



Basi di Dati

Il modello Entità-Relazione (parte I)

Marco Maratea

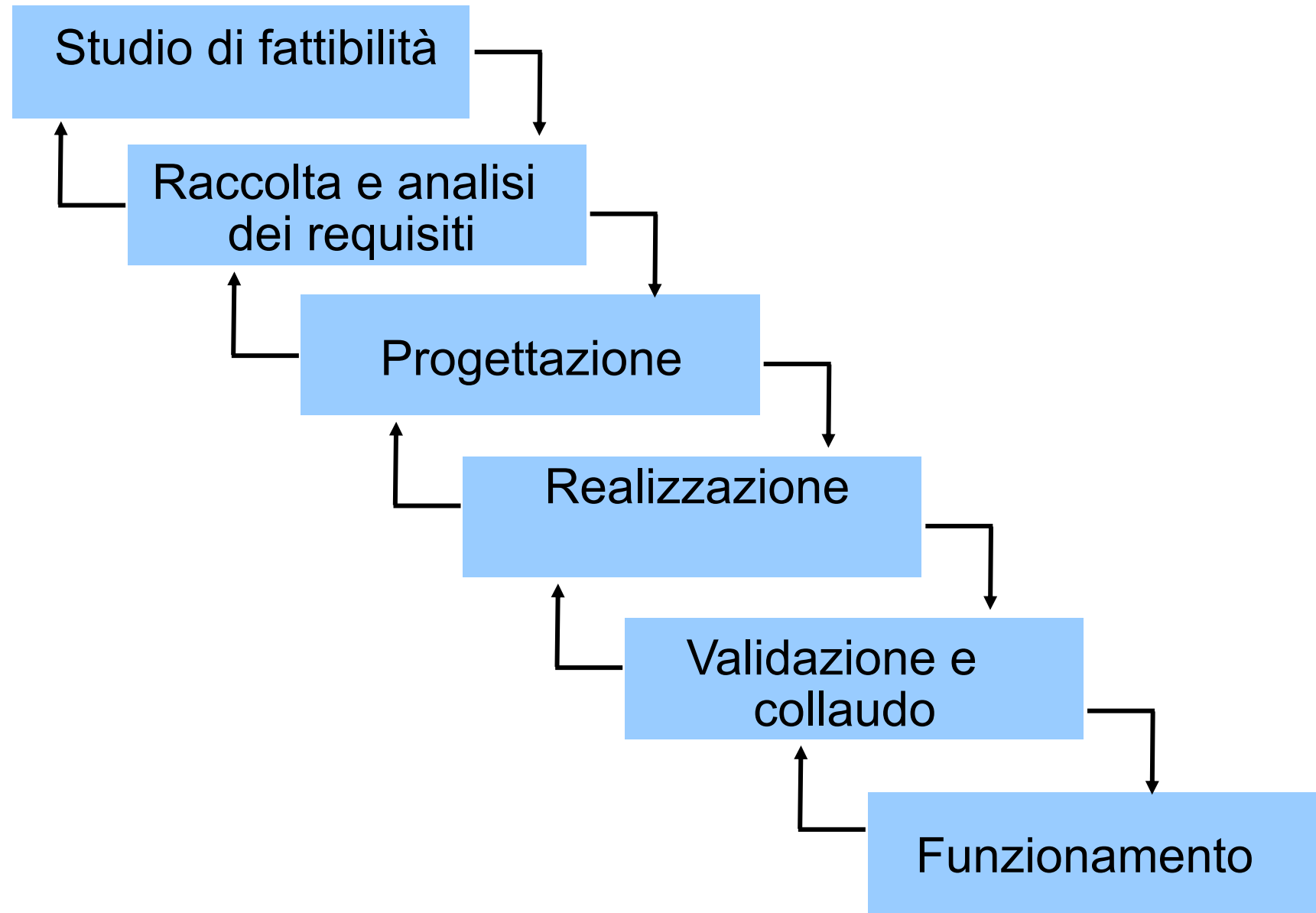
Laurea in Informatica, DeMaCS, UNICAL

26 Settembre 2025

Progettazione di basi di dati

- È una delle attività del processo di sviluppo dei sistemi informativi
- va quindi inquadrata in un contesto più generale:
- **il ciclo di vita dei sistemi informativi:**
 - Insieme e sequenzializzazione delle attività svolte da analisti, progettisti, utenti, nello sviluppo e nell'uso dei sistemi informativi
 - attività iterativa, quindi **ciclo**





Fasi (tecniche) del ciclo di vita



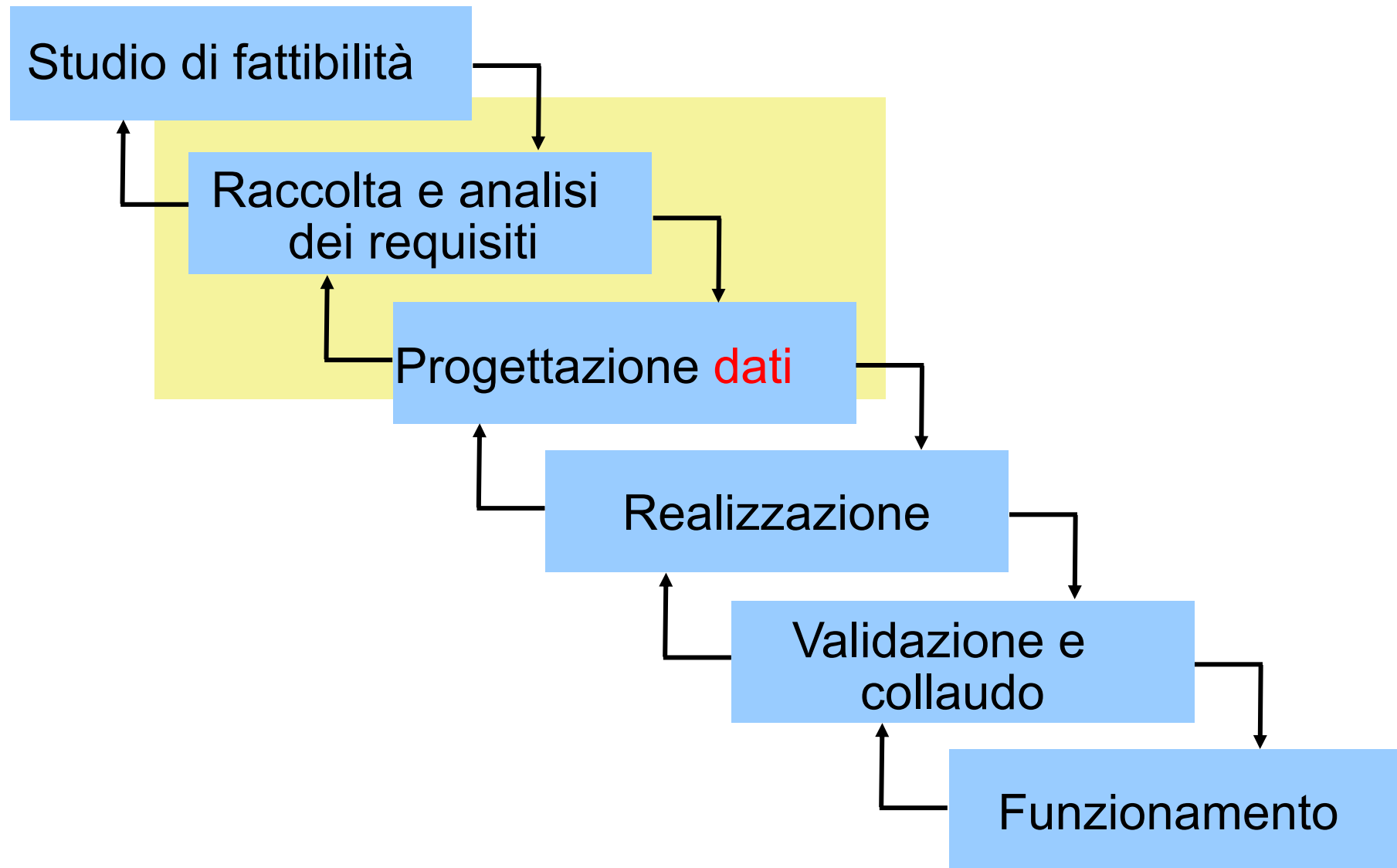
- Studio di fattibilità: definizione di costi e priorità, alternative
- Raccolta e analisi dei requisiti: studio delle proprietà del sistema, dei dati coinvolti e nelle loro relazioni
- Progettazione: di **dati** e funzioni
- Realizzazione: implementazione
- Validazione e collaudo: sperimentazione sul campo
- Funzionamento: il sistema diventa operativo

La progettazione di un sistema informativo riguarda due aspetti:

- ▶ **progettazione dei dati**
- progettazione delle applicazioni**

Ma:

- ▶ i dati hanno un ruolo centrale
 - i dati sono più stabili



Requisiti della base di dati

“CHE COSA”:

Progettazione
concettuale

Schema concettuale

Progettazione
logica

Schema logico

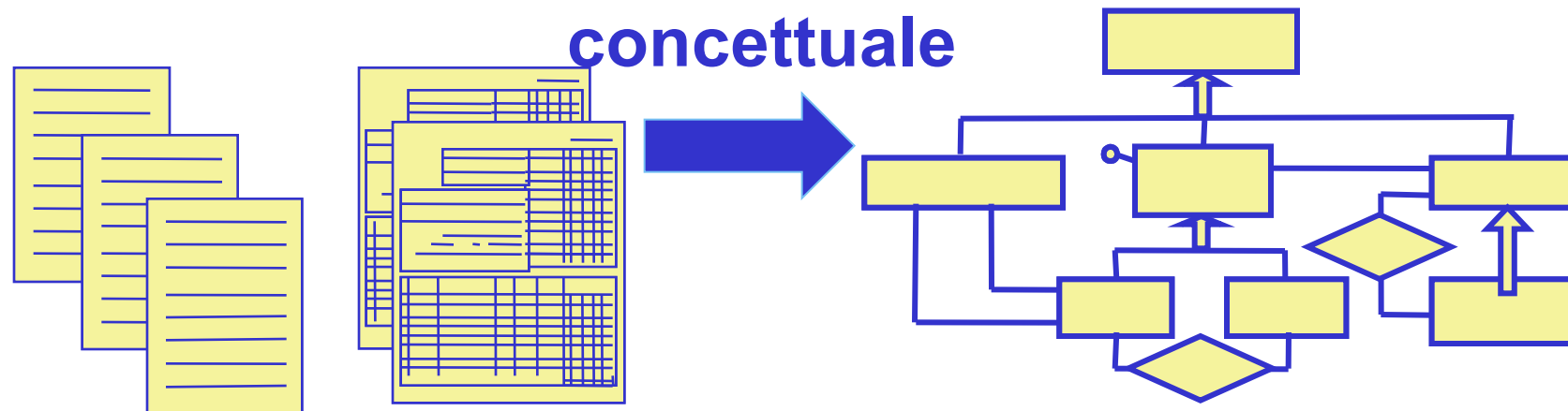
“COME”:

Progettazione
fisica

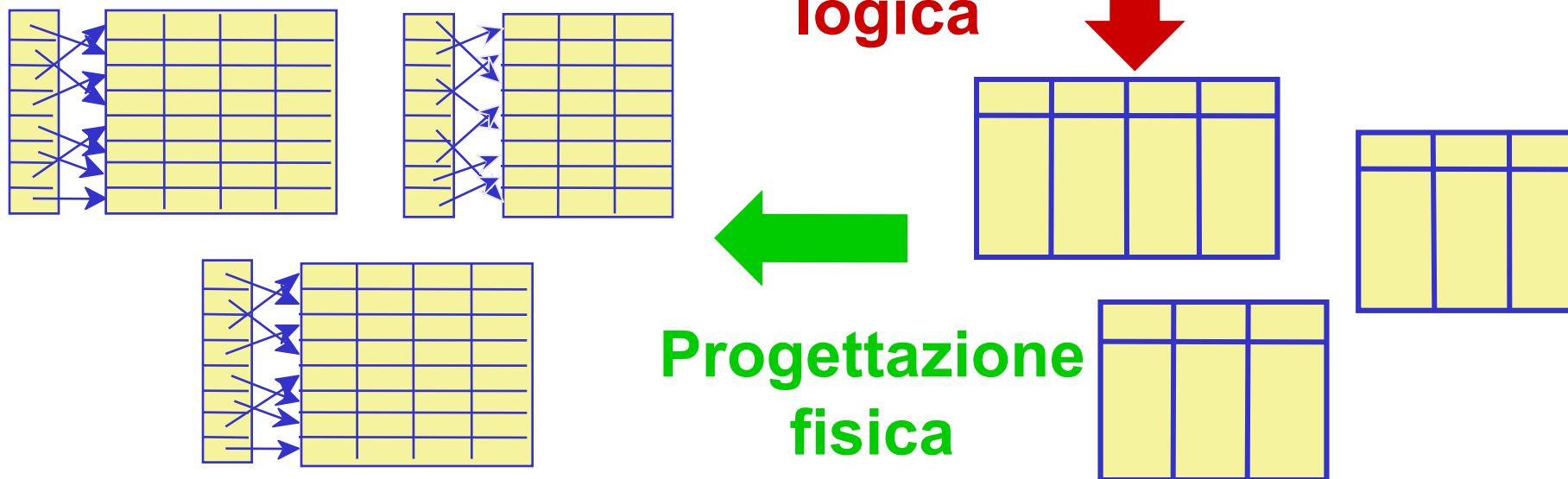
Schema fisico



Progettazione concettuale



Progettazione logica



Progettazione fisica

I prodotti della varie fasi sono
schemi di alcuni **modelli di dati**:

- Schema concettuale
- Schema logico
- Schema fisico



Modello dei dati

- insieme di costrutti utilizzati per organizzare i dati di interesse e descriverne la dinamica
- componente fondamentale: **meccanismi di strutturazione** (o **costruttori di tipo**)
- come nei linguaggi di programmazione esistono meccanismi che permettono di definire nuovi tipi, così ogni modello dei dati prevede alcuni costruttori
- ad esempio, il **modello relazionale** prevede il costruttore **relazione**, che permette di definire insiemi di record omogenei (tabelle)



Schemi e istanze

- In ogni base di dati esistono:
 - lo **schema**, sostanzialmente invariante nel tempo, che ne descrive la struttura (aspetto intensionale)
 - nel modello relazionale, le intestazioni delle tabelle
 - l'**istanza**, i valori attuali, che possono cambiare anche molto rapidamente (aspetto estensionale)
 - nel modello relazionale, il “corpo” di ciascuna tabella



Due tipi (principali) di modelli

- **modelli logici**: utilizzati nei DBMS esistenti per l'organizzazione dei dati
 - utilizzati dai programmi
 - indipendenti dalle strutture fisiche

esempi: **relazionale**, reticolare, gerarchico, a oggetti

- **modelli concettuali**: permettono di rappresentare i dati in modo indipendente da ogni sistema, e dal modello logico
 - cercano di descrivere i concetti del mondo reale
 - sono utilizzati nelle fasi preliminari di progettazione
 - non sono parte dei DBMS

il più noto è il modello **Entità-Relazione** (nel seguito verrà chiamato modello **Entity-Relationship** per non confondersi con la relazione del modello relazionale)



Modelli concettuali, perché?

- Proviamo a modellare una applicazione definendo direttamente lo schema logico della base di dati:
 - da dove cominciamo?
 - rischiamo di perderci subito nei dettagli
 - dobbiamo pensare subito a come correlare le varie tabelle (chiavi primarie etc.)
 - i modelli logici sono rigidi



Modelli concettuali, perché?

- servono per ragionare sulla realtà di interesse, indipendentemente dagli aspetti realizzativi
- permettono di rappresentare le classi di oggetti di interesse e le loro correlazioni
- prevedono efficaci rappresentazioni grafiche (utili anche per documentazione e comunicazione)



Modello Entity-Relationship (Entità-Relazione)

- Il più diffuso modello concettuale
 - Ne esistono molte versioni,
 - (più o meno) diverse l'una dall'altra



I costrutti del modello E-R

- Entità
- Relationship
- Attributo
- Identificatore
- Generalizzazione
-

Ogni costrutto ha una sua rappresentazione grafica.



Entità

- Classe di oggetti (fatti, persone, cose) della realtà di interesse con proprietà comuni e con esistenza “autonoma”
- Esempi:
 - impiegato, città, conto corrente, ordine, fattura



Entità ed occorrenza di entità

- **Entità:**
 - classe di oggetti, persone, ... "omogenei"
- **Occorrenza** (o istanza) di entità:
 - elemento della classe (l'oggetto, la persona, ..., non i dati)
- nello schema concettuale rappresentiamo le **entità**, non le singole istanze (rappresentazione ad alto livello)

Rappresentazione grafica di entità

Impiegato

Dipartimento

Città

Vendita



Entità, commenti

- Ogni entità ha un nome che la identifica univocamente nello schema:
 - nomi espressivi
 - opportune convenzioni
 - singolare

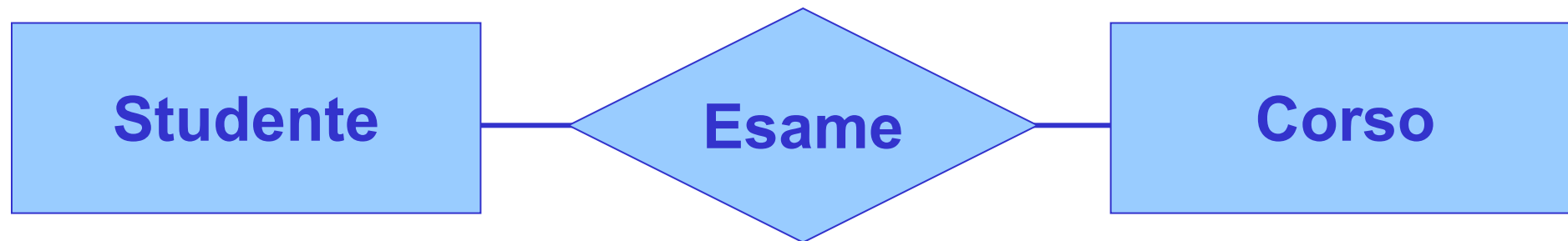


Relationship

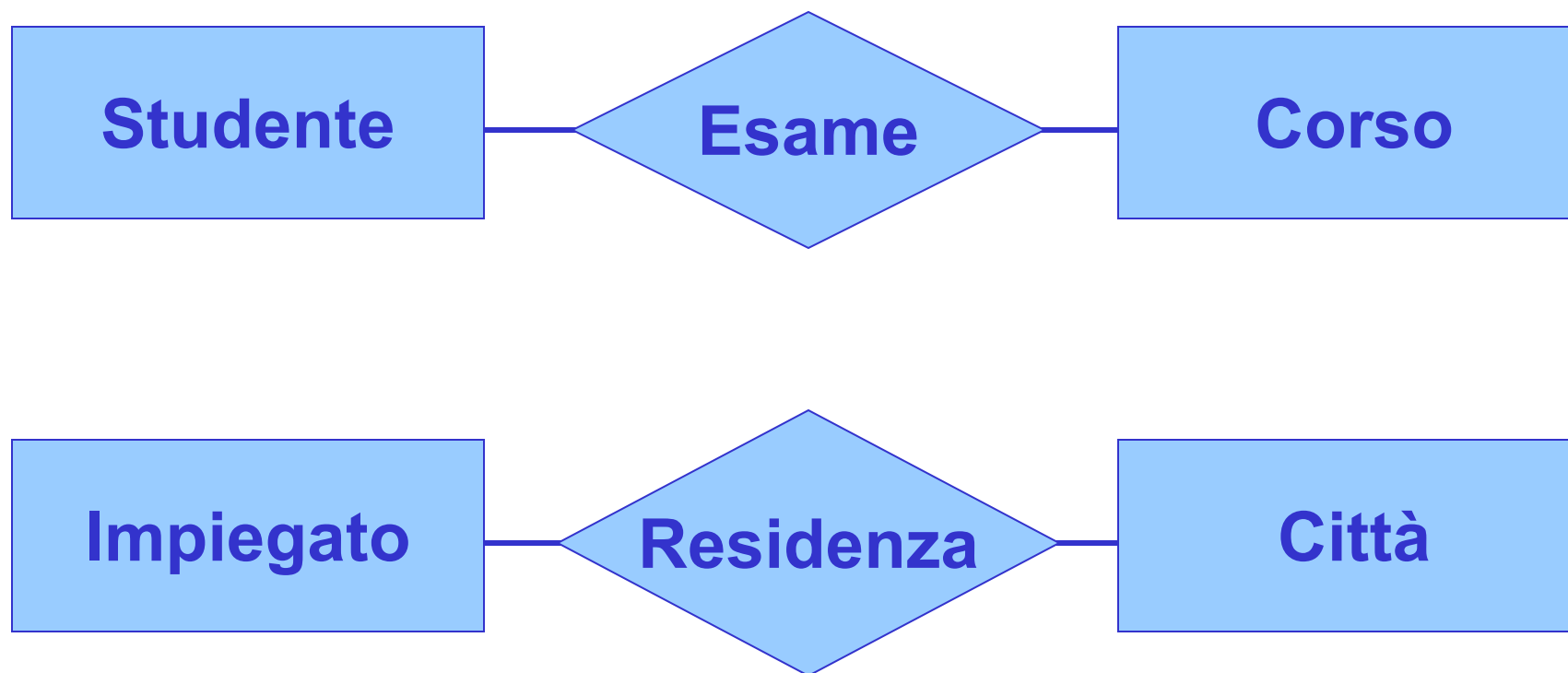
- Legame logico fra due o più entità, rilevante nell'applicazione di interesse
- Esempi:
 - Residenza (fra entità persona e città)
 - Esame (fra entità studente e corso)
- Chiamata anche:
 - **relazione**, correlazione, associazione



Uno schema E-R, graficamente



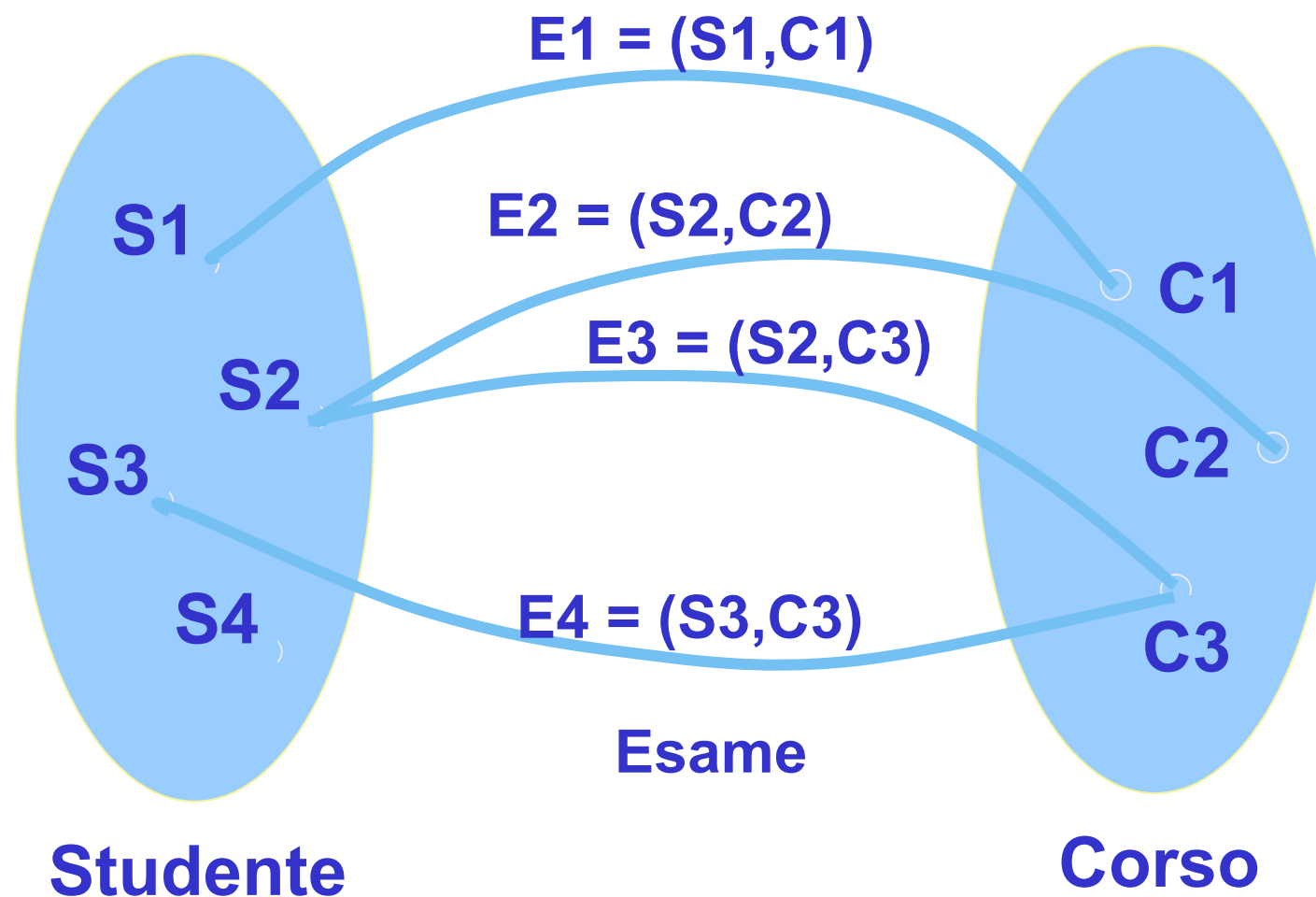
Rappresentazione grafica di relationship



Relationship, commenti

- Ogni relationship ha un nome che la identifica univocamente nello schema:
 - nomi espressivi
 - opportune convenzioni
 - singolare
 - sostantivi invece che verbi (se possibile), es: 'Residenza' invece che 'Risiede in'

Esempi di occorrenze

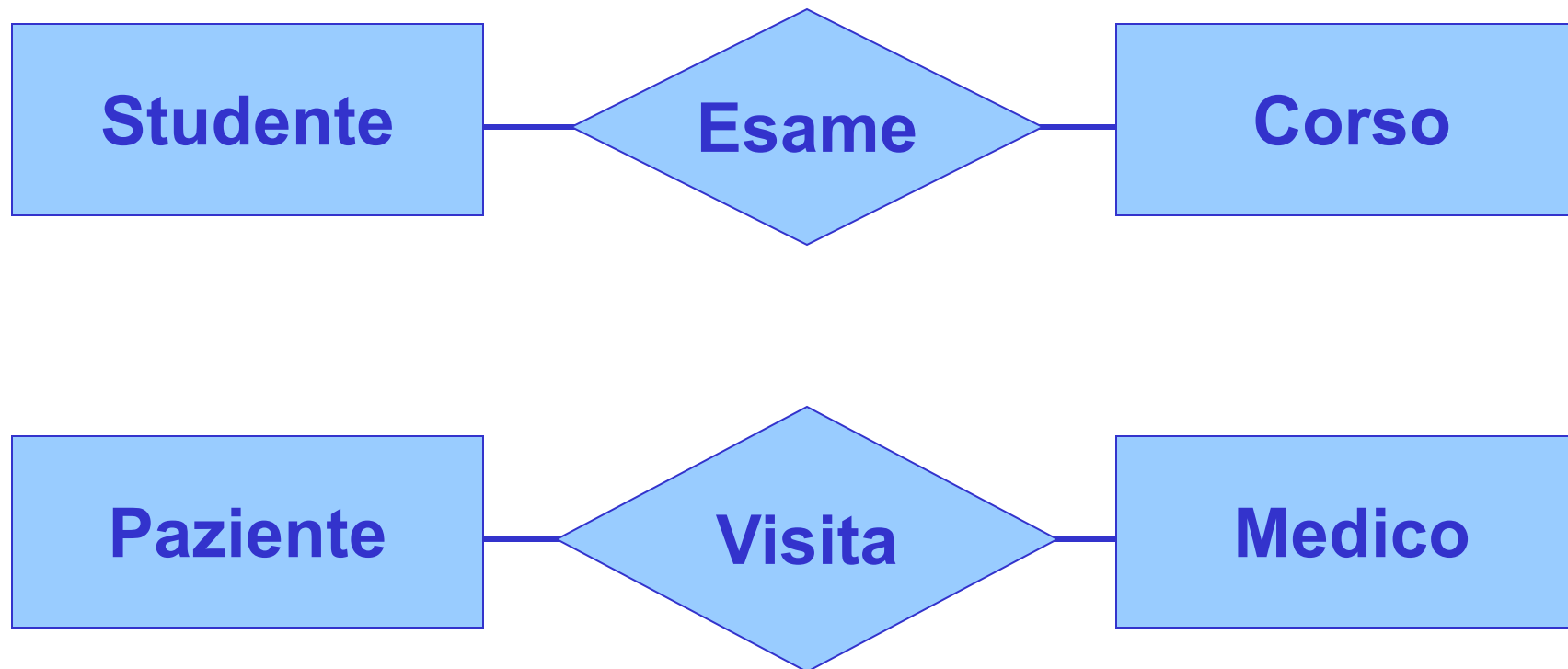


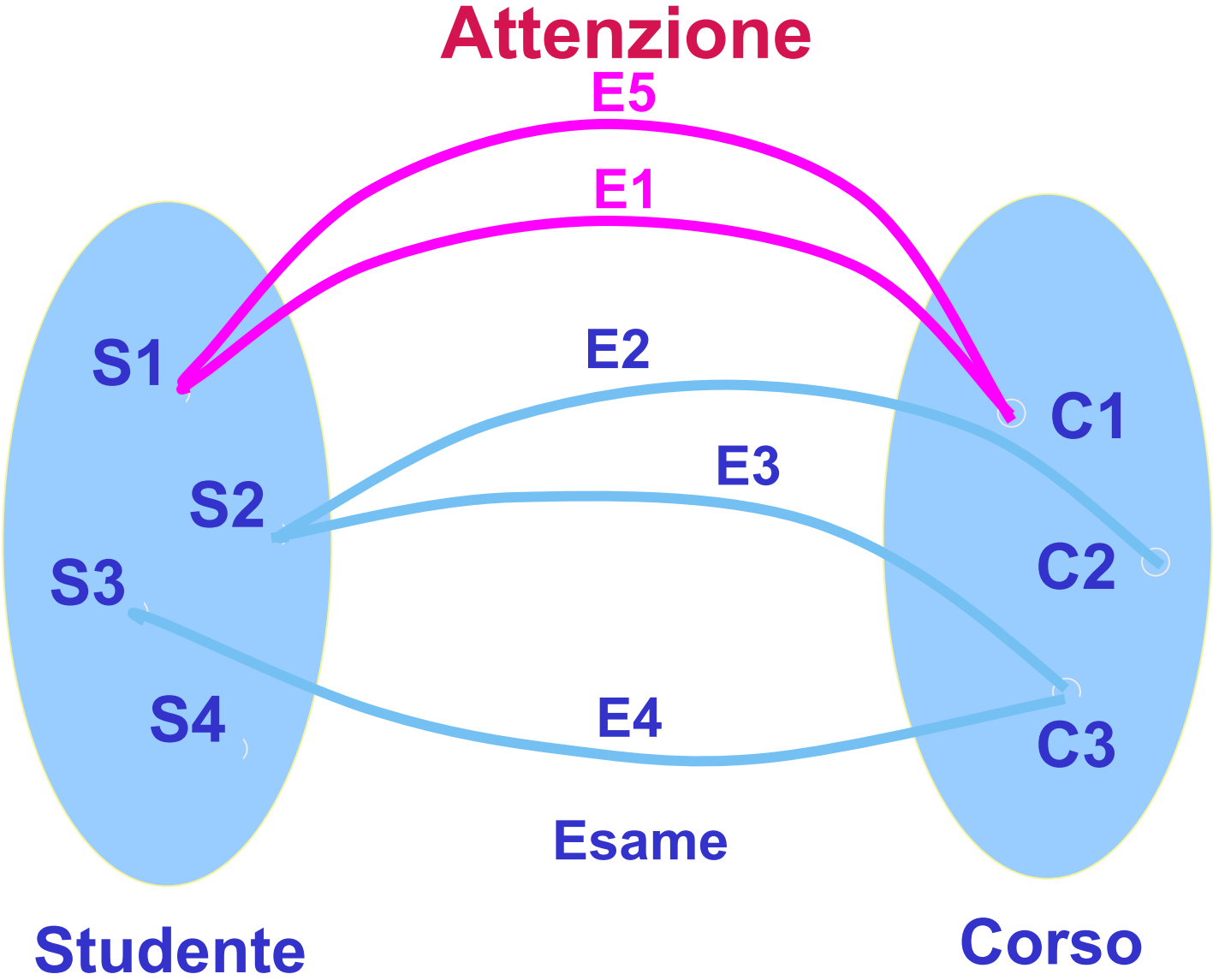
Relationship, occorrenze

- Una occorrenza di una relationship binaria è una coppia di occorrenze di entità, una per ciascuna entità coinvolta
- Una occorrenza di una relationship n-aria è una n-upla di occorrenze di entità, una per ciascuna entità coinvolta
- Le occorrenze di una relationship sono un insieme, quindi nell'ambito di una relationship non ci possono essere occorrenze (coppie, ennuple) ripetute

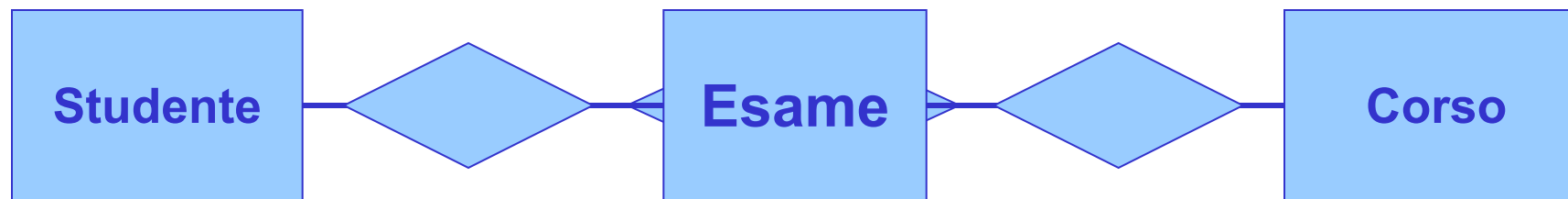


Relationship corrette? Dipende...



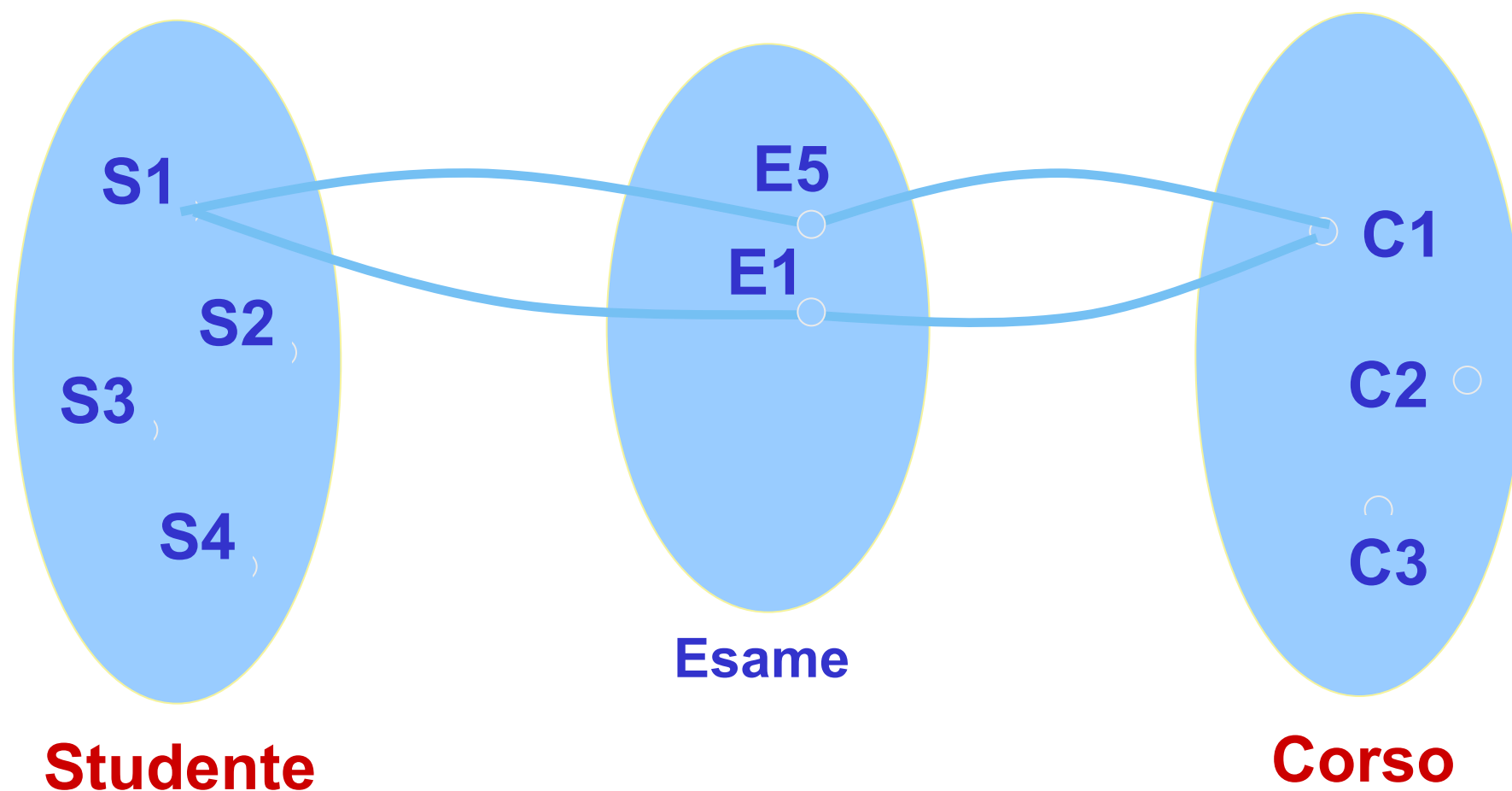


«Trasformiamo» la relationship

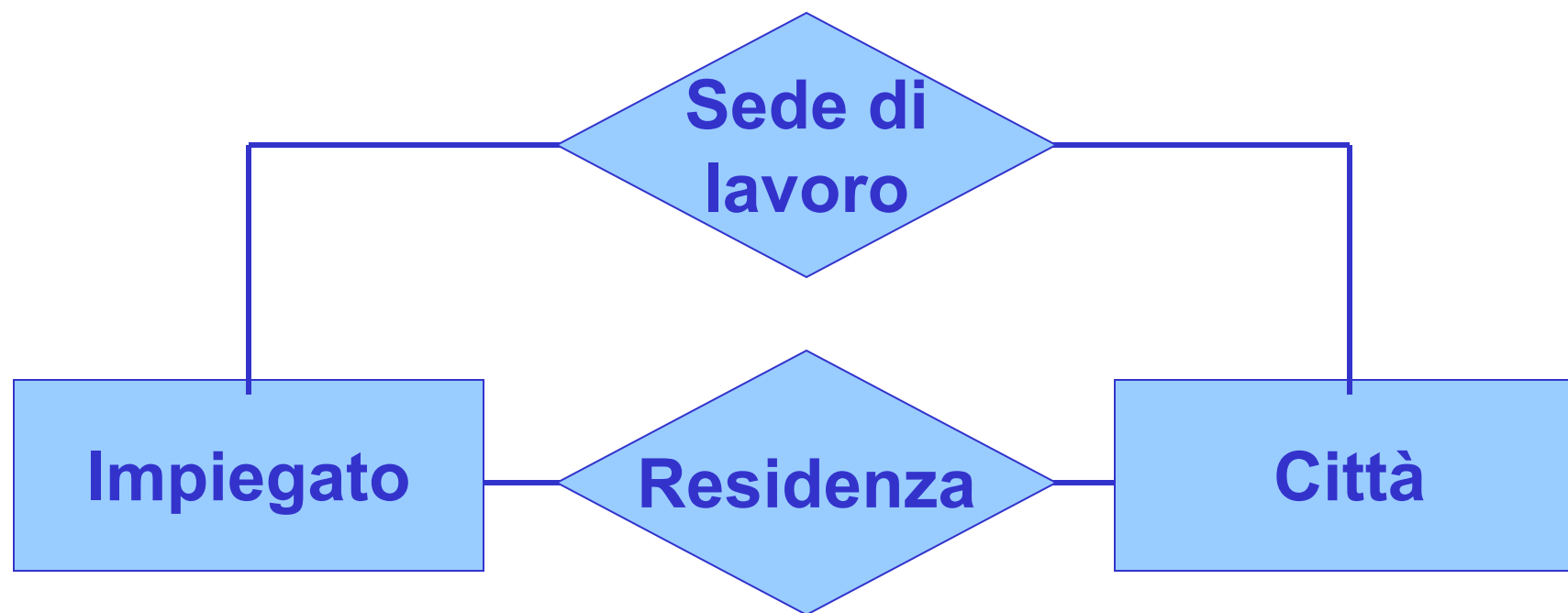


Le occorrenze di una relationship sono un insieme, quindi non possono essere ripetute. Se vogliamo esprimere il caso in cui uno studente ha sostenuto più volte lo stesso esame, dobbiamo passare a questa rappresentazione (o ad una relationship ternaria, che vedremo a breve).

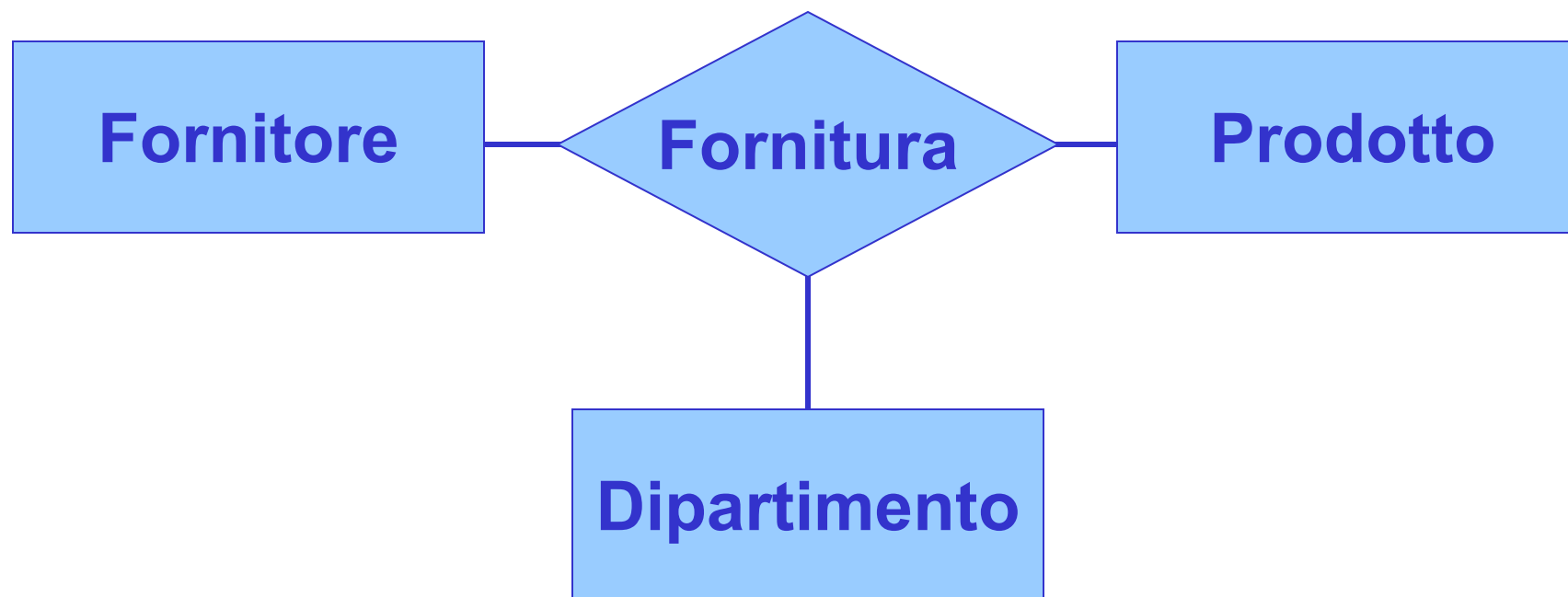
Con l'entità Esame



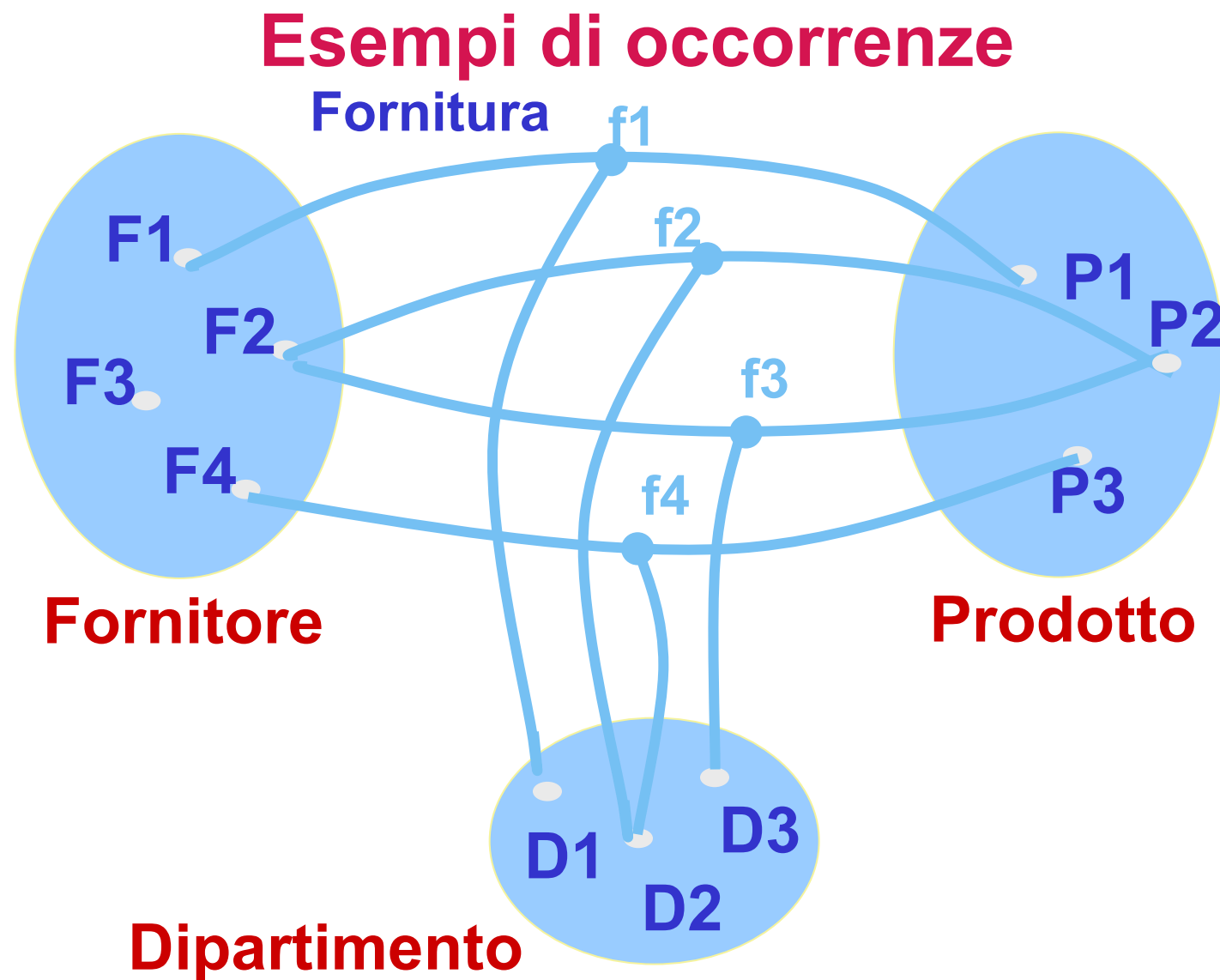
Due relationship sulle stesse entità



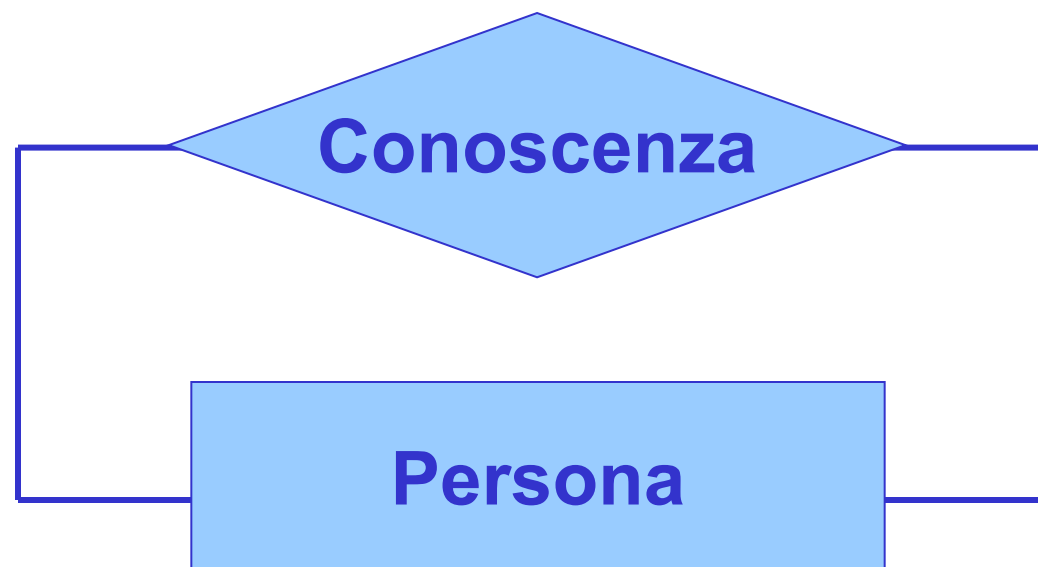
Relationship n-aria



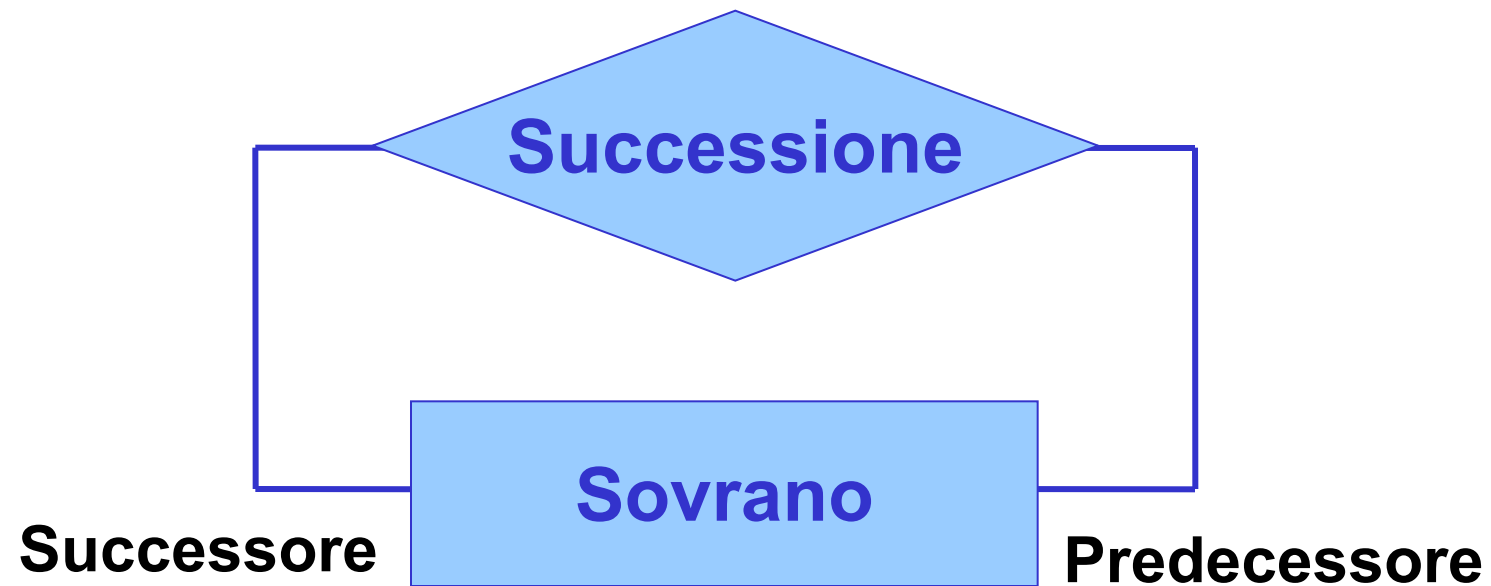
(Come anticipato, alternativa valida anche per il caso **Studente**, **Corso**, **Esame**)



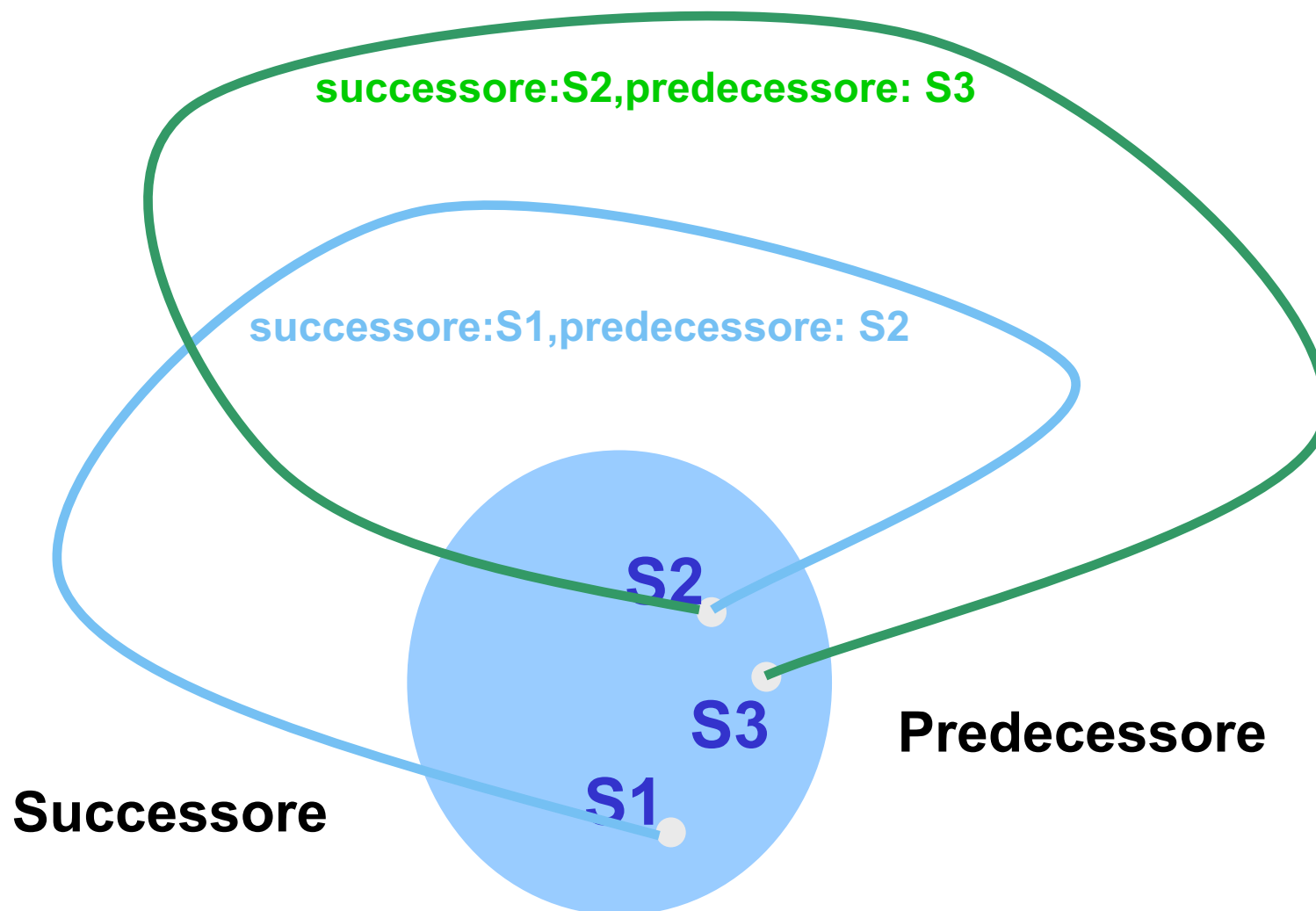
Relationship ricorsiva: coinvolge “due volte” la stessa entità



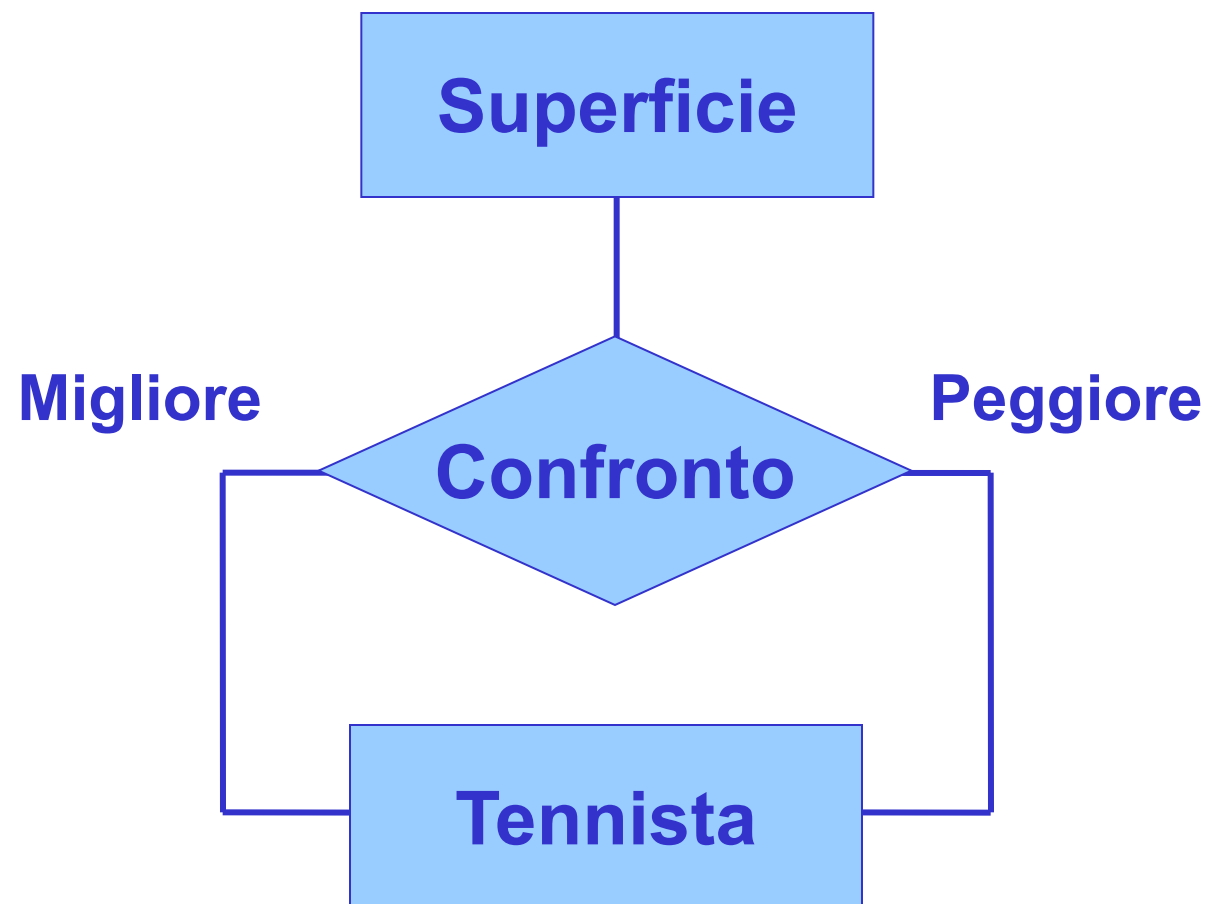
Relationship ricorsiva con “ruoli”



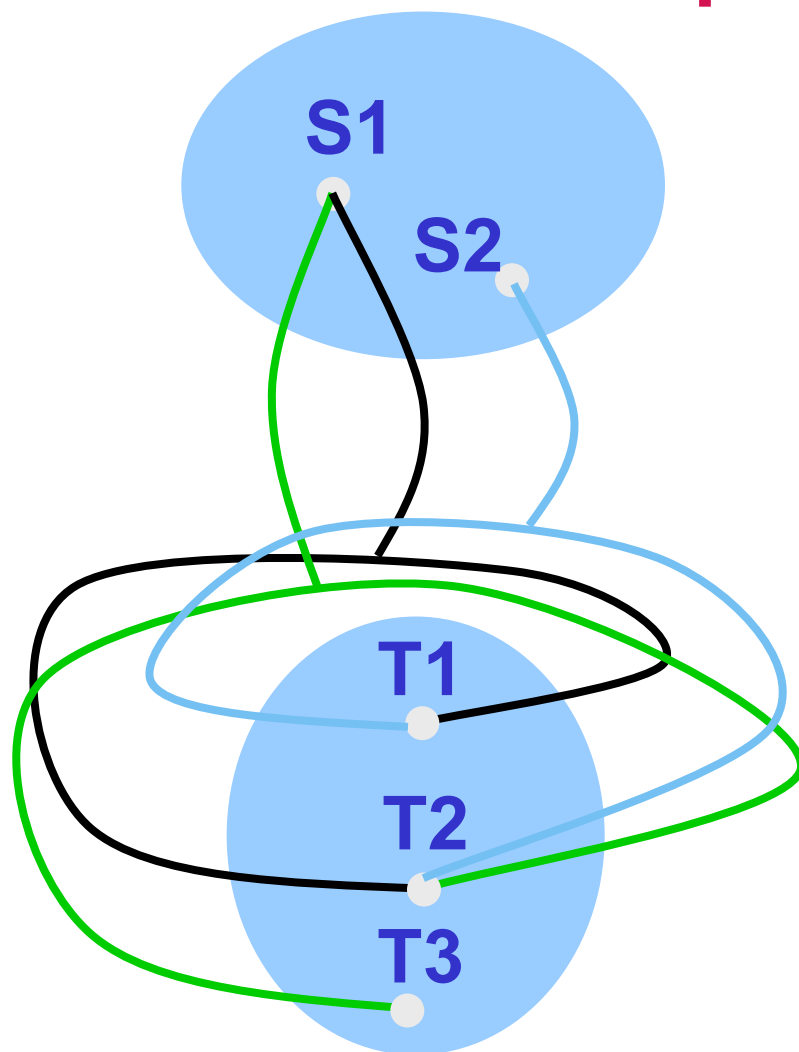
Esempi di occorrenze (c'e' un ordine: S3 -> S2 -> S1)



Relationship ternaria ricorsiva



Esempi di occorrenze



T1 è migliore di T2 su S2

T2 è migliore di T1 su S1

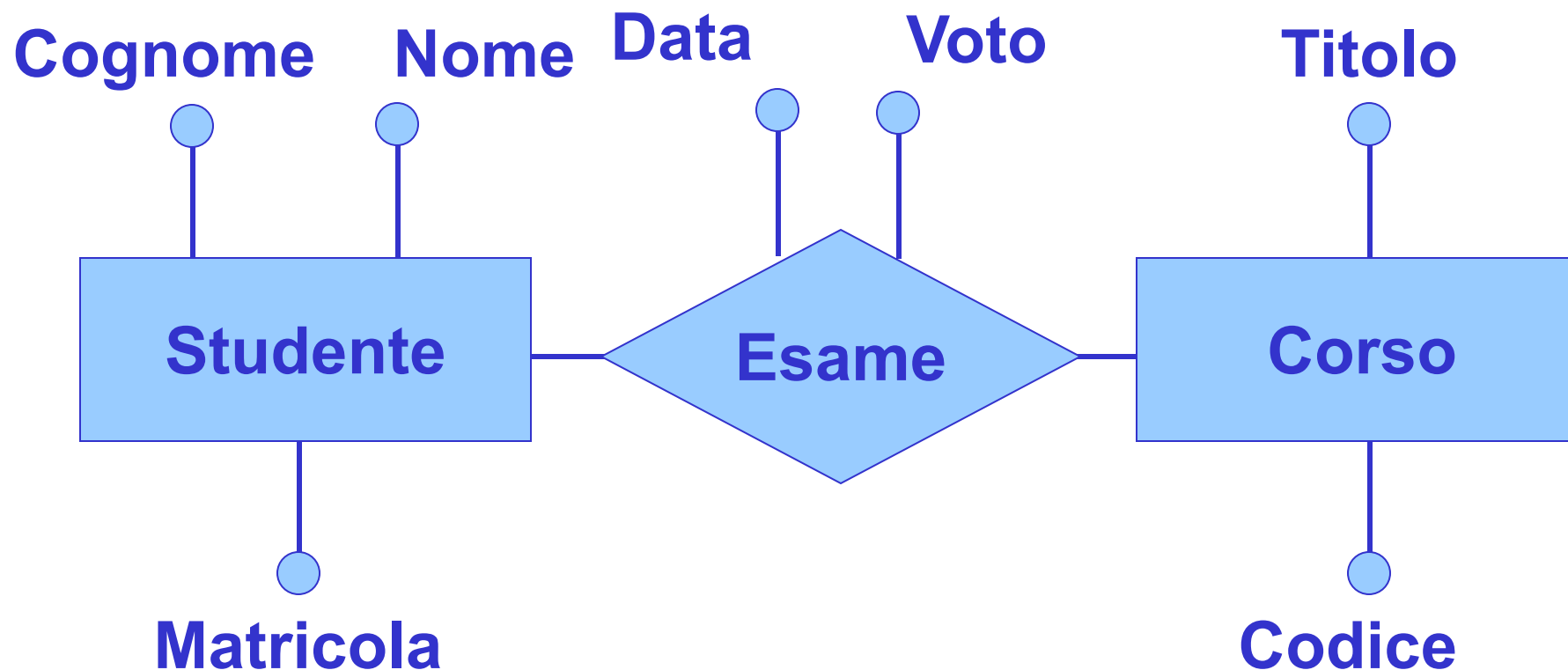
T3 è migliore di T2 su S1

Attributo

- Proprietà elementare di un'entità o di una relationship, di interesse ai fini dell'applicazione
- Associa ad ogni occorrenza di entità o relationship un valore appartenente a un insieme detto **dominio** dell'attributo



Attributi, rappresentazione grafica



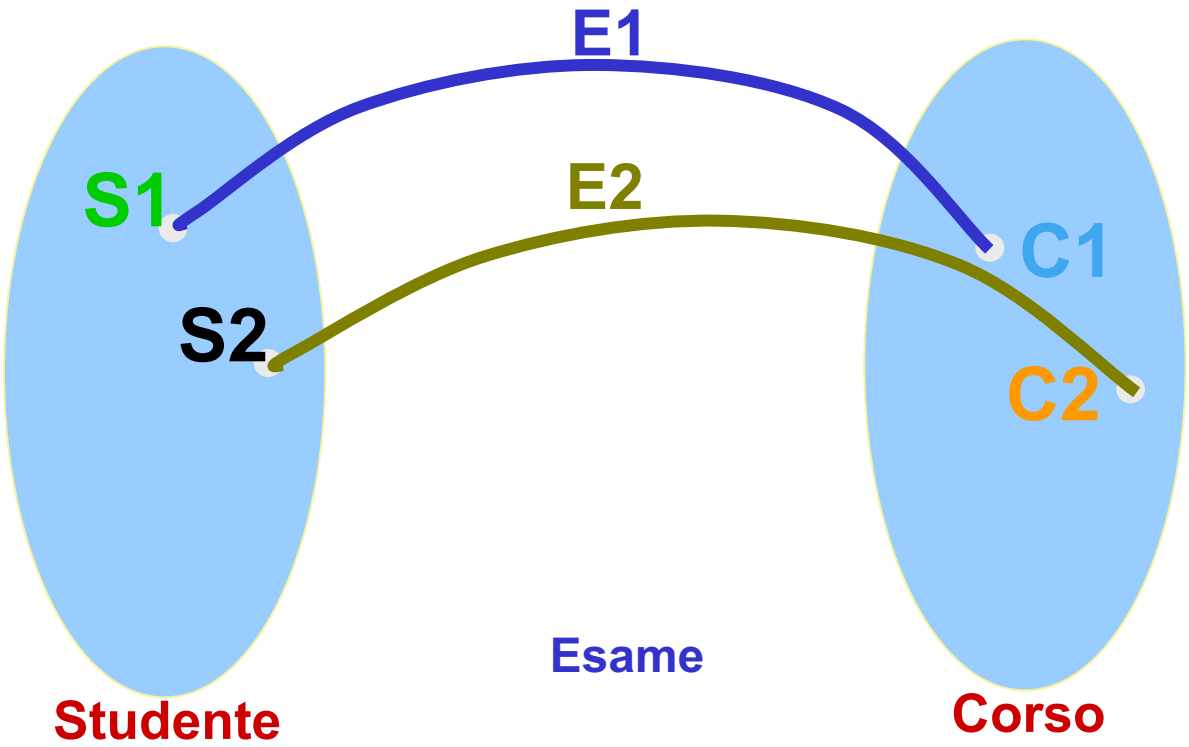
Esempi di occorrenze

Matricola: 34567
Cognome: Rossi
Nome: Mario

Data: 25/07/2004
Voto: 26

Codice: Inf205
Titolo: Basi di dati

Matricola: 46742
Cognome: Neri
Nome: Piero



Attributi composti

- Raggruppano attributi di una medesima entità o relationship che presentano affinità nel loro significato o uso
- Esempi:
 - Via, Numero civico e CAP formano un Indirizzo
 - Giorno, Mese ed Anno formano una Data



Rappresentazione grafica

