

Esempio di progettazione di un database con elencazione delle fasi progettuali

1. Definizione delle ENTITÀ

- **Studente:** dati anagrafici degli studenti
- **Insegnamento:** i corsi di studio (materie) tenuti dai docenti
- **Docente:** i professori che tengono gli insegnamenti e gli esami

2. Definizione degli ATTRIBUTI e CHIAVI delle entità

- **Studente** (Matricola_Studente, Cognome, Nome, Data_Nascita)
- **Insegnamento** (Codice_Insegnamento, Descrizione, Anno_di_Corso, Fondamentale)
- **Docente** (Matricola_Docente, Cognome, Nome, Data_Nascita)

Gli attributi sottolineati costituiscono le chiavi primarie delle entità. Non vi sono chiavi candidate, che altrimenti andrebbero evidenziate.

3. DIZIONARIO degli attributi

Attributo	Descrizione	Tipo	Vincolo / Note
ID	ID progressivo che identifica un record	DECIMAL(6,0)	Contatore automatico (per la Matricola)
Cognome	Cognome di una persona	VARCHAR(20)	
Nome	Nome di battesimo di una persona	VARCHAR(30)	
Data	Data cronologica	DATE	Formato ISO AAAA-MM-GG
Denominazione	Denominazione generica	VARCHAR(50)	Ad es. per la descrizione dell'insegnamento
Anno_di_Corso	Anno di corso dell'insegnamento	TINYINT(2) UNSIGNED	BETWEEN 1 AND 6
Flag	Eventuale obbligatorietà (Sì/No) dell'attributo	BOOLEAN	1=Sì 0=No
Codice_Insegnamento	Codice identificativo attribuito all'insegnamento	CHAR(3)	
Anno_Accademico	Anno iniziale e finale	CHAR(9)	Ad es. 2017/2018
Percentuale	Valore percentuale espresso in frazione dell'intero	FLOAT(3,2)	BETWEEN 0.00 AND 1.00 Ad es. 0.85 = 85%

4. Definizione delle RELAZIONI

	Studente	Insegnamento	Docente
Studente	—	R1 (0 ; m)	R2 (1 ; 1)
Insegnamento	R1 (0 ; n)	—	R3 (1 ; 1)
Docente	R2 (0 ; n)	R3 (0 ; n)	—

R1 [m : n] — Studente *segue* Insegnamento
R2 [1 : n] — Docente *assegnaTesi* a Studente
R3 [1 : n] — Docente *tiene* Insegnamento

5. VERIFICA delle RELAZIONI

La relazione R1, essendo multi-a-molti, va trasformata in due relazioni uno-a-molti collegate (sul lato molti) alla nuova Entità **Frequenza** (le frequenze ottenute dagli studenti per gli insegnamenti seguiti), avente gli attributi: (Studente, Insegnamento, Anno_Accademico, Perc_Frequenza) che assorbe le chiavi primarie Studente.Matricola_Studente (= Frequenza.Studente) e Insegnamento.Codice_Insegnamento (= Frequenza.Insegnamento) delle tabelle collegate alla relazione originaria. La chiave primaria è multi-attributo e deve comprendere necessariamente anche l'anno accademico potendo eventualmente uno studente rifrequentare un corso.

Qualora una relazione possieda degli attributi, va anch'essa scissa in due relazioni (1:1 oppure 1:m a seconda della cardinalità della relazione originaria) collegate ad una nuova tabella, avente come attributi di partenza le chiavi primarie delle tabelle collegate alla relazione originaria e gli attributi della stessa, nonché eventuali altri attributi ritenuti utili.

Sono quindi state aggiunte al progetto le seguenti nuove relazioni:

R1a [1 : n] — Uno Studente *frequenta un* Insegnamento per conseguire la Frequenza

R1b [1 : n] — Un Insegnamento *èFrequentato* da uno Studente per conseguire la Frequenza

in sostituzione della relazione R1 che viene eliminata.

	Studente	Insegnamento	Docente	Frequenza
Studente	—	R1 (0 ; m)	R2 (1 ; 1)	R1a (0 ; n)
Insegnamento	R1 (0 ; n)	—	R3 (1 ; 1)	R1b (0 ; m)
Docente	R2 (0 ; n)	R3 (0 ; n)	—	
Frequenza	R1a (1 ; 1)	R1b (1 ; 1)		

6. Controllo delle CHIAVI ESTERNE

Fatto ciò, per ciascuna relazione ora presente nel progetto va verificato che la chiave primaria dell'entità sul lato "uno" sia presente come chiave esterna nell'altra entità della relazione: in caso contrario, va aggiunta in modo che nell'implementazione del progetto nel modello relazionale il DBMS possa creare correttamente le relazioni tra le tabelle.

Dopo questa fase, l'elenco completo delle entità e relativi attributi è il seguente:

- **Studente** (Matricola_Studente: *ID*, *Cognome*: *Cognome*, *Nome*: *Nome*, *Data_Nascita*: *Data*)
- **Insegnamento** (Codice_Insegnamento: *Codice_Insegnamento*, *Descrizione*: *Denominazione*, *Anno_di_Corso*: *Anno_di_Corso*, *Fondamentale*: *Flag*)
- **Docente** (Matricola_Docente: *ID*, *Cognome*: *Cognome*, *Nome*: *Nome*, *Data_Nascita*: *Data*)
- **Frequenza** (Studente => *Studente.Matricola_Studente*, Insegnamento => *Insegnamento.Codice_Insegnamento*, Anno_Accademico: *Anno_Accademico*, *Perc_Frequenza*: *Percentuale*)

dove, per ciascun attributo, è indicato (dopo i due punti) il tipo (l'attributo del dizionario cui corrisponde) oppure — se si tratta di una chiave esterna — la Tabella.Chiave_Primary cui l'attributo è collegato.

7. NORMALIZZAZIONE delle entità

In quest'ultima fase, occorre verificare che ciascuna entità (che diventerà una tabella nel modello relazionale) sia in terza forma normale (3FN). Per fare ciò occorre procedere per gradi.

- (1) Innanzitutto, verificare che ciascuna entità (tabella) sia in 1FN, cioè che abbia la chiave primaria.
- (2) Successivamente verificare che ciascuna entità sia anche in 2FN, cioè che ciascun attributo NON CHIAVE dell'entità dipenda INTERAMENTE da tutte gli attributi della chiave (questo ovviamente se la chiave primaria è multi-attributo, altrimenti l'entità è anche già in 2FN).
- (3) Infine, un'entità è in 3FN se ciascun attributo NON CHIAVE dipende DIRETTAMENTE dalla chiave (mono o multi-attributo che sia), cioè un attributo non chiave non deve dipendere da un altro attributo non chiave ma solo e direttamente dalla chiave primaria.

Qualora un attributo non soddisfi i requisiti sopra elencati va rimosso dall'entità in esame e va inserito in una nuova entità, la cui chiave sarà costituita dagli attributi dai quali l'attributo rimosso dipende.

Questo conclude l'intera progettazione del database. Il progetto può quindi essere "disegnato" in forma grafica, rispettando le regole della metodologia Entity/Relationship (https://it.wikipedia.org/wiki/Modello_E-R):

