Corso di Informatica, Università degli studi di Udine

Laboratorio di Basi di Dati (a.a. 2024-2025) – Lezione 1 –

Luca Geatti
luca.geatti@uniud.it

21 Ottobre 2024

Informazioni sul docente

- **Nome:** Luca Geatti
  - RTD-a @ Departimento di Scienze Matematiche, Informatiche e Fisiche
- **E-Mail:** luca.geatti@uniud.it
- **Website:** `https://users.dimi.uniud.it/~luca.geatti/`
- **Ufficio:** . . . mandatemi una mail che ve lo spiego . . .
- **Ricevimento:**
  - il Giovedì 14:00-16:00
  - meglio su appuntamento
- **Annunci:** su Teams
- **Materiale del corso**: sulla mia pagina web e su Teams

Modalità di Lezione

E' un laboratorio: (36 ore)

- **interattività**
- poche lezioni frontali
- alcune lezioni dimostrative con l'uso del laptop (per il linguaggio R)
- **modalità:**
  - durante le ore di laboratorio (a parte le lezioni frontali e dimostrative) lavorate al **progetto**
  - il docente è qui a disposizione per chiarimenti
- **scenario ideale:** il progetto viene svolto interamente o per maggior parte durante le 36 ore di laboratorio
- suggerimento: sfruttate le ore di laboratorio

Lezioni

Quando?

- ogni Lunedì 15:30 - 18:30
- possibili cambiamenti per conciliare gli impegni accademici e istituzionali del prof. Montanari
  - controllate Teams

Dove?

- Aula C3

Registrazioni:

- Cercherò di caricare puntualmente le registrazioni di ogni lezione

Materiale del corso

Libro:

- Basi di dati Atzeni-Ceri-Fraternali-Paraboschi-Torlone Sesta edizione McGraw-Hill
- anche altre edizioni vanno bene; es.: Basi di dati: Modelli e linguaggi di interrogazione Atzeni-Ceri-Paraboschi-Torlone Terza edizione
- (attenti alla differente numerazione dei capitoli)

Altro materiale (slides, altri libri, software, ...) sarà disponibile su Teams o sulla mia pagina web.

Esame – Progetto

Progettazione e implementazione di una basi di dati

Gruppi di 3 o 4 persone

- pochissime eccezioni

Votazione:

- voto unico
  - esame di teoria (Prof. Montanari): 75%
  - progetto di laboratorio (Dr. Geatti): 25%
  - incremento di 1 punto per i gruppi che consegnano in tempo (entro i primi 2 appelli)
  - è necessario raggiungere almeno la sufficienza (18) sia nella teoria che nel laboratorio

Consegna progetto

La consegna del progetto si svolgerà nelle prossime ore di laboratorio:

- dopo alcune ore di lezione frontale (progettazione concettuale), i gruppi verranno qui per il ritiro del testo del progetto.
- mi appunterò
  - nome
  - cognome
  - numero di matricola
  - corso di studi

  di tutti i membri del gruppo

NO email per ritirare il testo del progetto.

Consegna progetto – 2

Assegnazione dei progetti ai gruppi casuale, basata su politica First-In-First-Served

Novità di quest'anno: potete proporre progetti *"alternativi"*:

- e.g., progetti su domini a scelta
- vanno prima di tutto discussi con me *durante le ore di laboratorio*.

Consegna progetto – 3

Tempistiche:

- **non** c'è nessuna scadenza
- l'esame di teoria viene conservato in attesa che il progetto venga ultimato e viceversa.
- **Suggerimento:** svolgere il progetto di pari passo con l'andamento del corso
  - statisticamente, i gruppi che consegnano entro i primi 2 appelli prendono un voto più alto di quelli che consegnano dopo.

Consegna progetto – 4

Materiale da inviare:

- Relazione (da consegnare a me via email in formato pdf) e codice (in formato di script eseguibili)
- Il focus deve cadere sulle fasi di *progettazione* (analisi dei requisiti, progettazione concettuale, logica e fisica)
- Tutto il codice deve essere *riproducibile* e *facilmente eseguibile*.
- Lunghezza: non c'è una regola ma tipicamente tra le 15 e le 30 pagine
  - scrivere tanto non significa aver fatto un buon progetto
  - scrivete poco ma bene

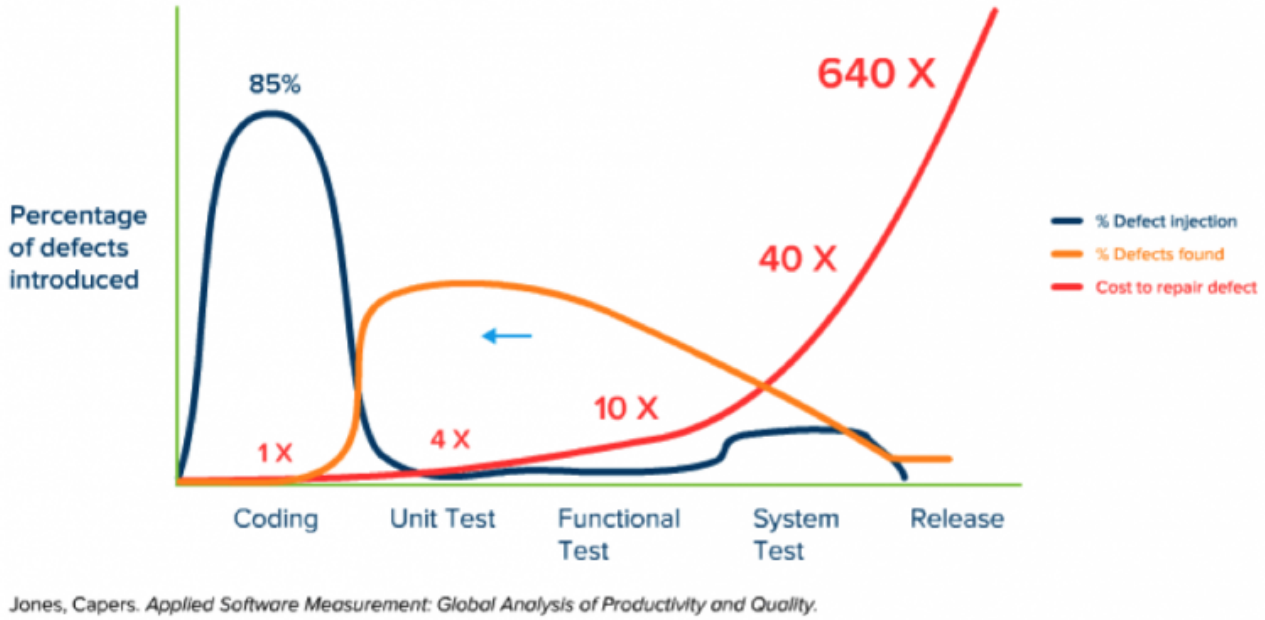
Consegna progetto – 5

Valutazione

- volta anche a valutare contributi individuali
- interazioni dirette durante le lezioni di laboratorio
- dopo la consegna riceverete una valutazione:
  - possono passare varie settimane dalla consegna alla valutazione
  - se non avete ancora svolto la prova scritta, il voto sarà conservato

Obiettivi del progetto

Importanza della corretta interazione tra le varie fasi della progettazione di una base di dati



Jones, Capers. *Applied Software Measurement: Global Analysis of Productivity and Quality.*

Un errore nelle fasi iniziali del ciclo di produzione del software ha conseguenze molto più gravi (difficoltà nella sua identificazione, risorse spese per correggerlo, propagazione alle fasi successive)

Obiettivi del progetto

Nel vostro progetto, dovrà essere chiaro che:

- Ogni fase produce uno o più documenti che rappresentano l'input per la fase successiva:
  1. analisi dei requisiti
  2. progettazione concettuale
  3. progettazione logica
  4. progettazione fisica
  5. implementazione
  6. analisi dei dati
- Importante: ogni fase rappresenta lo stesso insieme di requisiti! L'unica cosa che cambia è come sono rappresentati.
- Coerenza e uniformità tra documenti prodotti nelle diverse fasi

Obiettivi del progetto - 2

- *Prodotti della progettazione*: non solo SQL ma anche e SOPRATTUTTO gli schemi prodotti dalle varie fasi (es. schema concettuale, logico, ecc.). Importanti perché:
  - rappresentano l'unica documentazione su cui basare la fase successiva
  - sono fondamentali (a volte l'unico supporto) all'implementazione (es., per l'implementazione di query)
- Necessità di cicli tra fasi di progetto per raffinare/modificare prodotti di una fase precedente mentre si è in una fase successiva
- Dimostrare di avere ben chiare le difficoltà e gli aspetti concettualmente critici ed interessanti della progettazione (ridondanze, generalizzazioni, vincoli aziendali, ecc.)

Fasi del progetto & Struttura della relazione

1. Analisi dei requisiti
   - nel vostro caso, i requisiti saranno il testo del progetto
   - risolve eventuali ambiguità nei requisiti
   - produce un glossario
2. Progettazione concettuale
   - il modello concettuale dei dati (nel nostro caso, il modello Entità-Relazioni) tiene conto dei concetti principali da rappresentare (Entità) e delle loro relazioni
   - *non* si focalizza su *che tipo* di dati verranno rappresentati nè sul *come*
   - produce uno schema concettuale (schema E/R)

3. Progettazione logica
   - tiene conto del modello logico dei dati, più concreto perché legato alla famiglia di DBMS (nel nostro caso, DBMS relazionali), astrae da modello fisico (cosa/come)
   - produce uno schema logico (nel nostro caso, uno *schema relazionale*)
   - due sottofasi:
     3.1 ristrutturazione del modello E/R (analisi delle ridondanze basate su tabelle dei volumi e frequenza delle operazioni, eliminazione di tutti i costrutti dell'E/R non esprimibili nel relazionale, *e.g.*, le generalizzazioni)
     3.2 traduzione nello schema relazionale

4. Progettazione fisica
   - tiene conto di come vengono memorizzati fisicamente i dati
   - produce il *codice per la definizione delle tabelle* e per la definizione degli *indici*:
     - analisi e scelta di opportuni indici
     - definizione di relazioni e indici in SQL

Fasi del progetto & Struttura della relazione

5. Implementazione
   - creazione del DB
   - esecuzione del codice per la creazione delle tabelle
   - popolamento della base di dati (a mano o automatizzato, *e.g.*, usando il linguaggio R)
   - definizione di pochi (2 o 3) trigger risultanti dalla progettazione (è opportuno discutere prima con me quali trigger scrivere)
6. analisi dei dati in R
   - 2 o 3 esempi di query significative per una semplice analisi statistica (es.: trend o distribuzione di popolazione) realizzate interfacciando R al DBMS e visualizzazione del prodotto del risultato delle query attraverso opportuni grafic

Warnings

Warning:

- No ad aggiunte tecniche e complicate o estensioni troppo grandi del dominio. Ad esempio:
  - NON aggiungere 4 attributi derivati: 1 o 2 significativi bastano per mostrare che sapete eseguire l'analisi delle ridondanze
  - se si parla della gestione di aeroporti e compagnie aeree NON aggiungere (se non è nelle specifiche) tutta una parte nuova di dominio che riguarda, ad esempio, la gestione della vendita dei biglietti ai clienti
- *Query SQL*: discuterne prima con me.
  - 7 o 8 operazioni da implementare:
    - 1 o 2 di inserimento
    - 1 o 2 di aggiornamento o cancellazione
    - tutte le altre sono query/interrogazioni
  - non troppo semplici

Warnings

SI: aggiunte semplici di elementi mancanti importanti per analisi successive. Ad esempio:

- *aggiunta obbligatoria*: attributo derivato se non è presente nelle specifiche (con relative operazioni di lettura e modifica) per poter eseguire poi l'analisi delle ridondanze sull'attributo derivato aggiunto
- *aggiunta obbligatoria*: lista di operazioni frequenti sugli attributi derivati con relative frequenze (3 o 4 operazioni son sufficienti, ma che permettano l'analisi delle ridondanze)
- *aggiunta facoltativa*: aggiungere uno storico ad una relazione (*e.g.*, tra impiegato e datore di lavoro) se lo schema concettuale appare troppo semplice