

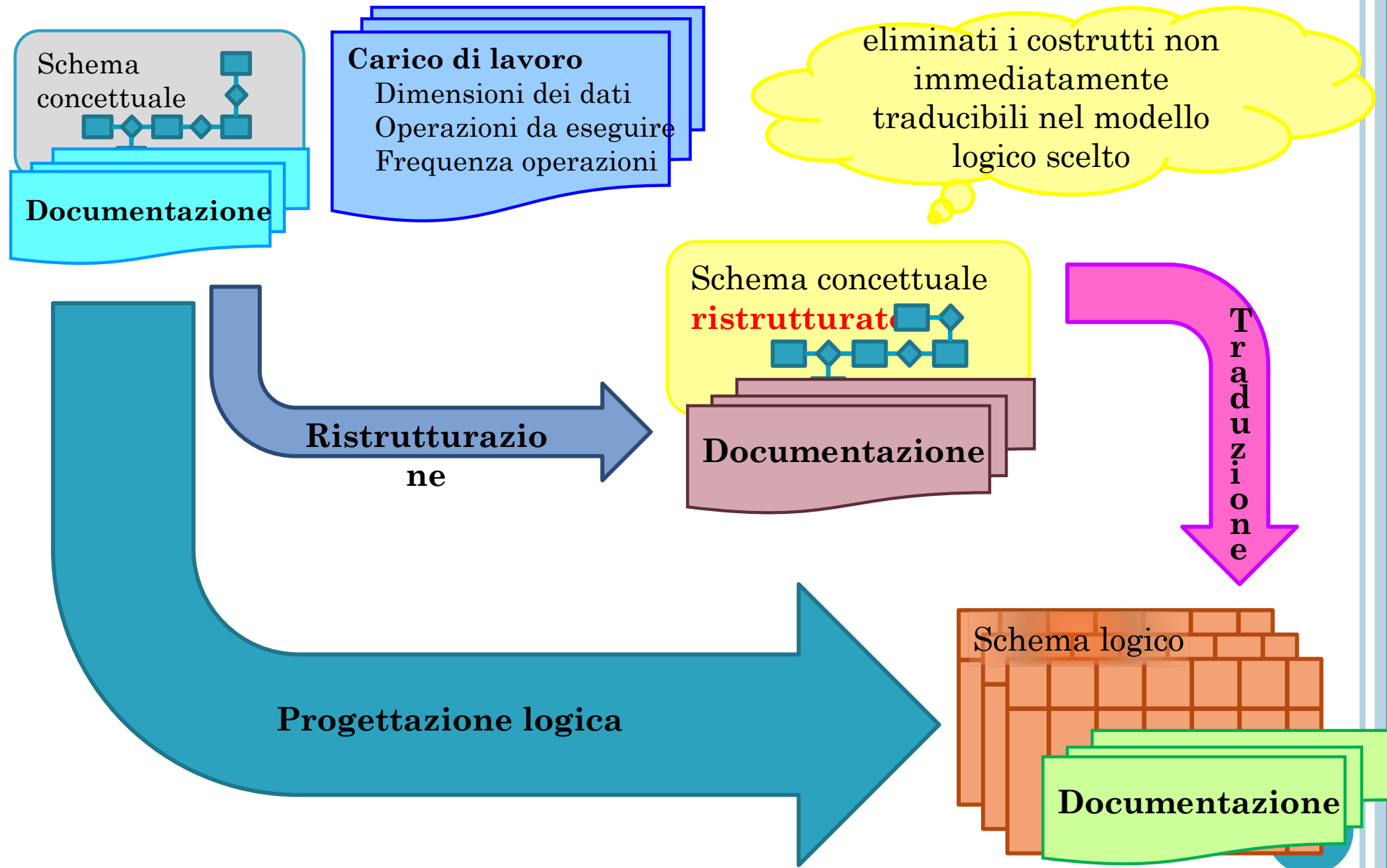


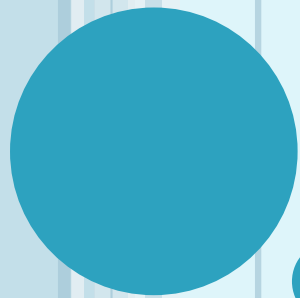
PROGETTAZIONE LOGICA

PROGETTAZIONE LOGICA

- Progettazione concettuale
 - input: specifica informale dei dati
 - output: schema concettuale
 - indipendente da ogni considerazione implementativa
 - **obiettivo primario**: rappresentazione non ambigua dei dati
- Progettazione logica
 - input: schema concettuale dei dati + **informazioni sul carico atteso (+ scelta del DBMS)**
 - output: schema logico per il DBMS prescelto
 - equivalente allo schema concettuale
 - ottimizzato per (lo specifico DBMS) e l'uso atteso
 - **obiettivo primario**: rappresentazione dei dati focalizzata alla realizzazione della base di dati e delle relative applicazioni

PROGETTAZIONE LOGICA - SCHEMA DEL PROCESSO





FASE DI RISTRUTTURAZIONE

4

Progettazione logica

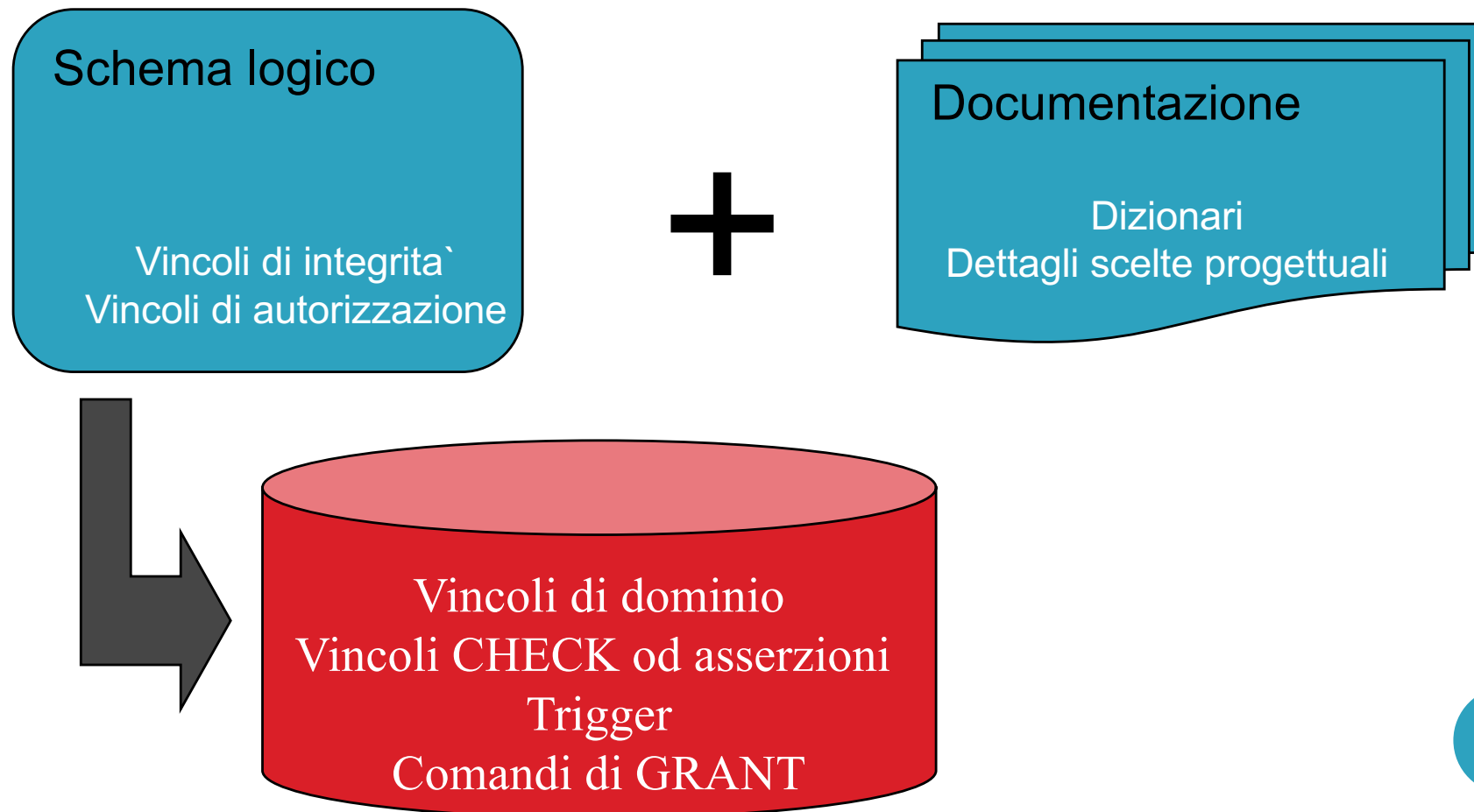
FASE DI RISTRUTTURAZIONE

- Genera lo **schema ER ristrutturato**
 - schema ER semplificato
 - equivalente a quello di partenza
- Scopo: semplificare la traduzione successiva
 - eliminazione dallo schema ER dei costrutti non **direttamente** rappresentabili nel modello relazionale
 - ristrutturazioni guidate da aspetti prestazionali
 - identificati dall'analisi del carico di lavoro
- Traduzione non sempre univoca
 - esigenze contrastanti
 - scelte del progettista sulla base della rilevanza attribuita alle singole esigenze

FASE DI TRADUZIONE

- Traduce lo schema ER ristrutturato in uno schema relazionale *equivalente*
 - traduzione basata su regole predefinite
 - in molti casi si possono applicare più regole
- Traduzione non sempre univoca
 - le scelte devono essere guidate da considerazioni prestazionali

PROGETTAZIONE LOGICA - OUTPUT



RISTRUTTURAZIONE

- Eliminazione dallo schema ER dei costrutti non direttamente rappresentabili nel modello relazionale
 - attributi composti
 - attributi con molteplicità > 1
 - generalizzazioni
- Ristrutturazioni per migliorare le prestazioni, suggerite dall'analisi del carico di lavoro
 - analisi della ridondanza
 - partizionamento/accorpamento di entità

ANALISI DELLA RIDONDANZA

- Ridondanza = la stessa informazione
 - viene rappresentata esplicitamente nello schema
 - può essere derivata da altre informazioni presenti nello schema
- Esempi
 - presenza di cicli tra le associazioni
 - presenza di attributi il cui valore può essere derivato da altri attributi ed/od associazioni
- Eliminare la ridondanza permette di semplificare lo schema ER ed il corrispondente schema relazionale

LA RIDONDANZA È UN MALE ASSOLUTO?

○ Svantaggi

- maggiore occupazione di spazio
- appesantimento delle procedure di aggiornamento

○ Vantaggi

- **può** rendere più efficienti **alcune** interrogazioni descritte nel carico di lavoro
 - se
 1. le interrogazioni vengono eseguite molto più frequentemente degli aggiornamenti
 2. le informazioni ridondanti sono di dimensione contenuta può essere giustificata

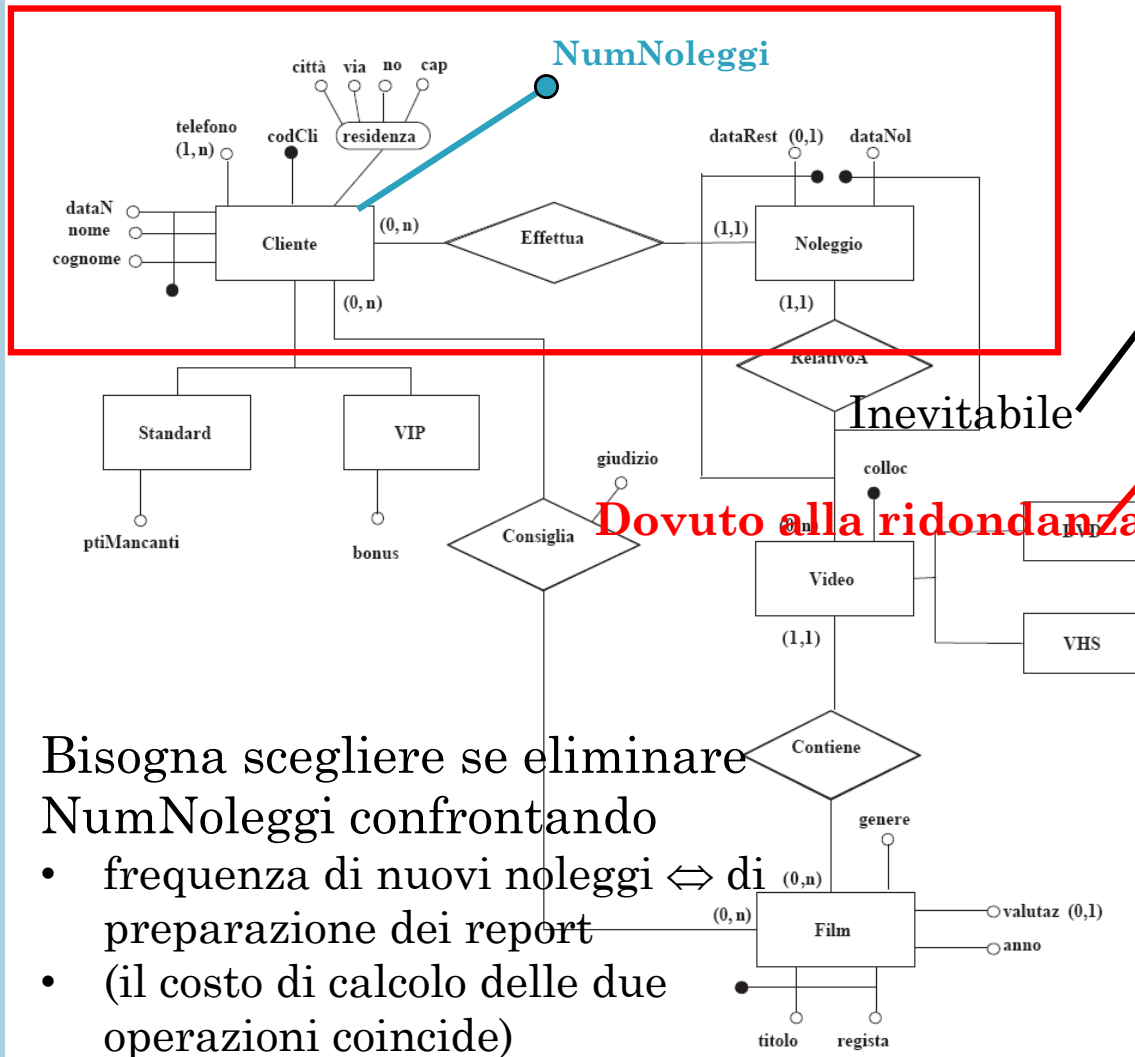
○ Morale: la ridondanza deve essere

- limitata ai casi in cui sia possibile ottenere un significativo beneficio in termini di tempo di esecuzione di interrogazioni eseguite frequentemente
- esplicitata nella documentazione
- gestita automaticamente durante gli aggiornamenti

ANALISI DELLA RIDONDANZA: RULE OF THUMB

- Le valutazioni da effettuare per decidere se eliminare entità o associazioni ridondanti si basano su stime
 - inerentemente soggettive
- Bisogna stimare
 - occupazione memoria del dato ridondante
 - costo per mantenere sincronizzati i dati ridondanti
 - **stima** della frequenza di aggiornamento
 - costo delle operazioni in presenza/assenza di ridondanza
 - **stima** della frequenza di tali operazioni
- Le stime sono basate su informazioni sul volume dei dati indicate nel carico di lavoro

ANALISI DELLA RIDONDANZA - ESEMPIO



Noleggio di un nuovo video da parte di un cliente

- aggiornamento istanze entità Noleggio
- aggiornamento del valore dell'attributo `numNoleggi` per il cliente considerato

Operazione frequente di stampa di un report contenente per ogni cliente il numero totale di noleggi

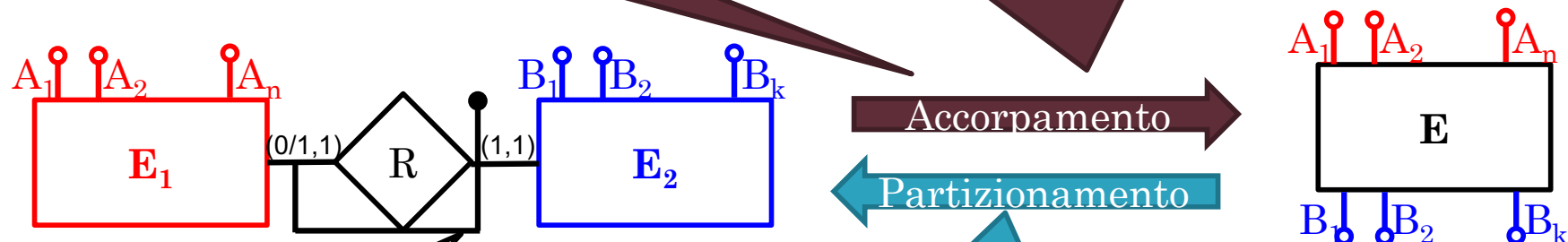
- `numNoleggi`
 - velocizza la preparazione dei report
 - richiede spazio limitato

PARTIZIONAMENTO/ACCORPAMENTO DI ENTITÀ

Sono una l'inverso dell'altra

Può generare attributi opzionali in caso di partecipazione opzionale all'associazione di E_1

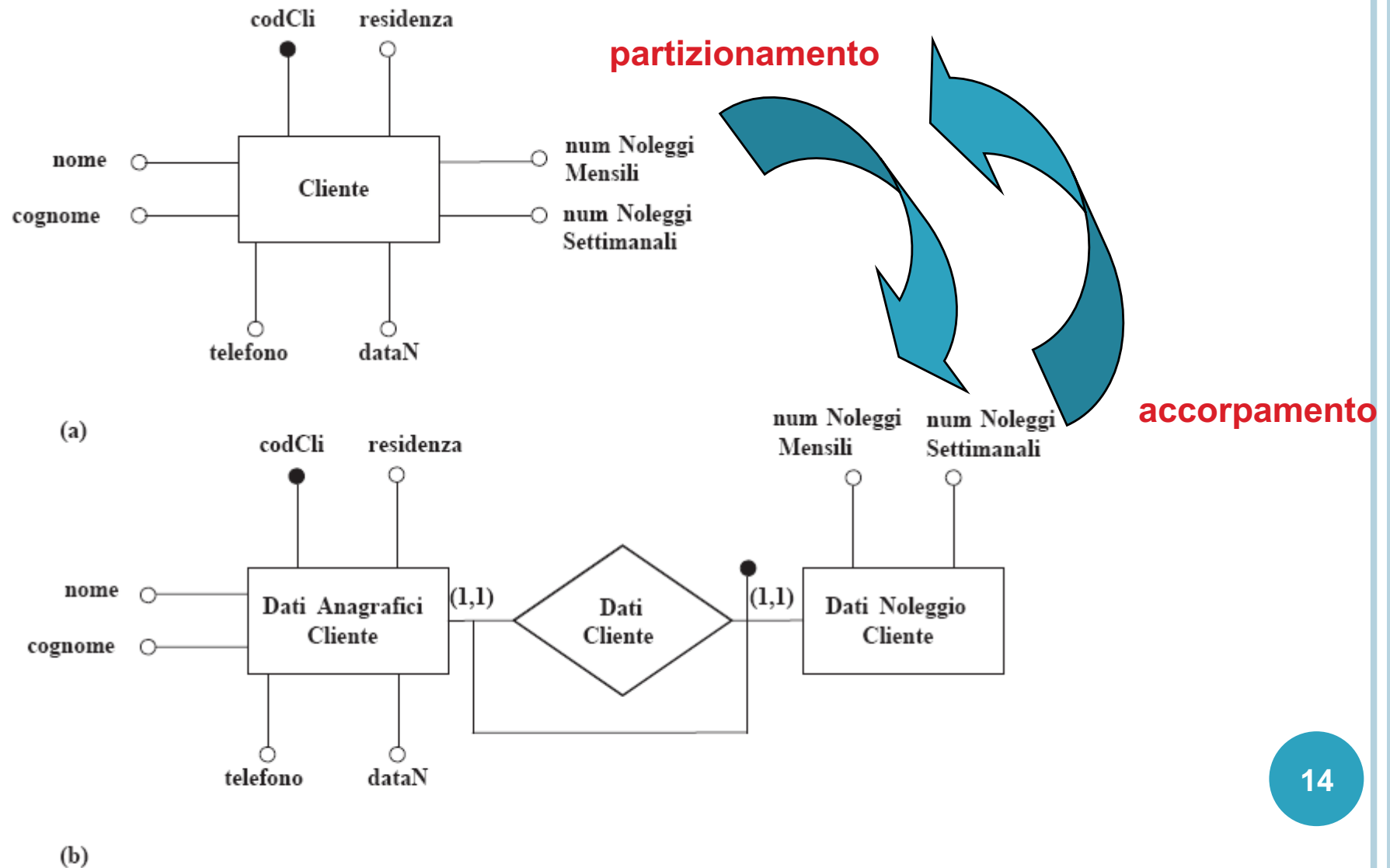
conveniente se ci sono operazioni frequenti che coinvolgono gli attributi sia di E_1 che di E_2
 \Rightarrow evita la navigazione dell'associazione



E_2 identificata esternamente da E_1

conveniente se ci sono operazioni frequenti che coinvolgono solo un sottoinsieme degli attributi di E

PARTIZIONAMENTO/ACCORPAMENTO DI ENTITA' - ESEMPIO



PARTIZIONAMENTO/ACCORPAMENTO DI ENTITÀ

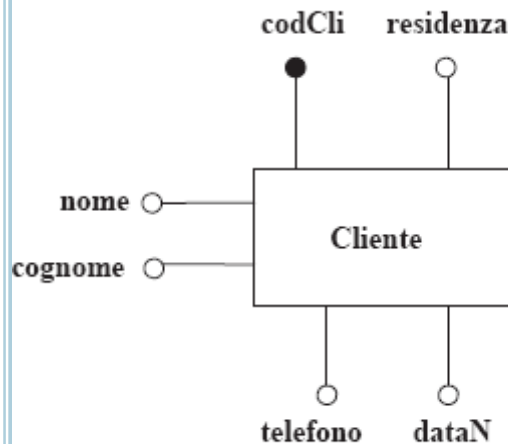
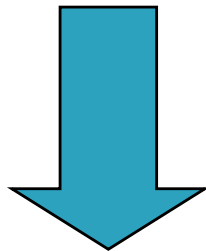
- Si possono eseguire già nella fase di ristrutturazione
- Spesso però si rimandano alla fase di progettazione fisica
 - disponibilità di ulteriori informazioni relative all'esecuzione delle interrogazioni

ELIMINAZIONE DEGLI ATTRIBUTI COMPOSTI

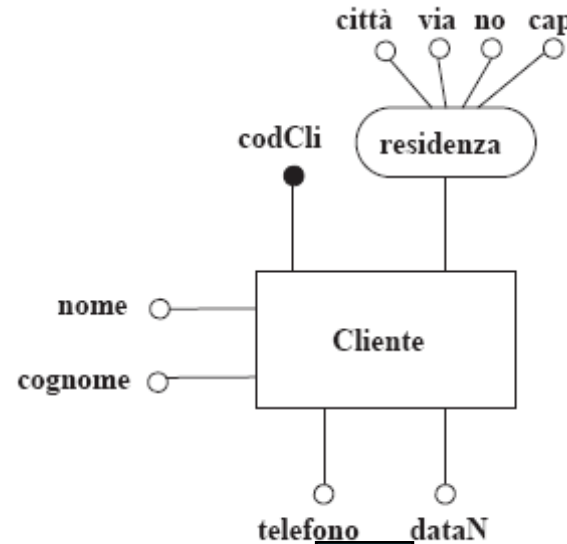
- Eliminazione di un attributo composto A da un'entità E
 - **Soluzione 1** *merge* dei sotto-attributi di A in un unico attributo semplice
 - diventa responsabilità delle applicazioni
 - garantire che il nuovo attributo contenga valori coerenti con la semantica dell'attributo composto ristrutturato
 - fare merge e unmerge
 - **Soluzione 2** considerare tutti i sotto-attributi di A come attributi di E
 - ridefinizione del dominio dell'attributo
 - si perde la relazione tra i sotto-attributi
 - **Soluzione 3** introdurre un'entità per rappresentare il tipo di A e associarla ad E
- Eventuali vincoli di cardinalità esistenti per l'attributo composto vengono associati a ciascuno dei nuovi attributi/all'associazione generati tramite la ristrutturazione
- Se le componenti dell'attributo composto sono a loro volta attributi composti, si riapplica la procedura

ELIMINAZIONE DEGLI ATTRIBUTI COMPOSTI - ESEMPIO

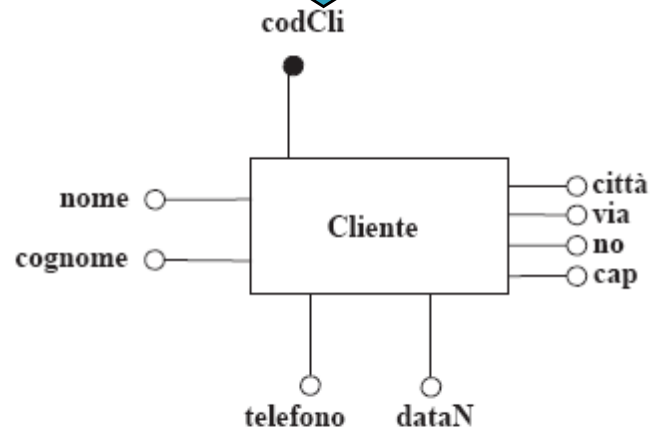
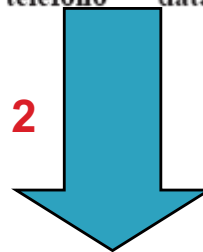
Soluzione 1



Dom(residenza) = string

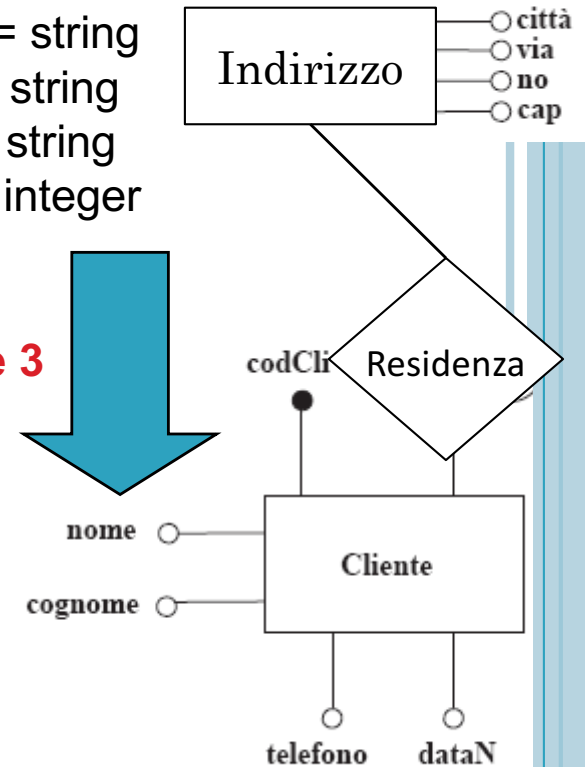
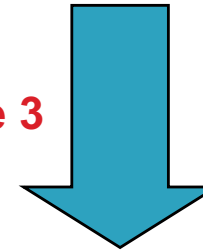


Soluzione 2



Dom(città) = string
Dom(via) = string
Dom(no) = string
Dom(cap) = integer

Soluzione 3

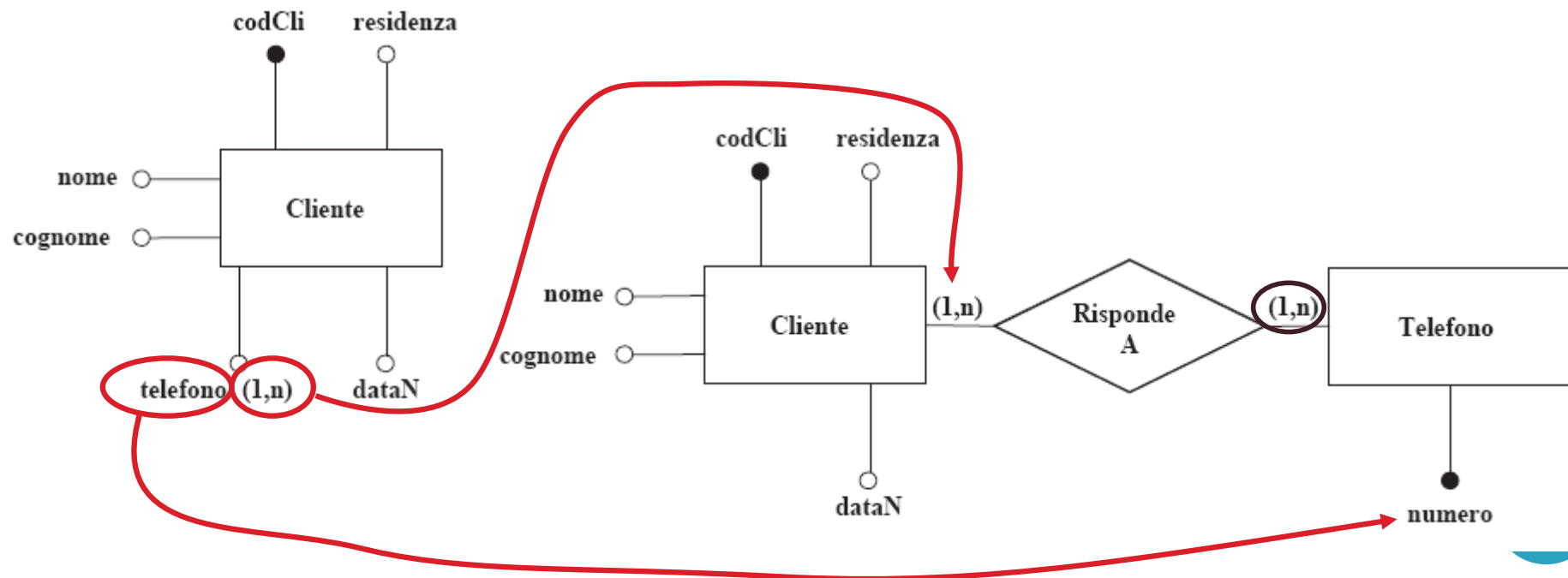


Dom(città) = string
Dom(via) = string
Dom(no) = string
Dom(cap) = integer



ELIMINAZIONE DEGLI ATTRIBUTI MULTI-VALORE

- Data un'entità E con un attributo multivalore A
 - si definisce una nuova entità E_A con un attributo monovalore A
 - si collegano E ed E_A tramite un'associazione R_A
- Vincoli di cardinalità rispetto alla nuova associazione:
 - per E stesso vincolo di cardinalità dell'attributo multi-valore
 - per E_A può essere in generale posto uguale a $(1,n)$



ELIMINAZIONE DELLE GERARCHIE DI GENERALIZZAZIONE

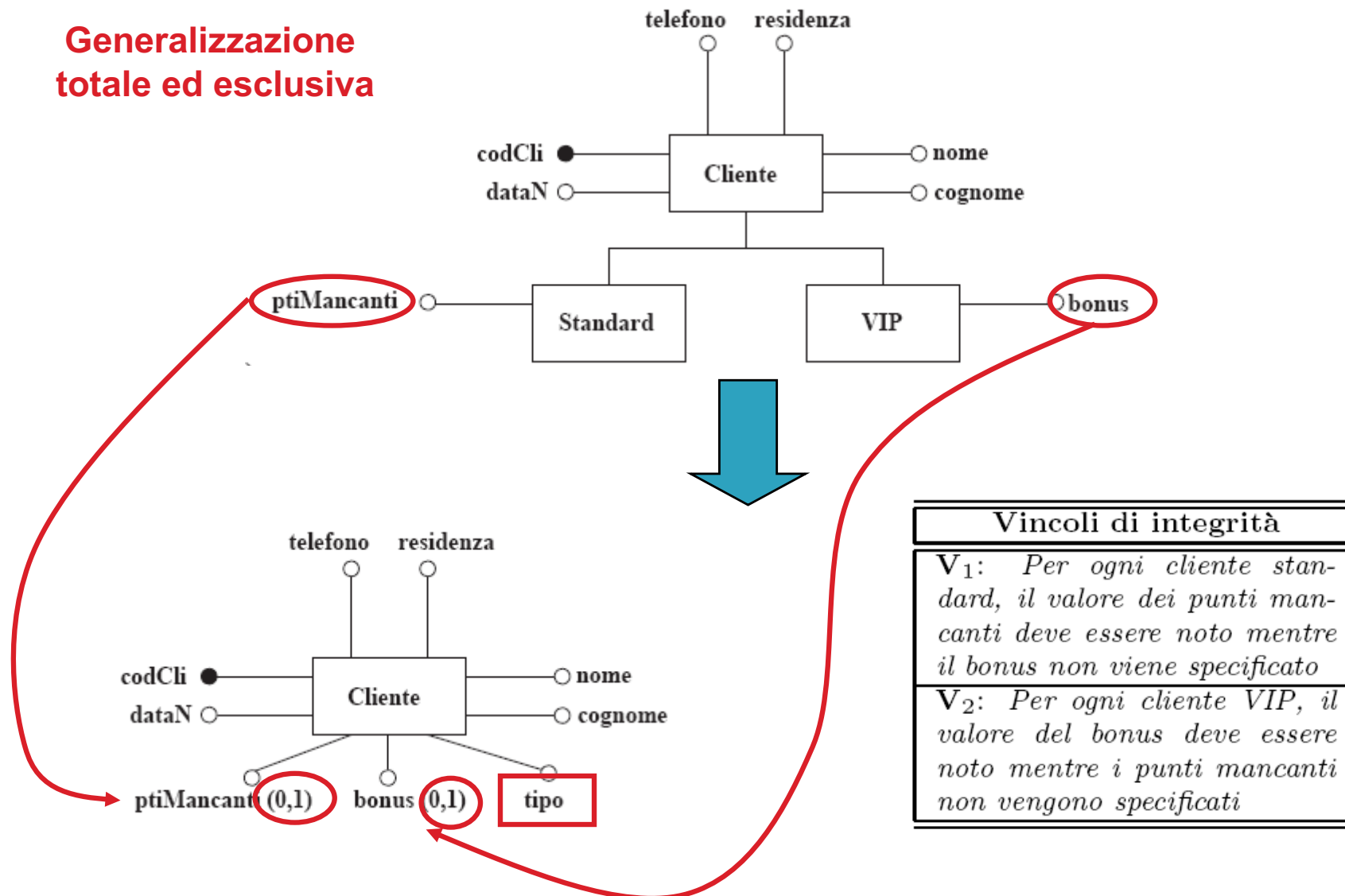
- Entità E generalizzazione di entità E_1, \dots, E_n
- Approccio
 - si estraggono dalla documentazione informazioni sul tipo di gerarchia (totale o parziale, esclusiva o condivisa)
 - si sceglie una soluzione di ristrutturazione, sulla base di
 - tipo di gerarchia
 - carico di lavoro
- Possibili soluzioni
 - eliminazione entità figlie
 - eliminazione entità padre
 - sostituzione della generalizzazione con associazioni

ELIMINAZIONE ENTITÀ FIGLIE

- Entità
 - E_1, \dots, E_n vengono eliminate
- Attributi
 - gli attributi di E_1, \dots, E_n loro attributi vengono inseriti in E come opzionali
 - si aggiunge ad E un attributo *tipo* che specifica da quale entità figlia E_i proviene l'istanza dell'entità padre nello schema ristrutturato
 - nel caso di generalizzazioni totali, non può mai essere nullo
 - nel caso di generalizzazioni parziali, un valore nullo indica un'istanza di E che, nello schema originario, non era istanza di nessuna E_i
 - nel caso di generalizzazioni condivise, sarà multi-valore
 - bisogna aggiungere un vincolo di integrità per garantire che se $\text{tipo} = E_i$, gli attributi obbligatori di E_i siano non nulli
 - bisogna aggiungere un vincolo di integrità per garantire che se un attributo di E_i è non nullo, allora $\text{tipo} = E_i$ (se la generalizzazione è condivisa, E_i è uno dei valori assunti da *tipo*)

ELIMINAZIONE ENTITA' FIGLIE

**Generalizzazione
totale ed esclusiva**



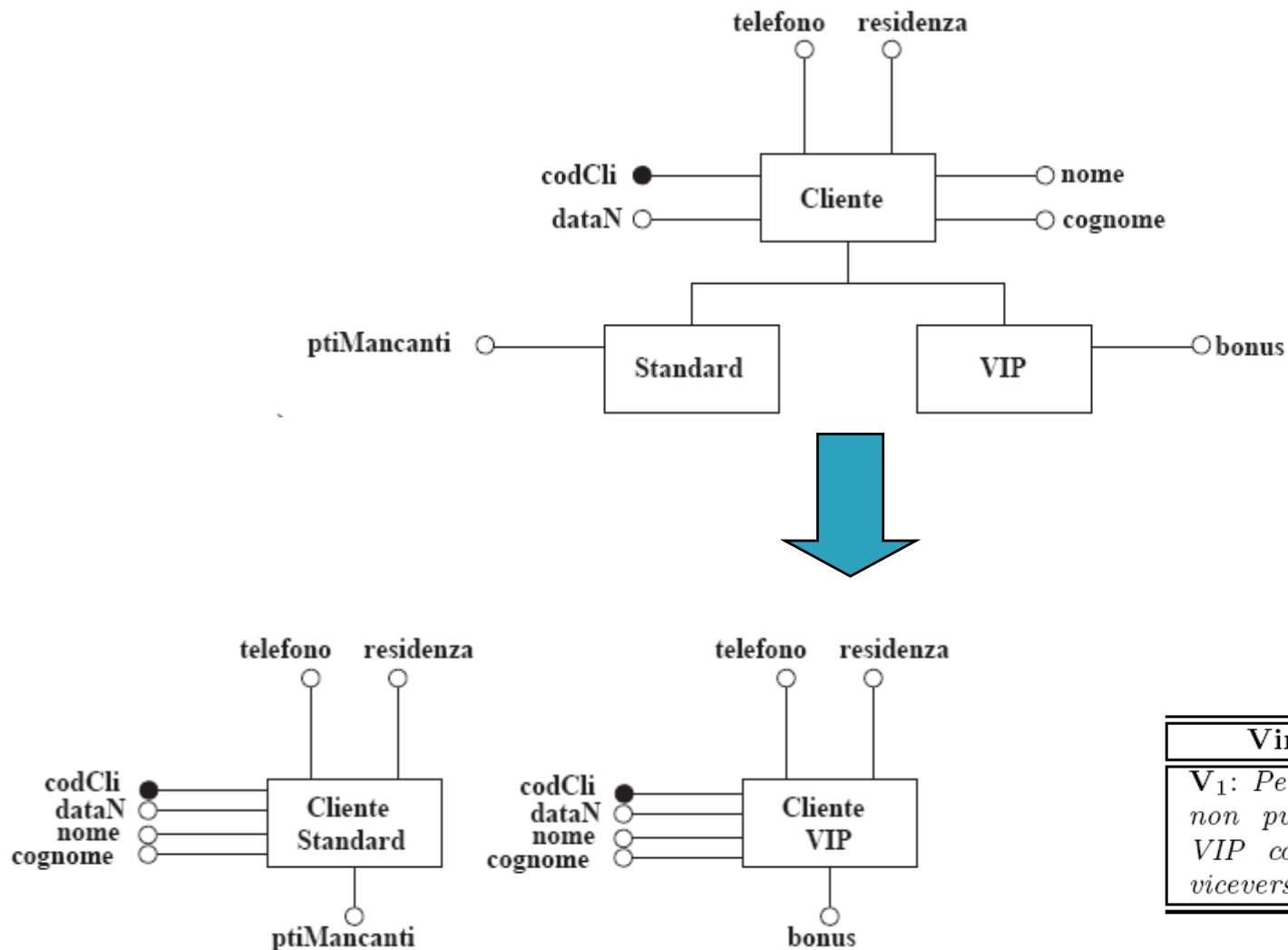
ELIMINAZIONE ENTITÀ FIGLIE - ASSOCIAZIONI

- La partecipazione (obbligatoria od opzionale) di un'entità figlia ad un'associazione diventa la partecipazione opzionale dell'entità padre alla stessa associazione
- Per ogni associazione, bisogna aggiungere un vincolo di integrità che indichi quali *tipi* di istanze dell'entità padre possono essere coinvolti nell'associazione

ELIMINAZIONE ENTITÀ PADRE

- Applicabile solo nel caso di generalizzazione **totale**
- Entità
 - eliminazione dell'entità padre E
- Attributi
 - inserimento degli attributi di E in ciascuna delle entità figlie
- Associazioni
 - ogni associazione a cui partecipava E viene sostituita con n nuove associazioni, una per ogni E_i
- Vincoli di integrità
 - se la generalizzazione esclusiva, vincolo per indicare che, nello schema ristrutturato, non possono esistere istanze di due entità figlie distinte aventi lo stesso valore per gli identificatori (ereditati dall'entità padre)
 - il vincolo di cardinalità di ciascuna entità figlia rispetto alla nuova associazione coinciderà con il vincolo di cardinalità dell'entità padre rispetto all'associazione eliminata
 - i vincoli di cardinalità delle altre entità diventeranno invece opzionali

ELIMINAZIONE ENTITA' PADRE - ESEMPIO



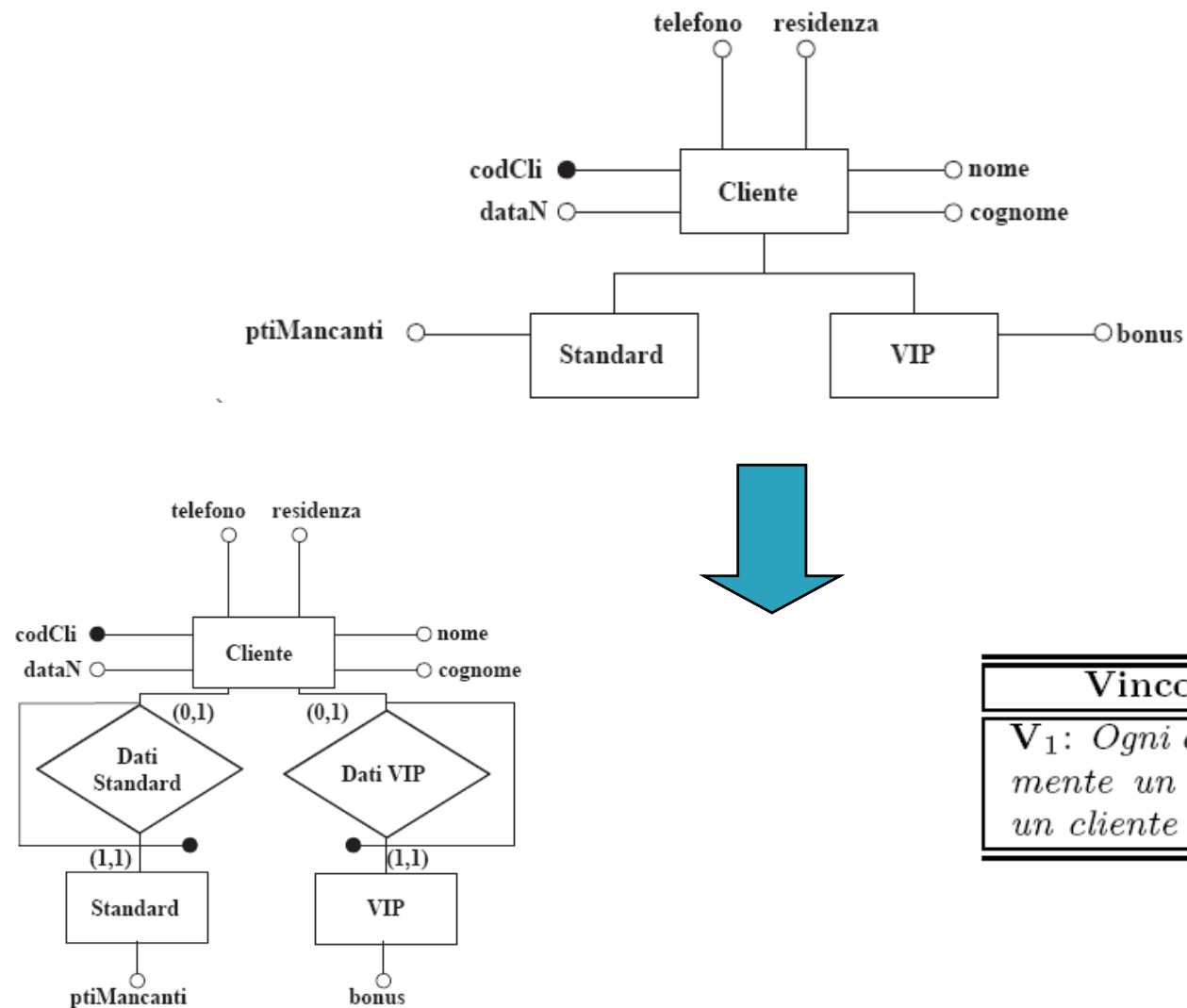
Vincoli di integrità

V_1 : Per ogni cliente standard, non può esistere un cliente VIP con lo stesso codice e viceversa

SOSTITUZIONE DELLA GENERALIZZAZIONE CON ASSOCIAZIONI

- Entità
 - non modificate
- Associazioni
 - la gerarchia viene sostituita da n associazioni R_i uno a uno, che collegano E con E_i
 - ciascun E_i è identificata esternamente da E e partecipa obbligatoriamente a R_i
 - la partecipazione di E a ciascun R_i è opzionale
- Vincoli di integrità
 - se la generalizzazione è esclusiva, un'istanza di E non può partecipare contemporaneamente a due o più associazioni R_i
 - se la generalizzazione è totale, ogni istanza di E deve partecipare obbligatoriamente ad almeno un'associazione R_i

SOSTITUZIONE DELLA GENERALIZZAZIONE CON ASSOCIAZIONI - ESEMPIO



Vincoli di integrità

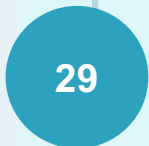
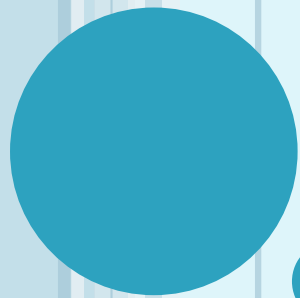
V₁: Ogni cliente è alternativamente un cliente standard od un cliente VIP

OSSERVAZIONI

- Eliminazione entità figlie
 - spreco di memoria per la presenza dei valori nulli
 - conveniente solo nel caso in cui le operazioni non fanno distinzione tra le varie sotto-entità
- Eliminazione entità padre
 - solo per generalizzazione totale
 - conveniente soprattutto nel caso in cui esistano operazioni che si riferiscono alle istanze di una specifica entità figlia
- Sostituzione con associazioni
 - preferibile alla soluzione di eliminare le entità figlie per quanto riguarda la quantità di memoria utilizzata
 - conveniente quando esistono delle operazioni che discriminano tra entità padre ed entità figlie

OSSERVAZIONI

- In alcune situazioni, può essere conveniente adottare soluzioni ibride
 - eliminazione di un sottoinsieme delle entità figlie, mantenendo le altre nello schema
- Generalizzazione a più livelli
 - si applicano le strategie proposte partendo dalle foglie della gerarchia complessiva
 - lo schema risultante dipenderà dal tipo della ristrutturazione applicata ad ogni livello

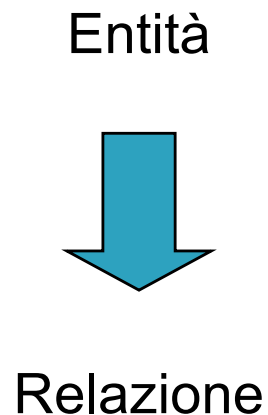


FASE DI TRADUZIONE

Progettazione logica

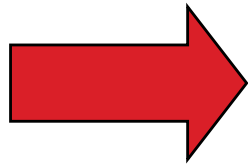
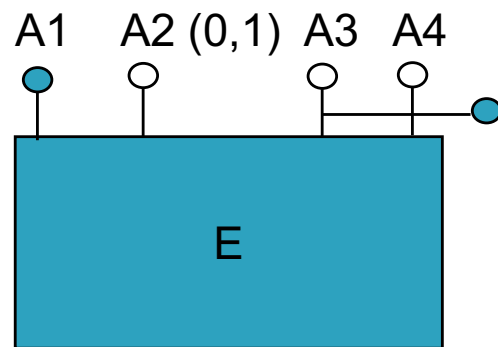
FASE DI TRADUZIONE

- Traduce lo schema ER ristrutturato in uno schema relazionale **equivalente**
- Si basa su un insieme di regole di traduzione
 - traduzione delle entità
 - traduzione delle associazioni
 - traduzione dei vincoli di integrità
 - ottimizzazioni finali



TRADUZIONE ENTITÀ

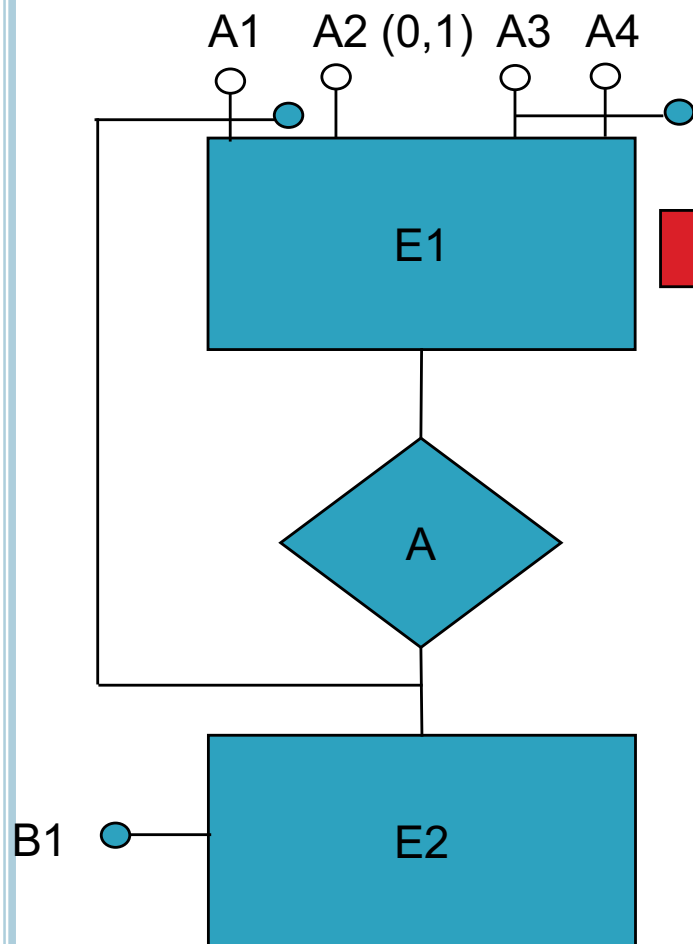
NO IDENTIFICATORI ESTERNI/MISTI



$E(A1/D1, A2_o/D2, A3/D3, A4/D4)$

- D_i estratti dalla documentazione generata dalla progettazione concettuale
- Vincolo di obbligatorietà per $A1, A3, A4$
- Chiavi candidate: $A1, (A3, A4)$

TRADUZIONE ENTITÀ IDENTIFICATORI ESTERNI/MISTI

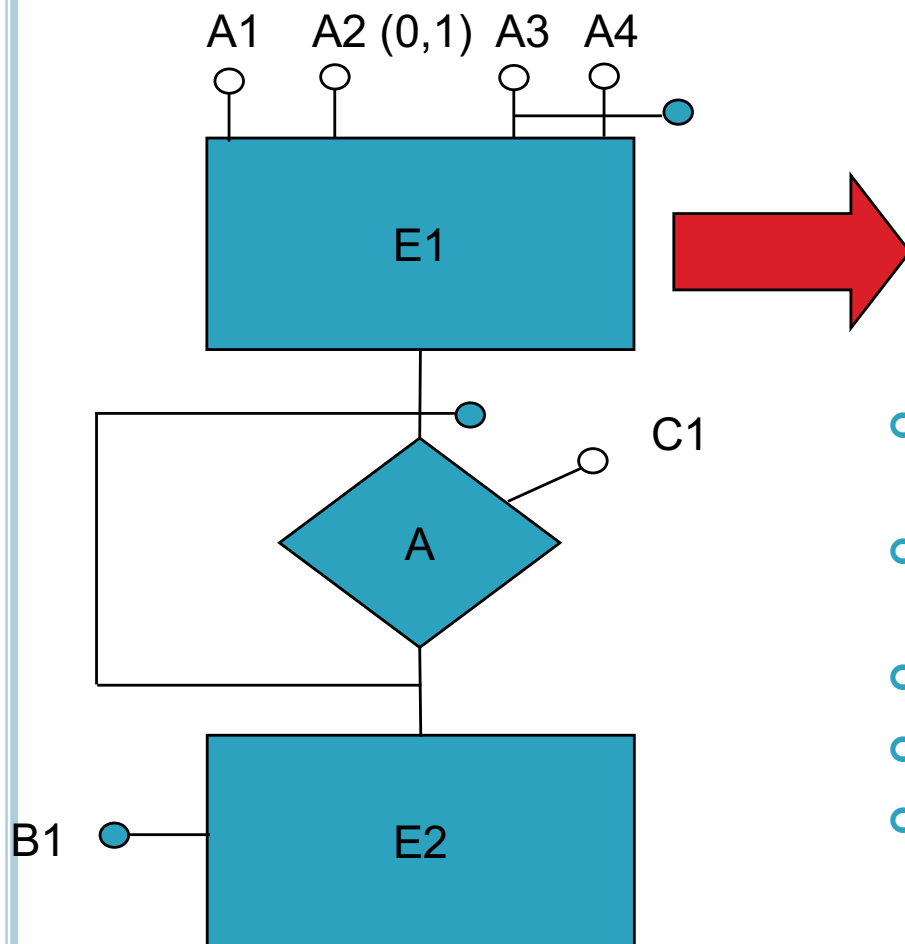


$E1(A1/D1, A2_0/D2, A3/D3, A4/D4, B1^{E2}/F1)$

$E2(B1/F1)$

- D_i , F_i estratti dalla documentazione generata dalla progettazione concettuale
- Vincolo di obbligatorietà per A1, A3, A4, B1
- Chiavi candidate per E2: B1
- Chiavi candidate per E1: (A1,B1), (A3,A4)
- A viene automaticamente rappresentata

TRADUZIONE ENTITÀ IDENTIFICATORI ESTERNI/MISTI



$E1(A1/D1, A2_o/D2, A3/D3, A4/D4, B1^{E2}/F1, C1)$

$E2(B1/F1)$

- D_i, F_i estratti dalla documentazione generata dalla progettazione concettuale
- Vincolo di obbligatorietà per A1, A3, A4, B1, C1
- Chiavi candidate per E2: B1
- Chiavi candidate per E1: B1, (A3,A4)
- A viene automaticamente rappresentata

TRADUZIONE ENTITÀ

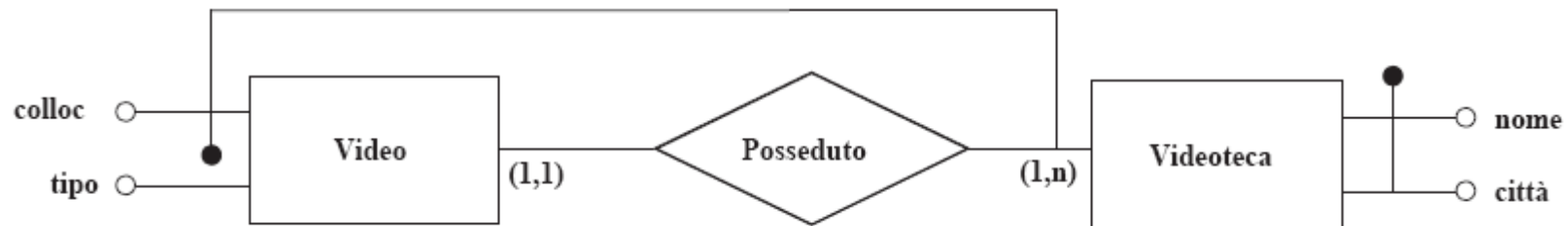
IDENTIFICATORI ESTERNI/MISTI 2

Il processo di eliminazione degli identificatori esterni o misti procede dalle foglie (le entità che hanno solo identificatori interni)

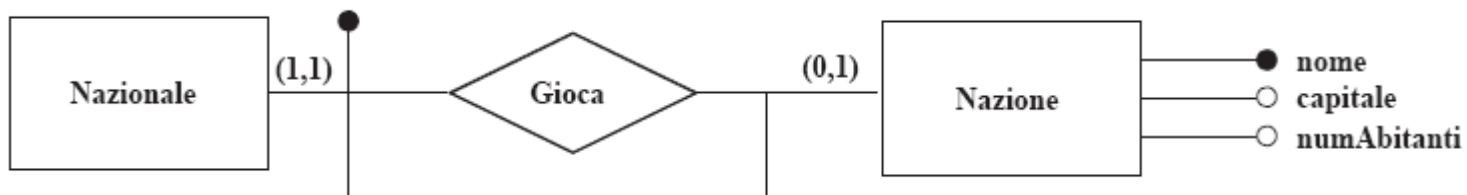
Se E2 è

- parte di un identificatore esterno/misto di E1
 - a sua volta identificato esternamente
- bisogna eliminare
- prima l'identificatore esterno/misto di E2
 - in modo da avere la chiave da usare nel passo successivo
 - poi quello di E1

TRADUZIONE ENTITÀ - ESEMPIO



Videoteca(nome, città)
 Video(colloc, nome^{Videoteca}, città^{Videoteca}, tipo)



Nazione(nome, capitale, numAbitanti)
 Nazionale(nome^{Nazione})

TRADUZIONE ENTITÀ

SCELTA CHIAVE PRIMARIA

- Ciascuna relazione potrebbe essere caratterizzata da più di una chiave
 - è necessario selezionare una di queste come chiave primaria
 - la scelta dipende da criteri di efficienza
- Analoghi criteri per determinare quale chiave di una relazione $R2$ usare come chiave esterna in una relazione $R1$
 - è preferibile definire una chiave esterna sulla base di una chiave primaria

TRADUZIONE ENTITÀ

SCELTA CHIAVE PRIMARIA - CRITERI

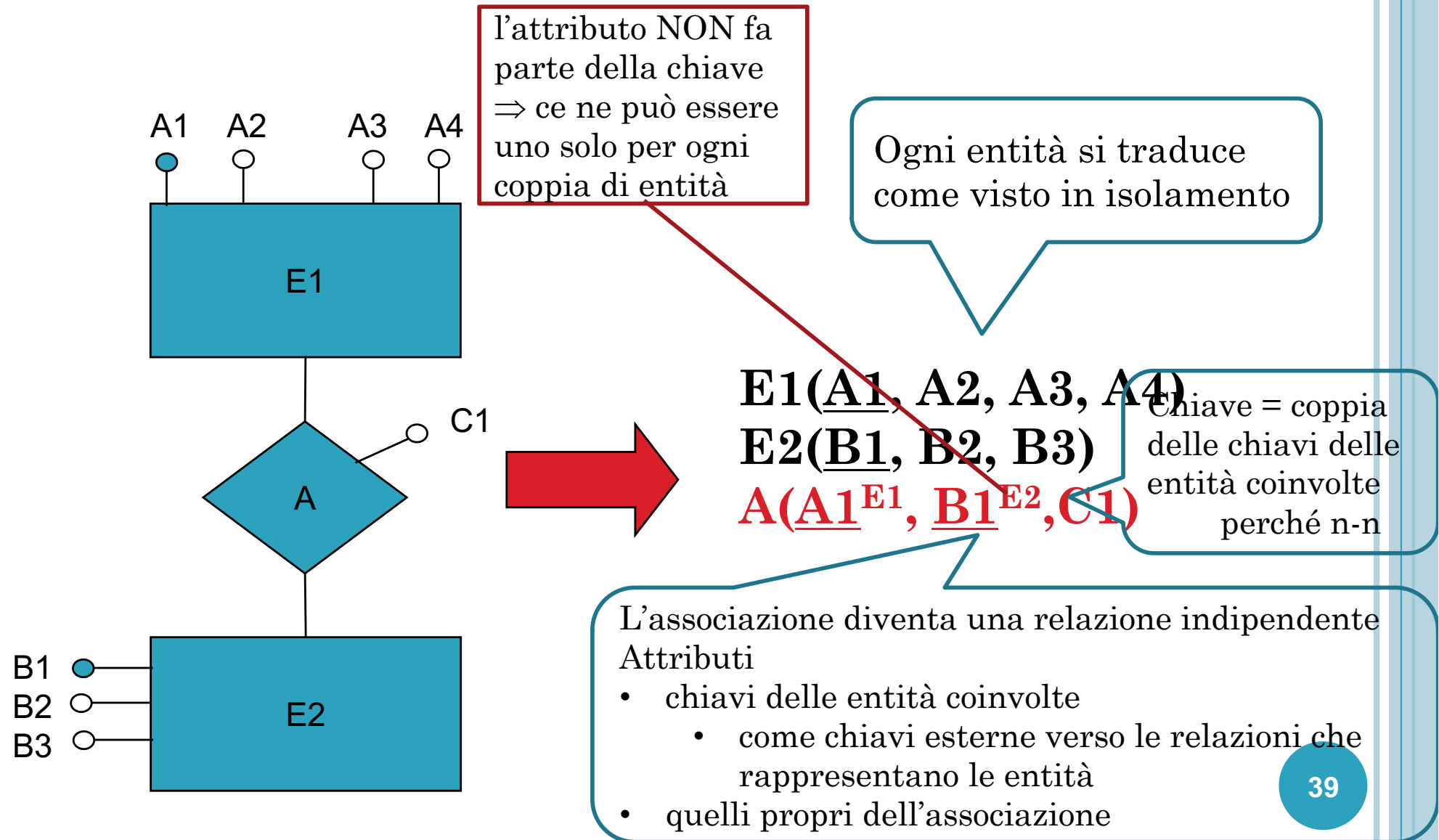
- gli identificatori che contengono attributi opzionali **non** possono essere selezionati come chiave primaria
- sono preferibili identificatori
 - composti da pochi piuttosto che molti attributi
 - identificatori che si stima vengano utilizzati da molte operazioni per accedere alle entità
- Se nessuna chiave candidata soddisfa i requisiti precedenti
 - si aggiunge un ulteriore attributo *codice* come chiave primaria
 - assegnare a *codice* valori speciali generati

TRADUZIONE ASSOCIAZIONI

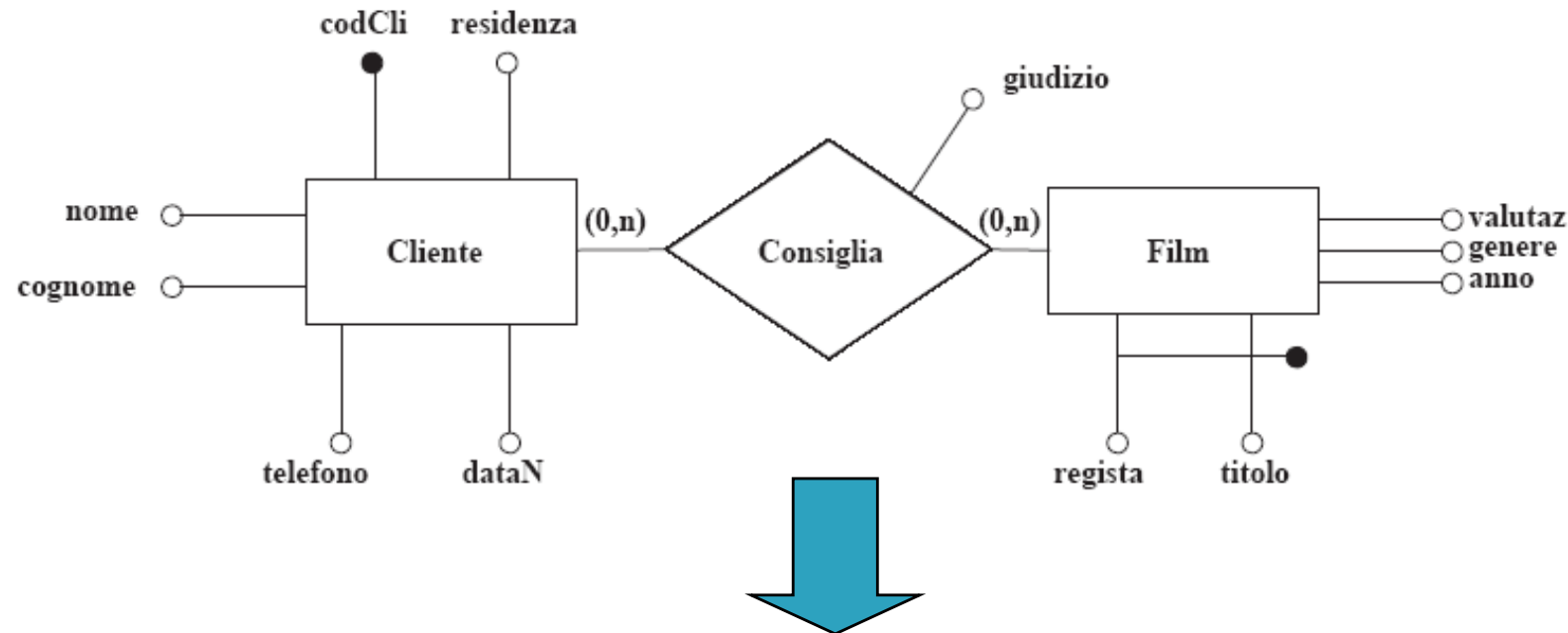
SEMPLIFICAZIONE

- Per semplicità
 - useremo entità con una sola chiave semplice
 - ometteremo la rappresentazione del dominio degli attributi
- Per verificare di aver capito
 - riformulate gli schemi usando entità con più chiavi candidate
 - non dovrebbe cambiare niente
 - riformulate gli schemi usando entità con la chiave primaria composta
 - completate con i domini degli attributi

TRADUZIONE ASSOCIAZIONE BINARIA MOLTI A MOLTI



TRADUZIONE ASSOCIAZIONE BINARIA MOLTI A MOLTI ESEMPIO



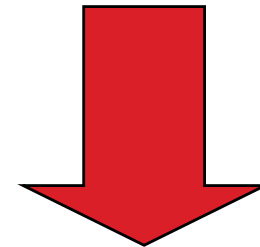
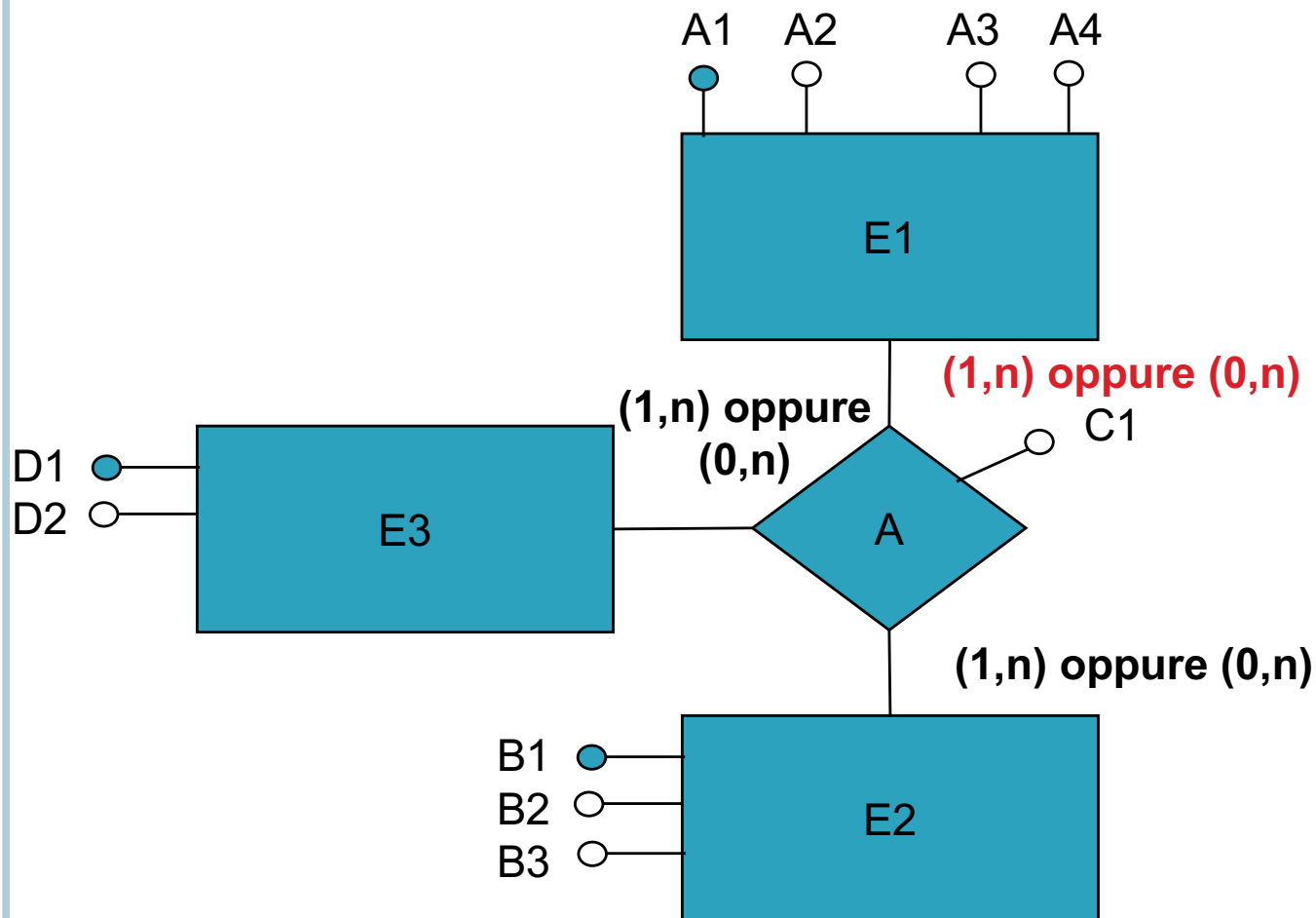
Cliente(codCli, nome, cognome, telefono, dataN, residenza)

Film(titolo, regista, anno, genere, valutaz)

Consiglia(codCli^{Cliente}, titolo^{Film}, regista^{Film}, giudizio)

TRADUZIONE ASSOCIAZIONE N-ARIA MOLTI A MOLTI

- In modo analogo ad associazioni binarie
- Caso molto comune



$E1(\underline{A1}, A2, A3, A4)$

$E2(\underline{B1}, B2, B3)$

$E3(\underline{D1}, D2)$

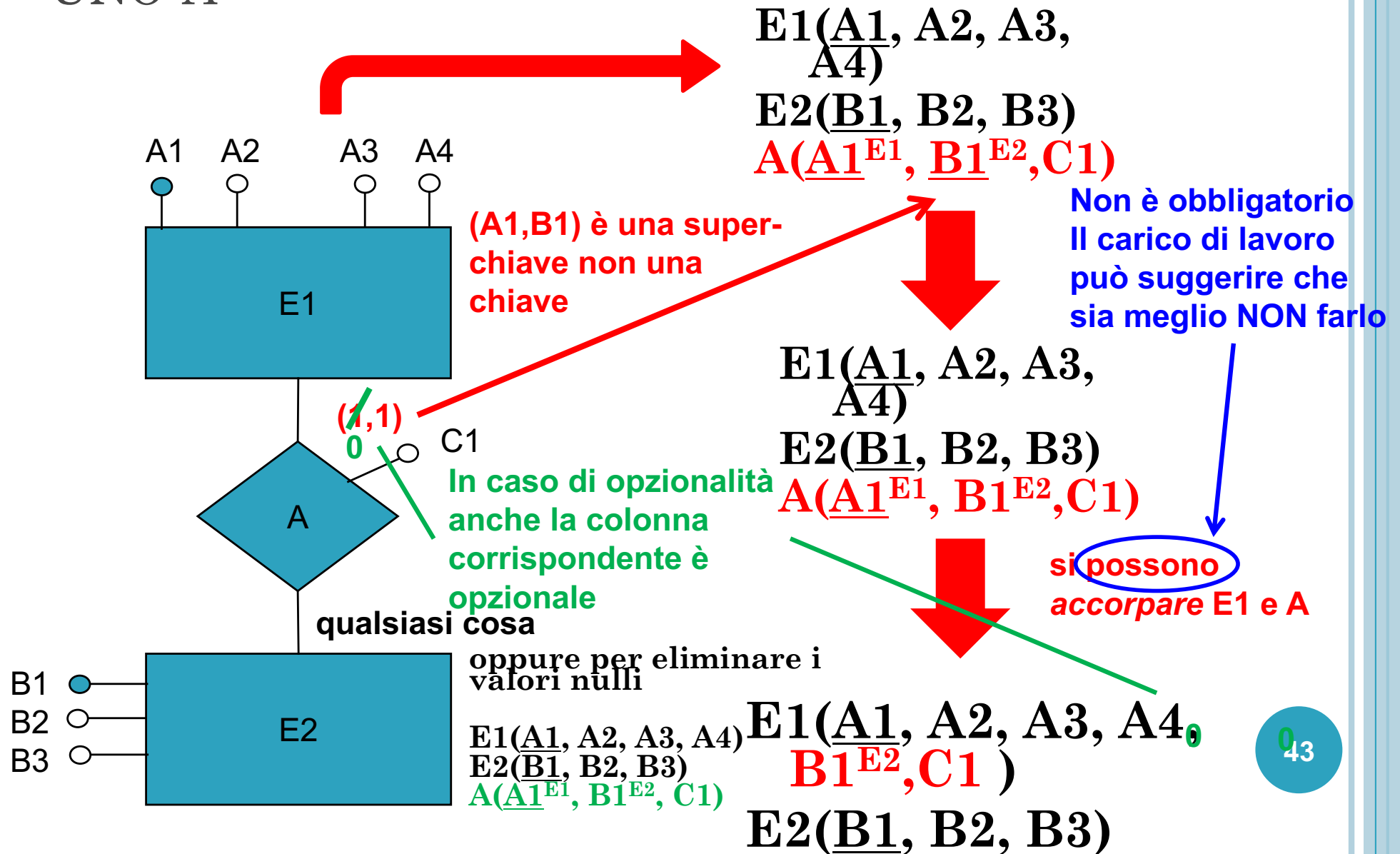
$A(\underline{A1}^{E1}, \underline{B1}^{E2}, \underline{D1}^{E3}, C1)$

TRADUZIONE GENERICA: PROBLEMATICHE

- Ogni navigazione di un'associazione richiede due join
 - computazionalmente pesante
- In alcuni casi (A1, B1, D1) è una super-chiave per A
 - La determinazione della chiave può avvenire dall'analisi di particolari vincoli di integrità
 - es. se la molteplicità massima di partecipazione di una delle entità è 1
- In casi di relazioni 1-1 l'informazione memorizzata è ridondante
 - può lo stesso essere utile mantenere questa traduzione per le stesse ragioni per cui si fa partizionamento
- In casi di relazioni 1-1 non si cattura il vincolo di unicità

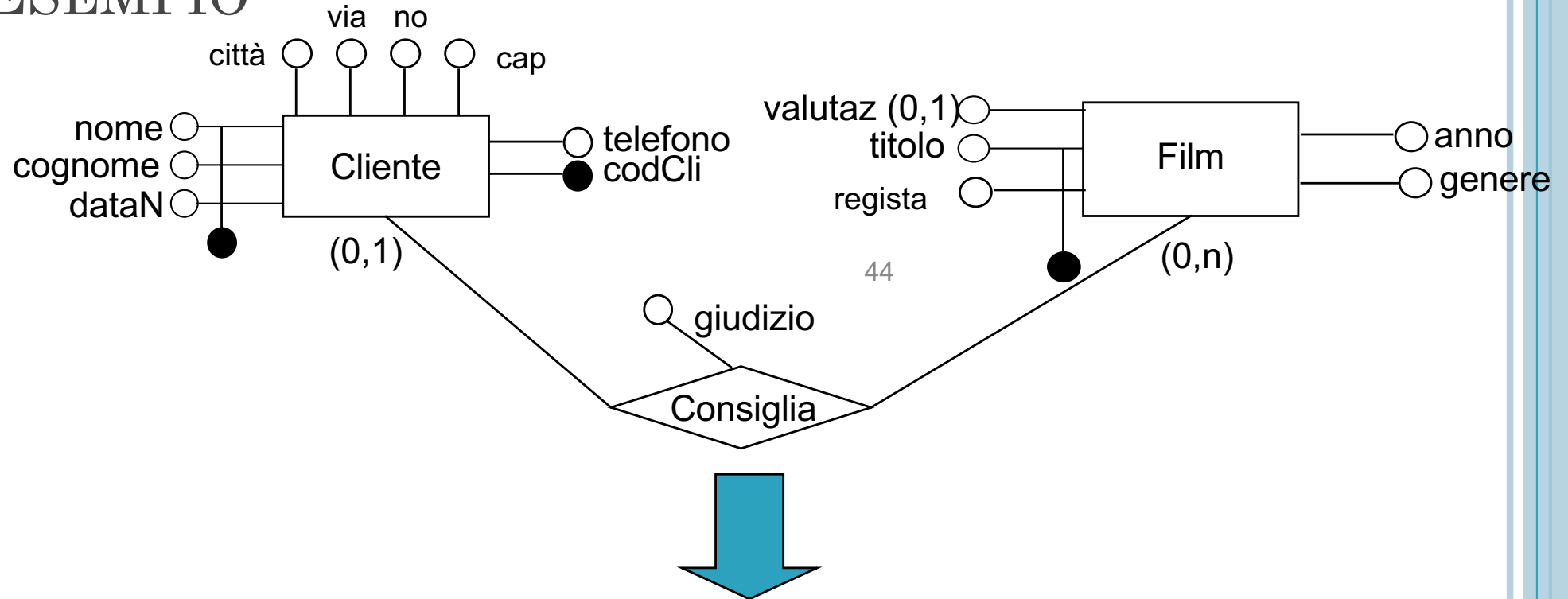
TRADUZIONE ASSOCIAZIONE BINARIA

UNO A *



TRADUZIONE ASSOCIAZIONE BINARIA UNO A *

ESEMPIO

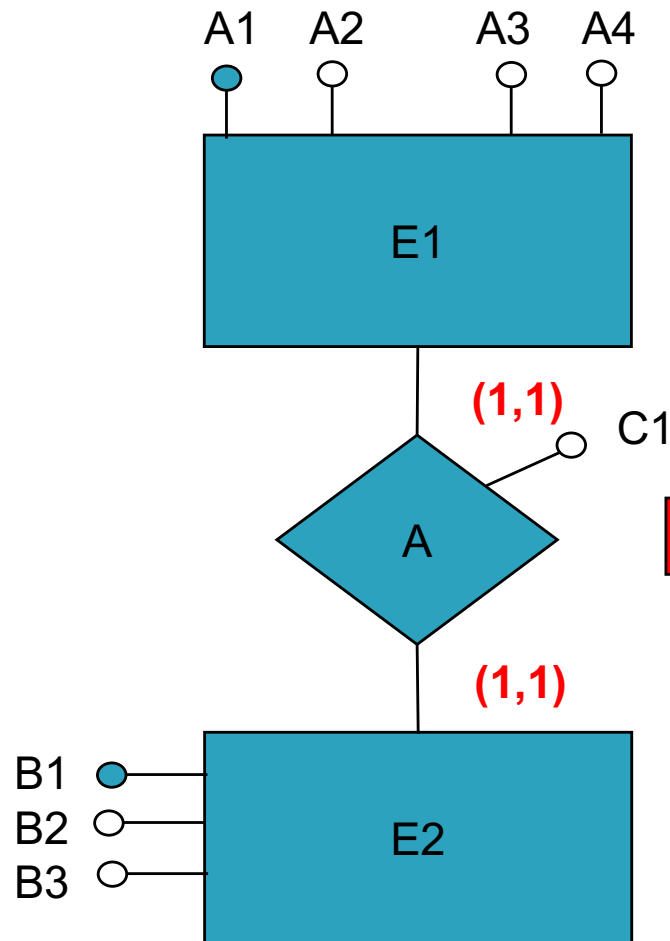


Film(titolo, regista, anno, genere, valutaz)

Cliente(codCli, nome, cognome, telefono, dataN, città, via, no, cap,
 titolo₀^{Film}, regista₀^{Film}, giudizio₀)

TRADUZIONE ASSOCIAZIONE BINARIA UNO A UNO

Se entrambe partecipano univocamente si può scegliere con quale accorpare l'associazione sulla base del carico di lavoro

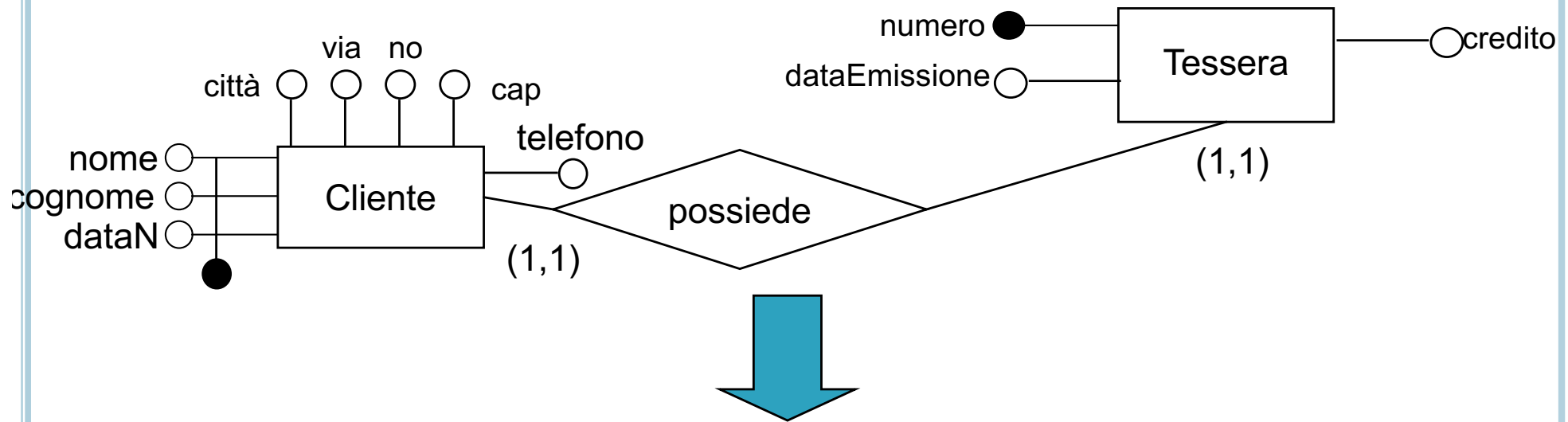


$E1(\underline{A1}, A2, A3, A4, \mathbf{B1^{E2}}, \mathbf{C1})$
 $E2(\underline{B1}, B2, B3)$

Oppure

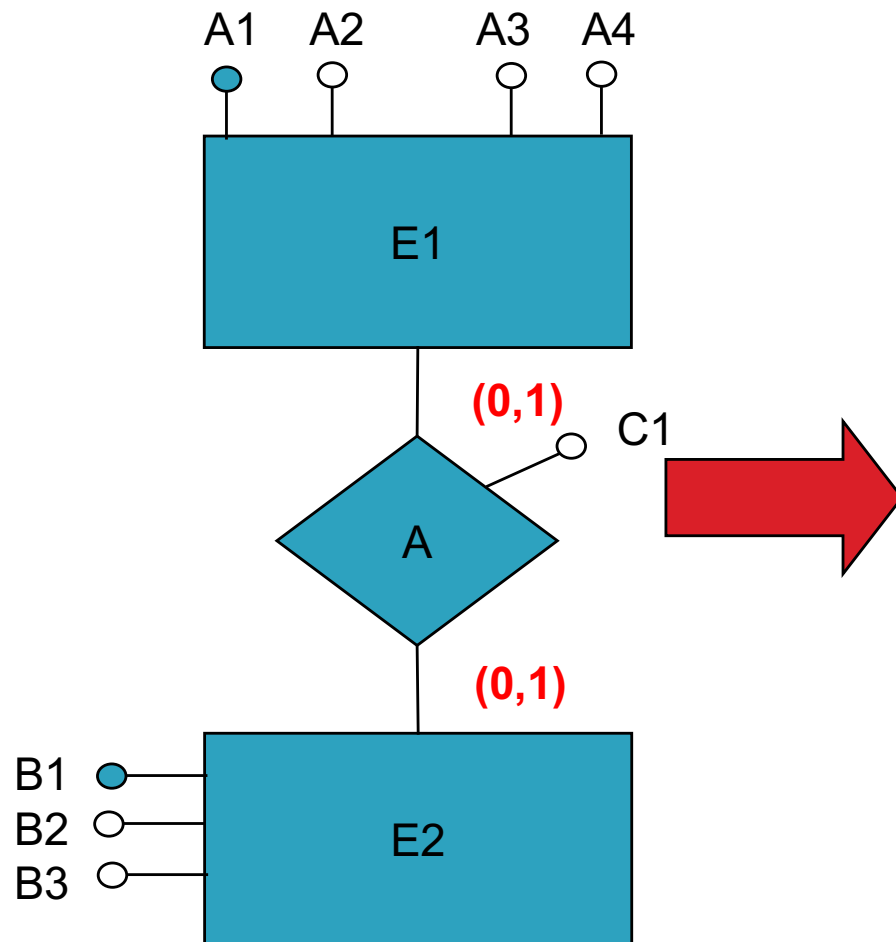
$E1(\underline{A1}, A2, A3, A4)$
 $E2(\underline{B1}, B2, B3, \mathbf{A1^{E1}}, \mathbf{C1})$

TRADUZIONE ASSOCIAZIONE BINARIA UNO A UNO ESEMPIO



Tessera(numero,dataEmissione,credito)
 Cliente(nome,cognome,telefono,dataN,città, via, no,cap,numero^{Tessera})
 Tessera(numero,dataEmissione,credito, nome^{Cliente},cognome^{Cliente},dataN^{Cliente})
 Cliente(nome,cognome,telefono,dataN,città, via, no,cap)
 Tessera(numero,dataEmissione,credito)
 Cliente(nome,cognome,telefono,dataN,città, via, no,cap)
Possiede(nome^{Cliente},cognome^{Cliente},dataN^{Cliente},numero^{Tessera}) oppure
Possiede(nome^{Cliente},cognome^{Cliente},dataN^{Cliente},numero^{Tessera})

TRADUZIONE ASSOCIAZIONE BINARIA UNO A UNO PARTECIPAZIONE OPZIONALE DI ENTRAMBE LE ENTITÀ



$E1(\underline{A1}, A2, A3, A4, \textcolor{red}{B1}^{\textcolor{red}{E2}}, \textcolor{red}{C1}_o)$
 $E2(\underline{B1}, B2, B3)$

oppure

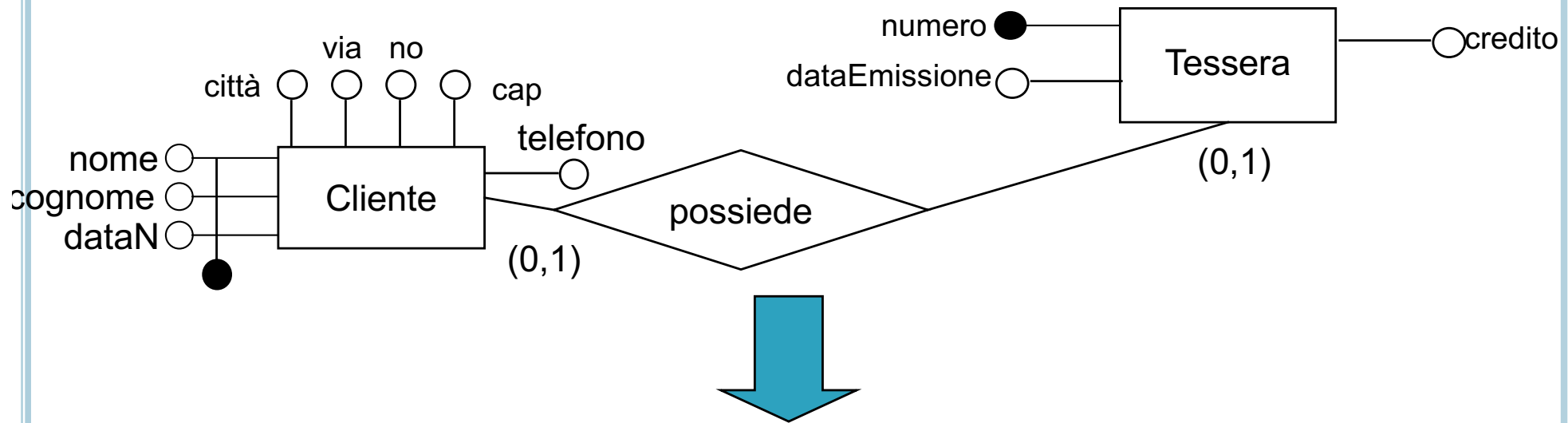
$E1(\underline{A1}, A2, A3, A4)$
 $E2(\underline{B1}, B2, B3, \textcolor{red}{A1}^{\textcolor{red}{E1}}, \textcolor{red}{C1}_o)$

alternativa per eliminare i
valori nulli

$E1(\underline{A1}, A2, A3, A4)$
 $E2(\underline{B1}, B2, B3)$
 $\textcolor{red}{A}(\textcolor{red}{A1}^{\textcolor{red}{E1}}, \textcolor{red}{B1}^{\textcolor{red}{E2}}, \textcolor{red}{C1})$ oppure
 $\textcolor{red}{A}(\textcolor{red}{A1}^{\textcolor{red}{E1}}, \textcolor{red}{B1}^{\textcolor{red}{E2}}, \textcolor{red}{C1})$

= si ritorna alla traduzione
standard

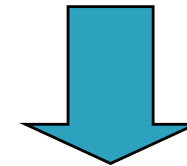
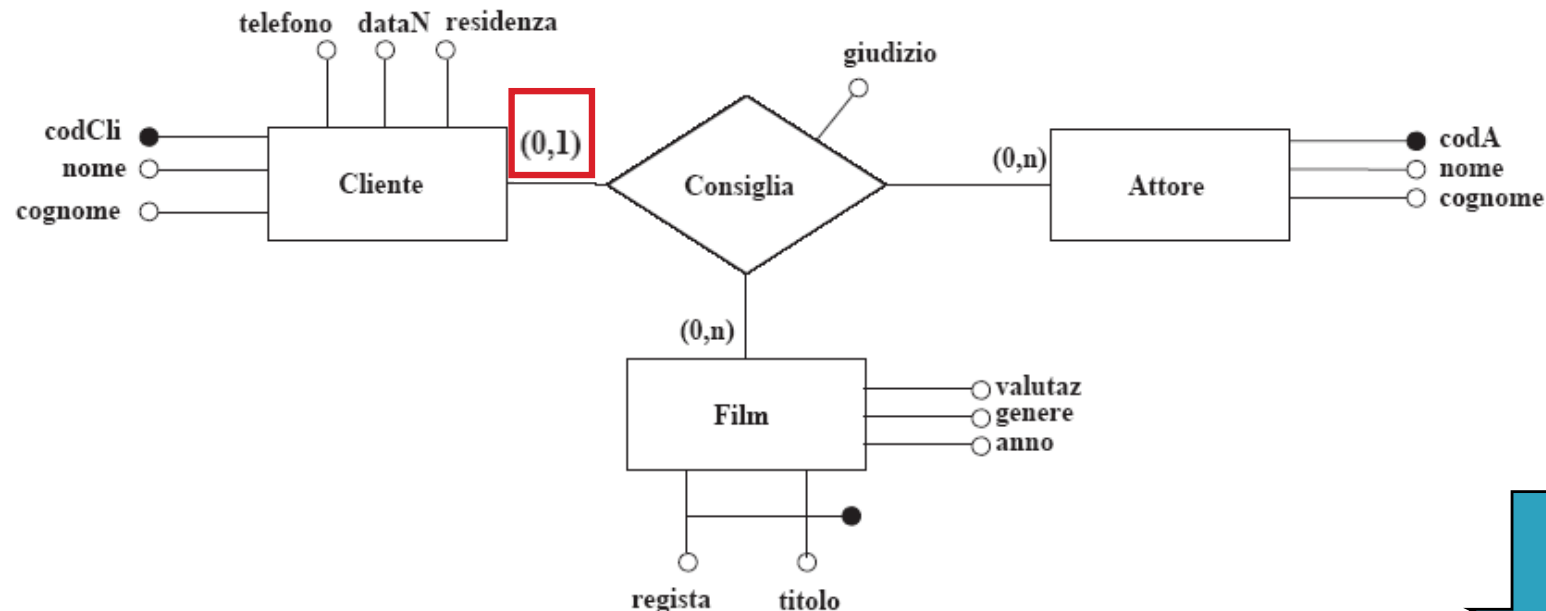
TRADUZIONE ASSOCIAZIONE BINARIA UNO A UNO ESEMPIO



Tessera(numero,dataEmissione,credito)
 Cliente(nome,cognome,telefono,dataN,città, via, no,cap,numero₀^{Tessera})
 Tessera(numero,dataEmissione,credito, nome₀^{Cliente},cognome₀^{Cliente},dataN₀^{Cliente})
 Cliente(nome,cognome,telefono,dataN,città, via, no,cap)
 Tessera(numero,dataEmissione,credito)
 Cliente(nome,cognome,telefono,dataN,città, via, no,cap)
Possiede(nome₀^{Cliente},cognome₀^{Cliente},dataN₀^{Cliente},numero₀^{Tessera}) oppure
Possiede(nome₀^{Cliente},cognome₀^{Cliente},dataN₀^{Cliente},numero₀^{Tessera})

TRADUZIONE ASSOCIAZIONE N-ARIA 1 A MOLTI

ESEMPIO

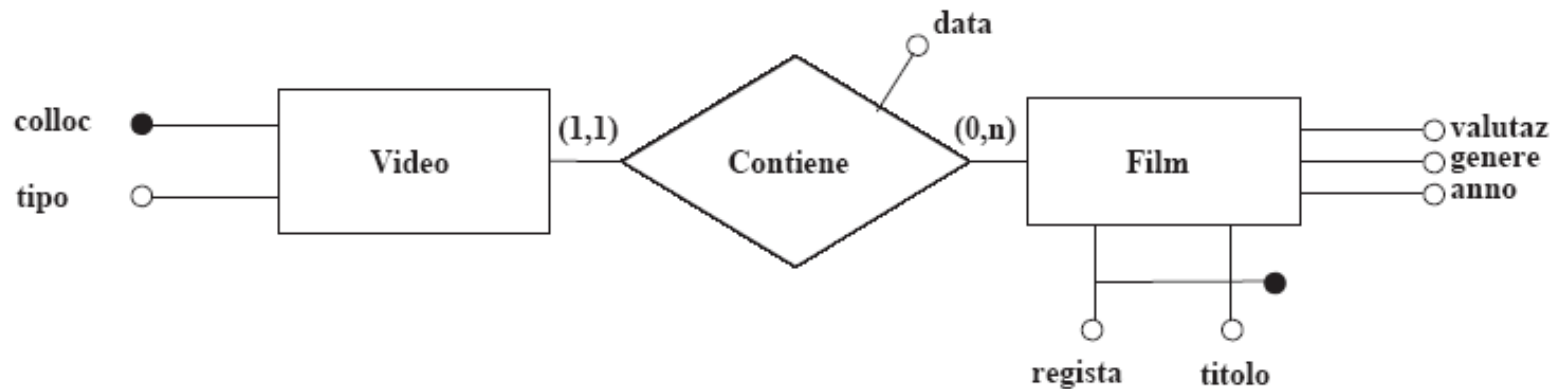


Cliente(codCli, nome, cognome, telefono, dataN, residenza,
 titolo_o^{Film}, regista_o^{Film}, codA_o^{Attore}, giudizio_o)
 Film(titolo, regista, anno, genere, valutaz)
 Attore(codA, nome, cognome)

Come si può ridurre la presenza di valori nulli?

TRADUZIONE ASSOCIAZIONE: PALESTRA

ESEMPIO 1

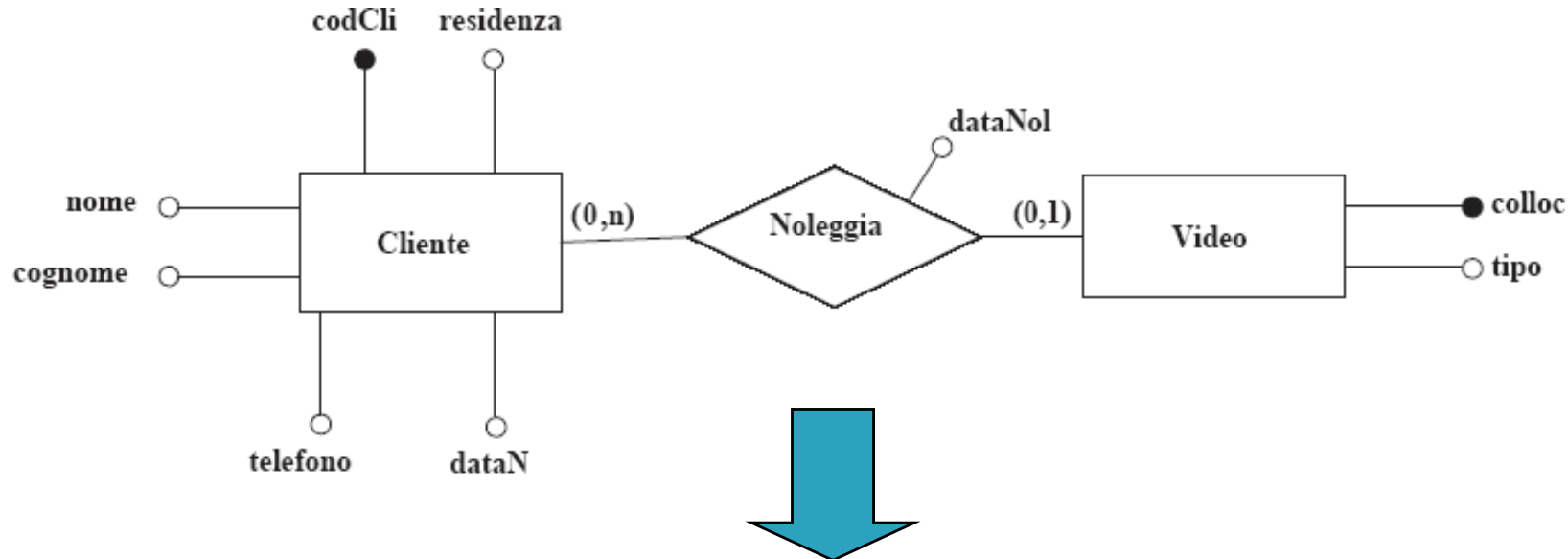


Film(titolo,registra,anno,genere,valutaz)

Video(colloc,tipo,titolo^{Film},registra^{Film},data)

TRADUZIONE ASSOCIAZIONE: PALESTRA

ESEMPIO 2

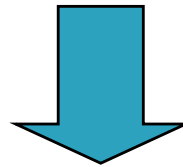
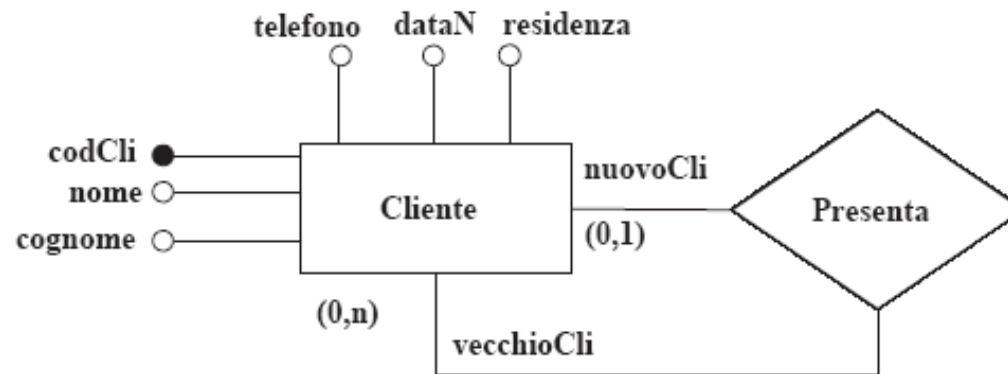


Video(colloc,tipo,codCli_o^{Cliente},dataNol_o)
 Cliente(codCli,nome,cognome,telefono,dataN,residenza)

Video(colloc,tipo)
 Cliente(codCli,nome,cognome,telefono,dataN,residenza)
 Noleggia(colloc^{Video},codCli^{Cliente},dataNol)

TRADUZIONE ASSOCIAZIONE: PALESTRA

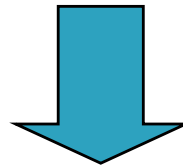
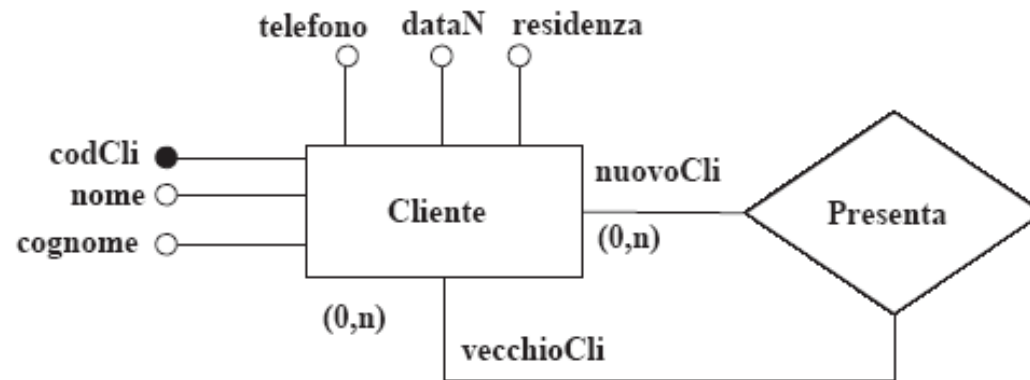
ESEMPIO 3



Cliente(codCli, nome, cognome, telefono, dataN, residenza, **vecchioCli**_o^{Cliente})
 oppure
 Cliente(codCli, nome, cognome, telefono, dataN, residenza)
Presenta(nuovoCli^{Cliente}, **vecchioCli**^{Cliente})

TRADUZIONE ASSOCIAZIONE: PALESTRA

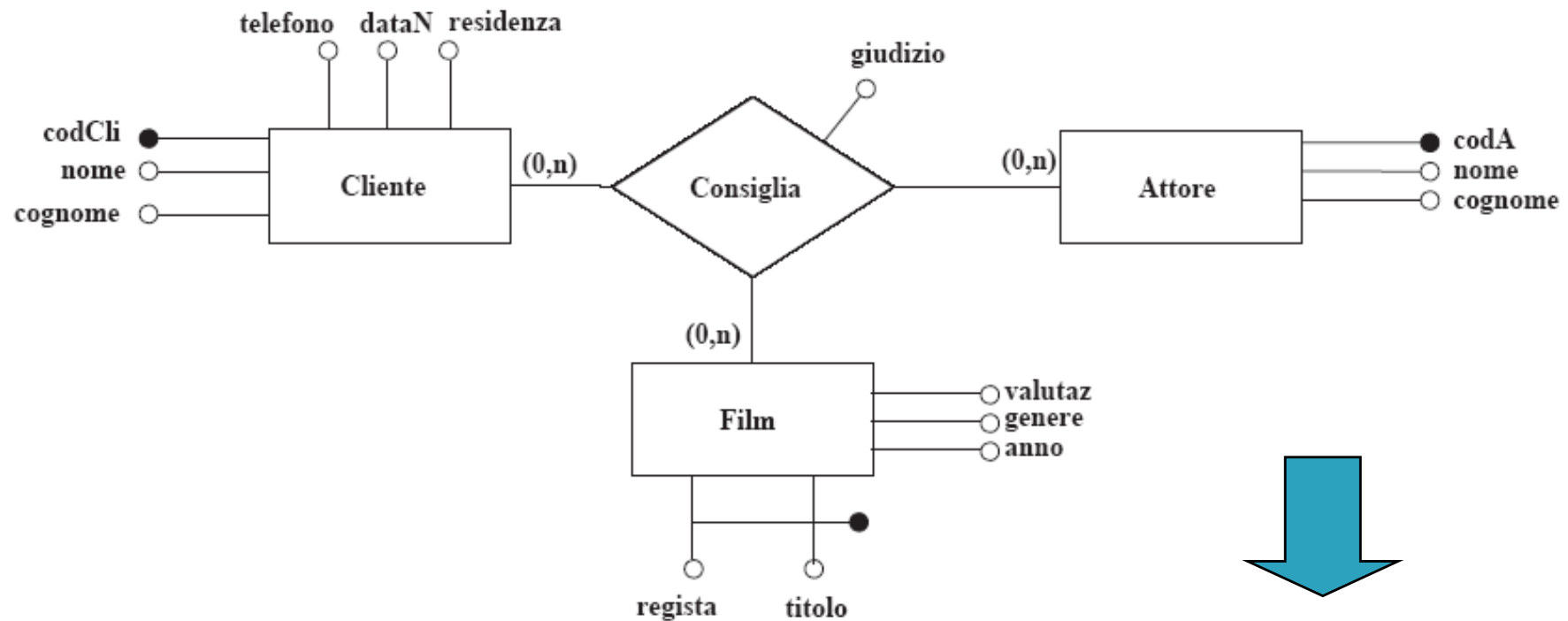
ESEMPIO 4



Cliente(codCli, nome, cognome, telefono, dataN, residenza)
Presenta(nuovoCli^{Cliente}, vecchioCli^{Cliente})

TRADUZIONE ASSOCIAZIONE: PALESTRA

ESEMPIO 5



Cliente(codCli,nome,cognome,telefono,dataN,residenza)

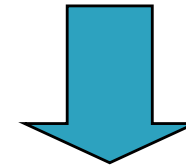
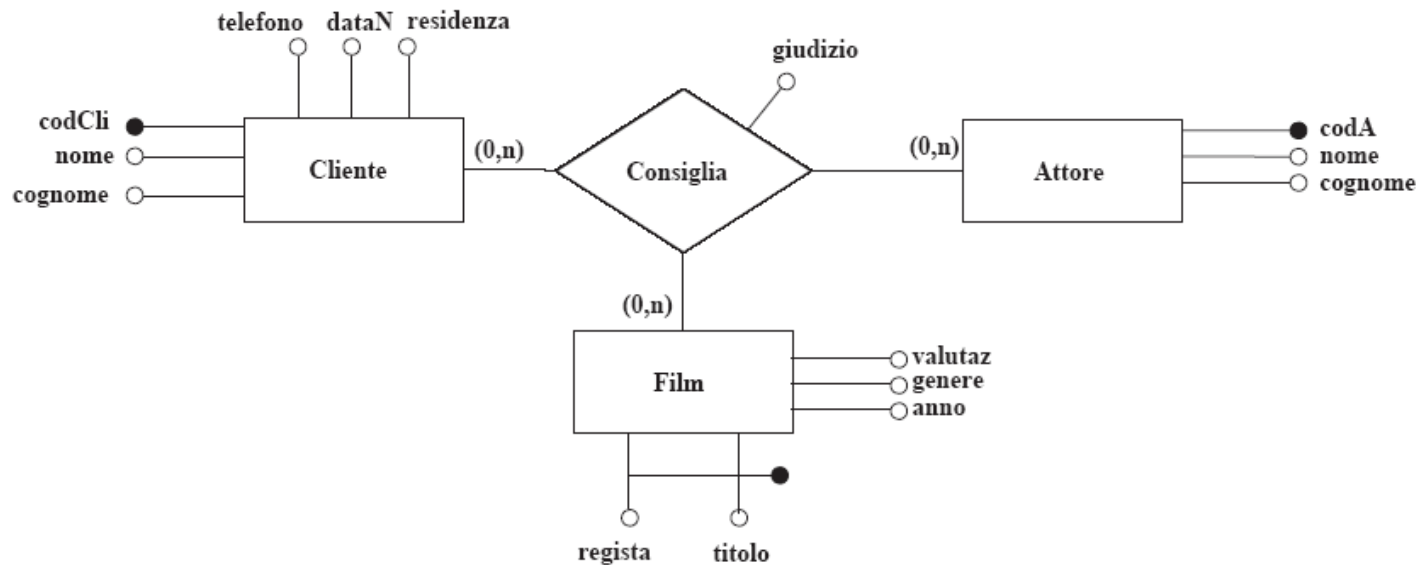
Film(titolo,regista,anno,genere,valutaz)

Attore(codA,nome,cognome)

Consiglia(codCli^{Cliente},titolo^{Film},regista^{Film},codA^{Attore},giudizio)

TRADUZIONE ASSOCIAZIONE: PALESTRA

ESEMPIO 6



Vincolo di integrità: un cliente può consigliare attori in un numero arbitrario di film, ma al più un film per ogni attore

Cliente(codCli, nome, cognome, telefono, dataN, residenza)

Film(titolo, regista, anno, genere, valutaz)

Attore(codA, nome, cognome)

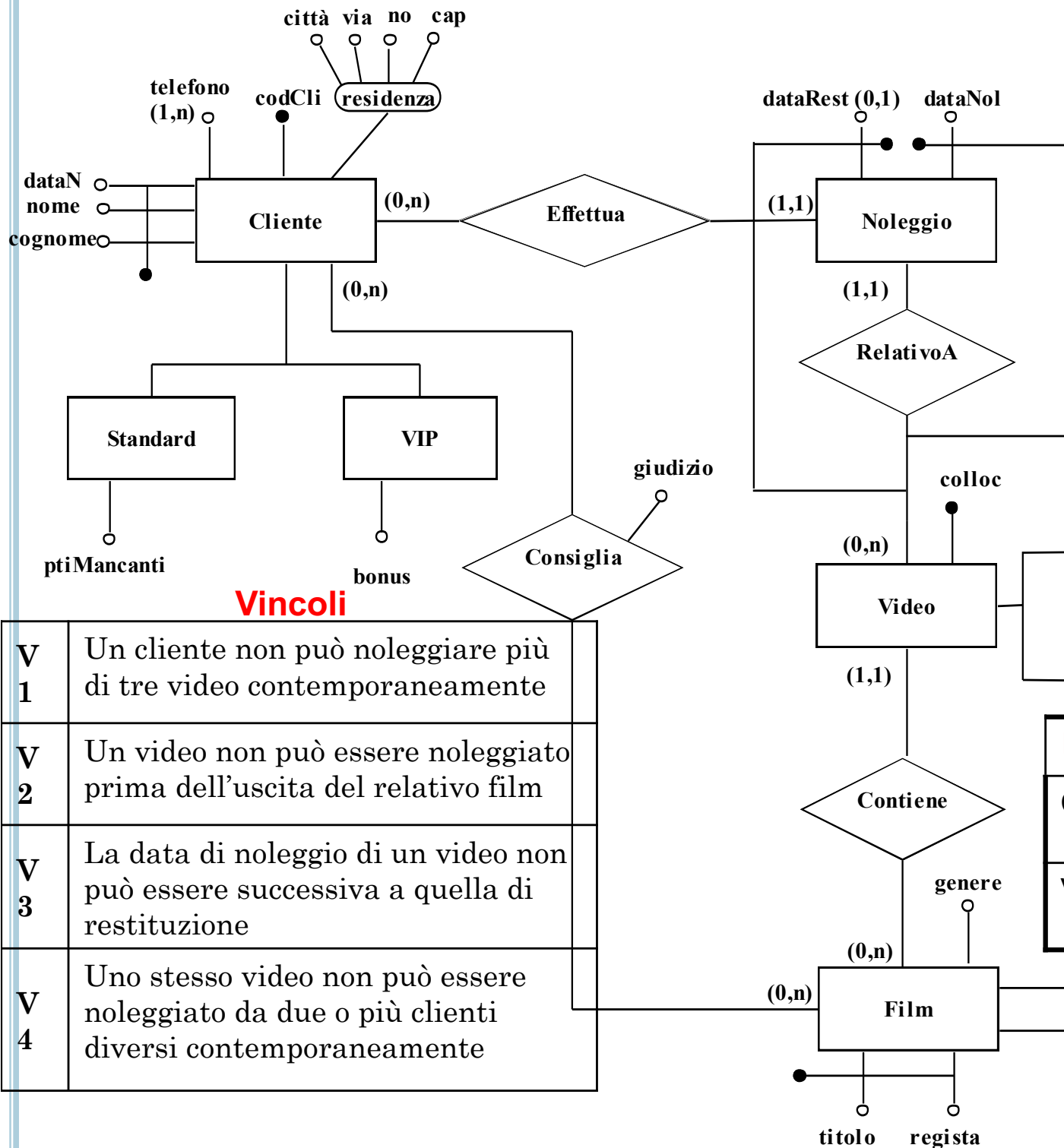
Consiglia(codCli^{Cliente}, titolo^{Film}, regista^{Film}, codA^{Attore}, giudizio)



UN ESEMPIO COMPLETO DI PROGETTAZIONE LOGICA

Progettazione logica

Schema concettuale



Vincoli

V1	Un cliente non può noleggiare più di tre video contemporaneamente
V2	Un video non può essere noleggiato prima dell'uscita del relativo film
V3	La data di noleggio di un video non può essere successiva a quella di restituzione
V4	Uno stesso video non può essere noleggiato da due o più clienti diversi contemporaneamente

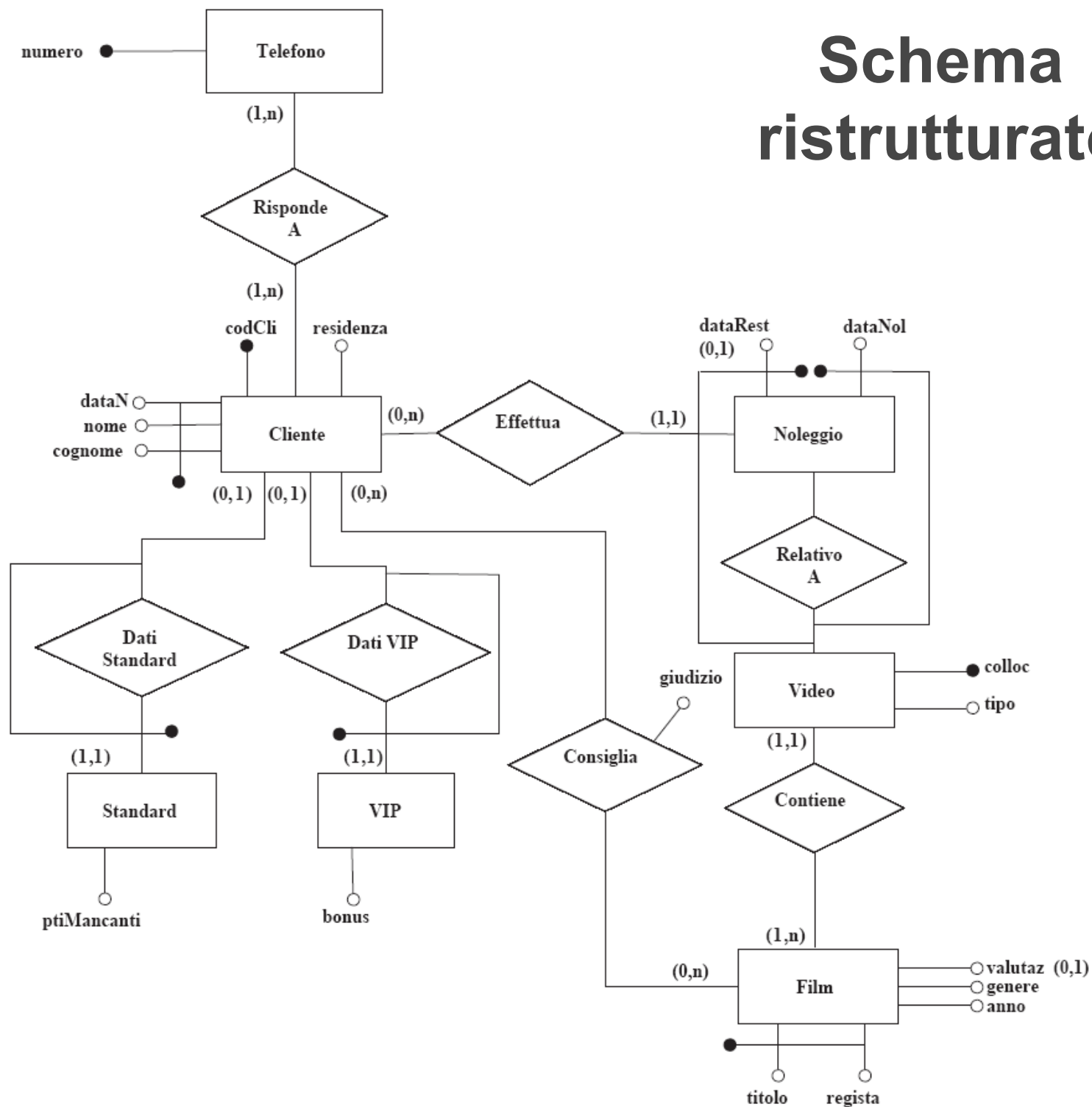
Gerarchie di generalizzazione

E. padre	E. figlie	Tipologia
Cliente	Standard, VIP	Totale esclusiva
Video	DVD, VHS	Totale esclusiva

CARICO DI LAVORO

- Operazione 1
 - Inserisce un nuovo video ed il corrispondente film, se non ancora presente (frequenza: 30 video/mese, 10 film/mese)
- Operazione 2
 - Inserisce un nuovo cliente, classificandolo come cliente standard, indicando tutti i suoi dati anagrafici (frequenza: 5 clienti/settimana)
- Operazione 3
 - Inserisce le informazioni relative ad un nuovo noleggio ed aggiorna i punti mancanti per accedere alla categoria VIP; quando i punti mancanti ad un cliente standard per accedere alla categoria VIP sono zero, il cliente cambia categoria e diventa un cliente VIP a cui viene assegnato un determinato bonus (frequenza: 200 noleggi/giorno)
- Operazione 4
 - Aggiorna i dati del noleggio al momento della restituzione del video da parte di un generico cliente (frequenza: 200 noleggi/giorno)
- Operazione 5
 - Stampa l'elenco dei nomi e delle residenze di tutti i clienti che hanno noleggiato almeno un video da più di 1 settimana e non l'hanno ancora restituito (frequenza: 1 stampa/giorno)
- Operazione 6
 - Stampa l'elenco dei nomi e delle residenze di tutti i clienti VIP, per inviare materiale informativo relativo alla situazione bonus (frequenza: 2 stampe/mese)

Schema ristrutturato



DOCUMENTAZIONE SU SCHEMA RISTRUTTURATO

Vincoli

V1: Un cliente non può noleggiare più di tre video contemporaneamente
V2: Un video non può essere noleggiato prima dell'uscita del relativo film
V3: La data di noleggio di un video non può essere successiva a quella di restituzione
V4: Uno stesso video non può essere noleggiato da due o più clienti diversi contemporaneamente
V5: Ogni cliente è alternativamente un cliente standard od un cliente VIP

SCHEMA RELAZIONALE

Cliente(codCli, *nome*, *cognome*, *dataN*, *residenza*)

Standard(codCli^{Cliente}, *ptiMancanti*)

VIP(codCli^{Cliente}, *bonus*)

Telefono(numero)

Video(colloc, *tipo*, *titolo*^{Film}, *regista*^{Film})

Film(titolo, regista, *anno*, *genere*, *valutaz_o*)

Noleggio(colloc^{Video}, dataNol, *codCli*^{Cliente}, *dataRest_o*)

RispondeA(codCli^{Cliente}, numero^{Telefono})

Consiglia(titolo^{Film}, regista^{Film}, codCli^{Cliente}, *giudizio*)

DOCUMENTAZIONE

Vincoli di integrità	Traduzione in SQL
V_1 : <i>Ogni cliente non può noleggiare più di tre video contemporaneamente</i>	Asserzione
V_2 : <i>Un video non può essere noleggiato prima dell'uscita del film a cui è relativo</i>	Asserzione
V_3 : <i>La data di noleggio di un video non può essere successiva alla data di restituzione</i>	Vincolo su relazione
V_4 : <i>Uno stesso video non può essere noleggiato da due o più clienti diversi contemporaneamente</i>	Asserzione
V_5 : <i>Ogni cliente è alternativamente un cliente standard od un cliente VIP</i>	Asserzione
V_6 : <i>Il tipo di un video è 'd', per i dvd, e 'v', per i vhs</i>	Vincolo su colonna
V_7 : <i>La valutazione di un film è un numero reale compreso tra 0 e 5</i>	Vincolo su colonna
V_8 : <i>Il giudizio espresso da un cliente su un film è un numero intero compreso tra 0 e 5</i>	Vincolo su colonna

OTTIMIZZAZIONI

- La relazione Telefono può essere eliminata
 - ogni numero di telefono che compare nella relazione Telefono compare anche nella relazione RispondeA in quanto la partecipazione dell'entità Telefono all'associazione RispondeA è obbligatoria
- Possiamo inserire un nuovo attributo codF in Film, come chiave primaria
 - La chiave attuale (di due attributi) diventerebbe chiave alternativa