

# Laboratorio di Basi di dati

## Ciclo di vita e progettazione dei sistemi informativi

Luca Anselma

[luca.anselma@unito.it](mailto:luca.anselma@unito.it)

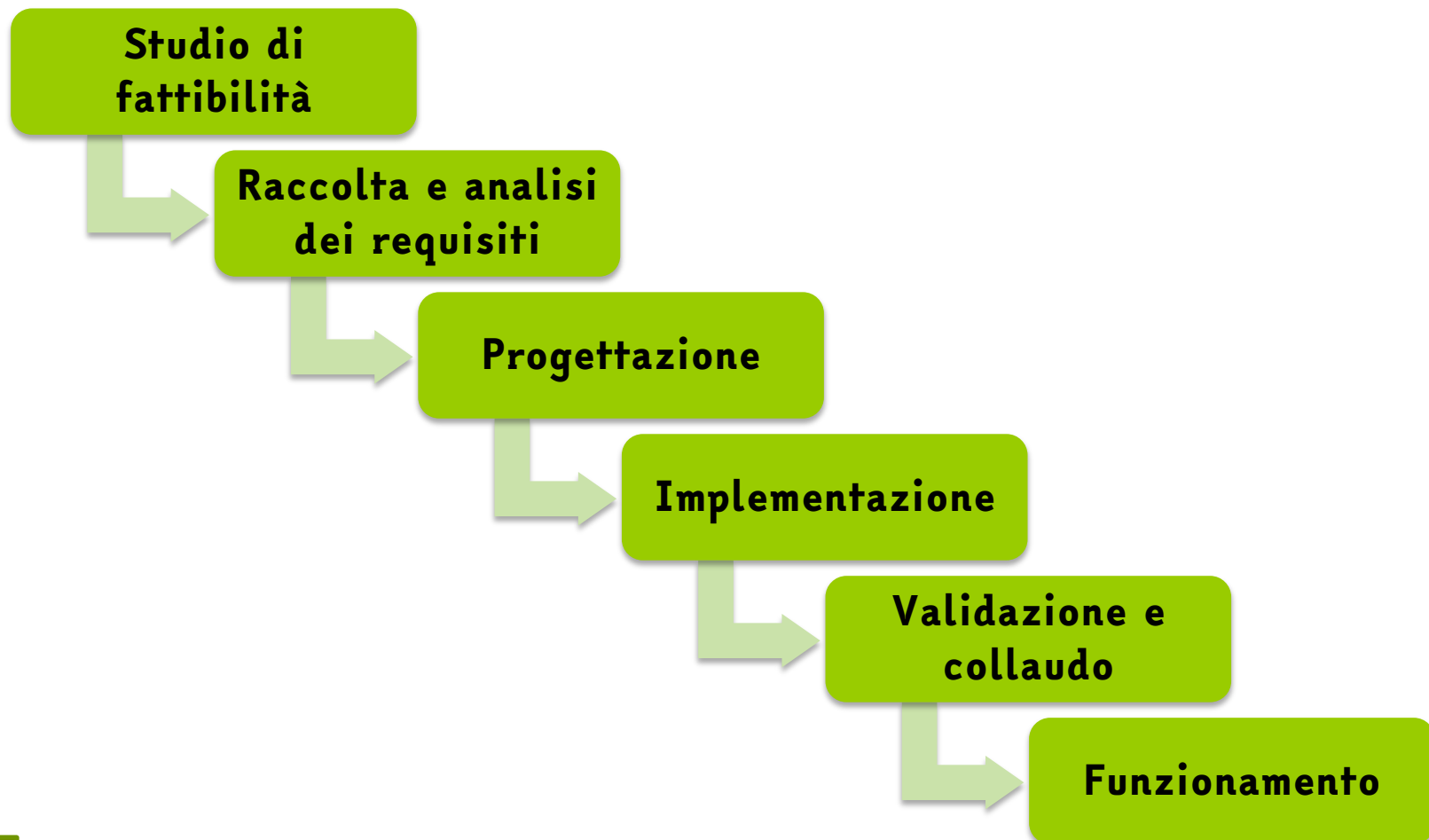


<https://dilbert.com/strip/1995-11-17>

# Ciclo di vita di un sistema informativo

---

Il ciclo di vita di una base di dati consiste di varie fasi. La progettazione è una di questa fasi.



# Ciclo di vita di un sistema informativo

---

- ❑ **Studio di fattibilità:** si stabiliscono le possibili soluzioni alternative e le priorità in relazione ai vincoli imposti (costo, tempo, qualità, funzionalità)
- ❑ **Raccolta e analisi dei requisiti:** si individuano le *proprietà* e *funzionalità* che il sistema informativo dovrà avere. Si definiscono inoltre i requisiti hardware e software. In questa fase tipicamente è necessario interagire strettamente con gli esperti di dominio, che conoscono le funzionalità del sistema informativo
- ❑ **Progettazione:** si costruisce un modello formale e dettagliato del sistema. Comprende progettazione dei dati e progettazione delle applicazioni

# Ciclo di vita di un sistema informativo

---

- ❑ **Implementazione:** si realizza il sistema informativo seguendo le specifiche di progetto: viene costruita e popolata la base di dati e scritto il codice dei programmi
- ❑ **Validazione e collaudo:** si verifica il corretto funzionamento e la qualità del sistema informativo
- ❑ **Funzionamento:** si “mette in produzione” il sistema realizzato. Successivamente manutenzione e revisioni
- ❑ Il ciclo di vita non è quasi mai sequenziale con il passaggio da una fase alla successiva perché spesso occorre rivedere decisioni prese in attività precedenti, quindi tipicamente contiene sottocicli
- ❑ Talvolta si usa prototipizzazione, cioè si realizzano velocemente versioni semplificate per sperimentare le funzionalità

# Le variabili fondamentali

---

- La progettazione di un sistema informativo è un'attività complessa
- Dipende da quattro variabili fondamentali:
  - **Costo**
  - **Tempo**
  - **Qualità**
  - **Funzionalità**

# Le variabili fondamentali

---

- ❑ Sarebbe bello minimizzare (o limitare) *tempo* e *costo* e massimizzare *qualità* e *funzionalità*
- ❑ Purtroppo non sono variabili indipendenti. Questo comporta il fatto che, se si impone un limite al costo e al tempo, bisogna scegliere o di implementare minori funzionalità di maggiore qualità o di diminuire la qualità a favore di un maggiore numero di funzionalità

# Progettare un sistema informativo

---

- La progettazione di un sistema informativo riguarda due aspetti principali:
  - **Dati**: la parte più *stabile*, la struttura dei dati subisce minime variazioni nel tempo
  - **Funzionalità**: la parte meno stabile. Nuove funzionalità vengono aggiunte e le presenti possono essere modificate
- Per es. la gestione patrimoniale: nonostante i patrimoni rimangano strutturalmente stabili, una serie di nuovi servizi è periodicamente attivata
- Ci concentreremo sui dati (raccolta e analisi dei requisiti e progettazione della base di dati). Le altre attività vengono trattate nei corsi di ingegneria del software
- Progettare un database relazionale è diverso da progettare applicazioni: mentre in un'applicazione è facile cambiare requisiti e funzionalità, è difficile/costoso fare evolvere gli schemi e le istanze dei database relazionali

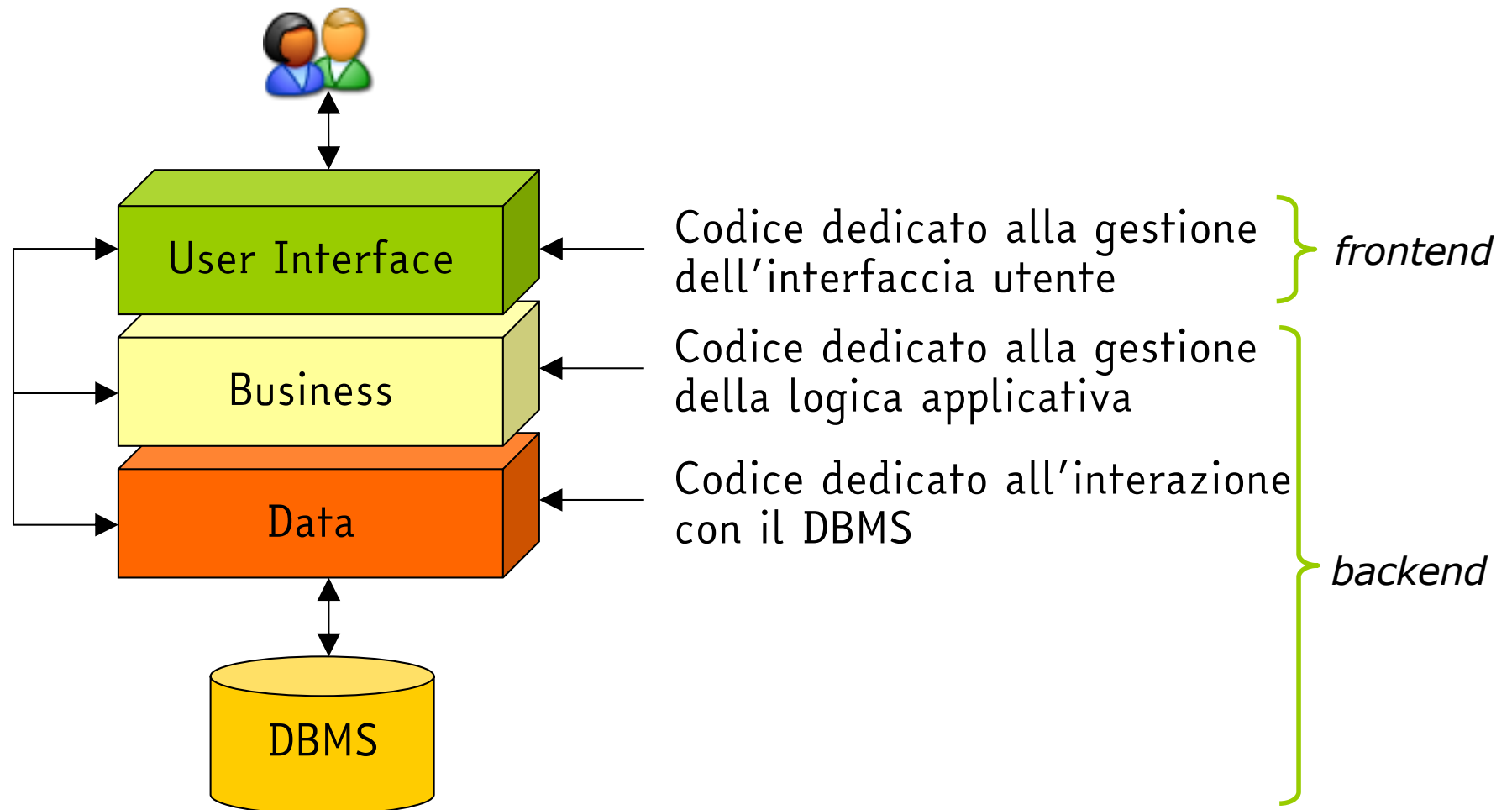
# Architettura di una applicazione

---

- ❑ Le “moderne” applicazioni sono progettate come una successione di strati (o livelli)
- ❑ Ogni strato comunica tipicamente solo con quelli adiacenti (sottostanti o sovrastanti)
- ❑ Ogni strato implementa una “interfaccia” per la comunicazione, ovvero un insieme di regole che gli strati adiacenti devono rispettare per comunicare con esso
- ❑ Ci sono diversi modelli per progettare applicazioni a livelli
- ❑ Uno dei modelli maggiormente affermati è quello 3-tier



# Modello 3-tier

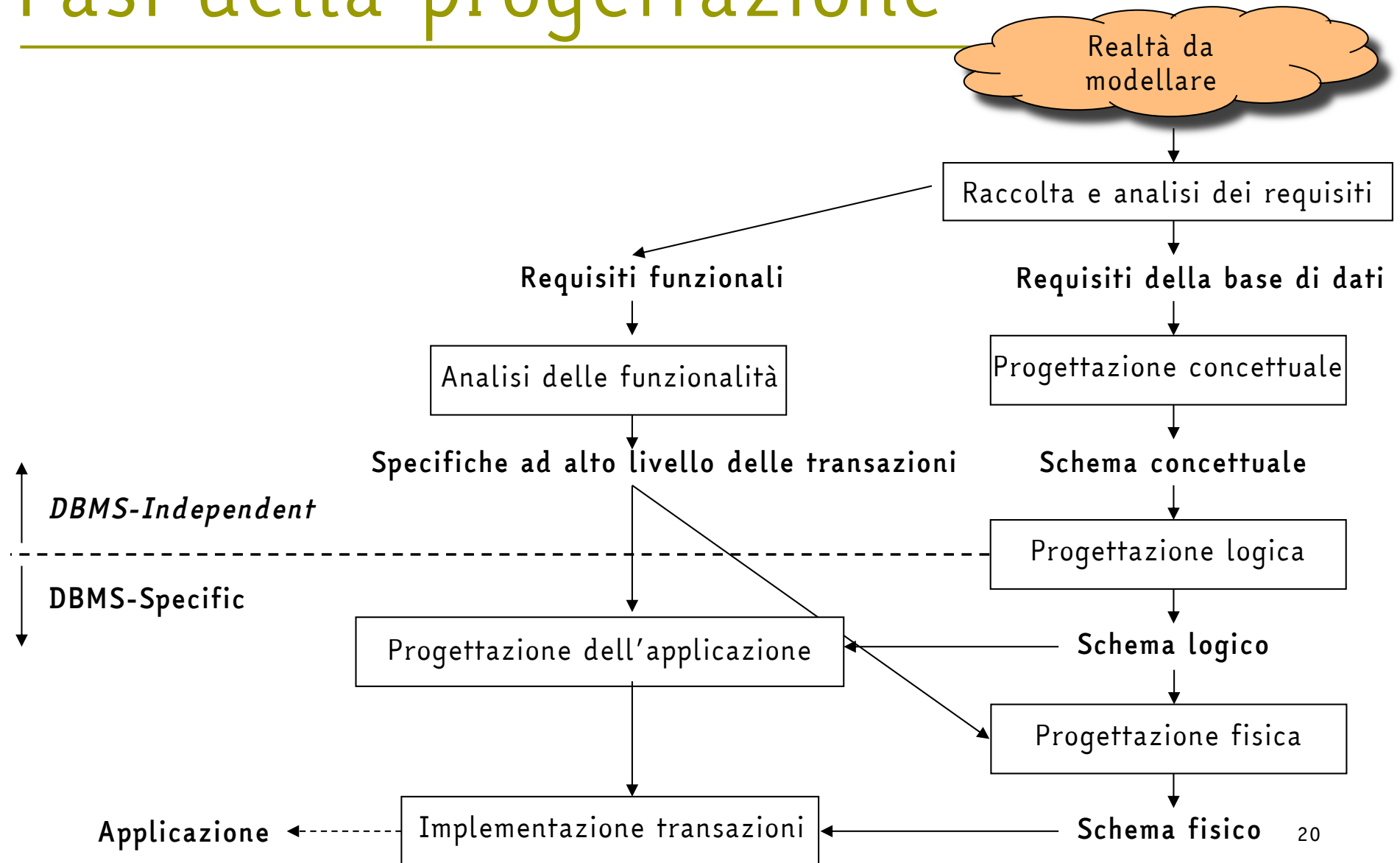


# Modello 3-tier

---

- ❑ Ogni strato implementa un insieme di *API* (*Application Programming Interface*) che gli strati adiacenti possono invocare
- ❑ Ci sono applicazioni 3-tier nelle quali la distinzione tra i tre livelli non è così netta

# Fasi della progettazione



# Modello logico

---

- ❑ Usa un modello dei dati, che è l'insieme dei costrutti utilizzati per organizzare i dati di interesse e descriverne la dinamica
- ❑ Componente fondamentale: **meccanismi di strutturazione** (analoghi ai costruttori di tipo dei linguaggi di programmazione)
- ❑ Ad esempio, il **modello relazionale** è un modello logico che prevede il costruttore **relazione**, il quale definisce insiemi di record omogenei

# Modello concettuale

---

- ❑ I modelli concettuali permettono di descrivere i dati in maniera *indipendente* dal modello logico
- ❑ Non sono disponibili nei DBMS (commerciali)
- ❑ Si usano per analizzare in modo rigoroso e astratto la realtà di interesse
- ❑ Nella fase di progettazione concettuale si rappresenta il contenuto informativo del DB. In questa fase il progettista **non deve preoccuparsi** (N.B. diverso da "può non preoccuparsi") delle modalità e dell'efficienza con cui le informazioni verranno codificate in un sistema reale

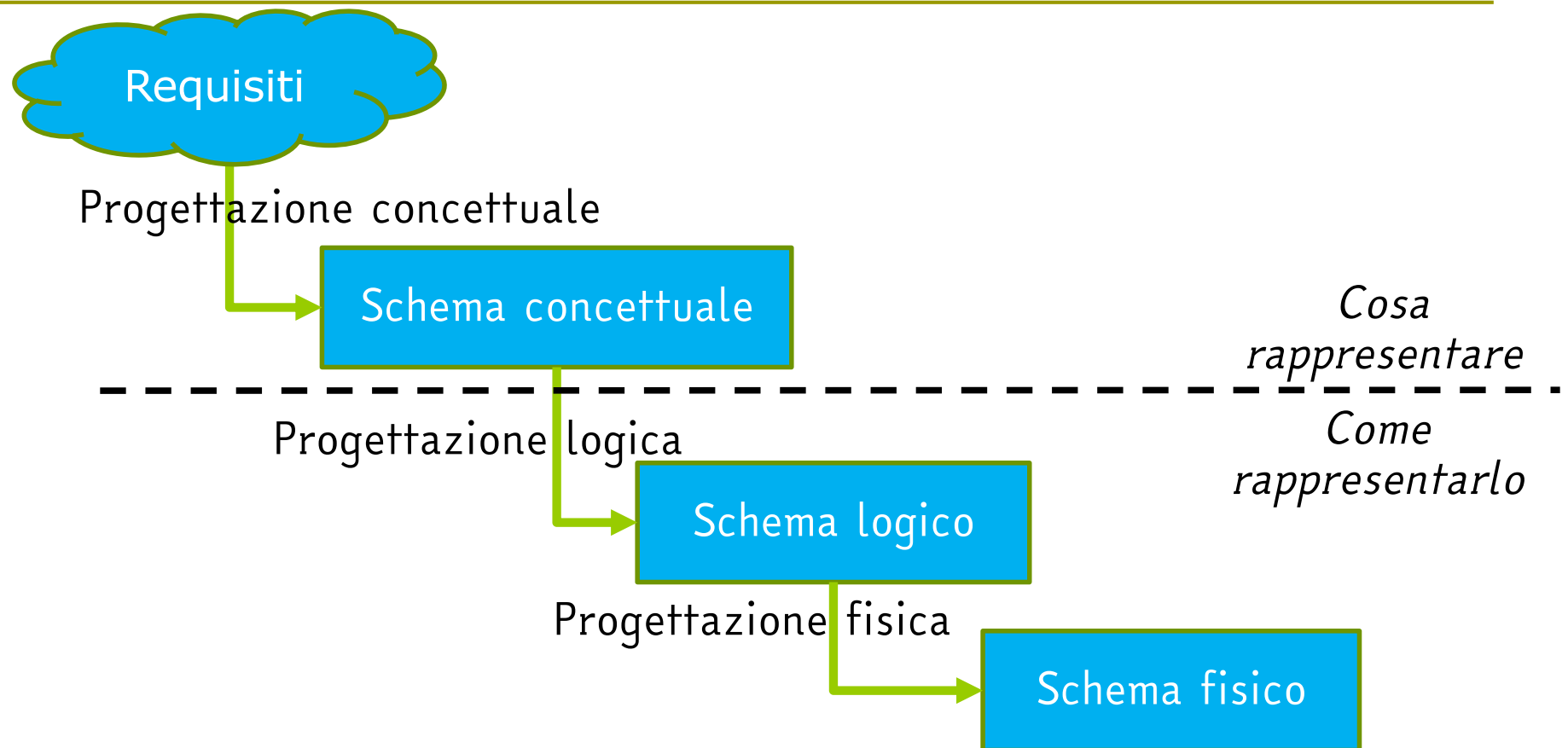
# Considerazioni metodologiche

---

Perché costruire prima uno schema concettuale dei dati e non direttamente lo schema logico (le tabelle del DB)?

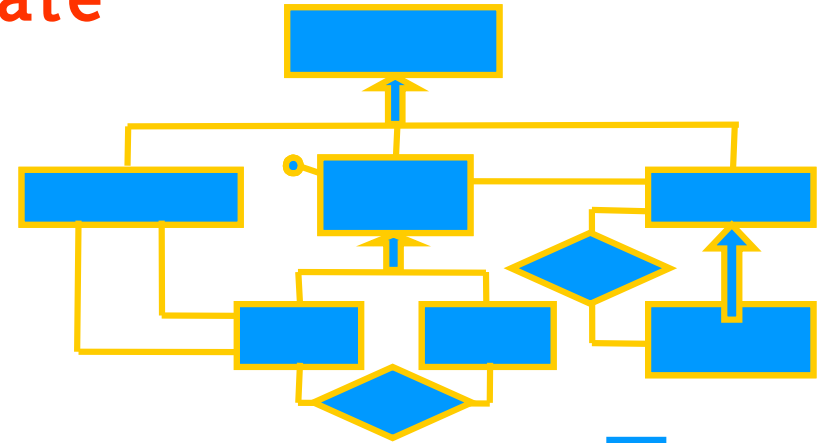
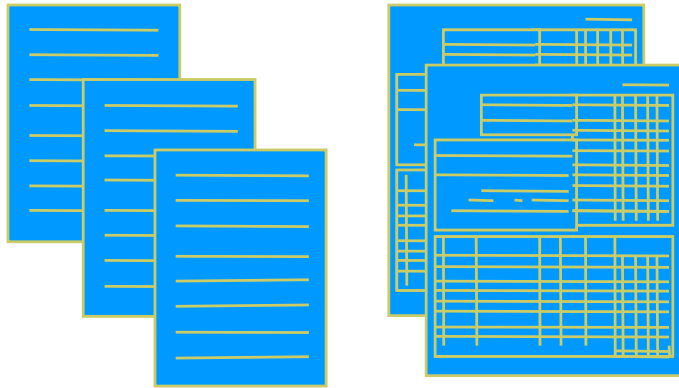
- ❑ I modelli concettuali sono utili per evitare errori nella modellazione
- ❑ Sono più semplici per i non-tecnici (e quindi sono utili a documentare e trasferire conoscenza)
- ❑ Sono DBMS-independent
- ❑ Generalmente sono più espressivi del modello logico

# Progettazione

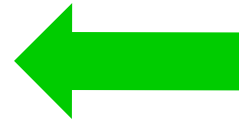
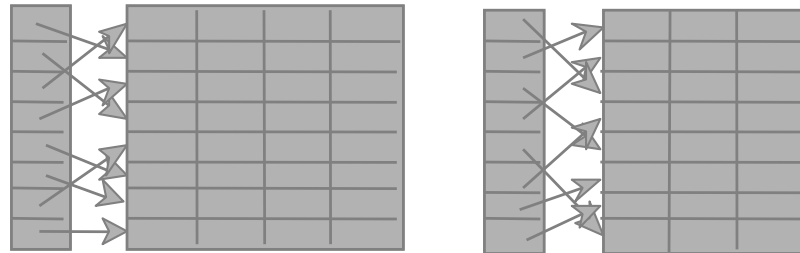


Tutti e tre gli schemi costituiscono il risultato della progettazione e sono utili come documentazione (schema concettuale), descrizione e riferimento per interrogazioni e aggiornamenti (schema logico) e realizzazione e implementazione del DB (schema fisico)

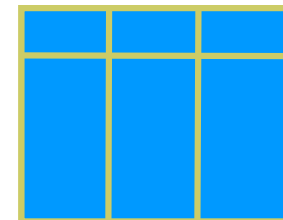
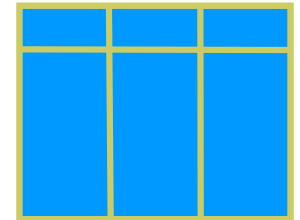
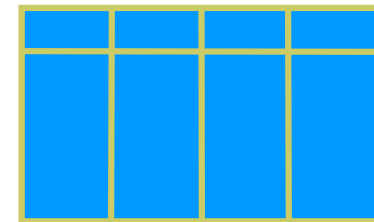
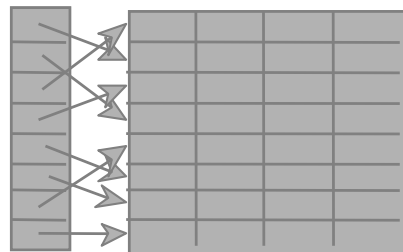
Progettazione  
concettuale



Progettazione  
logica



Progettazione  
fisica





# Reverse engineering

---

- ❑ Esistono situazioni in cui si ha a disposizione lo schema logico e si ha necessità di riprodurre uno schema concettuale
- ❑ È utile quando si vuole produrre documentazione (ad es. secondo il modello EER) a partire da una base di dati già esistente
- ❑ È esattamente il contrario di quello che normalmente si fa nella normale progettazione
- ❑ La problematica viene definita di reverse engineering
- ❑ Molti strumenti CASE semi-automatizzano anche questa fase