



MODELLO LOGICO

DEFINIZIONE

Il modello logico definisce la struttura degli elementi dei dati e le relazioni tra di essi.

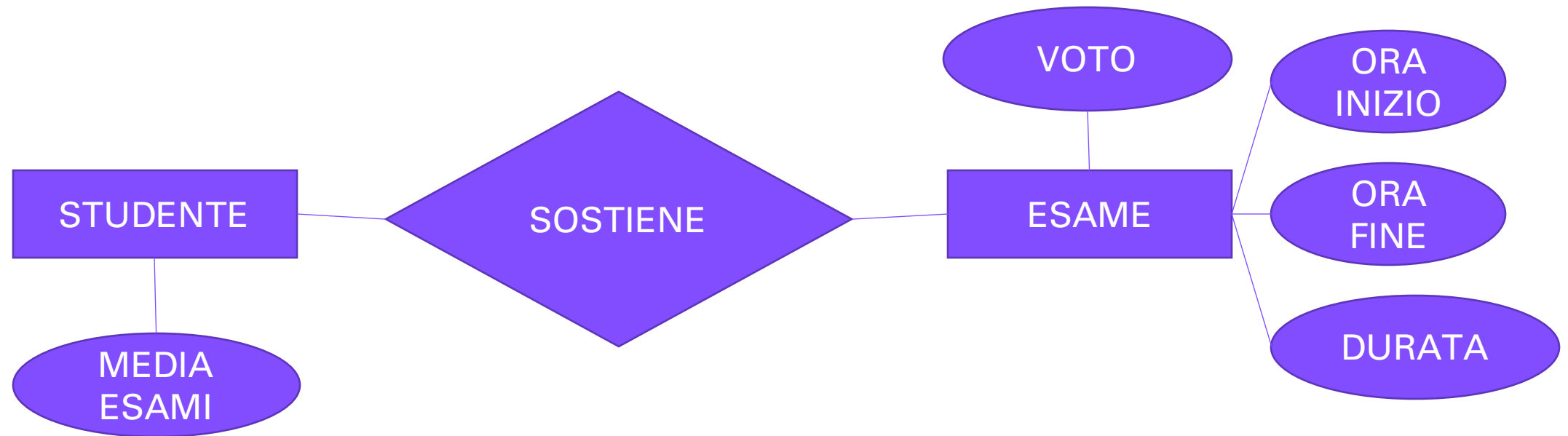
Anche in questo caso sono presenti attributi, entità e relazioni, ma il livello di dettaglio viene ulteriormente aumentato andando a specificare i tipi di dato che possono essere utilizzati e che verranno poi inseriti nelle tabelle da cui effettivamente è formato il database.

Come fare?

Si parte da un diagramma ER e si utilizzano una serie di tecniche che permettono di semplificare e trasformare una serie di entità e relazioni in tabelle, nelle quali si cerca principalmente di eliminare la ridondanza.

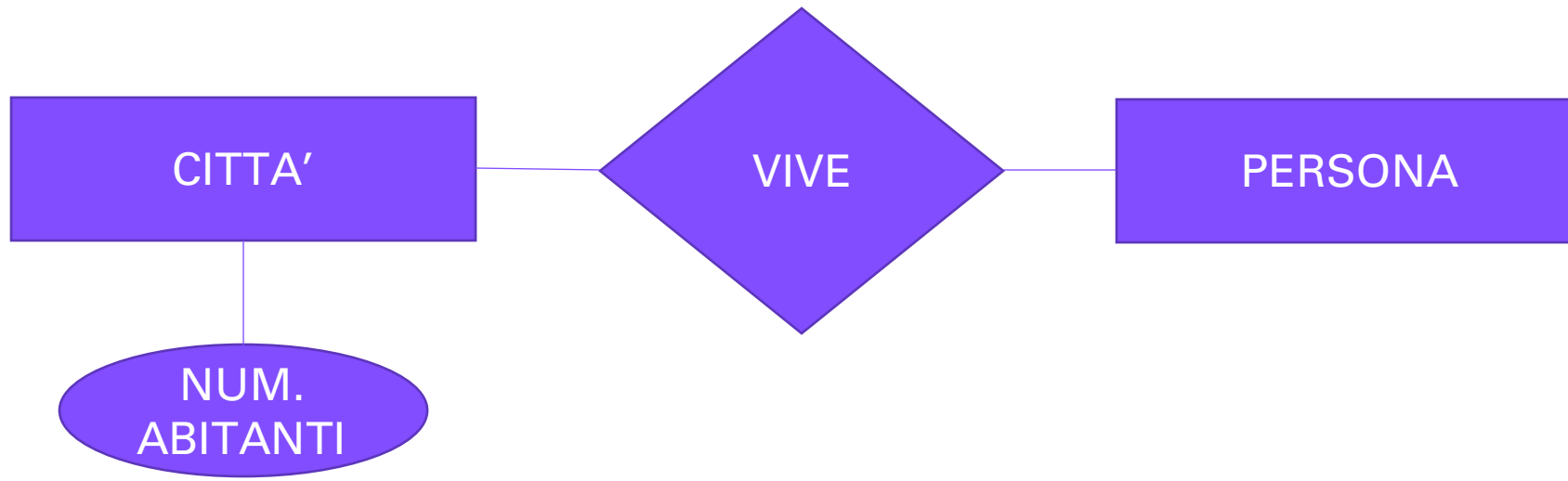
Ristrutturazione

Ristrutturazione del modello concettuale (eliminazione delle ridondanze o di attributi derivabili).



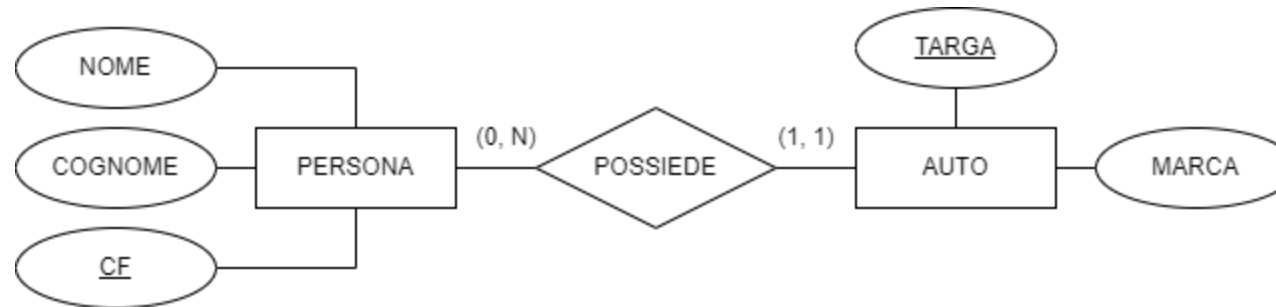
Ridondanza da conteggio

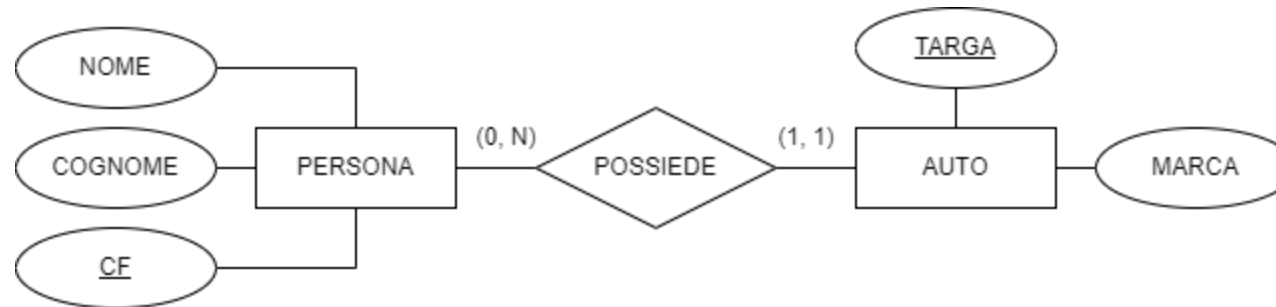
Il conteggio di occorrenze potrebbe essere un altro problema di ridondanza.



PRIMO PASSO

La traduzione del modello concettuale in modello logico si svolge trasformando ogni entità in una tabella, rappresentabile come una riga di testo che possiede il nome della tabella (solitamente con un nome plurale).





Persone (CF, Nome, Cognome)

Automobili (Targa, Marca, PersonaCF*)

La chiave primaria della tabella va sempre sottolineata!

E per le relazioni?

Quando siamo di fronte ad associazioni del tipo (1, N) o possiamo semplicemente includere in una tabella l'attributo che fa riferimento all'altra entità.

Automobili (Targa, Marca, Proprietario*)

Il termine proprietario viene indicato come una **chiave esterna** e conterrà il valore di chiave primaria di persona (solitamente si utilizza una freccia che punta alla chiave primaria di riferimento).

Esempio di tabella

Targa	Marca	Proprietario
A112ZX	Fiat	AGGFF...
387HHZ	Mercedes	KLSSHD...

E il contrario?

Se avessimo scelto la targa come chiave esterna di persona?

CF	Nome	Cognome	Targa
XYZ	Gino	Paolazzi	AB1234
XYZ	Gino	Paolazzi	ZZ5556
ABC	Mario	Rossi	555564

Notate il problema? In questo caso la soluzione non è ottima, perché rispetto a quella precedente aggiunge molta ridondanza delle informazioni.

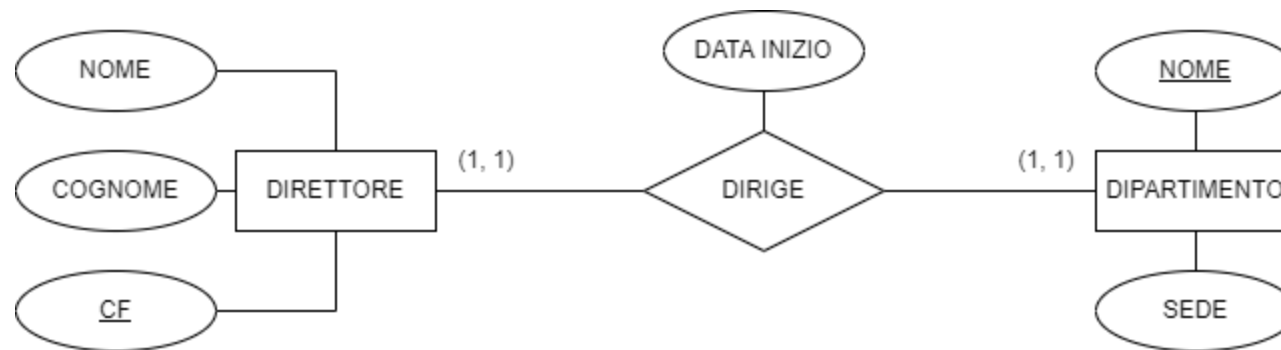
Regola cardinalità 1:N

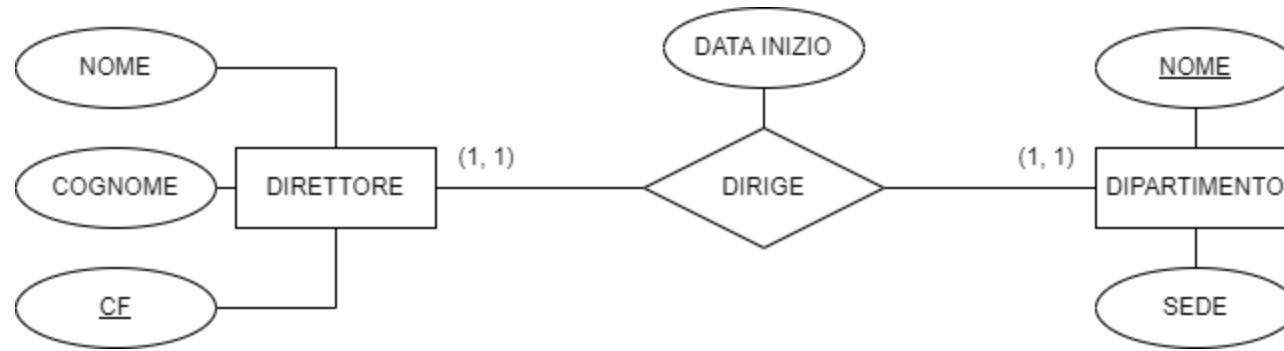
La chiave esterna viene aggiunta laddove la massima partecipazione dell'entità è pari a 1.

Mai dal lato della relazione dove la cardinalità massima è N.

Regola cardinalità 1:1

Con la cardinalità 1:1 è possibile raggruppare le informazioni all'interno di una singola tabella, oppure mantenere le informazioni in tabelle separate utilizzando come chiave esterna una qualsiasi delle primarie delle relazioni coinvolte.





Dirige (CF, Nome, Cognome, Data inizio, Nome_dip, sede)

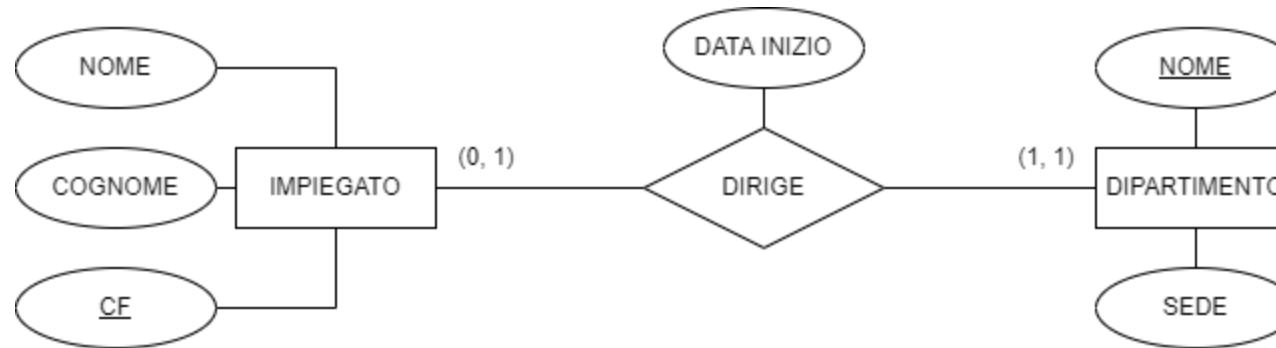
Oppure

Direttori (CF, Nome, Cognome)

Dipartimenti (Nome, Sede, Direttore*, Data inizio)

Partecipazione facoltativa

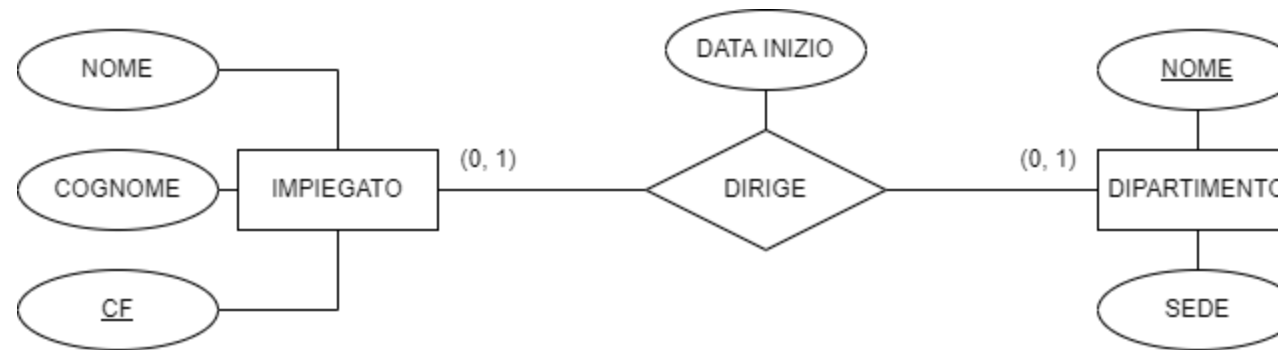
Nel caso in cui la partecipazione alla relazione sia facoltativa:



Impiegati (CF, nome, cognome)
Dipartimenti (Nome, sede, direttore*)

Partecipazione totalmente opzionale

Nel caso di doppia opzionalità:



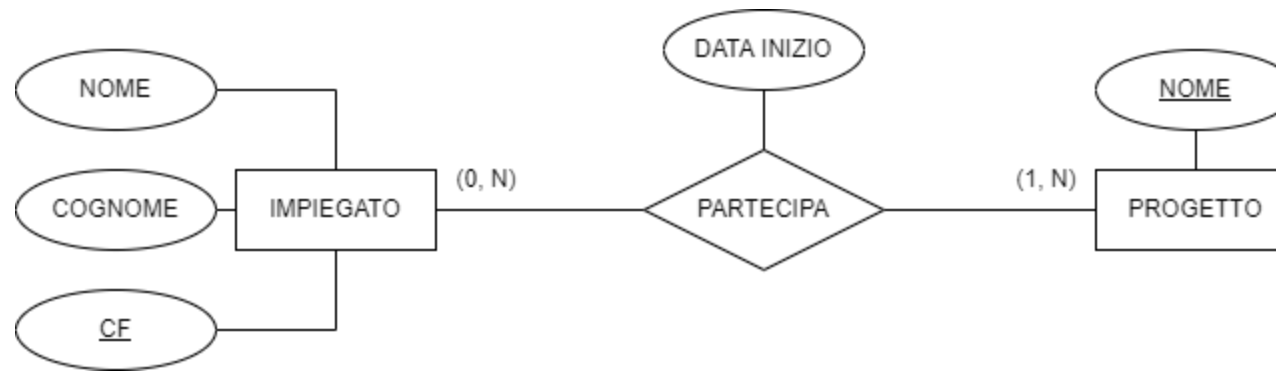
Impiegati (CF, nome, cognome)

Dipartimenti (Nome, sede)

Dirige (DIP*, Direttore*, DataInizio)

Relazioni N:N

Nel caso di relazioni che sono di tipo molti a molti:



Impiegati (CF, nome, cognome)

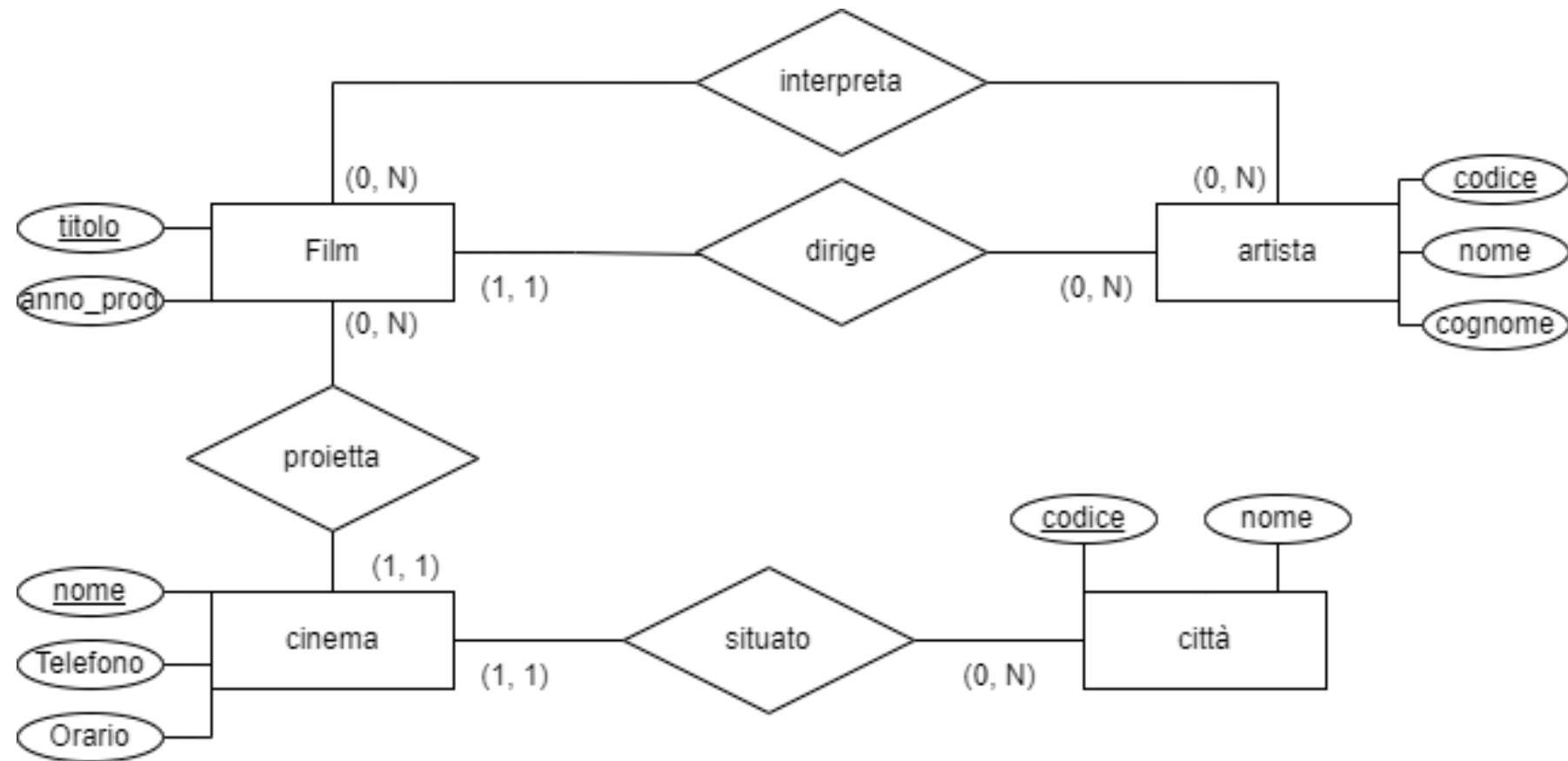
Progetti (Nome)

Partecipa (Impiegato*, progetto*, data_inizio)

Relazioni N:N

Si sceglie di creare una relazione aggiuntiva che contiene le chiavi esterne relative alle entità che prendono parte alla relazione e aggiungono gli attributi propri della relazione nello schema ER.

Esercizio 1



Soluzione

Artisti (Codice, Nome, Cognome)

Film (Titolo, Anno, CasaProduttrice, Direttore*)

Interpretazioni (Film*, Artista*)

Citta (Codice, Nome)

Cinema (Nome, Telefono, Orario, Indirizzo, Citta*, Film*)

E con le generalizzazioni?

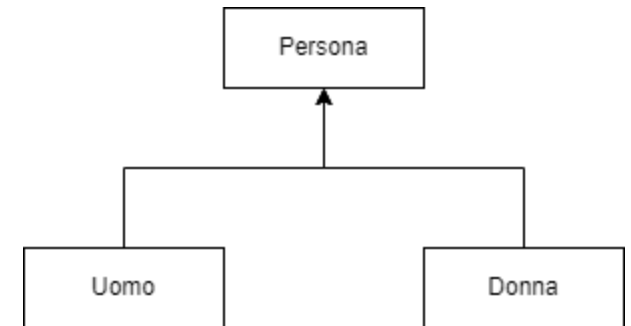
Dobbiamo eliminare le generalizzazioni effettuando una ricostruzione dello schema dove vengono inserite delle relazioni che specificano quali sono le entità collegate.

Esempio1

Nel caso di assenza di attributi una possibilità è quella di inserire un flag che identifica la divisione tra un'entità generalizzata e un'altra.

Attenzione però, il flag gestisce solitamente un numero limitato di generalizzazioni (ad esempio un valore binario per indicare uno o l'altro)

Persone (ID, Nome, Cognome, Sesso)



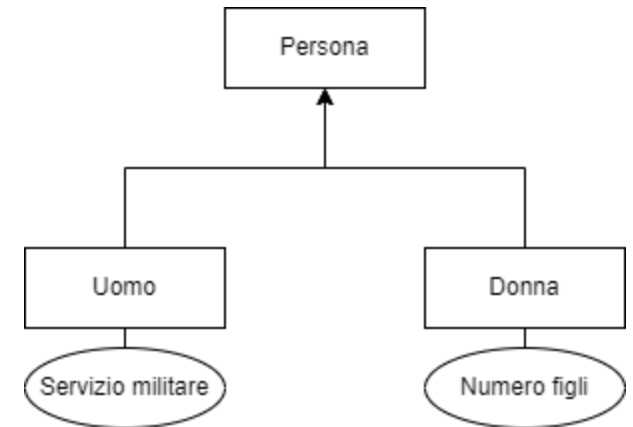
Esempio 2

Se invece le generalizzazioni possiedono degli attributi la ristrutturazione si deve effettuare aggiungendo una serie di relazioni.

Uomini (PersonalID*, Servizio_militare)

Donne (PersonalID*, Numero_figli)

Persone (ID, Nome, Cognome)



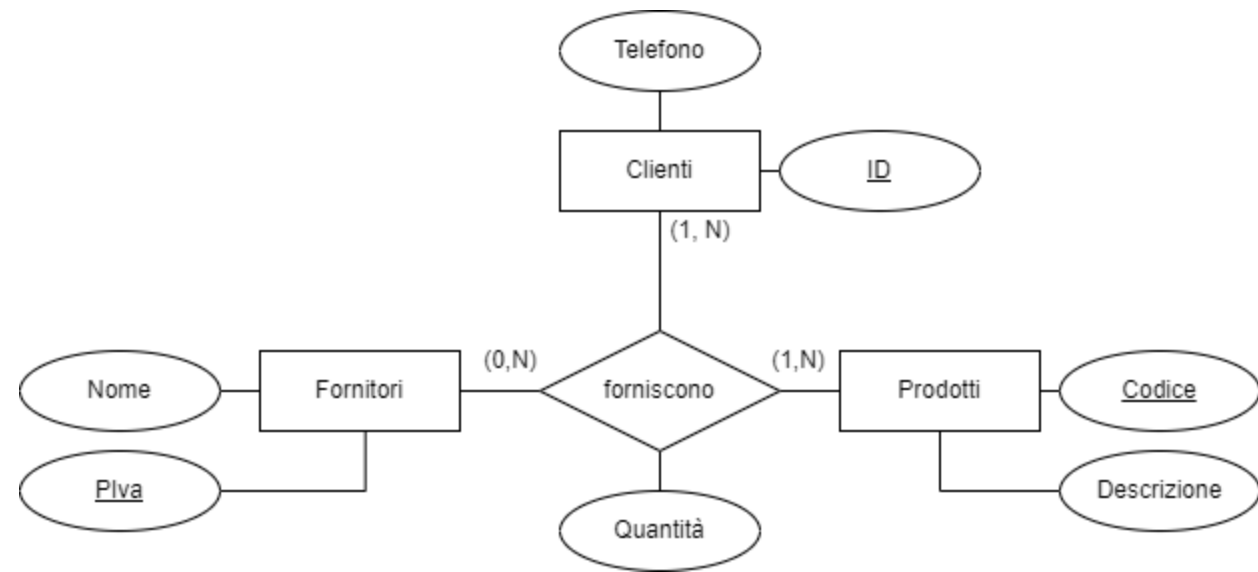
Relazioni N-Arie

Nel caso di una relazione che coinvolge più entità:

Clienti(ID, Telefono)

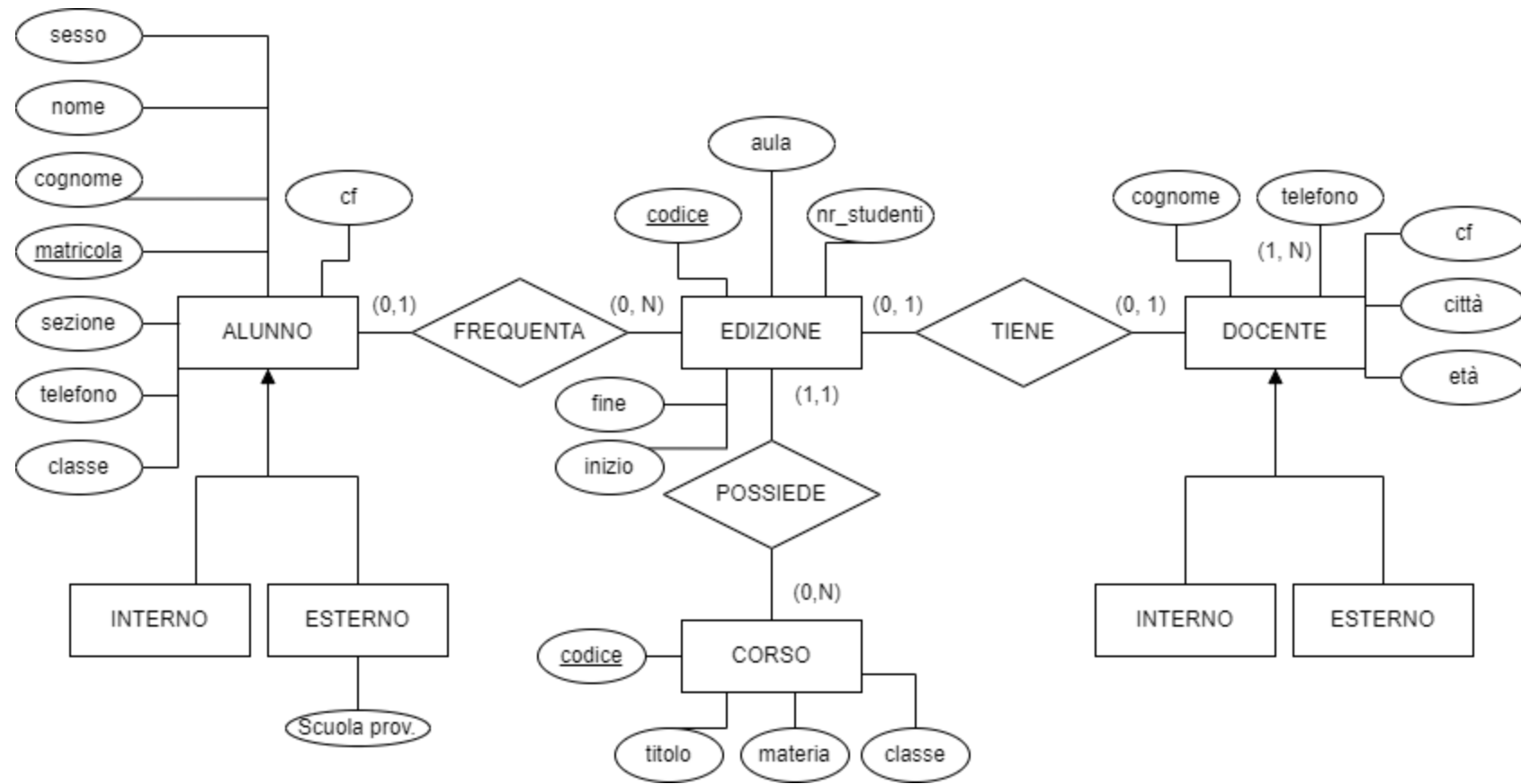
Fornitori (Plva, Nome)

Prodotti (Codice, Descrizione)



Forniture (Fornitore*, Cliente*, Prodotto*, Quantità)

Esercizio 2



Soluzione

Alunni (matricola, sesso, nome, cognome, sezione, telefono, classe, cf)

AlunniInterni (ID_alunno*)

AlunniEsterni (ID_alunno*, scuola_provenienza)

Edizioni (codice, aula, nr_studenti, fine, inizio, ID_corso*)

Corsi (codice, titolo, materia, classe)

Docenti (cf, cognome, città, età, interno)

Telefoni (ID_docente*, numero_telefono)

Frequenta (ID_alunno*, ID_edizione*)

DocenteEdizione (ID_edizione*, ID_docente*)

Le forme normali

Le 3 forme normali sono delle regole che assicurano che il nostro database rispecchi i principi di buona progettazione.

Si tratta di una specie di «cartina tornasole» che determina la qualità di uno schema relativo ad una base di dati relazionale.

In sostanza si tratta di buone pratiche da seguire che non sono obbligatorie per la realizzazione di un database, ma fortemente consigliate.

La prima forma normale (1FN)

La prima forma normale prevede i seguenti prerequisiti relativi alle tabelle:

- Deve esistere una chiave primaria che identifica univocamente un record
- Gli attributi devono essere atomici (non ulteriormente scomponibili)

Esempio

La seguente tabella rispetta la 1FN?

Persone (CF, nome, cognome, indirizzo)

Non lo è perché posso scomporre l'indirizzo in molti altri dati da memorizzare.

Persone (CF, nome, cognome, nazione, città, via, numero)

E per gli attributi con cardinalità?

Persone (CF, nome, cognome, Num_telefono)

In questo caso non posso inserire più di un numero di telefono, la possibile soluzione è:

Persone (CF, nome, cognome)

NumeriTelefono (CF, Num_telefono)

La seconda forma normale (2FN)

Il prerequisito è che la tabella sia in 1FN. Successivamente si applicano i seguenti requisiti:

- Non ci devono essere attributi che **dipendono funzionalmente** da una parte della chiave (solo con chiave multipla)

Mi devo assicurare che gli attributi non chiave dipendono solo da una parte della chiave.

Esempio

Ordini (CodOrdine, CodCliente, CodProd, data, quantità, descrizione)

Ad esempio in questo caso la descrizione del prodotto dipende dal codice che fa parte della chiave, quindi è una dipendenza funzionale e questo implica che la tabella non è in 2FN. Bisogna quindi cercare di interrompere questi legami di dipendenza funzionale.

Ordini (CodOrdine, data, CodCliente)

Prodotti (CodProd, descrizione)

ProdottiOrdinati (CodOrdine, CodProdotto, quantità)

Terza forma normale (3FN)

Una base dati è in terza forma normale se è in 2FN e per ogni dipendenza funzionale è vera una delle seguenti condizioni:

- X è una superchiave della relazione
- Y è membro di una chiave della relazione

Questo significa che tutti gli attributi non chiave dipendono esclusivamente dalla chiave e quindi non esistono attributi che dipendono da altri attributi non chiave.

Esempio

N. Ordine	Data	Cod cliente	Nome cliente	Città cliente
82001	18/09/2022	1111	Mario	Arco
82002	19/09/2022	1211	Giacomo	Riva del Garda

In questo caso abbiamo un problema perché il nome e la città del cliente dipendono dal codice cliente che chiaramente non è una chiave primaria della tabella. Posso facilmente risolvere isolando i clienti e lasciando nella tabella solo il codice cliente e ponendo in una tabella separata tutti i dettagli dei clienti.