Basi Di Dati e di conoscenza

Progettazione Logica

Progettazione base di dati

Progettare una base di dati significa definirne struttura, caratteristiche e contenuto. E quindi prevede l'uso di opportune metodologie. In base al grado di astrazione, la progettazione prevede:

- **Modello concettuale:** rappresenta la realta` dei dati e le relazioni tra essi attraverso uno schema
- **Modello logico:** **descrive il modo attraverso il quale i dati sono organizzati negli archivi del calcolatore**
- **Modello fisico:** descrive come i dati sono registrati nelle memorie di massa

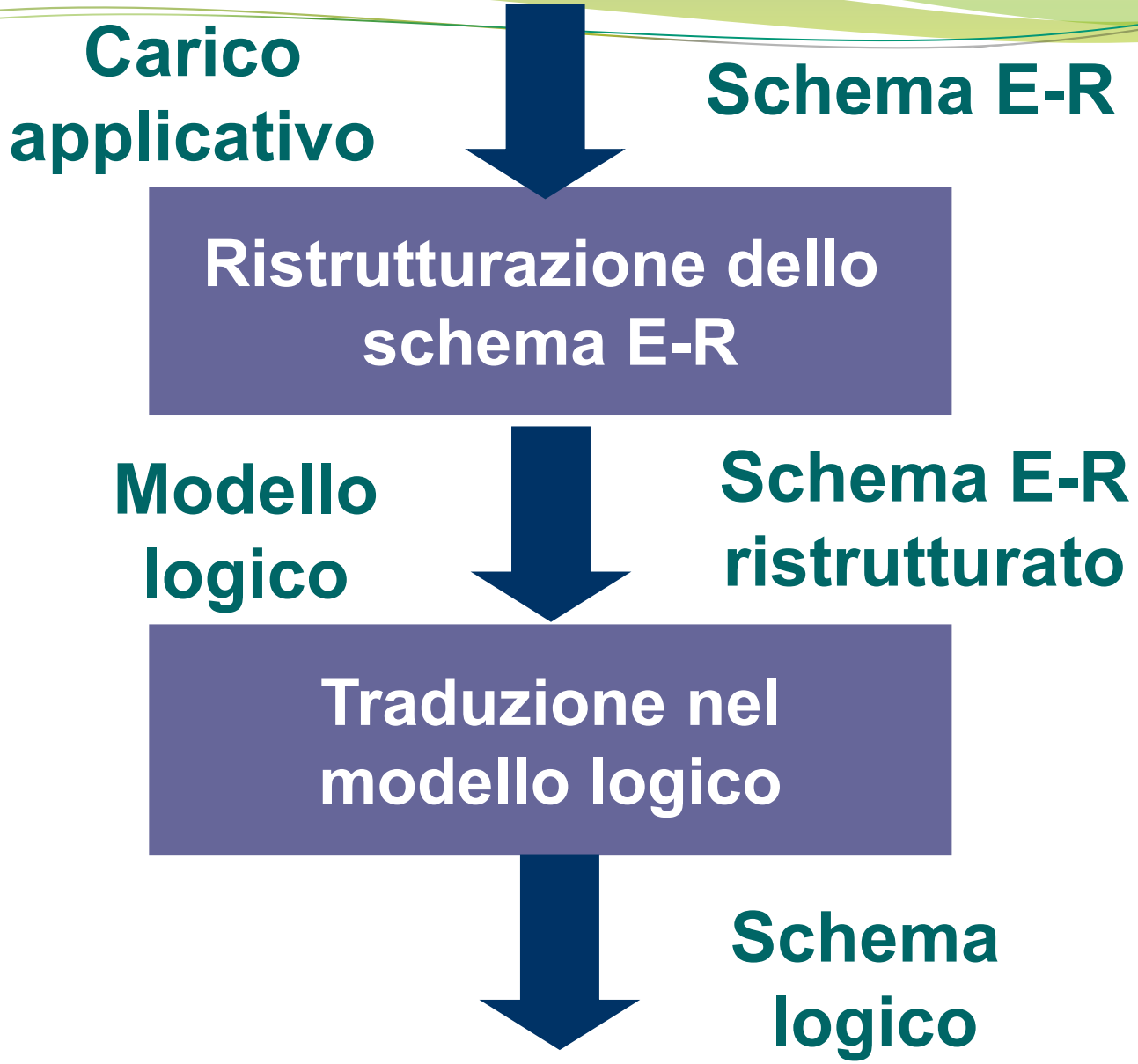
Progettazione Logica

Il prodotto della **progettazione logica** è uno schema logico che rappresenta le informazioni contenute nello schema E-R in modo corretto ed efficiente.

Richiede non solo una ***traduzione*** da uno schema all'altro ma anche una ***ristrutturazione*** dello schema.

Ad es. le generalizzazioni non hanno una traduzione diretta fra modello E-R e modello relazionale. Inoltre bisogna tenere conto delle prestazioni e quindi la riorganizzazione deve avere come obiettivo l'efficienza.

Come passo preliminare bisognerà eseguire un'analisi delle prestazioni attese dal sistema, per ottimizzarne la realizzazione.


Carico applicativo
Schema E-R
Ristrutturazione dello schema E-R
Modello logico
Schema E-R ristrutturato
Traduzione nel modello logico
Schema logico

Analisi delle prestazioni su schemi E-R

In tutti i sistemi software i due principali indici di prestazione che si considerano sono:

- **Costo delle operazioni**
- **Occupazione di memoria**

Nel caso di una base di dati, il costo di un'operazione è misurato in funzione del numero di occorrenze delle entità e delle associazioni fra entità coinvolte.

Per fare questa analisi è necessario conoscere:

- **Volume dei dati** (occorrenze, dimensioni degli attributi)
- **Caratteristiche delle operazioni** (tipo, frequenza, dati coinvolti)

Analisi delle prestazioni su schemi E-R

In genere basta limitarsi ad analizzare il costo delle operazioni più frequenti (regola dell'80/20 *"l'80% del tempo è occupato dal 20% delle operazioni"*)

Per descrivere volume dei dati e caratteristiche delle operazioni si possono costruire la ***tavola dei volumi***, in cui vengono riportati tutti i concetti dello schema con i volumi previsti a regime, e la **tavola delle operazioni**, in cui si riportano il tipo e la frequenza delle operazioni.

Per descrivere le operazioni si può usare il frammento dello schema E-R che comprende i concetti coinvolti nell'operazione, sul quale vengono riportati i cammini logici attraverso i quali si accede ai dati.

Nella tavola degli accessi si riportano i concetti coinvolti, il numero medio degli accessi e il tipo (lettura, scrittura).

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

Ristrutturazione di schemi E-R

Può essere divisa in più passi successivi:

- **Analisi delle ridondanze**: si decide se mantenere o eliminare le eventuali ridondanze.
- **Eliminazione delle generalizzazioni:** si sostituiscono le generalizzazioni con costrutti del modello logico utilizzato
- **Partizionamento/accorpamento di entità e associazioni**: si decide se raggruppare più concetti o partizionare in più concetti i concetti dello schema E-R
- **Scelta degli identificatori primari**: si selezionano gli identificatori per le entità che ne hanno più di uno.

Ridondanze

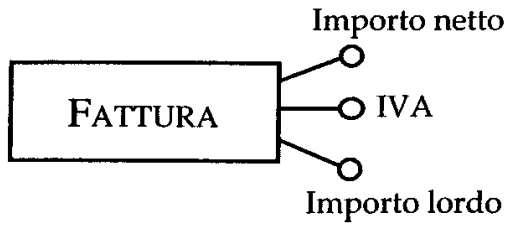
- **Vantaggi**
  - semplificazione delle interrogazioni
- **Svantaggi**
  - appesantimento degli aggiornamenti
  - maggiore occupazione di spazio

Analisi delle ridondanze

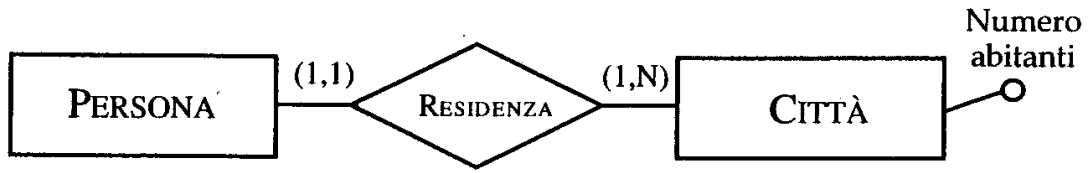
Le ridondanze di uno schema E-R nascono quando un concetto può essere derivato da altri.

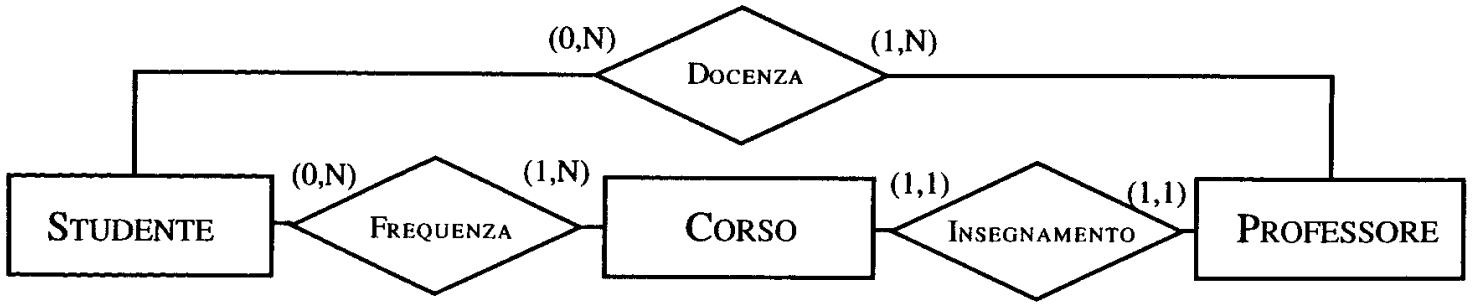
Esempi

- Un attributo può essere derivato da attributi della stessa entità o relazione
- Un attributo può essere derivato da attributi di altre entità e relazioni
- Un attributo può essere derivato mediante conteggio di occorrenze
- Associazioni possono essere derivate da altre associazioni in presenza di cicli


Importo netto
FATTURA
IVA
Importo lordo


Importo totale
ACQUISTO
(1,N)
COMPOSIZIONE
(1,N)
PRODOTTO
Prezzo


PERSONA
(1,1)
RESIDENZA
(1,N)
CITTÀ
Numero abitanti


(0,N)
DOCENZA
(1,N)
STUDENTE
(0,N)
FREQUENZA
(1,N)
CORSO
(1,1)
INSEGNAMENTO
(1,1)
PROFESSORE

Analisi di una ridondanza

Ridondanza

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

- **Operazione 1**: memorizza una nuova persona con la relativa città di residenza (500 volte al giorno)
- **Operazione 2**: stampa tutti i dati di una città (incluso il numero di abitanti) **(2 volte al giorno)**

Presenza di ridondanza

Operazione 1

- **S**: Scrittura
- **L**: Lettura

Concetto	Costrutto	Accessi	Tipo
Persona	Entità	1	S
Residenza	Relazione	1	S
Città	Entità	1	L
Città	Entità	1	S

Operazione 2

Concetto	Costrutto	Accessi	Tipo
Città	Entità	1	L

Assenza di ridondanza

Operazione 1

Concetto	Costrutto	Accessi	Tipo
Persona	Entità	1	S
Residenza	Relazione	1	S

Operazione 2

Concetto	Costrutto	Accessi	Tipo
Città	Entità	1	L
Residenza	Relazione	5000	L

Presenza di ridondanza

- **Costi**:
  - **Operazione 1**: 1500 accessi in scrittura e 500 accessi in lettura al giorno
  - **Operazione** 2: trascurabile.
- Contiamo doppi gli accessi in scrittura

Totale di 3500 accessi al giorno

Assenza di ridondanza

- **Costi**:
  - **Operazione 1**: 1000 accessi in scrittura
  - **Operazione 2**: 10000 accessi in lettura al giorno
- Contiamo doppi gli accessi in scrittura

Totale di 12000 accessi al giorno

Eliminazione delle gerarchie

- **Per eliminare le generalizzazioni (non presenti nel modello relazionale) si possono seguire 3 strategie**:

1. **Accorpamento delle figlie nel padre :**

   si usa una sola entità avente anche tutti gli attributi delle figlie più un attributo che indica il "tipo". (più memoria meno accessi)

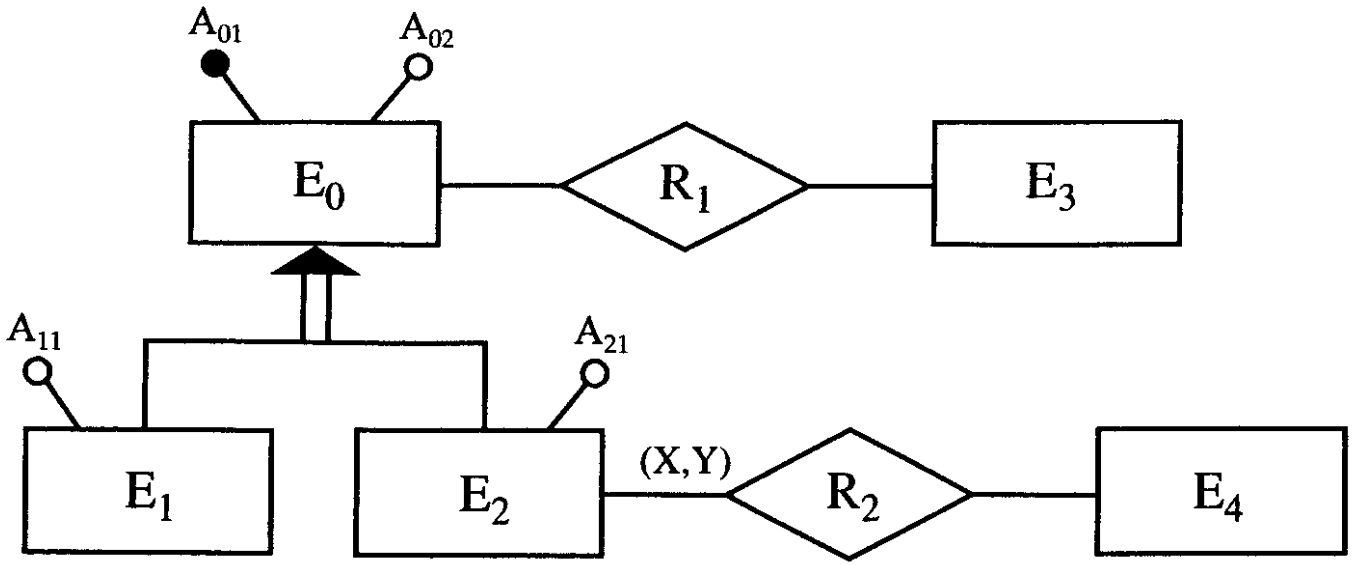
2. **Accorpamento del padre nelle figlie** :

   si elimina il padre ed i suoi attributi vengono inseriti in ciascuna delle figlie (possibile solo se la generalizzazione è totale, meno memoria rispetto al caso 1, meno accessi rispetto al 3).

3. **Sostituzione della generalizzazione con associazioni :**

   la generalizzazione si trasforma in una serie di associazioni, con il vincolo che ogni occorrenza della entità padre non può partecipare a più di una di tali associazioni (meno memoria rispetto a 1, ma più accessi per rispettare i vincoli aggiuntivi).

Eliminazione delle gerarchie

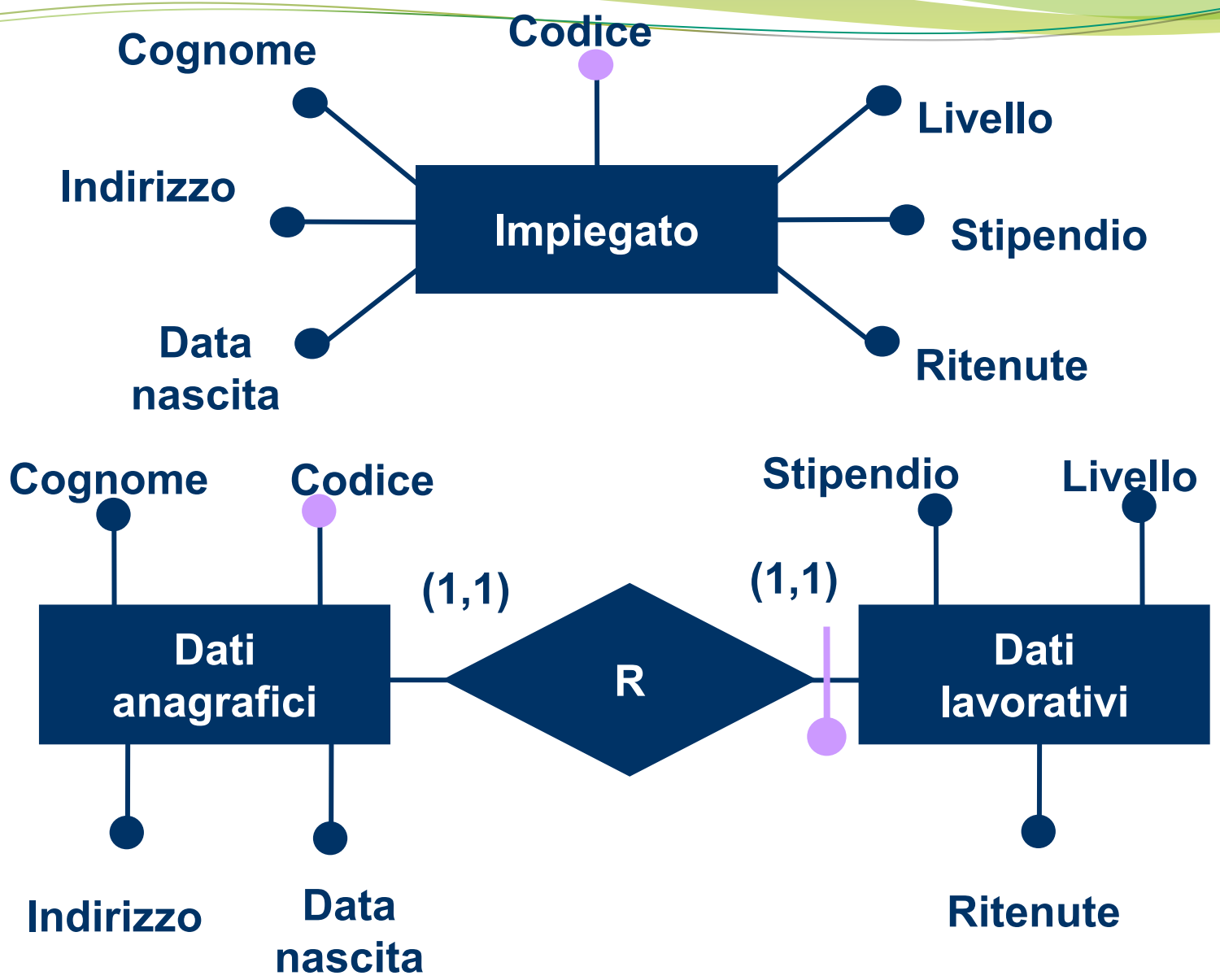


Partizionamento/accorpamento di concetti

- Gli accessi possono essere ridotti separando gli attributi di uno stesso concetto cui fanno accesso operazioni diverse e raggruppando attributi di concetti diversi cui fanno accesso le stesse operazioni.

Partizionamenti di entità:

- Si possono fare in modo verticale, decomponendo l'entità in più entità sulla base degli attributi, o in modo orizzontale, decomponendo l'entità sulla base delle sue occorrenze.
- Una decomposizione orizzontale corrisponde ad una generalizzazione. Costringe a duplicare tutte le associazioni cui l'entità originaria partecipa


Cognome
Codice
Livello
Indirizzo
Impiegato
Stipendio
Data nascita
Ritenute
Cognome
Codice
Stipendio
Livello
(1,1)
(1,1)
Dati anagrafici
R
Dati lavorativi
Indirizzo
Data nascita
Ritenute

Partizionamento/accorpamento di concetti

Eliminazione di attributi multivalore:

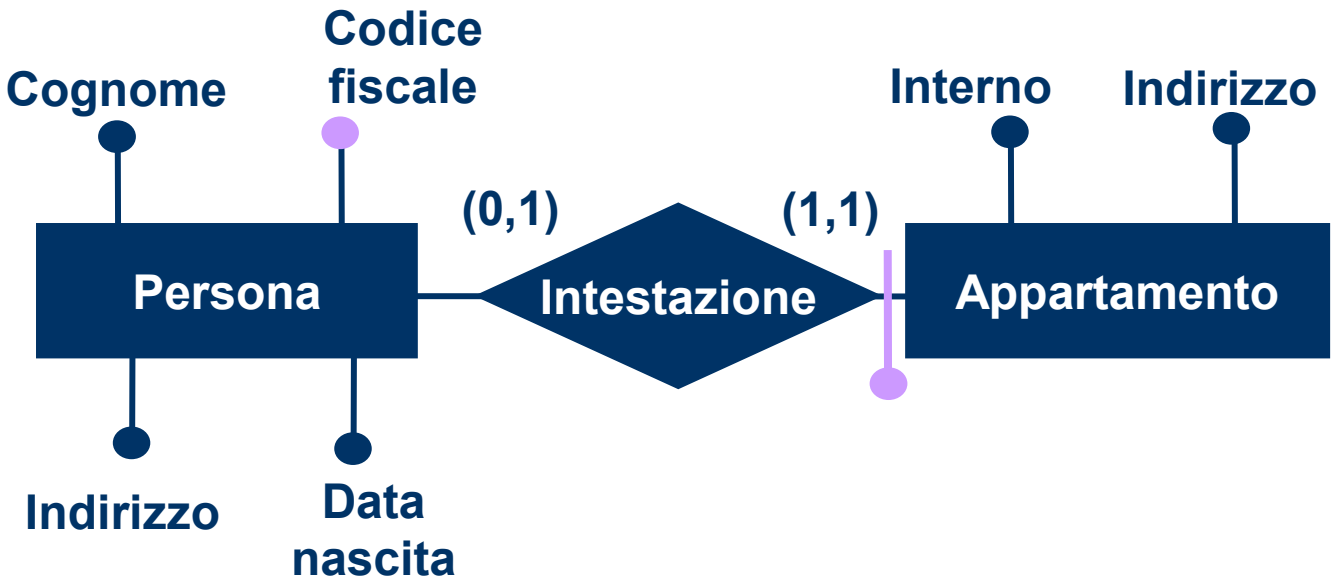
- il modello relazionale non prevede questo tipo di attributo e richiede che venga sostituito da una relazione o nella de**composizione diretta dell'attributo multivalore in più attributi**.

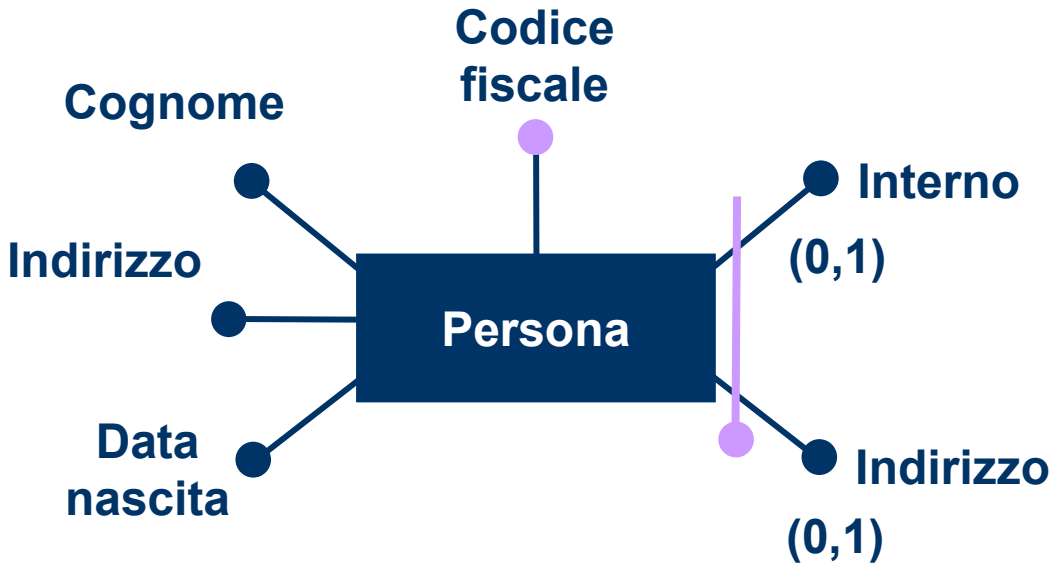
Accorpamento di entità:

- spesso conviene accorpare due entità coinvolte spesso nella stessa operazione in una sola entità. Questo può causare la presenza di valori nulli.
- **Si effettuano di solito su relazioni di tipo uno a uno,** per evitare la creazione di ridondanze.

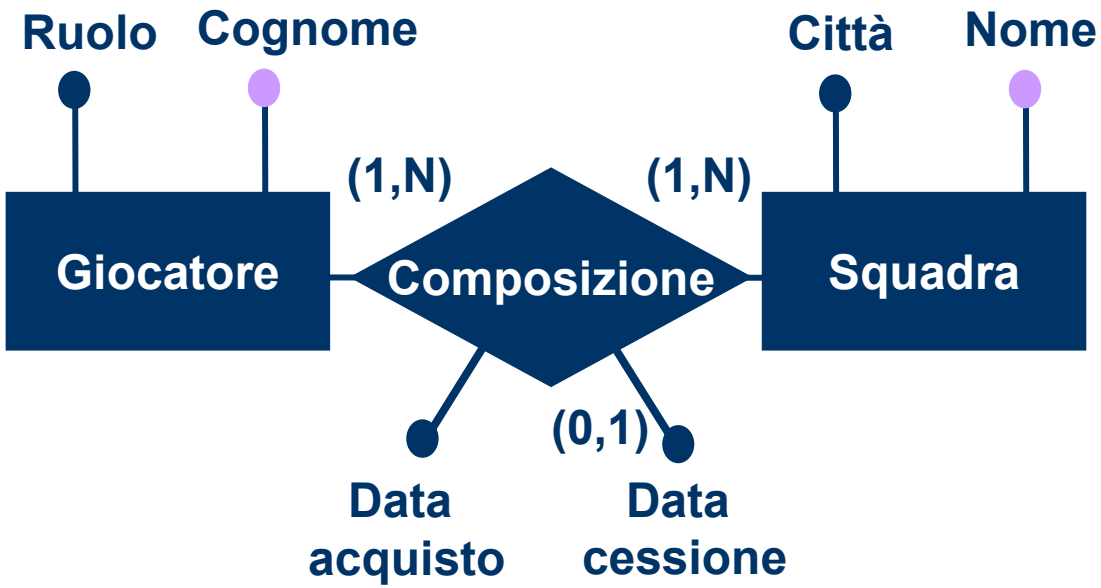
- E' possibile anche accorpare/partizionare associazioni.

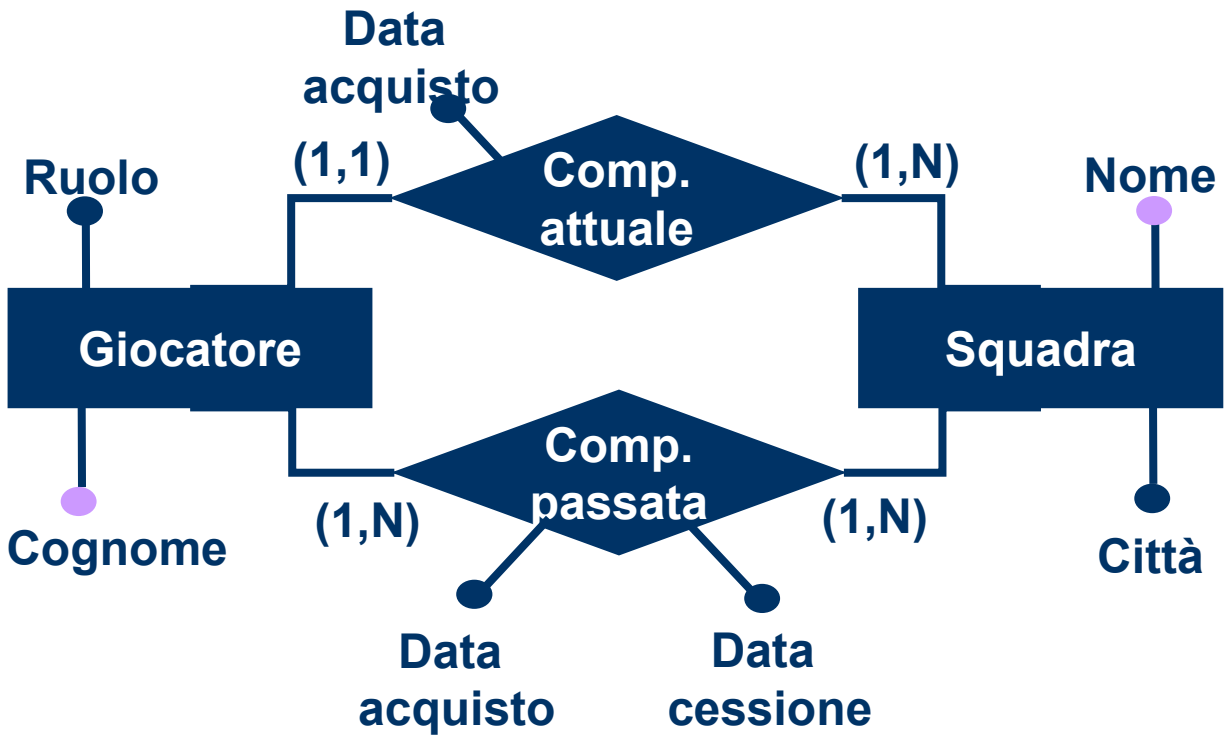
Attributi multivalore





Partizione associazioni





Scelta degli identificatori principali

E' molto importante per l'importanza rivestita dalle chiavi nel modello relazionale. Bisogna scegliere una chiave principale secondo i seguenti criteri:

1. Escludere gli attributi con valori nulli
2. Preferire identificatori con pochi attributi
3. Preferire identificatori interni rispetto ad identificatori esterni
4. Utilizzare identificatori che vengono utilizzati da molte operazioni per accedere alle occorrenze di una entità.
5. Se nessun identificatore soddisfa i criteri, è consigliabile crearne uno (codice identificativo)

Progettazione Logica: Traduzione

- **Una volta riorganizzato lo schema si può passare alla traduzione dal modello E-R a quello relazionale.**

Traduzione di associazioni molti a molti

Tipicamente, passando dal modello E-R a quello relazionale, si procede nel modo seguente:

1. ogni entità diventa una relazione sugli attributi dell'entità
2. ogni associazione diventa una relazione definita sugli attributi dell'associazione più gli identificatori delle entità coinvolte, che costituiranno la chiave della relazione.
3. Si instaura un vincolo di integrità referenziale verso gli identificatori delle entità coinvolte da parte dei corrispondenti attributi dell'associazione.
4. In alcuni casi (es. associazioni ricorsive) è necessario ridenominare gli attributi.

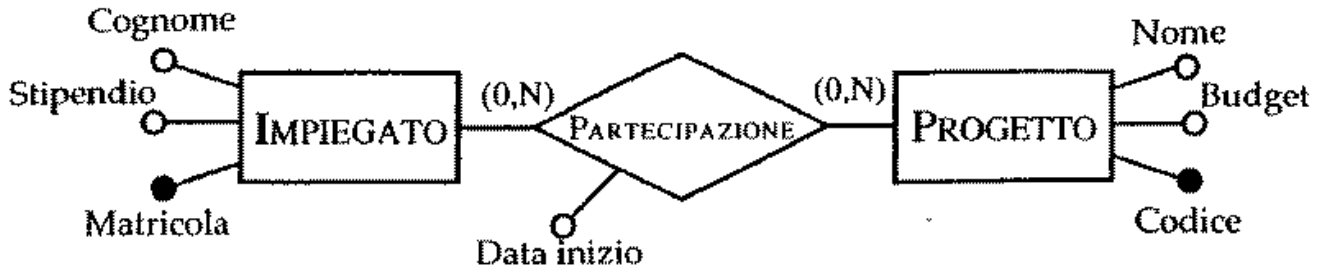


Figura 7.17 Schema E-R con associazione molti a molti

Lo schema relazionale che si ottiene è quindi il seguente:

IMPIEGATO(Matricola, Cognome, Stipendio)
PROGETTO(Codice, Nome, Budget)
PARTECIPAZIONE(Matricola, Codice), DataInizio)

con vincoli di integrità referenziale fra

- **Matricola in Partecipazione e (la chiave di) Impiegato**
- **Codice in Partecipazione e (la chiave di) Progetto**

Relazioni ricors

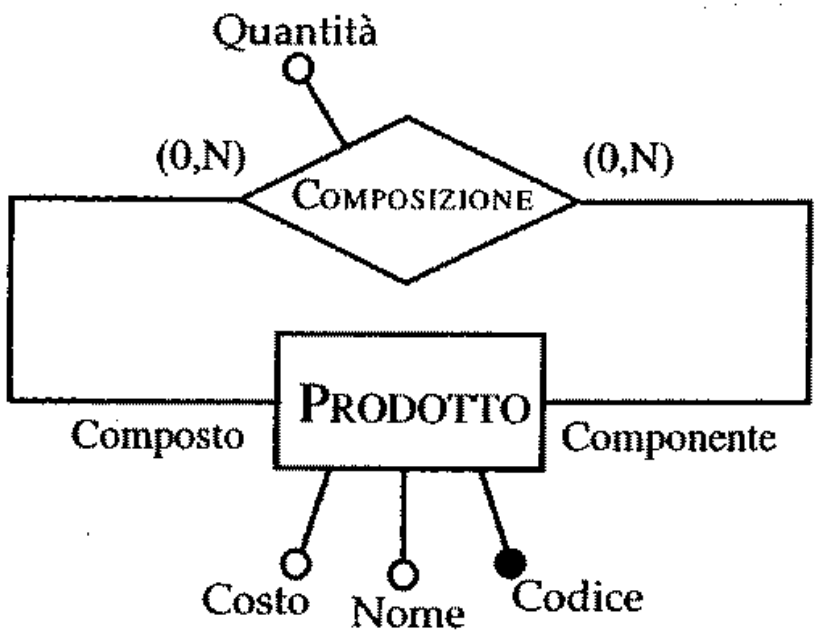
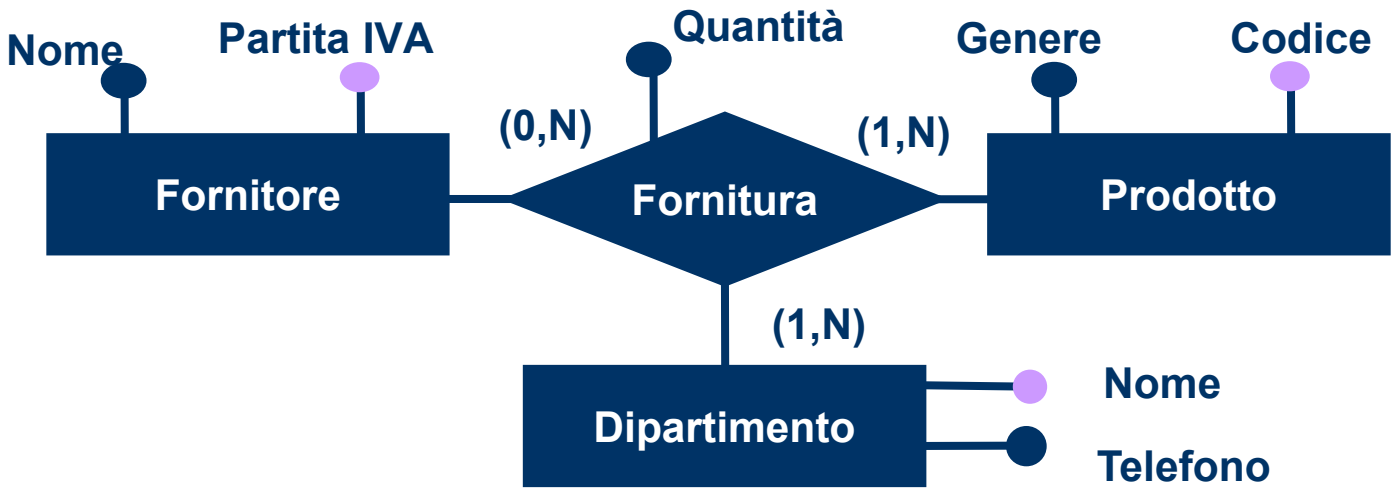


Figura 7.18 Schema E-R con associazione ricorsiva

Questo schema si traduce nelle due relazioni:

PRODOTTO(Codice, Nome, Costo)
COMPOSIZIONE(Composto, Componente, Quantità)

Associazioni n-arie



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

Traduzione di associazioni uno a molti

In questo caso si può usare una sola relazione per tradurre sia l'associazione che l'entità per la quale l'associazione può applicarsi una sola volta: la chiave sarà l'identificatiore dell'entità.

Es. giocatore, squadra e contratto: ogni giocatore può avere un solo contratto.

Si instaura un vincolo di integrità referenziale verso la chiave della entità (es. squadra) che può comparire più volte nella relazione da parte del corrispondente attributo della relazione che traduce sia l'altra entità che l'associazione.

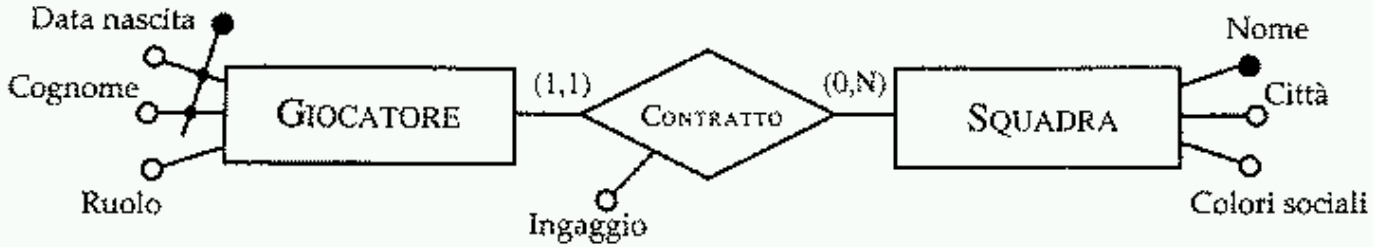


Figura 7.20 Schema E-R con associazione uno a molti

Secondo la regola vista per le associazioni molti a molti, la traduzione di questo schema dovrebbe essere la seguente:

GIOCATORE(Cognome, DataNascita, Ruolo)

SQUADRA(Nome, Città, ColoriSociali)

CONTRATTO(Giocatore, DataNascitaGiocatore, NomeSquadra, Ingaggio)

Meglio (più compatto):

- **Giocatore(Cognome, DataNasc, Ruolo, Squadra, Ingaggio)**
- **Squadra(Nome, Città, ColoriSociali)**
- con vincolo di integrità referenziale fra Squadra in Giocatore e la chiave di Squadra
- se la cardinalità minima della relazione è 0, allora Squadra in Giocatore deve ammettere valore nullo

Traduzione di entità con identificatore esterno

Poiché un'entità identificata esternamente tramite un'associazione partecipa all'associazione con cardinalità minima e massima pari ad uno, si ricade nel caso precedente della relazione uno a molti.

L'associazione e l'entità che la utilizza per avere un identificatore esterno vengono tradotti con la stessa relazione.

.

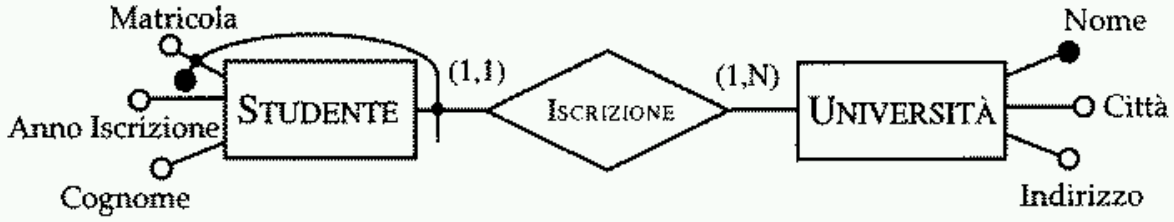


Figura 7.21 Schema E-R con identificatore esterno

STUDENTE(Matricola, NomeUniversità, Cognome, AnnoIscrizione)
UNIVERSITÀ(Nome, Città, indirizzo)

Traduzione di associazioni uno a uno

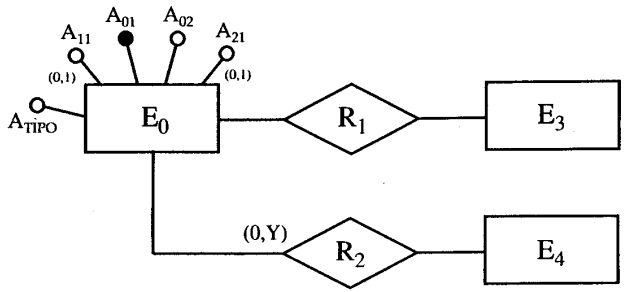
Se la partecipazione è obbligatoria per entrambe le entità, si può operare come nella traduzione delle associazioni uno a molti, però possiamo scegliere quale entità tradurre in una stessa relazione insieme all'associazione.

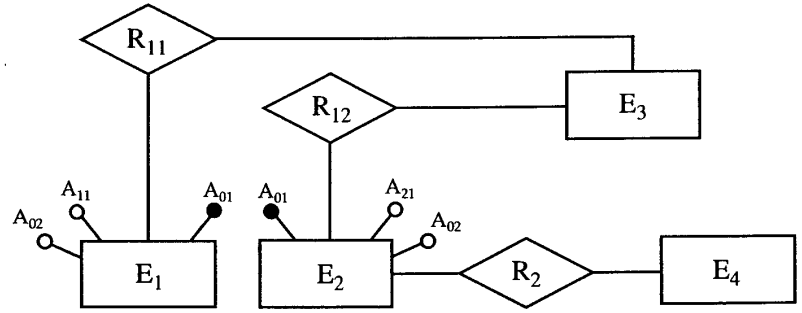
Se è opzionale per una sola entità, è preferibile scegliere di effettuare la fusione in modo da non avere valori nulli.

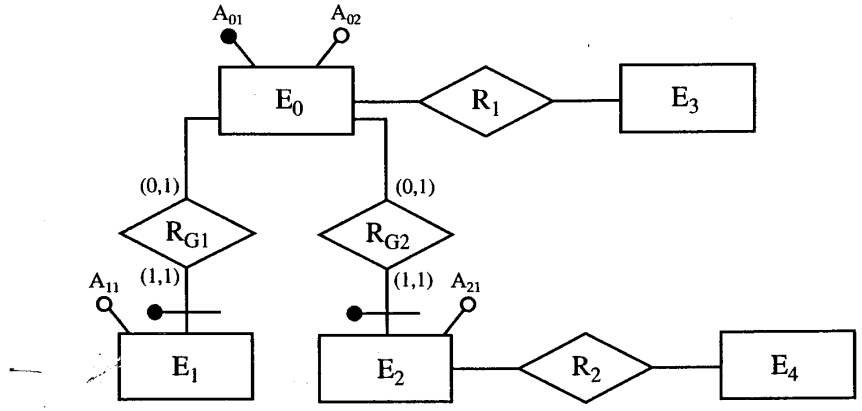
Se è opzionale per entrambe, con una traduzione mediante 3 relazioni separate si può evitare la presenza di valori nulli.

Traduzione

- Se si devono tradurre schemi complessi, si può procedere per gradi:
  - traduzione delle entità
  - traduzione delle entità con identificazioni esterne
  - traduzione delle associazioni rimaste (alcune sono già state tradotte al punto precedente)


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

ER-to-Relational Mapping strategy

- **ER-to-Relational Mapping Algorithm**
  Step 1: Mapping of Regular Entity Types
  Step 2: Mapping of Weak Entity Types
  Step 3: Mapping of Binary 1:1 Relation Types
  Step 4: Mapping of Binary 1:N Relationship Types.
  Step 5: Mapping of Binary M:N Relationship Types.
  Step 6: Mapping of Multivalued attributes.
  Step 7: Mapping of N-ary Relationship Types.

- **Mapping EER Model Constructs to Relations**
  Step 8: Options for Mapping Specialization or Generalization.
  Step 9: Mapping of Union Types (Categories).

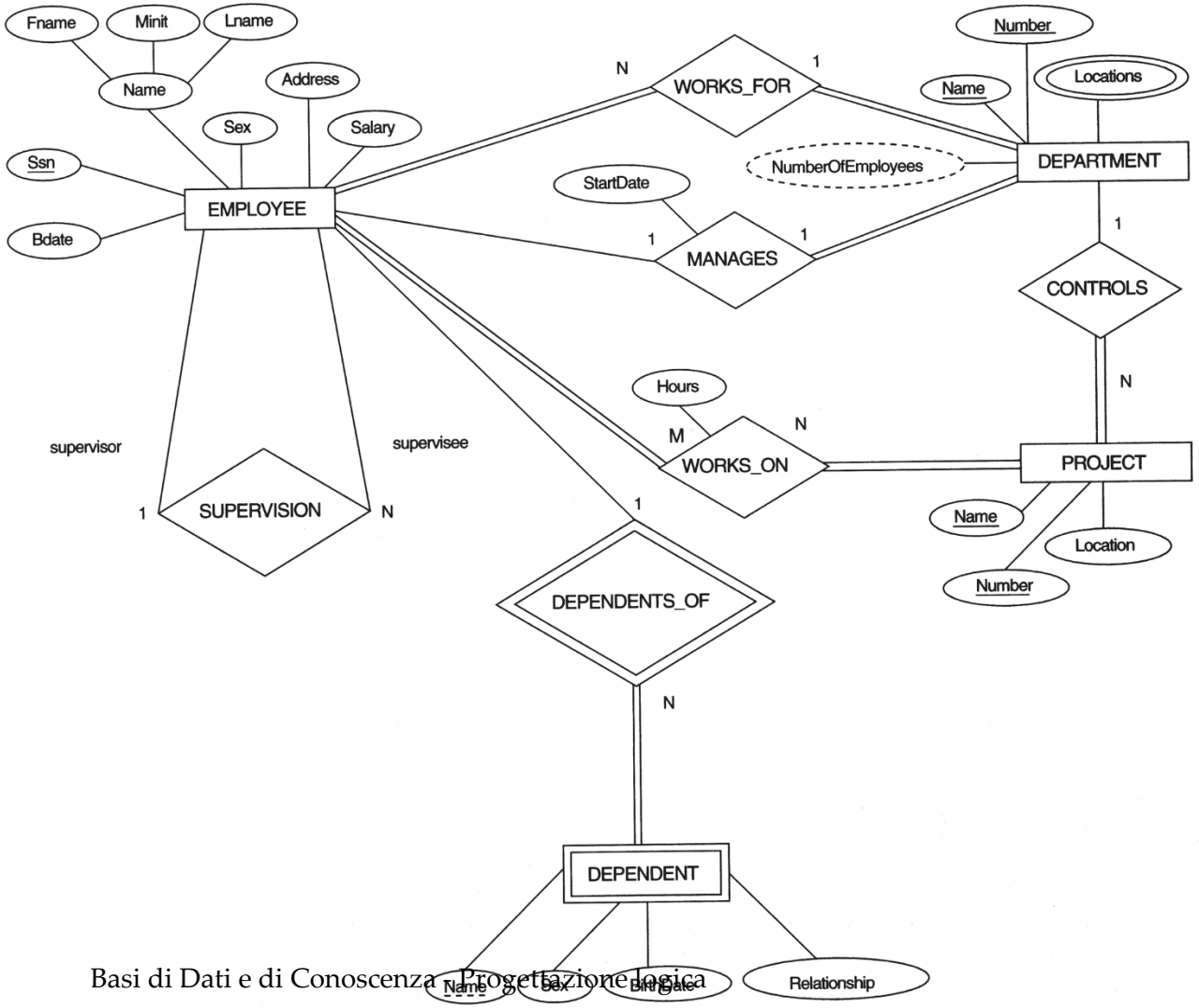
ER-to-Relational Mapping Algorithm

- **Step 1: Mapping of Regular Entity Types.**

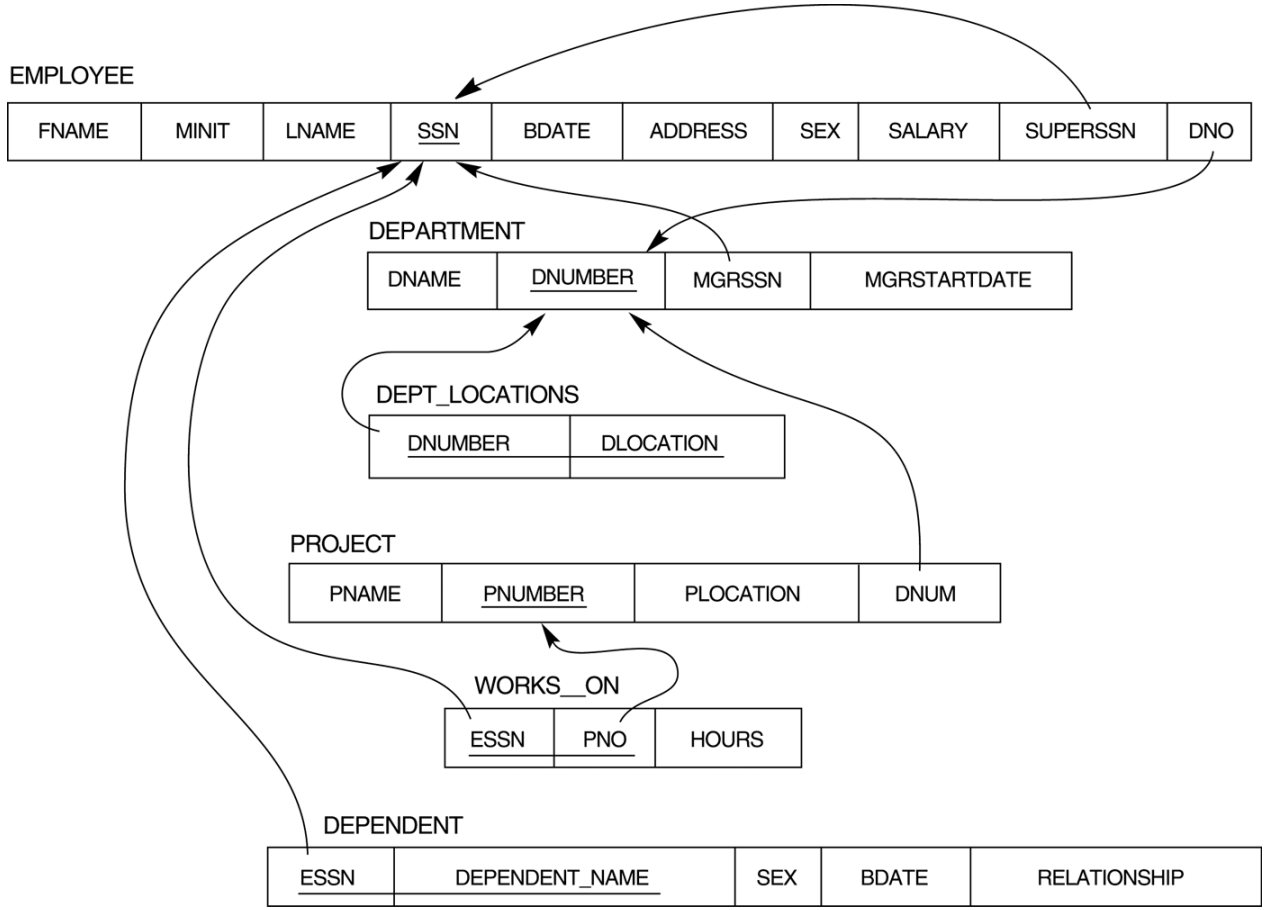
  - For each regular (strong) entity type E in the ER schema, create a relation R that includes all the simple attributes of E.
  - Choose one of the key attributes of E as the primary key for R. If the chosen key of E is composite, the set of simple attributes that form it will together form the primary key of R.

  Example: We create the relations EMPLOYEE, DEPARTMENT, and PROJECT in the relational schema corresponding to the regular entities in the ER diagram. SSN, DNUMBER, and PNUMBER are the primary keys for the relations EMPLOYEE, DEPARTMENT, and PROJECT as shown.

The ER conceptual schema diagram for the COMPANY database.



Result of mapping the COMPANY ER schema into a relational schema.



ER-to-Relational Mapping Algorithm (cont)

- **Step 2: Mapping of Weak Entity Types**

  - For each weak entity type W in the ER schema with owner entity type E, create a relation R and include all simple attributes (or simple components of composite attributes) of W as attributes of R.
  - In addition, include as foreign key attributes of R the primary key attribute(s) of the relation(s) that correspond to the owner entity type(s).
  - The primary key of R is the *combination of* the primary key(s) of the owner(s) and the partial key of the weak entity type W, if any.

  Example: Create the relation DEPENDENT in this step to correspond to the weak entity type DEPENDENT. Include the primary key SSN of the EMPLOYEE relation as a foreign key attribute of DEPENDENT (renamed to ESSN).

  The primary key of the DEPENDENT relation is the combination {ESSN, DEPENDENT_NAME} because DEPENDENT_NAME is the partial key of DEPENDENT.

ER-to-Relational Mapping Algorithm (cont)

- **Step 3: Mapping of Binary 1:1 Relation Types**

  For each binary 1:1 relationship type R in the ER schema, identify the relations S and T that correspond to the entity types participating in R. There are three possible approaches:

  (1) Foreign Key approach: Choose one of the relations-S, say-and include a foreign key in S the primary key of T. It is better to choose an entity type with *total participation* in R in the role of S.

  Example: 1:1 relation MANAGES is mapped by choosing the participating entity type DEPARTMENT to serve in the role of S, because its participation in the MANAGES relationship type is total.

  (2) Merged relation option: An alternate mapping of a 1:1 relationship type is possible by merging the two entity types and the relationship into a single relation. This may be appropriate when *both participations are total.*

  (3) Cross-reference or relationship relation option: The third alternative is to set up a third relation R for the purpose of cross-referencing the primary keys of the two relations S and T representing the entity types.

ER-to-Relational Mapping Algorithm (cont)

- **Step 4: Mapping of Binary 1:N Relationship Types.**
  - For each regular binary 1:N relationship type R, identify the relation S that represent the participating entity type at the N-side of the relationship type.
  - Include as foreign key in S the primary key of the relation T that represents the other entity type participating in R.
  - Include any simple attributes of the 1:N relation type as attributes of S.

  Example: 1:N relationship types WORKS_FOR, CONTROLS, and SUPERVISION in the figure. For WORKS_FOR we include the primary key DNUMBER of the DEPARTMENT relation as foreign key in the EMPLOYEE relation and call it DNO.

ER-to-Relational Mapping Algorithm (cont)

- **Step 5: Mapping of Binary M:N Relationship Types**.
  - For each regular binary M:N relationship type R, *create a new relation* S to represent R.
  - Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types; *their combination will form the primary key* of S.
  - Also include any simple attributes of the M:N relationship type (or simple components of composite attributes) as attributes of S.

  Example: The M:N relationship type WORKS_ON from the ER diagram is mapped by creating a relation WORKS_ON in the relational database schema. The primary keys of the PROJECT and EMPLOYEE relations are included as foreign keys in WORKS_ON and renamed PNO and ESSN, respectively.

  Attribute HOURS in WORKS_ON represents the HOURS attribute of the relation type. The primary key of the WORKS_ON relation is the combination of the foreign key attributes {ESSN, PNO}.

ER-to-Relational Mapping Algorithm (cont)

- **Step 6: Mapping of Multivalued attributes**.
  - For each multivalued attribute A, create a new relation R. This relation R will include an attribute corresponding to A, plus the primary key attribute K-as a foreign key in R-of the relation that represents the entity type of relationship type that has A as an attribute.
  - The primary key of R is the combination of A and K. If the multivalued attribute is composite, we include its simple components.

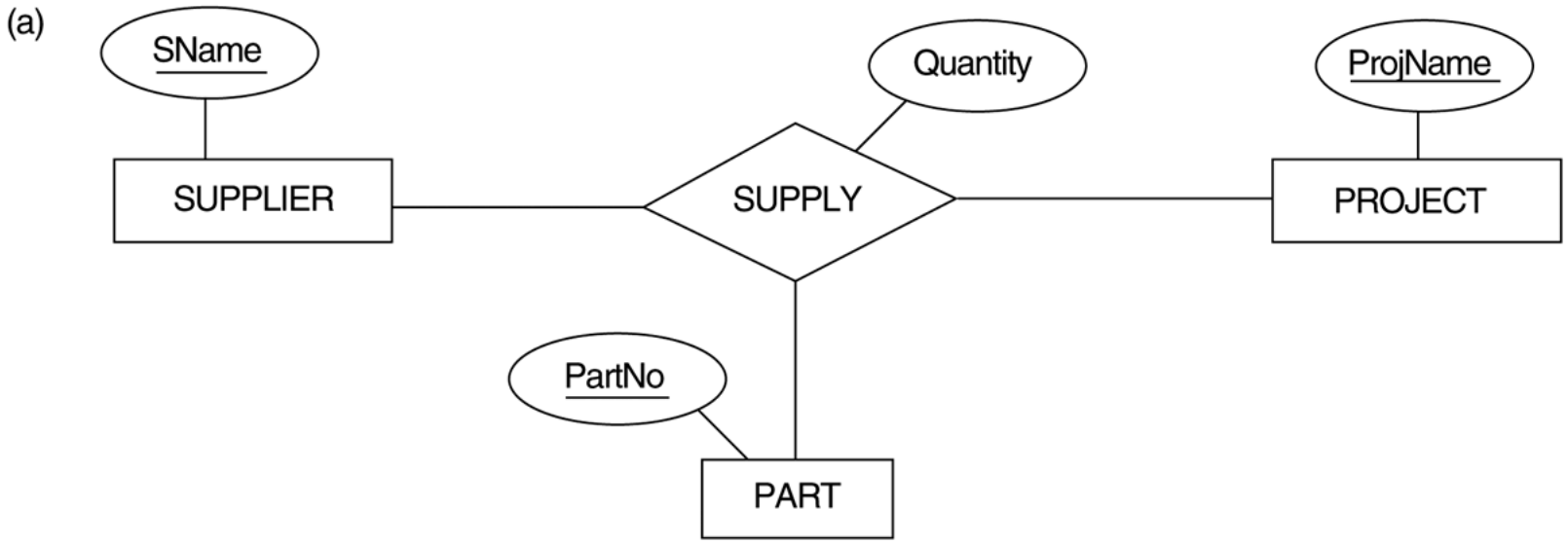
  Example: The relation DEPT_LOCATIONS is created. The attribute DLOCATION represents the multivalued attribute LOCATIONS of DEPARTMENT, while DNUMBER-as foreign key-represents the primary key of the DEPARTMENT relation. The primary key of R is the combination of {DNUMBER, DLOCATION}.

ER-to-Relational Mapping Algorithm (cont)

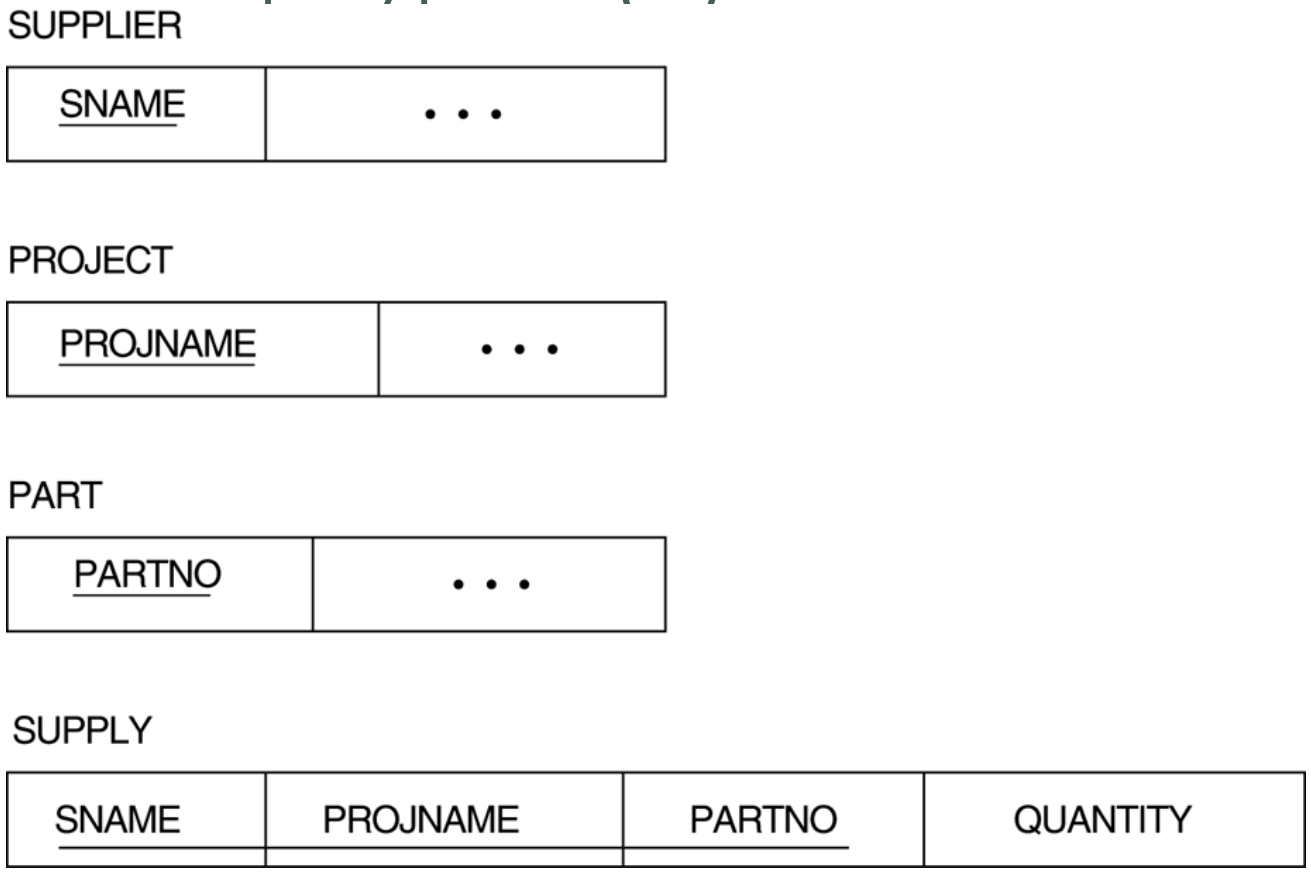
- **Step 7: Mapping of N-ary Relationship Types.**
  - For each n-ary relationship type R, where n>2, create a new relationship S to represent R.
  - Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types.
  - Also include any simple attributes of the n-ary relationship type (or simple components of composite attributes) as attributes of S.

  Example: The relationship type SUPPY in the ER below. This can be mapped to the relation SUPPLY shown in the relational schema, whose primary key is the combination of the three foreign keys {SNAME, PARTNO, PROJNAME}

Ternary relationship types. (a) The SUPPLY relationship.



Ternary relationship types. (a) The SUPPLY relationship.



Summary of Mapping constructs and constraints

Correspondence between ER and Relational Models

ER Model	Relational Model
Entity type	"Entity" relation
1:1 or 1:N relationship type	Foreign key (or "relationship" relation)
M:N relationship type	"Relationship" relation and two foreign keys
n-ary relationship type	"Relationship" relation and n foreign keys
Simple attribute	Attribute
Composite attribute	Set of simple component attributes
Multivalued attribute	Relation and foreign key
Value set	Domain
Key attribute	Primary (or secondary) key