

# Procedura un po' più generale, esempio

- R (CodCorso, Anno, Titolo, Crediti, Docente, NomeDoc, Semestre, Dipartimento, Indirizzo)
  - CodCorso → Titolo, Crediti, Dipartimento
  - CodCorso, Anno → Docente, Semestre
  - Docente → NomeDoc
  - Dipartimento → Indirizzo
- Chiave?
  - CodCorso, Anno

# Verifica della chiave

R (CodCorso, Anno, Titolo, Cred, Doc, NomeDoc, Sem, Dipart, Ind)

CodCorso → Titolo, Cred, Dipart

CodCorso, Anno → Doc, Sem

Doc → NomeDoc

Dipart → Ind

- CodCorso, Anno è chiave?

CodCorso, Anno

CodCorso, Anno, Titolo, Cred, Dipart

CodCorso, Anno, Titolo, Cred, Dipart, Doc, Sem

CodCorso, Anno, Titolo, Cred, Dipart, Doc, Sem, NomeDoc

CodCorso, Anno, Titolo, Cred, Dipart, Doc, Sem, NomeDoc, Ind

**Sì!**

# Variante nell'ordine delle dipendenze

R (CodCorso, Anno, Titolo, Cred, Doc, NomeDoc, Sem, Dipart, Ind)

Doc → NomeDoc

Dipart → Ind

CodCorso → Titolo, Cred, Dipart

CodCorso, Anno → Doc, Sem

- CodCorso, Anno è chiave?

CodCorso, Anno

CodCorso, Anno, Titolo, Cred, Dipart

CodCorso, Anno, Titolo, Cred, Dipart, Doc, Sem

CodCorso, Anno, Titolo, Cred, Dipart, Doc, Sem, NomeDoc

CodCorso, Anno, Titolo, Cred, Dipart, Doc, Sem, NomeDoc, Ind

**Sì!**

## Ricordiamo la definizione di chiave

- Insieme di attributi che identifica univocamente le ennuple ed è minimale rispetto a questa proprietà
- Per verificare che **CodCorso Anno** è chiave debbo verificare che
  - **CodCorso** da solo non è chiave
  - **Anno** da solo non è chiave

# Un insieme che non è chiave

R (CodCorso, Anno, Titolo, Cred, Doc, NomeDoc, Sem, Dipart, Ind)

CodCorso → Titolo, Cred, Dipart

CodCorso, Anno → Doc, Sem

Doc → NomeDoc

Dipart → Ind

- CodCorso è chiave?

CodCorso

CodCorso, Titolo, Cred, Dipart

CodCorso, Titolo, Cred, Dipart, Ind

**No!**

# Attenzione

- La verifica della chiave è semplice
- La ricerca non è semplice:
  - si può procedere per tentativi
  - conviene "intuire" e verificare

# Procedura un po' più generale, esempio

- R (CodCorso, Anno, Titolo, Crediti, Docente, NomeDoc, Semestre, Dipartimento, Indirizzo)

CodCorso → Titolo, Crediti, Dipartimento

CodCorso, Anno → Docente, Semestre

Docente → NomeDoc

Dipartimento → Indirizzo

- R (CC, A, T, Cr, Do, N, S, Di, I)

CC → T, Cr, Di

CC, A → Do, S

Do → N

Di → I

Corso

Edizione del corso

Docente

Dipartimento

# Procedura un po' più generale, esempio

R (CC, A, T, Cr, Do, N, S, Di, I)

~~CC~~ → T, Cr, Di

✓ CC, A → Do, S

~~Do~~ → N

~~Di~~ → I

Corso

Edizione del corso

Docente

Dipartimento

R1 (CC, T, Cr, Di, I)

✓ CC → T, Cr, Di

~~Di~~ → I

R2 (CC, A, Do, N, S)

✓ CC, A → Do, S

~~Do~~ → N

R3 (Di, I)

Di → I

OK

R4 (CC, T, Cr, Di)

CC → T, Cr, Di

OK

R5 (Do, N)

Do → N

OK

R2 (CC, A, Do, S)

CC, A → Do, S

OK

## Non sempre così facile

| Impiegato | Progetto | Sede   |
|-----------|----------|--------|
| Rossi     | Marte    | Roma   |
| Verdi     | Giove    | Milano |
| Verdi     | Venere   | Milano |
| Neri      | Saturno  | Milano |
| Neri      | Venere   | Milano |

~~X~~ Impiegato → Sede  
~~⊗~~ Progetto → Sede

# Applichiamo la procedura

| Impiegato | Progetto | Sede   |
|-----------|----------|--------|
| Rossi     | Marte    | Roma   |
| Verdi     | Giove    | Milano |
| Verdi     | Venere   | Milano |
| Neri      | Saturno  | Milano |
| Neri      | Venere   | Milano |

| Impiegato | Progetto |
|-----------|----------|
| Rossi     | Marte    |
| Verdi     | Giove    |
| Verdi     | Venere   |
| Neri      | Saturno  |
| Neri      | Venere   |

| Progetto | Sede   |
|----------|--------|
| Marte    | Roma   |
| Giove    | Milano |
| Saturno  | Milano |
| Venere   | Milano |

**Impiegato → Sede**

**??**

- La procedura non va bene:
  - non "conserva" le dipendenze
- Proviamo l'altra tecnica (una relazione per ogni dipendenza)

# Una relazione per ogni dipendenza

| <b>Impiegato</b> | <b>Progetto</b> | <b>Sede</b>   |
|------------------|-----------------|---------------|
| <b>Rossi</b>     | <b>Marte</b>    | <b>Roma</b>   |
| <b>Verdi</b>     | <b>Giove</b>    | <b>Milano</b> |
| <b>Verdi</b>     | <b>Venere</b>   | <b>Milano</b> |
| <b>Neri</b>      | <b>Saturno</b>  | <b>Milano</b> |
| <b>Neri</b>      | <b>Venere</b>   | <b>Milano</b> |

| <b>Impiegato</b> | <b>Sede</b>   |
|------------------|---------------|
| <b>Rossi</b>     | <b>Roma</b>   |
| <b>Verdi</b>     | <b>Milano</b> |
| <b>Neri</b>      | <b>Milano</b> |

| <b>Progetto</b> | <b>Sede</b>   |
|-----------------|---------------|
| <b>Marte</b>    | <b>Roma</b>   |
| <b>Giove</b>    | <b>Milano</b> |
| <b>Saturno</b>  | <b>Milano</b> |
| <b>Venere</b>   | <b>Milano</b> |

## Proviamo a ricostruire

| Impiegato | Sede   |
|-----------|--------|
| Rossi     | Roma   |
| Verdi     | Milano |
| Neri      | Milano |

| Progetto | Sede   |
|----------|--------|
| Marte    | Roma   |
| Giove    | Milano |
| Saturno  | Milano |
| Venere   | Milano |

| Impiegato | Progetto | Sede   |
|-----------|----------|--------|
| Rossi     | Marte    | Roma   |
| Verdi     | Giove    | Milano |
| Verdi     | Venere   | Milano |
| Neri      | Saturno  | Milano |
| Neri      | Venere   | Milano |
| Verdi     | Saturno  | Milano |
| Neri      | Giove    | Milano |

**Diversa dalla relazione di partenza!**

# Decomposizione senza perdita

- Una relazione  $r$  si **decompone** su  $X_1$  e  $X_2$  **senza perdita di informazione** se il join delle proiezioni di  $r$  su  $X_1$  e  $X_2$  è uguale a  $r$  stessa (cioè non contiene ennuple spurie)
- Una buona decomposizione deve essere senza perdita

# Purtroppo

- L'esempio appena visto ci costringe a dire che la normalizzazione non sempre funziona:
  - vorremmo avere una decomposizione
    - in BCNF
    - senza perdita
    - che "conservi" le dipendenze
  - Non sempre è possibile:
    - ci sono realtà difficili da modellare
  - Anche per questo, usiamo la normalizzazione come strumento di verifica e non di progettazione

# Decomposizione senza perdita

- Approfondiamo

| <u>IMP</u> | <u>PROGETTO</u> | <u>SEDE</u> |
|------------|-----------------|-------------|
| Rossi      | MARTE           | ROMA        |
| Verdi      | GIOVE           | MILANO      |
| Neri       | GIOVE           | MILANO      |

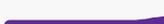
| <u>PROG SEDE</u> |        |   | <u>IMP SEDE</u> |        |
|------------------|--------|---|-----------------|--------|
| MARTE            | ROMA   | — | Rossi           | ROMA   |
| GIOVE            | MILANO | — | Verdi           | MILANO |
|                  |        | — | Neri            | MILANO |

Condizione sufficiente

- gli attributi comuni sono chiave per almeno una delle proiezioni

# Attenzione

- La condizione sufficiente non è necessaria

|   | IMP   | PROG   | SEDE   |
|---|-------|--------|--------|
|  | Rossi | GIOVE  | ROMA   |
|  | Neri  | MARTE  | MILANO |
|  | Neri  | VENERE | MILANO |
|  | Bruni | MARTE  | MILANO |
|  | Bruni | VENERE | MILANO |

| IMP   | SEDE   |  | PROG   | SEDE   |
|-------|--------|--|--------|--------|
| Rossi | ROMA   |  | GIOVE  | ROMA   |
| Neri  | MILANO |  | MARTE  | MILANO |
| Bruni | MILANO |  | VENERE | MILANO |

seconda condizione sufficiente

- le componenti delle eunuple sono indipendenti fra loro

nell'esempio

- ogni sede ha un insieme di impiegati.
- ogni sede ha un insieme di progetti.
- ogni impiegato lavora a tutti i progetti delle proprie sedi

# Riassumendo

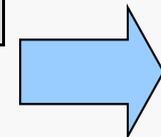
- La decomposizione senza perdita
  - è garantita se gli attributi comuni contengono una **chiave** per almeno una delle relazioni decomposte
  - Più in generale, si ha
    - quando le informazioni in  $X_1$  e  $X_2$  sono fra loro indipendenti

# Decomposizione senza perdita

| Impiegato | Progetto | Sede   |
|-----------|----------|--------|
| Rossi     | Marte    | Roma   |
| Verdi     | Giove    | Milano |
| Verdi     | Venere   | Milano |
| Verdi     | Giove    | Roma   |
| Verdi     | Venere   | Roma   |
| Bruni     | Saturno  | Milano |

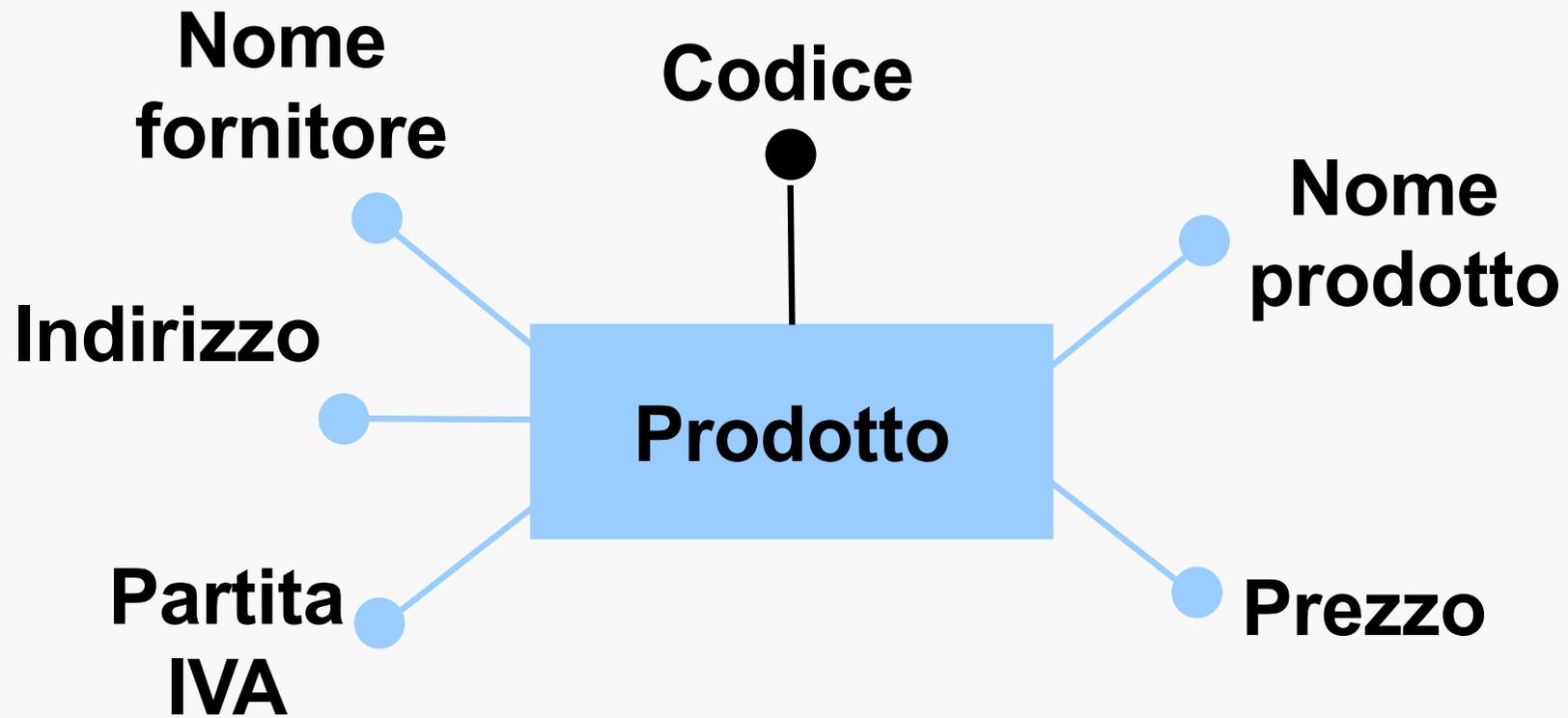
| Impiegato | Progetto |
|-----------|----------|
| Rossi     | Marte    |
| Verdi     | Giove    |
| Verdi     | Venere   |
| Bruni     | Saturno  |

| Impiegato | Sede   |
|-----------|--------|
| Rossi     | Roma   |
| Verdi     | Milano |
| Verdi     | Roma   |
| Bruni     | Milano |

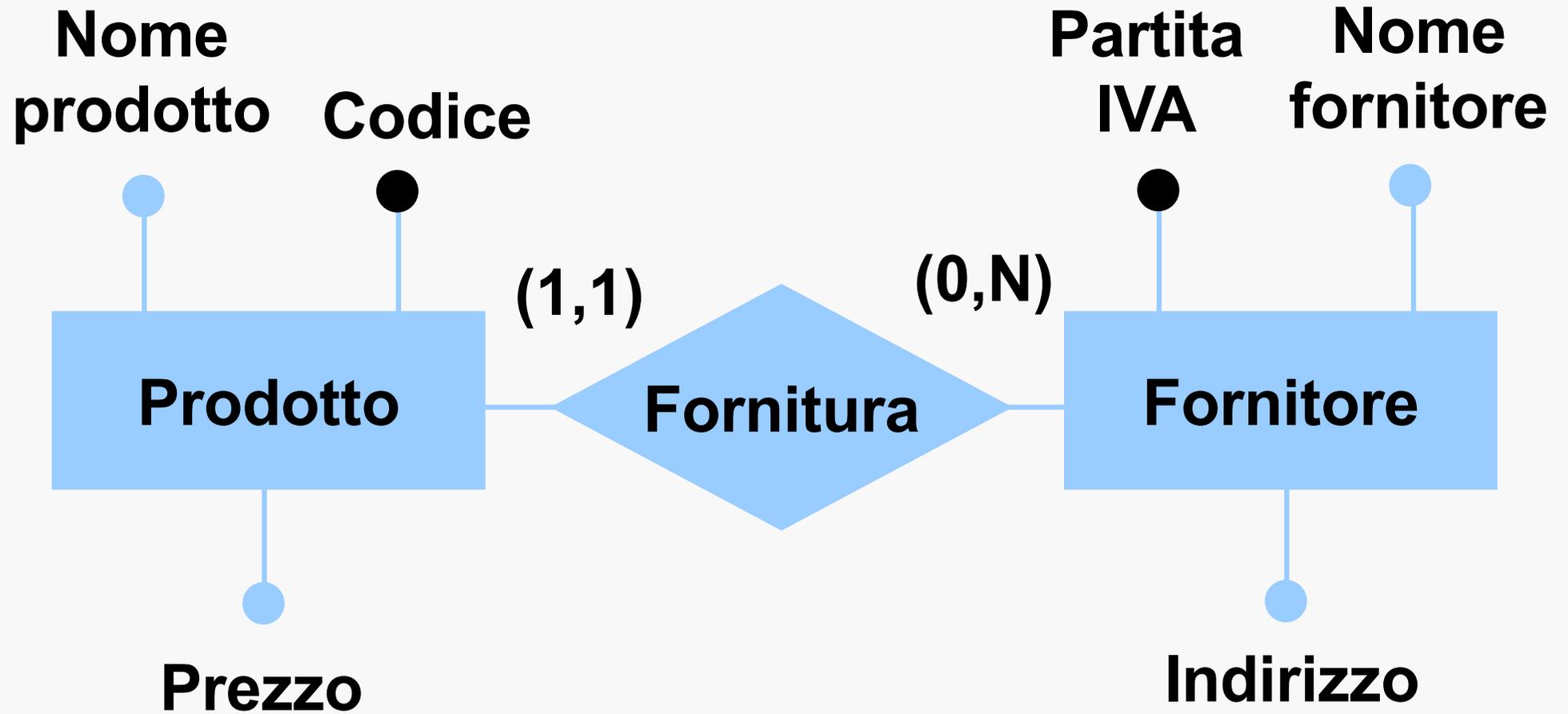


# Progettazione e normalizzazione

- La teoria della normalizzazione può essere usata nella progettazione logica per verificare lo schema relazionale finale
- Si può usare anche durante la progettazione concettuale per verificare la qualità dello schema concettuale



**PartitaIVA → NomeFornitore Indirizzo**



PRODOTTO

|               |        |        |       |        |     |
|---------------|--------|--------|-------|--------|-----|
| <u>CODICE</u> | NOME P | PREZZO | P.IVA | NOME F | IND |
|---------------|--------|--------|-------|--------|-----|

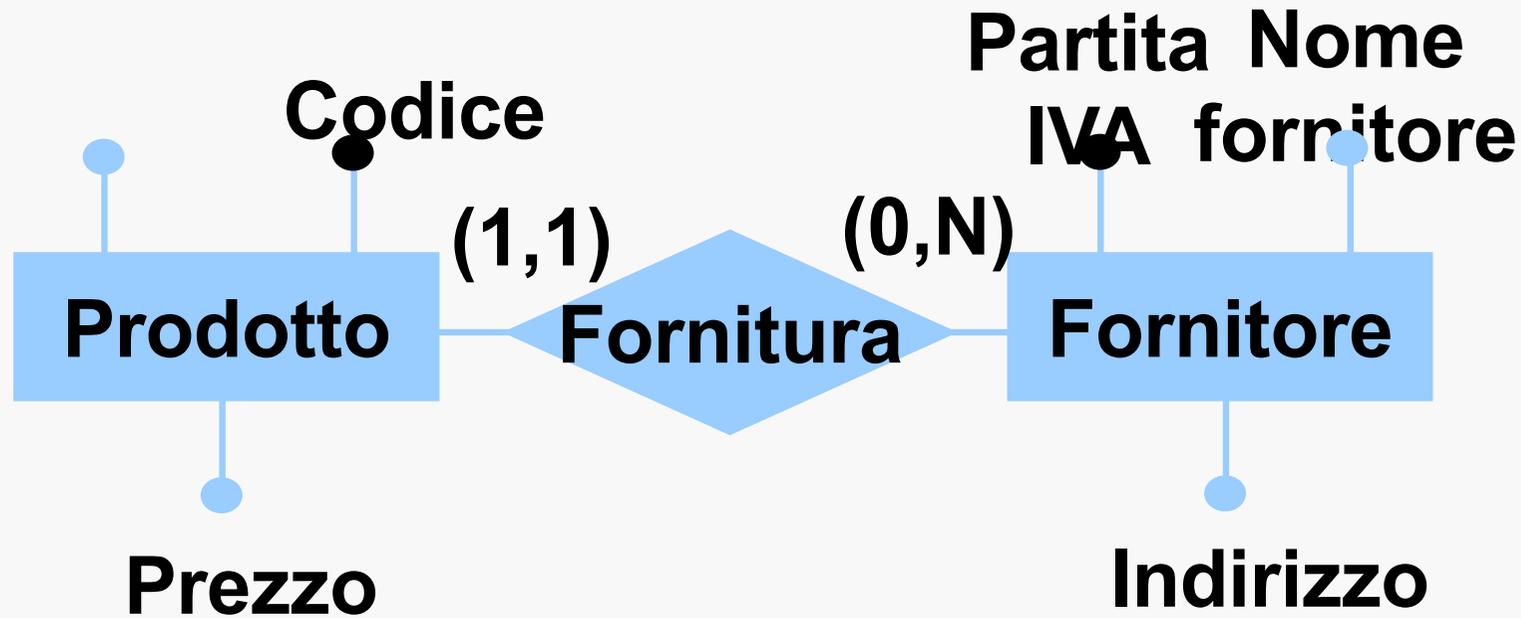
X P.IVA → NOME F, IND

PRODOTTO

|               |        |        |       |
|---------------|--------|--------|-------|
| <u>CODICE</u> | NOME P | PREZZO | P.IVA |
|---------------|--------|--------|-------|

FORNITORE

|       |        |     |
|-------|--------|-----|
| P.IVA | NOME F | IND |
|-------|--------|-----|

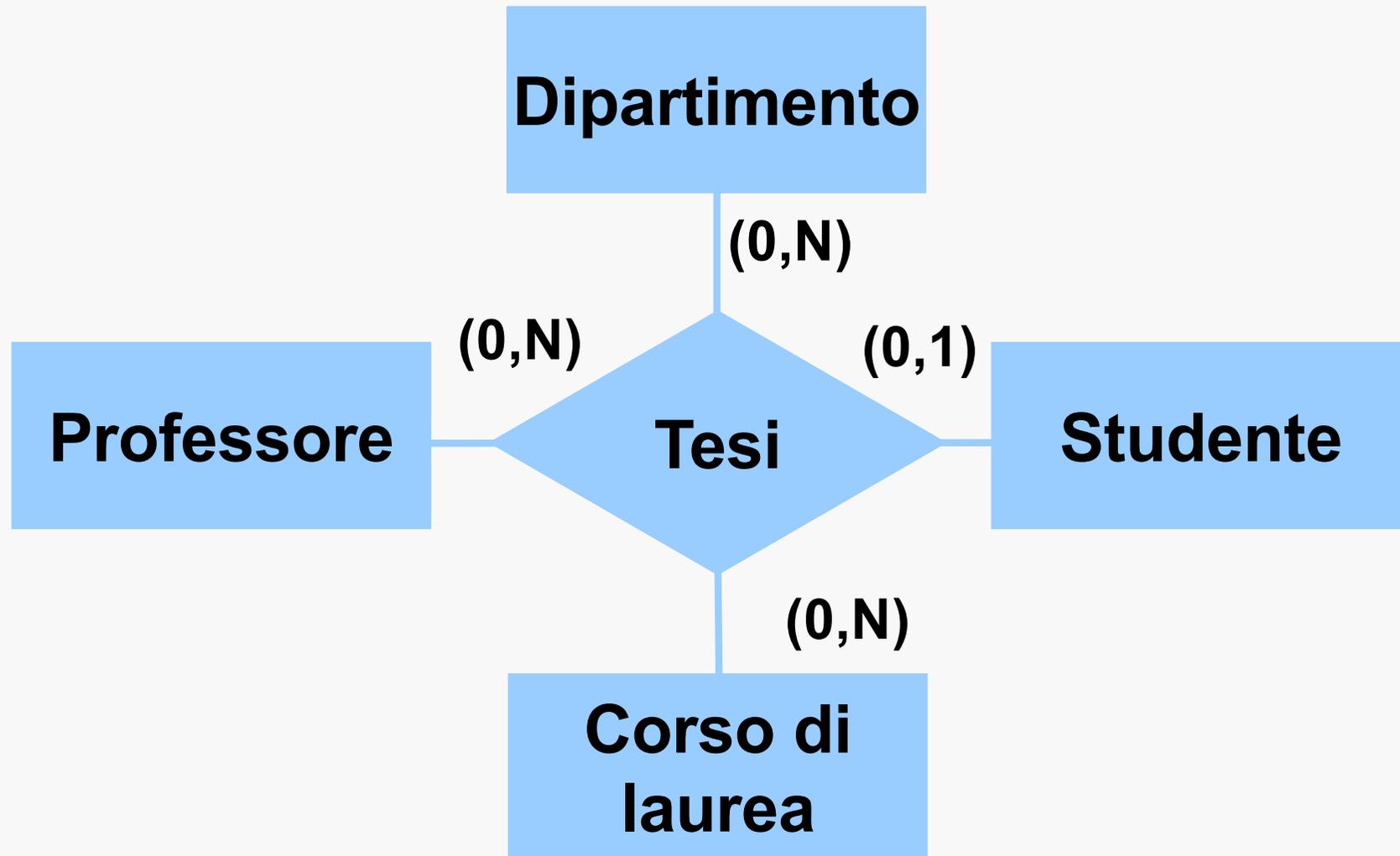


PRODOTTO

| <u>CODICE</u> | NOME P | PREZZO | P.IVA |
|---------------|--------|--------|-------|
|               |        |        |       |

FORNITORE

| <u>P.IVA</u> | NOME F | IND |
|--------------|--------|-----|
|              |        |     |



Assumiamo che TESI coinvolga

S C P D quando

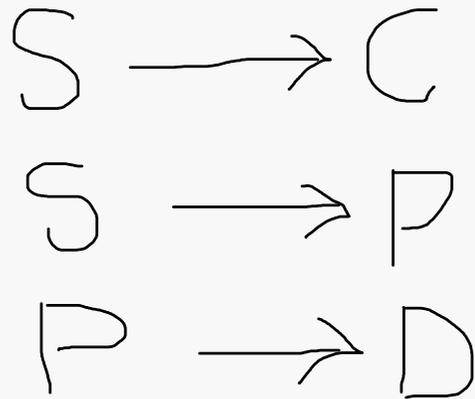
S svolge una tesi per il

CdL C con il prof P

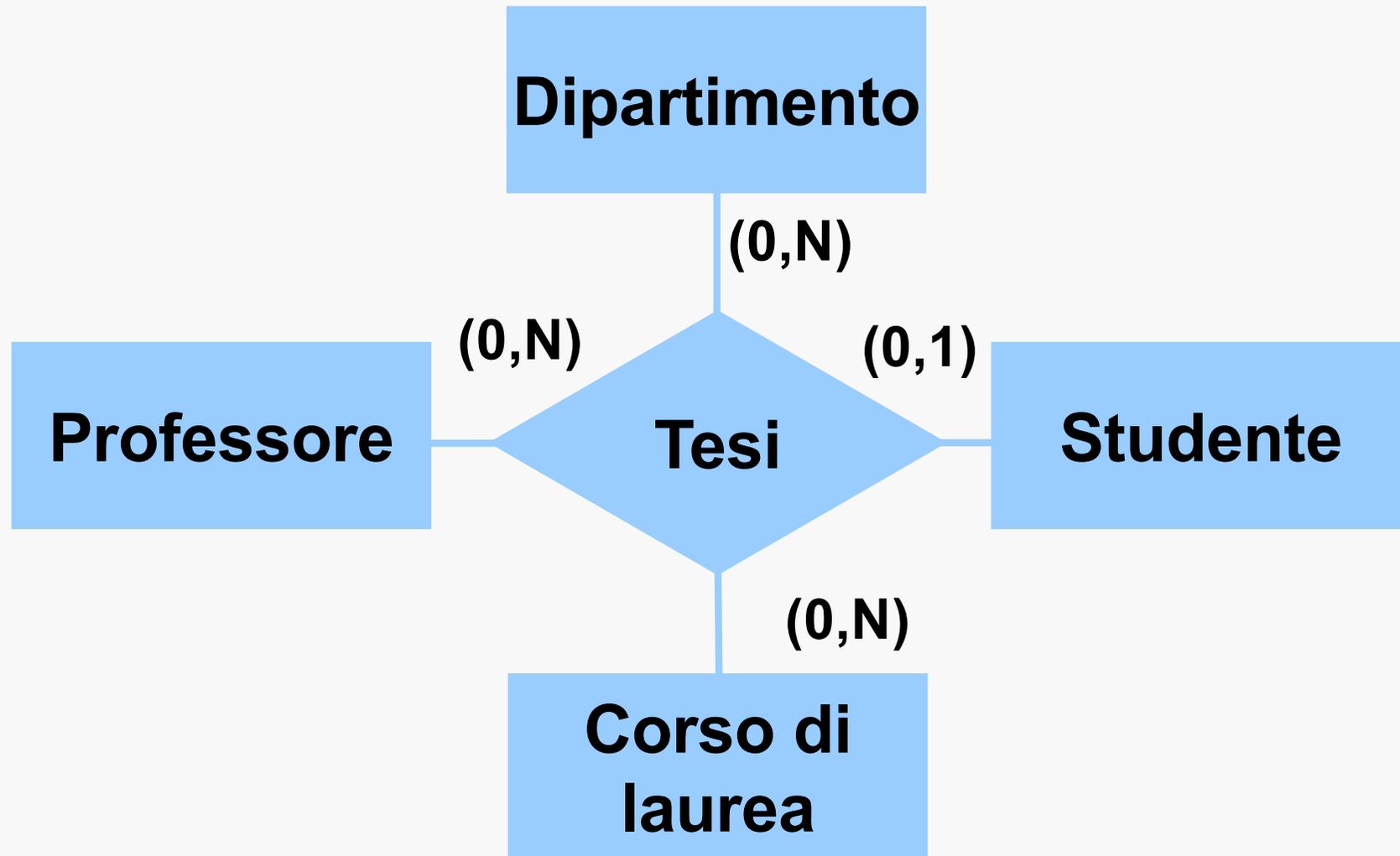
presso il dip. D (che è

il dip del prof P)

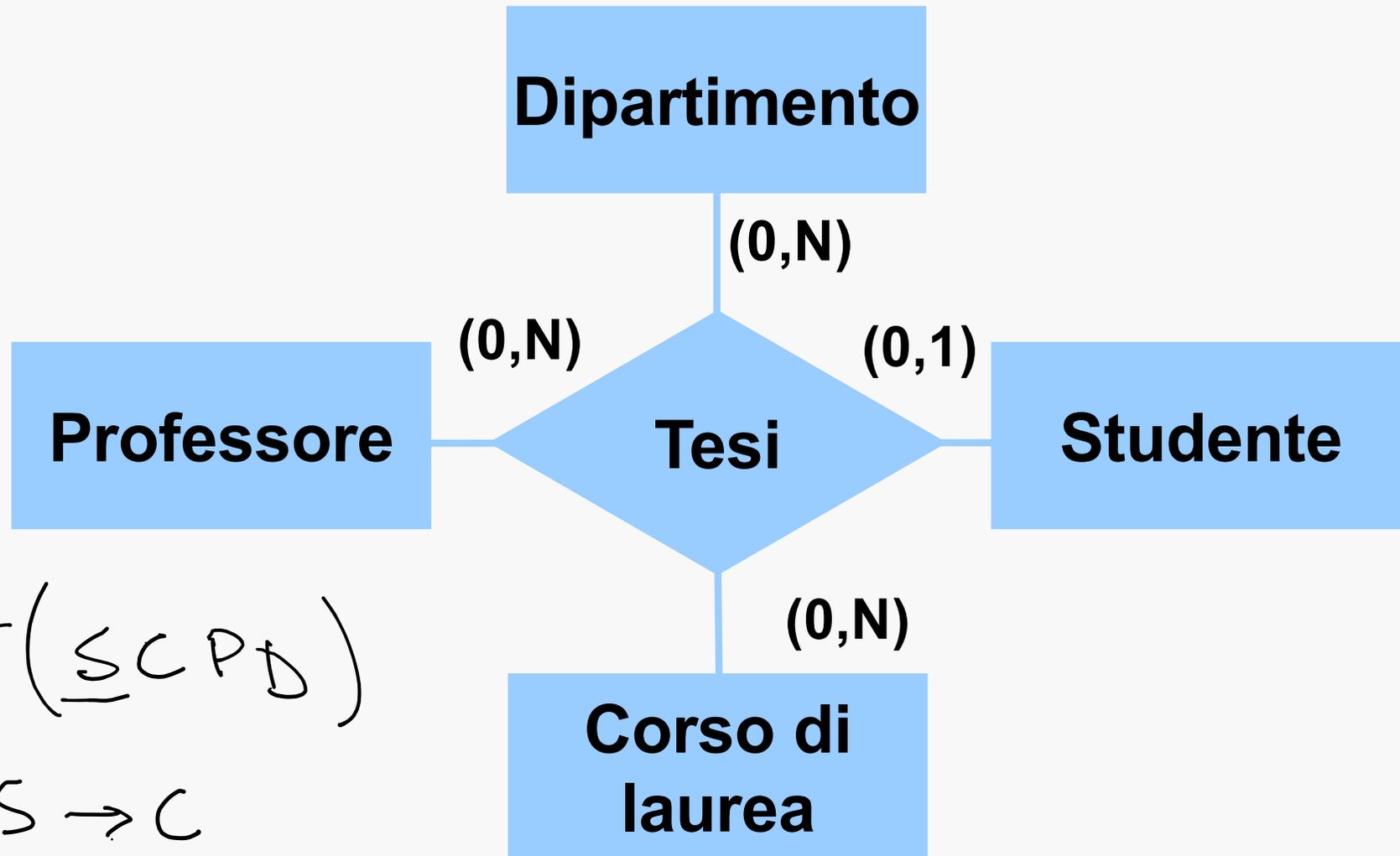
T



CHIAVE  
S



**Studente → Corso di laurea**  
**Studente → Professore**  
**Professore → Dipartimento**



$T(\underline{S}CPD)$

✓  $S \rightarrow C$

✓  $S \rightarrow P$

✗  $P \rightarrow D$

**Studente → Corso di laurea**  
**Studente → Professore**  
**Professore → Dipartimento**

T (S, C, P, D)

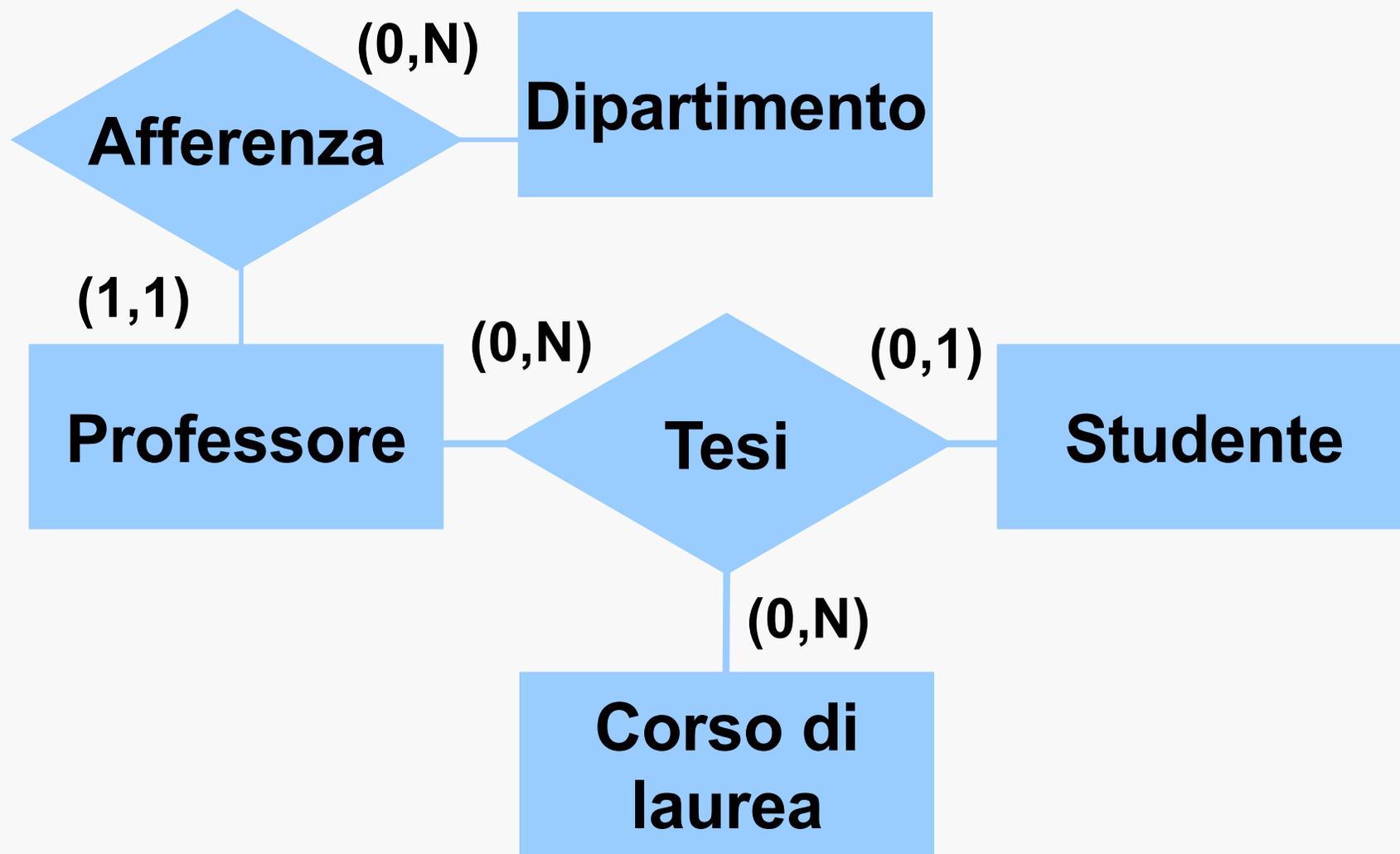
✓ S → C

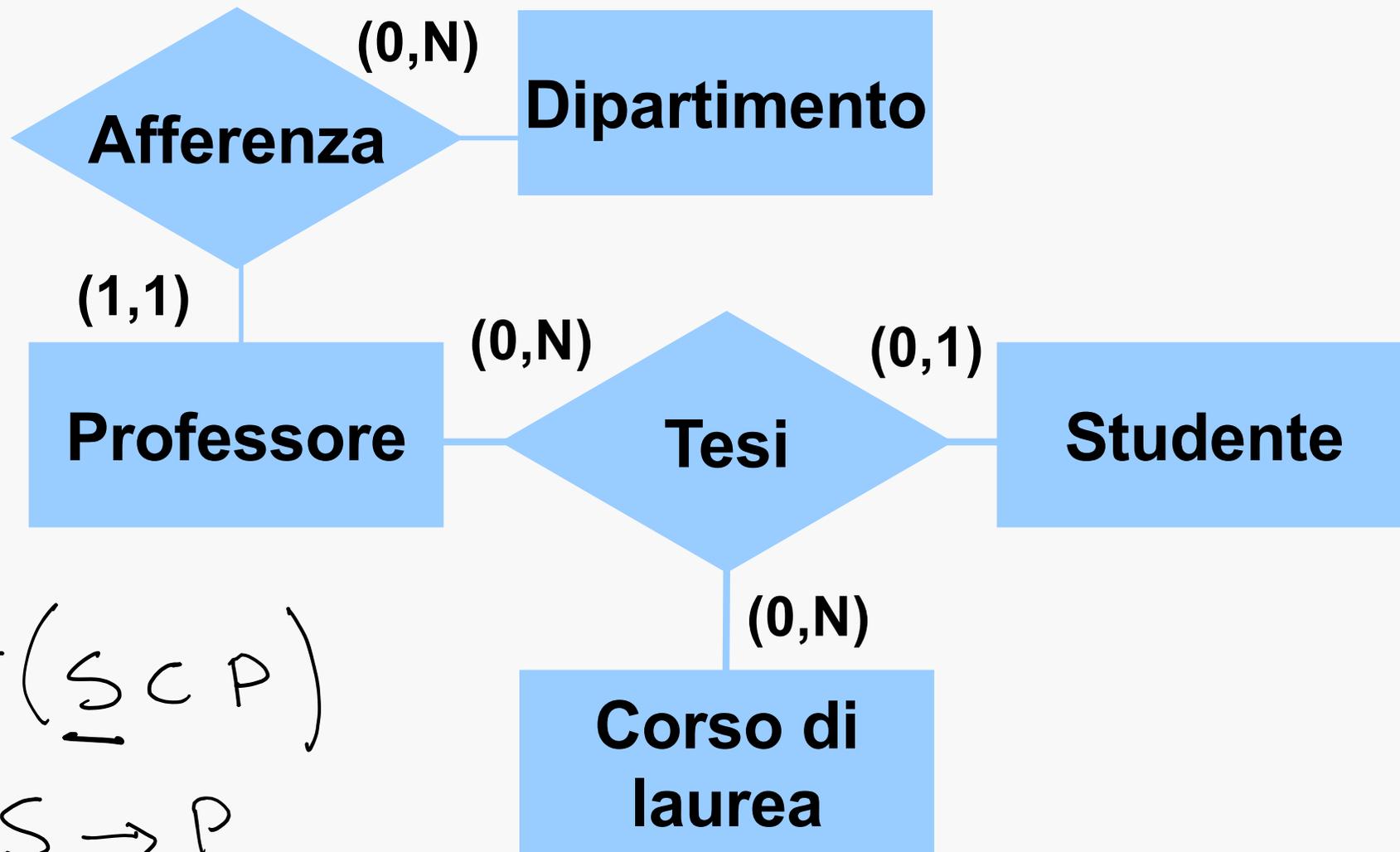
✓ S → P

✗ P → D

R<sub>1</sub> (S, C, P)

R<sub>2</sub> (P, D)





$T(S, C, P)$

✓  $S \rightarrow P$

✓  $S \rightarrow C$

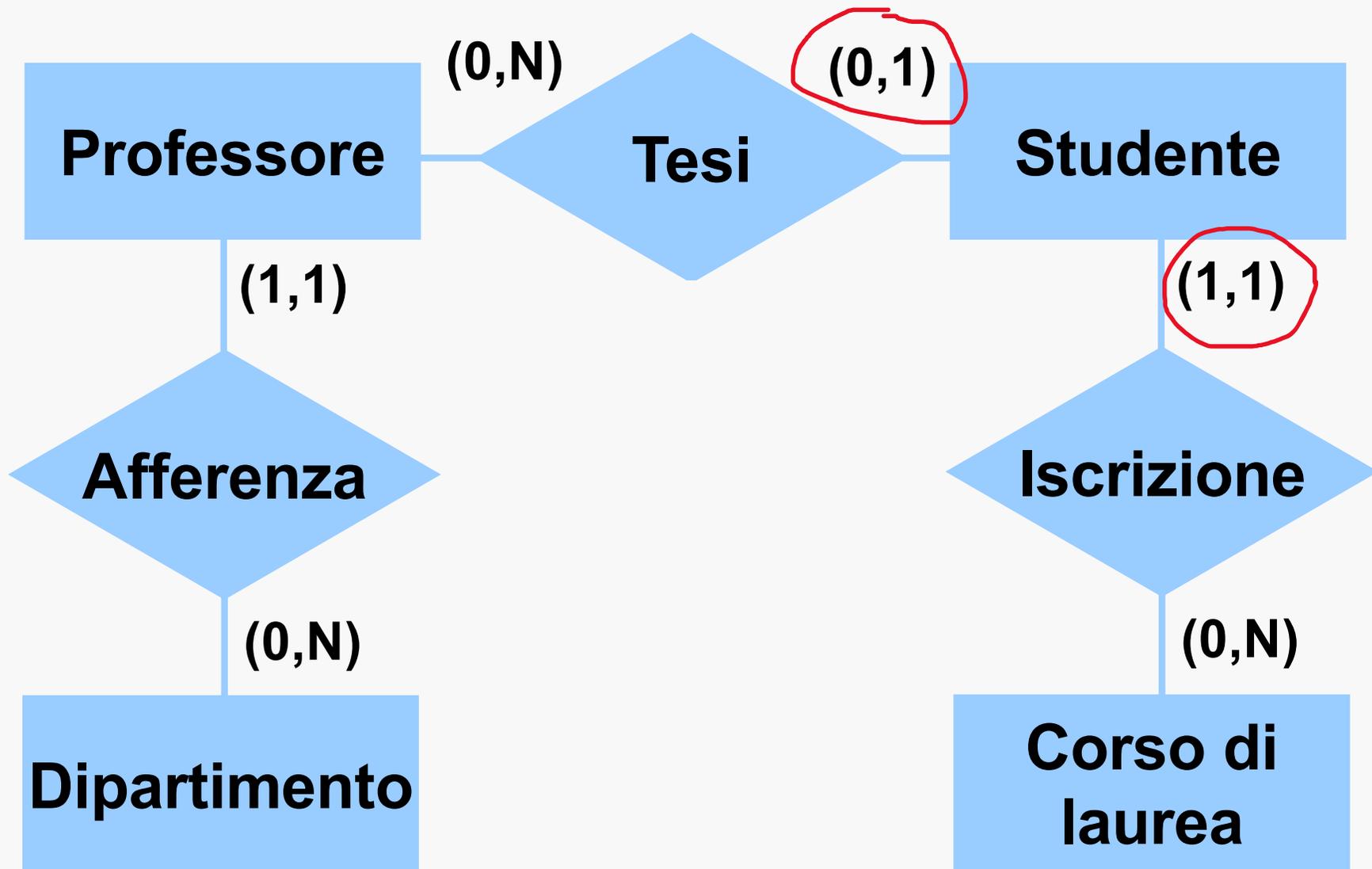
## Ulteriore analisi sulla base delle dipendenze

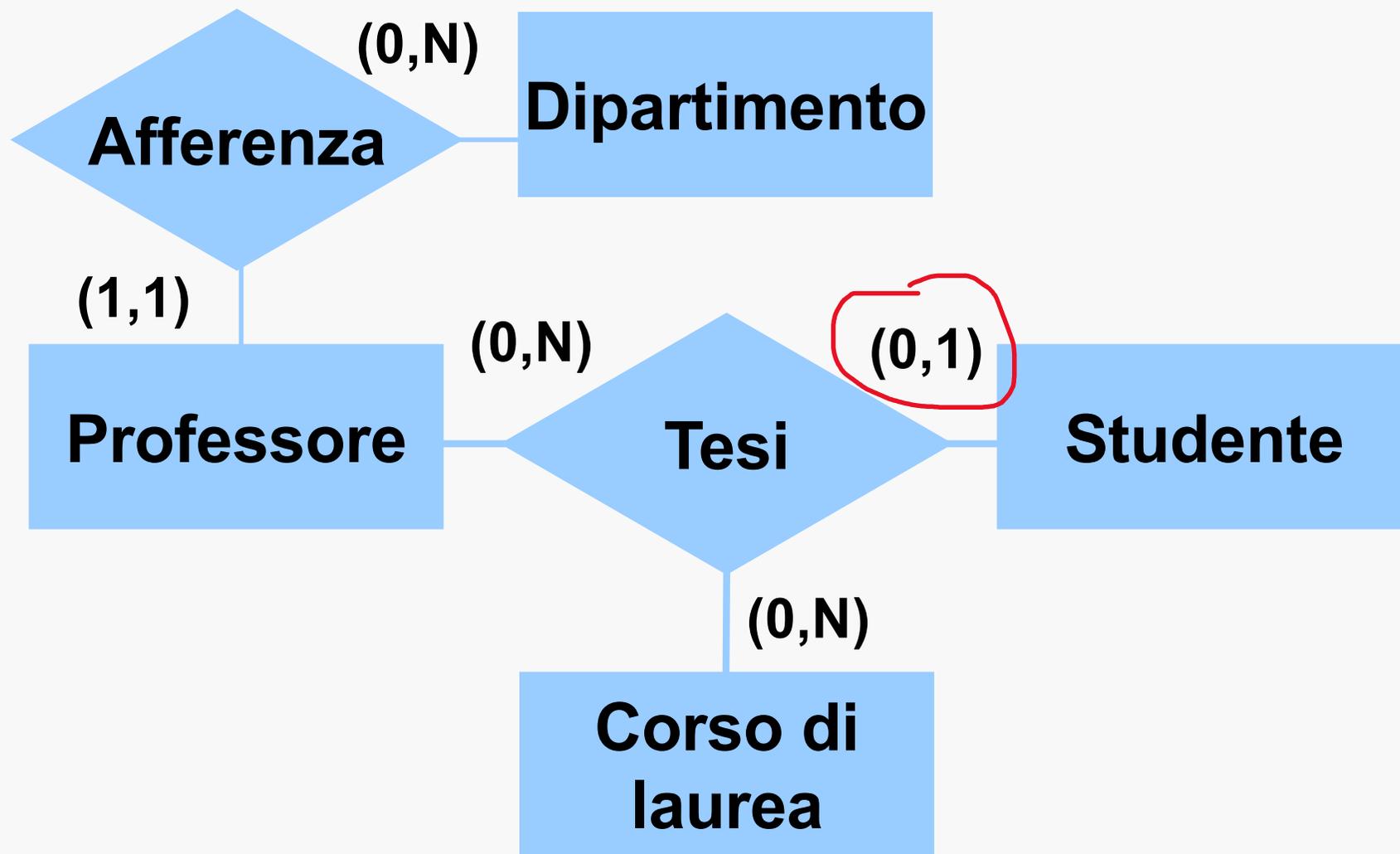
- La relationship **Tesi** (su S C P) è in BCNF sulla base delle dipendenze

Studente  $\rightarrow$  CorsoDiLaurea

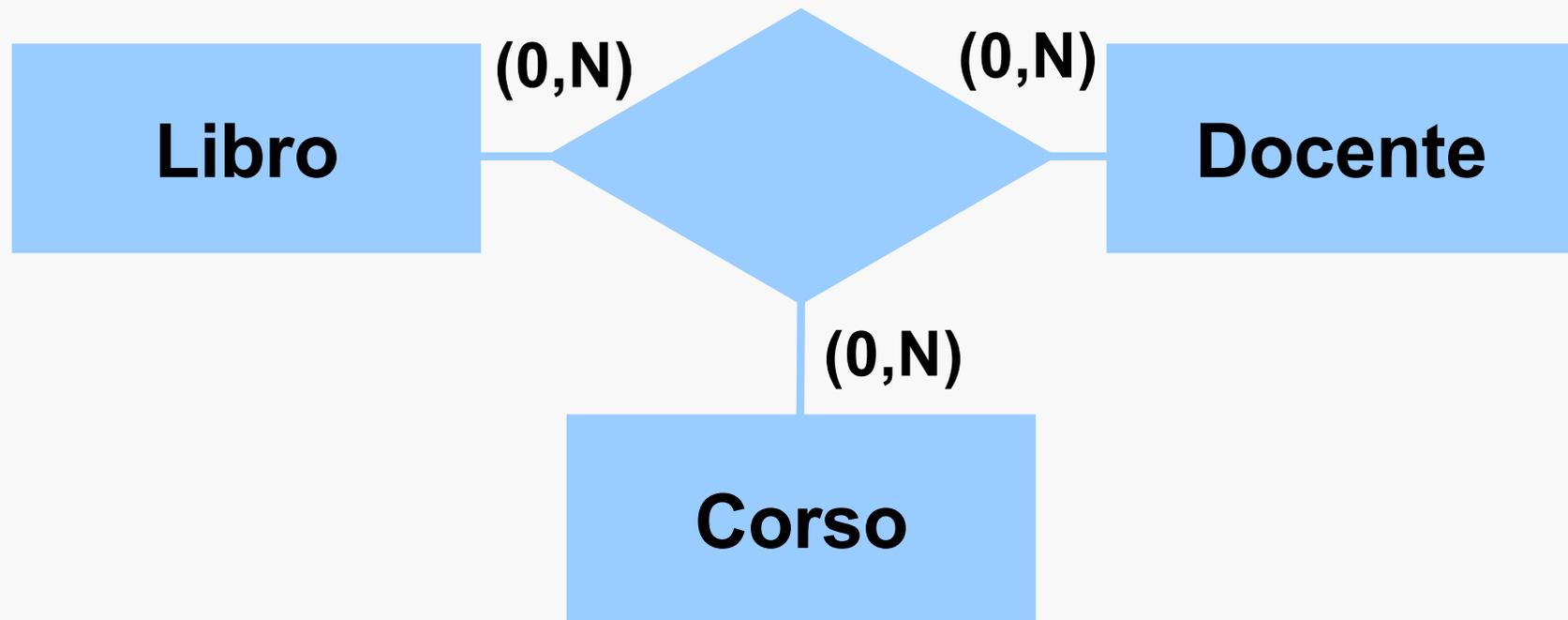
Studente  $\rightarrow$  Professore

- Però, le due proprietà sono indipendenti
- Questo suggerisce una ulteriore decomposizione





## Altro esempio, più sottile



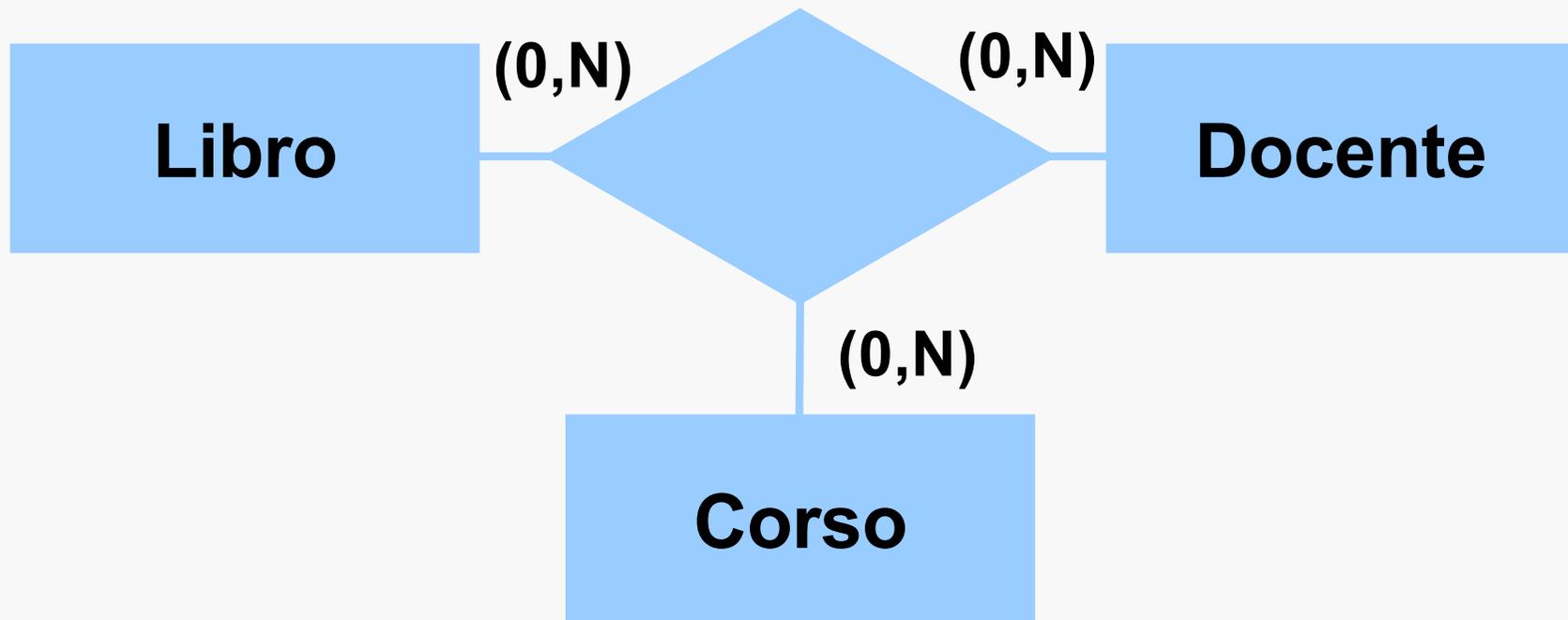
# Dettagli

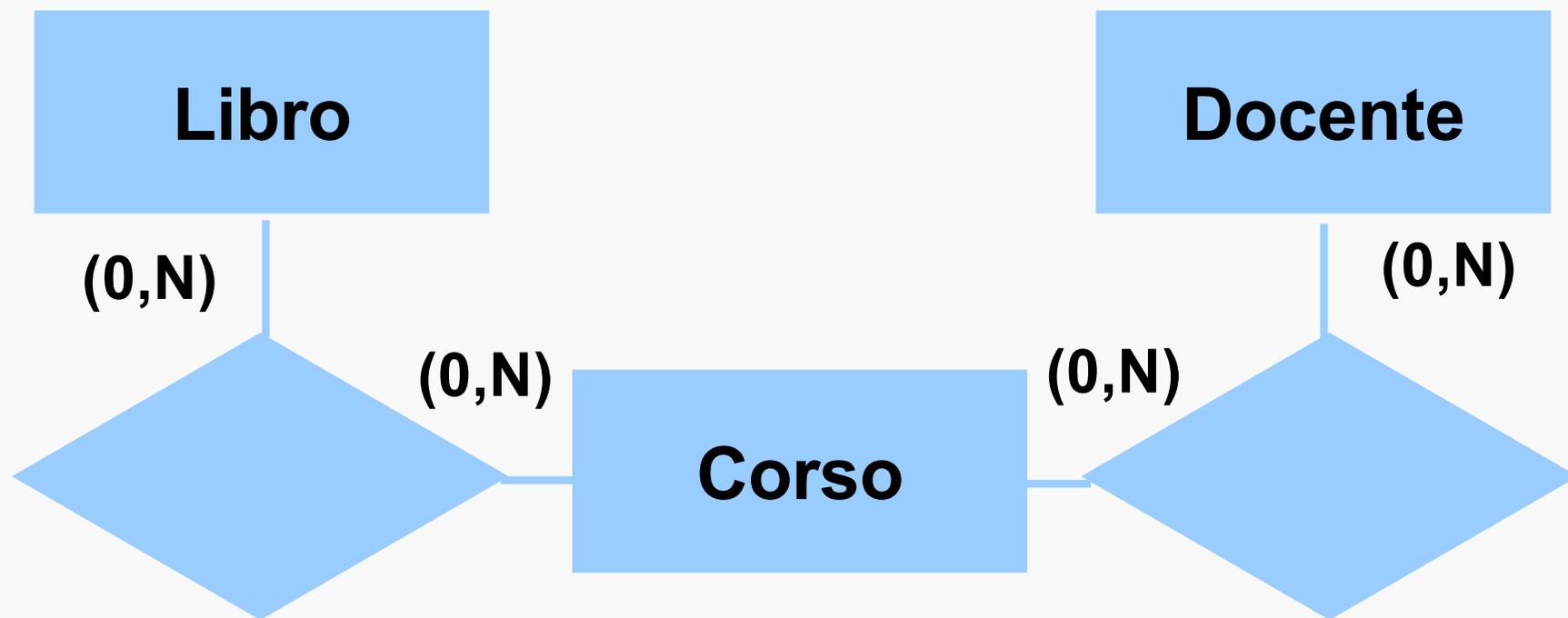
- Un corso ha uno o più docenti (ad esempio più canali) e uno o più libri di testo
  - ognuno dei docenti di un corso usa tutti i libri del corso
- Un docente può tenere più corsi
- Un libro può essere usato in più corsi

# In sostanza

- Non c'è legame diretto fra docente e libro
- I legami sono
  - fra corso e docente
  - fra corso e libro

## Quindi ...





# Perché possiamo decomporre?

- Perché le due relationship sono indipendenti!

# Abbiamo visto qualcosa di simile nel relazionale

| <b>Impiegato</b> | <b>Progetto</b> | <b>Sede</b>   |
|------------------|-----------------|---------------|
| <b>Rossi</b>     | <b>Marte</b>    | <b>Roma</b>   |
| <b>Verdi</b>     | <b>Giove</b>    | <b>Milano</b> |
| <b>Verdi</b>     | <b>Venere</b>   | <b>Milano</b> |
| <b>Verdi</b>     | <b>Giove</b>    | <b>Roma</b>   |
| <b>Verdi</b>     | <b>Venere</b>   | <b>Roma</b>   |
| <b>Bruni</b>     | <b>Saturno</b>  | <b>Milano</b> |

| <b>Impiegato</b> | <b>Progetto</b> |
|------------------|-----------------|
| <b>Rossi</b>     | <b>Marte</b>    |
| <b>Verdi</b>     | <b>Giove</b>    |
| <b>Verdi</b>     | <b>Venere</b>   |
| <b>Bruni</b>     | <b>Saturno</b>  |

| <b>Impiegato</b> | <b>Sede</b>   |
|------------------|---------------|
| <b>Rossi</b>     | <b>Roma</b>   |
| <b>Verdi</b>     | <b>Milano</b> |
| <b>Verdi</b>     | <b>Roma</b>   |
| <b>Bruni</b>     | <b>Milano</b> |

# Corso, libro e docente nel modello relazionale

| Corso | Docente | Libro  |
|-------|---------|--------|
| BD    | Atzeni  | ACFPT1 |
| BD    | Atzeni  | ACFPT2 |
| BD    | Torlone | ACFPT1 |
| BD    | Torlone | ACFPT2 |
| APS   | Cabibbo | Larman |

| Corso | Docente |
|-------|---------|
| BD    | Atzeni  |
| BD    | Torlone |
| APS   | Cabibbo |

| Corso | Libro  |
|-------|--------|
| BD    | ACFPT1 |
| BD    | ACFPT2 |
| APS   | Larman |

# Nota bene, riesco a ricostruire

| Corso | Docente | Libro  |
|-------|---------|--------|
| BD    | Atzeni  | ACFPT1 |
| BD    | Atzeni  | ACFPT2 |
| BD    | Torlone | ACFPT1 |
| BD    | Torlone | ACFPT2 |
| APS   | Cabibbo | Larman |

| Corso | Docente |
|-------|---------|
| BD    | Atzeni  |
| BD    | Torlone |
| APS   | Cabibbo |

| Corso | Libro  |
|-------|--------|
| BD    | ACFPT1 |
| BD    | ACFPT2 |
| APS   | Larman |

| Corso | Docente | Libro  |
|-------|---------|--------|
| BD    | Atzeni  | ACFPT1 |
| BD    | Atzeni  | ACFPT2 |
| BD    | Torlone | ACFPT1 |
| BD    | Torlone | ACFPT2 |
| APS   | Cabibbo | Larman |