PROGETTAZIONE LOGICA

Progettazione logica

Obiettivo: "tradurre" lo schema concettuale in uno schema logico che rappresenti gli stessi dati in maniera corretta ed efficiente

Input:

- schema concettuale
- informazioni sul carico applicativo
- modello logico

Output:

- schema logico, con vincoli e documentazione associata

Non si tratta di una pura e semplice traduzione per due motivi:

- alcuni aspetti non sono direttamente o univocamente rappresentabili
- è necessario prestare attenzione alle prestazioni

Usuale articolazione:

- ristrutturazione dello schema E-R
- traduzione nel modello logico (ed eventuale successiva ristrutturazione)

Ristrutturazione dello schema E-R

Motivazioni:

- semplificazione (eliminazione dei costrutti per i quali non esiste una traduzione immediata)
- trasformazione sulla base dei parametri quantitativi

Nota bene: unso schema E-R ristrutturato non è (più) uno schema concettuale nel senso stretto del termine, perché rappresenta i dati tenendo presenti aspetti realizzativi (facilità di traduzione e prestazioni)

Analisi delle prestazioni (su schemi E-R)

- le prestazioni non sono valutabili con precisione (perché dipendono da fattori non noti: resto del carico del sistema, parametri fisici, caratteristiche del DBMS)
- consideriamo "indici di prestazioni" per spazio e tempo:

  spazio dimensione dei dati

  tempo numero di occorrenze (di entità e relationship) visitate

  la schematizzazione sul tempo (che utilizzeremo in particolare) è molto approssimativa

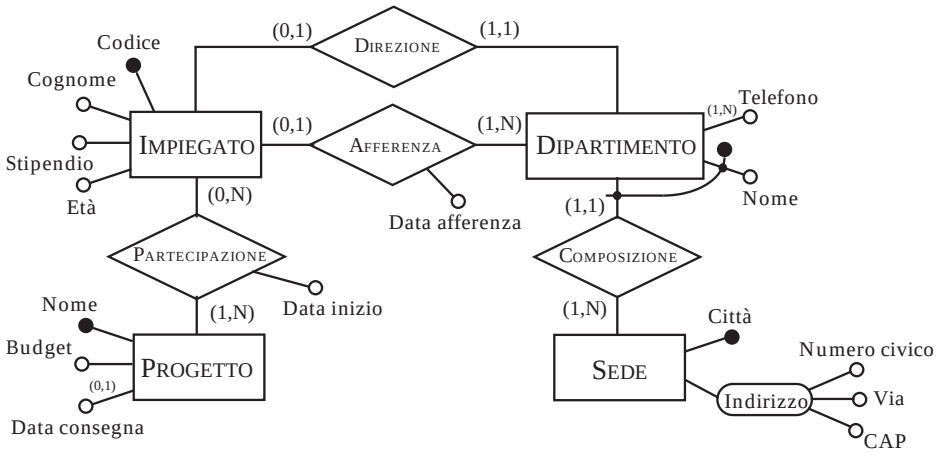
Informazioni sul carico applicativo

Volume dei dati

- numero (medio) di occorrenze di ogni entità e associazione dello schema,
- dimensioni di ciascun attributo (di entità o associazione)

Caratteristiche delle operazioni

- tipo dell'operazione (interattiva o batch),
- frequenza (numero medio di esecuzioni in un certo intervallo di tempo),
- dati (entità e/o associazioni) coinvolti: schema di operazione e tavola degli accessi



Operazione 1: assegna un impiegato a un progetto.

Operazione 2: trova i dati di un impiegato, del dipartimento nel quale lavora e dei progetti ai quali partecipa.

Operazione 3: trova i dati di tutti gli impiegati di un certo dipartimento.

Operazione 4: per ogni sede, trova i suoi dipartimenti con il cognome del direttore e l'elenco degli impiegati del dipartimento.

Tavola dei volumi

Concetto	Tipo	Volume
Sede	E	10
Dipartimento	E	80
Impiegato	E	2000
Progetto	E	500
Composizione	R	80
Afferenza	R	1900
Direzione	R	80
Partecipazione	R	6000

Tavola delle operazioni

Operazione	Tipo	Frequenza
Op. 1	I	50 al giorno
Op. 2	I	100 al giorno
Op. 3	I	10 al giorno
Op. 4	B	2 a settimana

Schema di operazione

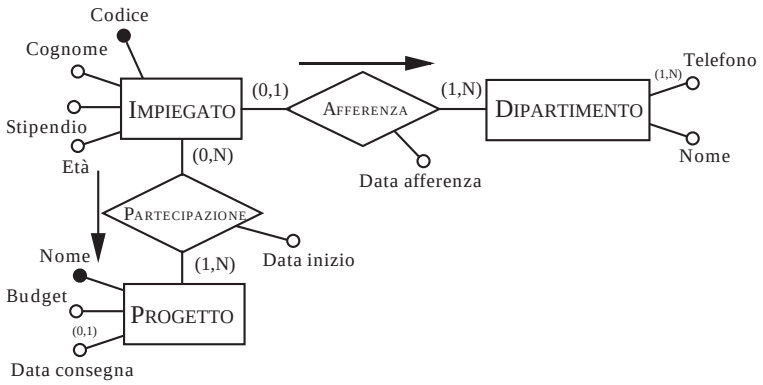


Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
Impiegato	Entità	1	L
Afferenza	Relazione	1	L
Dipartimento	Entità	1	L
Partecipazione	Relazione	3	L
Progetto	Entità	3	L

Ristrutturazione di schemi E-R

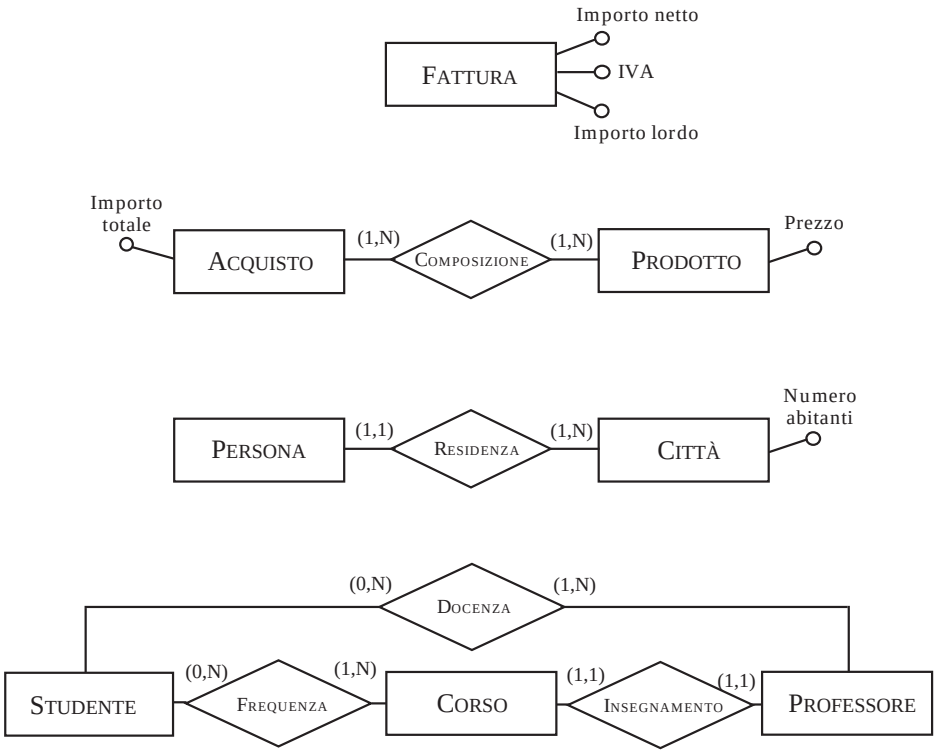
- Analisi delle ridondanze
- Eliminazione delle generalizzazioni
- Partizionamento/accorpamento di entità e associazioni
- Scelta degli identificatori primari

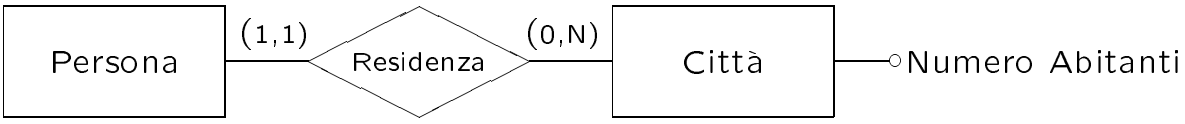
Analisi delle ridondanze

- a seconda delle metodologie seguite, possiamo avere schemi E-R che ammettono ridondanze oppure no
- in questa fase si decide se mantenere le ridondanze e/o se introdurne di nuove
- in uno schema concettuale, possiamo avere ridondanze per:
  - rappresentare informazioni significative ma derivate
- in uno schema logico, le ridondanze possono
  - semplificare le interrogazioni
  - appesantire gli aggiornamenti (sia come prestazioni sia come onere di programmazione) e occupare più spazio

Forme di ridondanza in uno schema E-R

- attributi derivabili, occorrenza per occorrenza, da altri attributi della stessa entità (o associazione)
- attributi derivabili da attributi di altre entità (o associazioni), di solito attraverso funzioni aggregative (somma, media, conteggio, ...)
- associazioni derivabili dalla composizione di altre associazioni in presenza di cicli





Operazione 1: memorizza una nuova persona con la relativa città di residenza.

Operazione 2: stampa tutti i dati di una città (incluso il numero di abitanti).

Tavola dei volumi

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

Tavola delle operazioni A

Operazione	Tipo	Frequenza
Op. 1	I	500 al giorno
Op. 2	I	10 al giorno

Tavola delle operazioni B

Operazione	Tipo	Frequenza
Op. 1	I	500 al giorno
Op. 2	I	1 al mese

Tavole degli accessi in presenza di ridondanza

Operazione 1			
Concetto	**Costr.**	**Acc.**	**Tipo**
Persona	E	1	S
Residenza	R	1	S
Città	E	1	L
Città	E	1	S

Operazione 2			
Concetto	**Costr.**	**Acc.**	**Tipo**
Città	E	1	L

Tavole degli accessi in assenza di ridondanza

Operazione 1			
Concetto	**Costr.**	**Acc.**	**Tipo**
Persona	E	1	S
Residenza	R	1	S

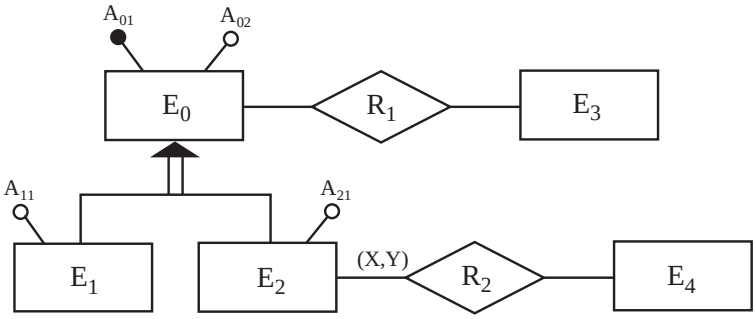
Operazione 2			
Concetto	**Costr.**	**Acc.**	**Tipo**
Città	E	1	L
Residenza	R	5000	L

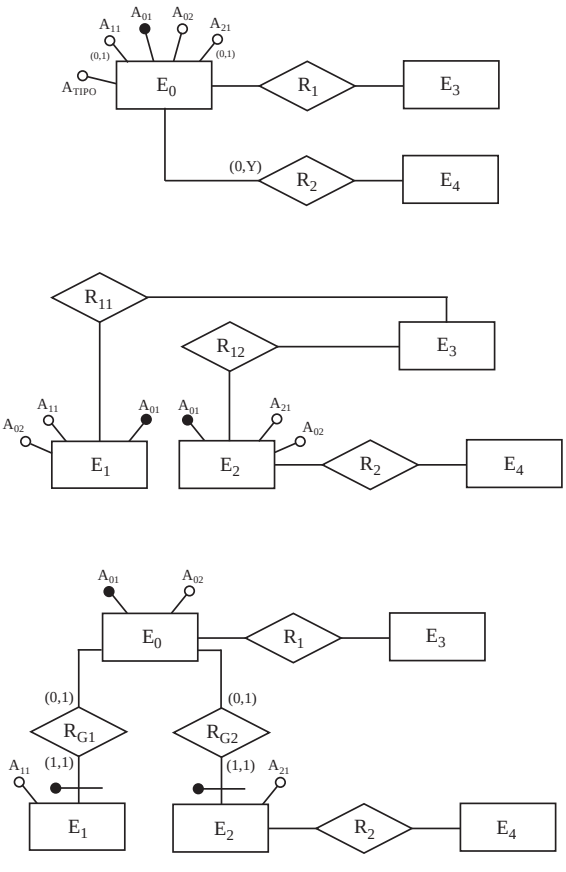
Eliminazione delle gerarchie

- il modello relazionale (come gli altri modelli logici) non ha costrutti per rappresentare direttamente le gerarchie
- entità e relationship sono direttamente rappresentabili
- conviene eliminare le gerarchie, sostituendole 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 associazioni


A01
A02
E0
R1
E3
A11
A21
E1
E2
(X,Y)
R2
E4


A11
A01
A02
A21
(0,1)
(0,1)
ATIPO
E0
R1
E3
(0,Y)
R2
E4
R11
R12
E3
A11
A01
A01
A21
A02
A02
E1
E2
R2
E4
A01
A02
E0
R1
E3
(0,1)
(0,1)
RG1
RG2
A11
(1,1)
(1,1)
A21
E1
E2
R2
E4

- la scelta fra le alternative si può fare con metodo simile a quello visto per l'analisi delle ridondanze (attenzione però: non è sufficiente far riferimento al numero di accessi)

- semplici regole generali:

  (1) conviene se gli accessi agli attributi di E1 e di E0 (nonché di E2 e E0) sono solitamente contestuali e le operazioni accedono sia alle occorrenze di E1 sia a quelle di E2

  (2) è possibile solo se la generalizzazione è completa e conviene se gli accessi a E1 sono distinti da quelli a E2

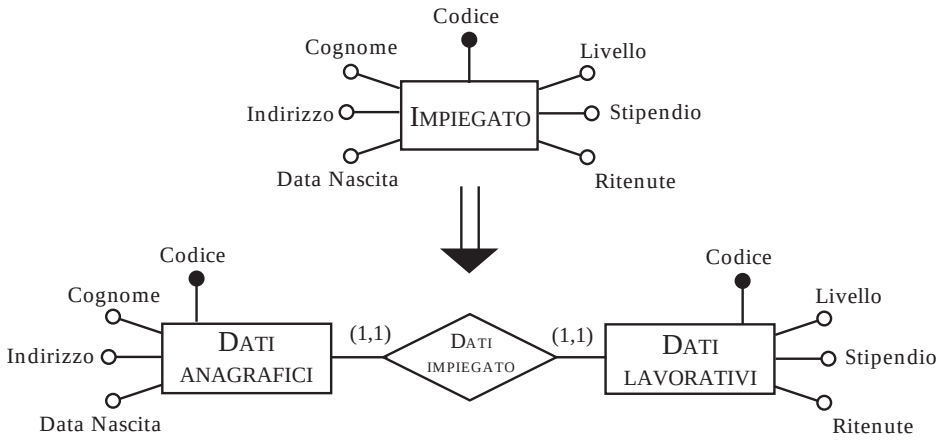
  (3) conviene se gli accessi alle entità figlie sono separati dagli accessi al genitore

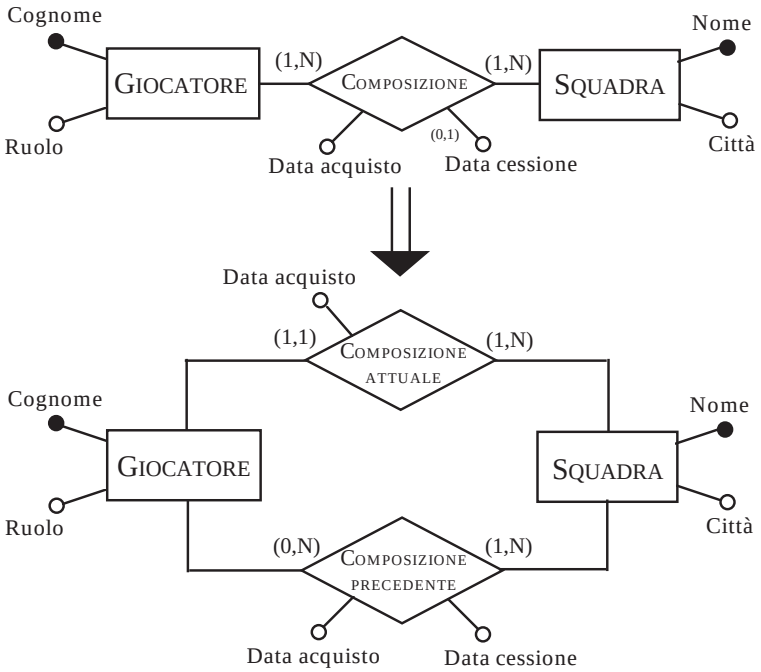
- sono possibili soluzioni interemdie

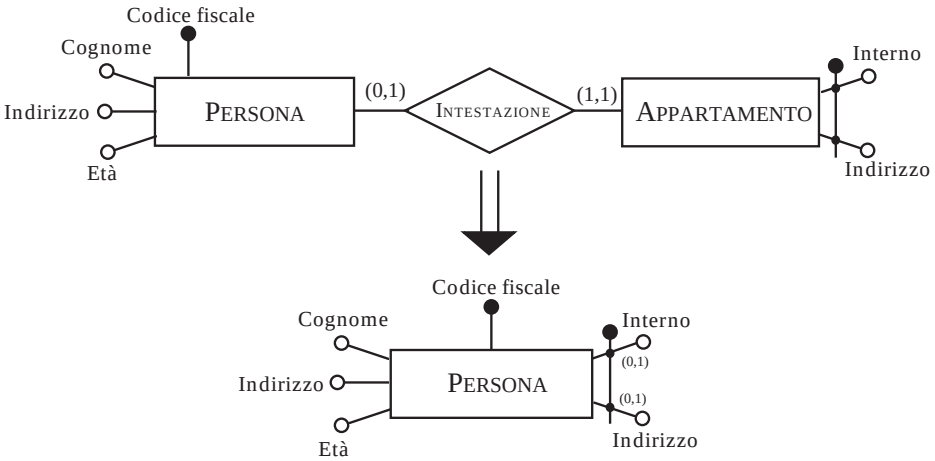
- nel caso di gerarchie a più livelli, le scelte possono essere diverse ai vari livelli

Partizionamento/accorpamento di concetti

- si basa su criteri simili a quelli utilizzati per le generalizzazioni
- esempi:
  - partizionamento (verticale) di entità
  - partizionamento (orizzontale) di relationship
  - accorpamento di entità




Cognome
Nome
GIOCATORE
(1,N)
COMPOSIZIONE
(1,N)
SQUADRA
Ruolo
(0,1)
Città
Data acquisto
Data cessione
Data acquisto
(1,1)
COMPOSIZIONE ATTUALE
(1,N)
Cognome
Nome
GIOCATORE
SQUADRA
Ruolo
Città
(0,N)
COMPOSIZIONE PRECEDENTE
(1,N)
Data acquisto
Data cessione


Codice fiscale
Cognome
Interno
Indirizzo
PERSONA
(0,1)
INTESTAZIONE
(1,1)
APPARTAMENTO
Età
Indirizzo
Codice fiscale
Cognome
Interno
(0,1)
Indirizzo
PERSONA
(0,1)
Età
Indirizzo

Scelta degli identificatori principali

- è indispensabile nella progettazione nel modello relazionale
- in altri contesti si tratta di una scelta legata alla struttura fisica (alla chiave primaria viene associato un cammino d'accesso privilegiato)

Criteri per la scelta:

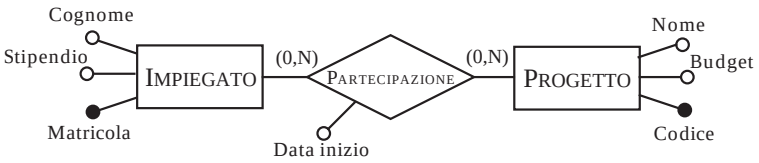
- assenza di valori nulli
- semplicità
- utilizzo nelle operazioni più frequenti o importanti
- preferenza per gli identificatori interni

Traduzione verso il modello relazionale

idea fondamentale:

- le entità diventano relazioni (sugli stessi attributi)
- le relationship diventano relazioni sulle chiavi delle relazioni che rappresentano le entità coinvolte (più gli attributi propri)

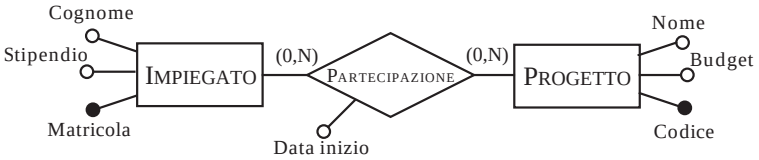
Entità e relationship molti a molti



Impiegato(<u>**Matricola**</u>, **Cognome**, **Stipendio**)
Progetto(<u>**Codice**</u>, **Nome**, **Budget**)
Partecipazione(<u>**Matricola**, **Codice**</u>, **DataInizio**)

con vincoli di integrità referenziale tra gli attributi **Matricola** e **Codice** di **Partecipazione** e le chiavi di **Impiegato** e **Progetto**

- può essere utile ridenominare gli attributi della chiave della relazione che rappresenta la relationship

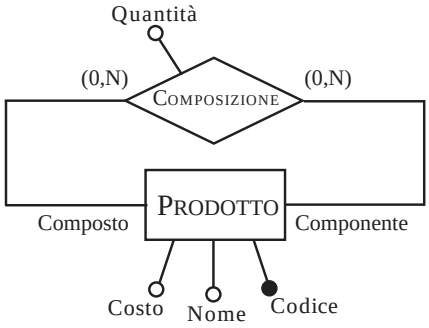


Impiegato(<u>**Matricola**</u>, **Cognome**, **Stipendio**)
Progetto(<u>**Codice**</u>, **Nome**, **Budget**)
Partecipazione(<u>**Impiegato**, **Progetto**</u>, **DataInizio**)

invece di

Partecipazione(<u>**Matricola**, **Codice**</u>, **DataInizio**)

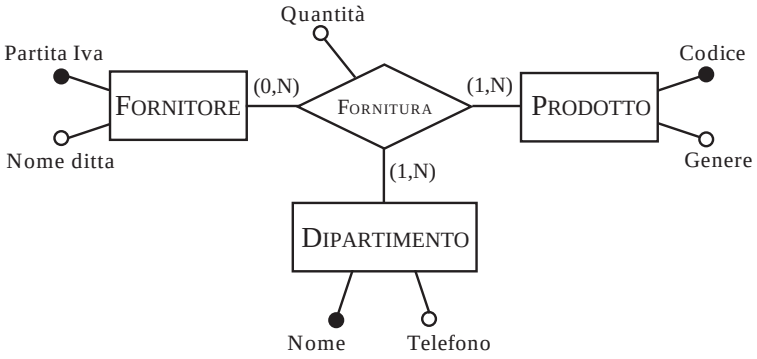
- la ridenominazione è essenziale per le relationship ricorsive:



Prodotto(<u>Codice</u>, Nome, Costo)
Composizione(<u>Composto, Componente</u>, Quantità)

con due vincoli di integrità referenziale

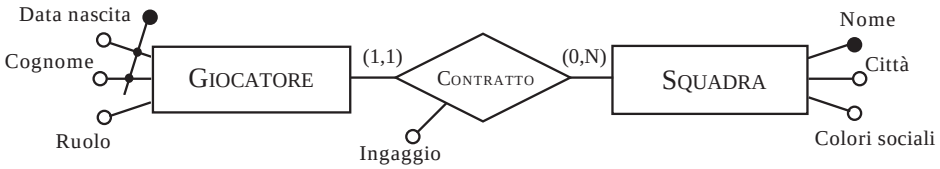
Relationship ternarie



Fornitore(<u>PartitaIVA</u>, NomeDitta)
Prodotto(<u>Codice</u>, Genere), Dipartimento(<u>Nome</u>, Telefono)
Fornitura(<u>Fornitore, Prodotto, Dipartimento</u>, Quantità).

con tre vincoli di integrità referenziale

Relationship uno a molti



- traduzione standard:

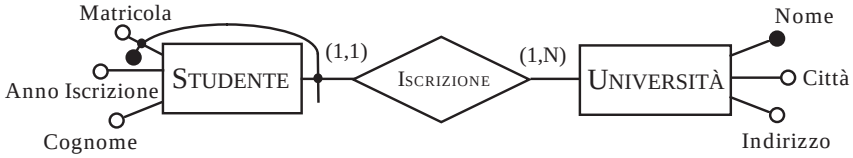
Giocatore(<u>**Cognome**, **DataNascita**</u>, **Ruolo**)
Contratto(<u>**Giocatore**, **DataNascitaGiocatore**</u>, **NomeSquadra**, **Ingaggio**)
Squadra(<u>**Nome**</u>, **Città**, **ColoriSociali**)

- le prime due relazioni hanno la stessa chiave; è possibile fonderle:

Giocatore(<u>**Cognome**, **DataNascita**</u>, **Ruolo**, **NomeSquadra**, **Ingaggio**)
Squadra(<u>**Nome**</u>, **Città**, **ColoriSociali**)

Entità con identificatore esterno

- sono coinvolte in relationship uno a molti
- si traducono con relazioni che contengono (come parte della chiave) anche la chiave della relazione identificante; si rappresenta così anche la relationship

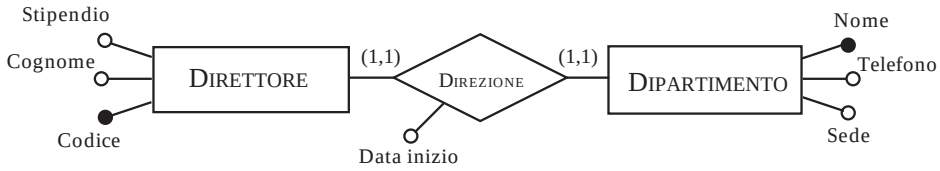


Studente(<u>**Matricola**, **NomeUniversità**</u>, **Cognome**, **AnnoIscrizione**)
Università(<u>**Nome**</u>, **Città**, **indirizzo**)

- con vincolo di integrità referenziale tra l'attributo **NomeUniversità** della relazione **Studente** e l'attributo **Nome** della relazione **Università**

Relationship uno a uno

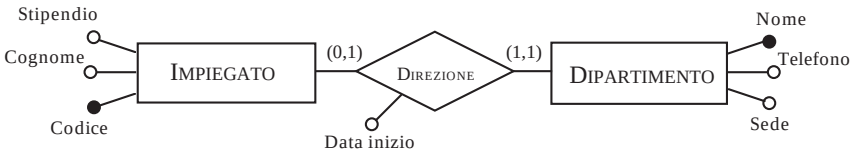
- varie possibilità, anche sulla base delle cardinalità minime; ad esempio



Direttore(<u>Codice</u>, Cognome, Stipendio, DipartimentoDiretto, InizioDirezione)
Dipartimento(<u>Nome</u>, Telefono, Sede)

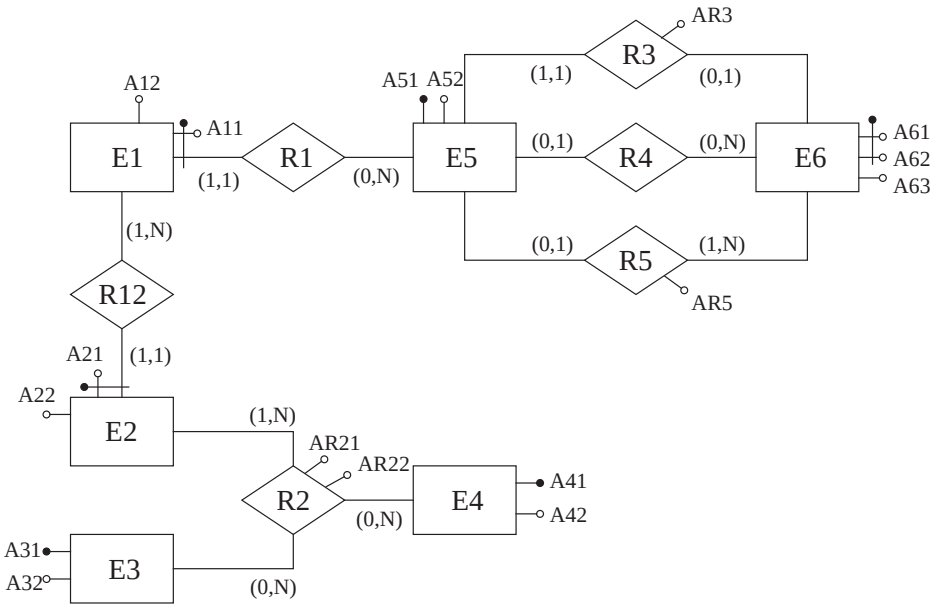
Direttore(<u>Codice</u>, Cognome, Stipendio)
Dipartimento(<u>Nome</u>, Telefono, Sede, Direttore, InizioDirezione)

- fondere tutto?



- una possibilità privilegiata:

Impiegato(<u>Codice</u>, Cognome, Stipendio)
Dipartimento(<u>Nome</u>, Telefono, Sede, Direttore, InizioDirezione)



E1(<u>A11, A51</u>, A12) E2(<u>A21, A11, A51</u>, A22)
E3(<u>A31</u>, A32) E4(<u>A41</u>, A42),
E5(<u>A51</u>, A52, A61R3, A62R3, AR3, A61R4, A62R4, A61R5, A62R5, AR5)
E6(<u>A61, A62</u>, A63) R2(<u>A21, A11, A51, A31, A41</u>, AR21, AR22)