

# Gestione dei Big Data

Matteo Aprile

Professore: Marco Zappatore, Antonella Longo

## INDICE

<b>I</b>	<b>Libri di testo consigliati</b>	<b>1</b>
<b>II</b>	<b>Databases - 28.09.22</b>	<b>1</b>
<b>III</b>	<b>Database System Concepts and Architecture - 28.09.22</b>	<b>1</b>
<b>IV</b>	<b>Data Modeling Using the Entity-Relationship (ER) Model - 28.09.22</b>	<b>2</b>
	<b>Riferimenti bibliografici</b>	<b>2</b>

### I. LIBRI DI TESTO CONSIGLIATI

- Fundamental of Database Systems, 7th ed, Elmasri, Navathe
- Data Warehouse Design, Rizzi, Golfarelli big dataL concepts technology and architecture 1st ed balusamy abirami gadomi

### II. DATABASES - 28.09.22

1. definizioni di base: dato: insieme di fatti conosciuti registrati con un significato. Sono detti dati grezzo visto che si suppone che andrò a elaborarlo, questo dato sarà poi archiviato, sarà un fatto conosciuto cioè degli eventi con un significato per un dato tipologia di utenti ed el conosciuto dato che c'è una sorgente che produce i dati con una certa velocità.

database: raccolta di dati altamente organizzati intercorrelati e strutturati. el una struttura con dei collegamenti strutturati tra i dati

dbms: data base management system: insieme di programmi per accedere ai dati e farci delle operazioni di 4 tipi: creazione, recupero, aggiornamento e cancellazione, ciclo CRUD.

esistono molti tipi di db i primi erano solo numerici o testuali, ora ci sono quelli multimediali, GIS Geographic Information Systems, data warehouses

opportuno vedere dei concetti di base. i dati hanno un ciclo di vita, il più semplice è: acquisizione (scattered data) -> aggregazione (integrated data) -> analisi (knowledge) -> finisce in un applicazione che genera dei log data che saranno poi acquisiti come scattered data

da un punto di vista computazionale queste fasi si devono prendere in un altro modo: 1. storage dei dati 2. formattazione e pulizia 3. capire cosa i dati ci dicono 3. se non mi bastano i dati che ho posso integrare dei dati

definizioni: database: collezione di dati collegati tra loro mini-world: parte del mondo real ai quali si riferiscono i dati presi. per creare un database vado a limitare la modellazione in un numero n di concetti

dbms: sistema che facilita mantenimento e gestione del database system: insieme di dbms con i dati

quando si ha un db abbiamo 3 livelli da considerare fisico (dove sono salvati logico: come sono collegati tra loro vista: rappresentazione che sarà diversa per ogni tipo di utente

da un punto di vista architetturale da un altro ho gli utenti e dall'altra i dati e i metadati (che descrivono i dati). avrò bisogno di un software per eseguire le query un dbms offre: di db mi consente di fare più del semplice salvataggio: definire modelli dati manipolarli processarli e condividerli

per interagire coi db possono essere tramite: query: accede a parti differenti di dati e formula una richiesta transazionale: legge dei dati ed aggiorna alcuni valori e li salva nel db

nei nostri mini-world avremo bisogno di identificare delle entità cioè i concetti di base che rappresentano una parte delle cose che inseriremo nel db relazione il resto sta nel connettere tra loro le entità, dette relazioni (relationships) (ER) le tabelle che ne derivano sono dette relation. tutto ciò deve derivare dai requisiti e non dall'esperienza personale.

le entità diventano delle tabelle dove inserisco i dati che ho a disposizione che saranno divise per righe (record) e colonne (attributi) i singoli elementi sono i dati grezzi catalogo dei dbms ci sono i vincoli creati da un software specifico, i tipi dei dati e la relazione di appartenenza degli attributi

concetto fondamentale: astrazione: ho dei dati e collegamenti tra varie entità e dei modi per salvarli. dobbiamo separare le cose per disporre di un modello senza che esso si occupi di come salvare i dati

modello concettuale:

modello fisico: definizione dei tipi di dati e dove sono conservati

controllo della concorrenza: garantire che tutte le transazioni sono correttamente eseguite recovery: se la transazione è stata eseguita e stata recorded nel database

### III. DATABASE SYSTEM CONCEPTS AND ARCHITECTURE - 28.09.22

data model: insieme di concetti che descrivono la struttura di un db, le operazioni e i vincoli applicati al db

data model structure e constraints: abbiamo dei costrutti che definiscono come collegare gli elementi definiti da: entità, record e tabella i vincoli devono essere sempre rispettati e bloccanti

data model operation: di base (CRUD) o definite dall'utente

modello dato di tipo concettuale: di alto livello e semantico

modello fisico: definisce come i dati sono salvati, basso livello

modello tipo implementativo: usati nel dbms

modello implementation: provide concepts that fall between the above two

modello autodescrivente: basati su XML

3 definizioni fondamentali: database schema: descrizione del database in termini di struttura tipo dati e vincoli

schema diagram: visione rappresentativa del db schema

schema construct: insieme tra schema ...

figura 2.1 schema diagram

database state: snapshot in istante t del db, si definisce quindi ai suoi contenuti

valid state: si definisce funzionante se il suo contenuto soddisfa i vincoli per quello schema

nota che: schema può essere detto intensional state può essere detto extensional

abbiamo 3 livelli di schema: interno (fisico): come i dati devono essere salvati e come posso accedere concettuale esterno: per descrivere le view dell'utente

per passare da uno schema ad un altro ho bisogno di un mapping per capire a cosa corrisponde un elemento, avremo che:

logical data independence: se voglio cambiare lo schema concettuale senza cambiar quello fisico

physical; devo cambiare lo schema fisico senza cambiare quello concettuale

quando ho un data manipulation language dml posso usarlo in autonomia o inserirlo in un programma facendolo tramite: embedded approach (tramite java, c ...) procedure call approach db programming language approach scripting languages

definizione data dictionary: insieme dove salvo sia lo schema che altre info

abbiamo più tipologie e di dbms: centralized: dove abbiamo tutta l'elaborazione su un unico nodo

2-tier: specializza in termini di server per ogni blocco di funzionalità che devo offrire

clients: per far accedere gli utenti

dbms server: per eseguire query e transazioni tramite API

three-tier: abbiamo 3 livelli: ...

(requisiti relazionali). dai data requirements ho, tramite il modellista, la conversione in un modello concettuale, poi abbiamo l'applicazione dell'algoritmo di mapping da parte del modellista il dbms si occuperà del physical design e il successivo internal schema. per definire le transazioni dal miniworld dobbiamo estrarre i functional requirements per effettuare una functional analysis che genera delle transazioni ad alto livello.

notiamo che le entità si scelgono in base ai sostantivi e le relazioni in base ai verbi

entità (sostantivi): oggetti o cose specifiche presenti nel minimondo che bisogna rappresentare

attributi (proprietà): descrittivi per ogni entità

record: insieme degli attributi che si danno ad un'entità

tipo dato: tipo per ogni attributo dato singolo: ha un unico valore

dato composto: dati da un insieme di più descrittori, possono formare una gerarchia, notazione ...(... , ...)

dato multivalore: attributi che hanno n-uple di valori, notazione ...

presa la mia entità possiamo identificare univocamente tutti i record tramite gli attributi chiave oppure un'unione tra l'attributo chiave ed un altro attributo.

le relazioni relationship sono quelle che collegano le entità. il grado di tipo della relazione e il numero di partecipanti a quella relazione. bisogna quindi identificare quante volte la relazione viene persorsa. ER lo scriviamo da sx a dx e da su a giù. i nomi delle entità vanno al singolare e verbi alla terza persona e attivi o passivi per capire da che parte si deve leggere la relazione. per la cardinalità posso dirla leggendo la relazione e chiedendomi quante volte è valida per l'entità opposta, cercando di capire un solo elemento quante entità può avere dell'altro

entità debole: entità che da sola non può esistere, quindi dipende da un'entità più forte

le cardinalità possono essere:

#### RIFERIMENTI BIBLIOGRAFICI

- [1] <https://en.wikibooks.org/wiki/LaTeX/Hyperlinks>

#### IV. DATA MODELING USING THE ENTITY-RELATIONSHIP (ER) MODEL - 28.09.22

parto dal minimondo che posso ricevere come descrizione scritta. nella gestione del minimondo serve capire cosa serve e cosa no questi sono i requisiti dati. quindi la mia applicazione deve gestire alcuni dati e poi parli vedere