli stessi principi di base, si raggruppano infatti sotto la dicitura di **movimento NoSQL (Not Only SQL)**.

68

L' Approccio

- I database NoSQL sono stati sviluppati dall'inizio per essere distribuiti e scalare orizzontalmente. Usano insiemi di server standard sia per i dati che le applicazioni.
- Il sistema scala facilmente aggiungendo nuovi server all'insieme precedente, i dati e le operazioni sono distribuiti sul nuovo cluster, infatti le applicazioni vedono un solo database distribuito.
- Naturalmente ci si aspetta che i server vadano in crash di tanto in tanto, quindi il database è costruito per tollerare i fallimenti ed essere in grado di recuperare i dati in modo affidabile.

69

Caratteristiche

- I database NoSQL presentano alcune caratteristiche comuni per quanto riguarda la scalabità e la performance.

70

- Auto-sharding – Il database NoSQL distribuisce automaticamente i dati tra i server, senza richiedere la partecipazione delle applicazioni. Infatti i database server possono essere aggiunti o rimossi senza che le applicazioni vengano interrotte.
- Replicazione – La maggior parte dei database NoSQL effettua replicazione dei dati, memorizzandone molte copie sui vari server dello stesso data center o su diversi data center. In questo modo viene garantita un'alta affidabilità e la possibilità di sopravvivere ai "disastri". Un database NoSQL opportunamente gestito non dovrebbe mai dovere essere messo fuori linea per recuperare da fallimenti.

71

- Query distribuite efficienti - Anche nei database relazionali è possibile effettuare sharding, ma in questo caso la possibilità di effettuare query complesse potrebbe essere ridotta. Nel caso di database NoSQL invece si mantiene la completa potenza espressiva delle query anche se il database è distribuito tra centinaia di server
- Caching trasparente – I sistemi NoSQL avanzati effettuano la memorizzazione dei dati necessari per le operazioni dalla memoria su disco alla memoria centrale in modo trasparente alle applicazioni e alla realizzazione delle operazioni. Invece nei database relazionali esiste un'infrastruttura per il caching separata su un server separato che quindi deve essere visto nella realizzazione delle operazioni.

72

Vantaggi dei sistemi NoSQL

- La **semplicità** di questi database è uno degli elementi fondamentali, è proprio questo che permette di scalare in orizzontale in maniera così efficiente, molti DBMS non relazionali, infatti, permettono di **aggiungere nodi a caldo in maniera impercettibile dall'utente finale**.
- E' possibile scegliere un database adatto alla mappatura più diretta alle object classes del proprio applicativo (Database ad oggetti). In questo modo si possono ridurre di molto i tempi dedicati allo sviluppo del metodo di scambio dati tra il database e l'applicativo stesso

73

Svantaggi dei sistemi NoSQL

- La semplicità di questi database, però, è dovuta anche alla **mancanza dei controlli fondamentali sull'integrità dei dati**: il compito ricade totalmente sull'applicativo che dialoga col database e che, ovviamente, dovrebbe essere testato in modo molto approfondito prima di essere messo in produzione.

74

Cont.

- La mancanza di uno standard universale (come può essere l' SQL) è un' altra delle pecche di questi database non relazionali, ogni database ha infatti le proprie API (Application programming Interface) e il suo metodo di storing e di accesso ai dati. Se lo sviluppo del database sul quale abbiamo basato il nostro applicativo venisse interrotto, il passaggio ad un altro database non sarebbe una cosa immediata, ma richiederebbe cambi più o meno profondi da apportare all' applicativo

75

Esempi

- **MongoDB** è un sistema orientato ai documenti ed è il più diffuso tra i DBMS non relazionali; è usato ad esempio da eBay
- **Cassandra**: è forse il più famoso tra i database non relazionali di ultima generazione, è adottato per gestire gigantesche quantità di dati da compagnie come **Facebook, Digg, Twitter, Rackspace** e molti altri.
  - Tra le sue caratteristiche principali ci sono lo **scaling orizzontale (out) e un'altissima ridondanza** (potrebbe anche cadere un intero data center, con altri nodi sparsi per il globo continuerebbe a funzionare tutto).

76

Conclusioni

- I database NoSQL non saranno i nuovi dominatori del mercato. I database relazionali saranno ancora usati in una grande varietà di applicazioni. Semplicemente non saranno più la scelta automatica.

77