

**Laurea in Informatica
A.A. 2024-2025**

Corso "Base di Dati"

Progettazione Logica

Prof. Massimiliano de Leoni



**UNIVERSITÀ
DEGLI STUDI
DI PADOVA**

Requisiti della base di dati

Progettazione
concettuale

“CHE COSA”:
analisi

Schema concettuale

Progettazione
logica

Schema logico

“COME”:
progettazione

Progettazione
fisica

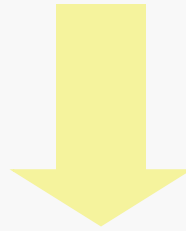
Schema fisico

Progettazione logica

- "Tradurre" lo schema concettuale in uno schema logico che rappresenti gli stessi dati in maniera corretta ed efficiente
- Osservazioni:
 - Alcuni aspetti non sono direttamente rappresentabili
 - È spesso necessario considerare le prestazioni

**Carico
applicativo**

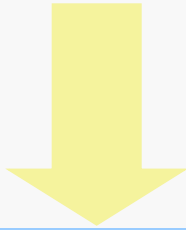
**Schema concettuale
E-R**



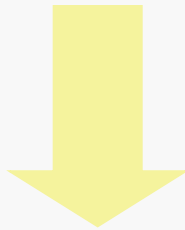
**Ristrutturazione dello
schema E-R**

**Modello
logico**

**Schema E-R
ristrutturato**



**Traduzione nel
modello logico**



**Schema
logico**

Ristrutturazione schema E-R

- Motivazioni:
 - Semplificare la traduzione
 - "Ottimizzare" le prestazioni
- Osservazione:
 - Uno schema E-R ristrutturato non è (più) uno schema concettuale nel senso stretto del termine

Attività della ristrutturazione

- Analisi delle ridondanze
- Eliminazione delle generalizzazioni
- Partizionamento/accorpamento di entità e relationship
- Scelta degli identificatori primari

Analisi delle ridondanze

- Una ridondanza in uno schema E-R è una informazione significativa ma derivabile da altre
- In questa fase si decide se eliminare le ridondanze eventualmente presenti o mantenerle (o anche di introdurne di nuove)

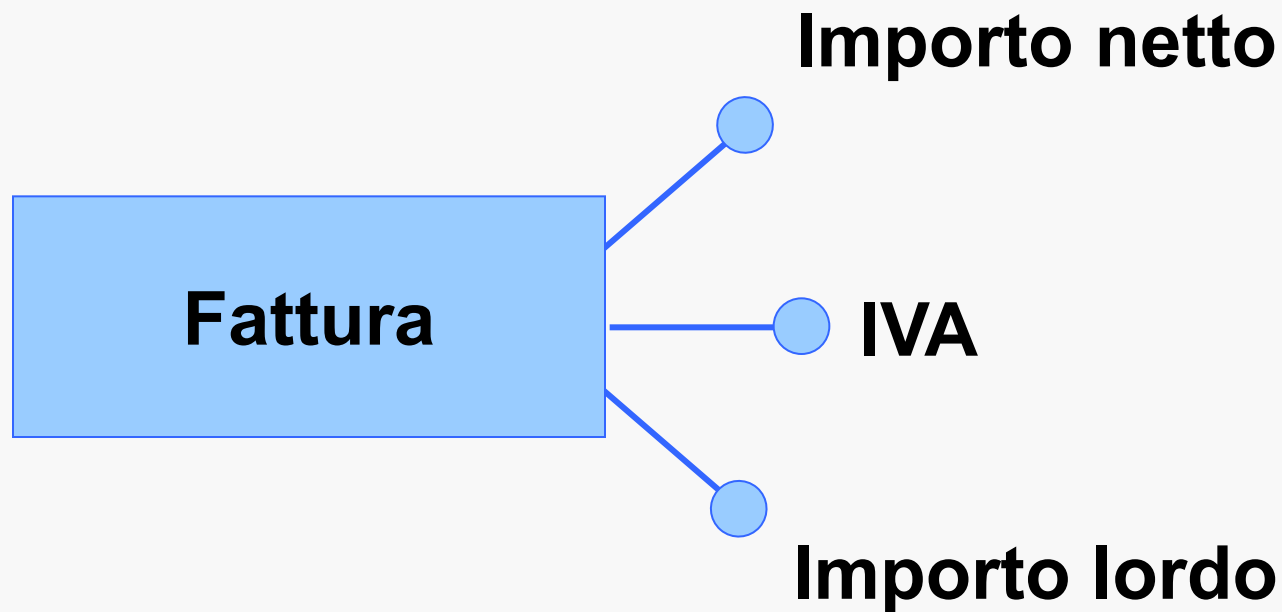
Ridondanze

- Vantaggi
 - semplificazione delle interrogazioni
- Svantaggi
 - appesantimento degli aggiornamenti
 - maggiore occupazione di spazio
 - rischi di inconsistenze

Forme di ridondanza in uno schema E-R

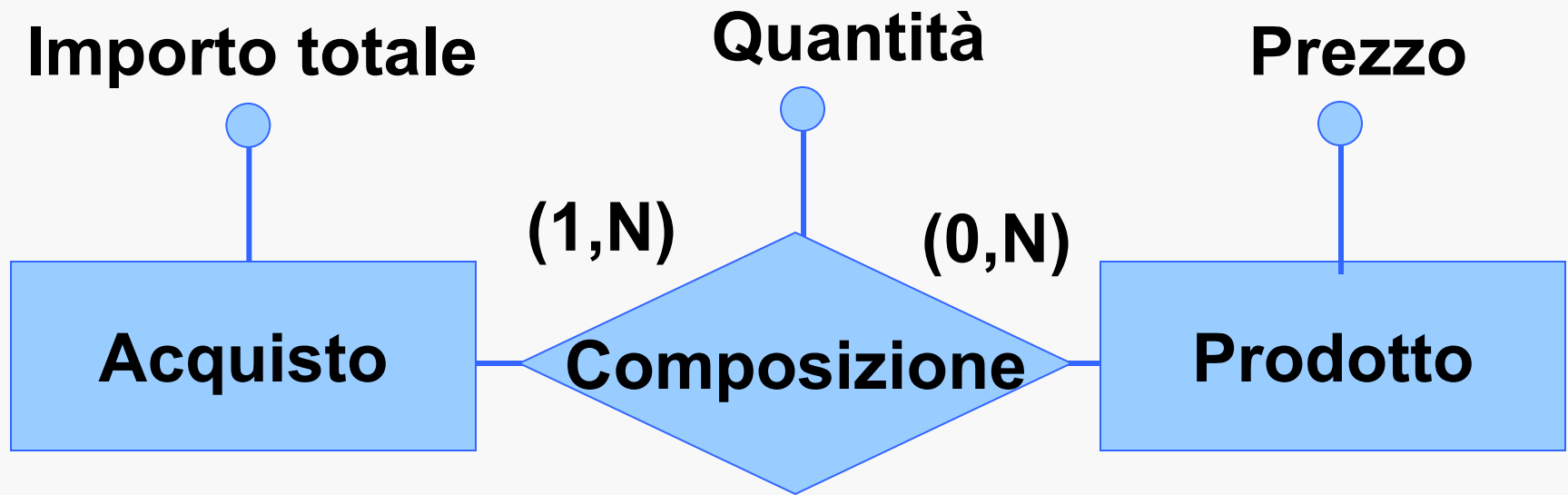
- Attributi derivabili:
 - Da altri attributi della stessa entità (o relationship)
 - Da attributi di altre entità (o relationship)
- «Relationship» derivabili dalla composizione di altre (più in generale: cicli di relationship)

Attributo derivabile

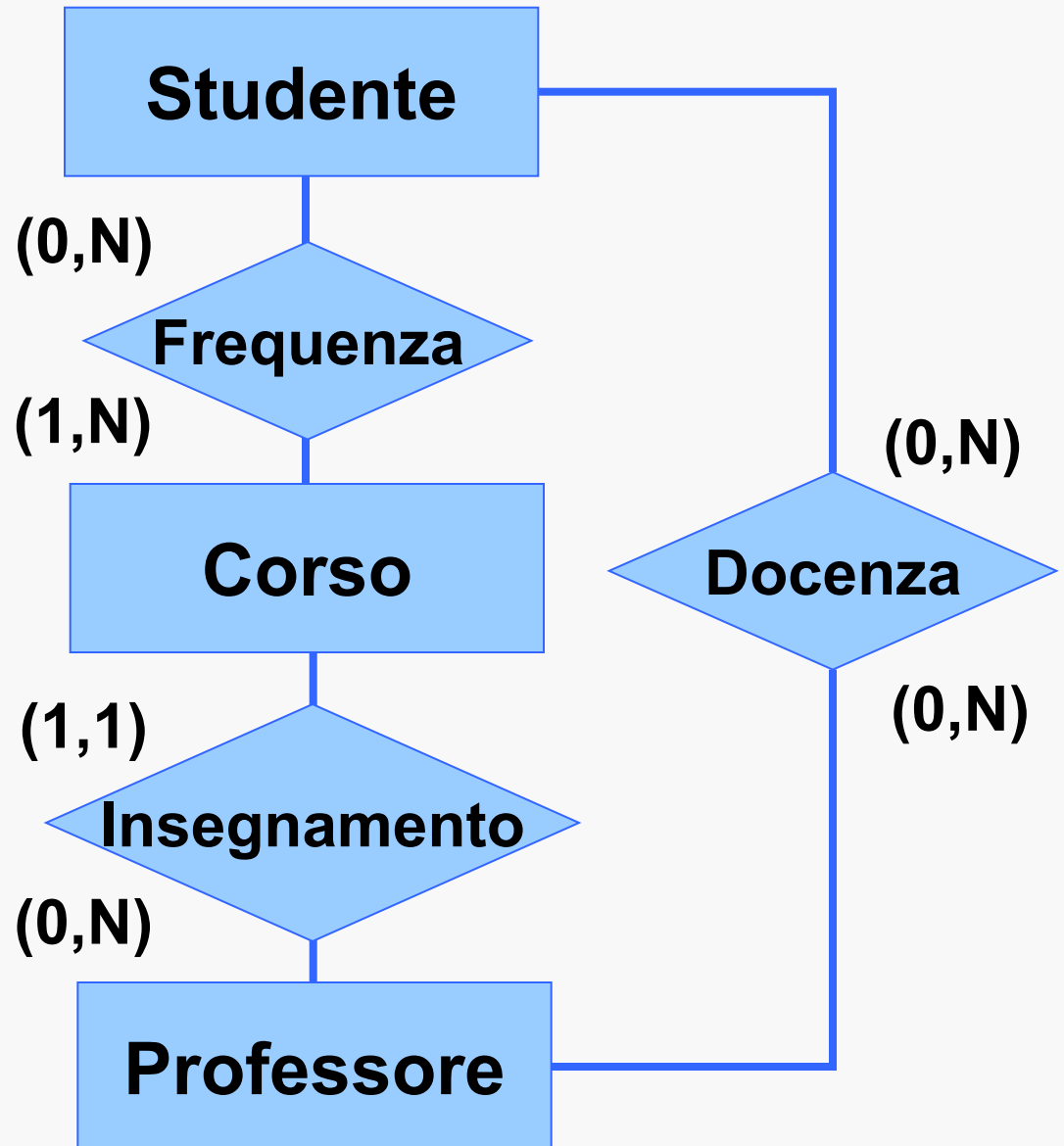


Importo netto + IVA. Importante avere la risonanza se il lordo è acceduto spesso

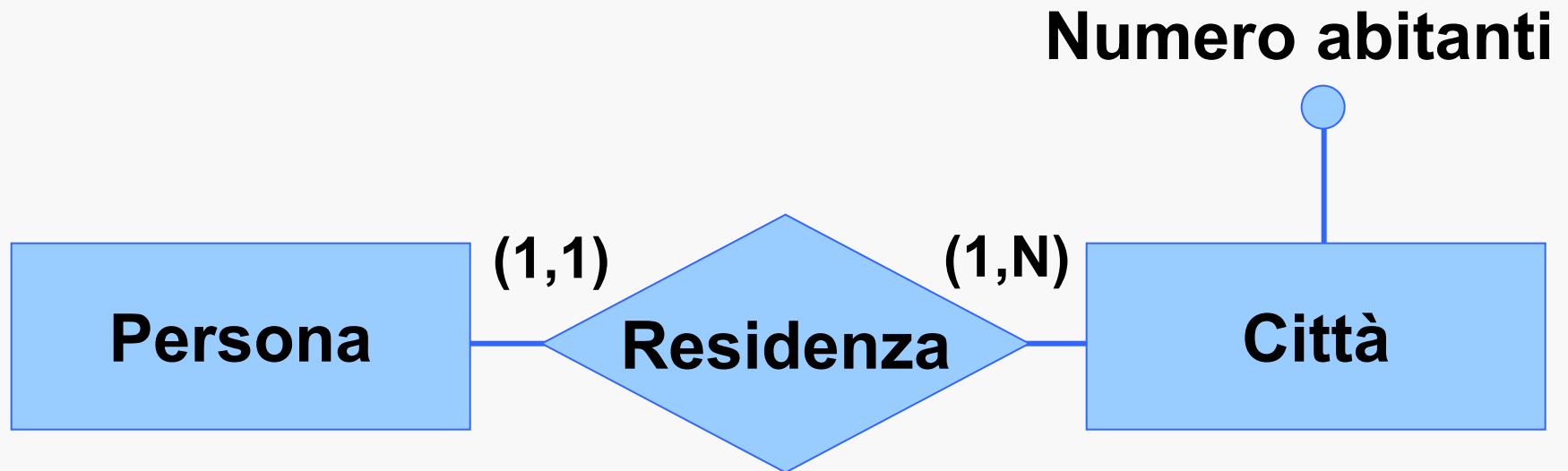
Attributo derivabile da altra entità



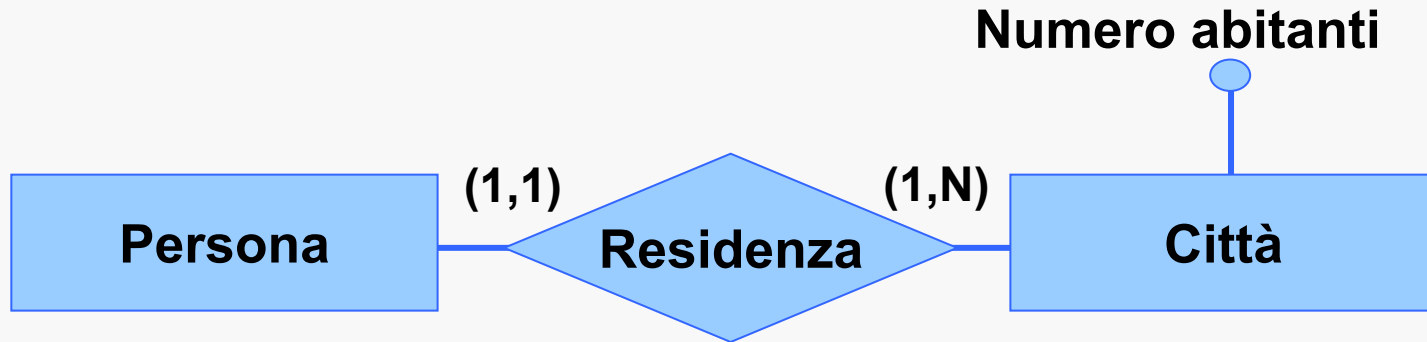
**Ridondanza dovuta a
ciclo**



Analisi di una ridondanza: è utile aggiungere «Numero abitanti»?



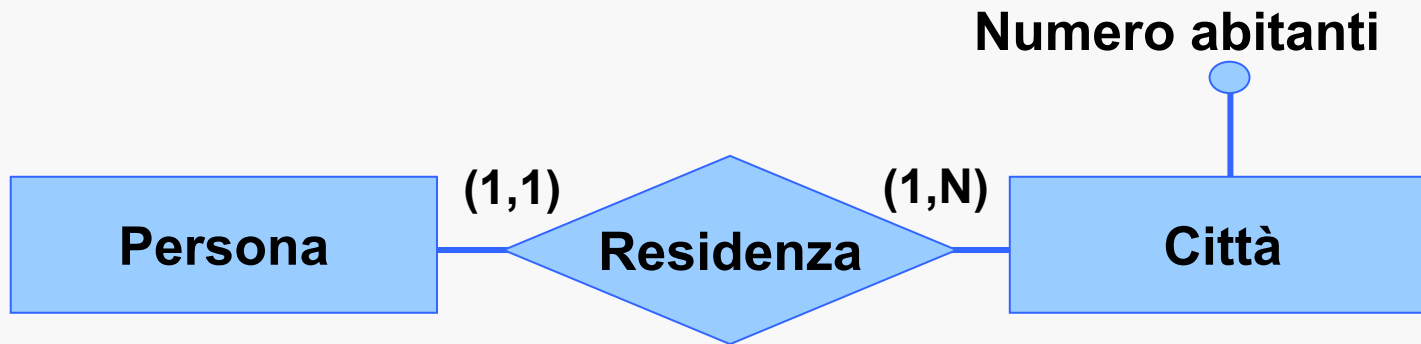
Concetto	Tipo	Volume
Città	E	200
Persona	E	1000000
Residenza	R	1000000



Concetto	Tipo	Volume
Città	E	200
Persona	E	1000000
Residenza	R	1000000

- **Operazione 1 (500 volte al giorno):** memorizza una nuova persona con la relativa città di residenza
- **Operazione 2 (2 volte al giorno):** stampa tutti i dati di una città (incluso il numero di abitanti, ca. $1M / 200 = 5000$ per città)

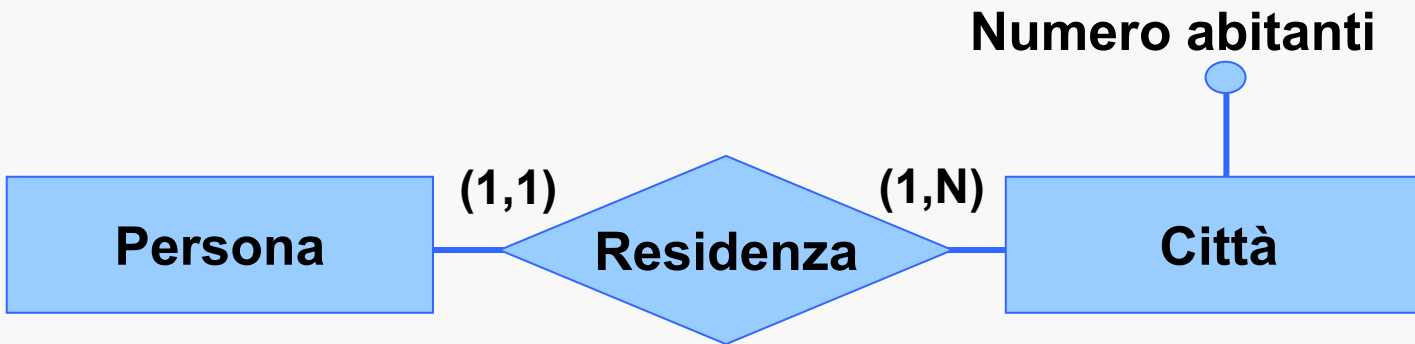
Presenza di ridondanza



- Operazione 1 (500 volte al giorno): memorizza una nuova persona con la relativa città di residenza

Concetto	Costrutto	Accessi	Tipo	
Persona	Entità	1	S	x 500 volte/giorno
Residenza	Relazione	1	S	x 500 volte/giorno
Città	Entità	1	L	x 500 volte/giorno
Città	Entità	1	S	x 500 volte/giorno

Presenza di ridondanza



- **Operazione 2 (2 volte al giorno):** stampa tutti i dati di una città (incluso il numero di abitanti, ca. $1M / 200 = 5000$ per città)

Operazione 2

Concetto	Costrutto	Accessi	Tipo
Città	Entità	1	L

x 2 volte/giorno

Presenza di ridondanza

- Costi:
 - Operazione 1: 1500 accessi in scrittura e 500 accessi in lettura al giorno
 - Operazione 2: 2 accessi in lettura.
- Assumendo costo doppio per gli accessi in scrittura

Il costo giornaliero è $1500 \times 2 + 500 + 2 = 3502$

Assenza di ridondanza



- **Operazione 1 (500 volte al giorno):** memorizza una nuova persona con la relativa città di residenza

Concetto	Costrutto	Accessi	Tipo	
Persona	Entità	1	S	x 500 volte/giorno x 500 volte/giorno
Residenza	Relazione	1	S	

Assenza di ridondanza



- **Operazione 2 (2 volte al giorno):** stampa tutti i dati di una città (incluso il numero di abitanti, ca. $1M / 200 = 5000$ per città)

Concetto	Costrutto	Accessi	Tipo	
Città	Entità	1	L	x 2 volte/giorno
Residenza	Relazione	5000	L	x 2 volte/giorno

Assenza di ridondanza

- Costi:
 - Operazione 1: 1000 accessi in scrittura
 - Operazione 2: 10002 accessi in lettura al giorno
- Assumendo costo doppio per gli accessi in scrittura

Il costo giornaliero è $1000 \times 2 + 10002 = 12002$
- Il costo giornaliero in presenza di ridondanza era 3502, che è più conveniente (implicazione: ridondanza utile)

Attività della ristrutturazione

- Analisi delle ridondanze
- Eliminazione delle generalizzazioni
- Partizionamento/accorpamento di entità e relationship
- Scelta degli identificatori primari

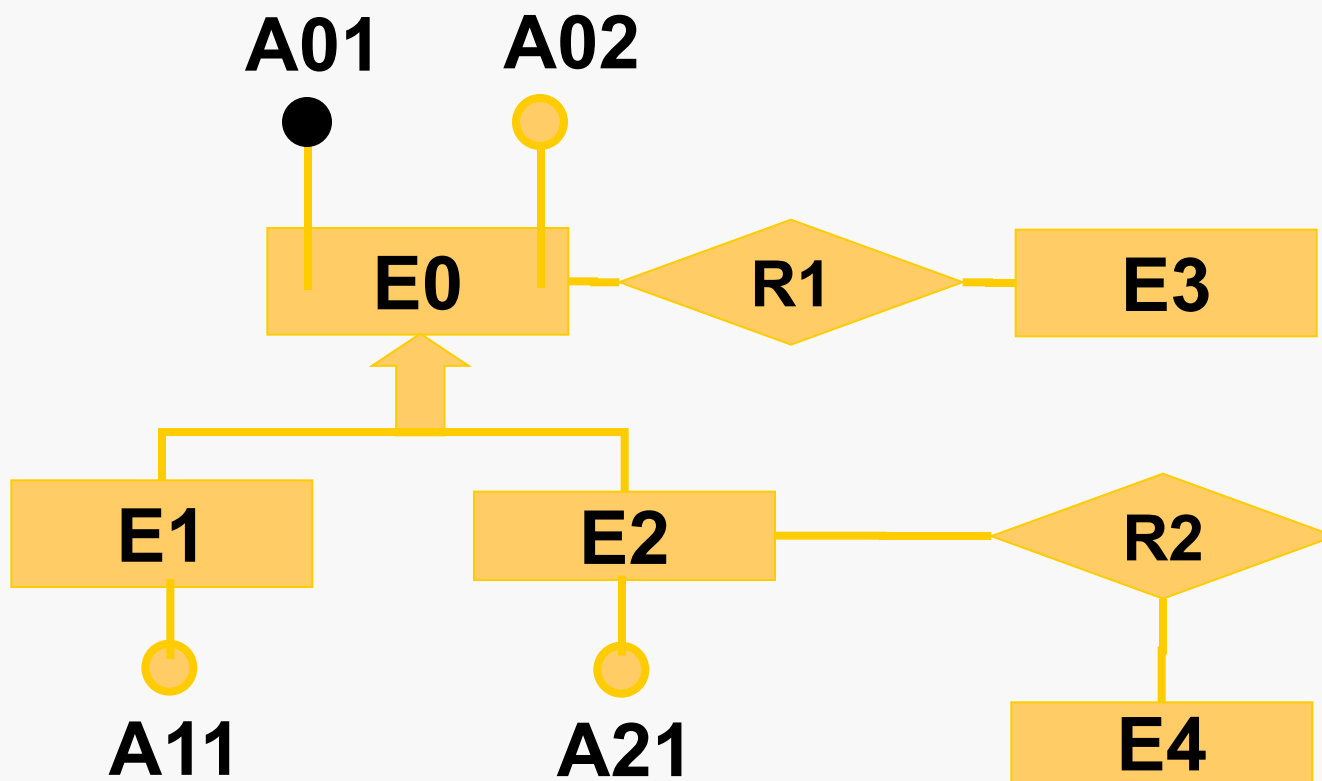
Eliminazione delle generalizzazioni / 1

- Il modello relazionale non può rappresentare direttamente le generalizzazioni
- Entità e relationship sono invece direttamente rappresentabili
- Conclusione: le generalizzazioni vanno sostituite con entità e relationship

Tre Possibilità

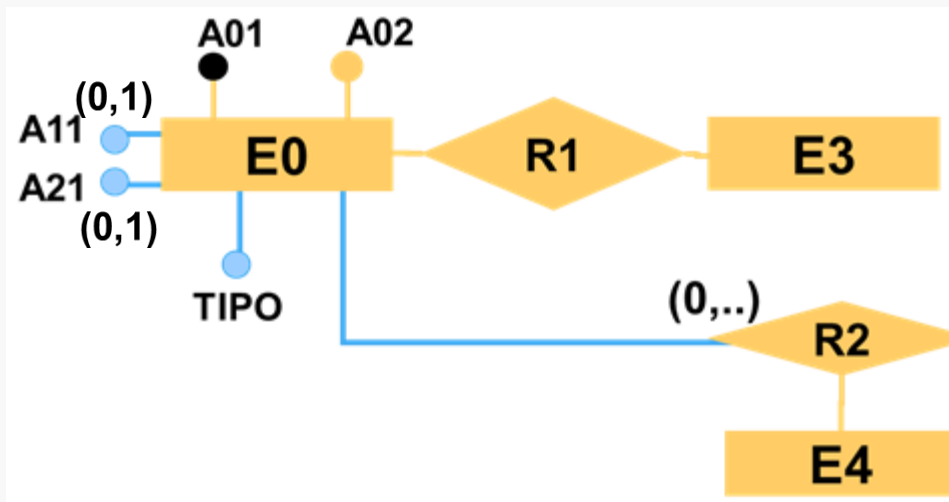
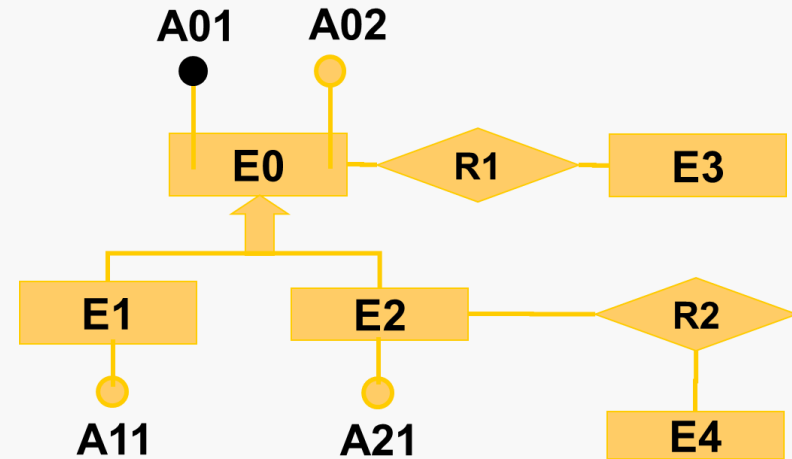
1. accorpamento delle figlie della generalizzazione nel genitore
2. accorpamento del genitore della generalizzazione nelle figlie
3. sostituzione della generalizzazione con relationship

Eliminazione delle generalizzazioni: Un esempio per le tre possibilità



Eliminazione delle generalizzazioni:

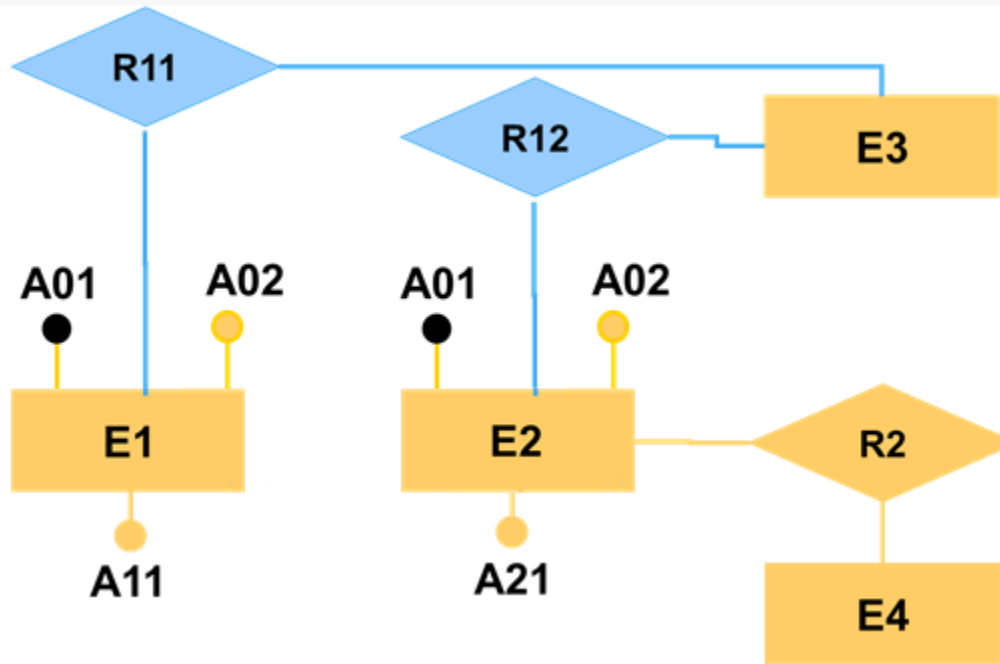
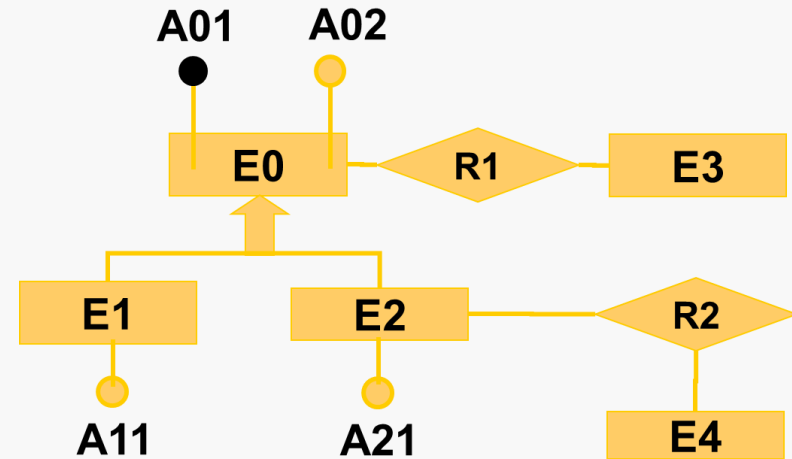
1. Accorpare le figlie nel padre



- Preferibile se gli accessi al padre e alle figlie sono contestuali
- Tabelle (es. E0) conterrà valori nulli.

Eliminazione delle generalizzazioni:

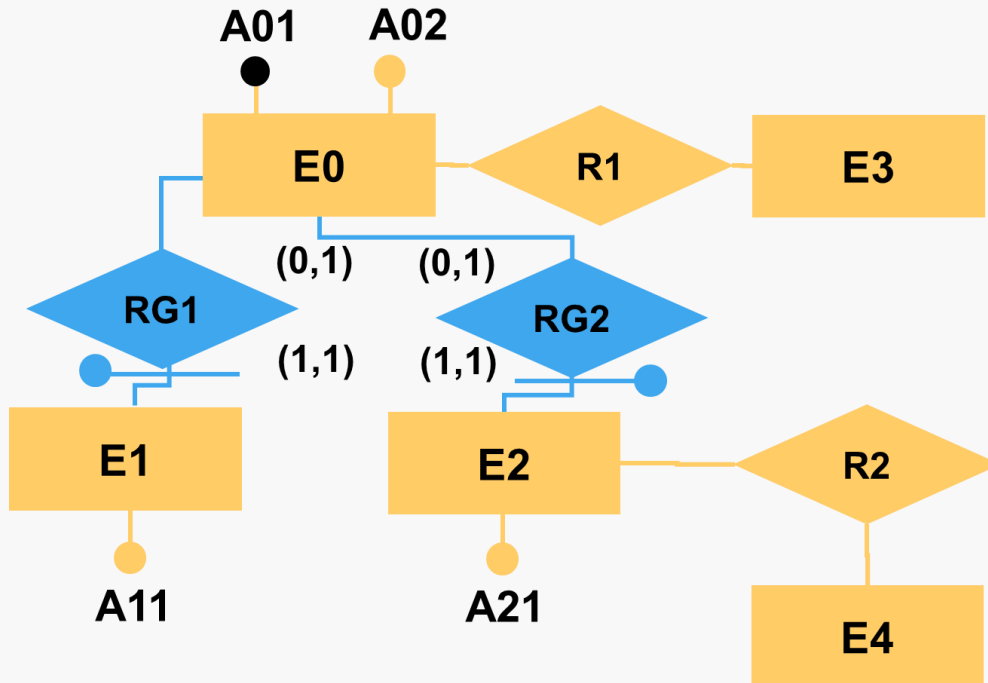
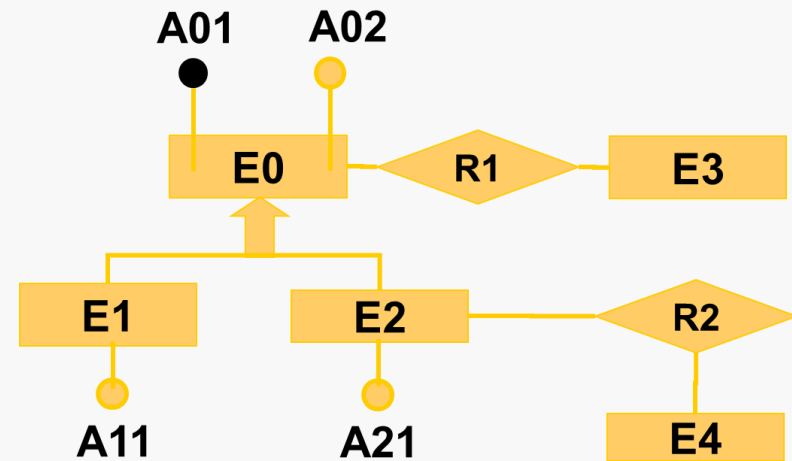
2. Accorpare il padre nelle figlie



- Preferibile se gli accessi al padre e alle figlie sono separati
- Possibile solamente se la generalizzazione è totale

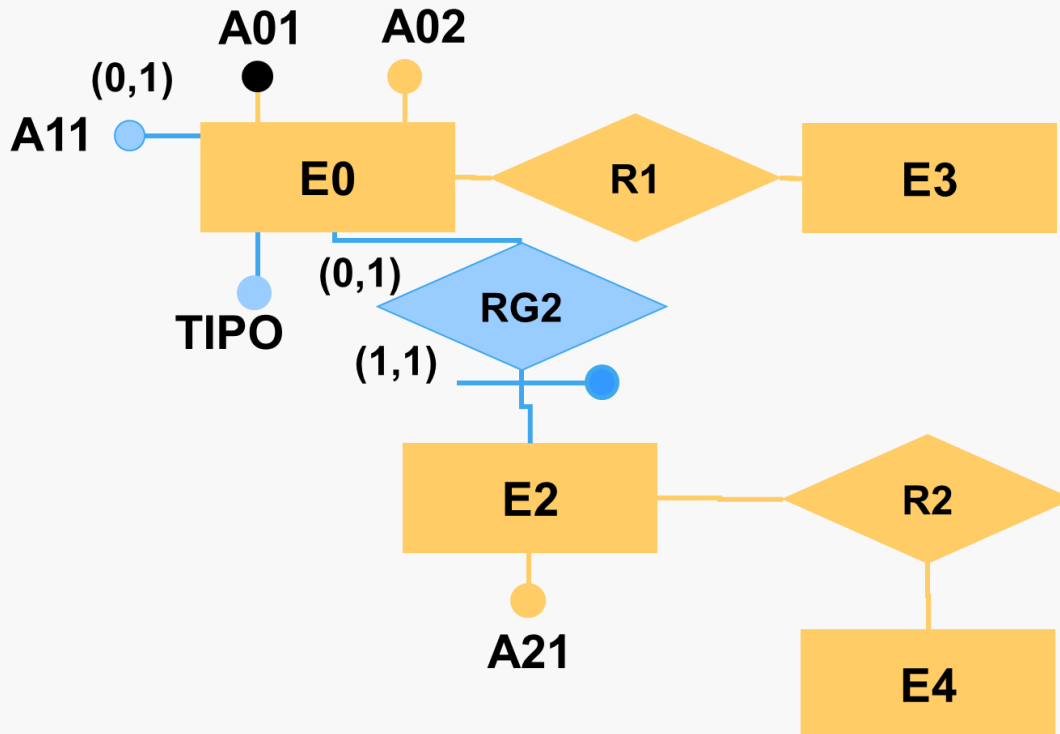
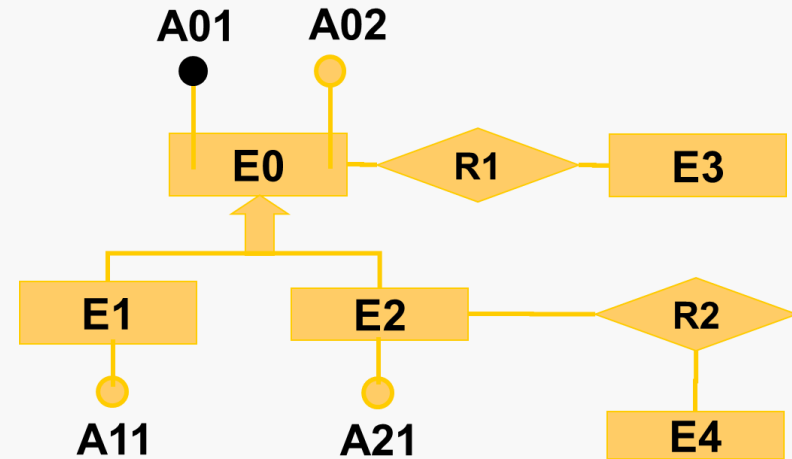
Eliminazione delle generalizzazioni:

3. Sostituire le generaliz. con relationships



- Preferibile se gli accessi al padre e alle figlie sono separati
- Va bene anche se la generalizzazione non è totale

Possibili soluzioni ibride!



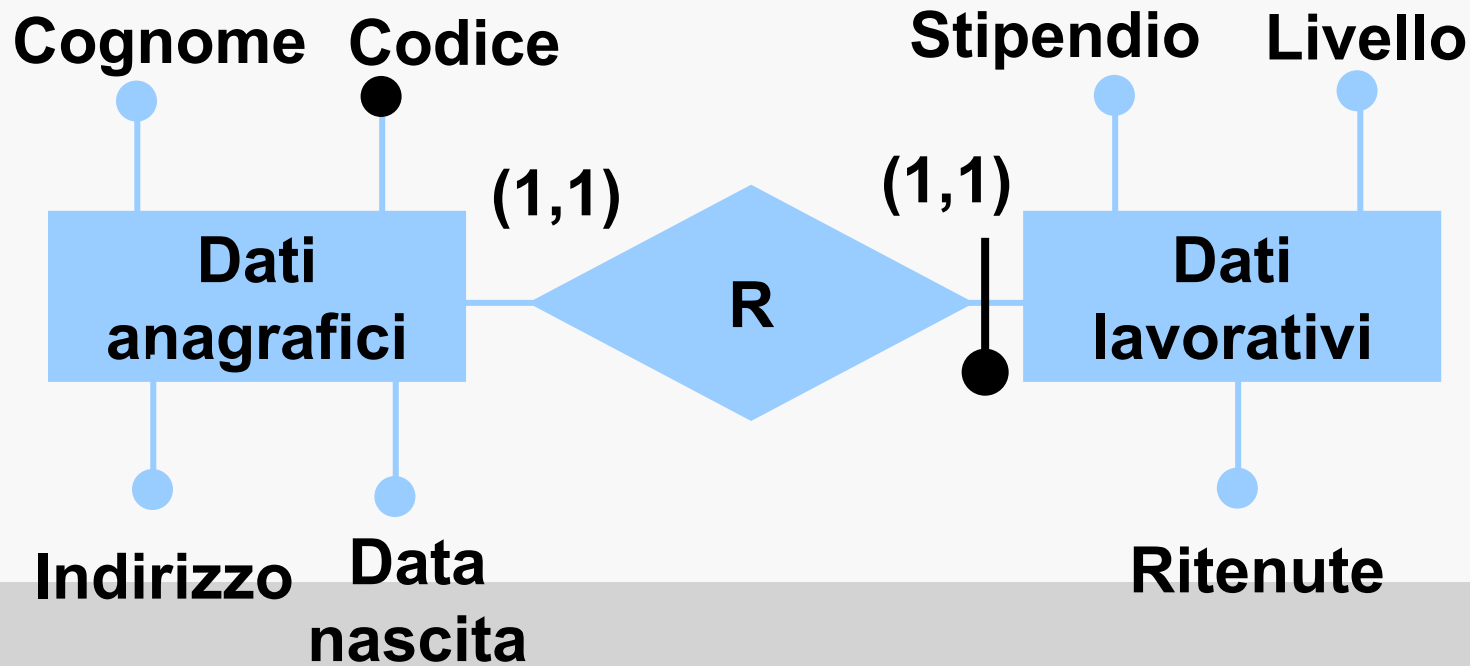
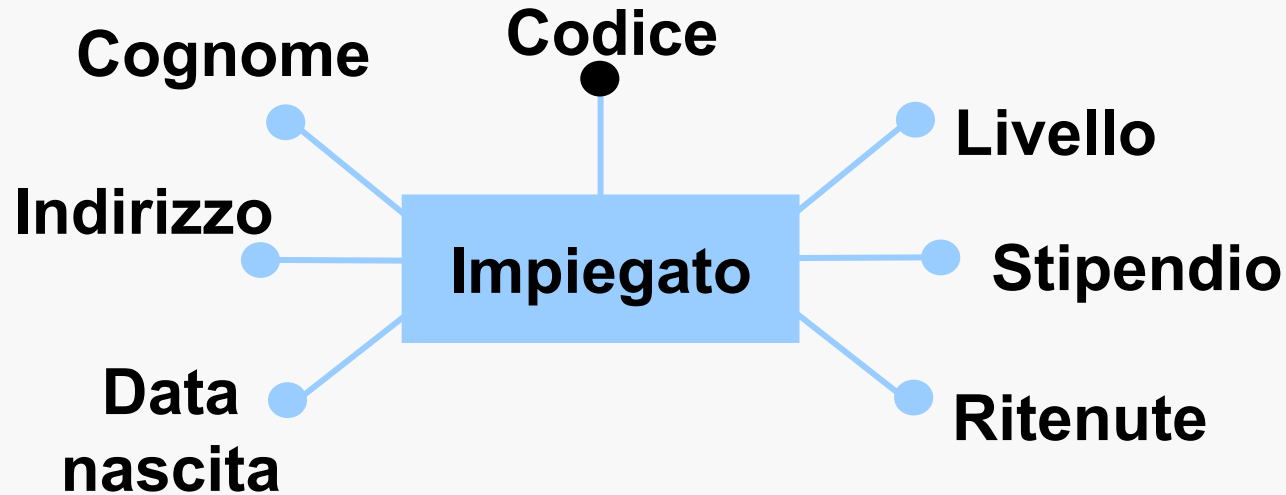
Attività della ristrutturazione

- Analisi delle ridondanze
- Eliminazione delle generalizzazioni
- Partizionamento/accorpamento di entità e relationship
- Scelta degli identificatori primari

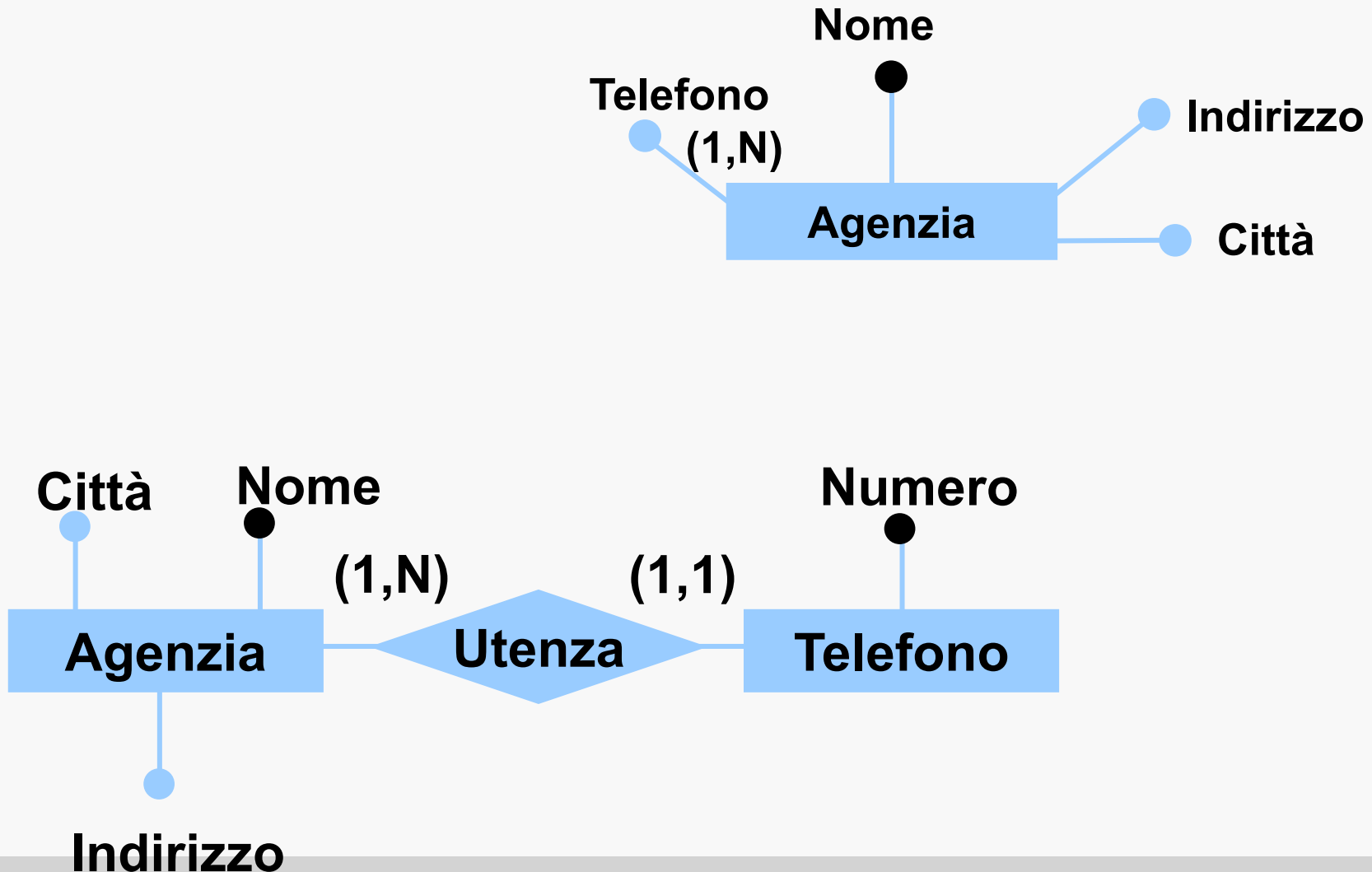
Attività di ristrutturazione

- Ristrutturazioni effettuate per rendere più efficienti le operazioni in base a un semplice principio
- Gli accessi si riducono:
 - separando attributi di un concetto che vengono acceduti separatamente
 - raggruppando attributi di concetti diversi acceduti insieme

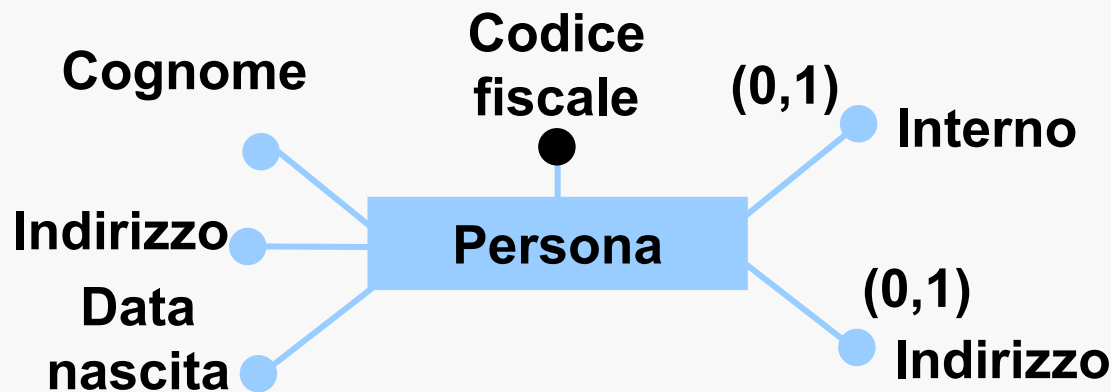
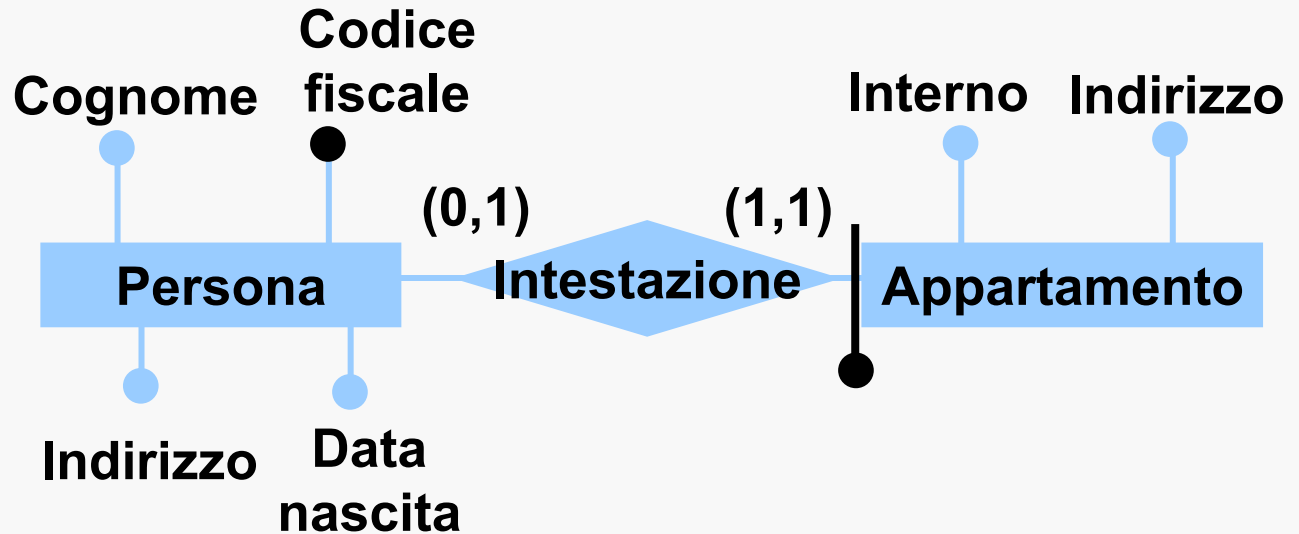
Partizionamento Verticale di Entità



Eliminazione di Attributi Multivalore



Accorpamento di entità/ relationship



Attività della ristrutturazione

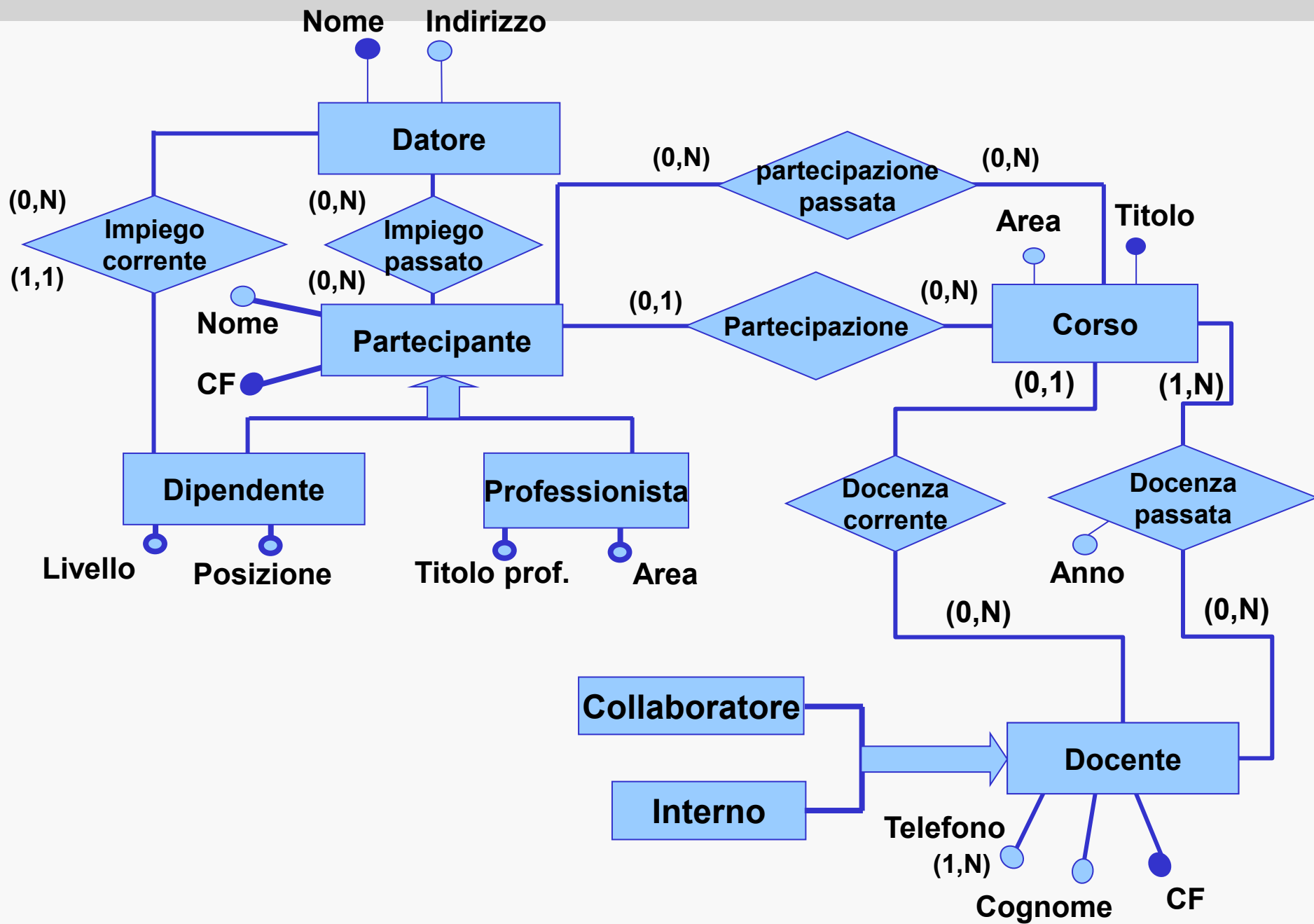
- Analisi delle ridondanze
- Eliminazione delle generalizzazioni
- Partizionamento/accorpamento di entità e relationship
- Scelta degli identificatori principali

Scelta degli Identificatori Principali

- Operazione indispensabile per la traduzione nel modello relazionale
- Criteri
 - Assenza di Opzionalità
 - Semplicità
 - Utilizzo nelle operazioni più frequenti o importanti

**Se non esistono identificatori per certe entità,
si aggiungono attributi con codici**

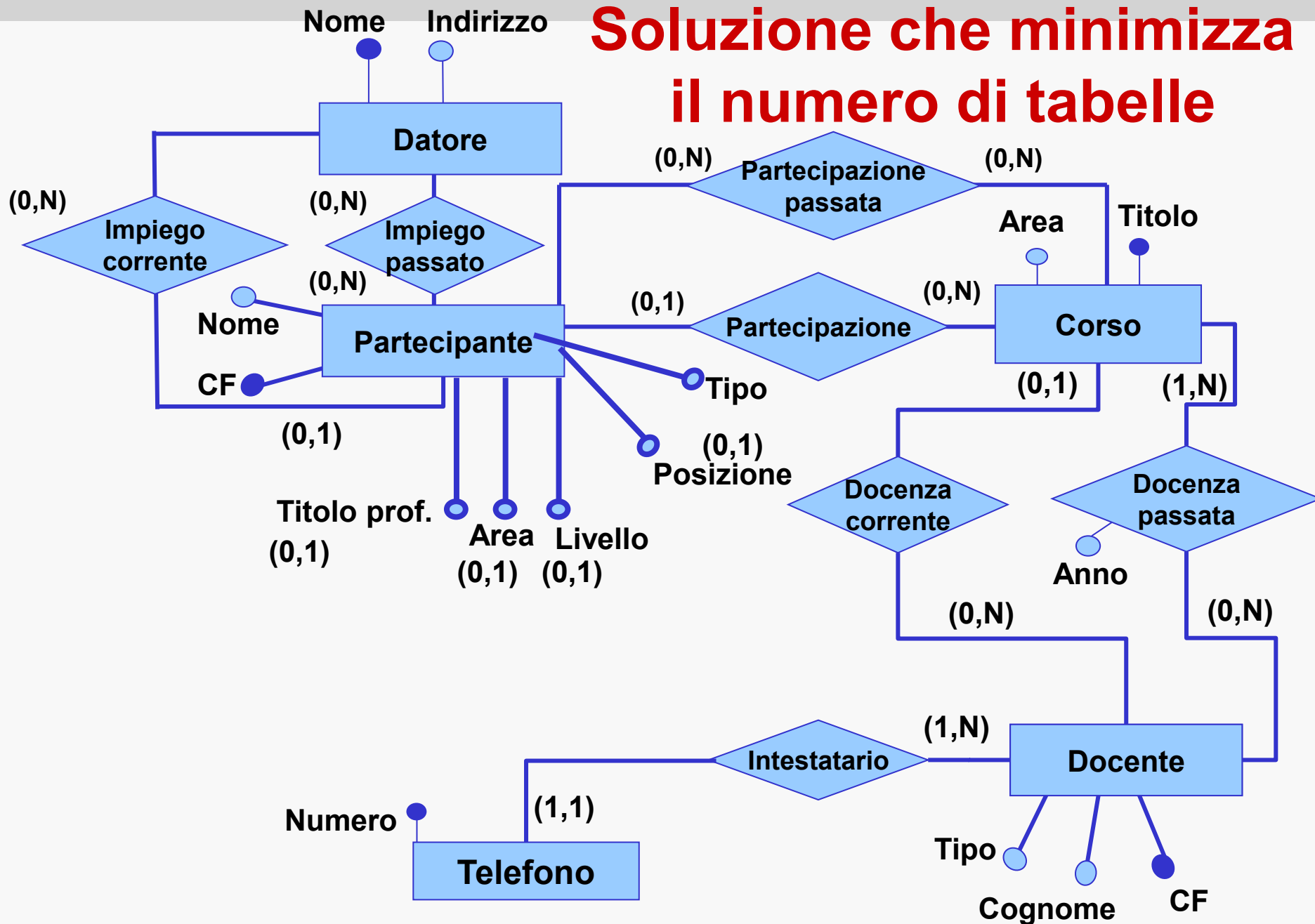
ESEMPIO DI RISTRUTTURAZIONE



Soluzione che minimizza i valori nulli



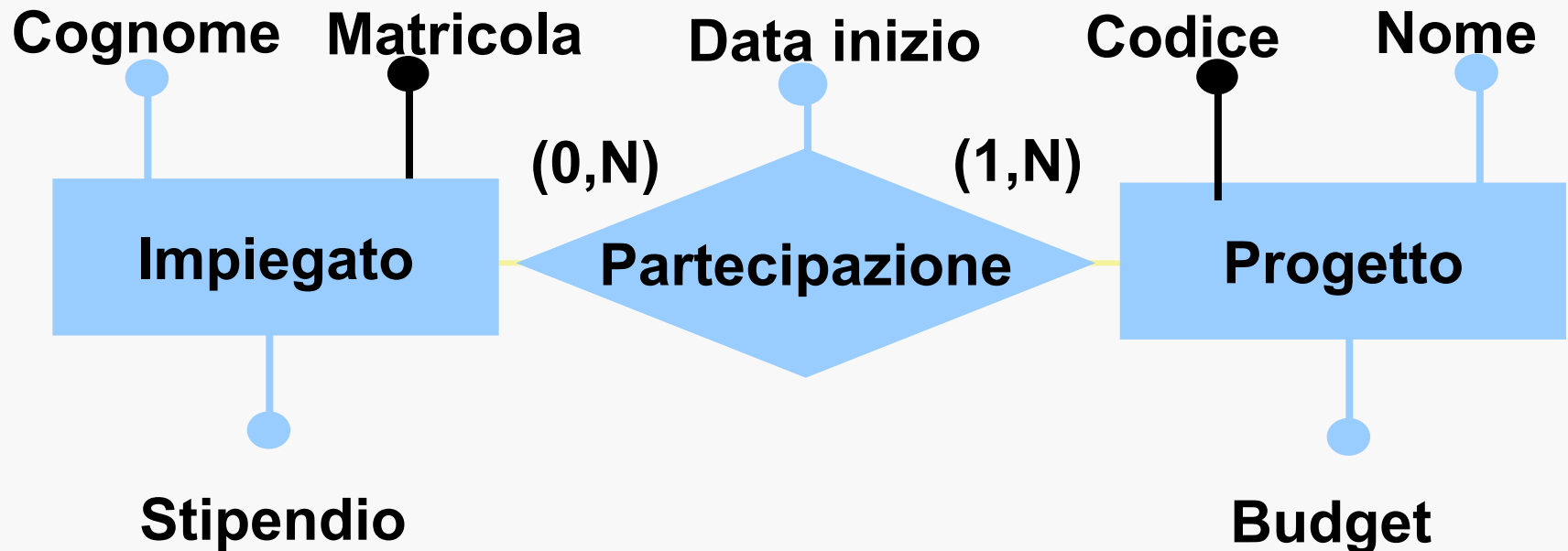
Soluzione che minimizza il numero di tabelle



Traduzione Verso il Modello relazionale

- Entità diventano relazioni sugli stessi attributi, usando gli identificatori come chiavi primarie
- Una relationship tra entità E_1, \dots, E_n diventa una relazione con attributi:
 - Identificatori di E_1, \dots, E_n (che diventano insieme chiave)
 - Attributi propri della relationship

Entità e Relationship molti a molti



Impiegato(Matricola, Cognome, Stipendio)
Progetto(Codice, Nome, Budget)
Partecipazione(Matricola, Codice, DataInizio)

Chiavi Esterne

Impiegato(Matricola, Cognome, Stipendio)

Progetto(Codice, Nome, Budget)

Partecipazione(Matricola, Codice, DataInizio)

Si aggiungono poi i vincoli di integrità referenziale

(Attributo_Rel_Esterna → Attributo_Rel_Referenziata)

Partecipazione.Matricola → Impiegato.Matricola

Partecipazione.Codice → Progetto.Codice

Meglio nomi più espressivi nelle relazioni derivate da relationships

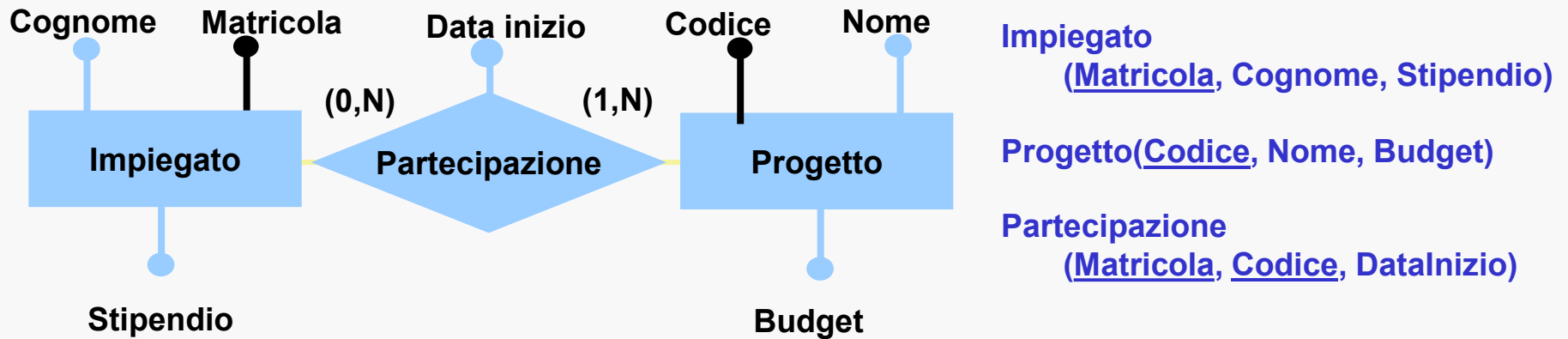
Impiegato(Matricola, Cognome, Stipendio)

Progetto(Codice, Nome, Budget)

Partecipazione(Matricola, Codice, DataInizio)

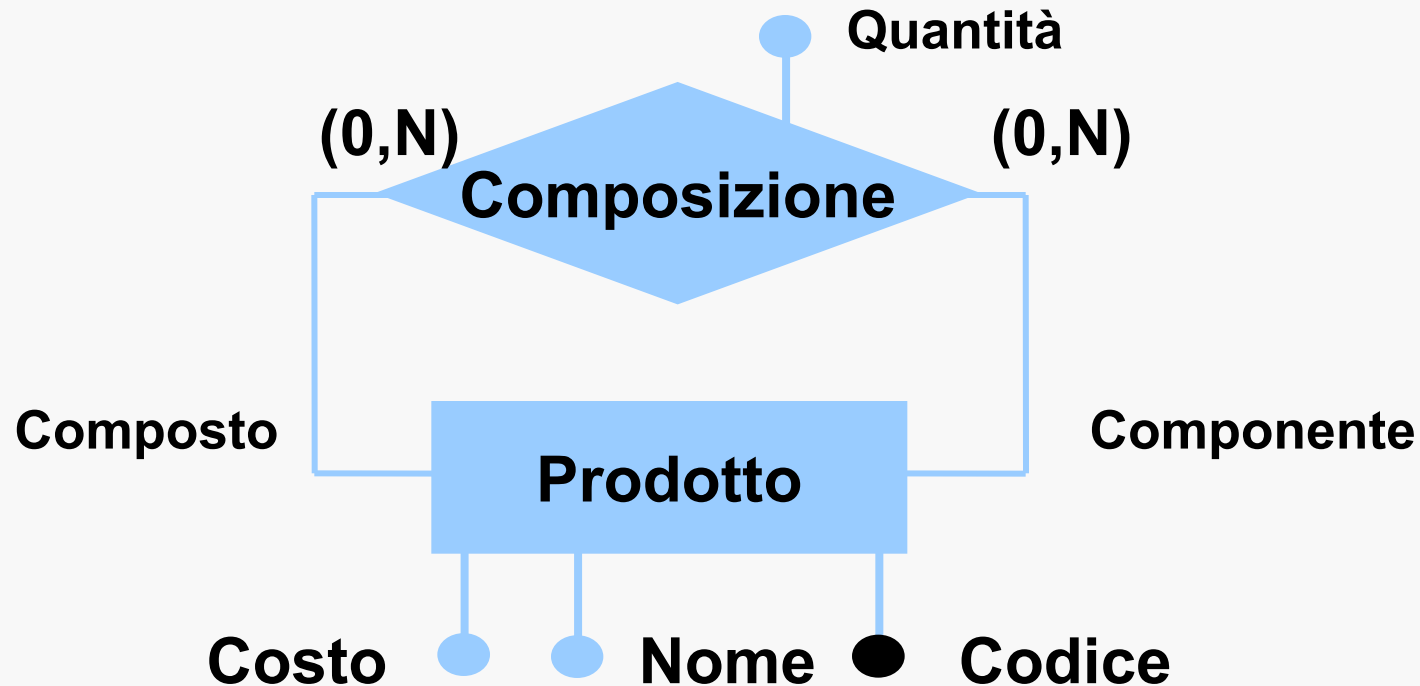
Partecipazione(Impiegato, Progetto, DataInizio)

Traduzione non garantisce i vincoli di cardinalità minima in relationship N-a-N!



- Esempio: Possibile tupla $(C, \dots, \dots) \in \text{Progetto}$ e nessuna tupla $(\dots, C, \dots) \in \text{Partecipazione}$
- I vincoli di CHECK non supportati da quasi nessun DBMS.

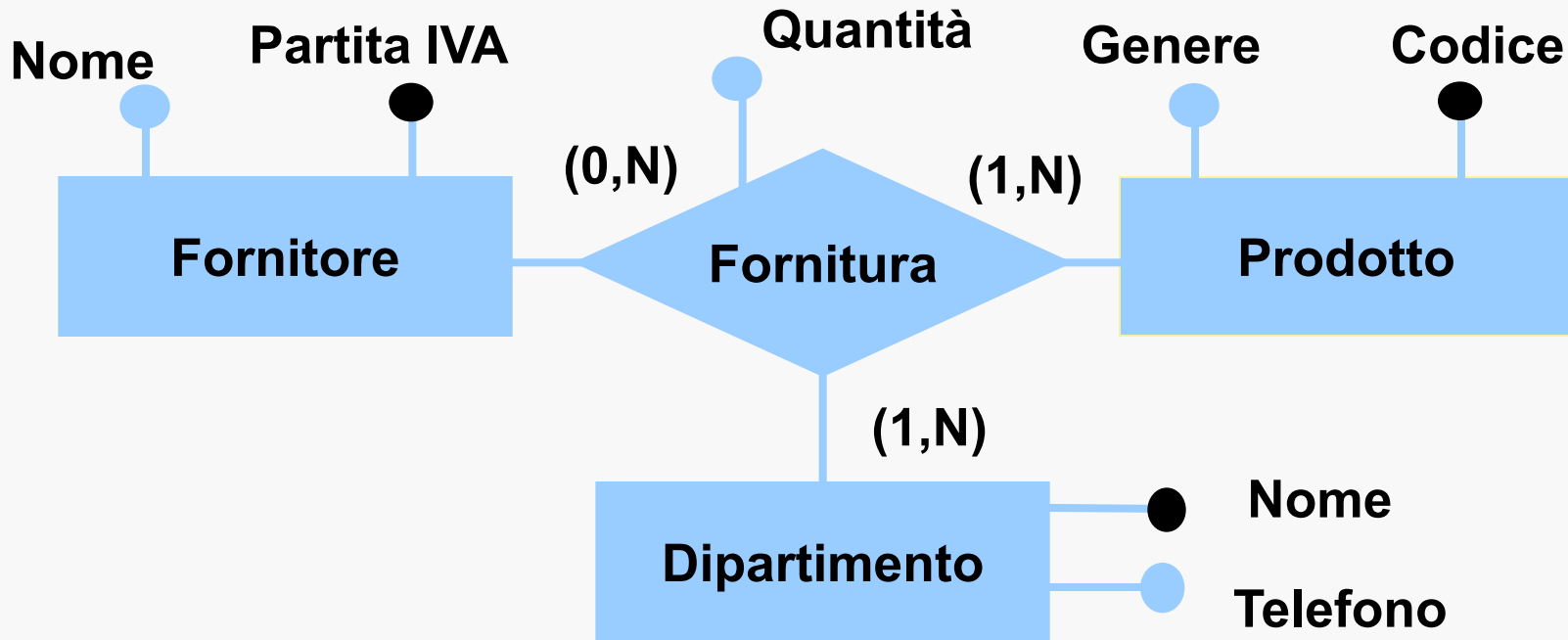
Esempio: Relationship ricorsive



Prodotto(Codice, Nome, Costo)
Composizione(Composto, Componente, Quantità)

Composizione.Composto → Prodotto.Codice
Composizione.Componente → Prodotto.Codice

Esempio: Relationship n-arie



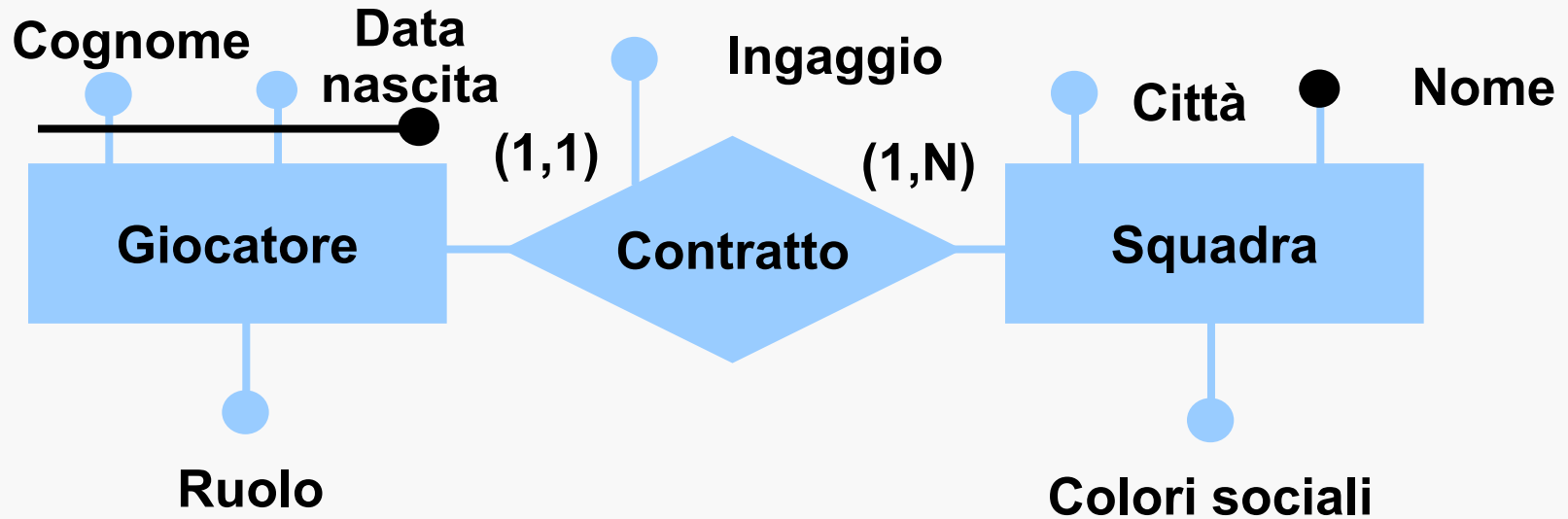
Fornitore (PartitaIVA, Nome)
Prodotto (Codice, Genere)
Dipartimento (Nome, Telefono)
Fornitura (Fornitore, Prodotto, Dipartimento, Quantità)

Fornitura.Fornitore → Fornitore.PartitaIVA

Fornitura.Prodotto → Prodotto.Codice

Fornitura.Dipartimento → Dipartimento.Nome

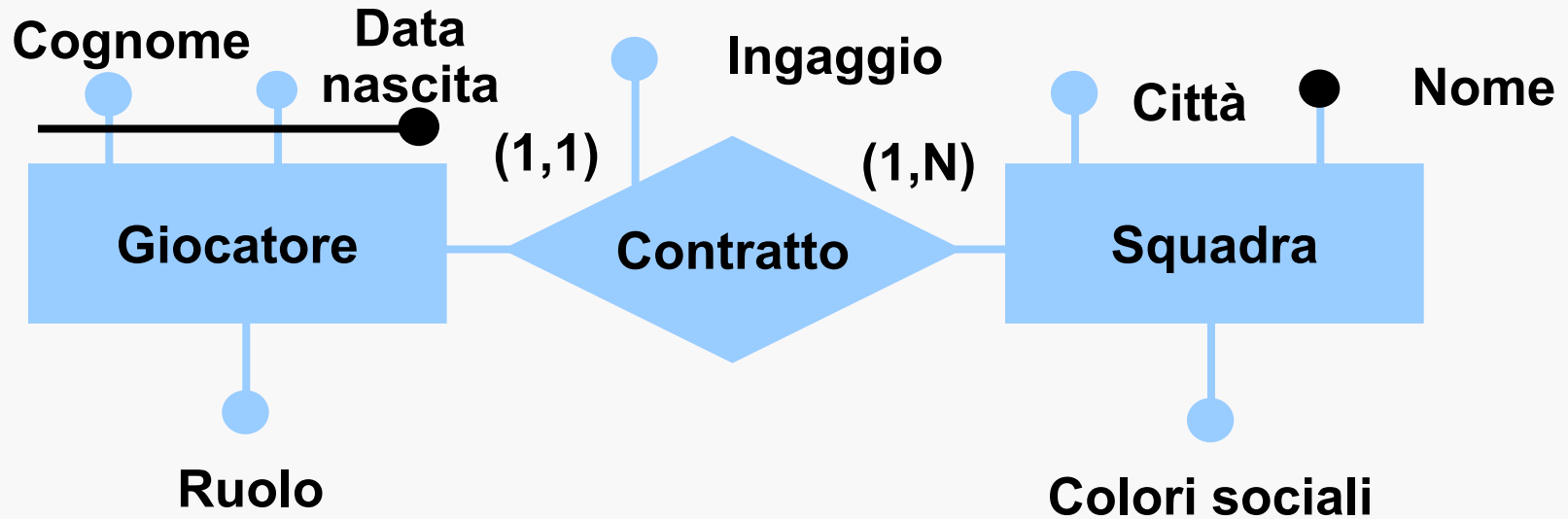
Relationship 1 a N: Soluzione Scorretta



Giocatore(Cognome, DataNascita, Ruolo)
Contratto(CognGiocatore, DataNascG, Squadra, Ingaggio)
Squadra(Nome, Città, ColoriSociali)

- Possibile aggiungere le seguenti tuple a Contratto:
(CG,DN,SQ1,3000) e (CG,DN,SQ2,4000)
- Violato il vincolo (1,1)

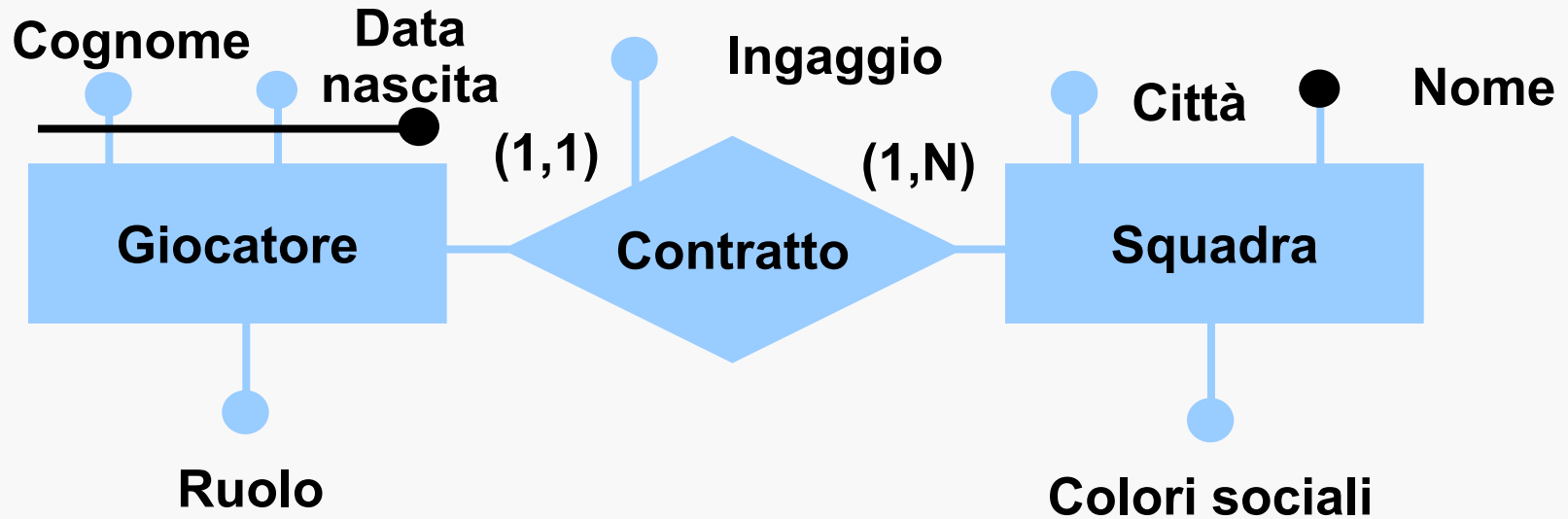
Relationship 1 a N: Soluzione Corretta



Giocatore(Cognome, DataNascita, Ruolo)
Contratto(CognGiocatore, DataNascG, Squadra, Ingaggio)
Squadra(Nome, Città, ColoriSociali)

Tuttavia, **Contratto** ha stessa Chiave di **Giocatore**: Ridondanza Non Necessaria

Relationship 1 a N: Soluzione Migliore



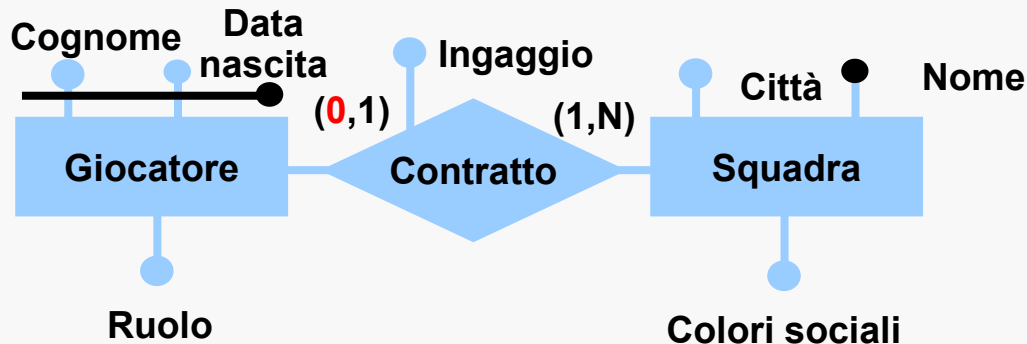
Giocatore(Cognome, DataNascita, Ruolo, **Squadra**, Ingaggio)

Contratto

Squadra(Nome, Città, ColoriSociali)

Cardinalità Minima Rappresentabile con partecipazione (x,1)

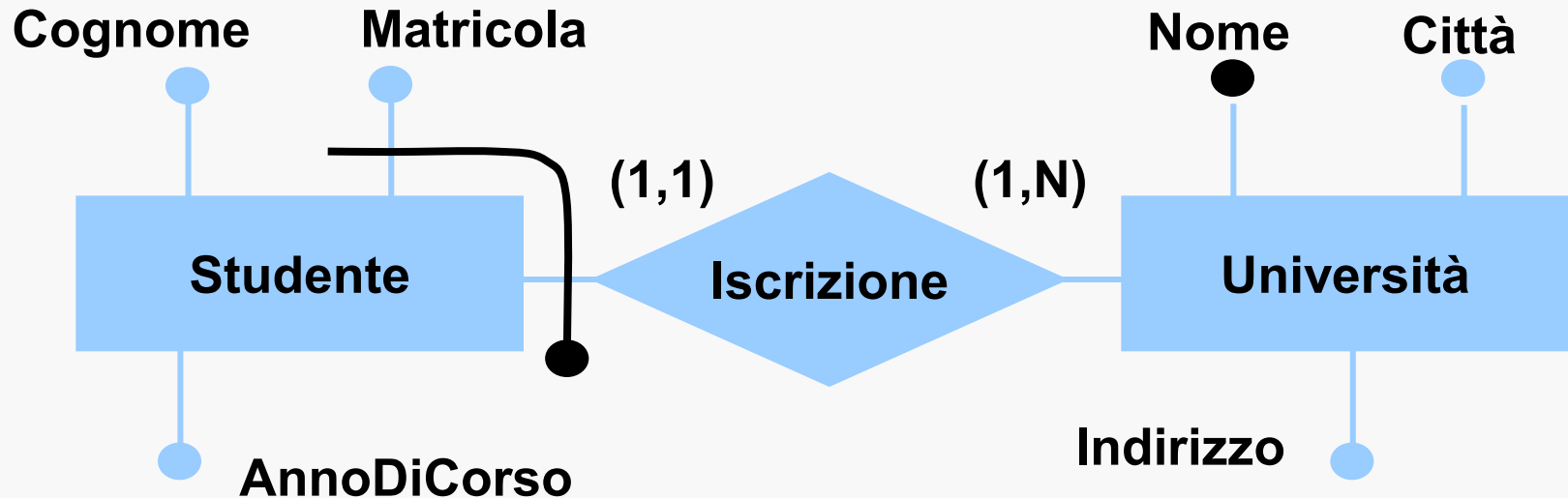
- La traduzione riesce a rappresentare efficacemente la cardinalità minima della partecipazione che ha 1 come cardinalità massima:
 - 0 : valore nullo ammesso
 - 1 : valore nullo non ammesso



Se cardinalità (0,1),
allora **Squadra** e
Ingaggio ammettono
valori NULL

Giocatore(CognGiocatore, DataNascG, Ruolo, Squadra, Ingaggio)
Squadra(Nome, Città, ColoriSociali)

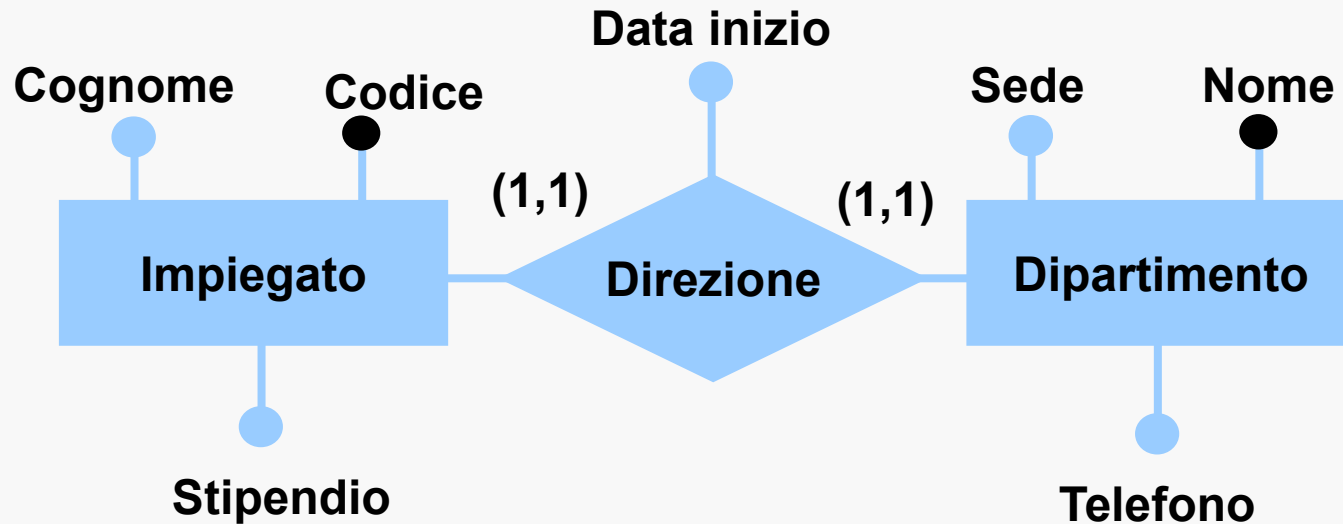
Entità con identificazione esterna



Studente(Matricola, Università, Cognome, AnnoDiCorso)
Università(Nome, Città, Indirizzo)

con vincolo di identità referenziale (chiave esterna):
Università → **Nome**

Relationship uno a uno / 1



Possibilità di fondere su Impiegato:

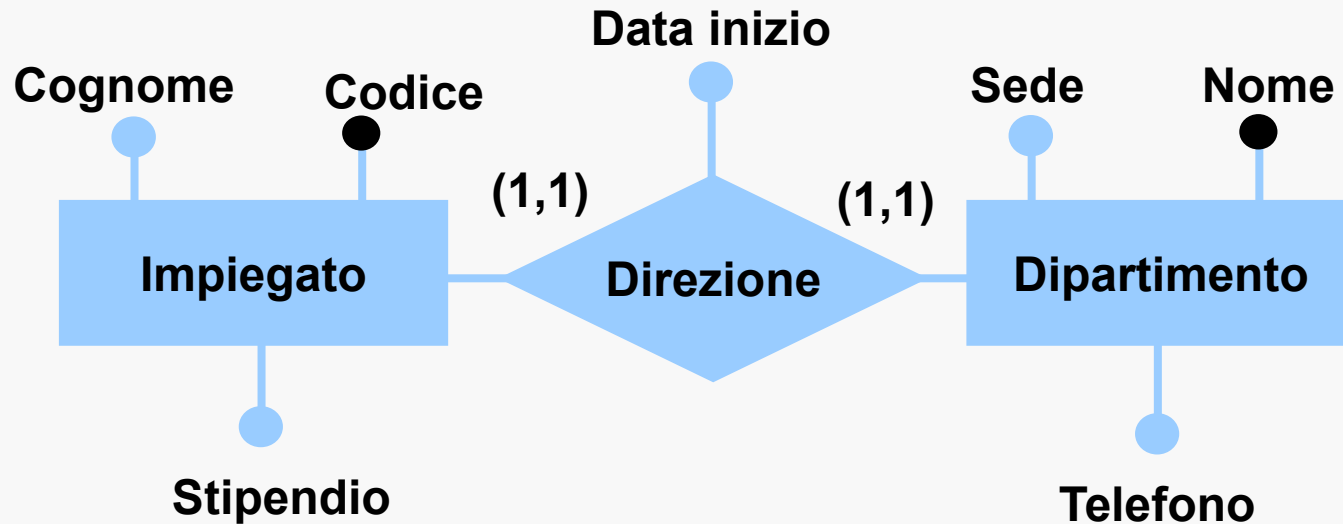
Impiegato (Codice, Cognome, Stipendio, **NomeDip**, **InizioD**)

Dipartimento (Nome, Sede, Telefono)

con vincoli:

1. **NomeDip** e **InizioD** non possono essere NULL
2. di chiave esterna: **NomeDip** -> **Nome**

Relationship uno a uno / 2



Possibilità di fondere su Dipartimento:

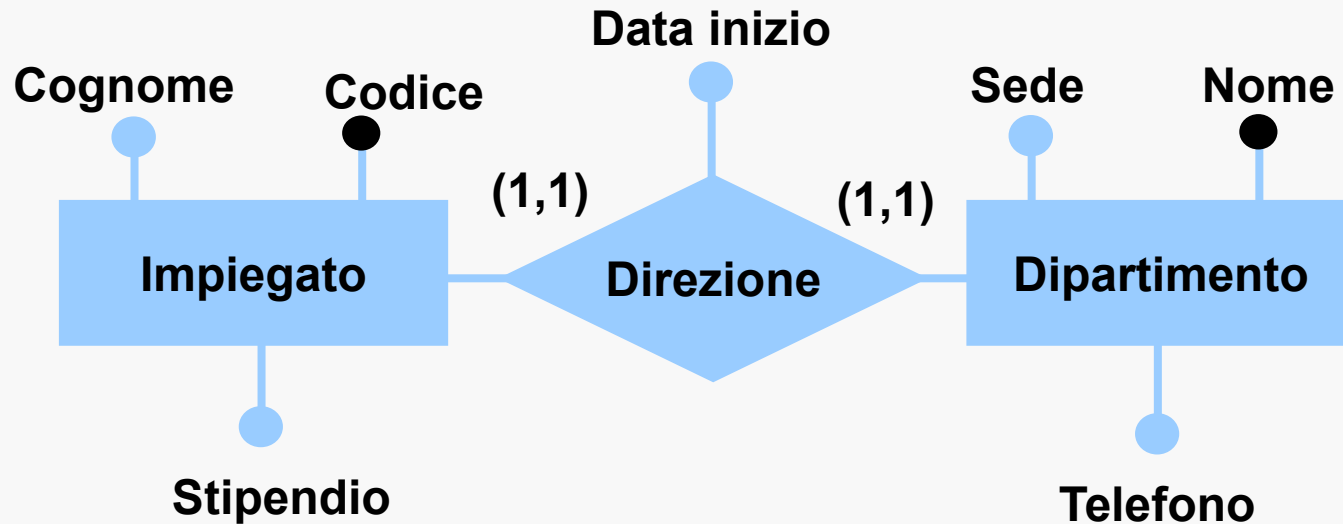
Impiegato (Codice, Cognome, Stipendio)

Dipartimento (Nome, Sede, Telefono, **InizioD**, **CodDirettore**)

con vincoli:

1. **CodDirettore** e **InizioD** non possono essere NULL
2. di chiave esterna: **CodDirettore** -> **Codice**

Relationship uno a uno / 3



Possibilità di fondere su Dipartimento e Impiegato:

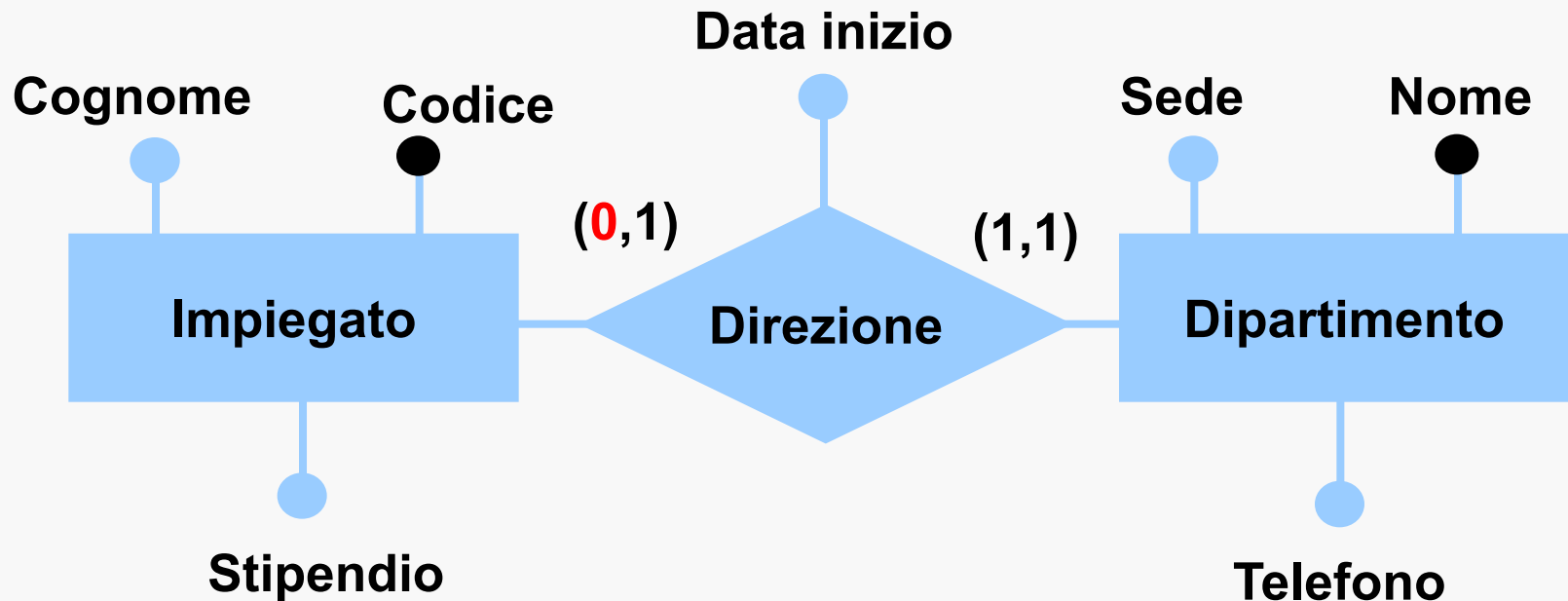
Impiegato (Codice, Cognome, Stipendio, **NomeDip**)

Dipartimento (Nome, Sede, Telefono, InizioD, **CodDirettore**)

con vincoli:

1. **CodDirettore** e **NomeDip** non possono essere NULL
2. di chiave esterna: **CodDirettore** -> **Codice**; **NomeDip** -> **CodDirettore**

Una possibilità privilegiata



Impiegato (Codice, Cognome, Stipendio)

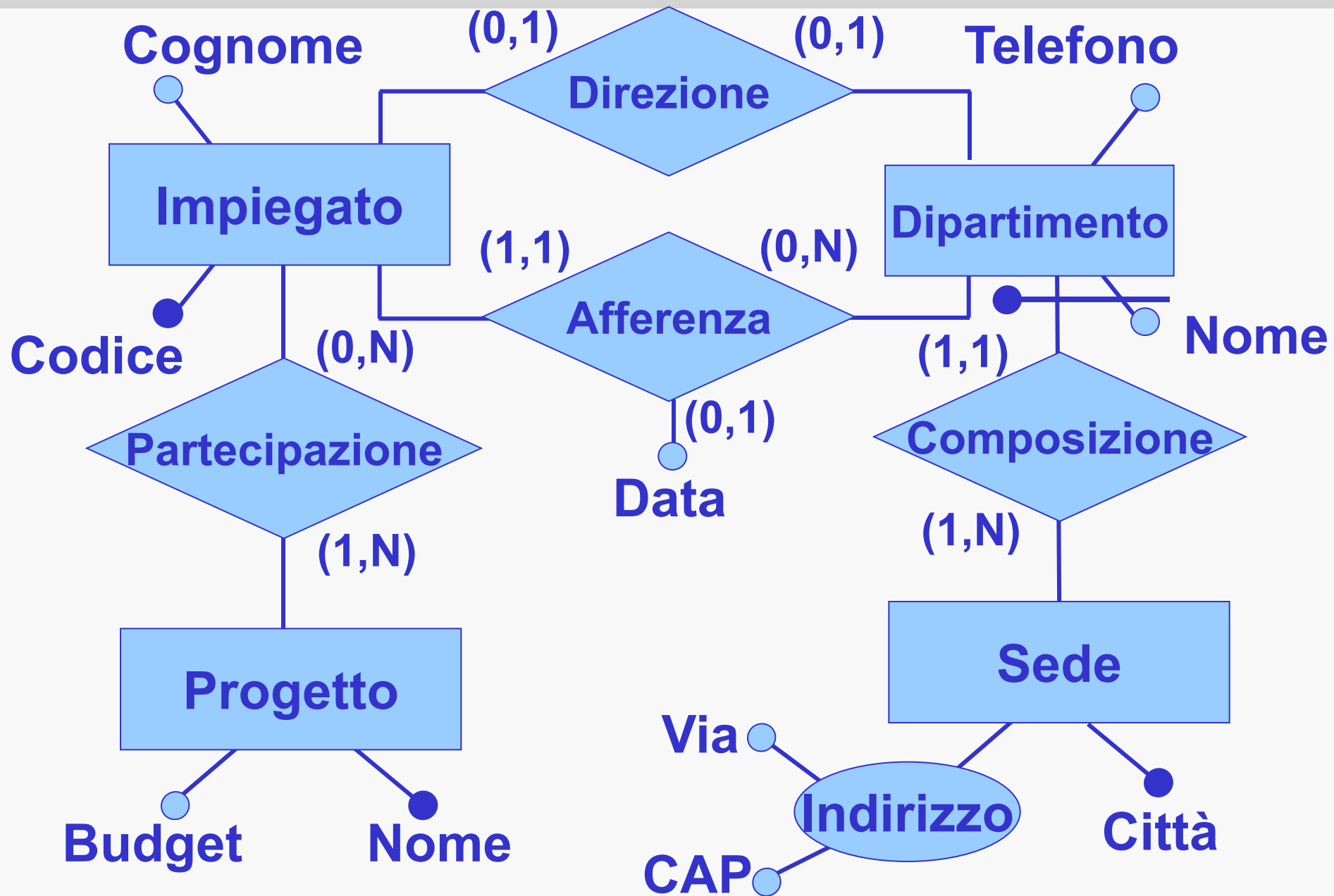
Dipartimento (Nome, Sede, Telefono, InizioD, **CodDirettore**)

con vincoli:

1. **CodDirettore** non può essere NULL
2. di chiave esterna: **CodDirettore** -> **Codice**

ESERCIZIO 1

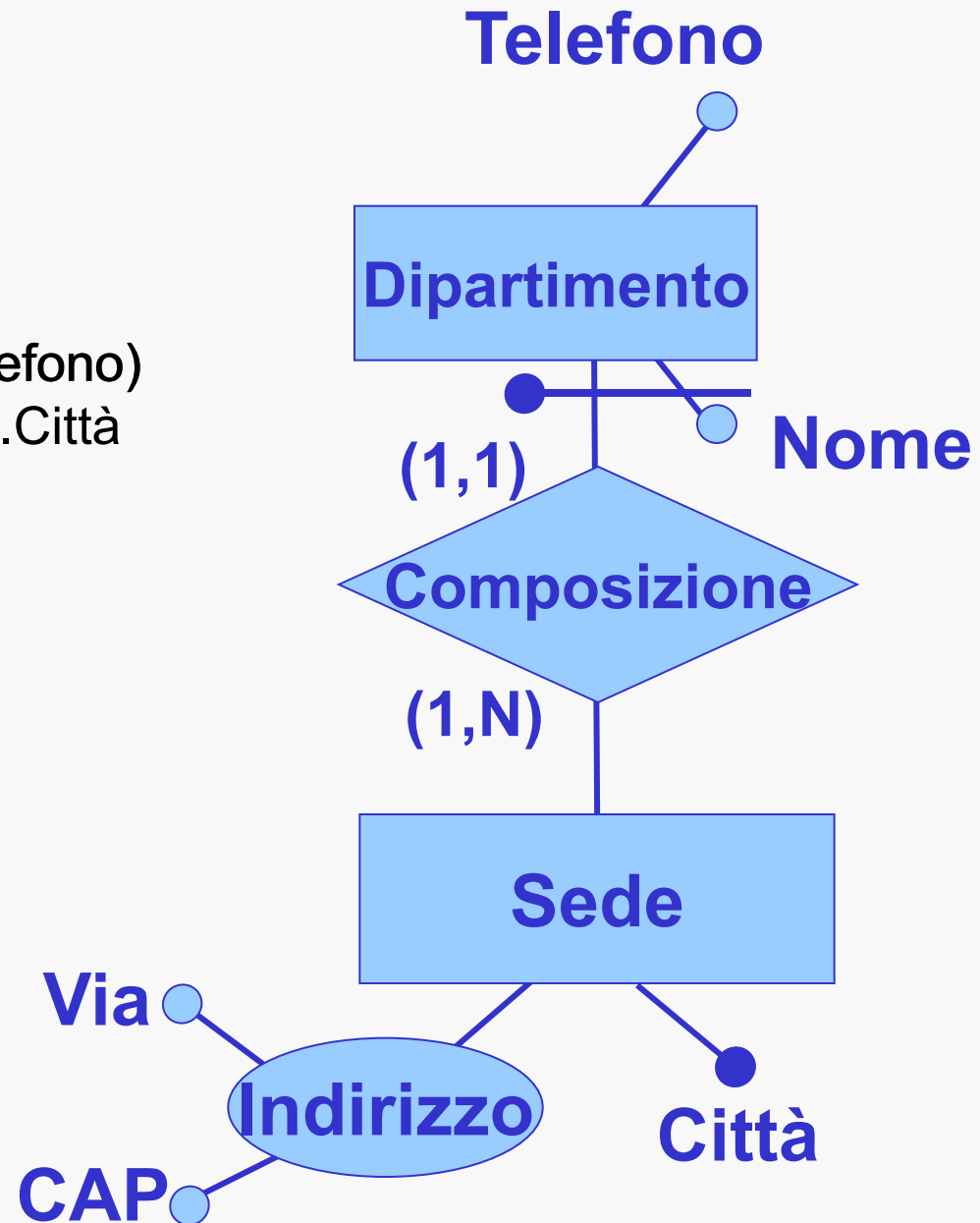
DI PROGETTAZIONE LOGICA

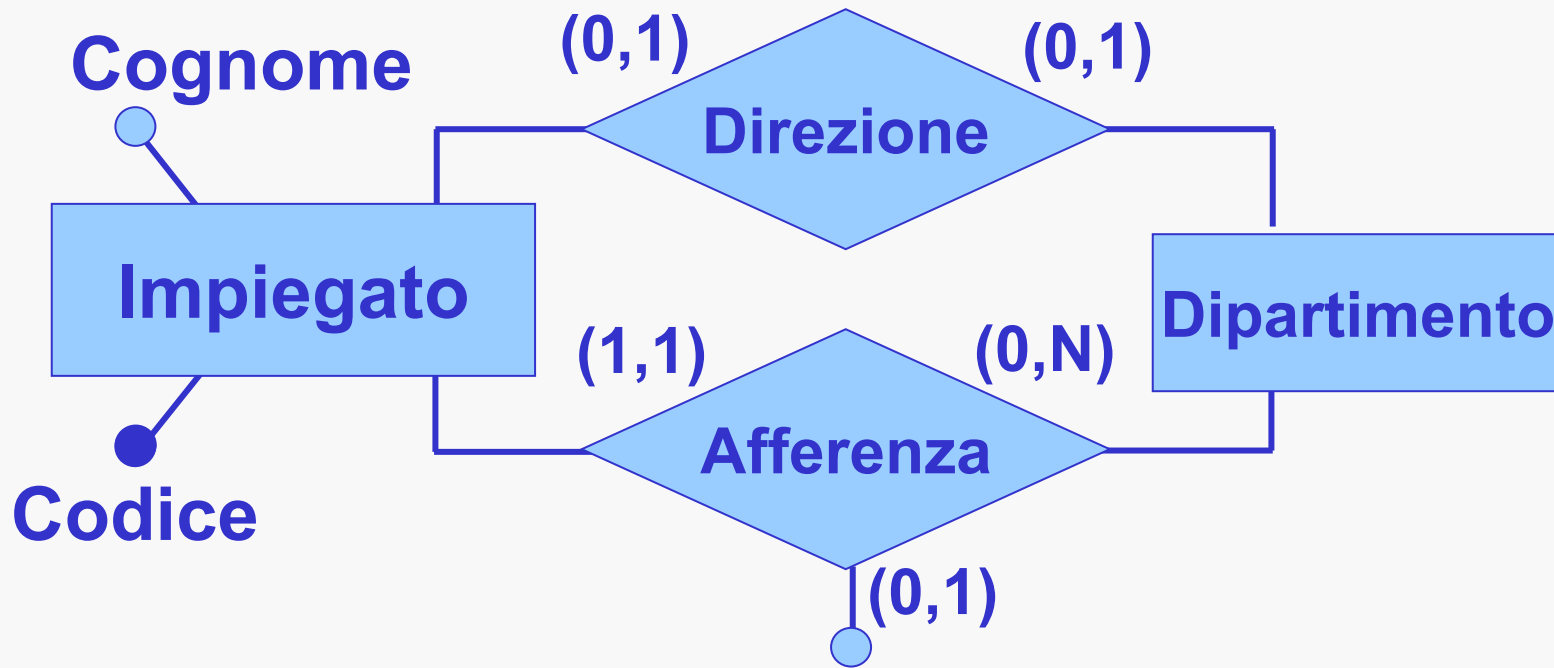


Sede(Città, Via, CAP)

Dipartimento(Nome, Città-Sede, Telefono)

- Dipartimento.Città-Sede → Sede.Città



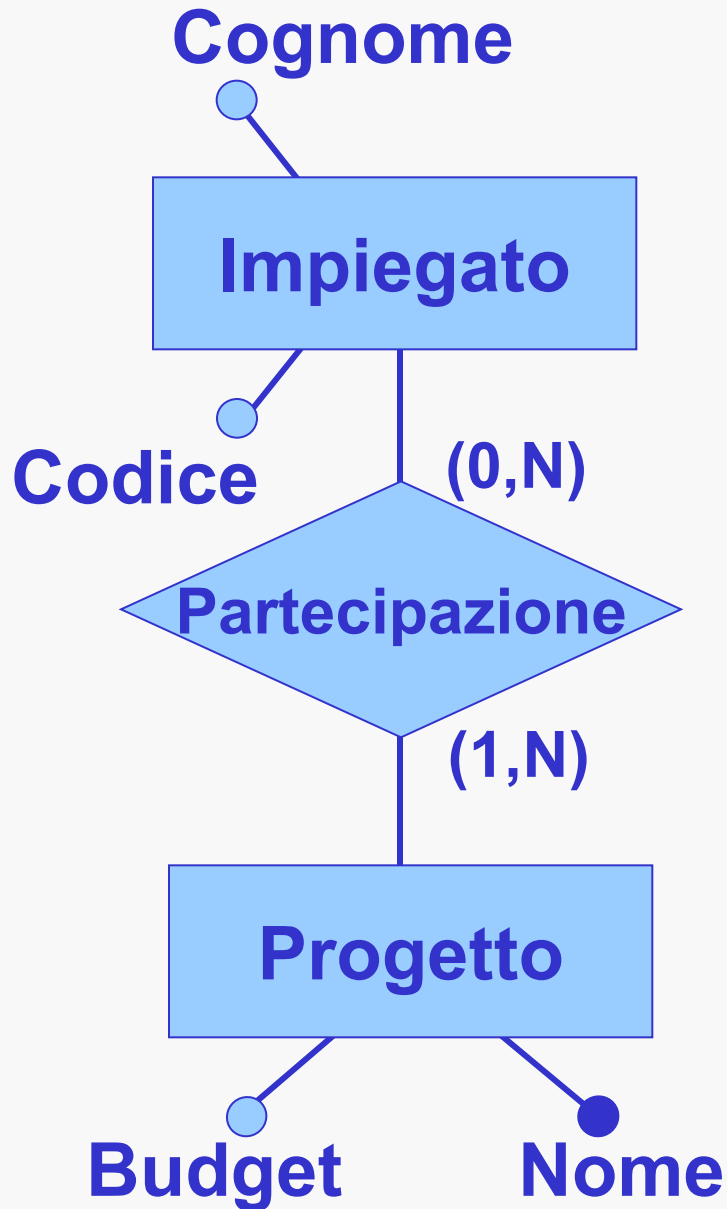


Dipartimento(Nome, Città, Sede, Telefono, Direttore)

- Dipartimento.Direttore \rightarrow Impiegato.Codice
- Direttore può essere NULL.

Impiegato(Codice, Cognome, DipartAffer, Sede, Data)

- Impiegato.(DipartAffer, Sede) \rightarrow
Dipartimento.(Nome,Città_Sede)
- Data può essere NULL



Impiegato(Codice, Cognome, DipartAffer, Sede, Data)

Progetto(Nome, Budget)

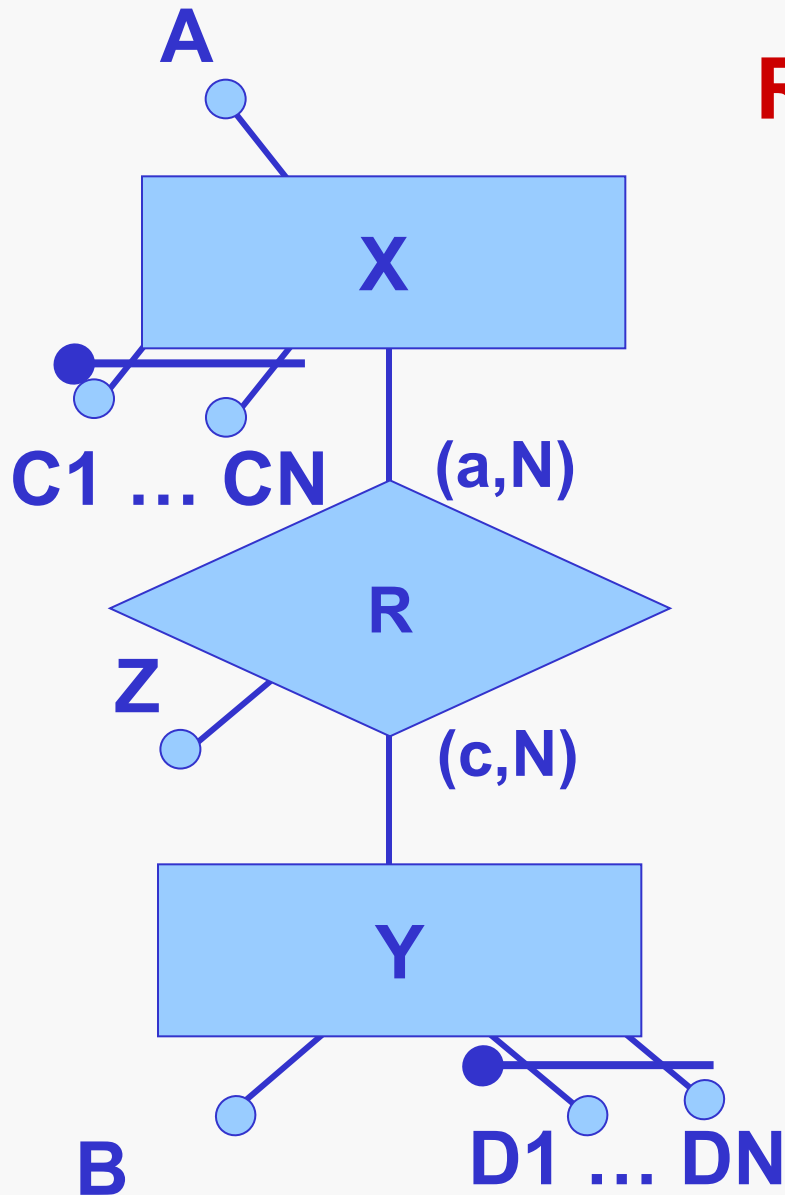
Partecipazione(Impiegato, Progetto)

- Partecipazione.Impiegato → Impiegato.Codice
- Partecipazione.Progetto → Progetto.Nome

Schema finale

- Impiegato(Codice, Cognome, DipartAffer, Sede, Data)
 - Impiegato.(DipartAffer,Sede) → Dipartimento.(Nome,Citta__Sede)
 - Data può essere NULL
- Dipartimento(Nome, Città Sede, Telefono, Direttore)
 - Dipartimento.Direttore → Impiegato.Codice
 - Direttore può essere NULL.
- Sede(Città, Via, CAP)
- Progetto(Nome, Budget)
- Partecipazione(Impiegato, Progetto)
 - Partecipazione.Impiegato → Impiegato.Codice
 - Partecipazione.Progetto → Progetto.Nome

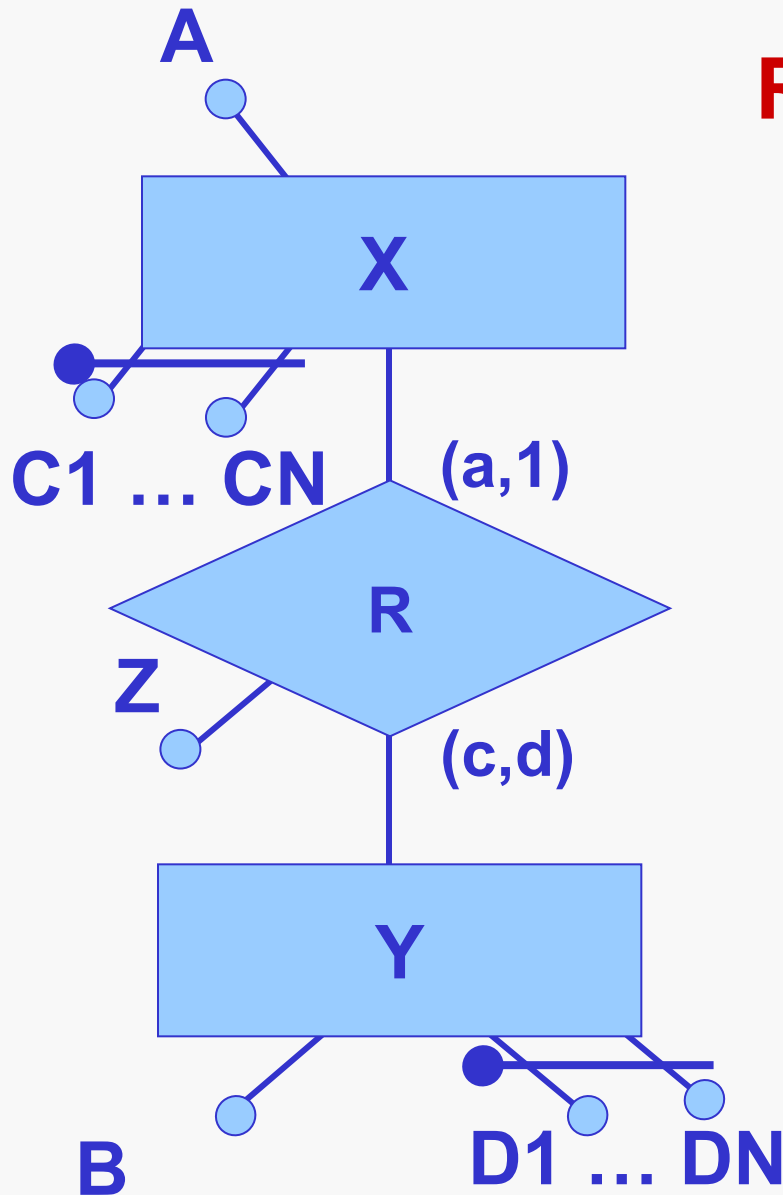
Riassunto della traduzione



$a \in \{0, 1\}, c \in \{0, 1\}$

- $X(\underline{C1}, \dots, \underline{CN}, A)$
- $Y(\underline{D1}, \dots, \underline{DN}, B)$
- $R(\underline{C1}, \dots, \underline{CN}, \underline{D1}, \dots, \underline{DN}, Z)$
 - $R.(C1, \dots, CN) \rightarrow A.(C1, \dots, CN)$
 - $R.(D1, \dots, DN) \rightarrow Y.(D1, \dots, DN)$

Riassunto della traduzione

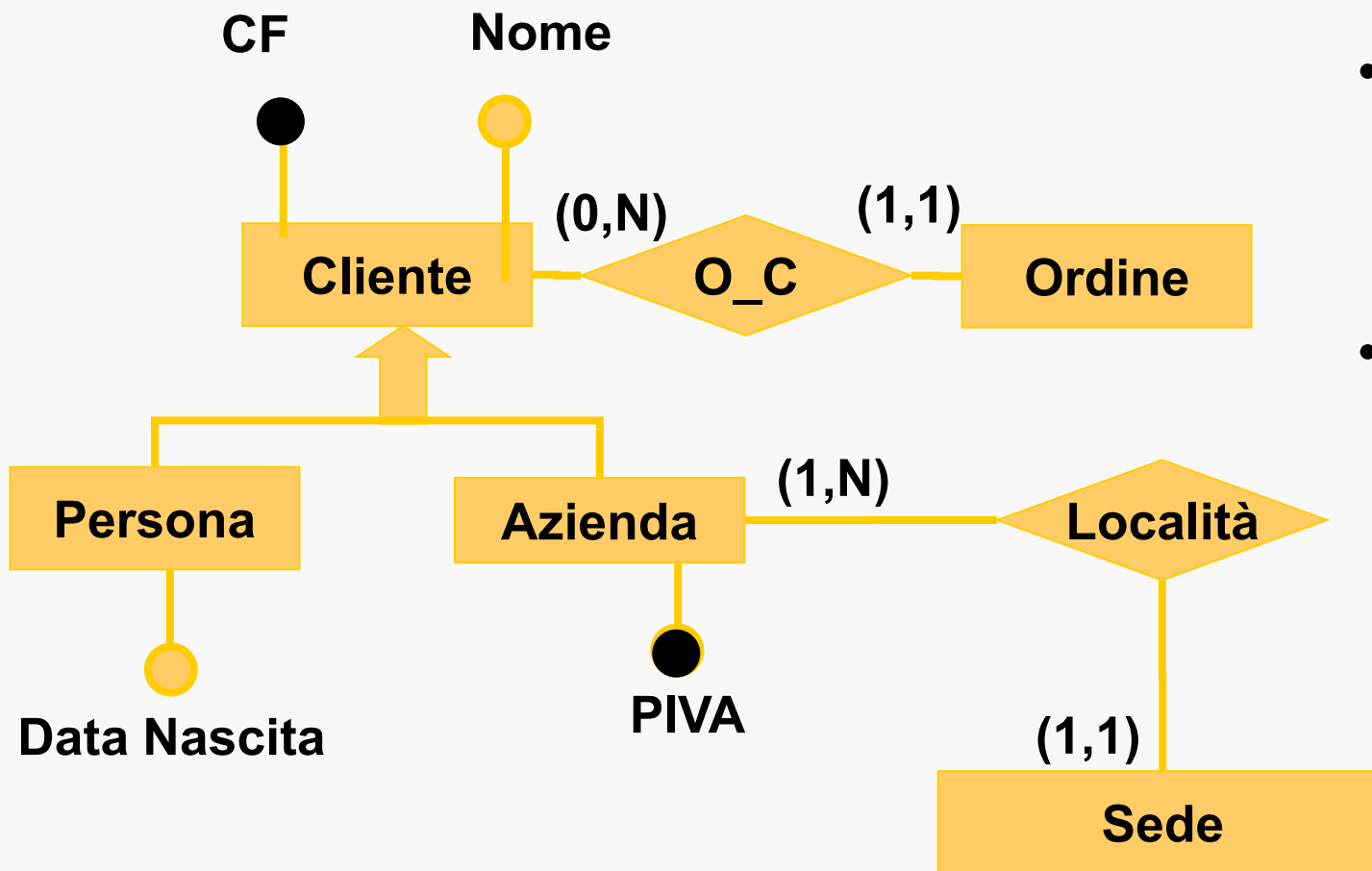


$a \in \{0,1\}, c \in \{0,1\}, d \in \{1,N\}$

- $X(\underline{C1}, \dots, \underline{CN}, A, \underline{D1}, \dots, \underline{DN}, Z)$
 - $X.(D1, \dots, DN) \rightarrow Y.(D1, \dots, DN$
- $Y(\underline{D1}, \dots, \underline{DN}, B)$
- Se $a=0$, gli attributi $D1, \dots, DN, Z$ ammettono valori nulli

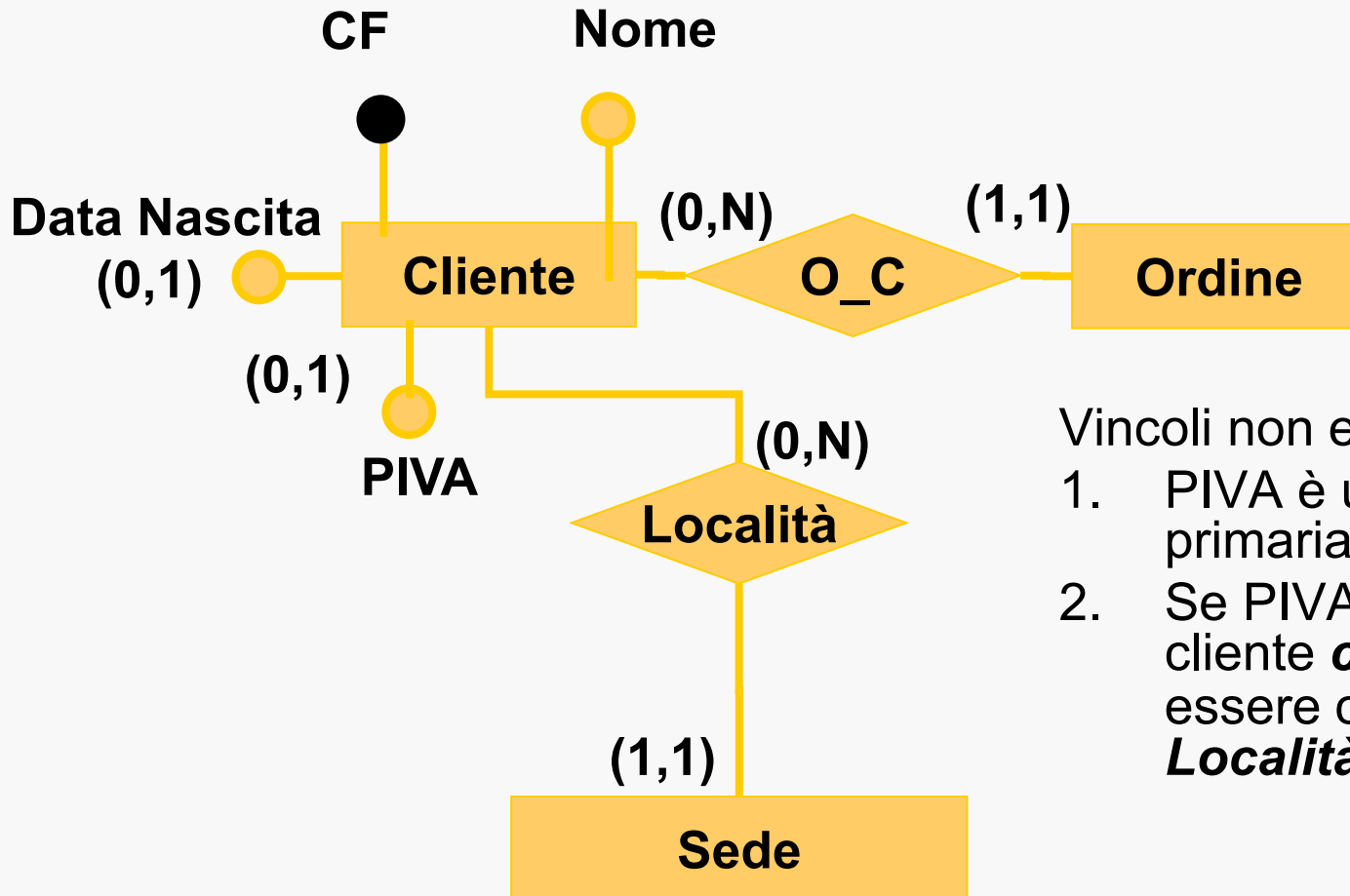
GENERALIZZAZIONE CON IDENTIFICATORI DIVERSI: UN ESEMPIO

Un figlio ha identificatore diverse dal padre



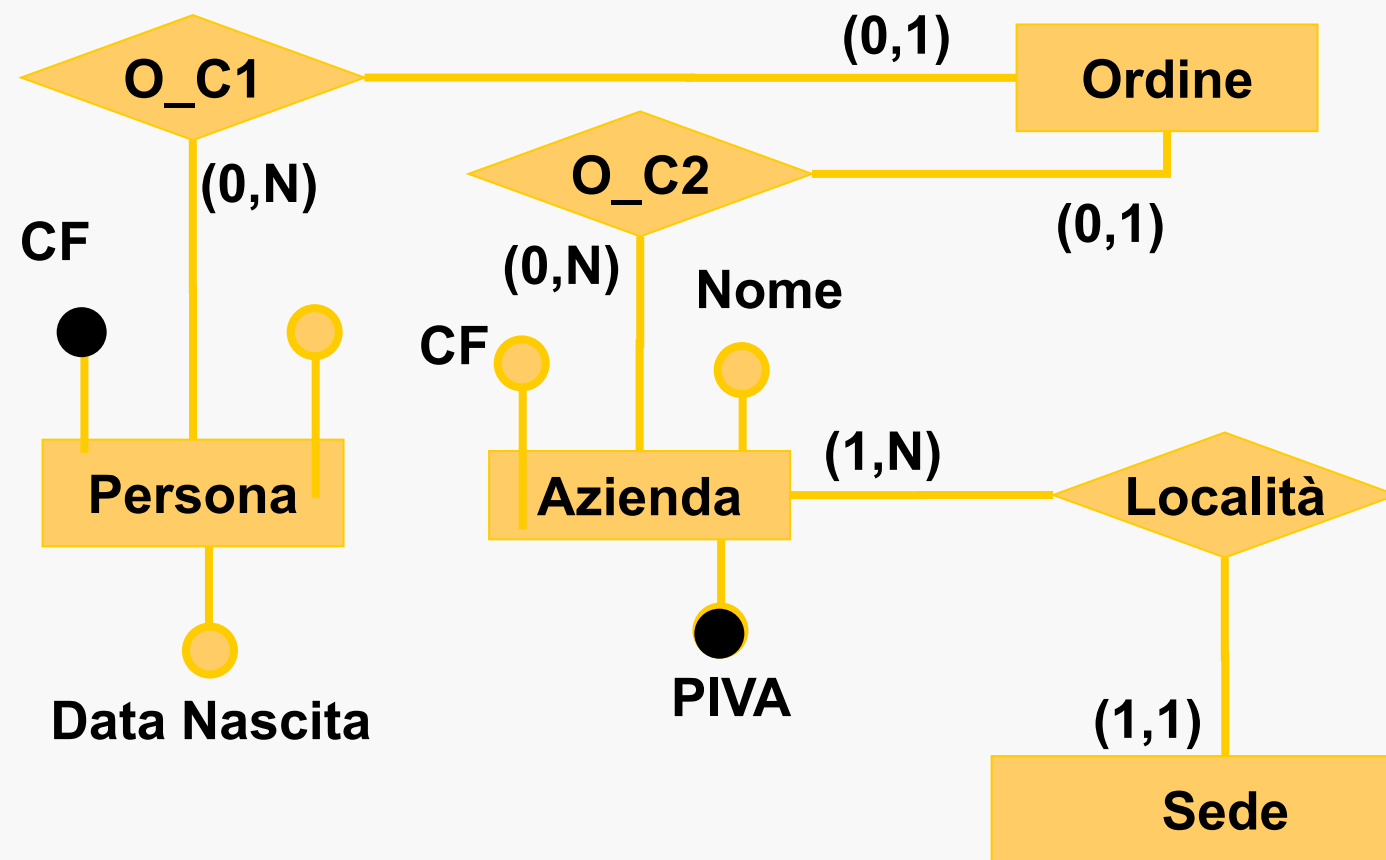
- *Azienda* ha un identificatore diverso da *Cliente*
- Comunque non ci possono essere due aziende con lo stesso CF

Possibilità di ristrutturazione 1: Accorpamento dei figli nel padre



- Vincoli non esprimibili in ER:
1. PIVA è una chiave (non primaria)
 2. Se PIVA è NULL per un cliente c , non ci possono essere coppie $(c,s) \in \text{Località}$

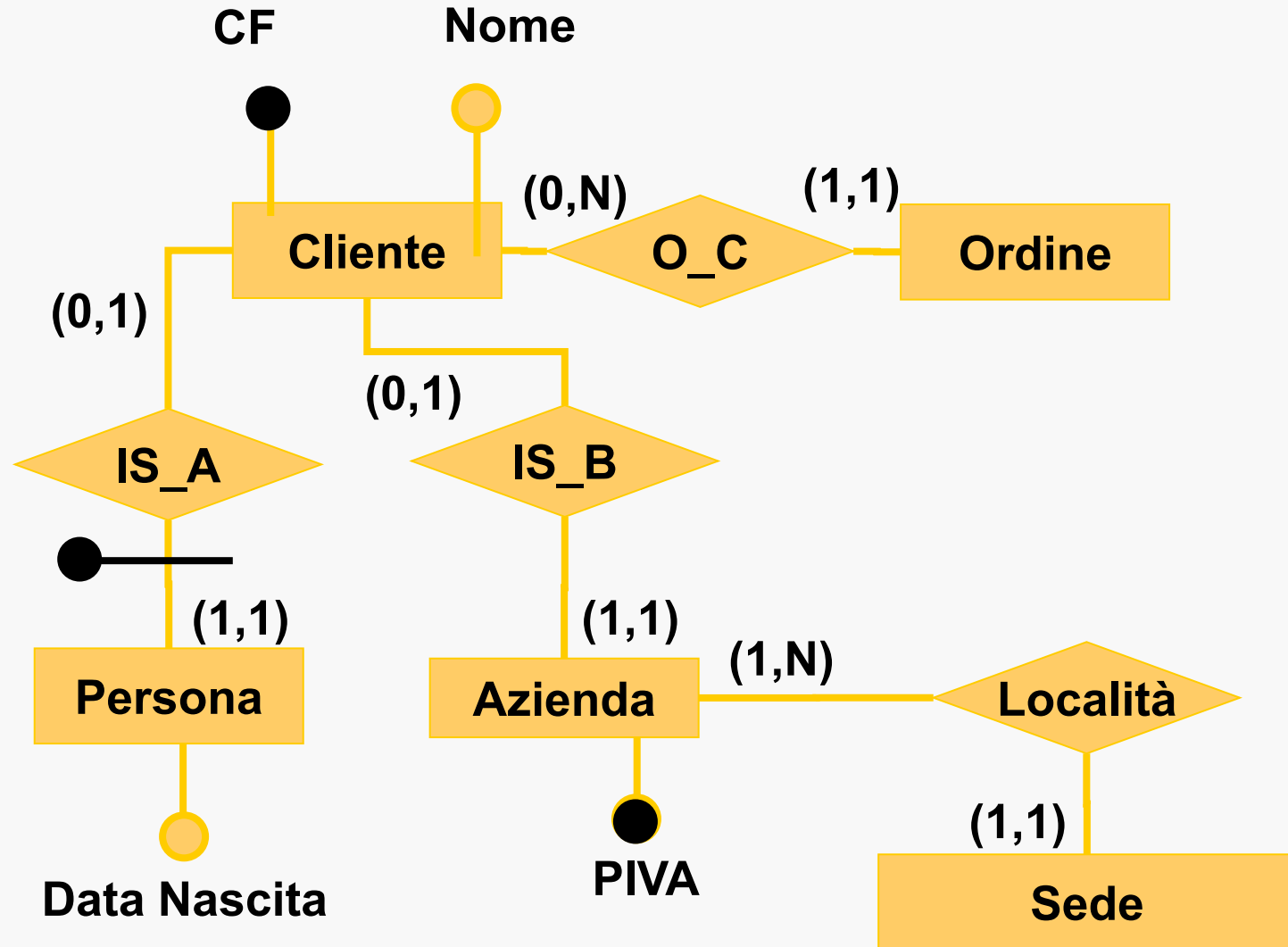
Possibilità di ristrutturazione 2: Accorpamento del padre nei figli



Vincoli non esprimibili in ER:

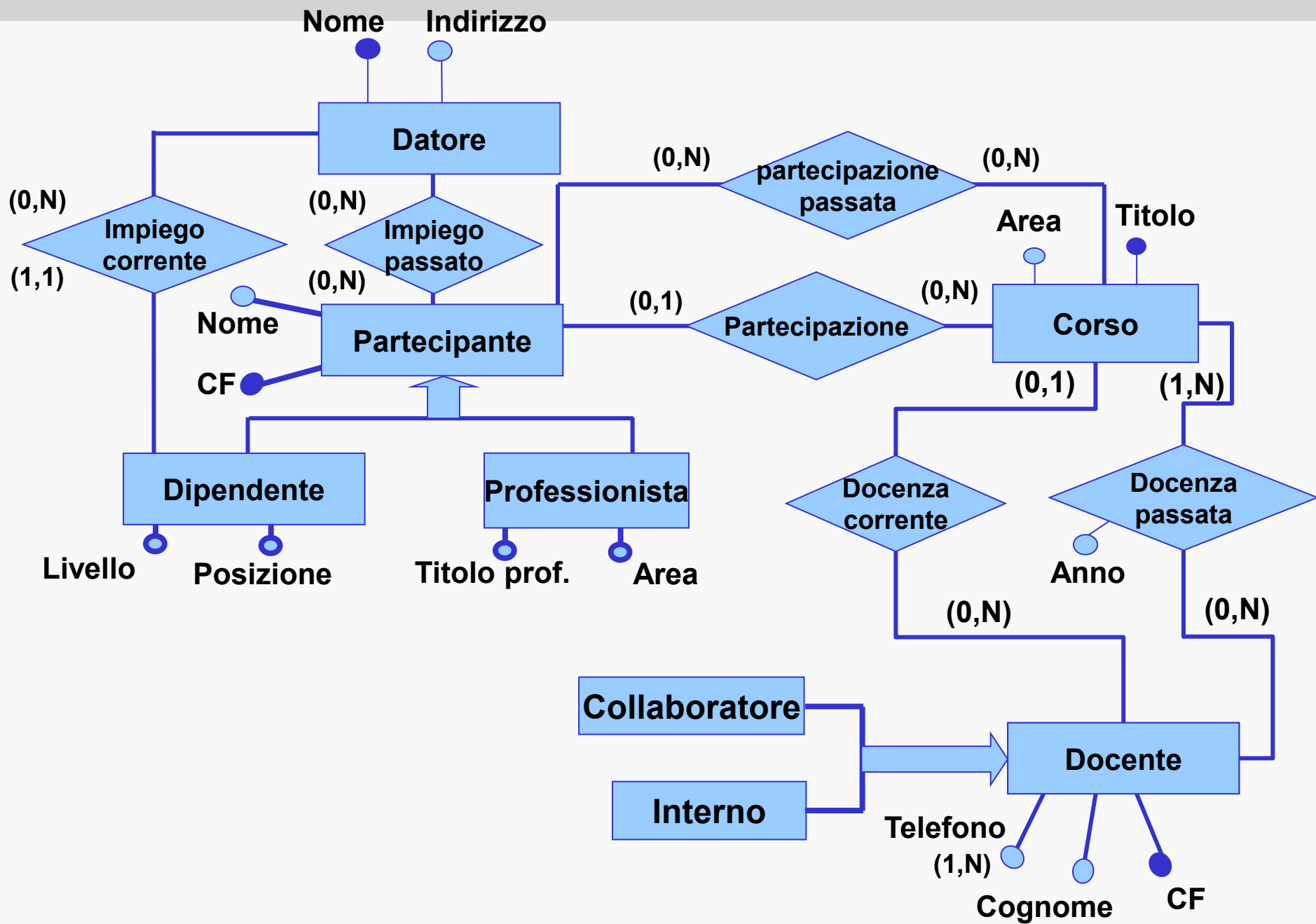
1. CF è chiave (non primaria) di Azienda
2. Un ordine deve "coinvolgere" una tra *Persona* e *Azienda*

Possibilità di ristrutturazione 3

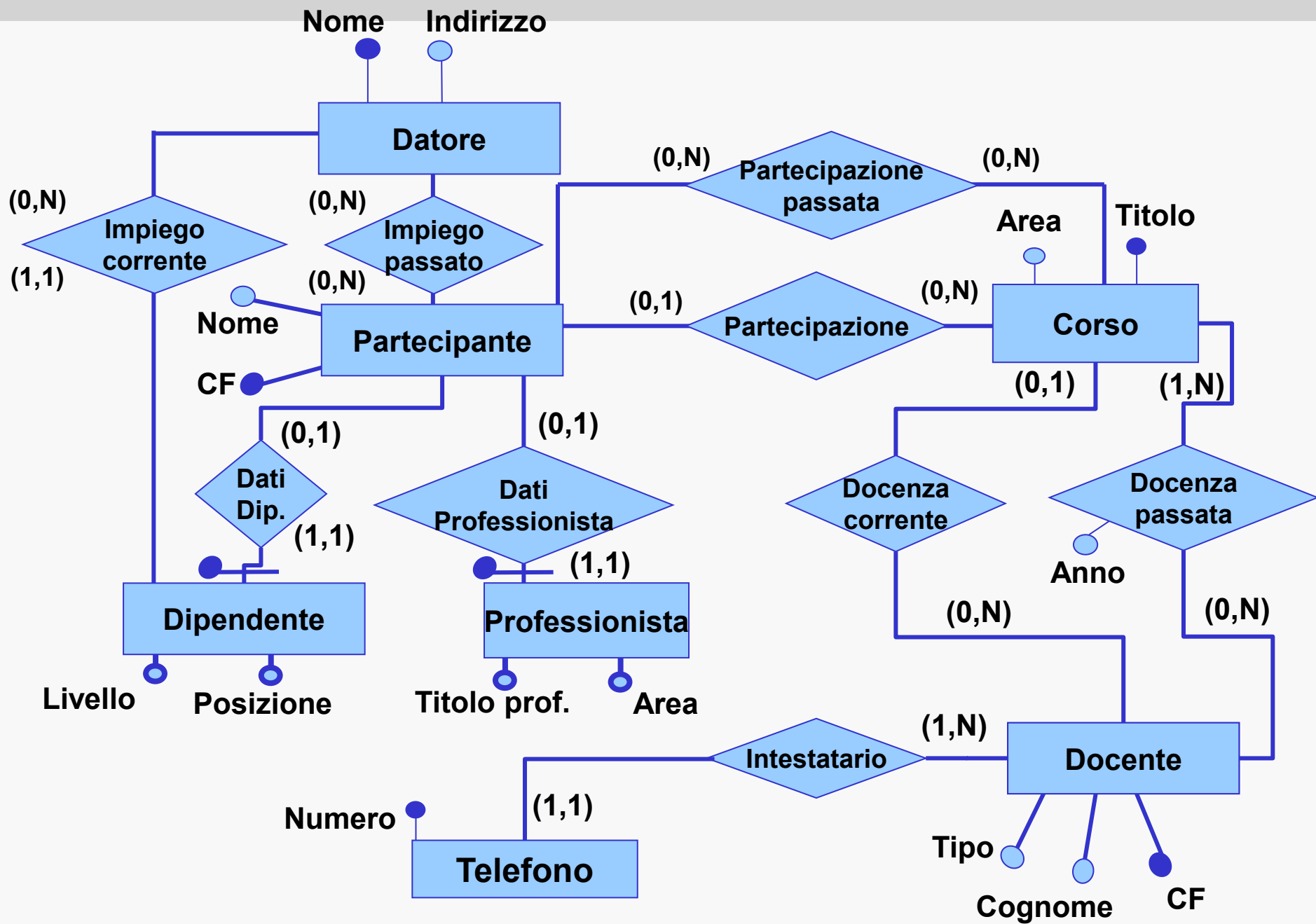


ESERCIZIO 2

DI PROGETTAZIONE LOGICA



**SOLUZIONE CHE MINIMIZZA I
VALORI NULLI**



Schema finale / 1

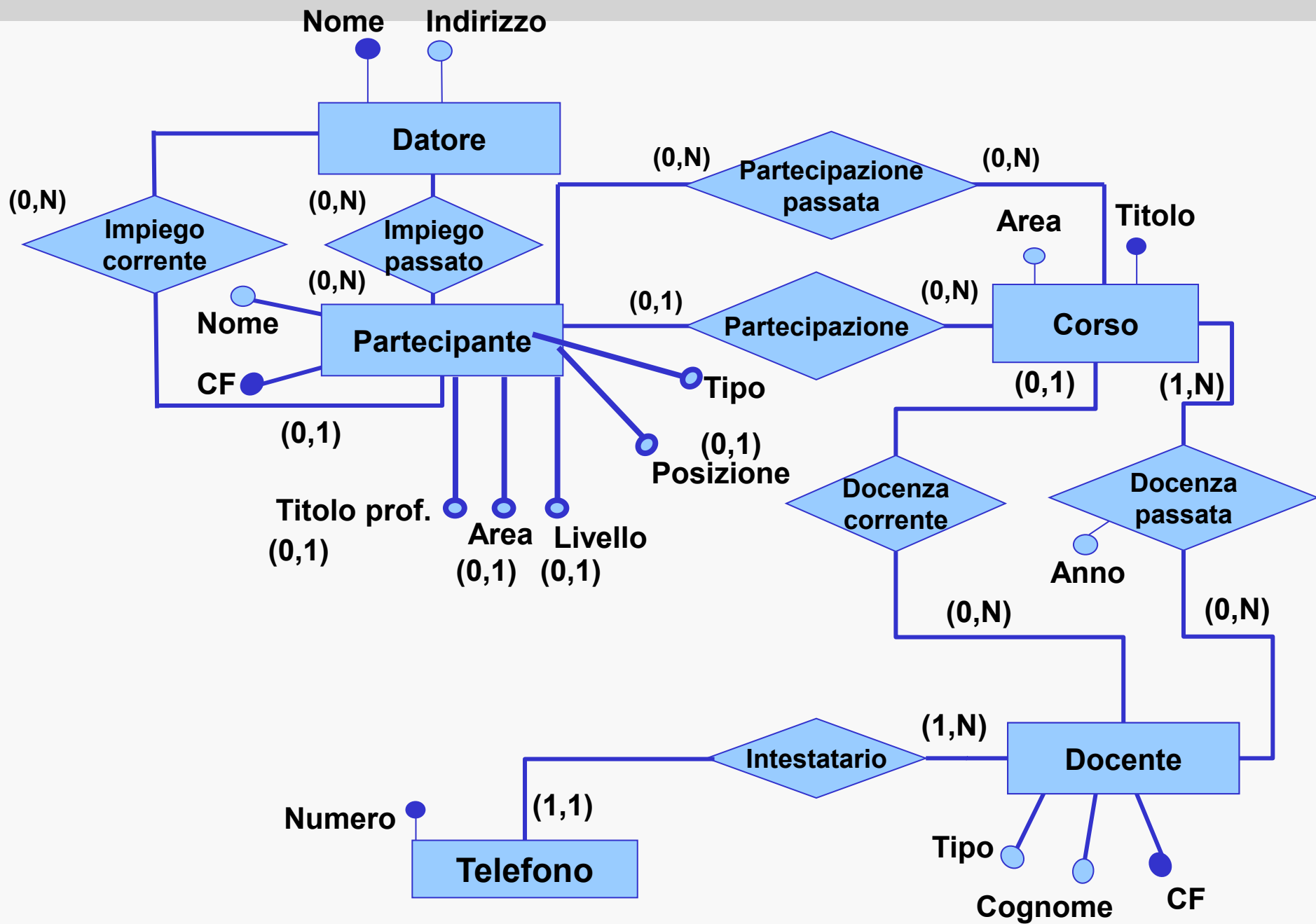
- Partecipante(CF, Nome, **Corso-Attuale**)
 - Partecipante.Corso-Attuale → Corso.Titolo
- Datore(Nome, Indirizzo)
- Corso(Titolo, Area, **CF-Docente-Corrente**)
 - Corso.CF-Docente-Corrente → Docente.CF
- Docente(CF, Cognome, Tipo)
 - Docente.Tipo ∈ { Collaboratore, Interno }
- Telefono(Numero, CF-Docente)
 - Telefono.CF-Docente → Docente.CF
- Dipendente(CF, Livello, Posizione, Datore)
 - Dipendente.CF → Partecipante.CF
 - Dipendente.Datore → Datore.Nome
- Professionista(CF, Titolo-Prof, Area)
 - Professionista.CF → Partecipante.CF

Gli attributi contrassegnati in rosso possono avere valori nulli

Schema finale / 2

- ImpiegoPassato(CF-Partecipante,Nome-Datore)
 - ImpiegoPassato.CF-Partecipante → Partecipante.CF
 - ImpiegoPassato.Nome-Datore → Datore.Nome
- PartecipazionePassata(CF-Partecipante,Titolo-Corso)
 - PartecipazionePassata.CF-Partecipante → Partecipante.CF
 - PartecipazionePassata.Titolo-Corso → Corso.Titolo
- DocenzaPassata(CF-Docente,Titolo)
 - DocenzaPassata.CF-Docente → CF.Docente
 - DocenzaPassata.Titolo → Corso.Titolo

**SOLUZIONE CHE RIDUCE IL
NUMERO DI TABELLE**



Schema finale / 1

- Partecipante(CF, Nome, Tipo,
Corso-Attuale, Titolo-Prof, Area, Posizione, Datore)
 - Tipo="Dipendente" \Leftrightarrow
Livello IS NOT NULL \wedge Posizione IS NOT NULL \wedge
Area IS NULL \wedge Datore IS NULL \wedge Titolo-Prof IS NULL
 - Tipo="Professionista" \Leftrightarrow
Livello IS NULL \wedge Posizione IS NULL \wedge
Area IS NOT NULL \wedge Datore IS NULL \wedge Titolo-Prof IS NOT NULL
 - Partecipante.Corso-Attuale \rightarrow Corso.Titolo
 - Partecipante.Datore \rightarrow Datore.Nome
- Datore(Nome, Indirizzo)
- Corso(Titolo, Area, CF-Docente-Corrente)
 - Corso.CF-Docente-Corrente \rightarrow Docente.CF
- Docente(CF, Cognome, Tipo)
 - Docente.Tipo \in { Collaboratore, Interno }
- Telefono(Numero, CF-Docente)
 - Telefono.CF-Docente \rightarrow Docente.CF

- I vincoli seguenti possono essere implementati con dei CHECK
 - Tipo="Dipendente" \Leftrightarrow
 Livello IS NOT NULL \wedge Posizione IS NOT NULL \wedge
 Area IS NULL \wedge Datore IS NULL \wedge Titolo-Prof IS NULL
 - Tipo="Professionista" \Leftrightarrow
 Livello IS NULL \wedge Posizione IS NULL \wedge
 Area IS NOT NULL \wedge Datore IS NULL \wedge Titolo-Prof IS NOT NULL
- In questo caso:


```

CHECK (
    (Tipo="Dipendente" AND Livello IS NOT NULL AND Posizione
    IS NOT NULL AND Area IS NULL AND Titolo-Prof IS NULL AND
    Datore IS NULL)
    OR
    (Tipo="Professionista" AND Area IS NOT NULL AND
    Titolo-Prof IS NOT NULL AND Livello IS NULL AND Posizione
    IS NULL Datore IS NOT NULL)
)
```

Schema finale / 2

- ImpiegoPassato(CF-Partecipante,Nome-Datore)
 - ImpiegoPassato.CF-Partecipante → Partecipante.CF
 - ImpiegoPassato.Nome-Datore → Datore.Nome
- PartecipazionePassata(CF-Partecipante,Titolo-Corso)
 - PartecipazionePassata.CF-Partecipante → Partecipante.CF
 - PartecipazionePassata.Titolo-Corso → Corso.Titolo
- DocenzaPassata(CF-Docente,Titolo)
 - DocenzaPassata.CF-Docente → CF.Docente
 - DocenzaPassata.Titolo → Corso.Titolo

Riferimenti

- Capitolo 8, escluso 8.6