

# La Progettazione di una basi di dati

Prof. Alfredo Pulvirenti

Prof. Salvatore Alaimo

(Atzeni-Ceri Capitolo 6-7)

# Sommario

- Progettazione di Basi di Dati. Metodologie e Modelli per la Progettazione
  - Il modello Entità- Relazione (E-R)
  - Progettazione Concettuale
  - Raccolta ed Analisi dei Requisiti
  - Strategie: Top-Down, Bottom-Up, Mista
  - Qualità di uno Schema Concettuale
- Progettazione Logica
  - Analisi delle Prestazioni di Schemi E-R
  - Ristrutturazione di Schemi E-R
  - Traduzione nel Modello Relazionale
- Esempi di Progettazione completa di sistemi DBMS

# Sommario

- Normalizzazione di Schemi Relazionali.
  - Dipendenze funzionali, Assiomi di Armstrong
  - Decomposizioni senza perdite (loss-less joins)
  - Decomposizioni che conservano le dipendenze funzionali
  - Forma Normale di Boyce-Codd. Terza Forma Normale

# Progetto della base di dati

- Come in ogni ciclo di vita di un sistema informativo, il progetto di una base di dati consiste dei seguenti passi:
  - Raccolta ed analisi dei requisiti
    - Individuare proprietà e funzionalità del sistema
  - Progettazione
    - individua l'organizzazione e la struttura della base di dati
    - schematizza le operazioni sui dati
  - Implementazione
  - Validazione e collaudo
  - Funzionamento e manutenzione

# Metodologie di progettazione

- Dal principio dell'ingegneria: separa il maniera netta le decisioni relative a cosa rappresentare in una base di dati (prima fase), da quelle relative a come farlo (seconda fase).
- Distinguiamo:
  - la progettazione concettuale;
  - la progettazione logica;
  - la progettazione fisica.

# La progettazione concettuale

- Scopo della progettazione concettuale è:
  - *tradurre la descrizione informale della realtà*, risultato dell'analisi dei requisiti del DB (tipicamente sotto forma di documenti e moduli di vario genere), *in uno schema formale* e completo che dovrà essere *indipendente dai criteri di rappresentazione del DBMS usato*.
- La descrizione formale fa riferimento ad un modello concettuale, cioè un insieme di concetti e notazioni standard adatti alla rappresentazione del dominio applicativo.

# Progettazione Logica

- Consiste nella traduzione dello schema concettuale in termini di un determinato **modello logico** (ad esempio il modello relazionale) di dati usato dal DBMS che si intende utilizzare. Il risultato è lo **schema logico**.
- Include anche l'**ottimizzazione** della rappresentazione in funzione delle operazioni eseguite (es. **normalizzazione**).

# Progettazione Fisica

- Si completa lo schema logico con la **specifica dei parametri fisici** di memorizzazione dei dati. Si produce lo **schema fisico** che fa riferimento ad un certo **modello fisico** dei dati che dipende dal DBMS scelto.



# Metodologie di progettazione

- La progettazione **concettuale**
- La progettazione **logica**
- La progettazione **fisica**

# L'analisi dei requisiti

- Deve:
  - individuare le **proprietà** e le **funzionalità** del sistema;
  - produrre una descrizione dei **dati** coinvolti e delle **operazioni** su di essi;
  - individuare (in linea di massima) i requisiti **software** ed **hardware** del sistema.

# Output dell'analisi

- **Studio di fattibilità** che stimi:
  - i costi in termini di budget, di impegno del personale;
  - le inefficienze temporanee dovute al cambio di sistema e di modalità di lavoro;
  - i benefici in termini di riduzione dei tempi di lavoro o migliore efficienza dei processi aziendali;
- **piano di sviluppo** del sistema con priorità e tempi di realizzazione.

# Procedure per l'analisi

- Per ogni settore aziendale in esame si procede con i seguenti passi:
  - si analizza il sistema informativo esistente, si intervistano i responsabili del settore;
  - si produce una prima versione dei requisiti in linguaggio naturale, raggruppando **frasi descrittive** relative a categorie diverse di **dati** e di **operazioni**.

# Procedure per l'analisi

- si analizzano le descrizioni per eliminare ambiguità provocate da:
  - pluralismo di percezione
  - incompletezze di descrizione
- ambiguità del tipo:
  - Omonimie
  - Sinonimie
  - conflitti di descrizione
  - similitudini

# Procedure per l'analisi

- si ricontrollano insieme ai responsabili di settore **le frasi** relative alle varie categorie di dati e alle operazioni che li coinvolgono
- si costruisce a partire dalle frasi verificate un **glossario di termini**;
  - Il glossario tipicamente per ogni termine contiene: la descrizione, l'elenco dei sinonimi e l'elenco dei termini collegati.
- si verifica la **completezza**
  - tutti gli aspetti importanti sono stati considerati
- si verifica la **consistenza** delle specifiche vedendo se:
  - tutti i termini sono stati definiti;
  - tutti i termini compaiono in delle operazioni;
  - le operazioni fanno riferimento a termini definiti.

# Esempio: “frasi descrittive di un magazzino”

- **il magazzino è composto da scaffali**
- **i fornitori forniscono prodotti**
- **i clienti ordinano prodotti**
- **gli scaffali contengono prodotti**
- **gli operai sono addetti agli scaffali**

# Glossario corrispondente

TERMINE	DESCRIZIONE	SINONIMI	LEGAME
fornitore	p. iva, denom., indirizzo, num. tel.		prodotto
cliente	p. iva, denom., indirizzo, num. tel.	acquirente	prodotto
prodotto	codice, nome, genere....	articolo voce	fornitore scaffale cliente
Scaffale	supporto numerato	ripiano (incertezza)	operaio prodotto
operaio	dati anagrafici, matricola, qualifica	addetto magazziniere	scaffale



# Il modello Entità-Relazione

- Il modello **Entità-Relazione** (E-R) è un **modello concettuale di dati** che contiene alcuni costrutti atti a descrivere la realtà in maniera semplice, indipendente dalla organizzazione dei dati nel computer.
- I costrutti sono:
  - **Entità**
  - **Associazioni (Relazioni)**
  - **Proprietà (Attributi)**
  - **Cardinalità**
  - **Identificatori**
  - **Generalizzazioni**
  - **Sottoinsiemi**

# Costrutti fondamentali

- CONOSCENZA ASTRATTA
  - entità
  - associazione
  - Proprietà
- CONOSCENZA CONCRETA
  - istanza di entità
  - istanza di associazione
  - proprietà delle istanze

# Entità – Istanza di Entità

- **Entità** sono classi di oggetti (cose, persone) che hanno proprietà comuni ai fini dell'applicazione di interesse che si intende modellare.
- **Istanza di Entità:** cosa (oggetto, persona) che esiste di per sé nel dominio applicativo, della quale si vogliono registrare fatti specifici e che può essere chiaramente identificata in modo da poterla distinguere dalle altre

# Esempi

- Entità:
  - Docente
  - Corso
  - Automobile
  - Studente
  - Volo
  - Percorso
- Istanza di Entità:
  - il docente Ferro
  - il corso Basi di Dati
  - l'auto AB111XY
  - lo studente 567345
  - il volo AZ3313
  - il percorso CT-PA

# Associazione - istanza di associazione

- Le **Associazioni** (o **Relazioni**) rappresentano legami logici fra due o più entità dell'applicazione.
  - Possono esserci più relazioni fra le stesse entità (associazioni ricorsive).
- **Istanza di Associazione**: fatto che descrive un'azione o una situazione e che stabilisce legami tra istanze di entità (associa, mette in relazione)

# Esempi

- Associazione:
  - Insegna
  - Appartiene
  - Ordina
  - Lavora
  - Viaggia sulla tratta
- Istanza di associazione:
  - Ferro insegna Basi di dati
  - Sicilia appartiene all'Italia
  - La ditta Rossi ordina PC
  - Bianchi lavora al magazzino 4
  - il TIR 542 viaggia sulla tratta CT-PA

# Proprietà delle istanze

- Dette anche attributi delle istanze: sono fatti che descrivono le caratteristiche delle istanze di entità e le caratteristiche delle istanze di associazione
- Le proprietà delle istanze assumono VALORI

# Esempi di proprietà delle istanze

- Proprietà di istanze di entità:
  - Ferro ha nome Alfredo
  - Il recapito della ditta Rossi è via Etnea 22
  - Formazione Analitica 1 si tiene al primo anno



# Esempi di proprietà delle istanze

- Proprietà di istanze di associazione
  - Ferro insegna basi di dati *nell'anno 1995*
  - Bianchi ha lavorato *3 ore* al magazzino 4
  - La ditta Rossi ordina 15 PC
  - Pappalardo supera Formazione Analitica 1 con 27

# Modello E-R

- Il modello E-R usa simboli grafici per favorire l'immediatezza della comprensione;
- Gli schemi E-R sono schemi grafici;
- Si possono aggiungere frasi di commento per rappresentare le caratteristiche non modellabili (ad esempio, vincoli complessi).

# Notazioni E-R: entità

**nome dell'entità**

**esempi:**

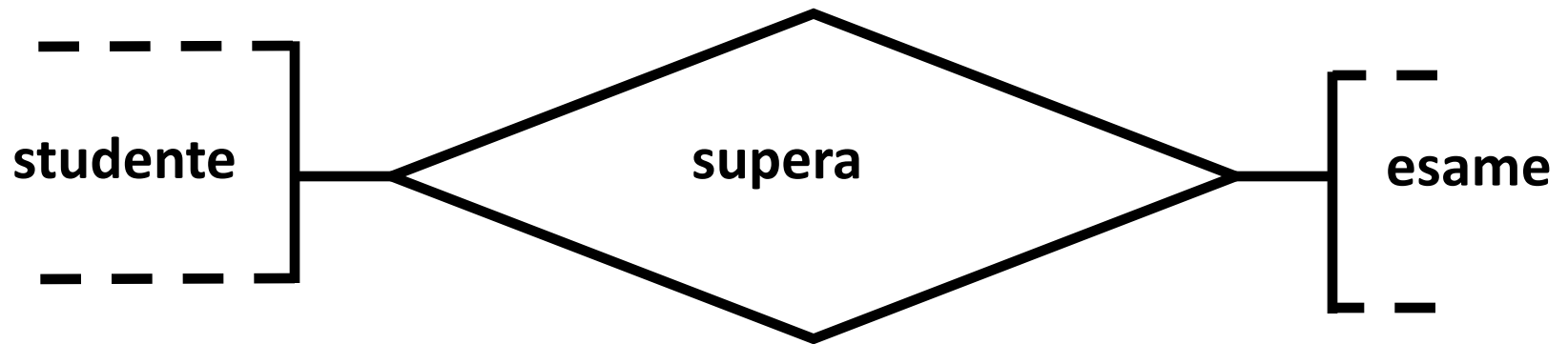
**studenti**

**materie**

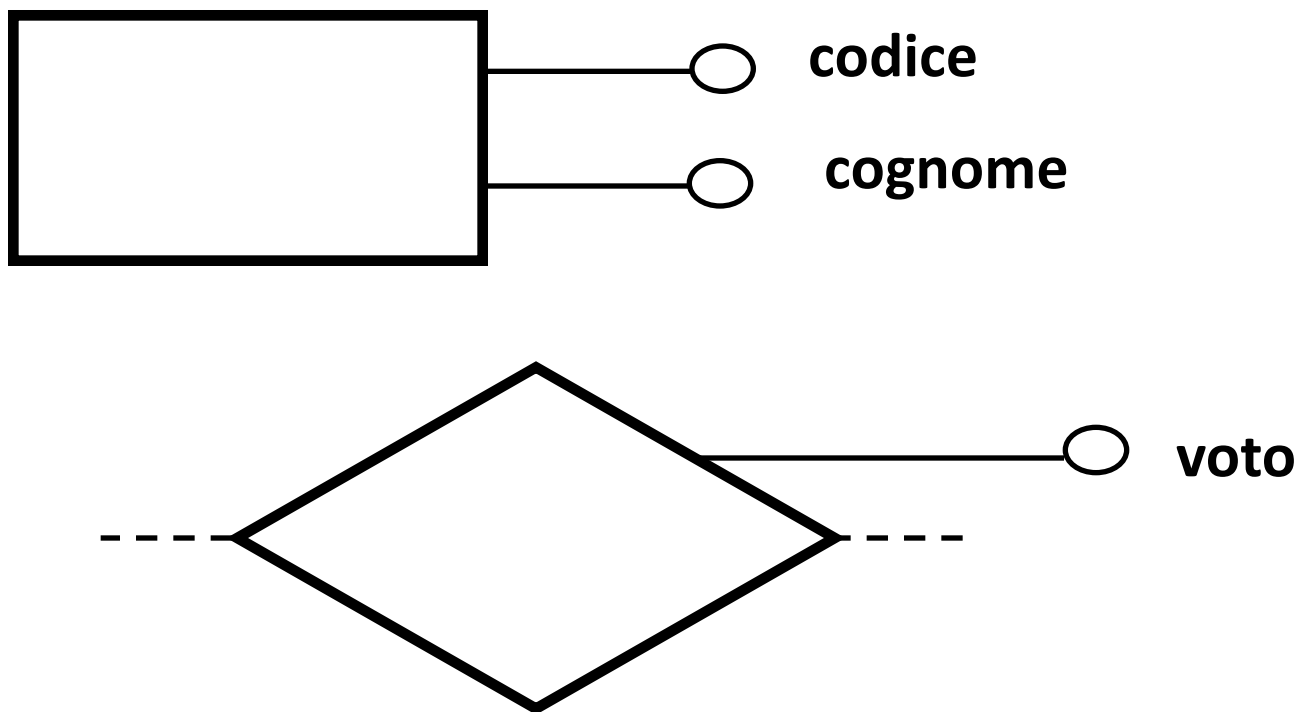
# Notazioni E-R: associazioni



**esempio:**



# Notazioni E-R: proprietà

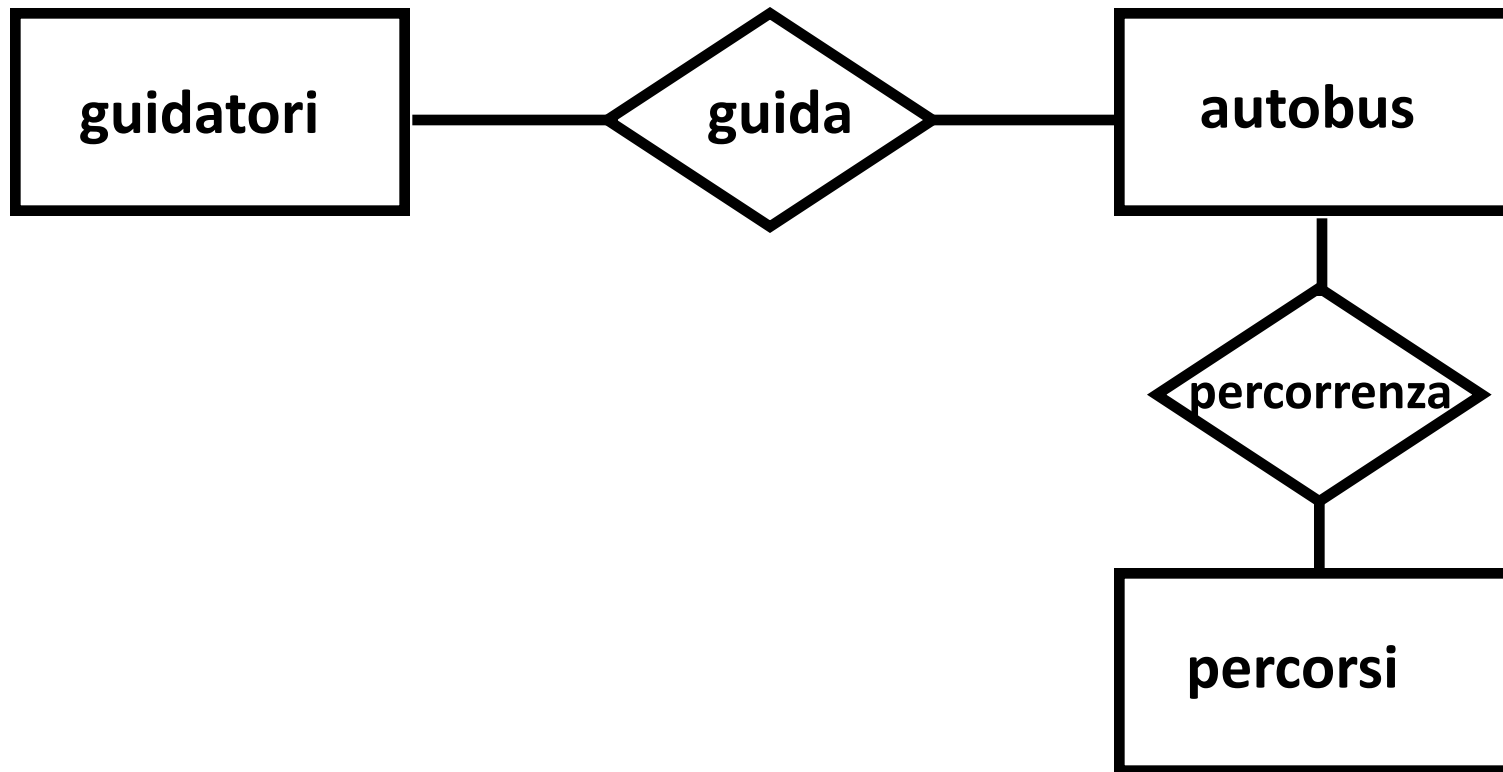


# Notazioni E-R: schemi scheletro

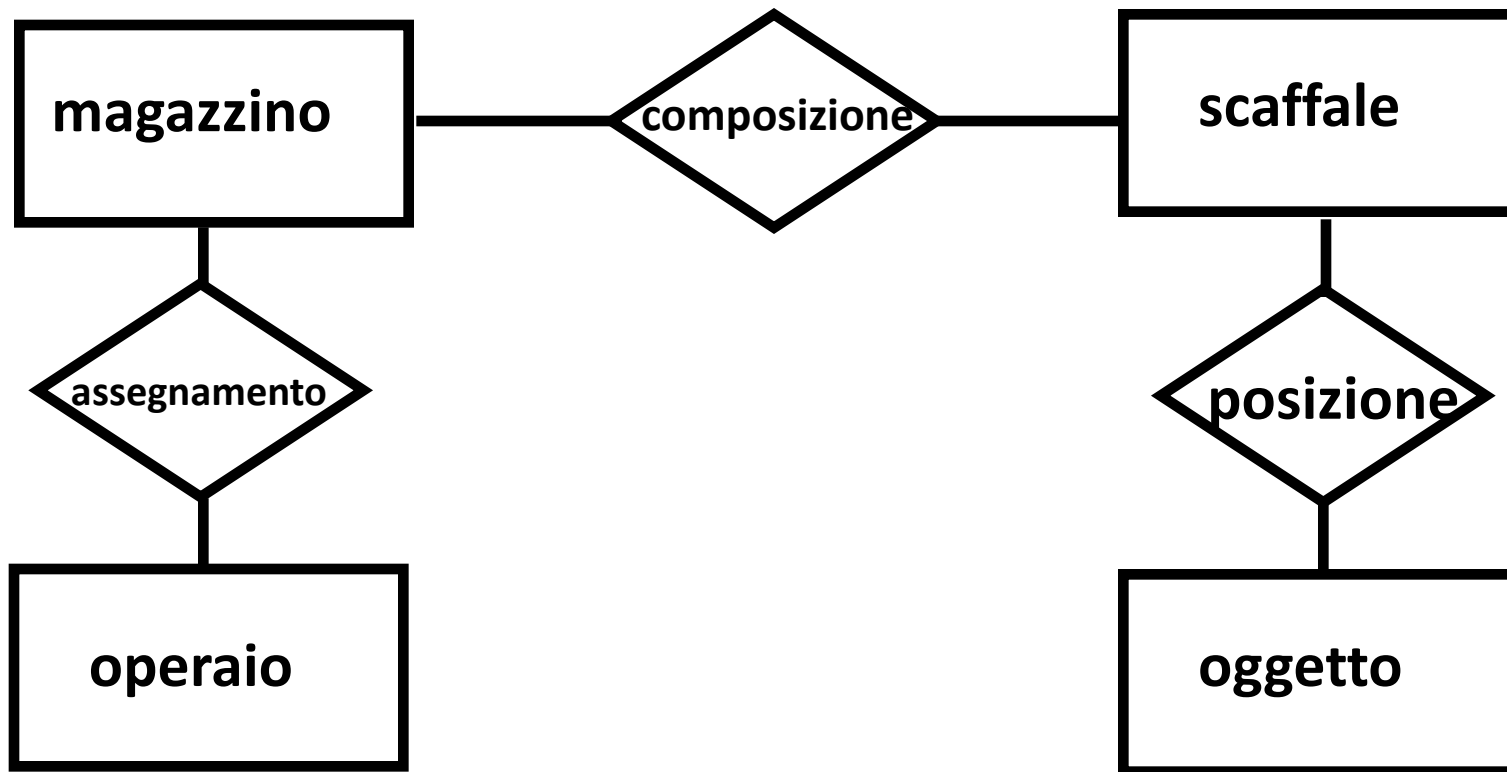
- Gli **schemi scheletro** descrivono una prima struttura di massima dello schema senza indicazioni sul TIPO delle entità e delle associazioni
- Descrivono in generale i collegamenti tra le entità di interesse e le associazioni che le legano

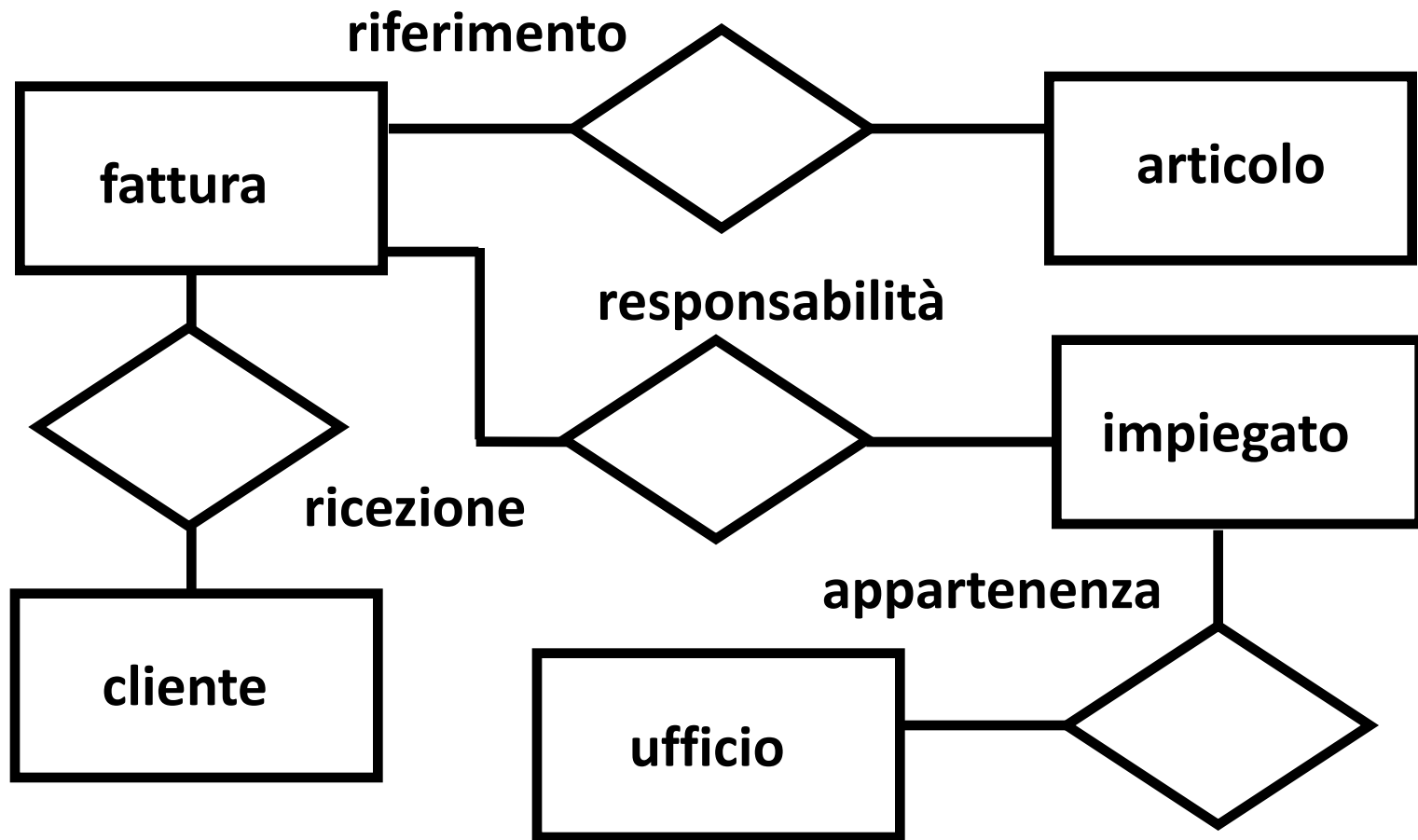
# Schemi scheletro (esempi)

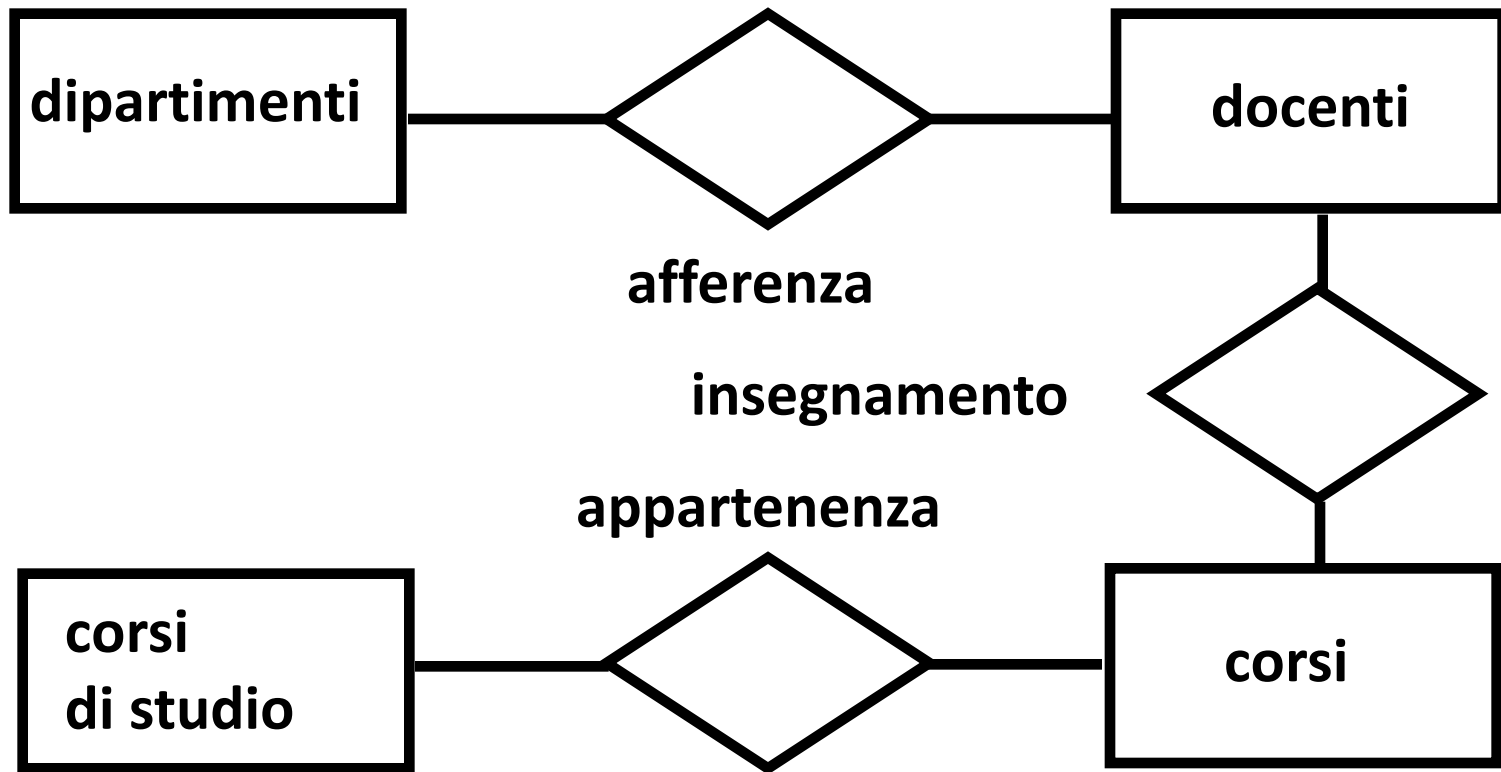




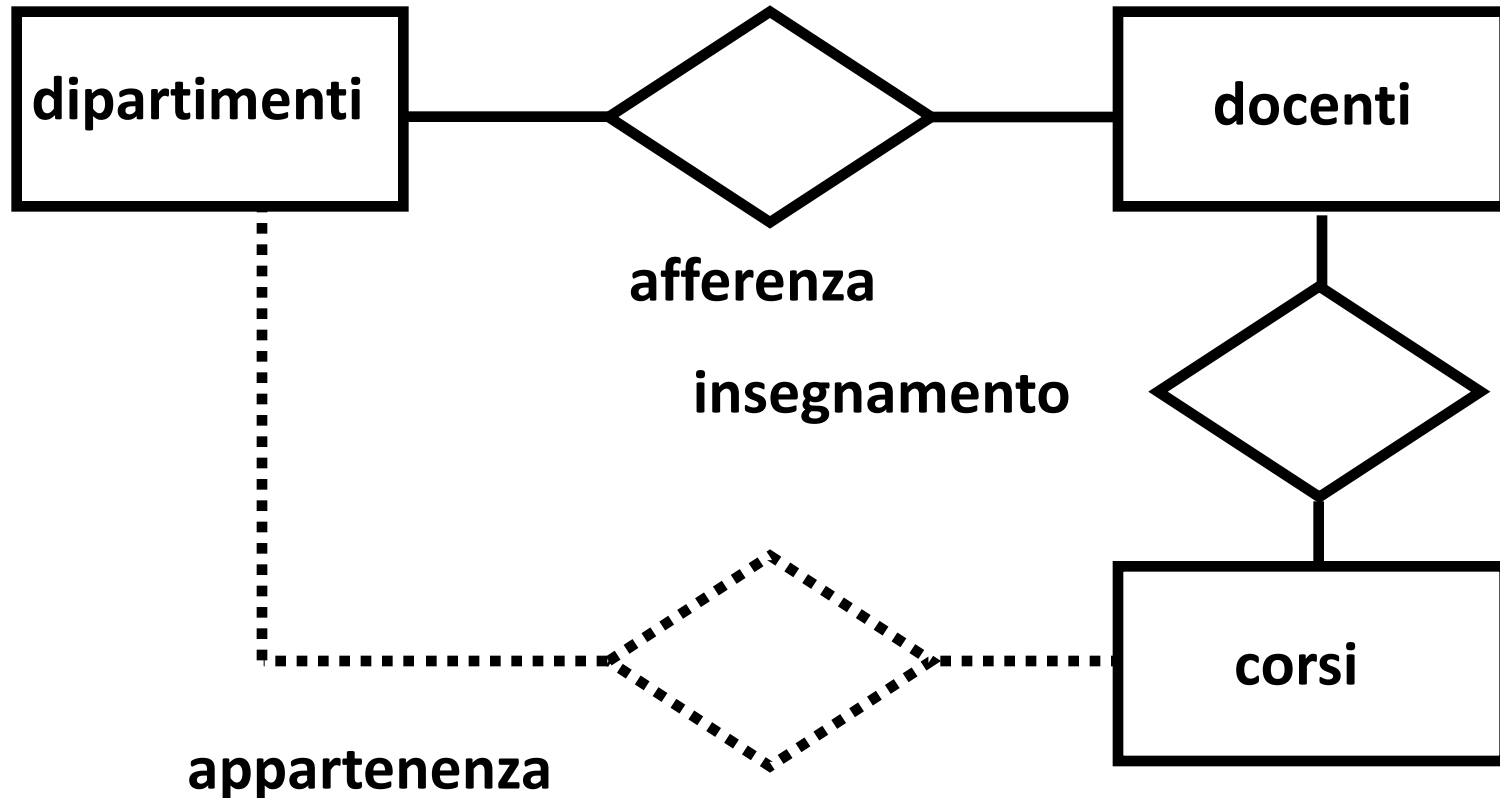




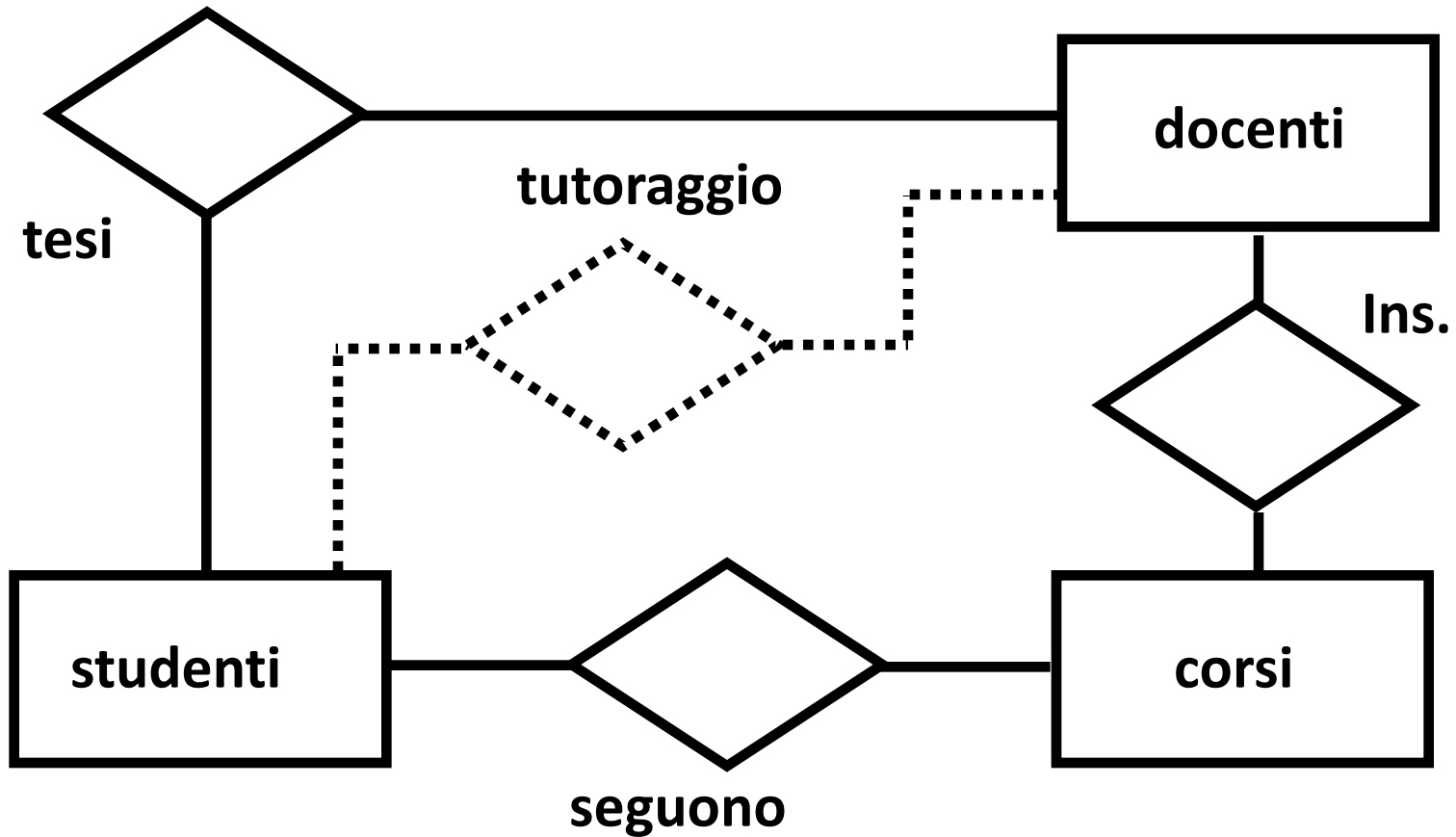




# Associazioni ridondanti?

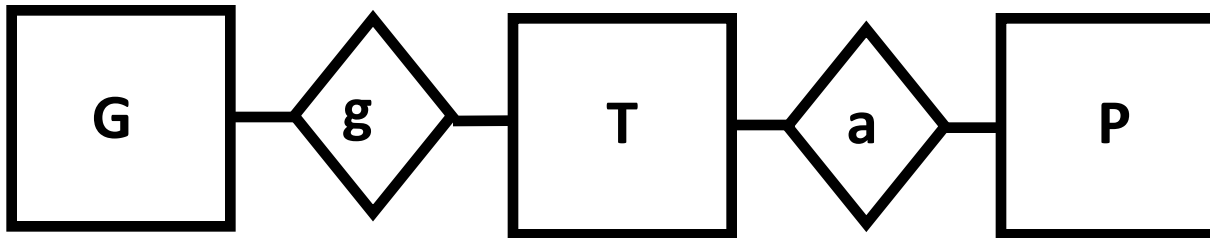


# Associazioni ridondanti?

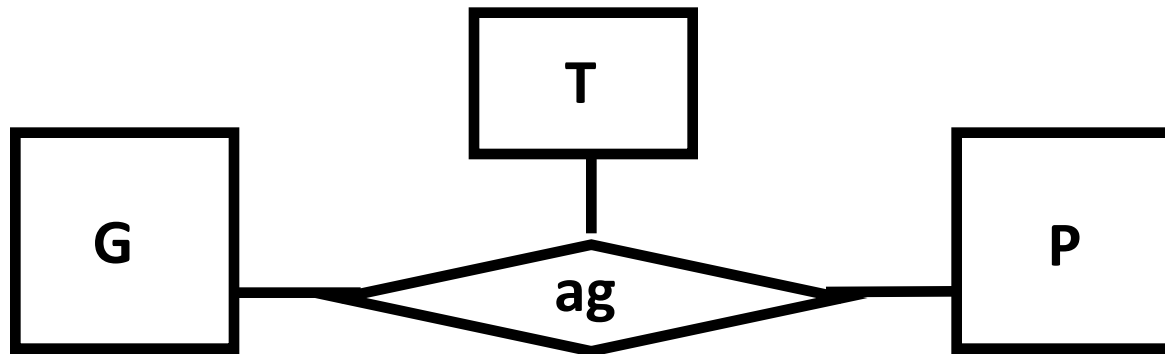


# Risoluzione delle ambiguità di modellazione

i Guidatori guidano TIR,  
i TIR sono assegnati a Percorsi



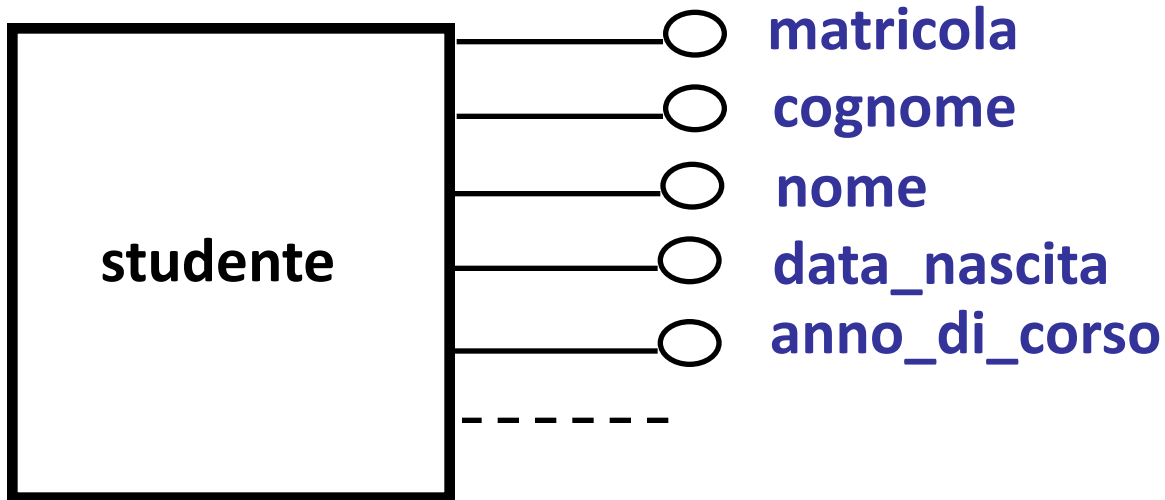
i Guidatori guidano TIR su Percorsi



# Le proprietà

- Gli schemi scheletro descrivono in generale i collegamenti tra le entità e le associazioni;
- Le entità e le associazioni devono essere descritte attraverso l'aggregazione di proprietà.

# Identificazione delle proprietà





# Proprietà



es.: matricola, cognome, voto

**multipla** (sono ammessi **n** valori)

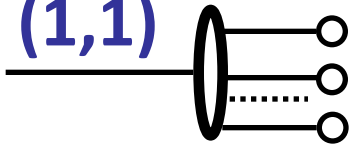
**(1,n)**



es.: titolo di studi, targa auto

**composta**

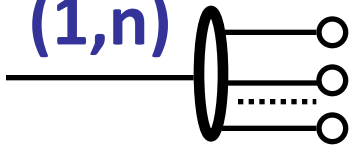
**(1,1)**



es.: data (gg,mm,aaaa), indirizzo,  
(denominazione, civico, cap)

**multipla composta**

**(1,n)**



es.: telefono (stato,  
città, numero)

Il simbolo **(n,m)** esprime la **cardinalità** minima e massima della proprietà

# Proprietà opzionali

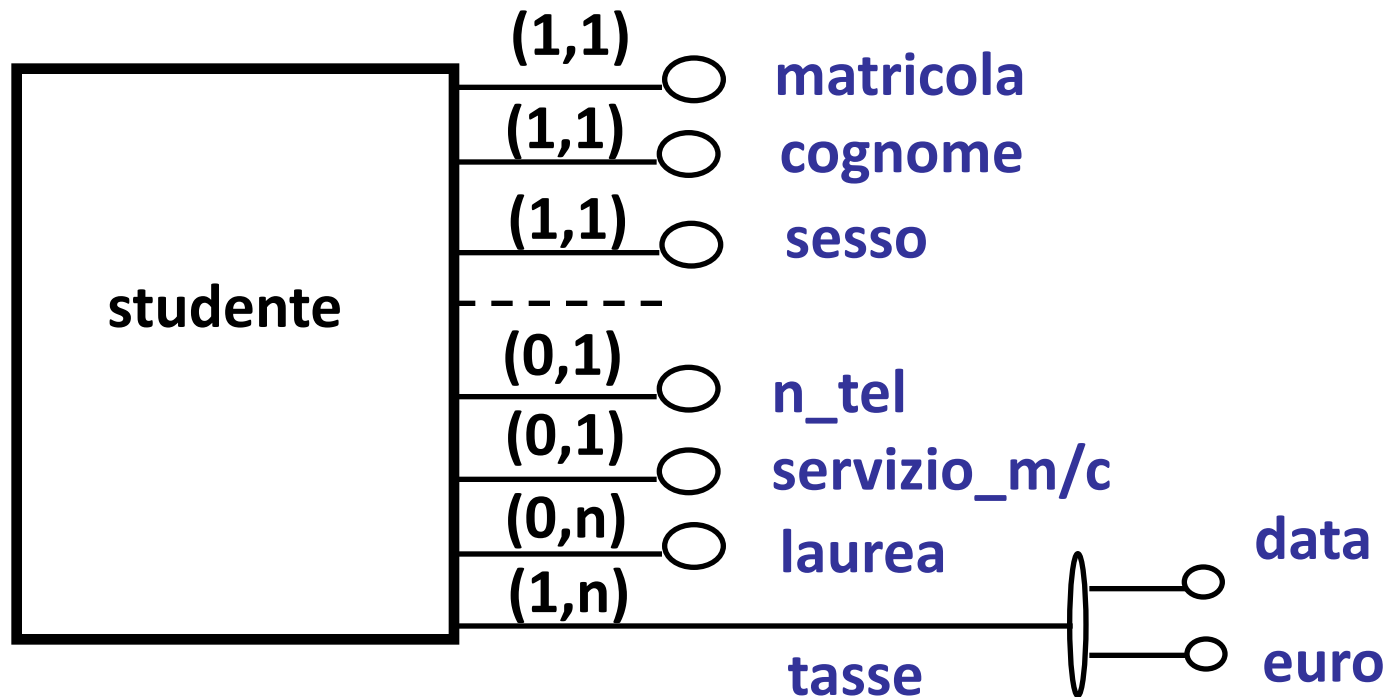
Proprietà **opzionale** (è ammessa la  
“non esistenza del valore”)

$(0,n)$   
\_\_\_\_\_○

$(0,1)$   
\_\_\_\_\_○

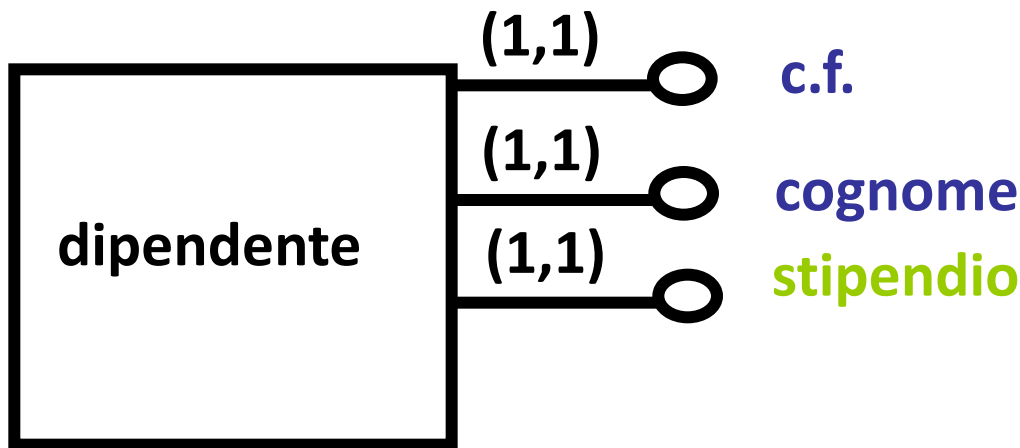
es.: tel., qualifica, targa

# Esempio di specifica delle proprietà



# Proprietà calcolate

Proprietà **calcolata (derivata)**: il valore è calcolato con un algoritmo, in alternativa la proprietà è **esplicita**.

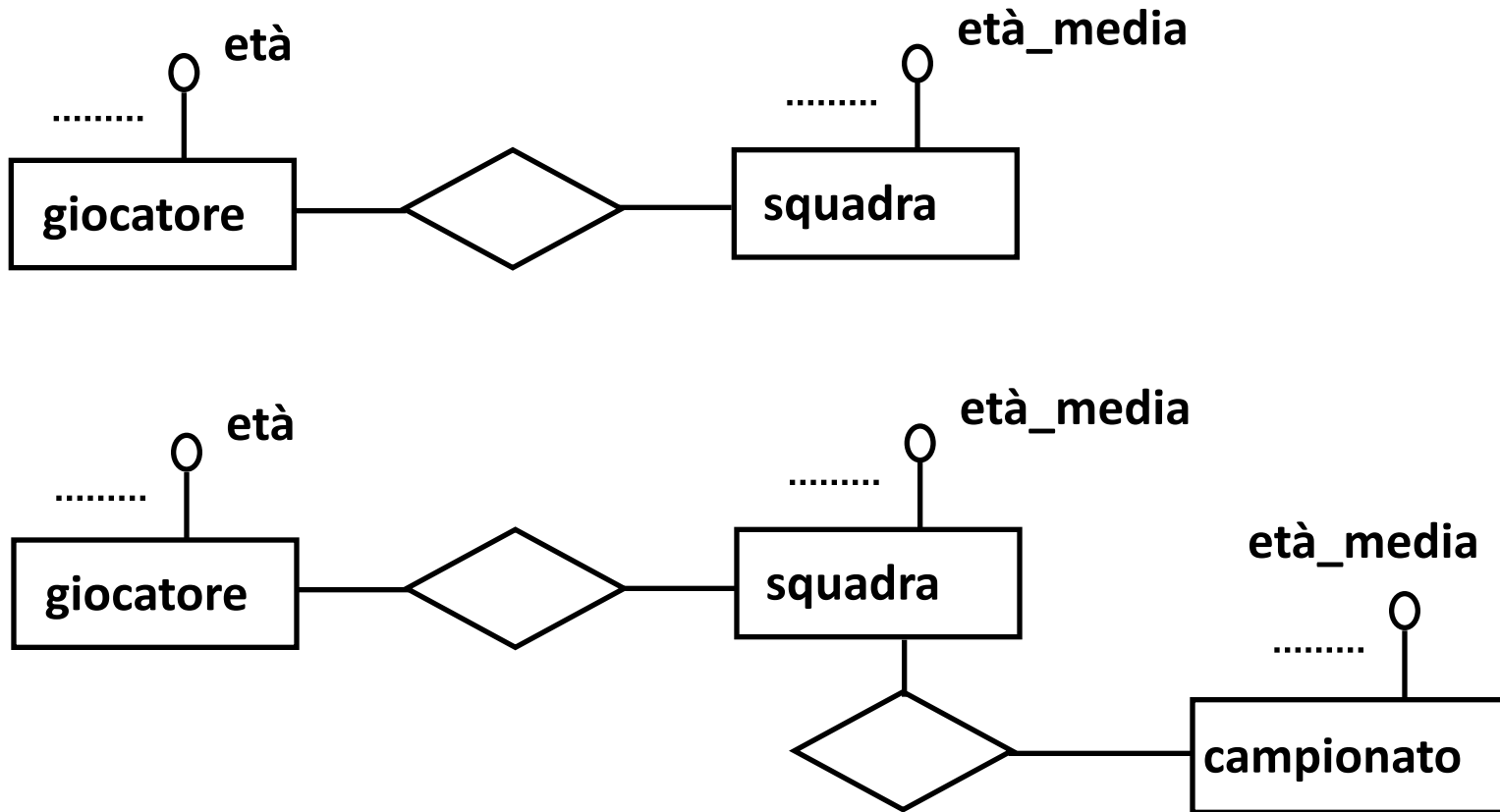


NB: la regola di calcolo va espressa a parte, usando un qualche linguaggio di specifica:

`stipendio = salario_giornaliero*presenze`

# Proprietà di sintesi


- valori medi , max e min ecc.

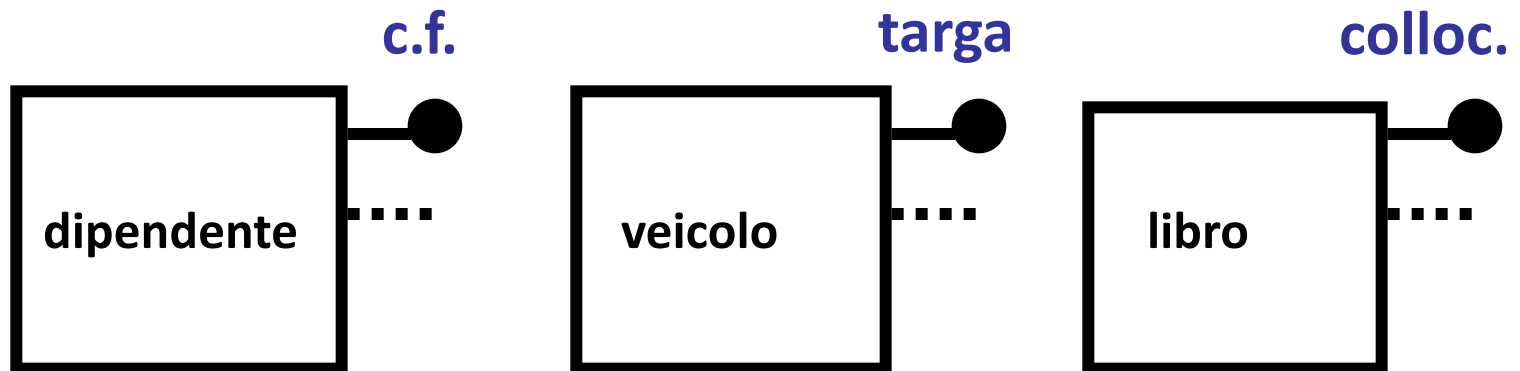


# Proprietà chiave

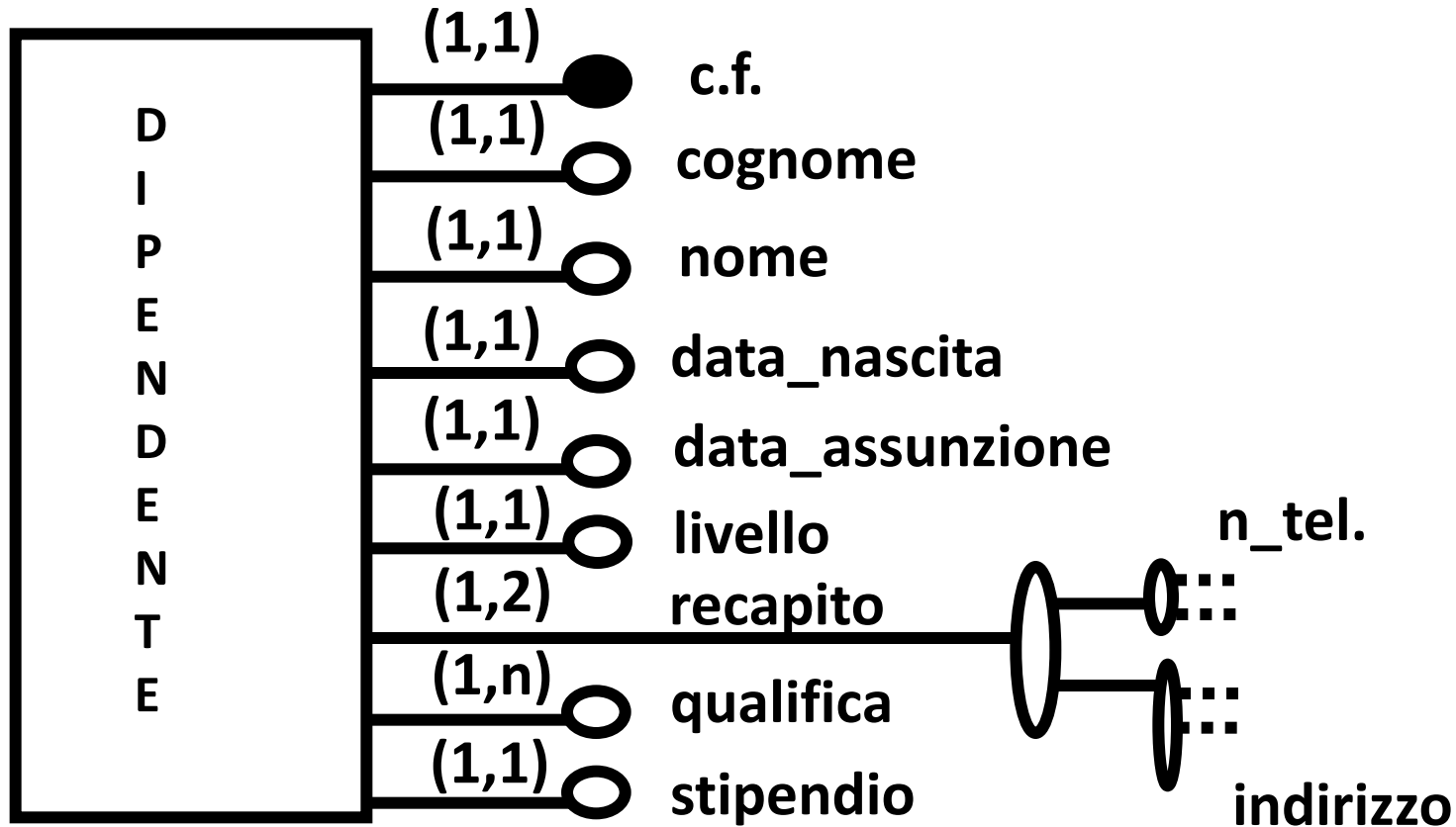
Una proprietà (attributo) **chiave** identifica in modo univoco la **singola istanza** di entità (o di associazione):

- obbligatoria, unica, esplicita
- può essere composta
- in generale non è modificabile

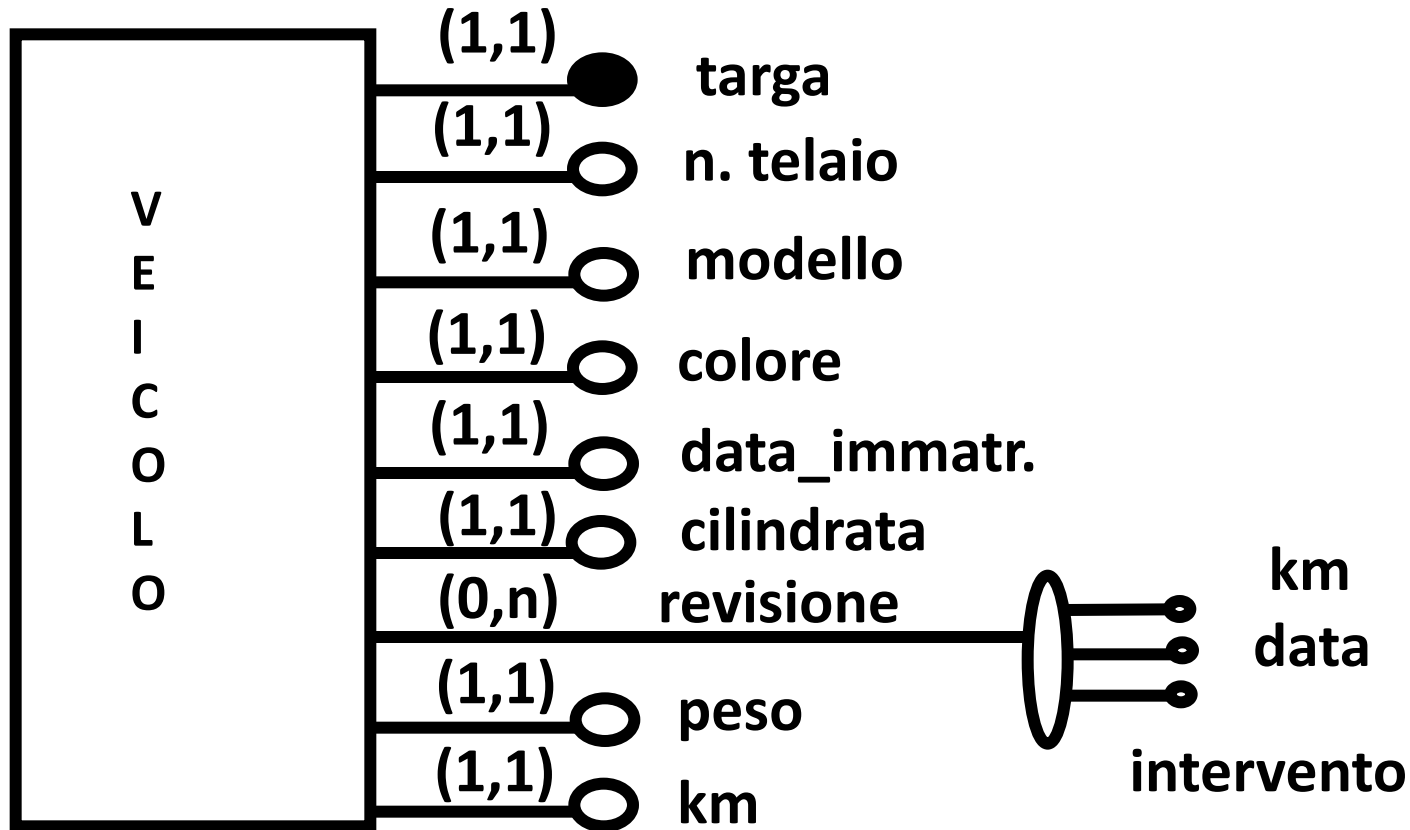
Simbolo = 



# Esempio



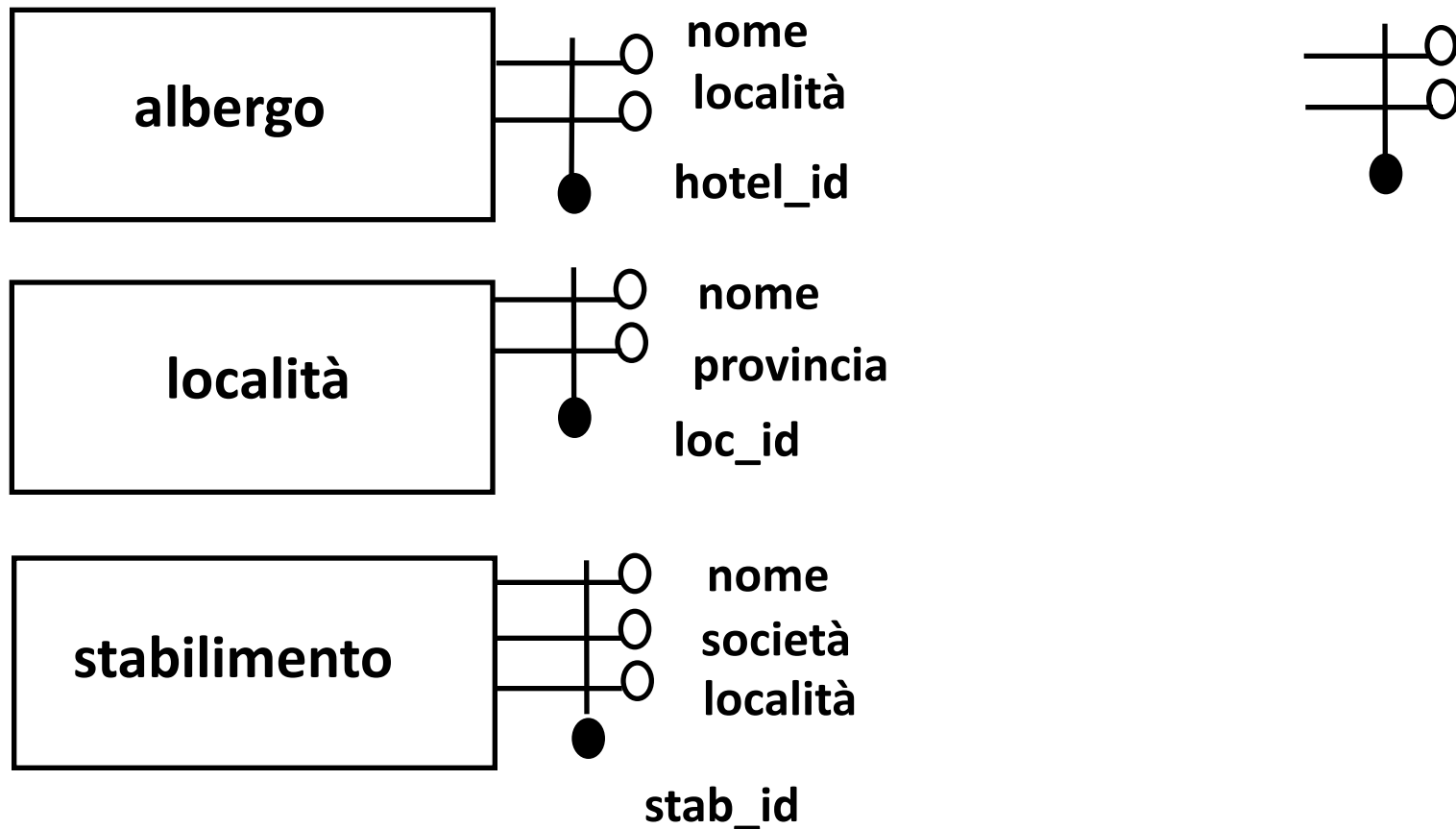
# Esempio





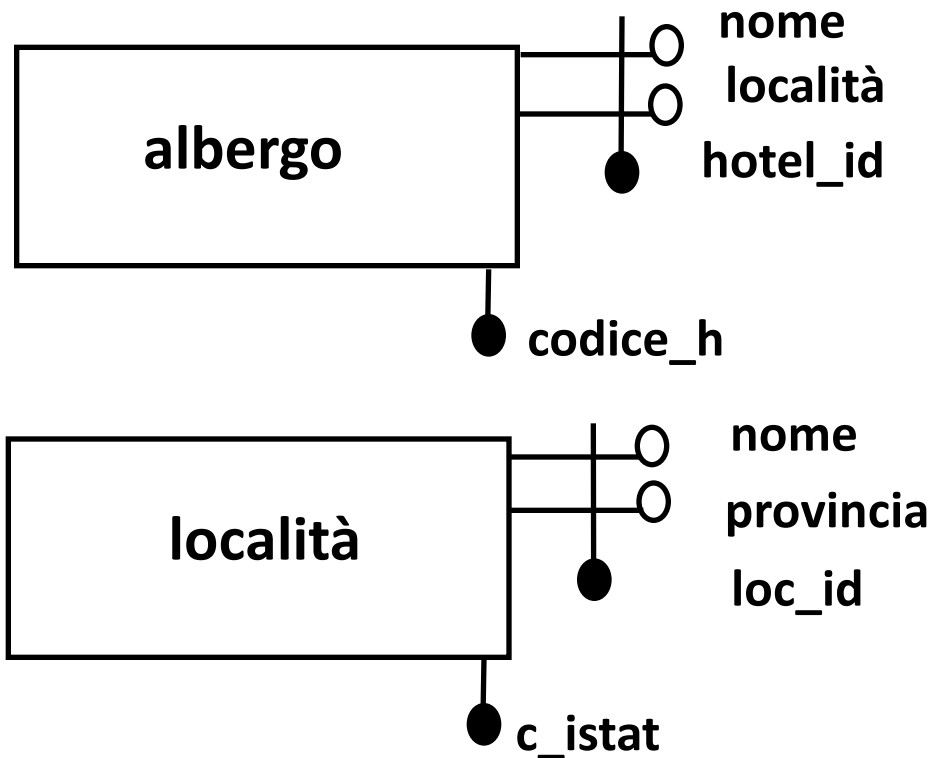
# Chiave composta

La chiave di un'entità può essere composta



# Chiavi alternative

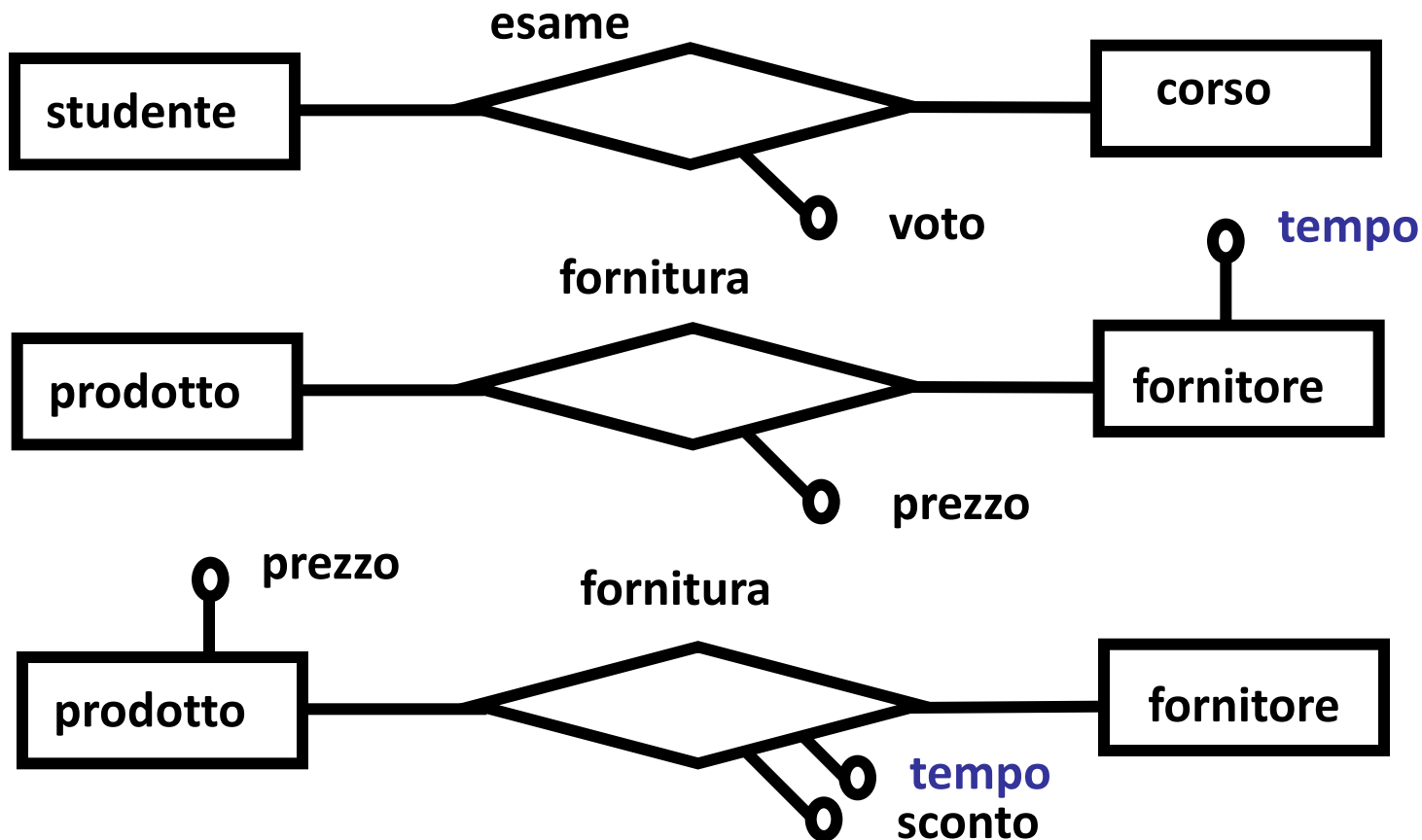
Chiavi **alternative** per una stessa entità



# Criteri di scelta della chiave

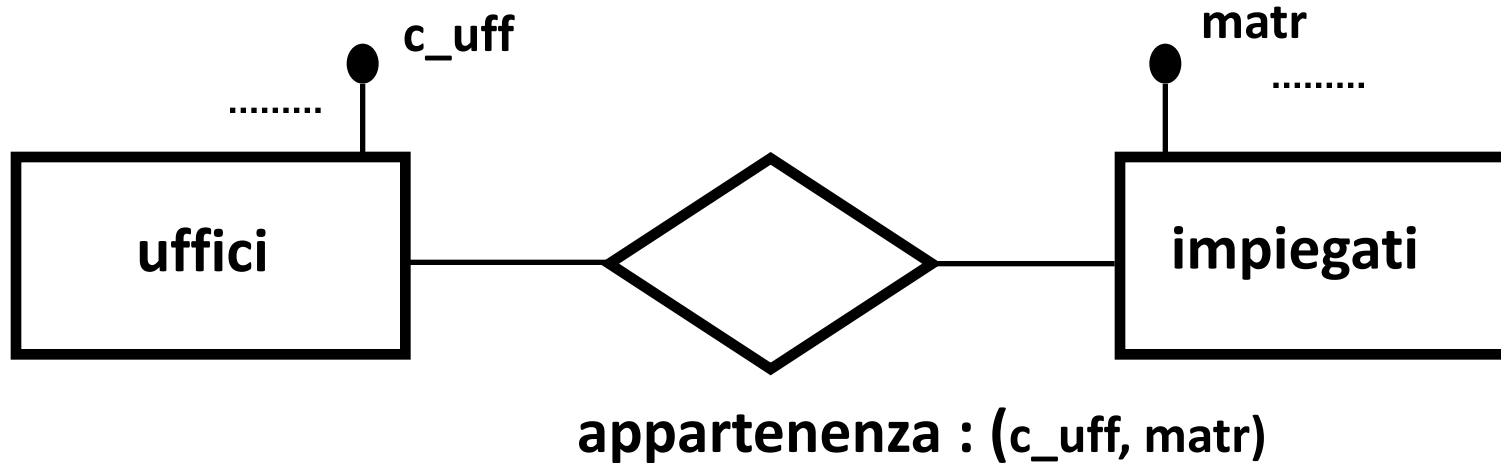
- Scelta condizionata da fattori esterni
  - es.:per gli studenti: codice fiscale o matricola?
  - velocità di digitazione (matricola)
  - controllo validità (codice fiscale)
- Spesso si fa uso di valori progressivi assegnati automaticamente dal sistema garantendo l'unicità (Ricordate AUTO\_INCREMENT)

# Proprietà delle associazioni

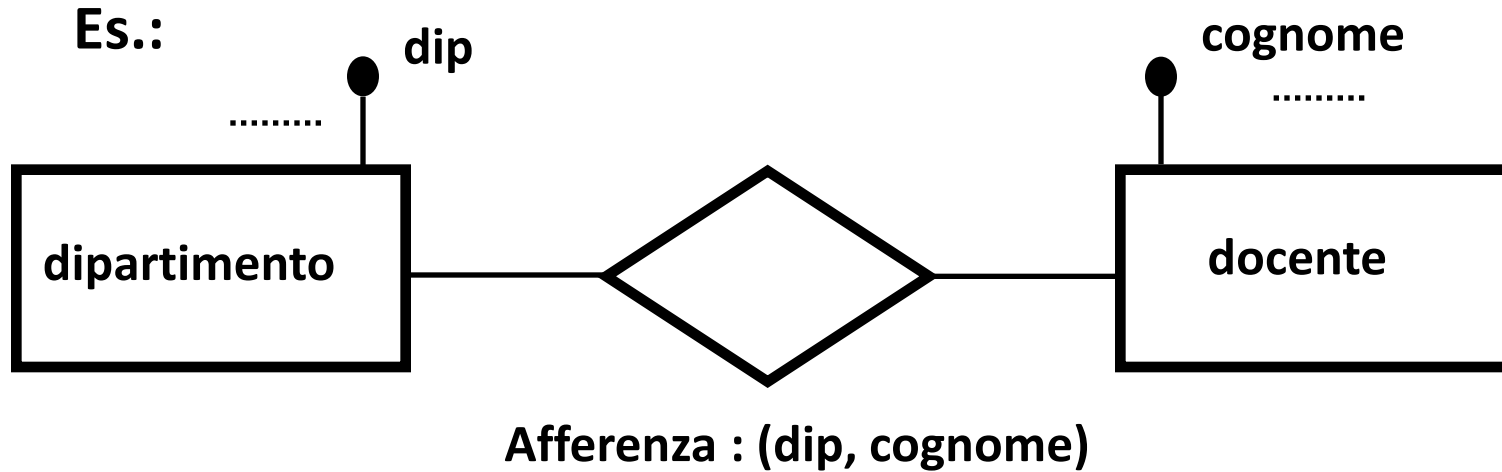


# La chiave delle associazioni

La chiave è **sempre composta** ed è definita come **l'insieme** delle chiavi delle entità partecipanti, es.:



# Esempio

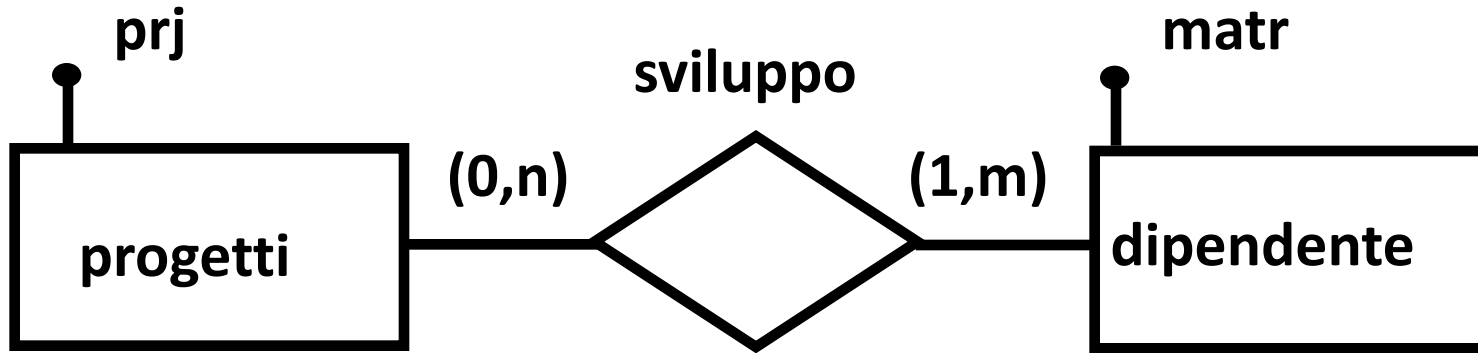


**ATTENZIONE:** la relazione eredita **sempre** le chiavi dalle entità partecipanti!  
Non si possono definire chiavi nelle associazioni!

# Cardinalità delle associazioni

- per cardinalità si intende il numero di volte che una data istanza di entità deve o può partecipare alla associazione
  - $(1,1)$  : obbligatoria, una sola volta
  - $(1,n)$  : obbligatoria, almeno una volta
  - $(0,1)$  : opzionale, una sola volta
  - $(0,n)$  : opzionale, n volte

# Associazioni N:M

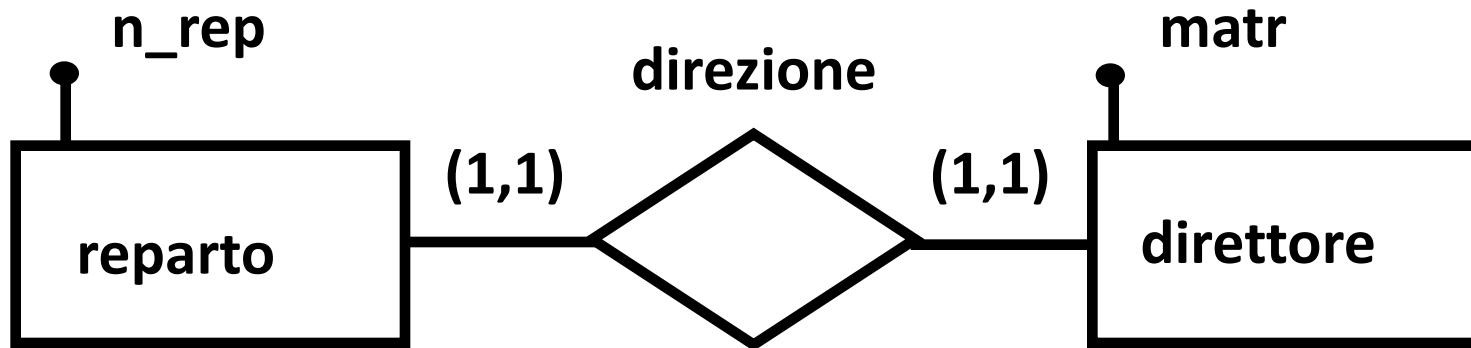


- **Con vincolo:** un dipendente **deve** partecipare ad almeno un progetto (**1**,m)
- **Senza vincolo:** ad un progetto **possono** partecipare dipendenti, ma può esistere anche un progetto senza dipendenti (**0**,n)

Chiamata anche: **associazione molti a molti**

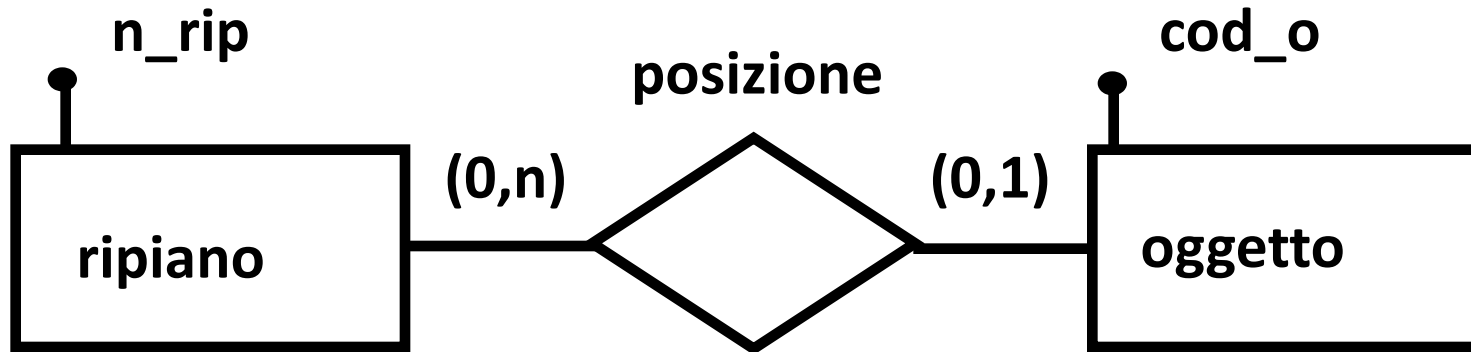


# Associazioni 1:1



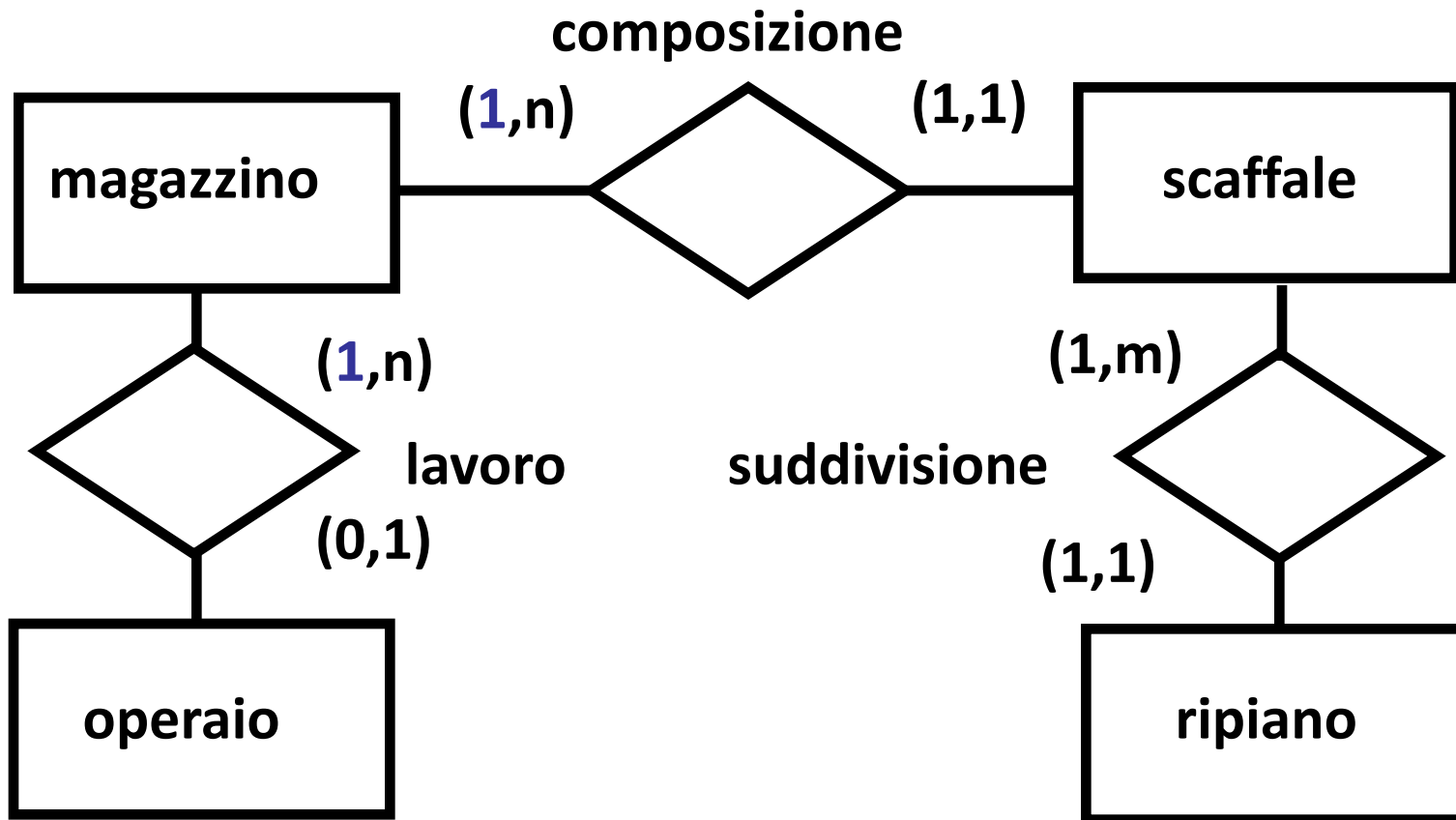
- Con doppio vincolo:
  - un reparto **deve** avere un direttore ed il direttore è **uno solo** (1,1)
  - un direttore **deve** dirigere uno ed **un solo** reparto (1,1)
- Il vincolo diventerebbe (0,1) se avessimo la generica entità “impiegato” al posto dell’entità “direttore” (alcuni impiegati non dirigono reparti)

# Associazioni N:1

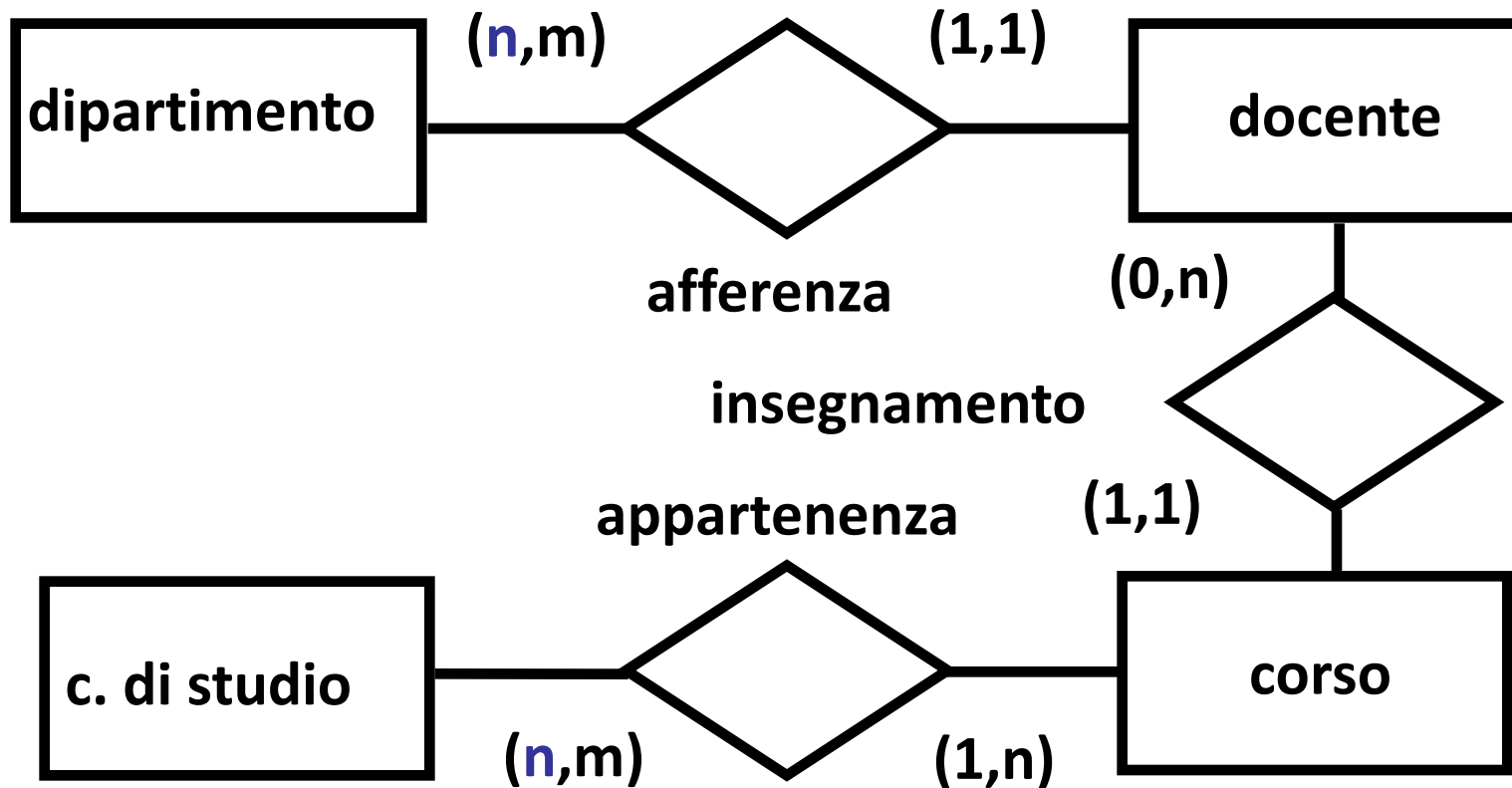


- **Con vincolo**: un oggetto **può** stare su un ripiano, ma su **un solo** ripiano (**0,1**)
- **Senza vincolo**: su un ripiano **possono** stare n oggetti
- È possibile vincolare la partecipazione, per esempio ponendo  $n = 5$ , **non più di** 5 oggetti per ripiano (**0,5**)
- Chiamata anche: associazione molti a uno

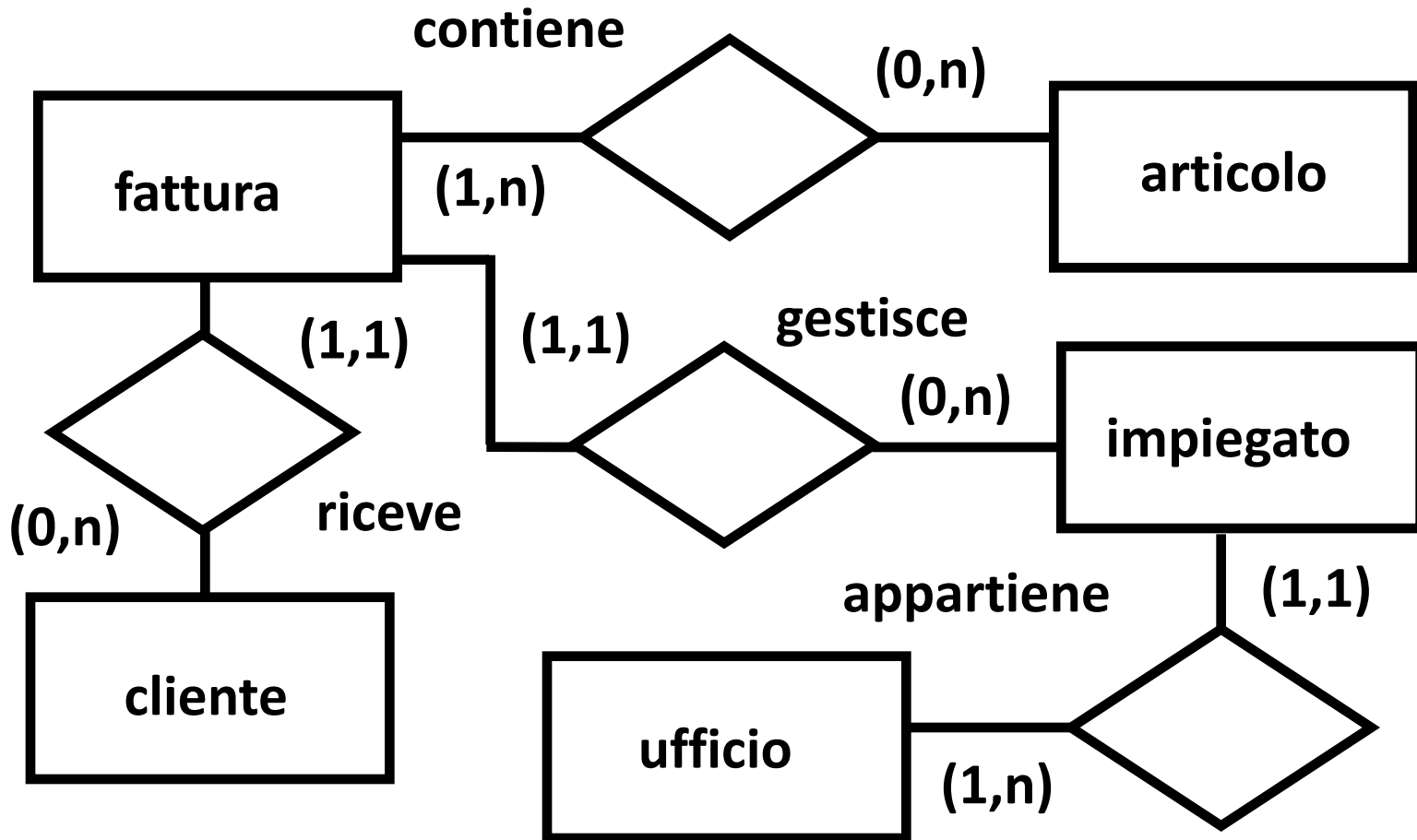
# Gestione magazzino



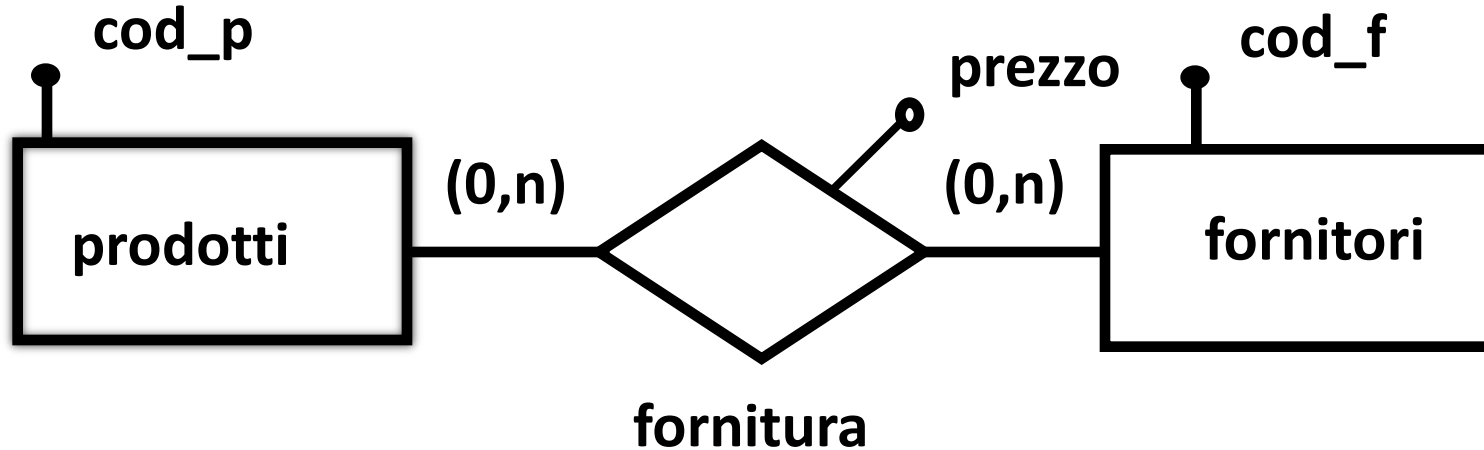
# Base dati universitaria



# Contabilità

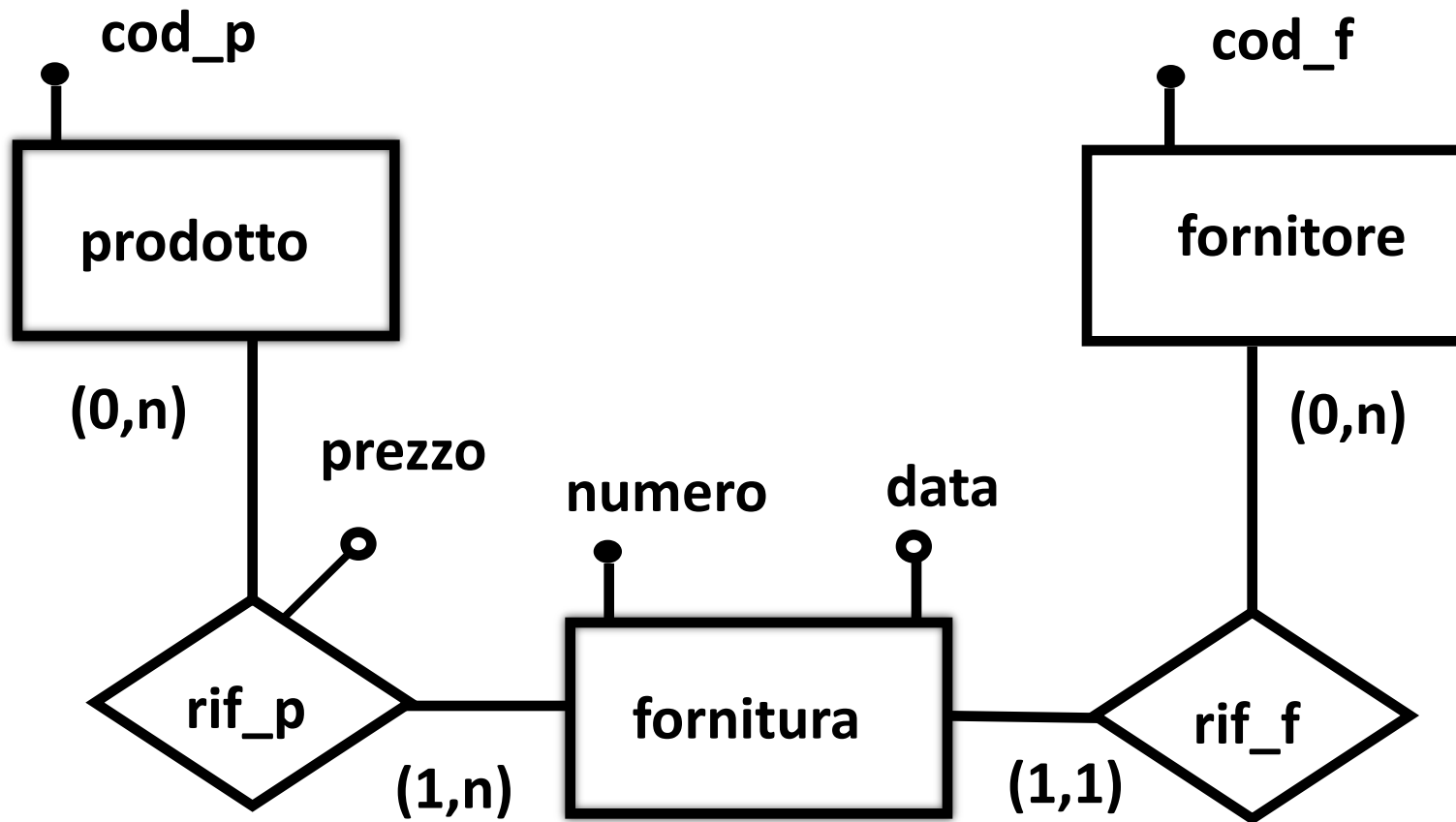


# Fornitura

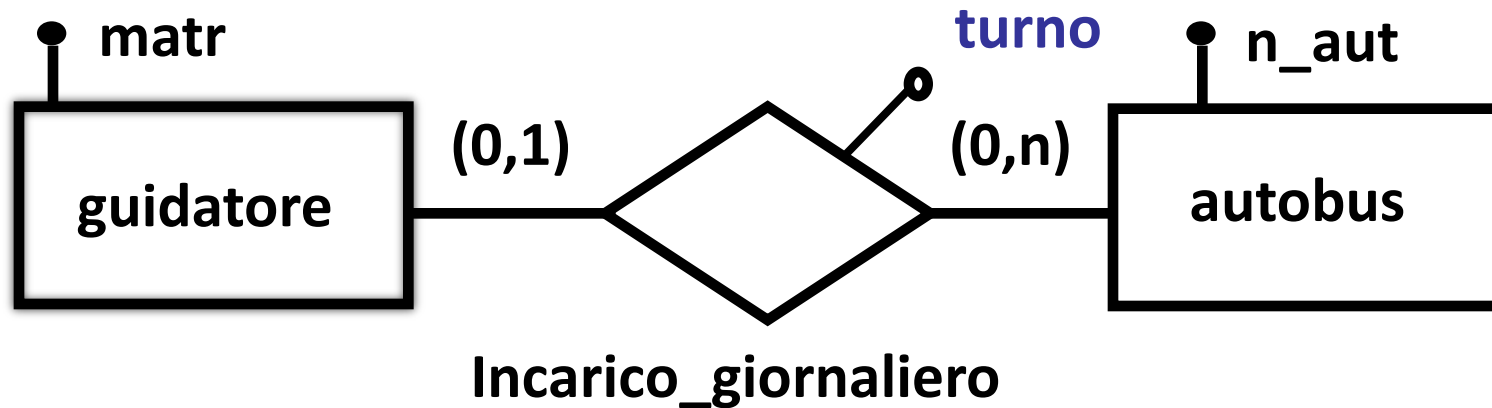


**NOTA:** questo modello non consente a un fornitore di fornire più volte lo stesso prodotto a seguito di ordini diversi perché si avrebbe una violazione dell'unicità delle chiavi

# Fornitura / 2



# Trasporti

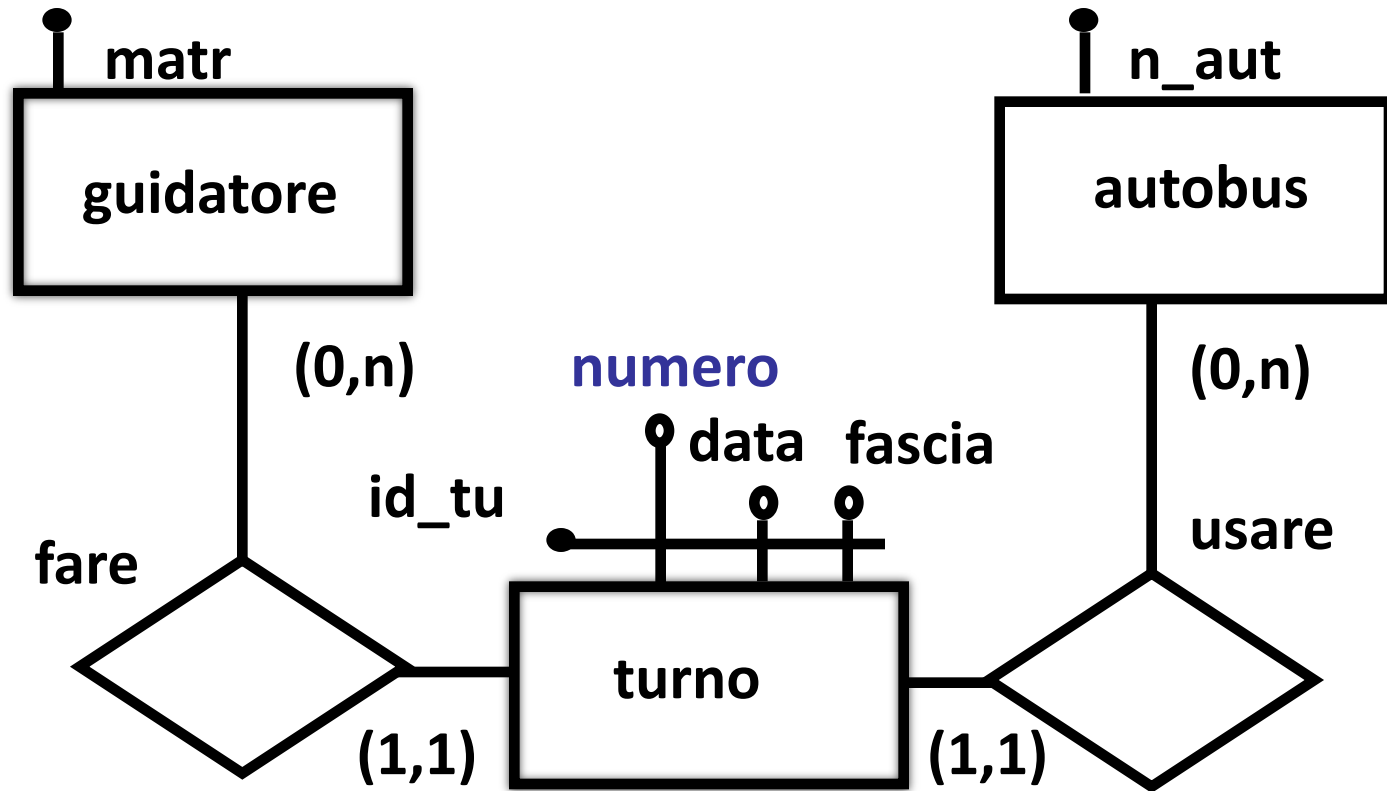


- **Attenzione all'interpretazione:**

questo è il programma giornaliero, in cui ogni guidatore è assegnato a un solo bus e fa un solo turno  
non rappresenta l'archivio dei turni svolti da un guidatore!



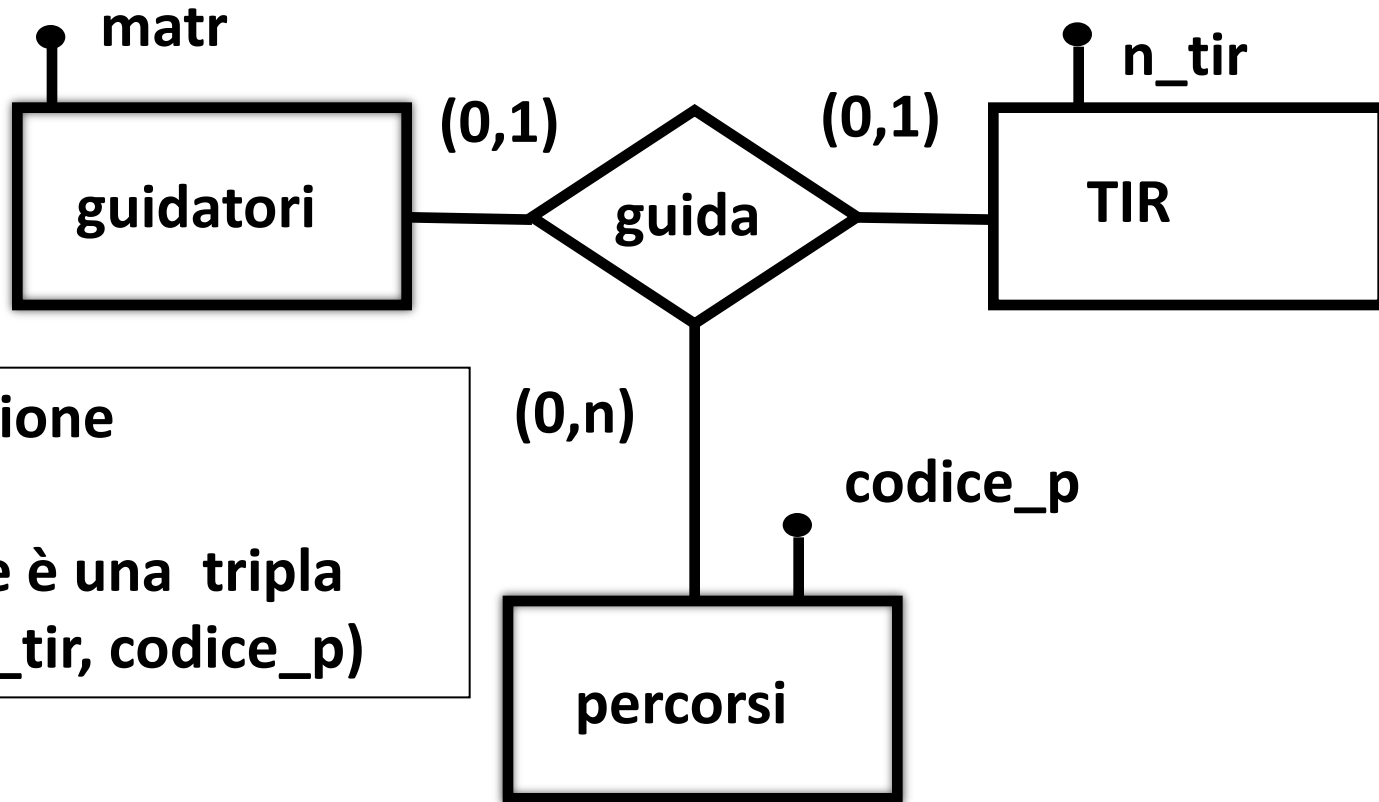
# Trasporti / 2



[Archivio assegnamenti:](#)

ci possono essere più turni nella stessa data e fascia oraria, svolti da guidatori diversi su bus diversi

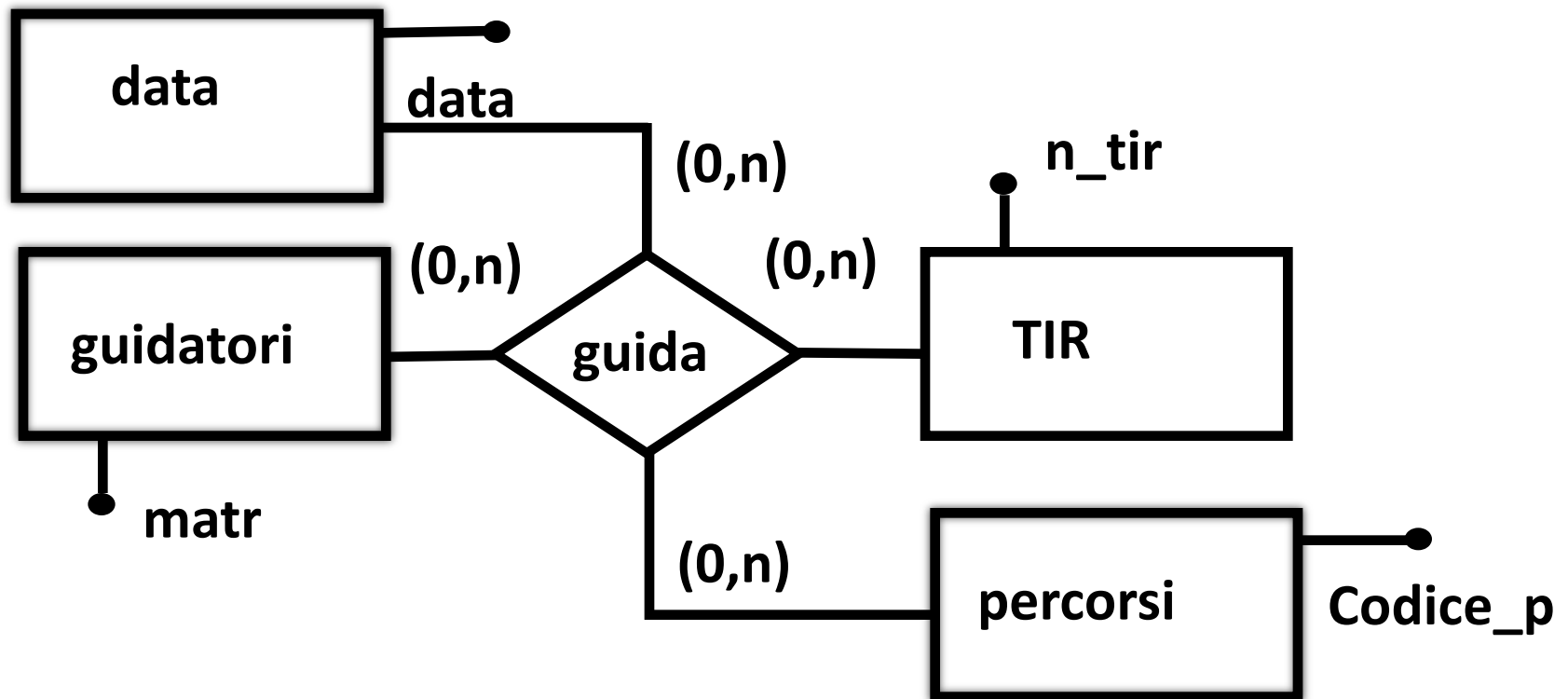
# Trasporti bis / 1



associazione  
ternaria  
la chiave è una tripla  
(matr, n\_tir, codice\_p)

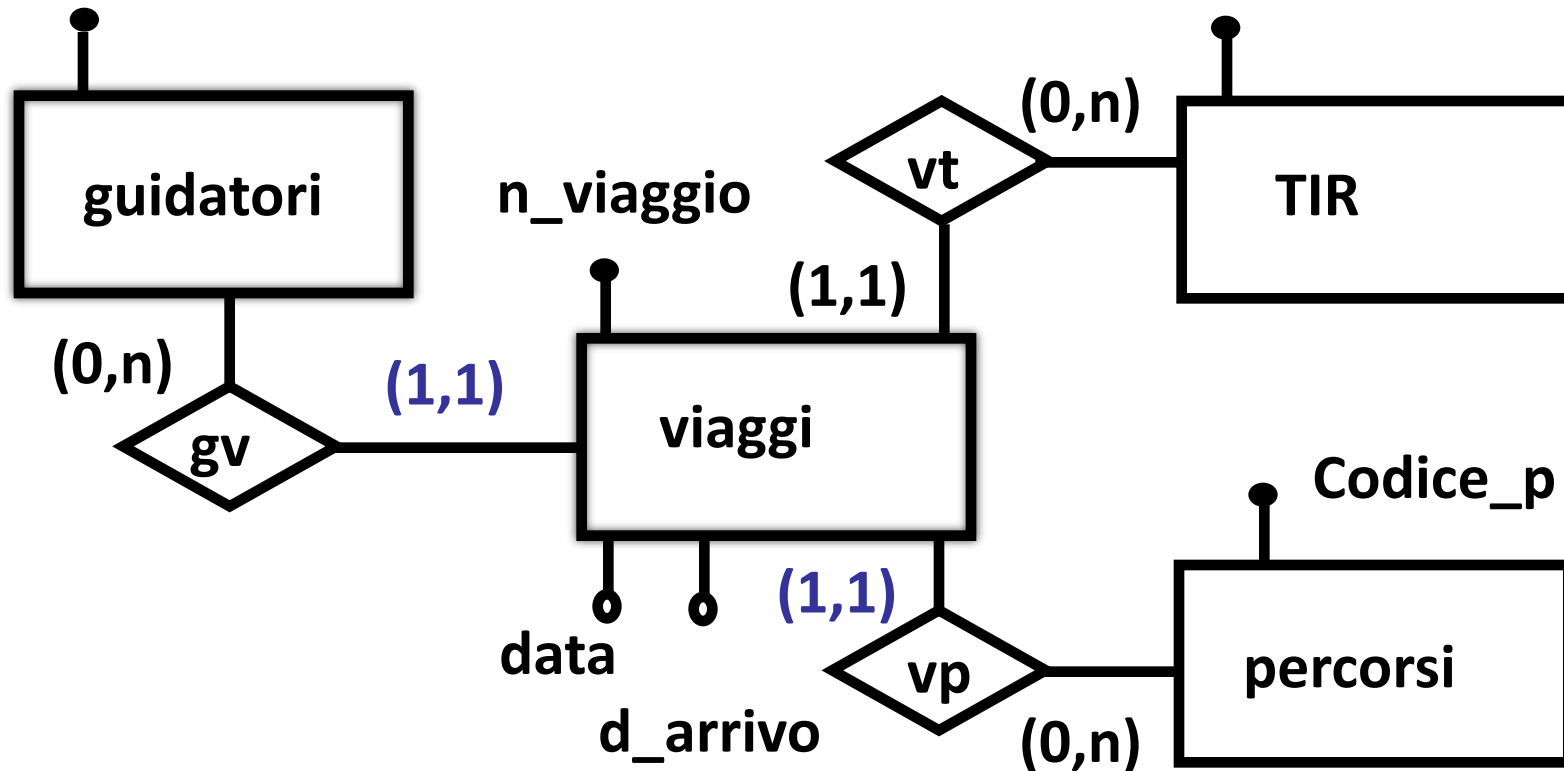
**NOTA:** descrive la situazione di un solo giorno, perché un guidatore non può condurre lo stesso TIR sullo stesso percorso più di una volta

# Trasporti bis / 2



- un guidatore può condurre lo stesso TIR sullo stesso percorso più di una volta, ma in date diverse
- questa rappresentazione nasconde la presenza di un'entità

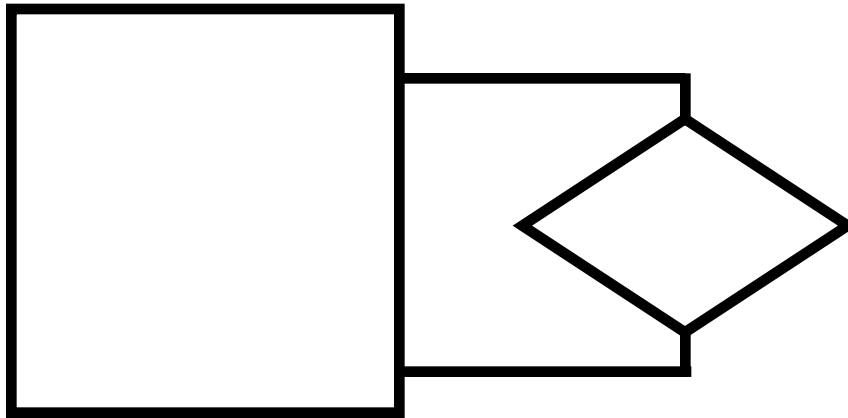
# Trasporti bis / 3



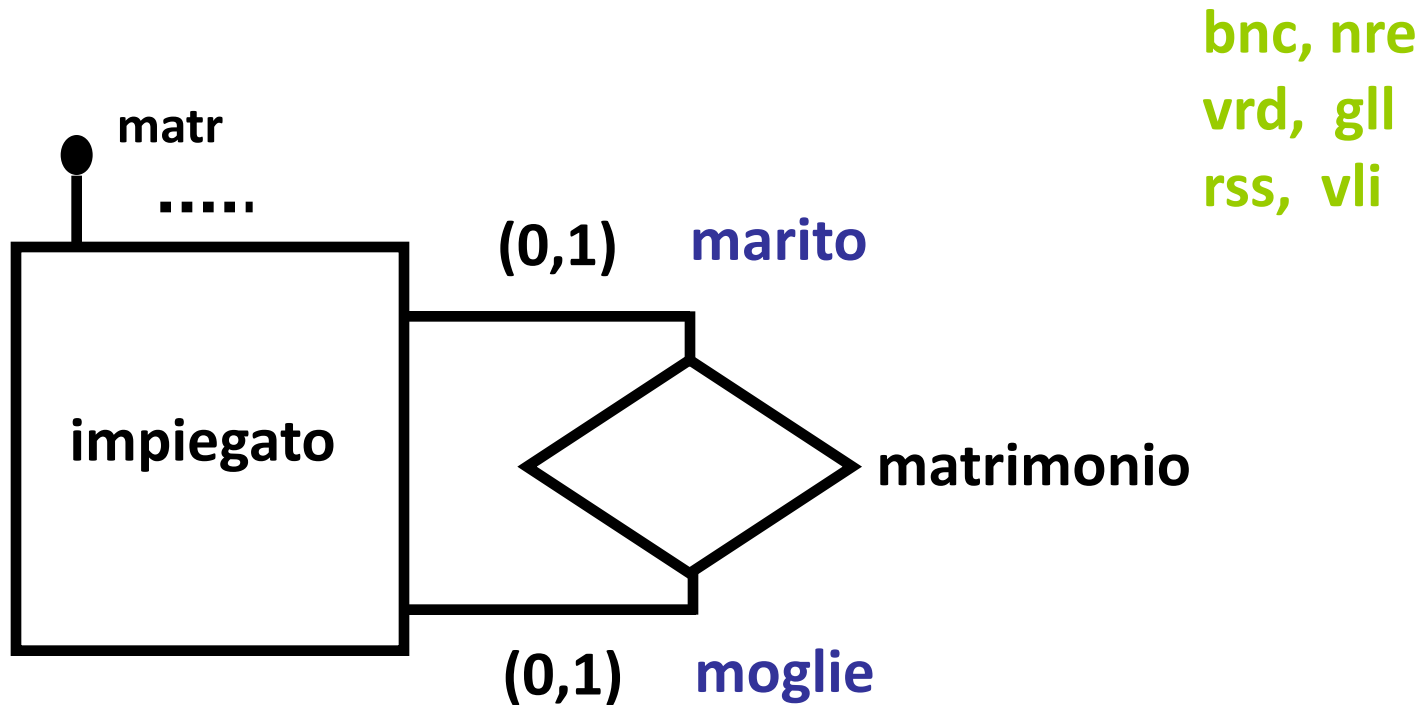
- Potremmo anche mettere  $(1,n)$  al posto di  $(1,1)$ , per esempio per rappresentare viaggi con più di un guidatore

# Le auto-associazioni

Associazioni aventi come partecipanti  
istanze provenienti dalla stessa entità  
(chiamate anche unarie o ad anello):

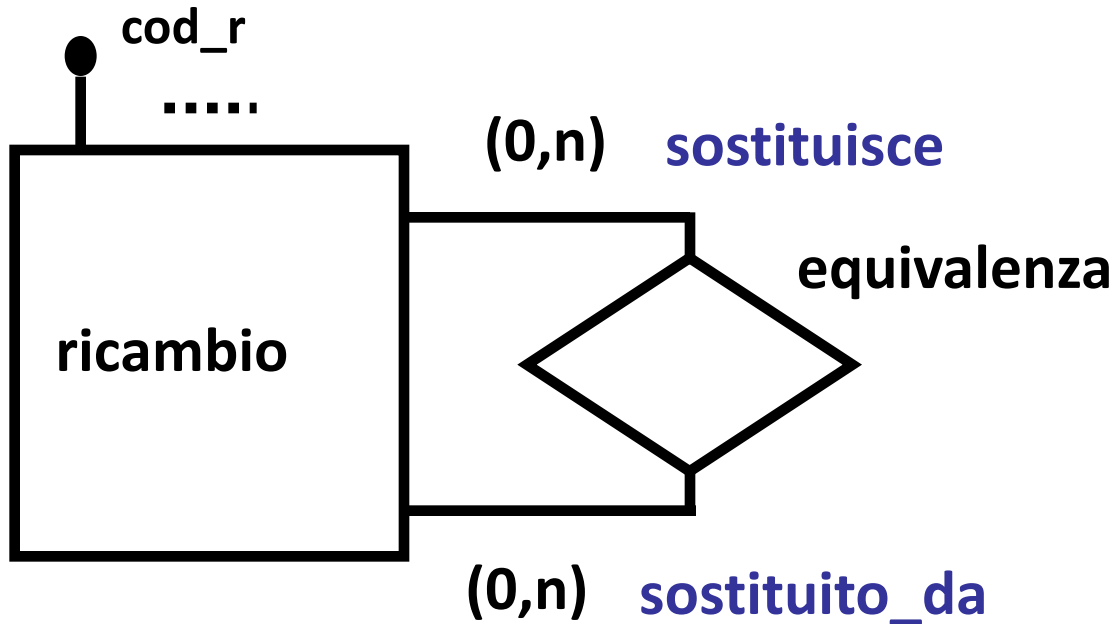


# Auto-associazioni 1:1



Sul ogni ramo può essere riportato il “**ruolo**”  
del partecipante all’associazione

# Auto-associazioni n:m



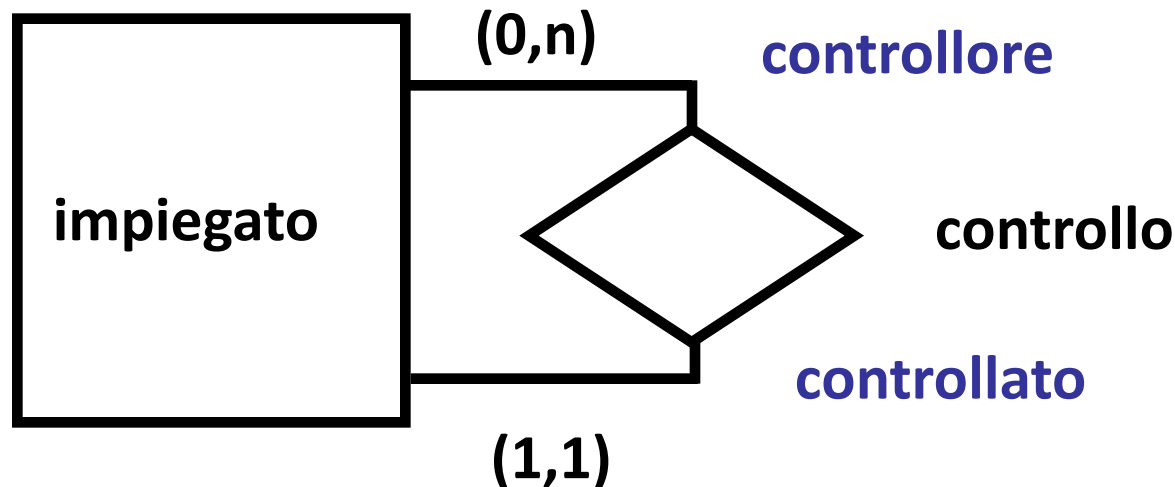
NB: può essere interpretata  
come **bi**- o **mono**-direzionale

p22	m89
k45	s56

p22	m89
k45	s56
m89	p22
s56	k45

# Auto-associazioni ricorsive

esempio: gerarchia (1:n)



grafo diretto aciclico (albero)