

Progettazione Concettuale di Basi di Dati

Strategie, Pattern e Metodologie

Prof. Fedeli Massimo - Tutti i diritti riservati

Fabbrica Digitale

16 dicembre 2025

- 1 Analisi dei Requisiti
- 2 Design Patterns ER
- 3 Strategie di Progettazione
- 4 Metodologia e Qualità
- 5 Caso di Studio: Training Company

Il Ciclo di Vita della Progettazione

La progettazione di un database si articola in fasi distinte:

- **Analisi dei Requisiti ("COSÀ"):** Raccogliere e analizzare le necessità.
- **Progettazione Concettuale ("COSÀ"):** Schema Concettuale (indipendente dal DBMS).
- **Progettazione Logica ("COME"):** Schema Logico (es. Relazionale).
- **Progettazione Fisica ("COME"):** Schema Fisico (file, indici).

Copre compiti interconnessi:

- ① Raccolta dei requisiti (elicitation).
- ② Analisi dei requisiti.
- ③ Costruzione del glossario.
- ④ Costruzione dello schema concettuale.

Da dove provengono le informazioni?

- **Utenti e clienti:** tramite interviste.
- **Documentazione specifica:** documenti ad hoc.
- **Documentazione esistente:** normative, leggi, regolamenti interni, processi aziendali.
- **Soluzioni preesistenti:** software legacy.
- **Modulistica:** form cartacei o digitali in uso.

Attenzione

L'acquisizione è un'attività difficile e non standardizzata. I requisiti iniziali spesso necessitano di raffinamenti successivi.

- Utenti diversi forniscono informazioni diverse.
- Livello manageriale: visione ampia ma poco dettagliata.
- Livello operativo: visione dettagliata ma limitata.

- Effettuare frequenti controlli di comprensione e coerenza.
- Utilizzare esempi di *Use Case* (casi generici e casi limite).
- Chiedere definizioni esplicite e classificazioni.
- Chiedere di distinguere tra aspetti essenziali e periferici.

Regole empiriche per trattare i testi dei requisiti:

- Scegliere il giusto livello di astrazione.
- Mantenere la struttura delle frasi il più standard possibile.
- Spezzare le frasi troppo lunghe o complesse.
- Distinguere le frasi che descrivono ****dati**** da quelle che descrivono ****funzioni****.

Il glossario è fondamentale per uniformare il linguaggio.

- Costruire un glossario per i termini chiave.
- Unificare omonimi e sinonimi sotto un unico termine.
- Chiarire esplicitamente le relazioni tra i termini.
- Ordinare le frasi per concetto.

Esempio di Glossario

Termine	Descrizione	Sinonimo	Relato a
Partecipante	Chi prende parte ai corsi	Studente	Corso, Società
Docente	Il titolare del corso	Insegnante	Corso
Corso	Corso interno, può avere edizioni	Workshop	Docente
Società	Luogo di lavoro attuale o passato	Luogo	Partecipante

Organizzare i requisiti in gruppi omogenei di frasi facilita la progettazione.

- Frasi generali (visione d'insieme).
- Frasi relative ai Partecipanti.
- Frasi relative ai Corsi.
- Frasi relative ai Docenti.

[Immagine suggerita: Diagramma a blocchi che mostra il raggruppamento delle frasi per argomento]

Dai Requisiti allo Schema Concettuale

Come mappare i termini nei costruttori ER?

- **Entità:** se il termine ha proprietà rilevanti e descrive oggetti autonomi.
- **Attributo:** se è un termine semplice senza ulteriori specifiche.
- **Relazione (Associazione):** quando un termine collega altri termini.
- **Generalizzazione:** quando un termine è un caso più generale di un altro.

- Sono "best practices" comuni nella progettazione software per problemi ricorrenti.
- Gli ingegneri del software li usano quotidianamente.
- Esistono pattern specifici per la modellazione ER (Entity-Relationship).

Pattern: Reificazione di Attributi

Quando un attributo nasconde concetti complessi, deve diventare un'Entità.

Esempio: "Azienda" come semplice stringa vs "Azienda" come entità relazionata.



L'azienda è un concetto diverso dall'impiegato (es. quando le aziende sono coinvolte in altre relazioni).

Pattern: Part-of (Composizione/Aggregazione)

Le relazioni (1,N) possono rappresentare concetti di "parte di".

- **Composizione:** Un cinema è fatto di sale (senza cinema, la sala non ha senso).
- **Aggregazione:** Un team è composto da esperti (l'esperto esiste anche senza il team).



Pattern: Instance-of

A volte servono due entità distinte: una per la rappresentazione astratta e una per l'istanza concreta.

Esempio:

- **Volo (Flight)**: Il concetto astratto (es. Volo AZ123 Roma-Milano delle 8:00).
- **Schedulazione (Scheduled)**: L'evento specifico in una data precisa.



Pattern: Reificazione di Relazioni Binarie (1)

Quando una relazione ha attributi complessi o deve essere collegata ad altre entità, può essere trasformata (reificata) in un'entità.

Esempio Studente-Esame-Lezione: Un esame collega uno studente e una lezione e ha un voto.



Pattern: Reificazione di Relazioni Ricorsive

Esempio Partita (Match): Una partita potrebbe essere una relazione binaria tra due squadre (Team). Tuttavia, poiché due squadre possono giocare contro più volte, è meglio reificare il concetto di "Match".

[Inserire qui diagramma ER slide 30: Team - Home/Visiting - Match]

Pattern: Reificazione di Relazioni (Strumenti Musicali)

Un musicista suona uno strumento in un'orchestra.

- Un musicista suona uno strumento.
- Lo stesso strumento (fisico o tipologia) può essere suonato molte volte.



Pattern: Caso Specifico (Sottoinsiemi)

Le generalizzazioni sono usate per definire casi specifici di un'entità.

Esempio: Non tutti gli impiegati gestiscono progetti. Solo i "Manager" lo fanno.



Pattern: Storizzazione (History)

Come mantenere traccia dei cambiamenti nel tempo?

- **Dati anagrafici:** Indirizzo corrente vs indirizzi passati.
- **Stato software:** Versione corrente vs versioni legacy.

Si usano generalizzazioni per separare lo stato "Corrente" dallo stato "Storico".



Pattern: Storizzazione (Employment)

Esempio complesso: Storia lavorativa di un impiegato.

- Un impiegato può avere un impiego corrente e molti passati.
- Si può modellare con due relazioni distinte o reificando l'impiego (Employment).



Pattern: Estensione di un Concetto

Le generalizzazioni possono essere usate per estendere implementazioni correnti.

Esempio: Un progetto "Accettato" richiede informazioni aggiuntive (data inizio, fondi) rispetto a un progetto generico in fase di proposta.



Pattern: Relazioni Ternarie

Quando tre entità sono coinvolte simultaneamente.

Esempio: Un impiegato lavora su un task in un ufficio specifico.

- Gli impiegati lavorano su vari task in diversi uffici.
- Gli uffici ospitano diversi impiegati su vari task.



Reificazione Relazione Ternaria (1)

Spesso le ternarie vengono reificate per maggiore chiarezza o per aggiungere vincoli.
Ogni "Lavoro" (Work) è definito da un Impiegato che lavora in un dato Ufficio per un dato Task.



Reificazione Relazione Ternaria (2)

Se un task può essere svolto da un solo operatore e solo in un ufficio, la reificazione può essere semplificata in una catena di relazioni binarie.



Come trasformare i requisiti in uno schema ER? Esistono tre strategie principali:

- 1 Top-down
- 2 Bottom-up
- 3 Inside-out

Strategia Top-down

Si parte da concetti astratti e macroscopici per scendere nel dettaglio.

- Schema Draft → Schema Intermedio → Schema Finale.
- Si parte con poche entità molto generiche.



Esempio Top-down (Raffinamento)

- 1 Concetto iniziale: **Exam**.
- 2 Raffinamento: Trasformazione in relazione tra **Student** e **Lecture**.



Esempio Top-down (Attributi e Generalizzazione)

- 1 Concetto: **Employee**.
- 2 Aggiunta dettagli: Attributi (Surname, Age, Wage).
- 3 Oppure specializzazione: **People** diventa generalizzazione di **Man** e **Woman**.

Si parte dai dettagli (attributi, specifiche) per aggregarli in concetti più ampi.

- Requisiti elementari → Schemi parziali → Integrazione.



Esempio Bottom-up

- 1 Requisiti su un impiegato → Creazione Entità **Employee**.
- 2 Requisiti su esami → Creazione relazione **Student-Exam-Lecture**.
- 3 Unione di attributi o entità separate in gerarchie (Man/Woman → People).

È una variante del bottom-up. Si parte dai concetti più importanti ("nucleus") e ci si espande "a macchia d'olio" seguendo le relazioni.

- 1 Identificazione Entità principale (es. **Employee**).
- 2 Aggiunta attributi (Code, Surname).
- 3 Scoperta relazioni: appartiene a **Dept**, è supervisionato da...
- 4 Espansione: Dept ha un nome, Employee lavora a un **Project**, ecc.

Esempio Inside-out (Espansione)

Si naviga attraverso le associazioni:

- Employee \leftrightarrow Dept (Belonging)
- Employee \leftrightarrow Project (Enrollment)
- Dept \leftrightarrow Office (MadeOf)



La Strategia "Mista" (Best Practice)

Nella pratica si usa uno stile misto:

- 1 **Sketch iniziale:** Creare uno schema di massima con le entità più rilevanti (Top-down/Inside-out).
- 2 **Decomposizione:** Dividere i requisiti in sottogruppi.
- 3 **Raffinamento:** Espandere i dettagli (Bottom-up), integrare le parti.

- ➊ **Analisi dei Requisiti:** Glossario, pulizia ambiguità, raggruppamento frasi.
- ➋ **Caso Base (Sketch):** Schema bozza con concetti principali.
- ➌ **Decomposizione:** Partizionamento dei requisiti per aree di interesse.
- ➍ **Iterazione (per ogni area):**
 - Raffinare i concetti base usando i requisiti.
 - Aggiungere concetti mancanti.
- ➎ **Integrazione:** Unire i sotto-schemi.
- ➏ **Analisi di Qualità:** Verifica finale.

Uno schema è buono se soddisfa:

- **Correttezza (Correctness):** Uso corretto dei costrutti sintattici e semantici.
- **Completezza (Completeness):** Tutti i requisiti sono rappresentati.
- **Chiarezza (Clarity):** Lo schema è leggibile e comprensibile.
- **Minimalità (Minimality):** Evitare ridondanze inutili.

Caso di Studio: Training Company

Applichiamo la metodologia a un caso reale.

Obiettivo: Un database per una società di formazione.

- Gestione corsi, lezioni e insegnanti.
- 5000 studenti, 200 corsi, 300 insegnanti.

Step 1: Sketch Iniziale

Partiamo dalle entità più rilevanti citate esplicitamente o implicitamente.

- Lecture (Lezione/Corso)
- Participant (Studente)
- Lecturer (Docente)

Relazioni base: "Presence" (Partecipazione) e "Teaching" (Insegnamento).

*[Inserire schema slide 66: Participant - Lecture
- Lecturer]*

Step 2: Raffinamento Partecipanti (1)

Requisiti: Dati anagrafici (ID, Tax code, name, age...), datore di lavoro, corsi passati e correnti con voto.

Analisi:

- Distinguere corsi attuali e passati (Storicizzazione).
- Attributi complessi.

Step 2: Raffinamento Partecipanti (2)

Specifiche:

- Se studente è *freelance*: area di interesse, titolo onorifico.
- Se studente è *dipendente*: livello, posizione, struttura aziendale.

→ Suggerisce una **Generalizzazione** (Employee vs Freelance).

Step 2: Raffinamento Partecipanti (3) - Employer

Requisiti: Memorizzare datori di lavoro attuali e passati (Nome, indirizzo, telefono).

→ L'azienda (Employer) diventa un'Entità a sé stante. → Relazione storicizzata con il Partecipante (CurrentEmpl vs PastEmpl).



Step 3: Raffinamento Corsi

Requisiti:

- Corsi hanno codice, titolo.
- Diverse **edizioni** (data inizio/fine, numero partecipanti).
- Ogni edizione ha lezioni in giorni/stanze specifici.

Modellazione: Lecture (generico) → Edition (specifico nel tempo) → Lesson (singola lezione giornaliera). Usiamo pattern Part-of o entità deboli.



Step 4: Raffinamento Docenti

Requisiti:

- Dati anagrafici.
- Corsi che possono insegnare (skill) vs corsi che hanno insegnato (storia).
- Possono essere dipendenti interni o contractor esterni.

Modellazione: Generalizzazione (Independent vs House Employee). Relazioni multiple con i corsi.



Step 5: Integrazione (Schema Intermedio)

Collegiamo i pezzi.

- Participant si lega a Edition (CurrentPresence) e a Lecture/Edition passate (PastPresence).
- Lecturer si lega a Edition (Current teaching) e Lecture (History/Ability).



Il risultato finale unisce:

- Gerarchia dei Partecipanti (Employee/Freelance).
- Relazione con Employer.
- Struttura del Corso (Lecture-Edition-Lesson).
- Gerarchia Docenti.
- Relazioni di frequenza e insegnamento.



- La progettazione concettuale è un processo iterativo.
- L'uso di pattern semplifica problemi complessi (es. storicizzazione).
- Una metodologia mista (Sketch + Raffinamento) è la più efficace.
- La qualità si misura in completezza, correttezza e minimalità.

Grazie per l'attenzione