



# Basi Dati

Uso del modello entità relazione

a.a. 2021/2022  
Prof.ssa G. Tortora

# Progettazione di un Database

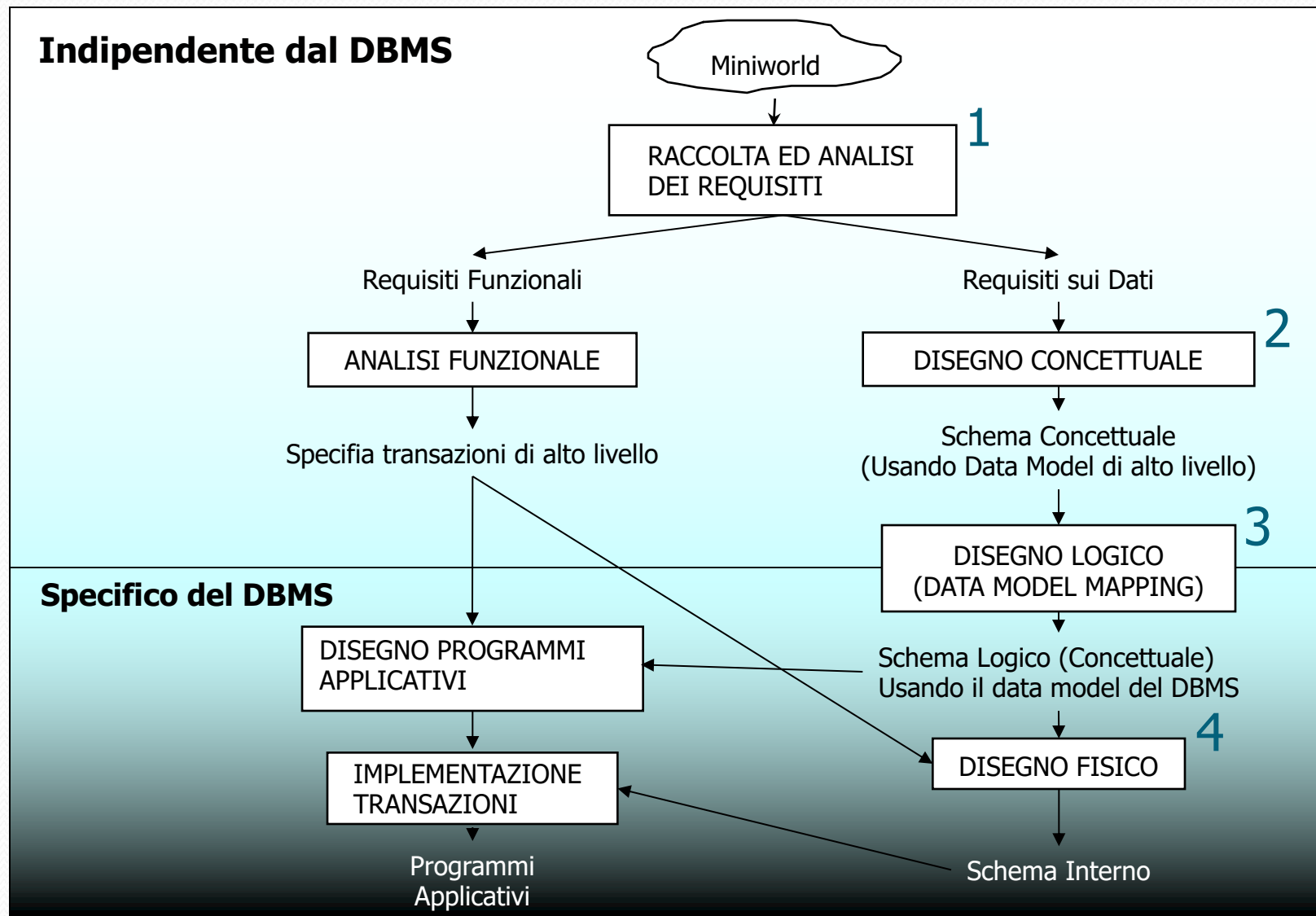


# Il processo di progettazione del Database

Il processo consta di quattro fasi:

1. Raccolta e analisi dei requisiti
2. Disegno del database concettuale
3. Disegno del database logico
4. Disegno del database fisico

# Le fasi di progettazione di un database





# 1. Raccolta e analisi dei requisiti

- Il progettista del DB intervista i potenziali utenti dei DB per capire e documentare i requisiti utente.

*Output:*

- Requisiti sui dati
- Requisiti funzionali
  - Operazioni definite dall'utente (transazioni) che saranno applicate al DB.
    - *Es:* aggiornamenti, ricerche, ...
  - Specifica dei requisiti funzionali attraverso Data Flow Diagrams.

## 2. Disegno del database concettuale

- Creazione dello schema concettuale, usando un data model concettuale ad alto livello.

### *Output:*

- Descrizione concisa dei requisiti utente dei dati inclusiva di una descrizione dettagliata di tipi di dati, relazioni e vincoli fatta usando i concetti del data model ad alto livello.
  - Poiché non vi sono dettagli implementativi, tale descrizione può essere usata per comunicare con utenti non tecnici.
  - Tale documentazione può essere usata anche come riferimento per assicurarsi di aver considerato tutti i requisiti dell'utente.



# Disegno del database concettuale (2)

- Questo approccio abilita il progettista a concentrarsi sui dati ignorando i dettagli di memorizzazione.
- Dopo aver disegnato lo schema concettuale, le operazioni di base del data model possono essere usate per specificare le operazioni di alto livello individuate durante l'analisi funzionale (*passo 1*).

### 3. Disegno del database logico

- Implementazione del database usando un DBMS commerciale.
  - Detto anche **data model mapping**.
  - Molti DBMS usano un data model di implementazione, in modo da trasformare facilmente lo schema concettuale da data model ad alto livello in data model di implementazione.

#### *Output:*

- Schema del database nel data model di implementazione specifico del DBMS scelto.



## 4. Disegno del database fisico

- Specifica delle strutture di memorizzazione interne, degli access path e dell'organizzazione dei file.
- Progettazione dei programmi applicativi e implementazione delle transazioni.

# Il processo di progettazione del database

1. **Raccolta e analisi dei requisiti**
  2. **Disegno del database concettuale**
  3. Disegno del database logico
  4. Disegno del database fisico
- La prima fase mal si presta all'utilizzo di rigorosi formalismi, essendo essenzialmente un grosso documento di testo.
  - La seconda fase, invece, permette la definizione di modelli formali per supportare il lavoro del progettista. Il più diffuso di questi modelli è l'**Entity-Relationship (ER)** ...



# Il modello Entity-Relationship (ER)





# E.R.: Introduzione

- Il modello Entità-Relazione (ER) è un diffusissimo data model di alto livello, estesamente utilizzato per definire lo schema concettuale di un database.
- È stato concepito per essere più vicino ai concetti “*umani*”, e quindi facilmente comprensibile anche ad utenti non tecnici.
- Il modello ER ha avuto una grandissima diffusione principalmente per i formalismi grafici semplici e chiari che incorpora.

## E.R.: Introduzione (2)

- Il modello ER è utilizzato in molti tool per la progettazione di database.
  - *Es.* Platinum ER-Win.
- Esistono degli algoritmi per convertire automaticamente un modello ER in uno schema di database per DBMS commerciali.
- Fu introdotto da Chen nel 1976.
- È stato migliorato negli anni da Chen ed altri (tra cui Elmasri), portando all'Enhanced-ER (EER).



## E.R.: Introduzione (3)

- Il modello ER descrive i dati con tre concetti fondamentali:
  - Entità
  - Attributi
  - Relazioni



# Entità

- Le entità corrispondono a **classi di oggetti** del mondo reale (fatti, persone, ...) che hanno proprietà omogenee, comuni ed esistenza “*autonoma*” ai fini dell’applicazione di interesse.
  - Un’entità può essere un oggetto fisico (casa, impiegato, ...) o un oggetto concettuale (un lavoro, una società, ...).
- Ogni entità ha un nome che la identifica univocamente nello schema e viene rappresentata graficamente con un **rettangolo** con il nome dell’entità all’interno.

Impiegato

Rappresentazione ER  
dell’entità “*Impiegato*”

Studente

Rappresentazione ER  
dell’entità “*Studente*”

# Attributi

- Ogni entità ha delle proprietà dette **attributi**.
  - *Es:* l'entità *Impiegato* ha attributi **nome**, **età**, **indirizzo**, **salario**, **telefono**, ...
- Ogni entità è caratterizzata da un valore per i suoi attributi.
- Ogni attributo ha un nome che lo identifica e viene rappresentata graficamente con un'**ellisse** contenente il nome dell'attributo, collegata all'entità cui si riferisce.

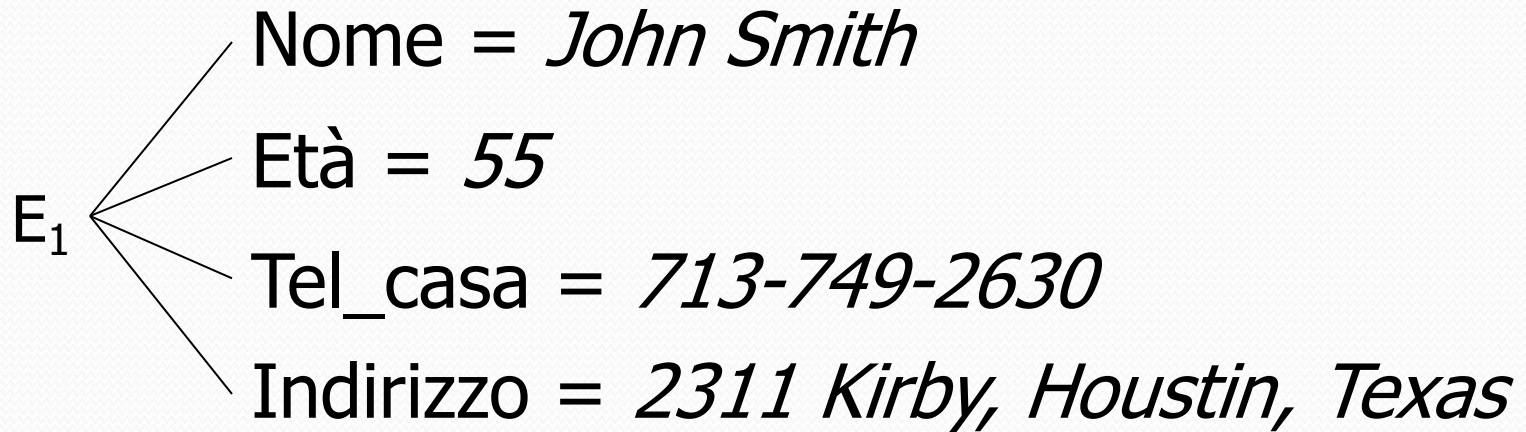


Rappresentazione ER  
dell'entità "*Studente*" con gli  
attributi "*Nome*" e "*Matricola*"



# Entità e Attributi: *Esempio*

- ***Esempio:*** entità *Impiegato*





# Tipo di Entità

- Entità con gli stessi attributi di base sono raggruppati in un **tipo di entità**.
- **Esempio:** Tutte le persone che lavorano per un dipartimento possono essere definite con l'entità *Impiegato*.

Nome del Tipo di Entità:  
(Schema o Intensione)

**Impiegato**

Nome, Età, Stipendio

Insieme di Entità  
(Estensione)

$e_1$  ■  
(John Smith, 55, 80k)

$e_2$  ■  
(Fred Brown, 40, 30k)

$e_3$  ■  
(Judy Clark, 25, 20k)

⋮

Un tipo di entità  
descrive lo **schema**  
(o **intenzione**) per  
un insieme di entità

Le entità individuali  
di un particolare tipo  
di entità sono  
raggruppate in una  
collezione o insieme  
detta **estensione** del  
tipo di entità.

# Tipi di Attributi

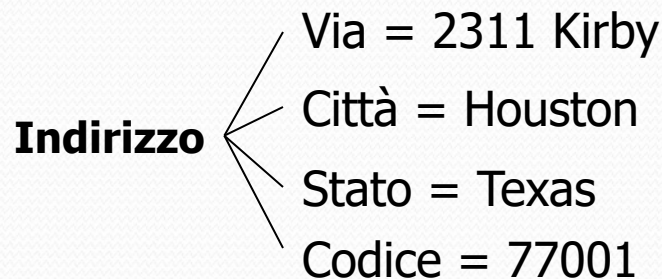
- Nel modello E-R abbiamo diversi tipi di attributi:

Divisibile?	Più valori?	Calcolabile?
Semplice	Single-valued	Memorizzato
Composto	Multivalued	Derivato



# Attributi Semplici e Composti

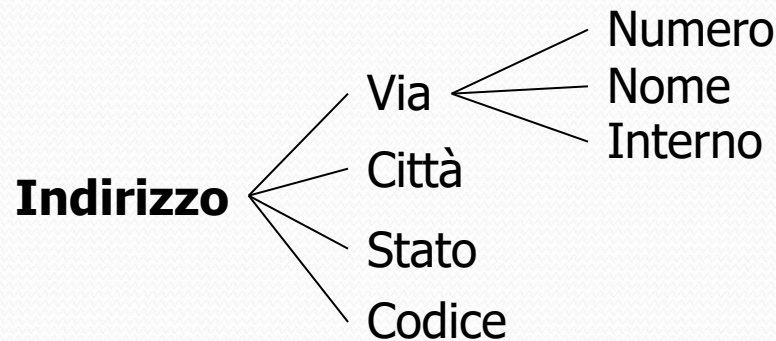
- *Attributi semplici*: ogni entità ha un valore singolo (atomico) per tale attributo.
- *Attributi composti*: possono essere divisi in sottoparti, che rappresentano informazioni di base con loro significati indipendenti.
  - *Esempio*: l'attributo **Indirizzo**:





# Attributi Semplici e Composti (2)

- Gli attributi composti possono formare una gerarchia:



- L'utilizzo di un attributo semplice o di uno composto dipende dalla necessità o meno di trattare separatamente le sottoparti.

# Single-Valued e Multivalued

- *Attributi single-valued:*
  - hanno un solo valore per ciascuna entità.
  - **Esempio:** l'età di un impiegato.
- *Attributi multivalued:*
  - può avere un insieme di valori per la stessa entità.
  - **Esempio:** l'attributo **Colore** per un'entità auto. Un'auto può avere più colori.
- Può essere determinato un limite inferiore e un limite superiore al numero di valori per un'entità.



# Memorizzati e derivati

- *Attributi derivati*: alcuni attributi possono essere in relazione tra loro.
  - **Esempio**: l'età e la data di nascita:
    - Età                      **attributo derivato**
    - Data\_nascita        **attributo memorizzato**
  - Alcuni valori di attributi possono essere derivati da entità correlate.  
**Esempio**:  
Numero\_di\_Impiegati di una entità “*dipartimento*” può essere derivato contando il numero di impiegati relati al (che lavorano per) dipartimento.
- Alcuni attributi possono avere valore **null** col significato di “non noto” o “mancante” o “non applicabile”.

# Attributi chiave di un tipo di entità

- Un importante vincolo sulle entità di un tipo di entità è la chiave o vincolo di unicità.
  - L'attributo chiave di un tipo di entità è un attributo che deve avere un valore univoco per cui ogni entità.
  - ***Esempio:*** Il codice fiscale di una persona.
- Talvolta più attributi insieme formano una chiave:
  - In tal caso tali attributi possono essere raggruppati in un attributo composto che diventa chiave.
- Alcuni tipi di entità possono avere più di un attributo chiave.
- In notazione ER l'attributo chiave è rappresentato **sottolineato** nell'ovale.



# Esempio: Entità AUTO

AUTO

Targa(Numero,Provincia), Telaio, Marca, Modello, Anno\_imm, {Colore}

Car 1 .

((ASC 123, TEXAS), tk629, Ford Mustang, convertible, 1989, {red, black})

Car 2 .

((ABO 123, NEW YORK), WPS872, N~san sontra 2~cor, 1992, {blue})

Car3 .

((VYS 720, TEXAS), TD729, Chrysler LeBaron, 4dcor, 1993, {white, bue})

Gli attributi multivalued sono mostrati tra parentesi { }.

Le componenti di un attributo composto sono mostrate tra parentesi ().

# Dominio di un attributo

- Ciascun attributo semplice è associato a un **dominio** o **insieme di valori** che rappresenta l'insieme dei valori che l'attributo può assumere.
- **Esempio:** l'età dell'impiegato può variare nel dominio (16, 70).
- *Matematicamente:*
  - Un attributo semplice A, del tipo di entità E, avente dominio V, è una funzione:  
 $A: E \rightarrow P(V)$        $P(V)$  insieme potenza dei sottoinsiemi di V:  
Un valore **null** è rappresentato con  $\emptyset$ .
  - $A(e)$  il valore dell'attributo A per l'entità e,  
 $A(e)$  = singleton per attributi single-valued.



## Dominio di un attributo (2)

- Per un attributo composto A, del tipo di entità E, il dominio V è il prodotto cartesiano

$$V = P(V_1) \times P(V_2) \times \dots \times P(V_n)$$

dove  $V_i$  è il dominio dell'attributo semplice  $i$ -mo di A,

- **Esempio:** l'indirizzo telefonico di una persona con più residenze, dove ogni residenza può avere più telefoni, è specificato come segue:

{ IndTelefono ({Telefono (Prefisso, Numero)}, Indirizzo  
(Via (Numero, Nome, Interno), Città, Stato, Codice) }

*Esempio:* il database Company



# Requisiti per il database Company

- La compagnia è organizzata in DIPARTIMENTI. Ogni Dipartimento ha un nome, un numero ed un impiegato che lo gestisce. Bisogna tener traccia della data di insediamento del manager. Un dipartimento può avere più locazioni.
- Ogni dipartimento controlla una serie di PROGETTI. Ogni progetto ha un nome, un numero ed una singola locazione.
- Per IMPIEGATO si tiene traccia di: nome, SSN, indirizzo, salario, sesso e data di nascita. Ogni impiegato lavora per un dipartimento e può lavorare su più progetti. Teniamo traccia del numero di ore settimanali che un impiegato spende su un progetto e del supervisore di ogni impiegato.
- Ogni impiegato ha una serie di PERSONE A CARICO. Per ogni persona a carico, registriamo: nome, sesso, data di nascita e parentela con l'impiegato.

# Disegno concettuale del database Company

- Descriviamo i tipi di entità per il database COMPANY.
- In accordo ai requirement possiamo identificare quattro tipi di entità:

## 1. DIPARTIMENTO

Nome, Numero, {Sedi}, Manager, Datains\_manager

Nome e Numero sono entrambi attributi chiave.

## 2. PROGETTO

Nome, Numero, Luogo, Dip\_controllo

Nome e Numero sono entrambi attributi chiave.



# Disegno concettuale del database Company (2)

## 3. IMPIEGATO

Nome, SSN, Sesso, Indirizzo, Stipendio, DataNascita, Dipartimento, Supervisore

SSN è un attributo chiave.

Nome e Indirizzo sono attributi composti (occorrerebbe verificare con l'utente se ha bisogno di riferire ai componenti individuali).

## 4. PERS\_A\_CARICO

Sesso, DataNascita, Impiegato, Nome\_pers\_carico, Parentela

## Disegno concettuale del database Company (3)

- Dobbiamo però rappresentare:
  - Il fatto che un impiegato può lavorare su più progetti.
  - Il numero di ore settimanali di un impiegato su ciascun progetto.
- Si può aggiungere un attributo a IMPIEGATO “*Lavora\_su*” composto di due componenti semplici (**Progetto**, **Ore**):

### IMPIEGATO

Nome (FName, Minit, LName), SSN, Sesso, Indirizzo, Stipendio, DataNascita, Dipartimento, Supervisore, {Lavora\_su(Progetto, Ore)}

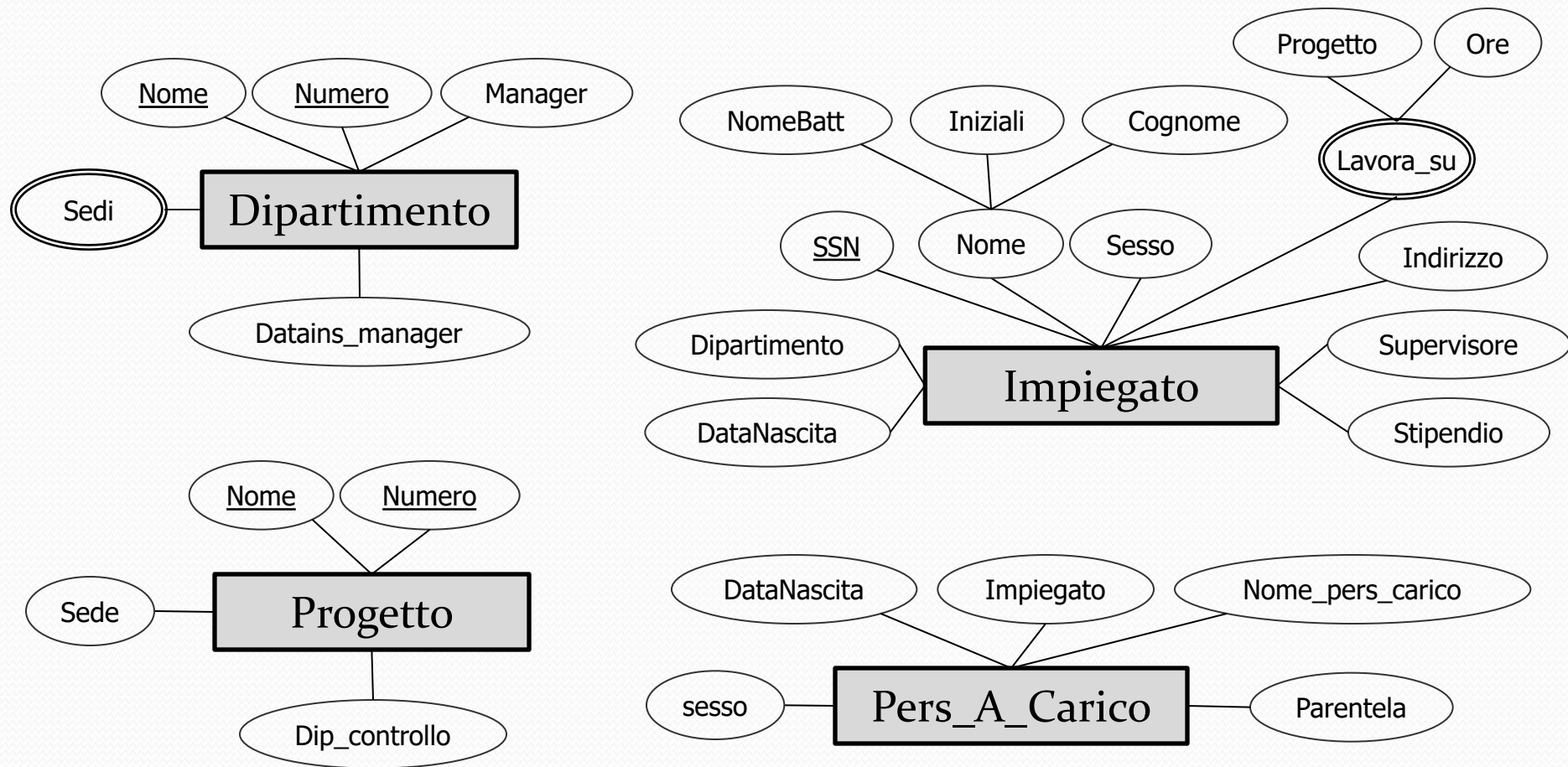
- In alternativa, le stesse informazioni si potrebbero mantenere nel tipo di entità PROGETTO con un attributo composto:
  - **Addetti** (**Impiegato**, **Ore**)



## Disegno concettuale del database Company (4)

- Esistono varie relazioni implicite:
  - L'attributo Manager di DIPARTIMENTO si riferisce a un impiegato che gestisce il dipartimento;
  - L'attributo Dip\_controllo di PROGETTO si riferisce al dipartimento che controlla il progetto;
  - L'attributo Dipartimento di IMPIEGATO si riferisce al dipartimento per cui lavora l'impiegato;
  - ...
- Nelle disegno iniziale queste associazioni tra entità sono rappresentabili come attributi, ma durante il processo di raffinamento nel modello ER questi riferimenti dovrebbero essere rappresentati come **relazioni**.

# Progettazione preliminare per il db COMPANY





# Relazioni, ruoli e vincoli strutturali



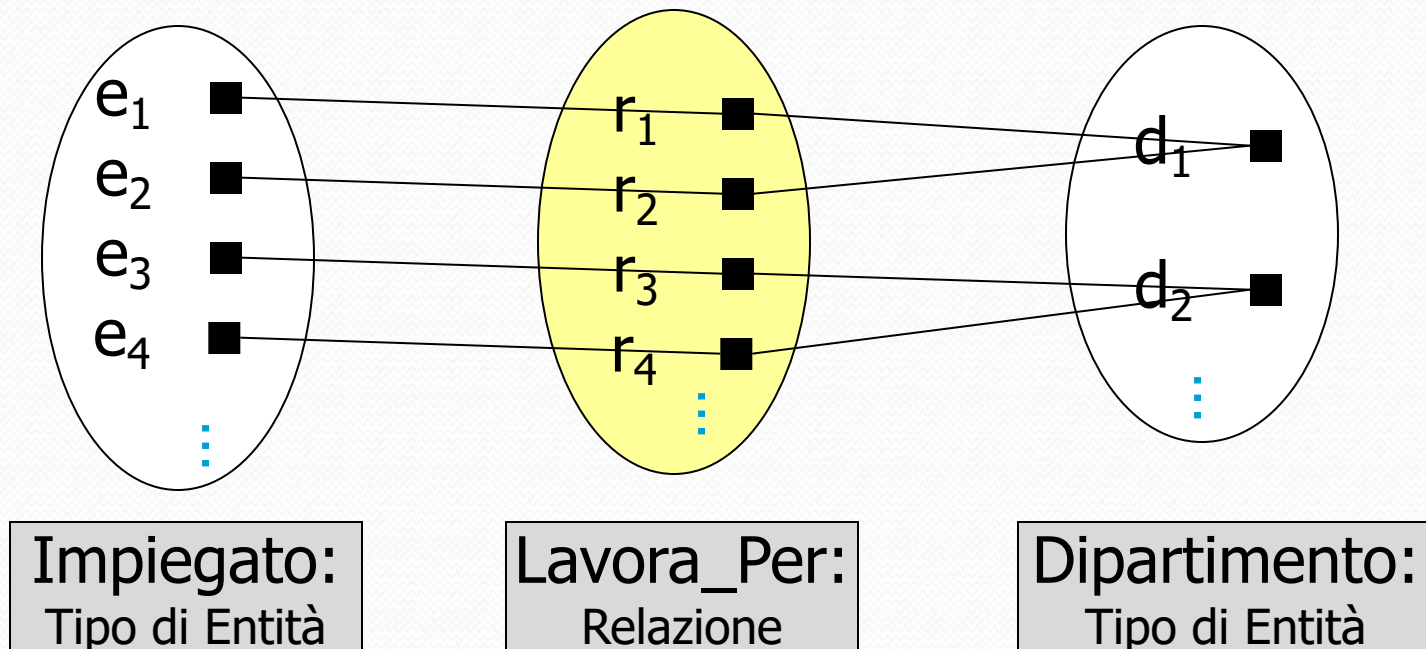
# Tipi e istanze di relazioni

- Le relazioni corrispondono a **legami logici** tra entità, significativi ai fini dell'applicazione di interesse.
- Un **tipo di relazione** è un'associazione tra  $n$  tipi di entità  $E_1, E_2, \dots, E_n$ .
- Le occorrenze o **istanze di relazione** associano  $n$  entità dei tipi di relazione richiesti.
- Ogni tipo di entità è detto **partecipare** al tipo di relazione.
- Il **grado** di un tipo di relazione è il numero di entità che vi partecipano. Se il grado è 2, la relazione è detta **binaria**.



# Tipi e istanze di relazioni (2)

- **Esempio:** vogliamo rappresentare il fatto che ogni impiegato  $e_i$  lavora per un dipartimento  $d_j$ .
- Definiamo il tipo di relazione LAVORA\_PER tra i due tipi di entità IMPIEGATO e DIPARTIMENTO: ogni relazione  $r_i$  associa una entità IMPIEGATO  $e_i$  con una entità DIPARTIMENTO  $d_j$ .



# Tipi e istanze di relazioni (3)

*Matematicamente:*

- R è un insieme di **istanze di relazione**  $r_i$  dove ogni  $r_i$  associa  $n$  entità  $(e_1, e_2, \dots, e_n)$ , e ciascuna entità  $e_j$  in  $r_i$  è un membro del tipo di entità  $E_j$ , con  $1 \leq j \leq n$ .

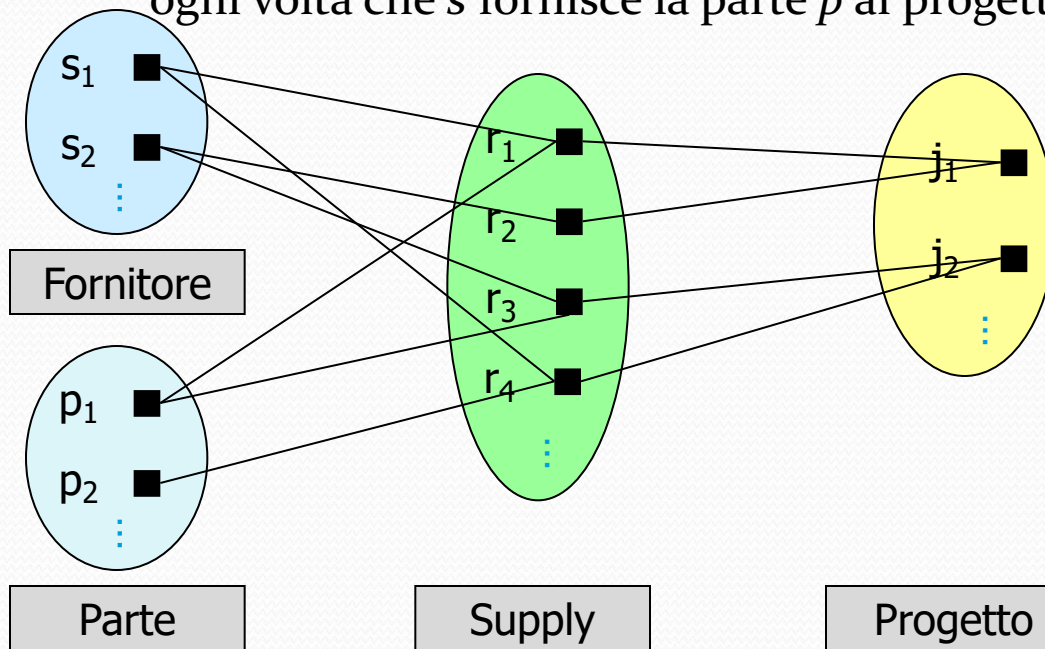
Quindi un tipo di relazione è una funzione matematica su  $E_1, E_2, \dots, E_n$  o alternativamente può essere definita come un **sottoinsieme del prodotto cartesiano**  $E_1 \times E_2 \times \dots \times E_n$ .

- Ciascun  $E_j$  è detto **partecipare** al tipo di relazione R e analogamente ogni entità individuale  $e_j$  è detta partecipare all'istanza di relazione  $r_i = (e_1, e_2, \dots, e_n)$ .



# Grado di un tipo di relazione

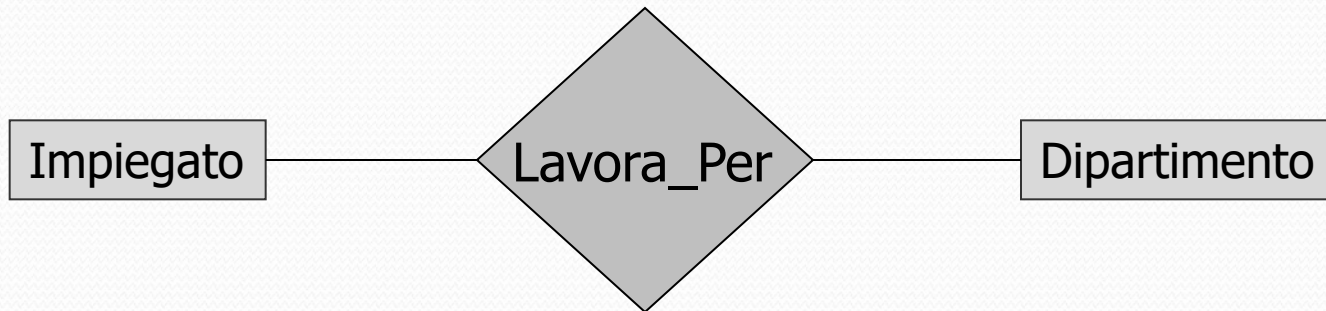
- Il grado di un tipo di relazione è il numero di tipi di entità partecipanti.
- **Esempio:**
  - LAVORA\_PER è di grado 2 (binaria).
  - La relazione SUPPLY è un tipo di relazione ternaria, dove ogni istanza di relazione  $r_i$  associa tre entità, un fornitore  $s$ , una parte  $p$  e un progetto  $j$  ogni volta che  $s$  fornisce la parte  $p$  al progetto  $j$ .



Le relazioni possono essere di qualsiasi grado ma le più ricorrenti sono quelle binarie.

# Rappresentazione di Relazioni

- In uno schema ER, un tipo di relazione ha un nome che lo identifica univocamente e viene rappresentato graficamente con un **rombo**, contenente il nome della relazione, e da linee che lo collegano ai tipi di entità che mette in relazione.



Rappresentazione ER della relazione  
*"Impiegato lavora per Dipartimento"*



# Relazioni come attributi

- A volte (soprattutto in fase di disegno iniziale) può essere conveniente considerare un tipo di relazione come un attributo di una delle entità partecipanti, per semplificare la definizione dello schema.
- Le relazioni verranno poi esplicitate durante il raffinamento del progetto.
- Quando si considera una relazione binaria come attributo esistono ovviamente due alternative, in base a quale entità viene scelta per contenere l'attributo.

# Relazioni come attributi: *Esempio*

*Il tipo di relazione LAVORA\_PER può essere rappresentato:*

- Tramite un attributo Dipartimento nel tipo di entità IMPIEGATO. Per ogni entità impiegato si riferisce all'entità dipartimento in cui lavora. Il dominio dell'attributo Dipartimento è l'insieme di tutte le entità DIPARTIMENTO,

*oppure*

- tramite un attributo multivalued Impiegati del tipo di entità DIPARTIMENTO. Per ogni entità dipartimento il valore dell'attributo Impiegati è l'insieme degli impiegati che lavorano in quel dipartimento. Il dominio dell'attributo Impiegati è l'insieme delle entità IMPIEGATO.

- Se entrambi gli attributi sono utilizzati per rappresentare la relazione LAVORA\_PER, allora essi sono vincolati ad essere l'uno l'inverso dell'altro.



# Nomi di Ruolo

- Ogni entità che partecipa a qualche tipo di relazione riveste un **ruolo** particolare nella relazione.
  - Il **nome di ruolo** specifica il ruolo che riveste in ciascuna istanza di relazione una entità partecipante.
- **Esempio:** nel tipo di relazione LAVORA\_PER:
  - IMPIEGATO gioca il ruolo di *impiegato* o di *addetto*.
  - DIPARTIMENTO gioca il ruolo di *dipartimento* o di *datore di lavoro*.

# Relazioni Ricorsive

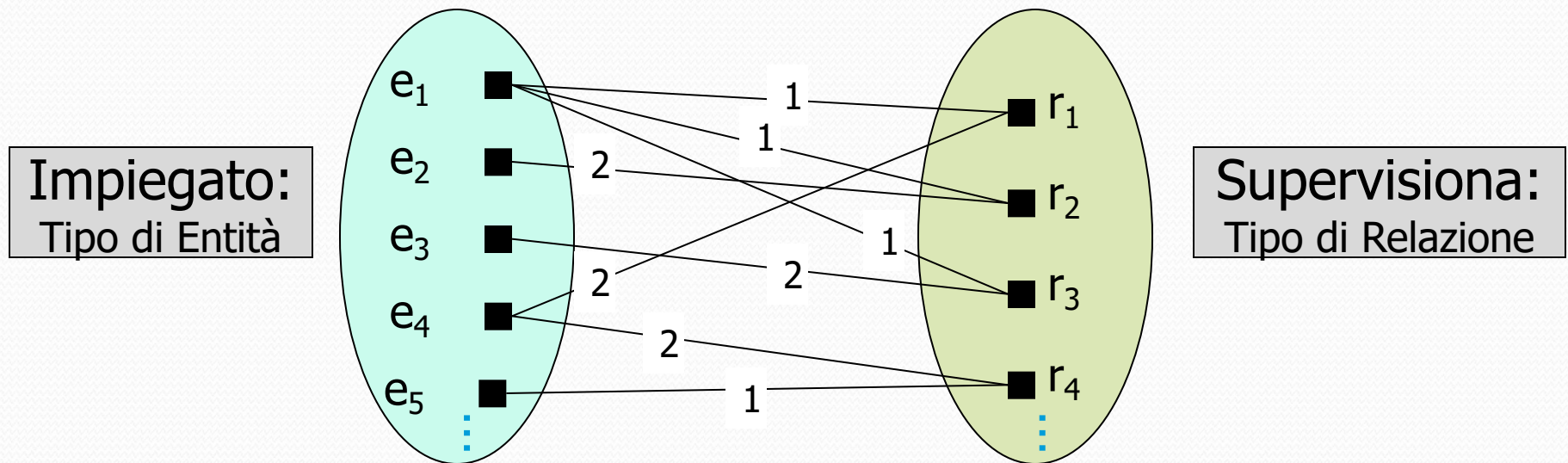
- Quando lo stesso tipo di entità partecipa ad un tipo di relazione più di una volta con ruoli diversi, il nome del ruolo diventa essenziale per distinguere il significato di ciascuna partecipazione.
- Questi tipi di relazioni sono detti **ricorsivi**.



# Relazioni Ricorsive: *Esempio*

**Esempio:** il tipo di relazione SUPERVISIONE IMPIEGATO : IMPIEGATO

- mette in relazione un supervisore e un subordinato, dove entrambe le entità supervisore e subordinato sono membri dello stesso tipo di entità IMPIEGATO. Quindi l'entità IMPIEGATO partecipa due volte al tipo di relazione SUPERVISIONE, una volta col ruolo di supervisore e una col ruolo di subordinato.



Le linee '1' rappresentano il ruolo di supervisore, quelle '2' il ruolo di subordinato.

# Vincoli sui tipi di relazioni

- Ogni tipo di relazione ha un insieme di vincoli che limitano le combinazioni possibili di entità che possono partecipare ad istanze della relazione.
- Questi vincoli sono determinati dalla situazione del miniworld che le relazioni rappresentano.
- **Esempio:** se la società ha la regola che ogni impiegato deve lavorare esattamente per un dipartimento, si vuole descrivere questo vincolo nello schema.

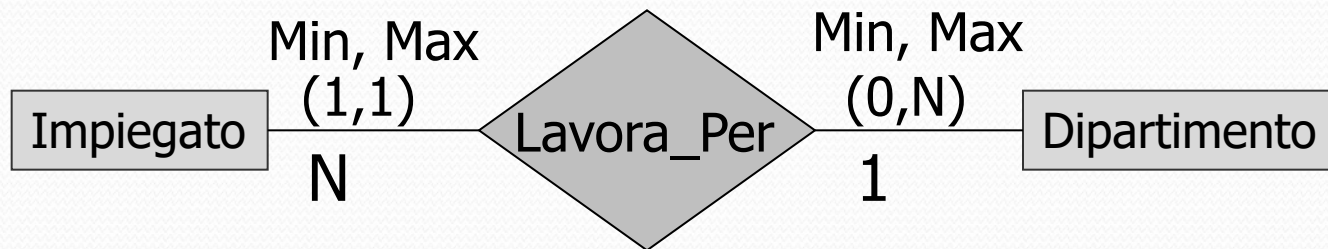


# Tipi di Vincoli

- Si distinguono due tipi di vincoli:
  1. **Rapporto di cardinalità**
    - specifica il numero di istanze di relazione in cui un'entità può partecipare.
  2. **Vincolo di partecipazione**
    - specifica se l'esistenza di una entità dipende dal fatto di essere relata a un'altra entità attraverso il tipo di relazione.
    - Due tipi di vincoli di partecipazione:
      - partecipazione totale o dipendenza di esistenza,
      - partecipazione parziale.
- I vincoli rapporto di cardinalità e vincolo di partecipazione sono detti **vincoli strutturali** di un tipo di relazione.

# Rapporto di cardinalità

- Deve essere indicato per ciascun tipo di entità che partecipa ad una relazione, e permette di specificare il **numero minimo** e **massimo** di istanze di relazione a cui le occorrenze delle entità coinvolte possono partecipare.



- Rappresentazione ER della relazione “*Impiegato lavora per Dipartimento*”, con rapporto di cardinalità N:1 – N impiegati lavorano per un dipartimento:
  - MAX:** Ogni dipartimento può avere numerosi impiegati, e ciascun impiegato lavora per un solo dipartimento.
  - MIN:** Un dipartimento potrebbe non avere impiegati, mentre un impiegato deve sempre essere assegnato ad un dipartimento.



## *Esempio:* relazione lavora\_per

*Esempio:* Impiegato dipartimento

(rossi, ricerca)

(bianchi, ricerca)

(neri, amministrazione)

(verdi, ricerca)

- Rossi è presente una volta nella relazione.
- Ricerca è presente 3 volte.

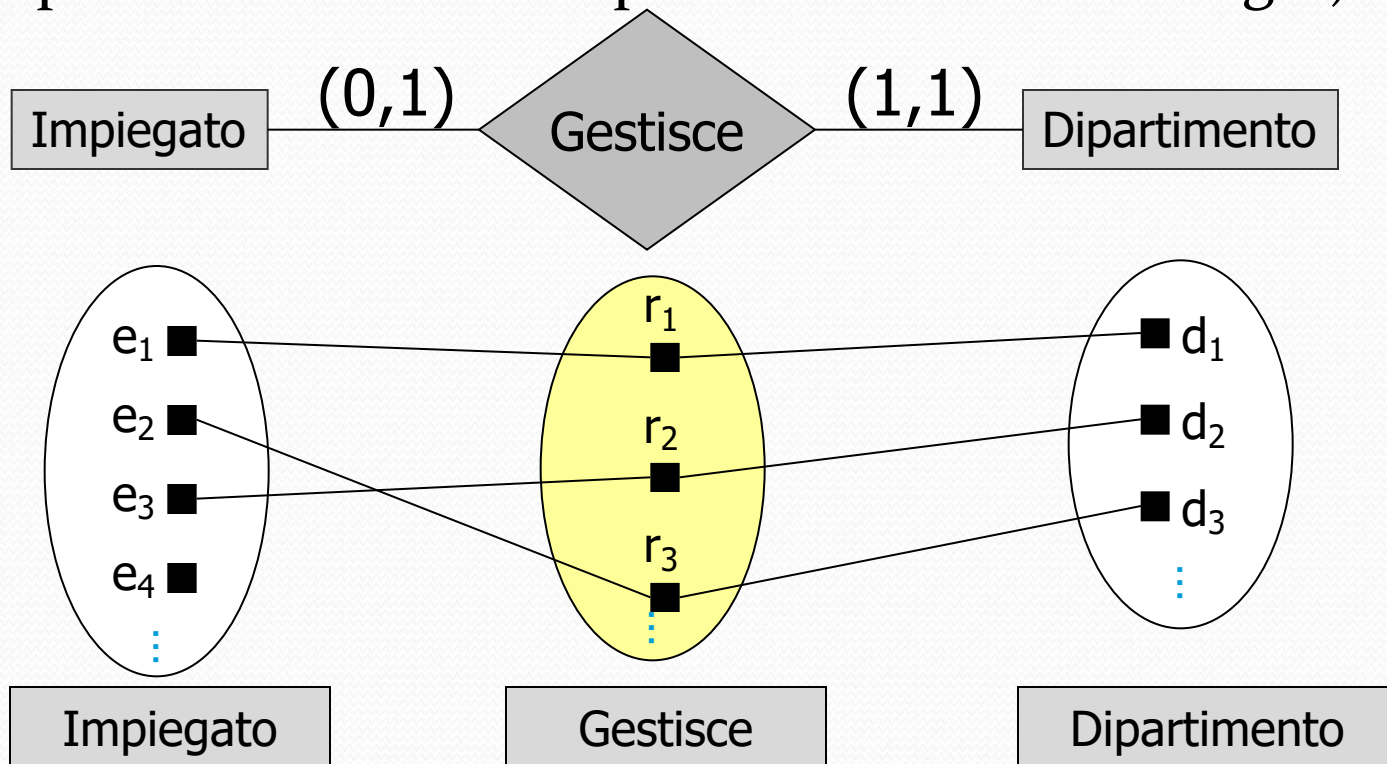
## Rapporto di cardinalità (2)

- È possibile assegnare un qualunque intero non negativo a un rapporto di cardinalità, con l'ovvio vincolo che la cardinalità minima deve essere minore o uguale alla cardinalità massima.
- Nella maggior parte dei casi si utilizzano solo tre valori: 0, 1 e N:
  - Il valore 0 per la cardinalità minima indica una partecipazione **opzionale** del tipo di entità alla relazione.
  - Il valore 1 per la cardinalità minima indica una partecipazione **obbligatoria** del tipo di entità alla relazione.
- La cardinalità minima può eventualmente essere omessa, quella massima deve essere sempre specificata.



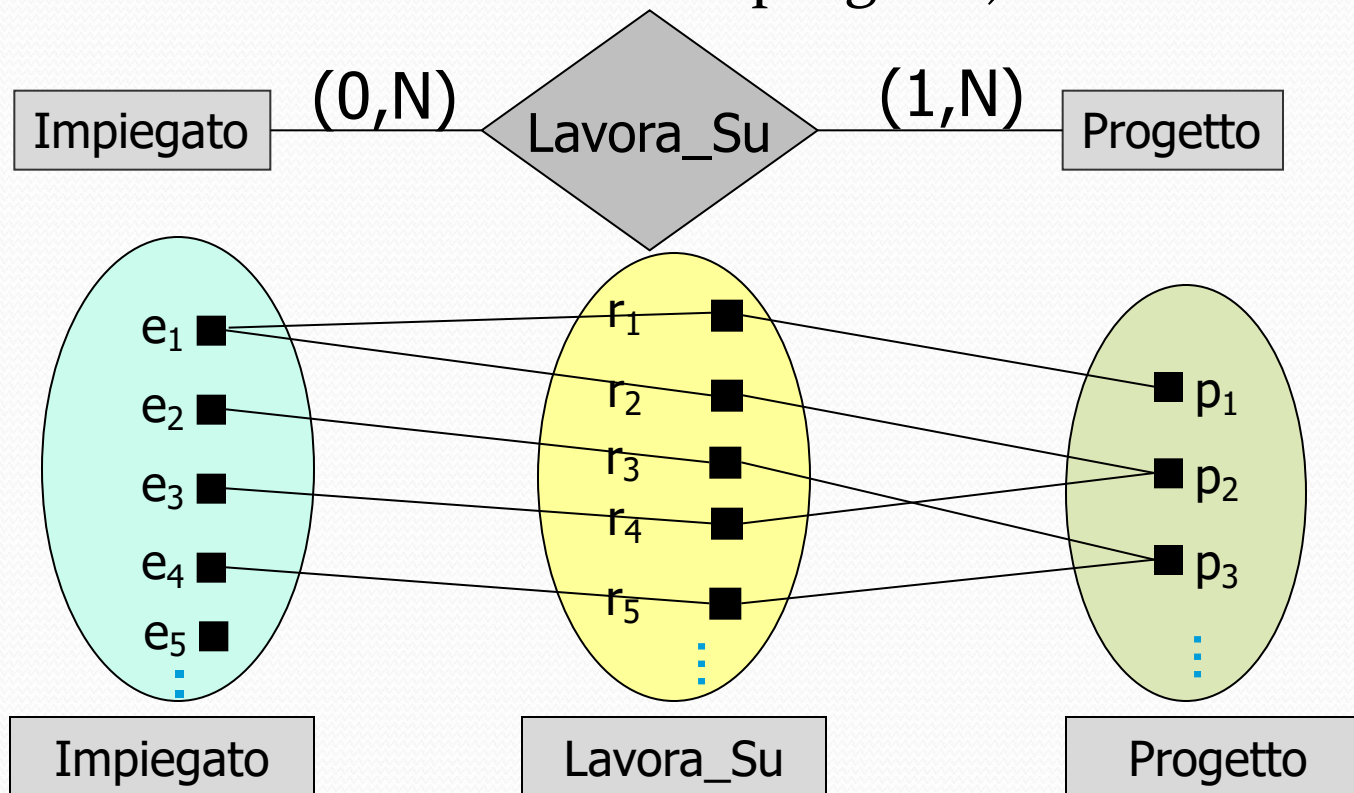
# Rapporto di cardinalità: *Esempio*

- La relazione binaria GESTISCE tra IMPIEGATO e DIPARTIMENTO è di rapporto di cardinalità 1:1 (un impiegato può gestire al più un dipartimento, ed un dipartimento deve sempre avere un solo manager).



# Rapporto di cardinalità: *Esempio*

- La relazione binaria LAVORA\_SU tra IMPIEGATO e PROGETTO è di rapporto di cardinalità M:N, (poiché un impiegato può lavorare su più progetti, e più impiegati possono lavorare sullo stesso progetto).





# Vincoli di partecipazione

- Specifica se l'esistenza di una entità dipende dal fatto di essere relata a un'altra entità attraverso il tipo di relazione.
- Due tipi di vincoli di partecipazione:
  - partecipazione totale o dipendenza di esistenza,
  - partecipazione parziale.

# Vincoli di partecipazione: *Esempio*

- ***Esempio:*** Ipotizziamo che ogni impiegato deve lavorare per un dipartimento. Allora esiste una entità impiegato solo se partecipa a un'istanza di relazione LAVORA\_PER.
- Partecipazione totale significa che ogni entità nell'insieme **totale** delle entità impiegato deve essere relata a una entità dipartimento:
  - Nella relazione GESTISCE, poiché non tutti gli impiegati gestiscono un dipartimento, il vincolo è parziale in quanto solo un sottoinsieme delle entità impiegato è relato ad entità dipartimento tramite GESTISCE.
- **Notazione:** Nei diagrammi la partecipazione totale è rappresentata con una linea doppia, quella parziale con una linea singola.



# Attributi di Tipi di Relazioni

- I tipi di relazioni possono avere attributi simili a quelli dei tipi di entità.
- *Esempio:*
  - La relazione LAVORA\_SU può includere un attributo **ore** per mantenere il numero di ore settimanali che l'impiegato dedica al progetto.
  - Analogamente per un attributo **data\_ins** di insediamento del manager nella reazione GESTISCE.

## Attributi di Tipi di Relazioni (2)

- Gli attributi dei tipi di relazioni 1:1 possono essere trasferiti a uno dei tipi di entità partecipanti.
  - *Es:* per il tipo di relazione GESTISCE, la data di insediamento può essere l'attributo di IMPIEGATO oppure di DIPARTIMENTO, perché si tratta di un tipo di relazione 1:1.
- Per un tipo di relazione 1:N, un attributo di relazione può essere trasferito solo del tipo di entità dalla parte N della relazione.
  - *Es:* nei tipo di relazione LAVORA\_PER un attributo **data\_inizio** per inizio del rapporto dell'impiegato con il dipartimento, può essere incluso come un attributo del tipo di entità IMPIEGATO.
- La scelta di considerare l'attributo nel tipo di entità o nel tipo di relazione è una scelta *soggettiva* del progettista di DB.



## Attributi di Tipi di Relazioni (3)

- Per tipi di relazioni M:N non è sempre possibile mantenere l'attributo in una delle due entità partecipanti in quanto tale attributo può essere determinato dalla combinazione delle entità (*ad esempio* l'attributo **ore** della relazione M:N LAVORA\_SU è determinato dalla combinazione impiegato-progetto).

# Tipi di entità deboli





# Tipi di Entità Deboli








- Sono tipi di entità che possono non avere attributi chiave propri.
- Sono relati a entità specifiche da un altro tipo di entità (**possessore di identificazione**) in combinazione con alcuni valori dei suoi attributi.
- La relazione viene detta **relazione di identificazione**.
- Un tipo di entità debole ha sempre un vincolo di partecipazione totale.
- **Esempio:**
  - PERS\_A\_CARICO e IMPIEGATO. Due persone a carico possono avere lo stesso **nome**, **DataNascita**, **sex** e **parentela** ma sono individuati dalle entità IMPIEGATO a cui sono relate.  
Ciascuna entità impiegato è detta **possedere** le entità dipendenti ad essa relate.

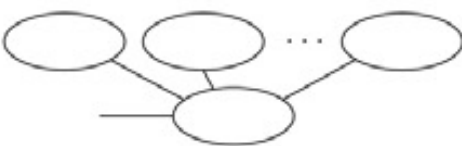

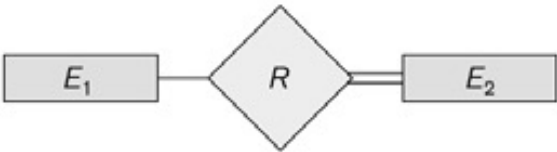

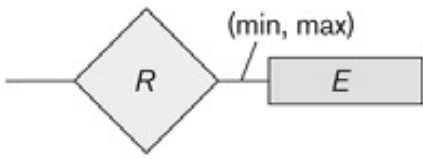
## Tipi di Entità Deboli (2)

- Un tipo di entità debole normalmente ha una **chiave parziale** che è l'insieme di attributi che identifica univocamente due entità relate alla stessa entità possessore  
(es. l'attributo del nome della persona a carico **Nome\_pers\_carico** è una chiave parziale).
- Tipi di entità deboli possono essere anche rappresentati come attributi composti multivalued.
- La scelta della rappresentazione è lasciata al progettista del DB.
  - Un approccio è scegliere la rappresentazione di tipo di entità debole se ha molti attributi e partecipa indipendentemente ad altri tipi di relazioni oltre quella di identificazione.
  - In generale può essere definito qualsiasi numero di livelli di tipi di entità deboli.




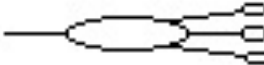






# Notazione per i diagrammi ER (Navathe)

Symbol	Meaning
	Entity
	Weak Entity
	Relationship
	Identifying Relationship
	Attribute
	Key Attribute
	Multivalued Attribute

Symbol	Meaning
	Composite Attribute
	Derived Attribute
	Total Participation of $E_2$ in $R$
	Cardinality Ratio 1 : N for $E_1:E_2$ in $R$
	Structural Constraint (min, max) on Participation of $E$ in $R$

# Notazione per i diagrammi ER (Atzeni)

Entità	
Relazione	
Attributo	
Attributo composto	
Cardinalità relazione	
Cardinalità attributo	
Identificatore interno	
Identificatore esterno	



# Raffinamento del modello ER



# Raffinamento del modello ER

- Si procede a cambiare gli attributi che rappresentano relazioni in tipi di relazioni.
- I rapporti di cardinalità e il vincolo di partecipazione di ciascun tipo di relazione si determinano a partire dai requisiti;

*Se necessario si consulta l'utente.*

- Nell'esempio specifichiamo i seguenti tipi di relazioni:

1. GESTISCE, tipo di relazione 1:1 tra IMPIEGATO e DIPARTIMENTO.

**Partecipazione:**

IMPIEGATO: parziale

DIPARTIMENTO: (richiesto a utente) totale

**Attributo:** data\_Ins



# Raffinamento del modello ER (2)

- 2. LAVORA\_PER, tipo di relazione 1:N tra DIPARTIMENTO e IMPIEGATO

**Partecipazione:**

DIPARTIMENTO : totale

IMPIEGATO : totale

- 3. CONTROLLA, tipo di relazione 1:N tra DIPARTIMENTO e PROGETTO

**Partecipazione:**

DIPARTIMENTO : totale

PROGETTO : (richiesto a utente) parziale

# Raffinamento del modello ER (3)

- 4. SUPERVISIONE, tipo di relazione 1:N tra IMPIEGATO (ruolo supervisore) e IMPIEGATO (ruolo subordinato)

**Partecipazione:**

IMPIEGATO: (richiesto a utente) parziale

IMPIEGATO: (richiesto a utente) parziale

(non tutti gli impiegati sono supervisori e non tutti gli impiegati hanno un supervisore)

- 5. LAVORA\_SU, tipo di relazione M:N tra IMPIEGATO e PROGETTO.

**Partecipazione:**

IMPIEGATO totale

PROGETTO totale

**Attributo:** Ore



# Raffinamento del modello ER (4)

6. A\_CARICO, tipo di relazione 1:N tra IMPIEGATO e PERS\_CARICO.

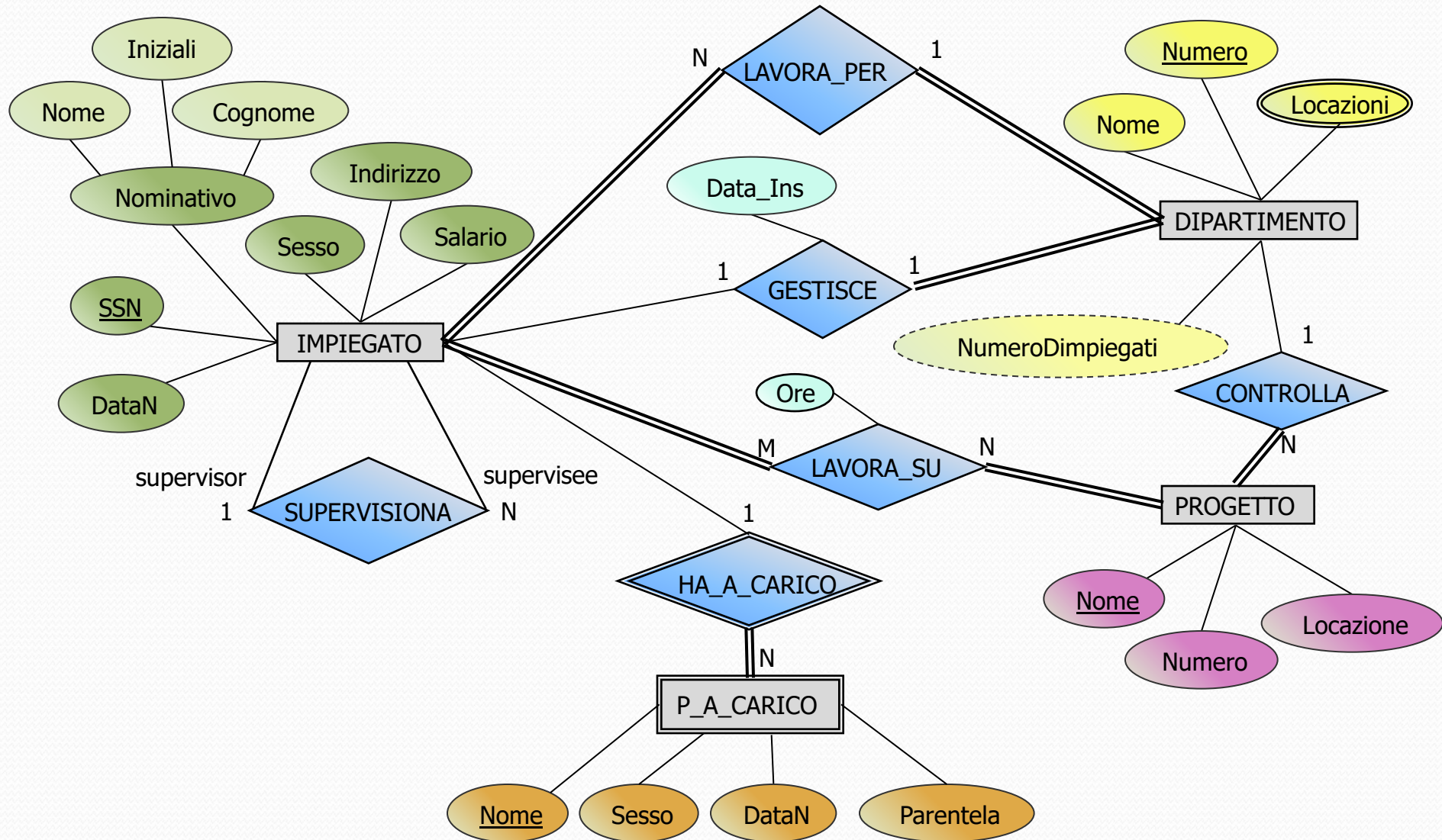
**Partecipazione:**

IMPIEGATO parziale

PERS\_A\_CARICO : totale

- Si eliminano quindi tutti gli attributi che sono stati convertiti in relazioni.  
È importante eliminare la ridondanza, che può essere eventualmente aggiunta in fasi successive.

# Modello ER risultante





# Convenzioni sulla notazione dei diagrammi ER

- **Nomi** per tipi di entità.
- **Verbi** per tipi di relazioni.
- Lettura dello schema *da sinistra a destra* e *dall'alto verso il basso*.

# Metodologia da seguire per il design di un modello concettuale

- Identificare i tipi di entità.
- Identificare i tipi di relazioni.
- Identificare ed associare gli attributi ai tipi di entità ed ai tipi di relazione.
- Determinare i domini degli attributi.
- Determinare le chiavi candidate e la chiave primaria.
- Controllare la ridondanza:
  - Riesaminare le relazioni 1-1.
  - Rimuovere le relazioni ridondanti.
- Validare il modello concettuale considerando le transazioni utente.



# Le Business Rule



# Business rule: Introduzione

- Business rule:
  - Strumento usato dagli analisti dei sistemi informativi per la descrizione di proprietà di una applicazione che non si riesce a rappresentare completamente nel modello E-R.
  - Costituisce una documentazione di supporto per facilitare l'interpretazione dello schema e descrivere le proprietà dei dati non esprimibili nel modello E-R.
- Possono essere di tre tipi:
  1. Descrizione di un concetto rilevante
  2. Vincolo di integrità sui dati
  3. Derivazione (espressione di un concetto di inferenza o calcolo)



# 1. Descrizione di un concetto rilevante

- È espresso mediante frasi in linguaggio naturale o sottoforma di glossari (dizionario dei dati).
- Il dizionario dei dati è composto da due tabelle:
  - **Tavola delle entità:** Nome, descrizione informale in linguaggio naturale, elenco degli attributi con eventuali descrizioni, possibili identificatori.
  - **Tavola delle relazioni:** Nome, descrizione informale in linguaggio naturale, elenco degli attributi con eventuali descrizioni, elenco delle entità coinvolte con relative cardinalità di partecipazione.

# Esempio di dizionario dei dati

Entità	Descrizione	Attributi	Identificazione
Impiegato	Impiegato che lavora nella azienda.	Codice, Cognome, Stipendio, Età	Codice
Progetto	Progetti aziendali sui quali lavorano gli impiegato.	Nome, Budget, Data consegna	Nome
Dipartimento	Dipartimento delle sedi dell'azienda	Telefono, Nome	Nome, Sede
Sede	Sede dell'azienda in una certa città	Città, Indirizzo (Numero, Via, CAP)	Città

Relazione	Descrizione	Entità coinvolte	Attributi
Direzione	Associa un dipartimento al suo direttore	Impiegato (0, 1), Dipartimento (1, 1)	
Afferenza	Associa un impiegato al suo dipartimento	Impiegato (0, 1), Dipartimento (1, N)	Data afferenza
Partecipazione	Associa agli impiegati i progetti sui quali lavorano	Impiegato (0, N), Progetto (1, N)	Data inizio
Composizione	Associa una sede ai dipartimenti di cui è composta	Dipartimento (1, 1), Sede (1, N)	



## 2. Vincolo di integrità sui dati

- Sono espresse sottoforma di *asserzioni atomiche* (non decomponibili a loro volta in asserzioni) che devono essere sempre verificate nella base di dati.
- Sono enunciate in maniera *dichiarativa* del tipo:
  - *<concetto> **deve/non deve** <espressione su concetti>*
- I concetti coinvolti devono apparire o essere derivabili da concetti nello schema E-R.

# Esempi di regole di vincolo

- (RV<sub>1</sub>) *il direttore di un dipartimento **deve** afferire a tale dipartimento.*
- (RV<sub>2</sub>) *un impiegato **non deve** avere uno stipendio maggiore del direttore del dipartimento al quale afferisce.*
- (RV<sub>3</sub>) *un dipartimento con sede a Roma **deve** essere diretto da un impiegato con più di dieci anni di anzianità.*



### 3. Derivazione

- Regole aziendali espresse specificando operazioni che permettono di ottenere il concetto derivato.
- Sono enunciate con espressioni del tipo:
  - *<concetto> si ottiene <operazioni su concetti>*

# Esempi di regole di derivazione

- (RD<sub>1</sub>) *il budget di un progetto **si ottiene** moltiplicando per 3 la somma degli stipendi degli impiegati che vi partecipano.*



# Business rule: Implementazione

- Lo schema concettuale viene tradotto in una base di dati (progettazione logica e fisica).
- Le business rule non descrittive (vincoli o derivazioni) vanno codificate mediante:
  - Clausole SQL all'atto della definizione dello schema;
  - Trigger;
  - Procedure scritte in qualche linguaggio di programmazione.