

# Introduzione al corso di Laboratorio di Basi di Dati

Dario Della Monica

21 ottobre 2021

# Informazioni docente

Dario Della Monica

- ▶ **email:** `dario.dellamonica@uniud.it`
- ▶ **web:** `http://users.dimi.uniud.it/~dario.dellamonica/`
  - ▶ [Home] >> Teaching >> Laboratorio di Basi di Dati (2021-2022)
- ▶ **ufficio:** NN1, 2° piano, nodo nord, polo scientifico “Rizzi”
- ▶ altre info (cell, skype, ...) sulla mia pagina web
- ▶ **ricevimento:** martedì, 16:00–18:00 o (meglio) su appuntamento
- ▶ annunci sul “post-it” nella mia homepage



# Materiale corso

- *Basi di dati*  
Atzeni-Ceri-Fraternali-Paraboschi-Torlone  
Quinta edizione  
McGraw-Hill

altre edizioni vanno bene lo stesso; 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](#), ...) disponibile sulla mia pagina web

# Software

- ▶ **ChronoGeoGraph** (by dott.ssa **Donatella Gubiani**): strumento software per la progettazione concettuale di basi di dati spatio-temporali (**schemi Entità-Relazioni**)
  - ▶ scaricabile dalla mia pagina web
  - ▶ sembra non funzionare in WinXP e in Win10
  - ▶ potete usare qualsiasi altro tool preferiate a patto che la notazione sia la stessa (o quasi) — se avete dubbi ne parliamo
    - ▶ ERDplus — <https://erdplus.com/>
    - ▶ Draw.io — <https://app.diagrams.net/>
    - ▶ Diagram Editor — <https://www.diagrameditor.com/>
    - ▶ ... ce ne sono molti altri sul web; si accettano suggerimenti
- ▶ **PostgreSQL**: DataBase Management System (**DBMS**) gratuito e open-source
  - ▶ <https://www.postgresql.org/>
  - ▶ per tutti i sistemi operativi (windows, linux, mac)
- ▶ **pgAdmin**: piattaforma per amministrazione e sviluppo di basi di dati in PostgreSQL (**interfaccia grafica per PostgreSQL**)
  - ▶ <https://www.pgadmin.org/>
  - ▶ per tutti i sistemi operativi (windows, linux, mac)
- ▶ Istruzioni per installazione e configurazione di PostgreSQL e pgAdmin sulla mia pagina web (in inglese)
- ▶ **R** e **RStudio**: linguaggio di programmazione orientato ad analisi statistiche e produzione di grafici e relativo ambiente di lavoro user-friendly



# Modalità delle lezioni

- ▶ si tratta di un laboratorio
  - ▶ poche lezioni frontali
  - ▶ alcune lezioni dimostrative con l'uso del laptop (per il linguaggio R)
  - ▶ gli studenti lavorano al progetto ed il docente è a disposizione per chiarimenti
  - ▶ scenario ideale: il progetto viene svolto interamente o per maggior parte durante le 36 ore di laboratorio
- ▶ lezioni: quando e dove?
  - ▶ giovedì 15:30-18:30
  - ▶ possibili vari cambiamenti per conciliare gli impegni accademici e istituzionali del prof. Montanari
  - ▶ su Microsoft Stream sono disponibili le registrazioni delle lezioni dell'anno scorso
    - ▶ ad esempio, ecco i link alla prima lezione dell'a.a. 2020/2021:  
[1<sup>a</sup> parte](#), [2<sup>a</sup> parte](#), [3<sup>a</sup> parte](#)
- ▶ comunicazioni
  - ▶ annunci in classe, pagina web docente
- ▶ si raccomanda interattività

# Esame: progetto di gruppo

- ▶ progettazione e implementazione di una basi di dati
- ▶ gruppi di 3/4 persone
  - ▶ si possono fare **poche eccezioni...** ma **poche**
- ▶ voto unico (esame di Basi di Dati e Laboratorio)
  - ▶ peso del voto dell'esame di teoria: 75%
  - ▶ peso del voto del progetto: 25%
  - ▶ bonus opzionale a discrezione dei docenti
  - ▶ è necessario raggiungere almeno la sufficienza (18) sia nella teoria che nel laboratorio

# Progetto “standard” ...

- ▶ formare i gruppi e registrarsi via email (a partire già da oggi)
  - ▶ UNA email per gruppo, né più né meno
  - ▶ inviata da un membro del gruppo
  - ▶ contenente la lista di tutti i membri del gruppo (nomi completi, mittente incluso)
- ▶ assegnazione dei progetti ai gruppi casuale, basata su politica First-In-First-Served

... e progetti “alternativi” (da discutere con il docente)

- ▶ progetti su domini a scelta
- ▶ migliorare il tool ChronoGeoGraph

**IMPORTANTE!!!** tutte le comunicazioni che riguardano il progetto devono includere in Cc: tutti i membri del gruppo

# Informazioni di carattere organizzativo sul progetto

- ▶ le **tempistiche** per la consegna del progetto sono flessibili, l'esame di teoria viene conservato in attesa che il progetto venga ultimato e viceversa.
  - ▶ **ATTENZIONE!!!** Garanzia solo per quest'anno accademico. Se dovesse cambiare il docente... nuovo docente, nuove regole
  - ▶ **SUGGERIMENTO:** svolgere il progetto di pari passo con l'andamento del corso
  - ▶ la registrazione avviene durante il primo appello successivo
- ▶ **materiale** da inviare
  - ▶ relazione (da consegnare via email **in formato pdf**)
  - ▶ lunghezza: non c'è una regola ma tipicamente tra le 15 e le 30 pagine
  - ▶ il **focus** deve cadere sulle fasi di progettazione (analisi dei requisiti, progettazione concettuale, logica e fisica) piuttosto che sull'implementazione che ha un'importanza ridotta in questo corso
- ▶ **valutazione** (volta 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):
    - a. potete accettare la valutazione
    - b. oppure potete scegliere di sostenere una discussione orale basata sulla relazione e sull'implementazione della base di dati



# Obiettivi del progetto (1)

- ▶ importanza della corretta interazione tra le varie fasi della progettazione (**ingegnerizzazione del software**)
  - ▶ ogni fase produce uno o più documenti che rappresentano l'input per la fase successiva
  - ▶ coerenza e uniformità tra documenti prodotti nelle diverse fasi
  - ▶ prodotti della progettazione: **non solo SQL ma anche e SOPRATTUTTO** gli schemi prodotti al termine (o durante) le varie fasi (es. schema concettuale, logico, ecc.). Importanti perché:
    - ▶ rappresentano l'unica documentazione su cui basare la fase successiva
    - ▶ sono 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

Considerate che un errore nelle fasi iniziali del ciclo di produzione del software a conseguenze molto più gravi (difficoltà nella sua identificazione, risorse spese per correggerlo)

## Obiettivi del progetto (2)

- ▶ integrazione specifiche per coprire situazioni e costrutti studiati durante il corso. **Attenzione** ad usare livello di complessità adeguato
  - ▶ NO: aggiunte tecniche e complicate o estensioni troppo grandi del dominio. Ad esempio:
    - ▶ NON aggiungere 4 attributi derivati: 1 o 2 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
  - ▶ SI: aggiunte semplici di elementi mancanti importanti per analisi successive. Ad esempio:
    - ▶ aggiungere un 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
    - ▶ aggiungere uno storico ad una relazione (es.: tra impiegato e datore di lavoro)
- ▶ dimostrare di avere ben chiare le difficoltà e gli aspetti **concettualmente** critici ed interessanti della progettazione (ridondanze, generalizzazioni, vincoli aziendali, relazioni ternarie, ecc.)

# Fasi del progetto/struttura della relazione (1)

→ traduciamo i requisiti in E/R (perdiamo espressività ogni volta a vantaggio della chiarezza)

## 1. raccolta e analisi dei requisiti (dati e operazioni)

↳ produrre un documento delle specifiche prendendo quello che ho disambiguandolo.

## 2. progettazione concettuale (dati)

ER + VINCOLI AZIENDALI → RELAZ + VINC. AZIENDALI NUOVI  
↳ risolti in fase di progettaz.  
con i TRIGGER

- ▶ tiene conto del modello concettuale dei dati, totalmente astratto, astrae da modello logico e fisico (**cosa**)
- ▶ produce uno **schema concettuale**
  - ▶ **diagramma E/R** (o schema E/R)

## 3. progettazione logica (**dati**, operazioni usate per ottimizzazioni)

- ▶ 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** (noi adottiamo il paradigma **relazionale**)
  - a. ristrutturazione del modello E/R (analisi delle ridondanze basate su tavole dei valori e delle operazioni, eliminazione delle generalizzazioni, ecc. — di tutti i costrutti dell'E/R non esprimibili nel relazionale)
  - b. traduzione nello **schema relazionale**

# Fasi del progetto/struttura della relazione (2)

## 4. progettazione fisica (**dati**, operazioni usate per ottimizzazioni)

- ▶ tiene conto di come vengono memorizzati fisicamente i dati (**come**)
- ▶ produce il codice per la definizione delle relazioni ma **soprattutto** per la definizione degli **indici**
  - a. analisi e scelta di opportuni indici
  - b. definizione di relazioni e indici in **SQL**

## 5. implementazione (**operazioni**)

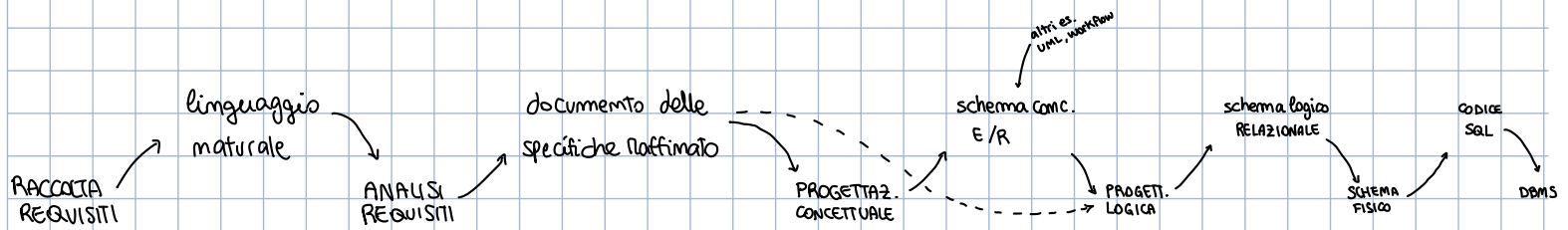
- ▶ popolamento della base di dati (a mano o automatizzato, es.: R)
- ▶ definizione di pochi (2/3) trigger risultanti dalla progettazione
- ▶ definizione di poche (2/3) query significative (operazioni frequenti)

## 6. analisi dei dati in R

- ▶ 2/3 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 grafici

# Prossime lezioni

- ▶ progettazione concettuale di una base di dati
  - ▶ analisi e raccolta dei requisiti
  - ▶ pattern di progetto
  - ▶ strategie di progetto
  - ▶ qualità di uno schema concettuale
- ▶ cap. 7 “Progettazione concettuale”  
(Atzeni, Ceri, Fraternali, Paraboschi, Torlone, *Basi di dati*, 5° ed.)  
attenti a numerazione diversa in altre edizioni
- ▶ nota: le slides si riferiscono al capitolo 6 “Conceptual design”  
di una versione precedente (in inglese) del testo



livello utenti  
(view model)

→ indep. logica: se faccio una modifica alla struttura, l'utente non deve percepire le modifiche garantito dalle applicaz./pagine WEB

livello logico-concett.

→ indep. fisica: cambiando S.O., i nostri comandi SQL andranno bene anche nel nuovo sistema garantito dal DBMS

livello fisico

