



La progettazione logica: il modello relazionale

1 Introduzione

- La **progettazione logica relazionale** consiste nella conversione del diagramma ER in un insieme di tabelle detto **schema logico relazionale** e nella definizione delle operazioni da compiere su di esso.

Schema concettuale: Diagramma ER



Progettazione Logica



Schema logico: Insieme di Tabelle



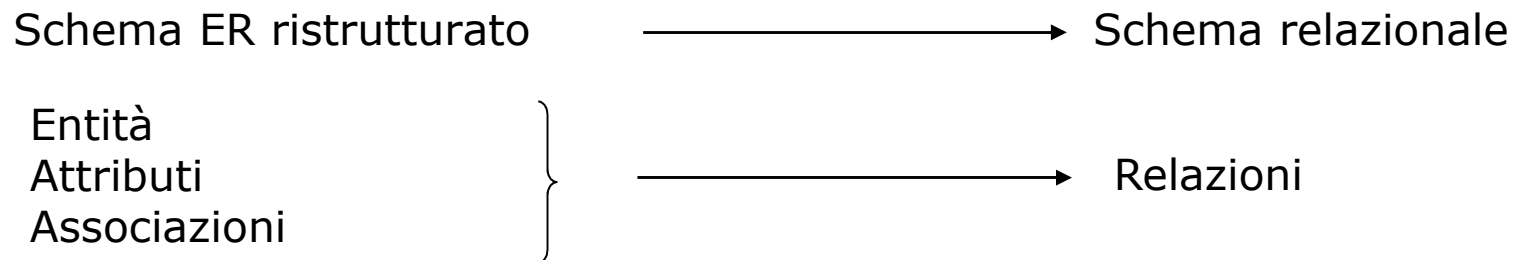
1 Introduzione

- La progettazione logica è costituita fundamentalmente da quattro fasi principali:
 - fase di **Ristrutturazione**
 - fase di **Traduzione**
 - Definizione delle **Operazioni sulle Relazioni**
 - Verifiche di **Normalizzazione**
- 1. Ristrutturazione:** eliminazione dallo schema ER di tutti i costrutti che non possono essere direttamente rappresentati nel modello relazionale
 - eliminazione degli attributi composti e multipli
 - eliminazione delle gerarchie di generalizzazione
 - eliminazione degli identificatori esterni (non trattato)
- risultato: schema ER ristrutturato



1 Introduzione

2. Traduzione: traduzione con l'utilizzo di regole di trasformazione di entità, attributi e associazioni dello schema ER in relazioni del modello relazionale



○ Risultato: Schema Relazionale



1 Introduzione

- 3. Operazioni sulle Relazioni:** Definizione delle operazioni che consentono di **interrogare** una base di dati relazionale, consentendo di ottenere le informazioni desiderate estraendo dalle tabelle delle sottotabelle ridotte. Oppure **combinando opportunamente** una o più tabelle per generare nuove tabelle contenenti i dati desiderati.
- 4. Normalizzazione:** Procedimento che consente di verificare se lo schema relazionale corrisponde ai "canoni standard" di correttezza della base di dati avvalendosi di un preciso insieme di regole, riportare le tabelle in quelle che sono definite "forme normali" delle tabelle relazionali.

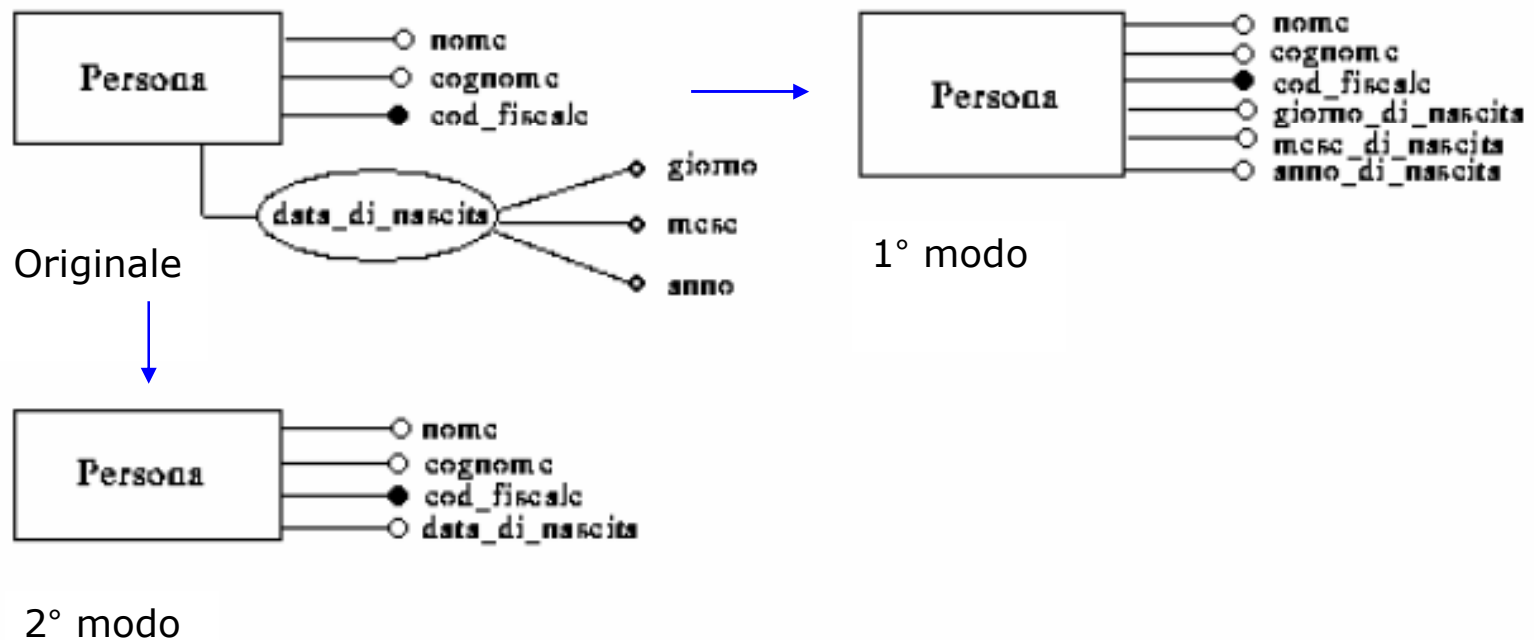


1.1 Ristrutturazione (non presente sul libro)

- Eliminazione degli **attributi composti** e degli **attributi multipli**:
- Il modello relazionale consente solo la specifica di attributi semplici e monovalore
 - per eliminare gli **attributi composti** si può procedere in due modi (ricorsivamente):
 1. Si considerano tutti i sottoattributi come attributi dell'entità
 2. Si eliminano i sottoattributi e si considera l'attributo composto come un attributo semplice (ridefinizione del dominio)

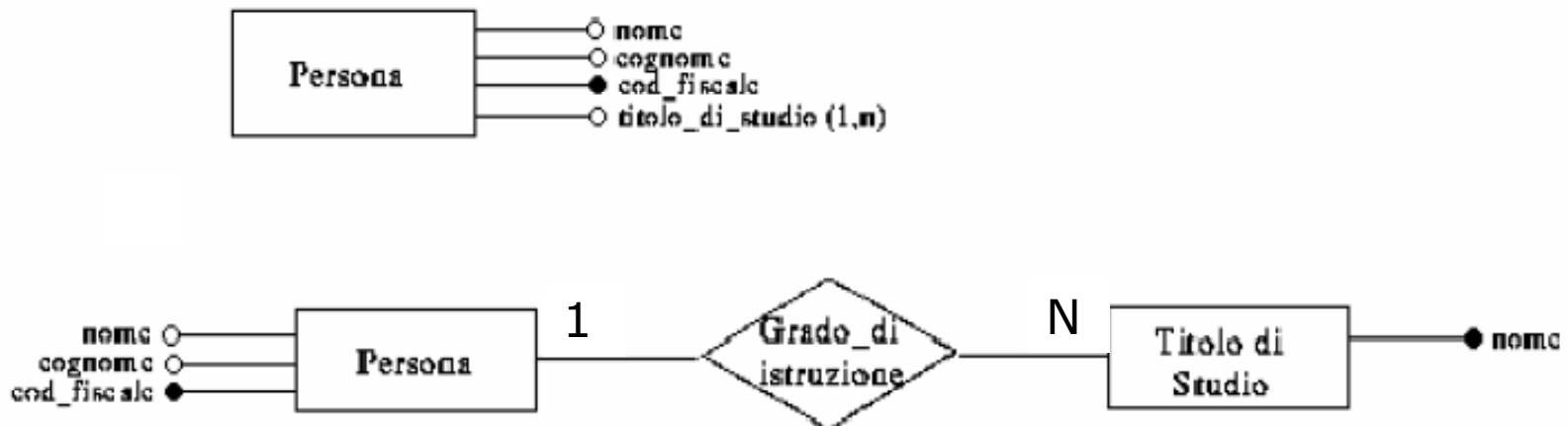
1.1 Ristrutturazione (non presente sul libro)

- Esempio di **eliminazione attributo composto**:



1.1 Ristrutturazione (non presente sul libro)

- per eliminare gli **attributi multipli** si definisce una nuova entità, collegata all'entità di partenza con un'associazione, che modella l'attributo multivalore mediante un attributo a valore singolo l'associazione introdotta sarà ovviamente uno a molti
- Esempio di **eliminazione attributo multiplo**:





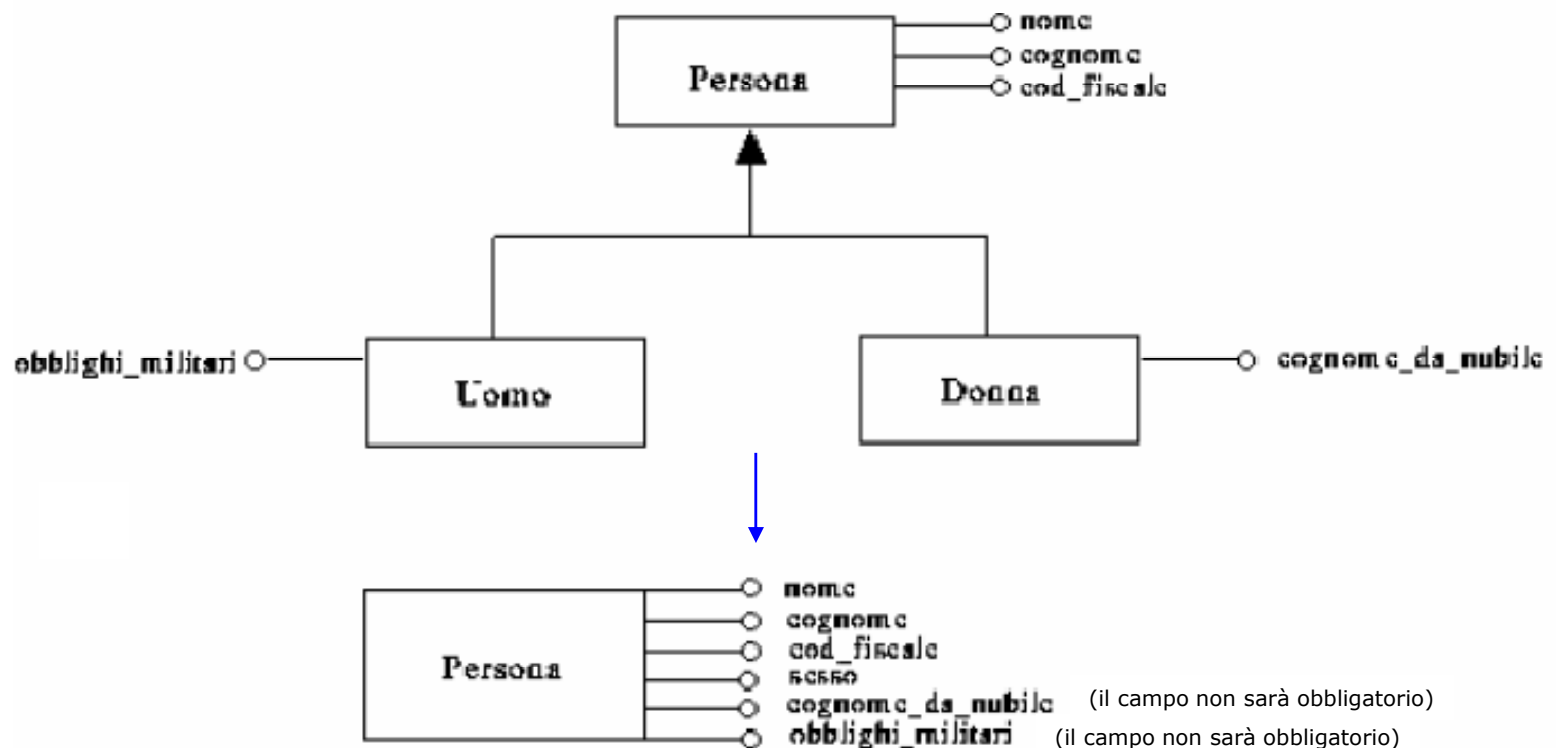
1.1 Ristrutturazione (non presente sul libro)

- Eliminazione delle **gerarchie di generalizzazione** :
- Il modello relazionale non prevede gerarchie di generalizzazione
 - Vi sono quindi tre alternative per risolvere il problema:
 1. eliminazione delle entità figlie
 2. eliminazione dell'entità padre
 3. sostituzione della generalizzazione con associazioni (non tratteremo)

1.1 Ristrutturazione (non presente sul libro)

1. eliminazione delle entità figlie

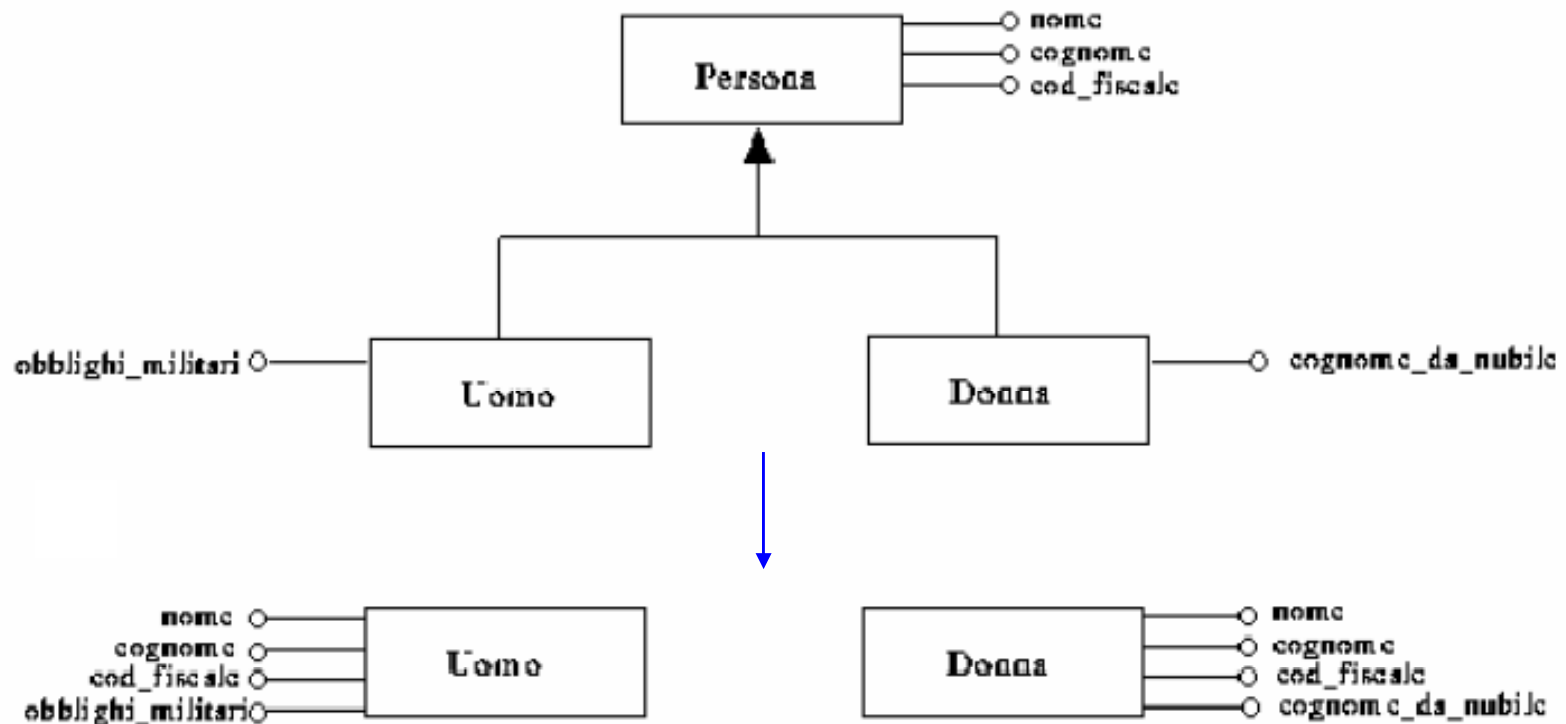
Esempio:



1.1 Ristrutturazione (non presente sul libro)

2. eliminazione dell'entità padre

Esempio:



1 Le relazioni

- Definizione: Una relazione "R" definita su di un insieme di domini D_1, D_2, \dots, D_n è un sottoinsieme del prodotto cartesiano $D_1 \times D_2 \times \dots \times D_n$ che può essere espresso con la scritta:

$$R \subseteq D_1 \times D_2 \times \dots \times D_n$$

Esempio:

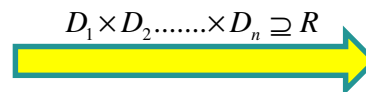
$$D_1 = \{a, b, c\}$$

$$D_2 = \{2, 7\}$$

$$D_3 = \{x, y\}$$

Prodotto Cartesiano

D1	D2	D3
a	2	x
a	2	y
a	7	x
a	7	y
b	2	x
b	2	y
b	7	x
b	7	y
c	2	x
c	2	y
c	7	x
c	7	y



Relazione

R		
D1	D2	D3
a	7	X
a	2	Y
b	7	X
c	7	Y
b	7	Y
c	2	X
c	2	y



1 Le relazioni

- Gli insiemi **D** sono detti domini della relazione (carattere, intero, reale, booleano, ecc..)
- Due domini possono quindi avere lo stesso tipo ma nomi diversi.
- **n** è detto grado della relazione rappresenta il numero di insiemi D che vengono considerati (numero di colonne della relazione che ne deriva)
- Viene definito schema di una relazione il nome di una relazione e la lista dei suoi attributi racchiusi tra parentesi tonde e separate da virgole.

`<NomeRelazione>(<Attributo1>[:<Tipo1>],...,<AttributoN>[:<TipoN>])`

- Per convenzione vengono indicati i nomi delle relazioni in maiuscolo e gli attributi con la sola iniziale maiuscola. I <Tipi> rappresentano i tipi di dato degli attributi, se non indispensabili, di solito si omettono

1 Le relazioni

- Esempio:

PERSONA (Cognome:String(30), Nome: String(20), Età:Intero, Sesso:Booleano)

Più in breve:

PERSONA (Cognome, Nome, Età, Sesso)

PERSONA			
Cognome	Nome	Età	Sesso

$$(d_1, d_2, \dots, d_n) \quad d_1 \in D_1, d_2 \in D_2, \dots, d_n \in D_n$$



1 Le relazioni

- Gli elementi di una relazione (insieme dei valori di una riga) sono detti **ennuple** o **tuple** e vengono indicate con (d_1, d_2, \dots, d_n) con $d_1 \in D_1, d_2 \in D_2, \dots, d_n \in D_n$
- **L'istanza di una relazione** è l'insieme delle sue tuple in un determinato istante di tempo. L'istanza rappresenta il significato **estensionale** della relazione
- Le relazioni del modello relazionale sono **variabili nel tempo**
- Il numero **m** di tuple, presenti in un dato istante in una relazione R, viene detto cardinalità (corrente) della relazione e indicato con $\text{Card}(R)$.
- Una relazione può essere rappresentata in tre modi diversi:
 1. Per elencazione
 2. In forma insiemistica
 3. In forma tabellare

1 Le relazioni

1. Per elencazione:

PERSONA (Cognome, Nome, Età, Settore)=

{(Rossi, Paolo, 30, Maschio),
(Bianchi, Antonio, 23, Maschio),
(Neri, Ada, 35, Femmina)}

3 tuple

Singola tupla

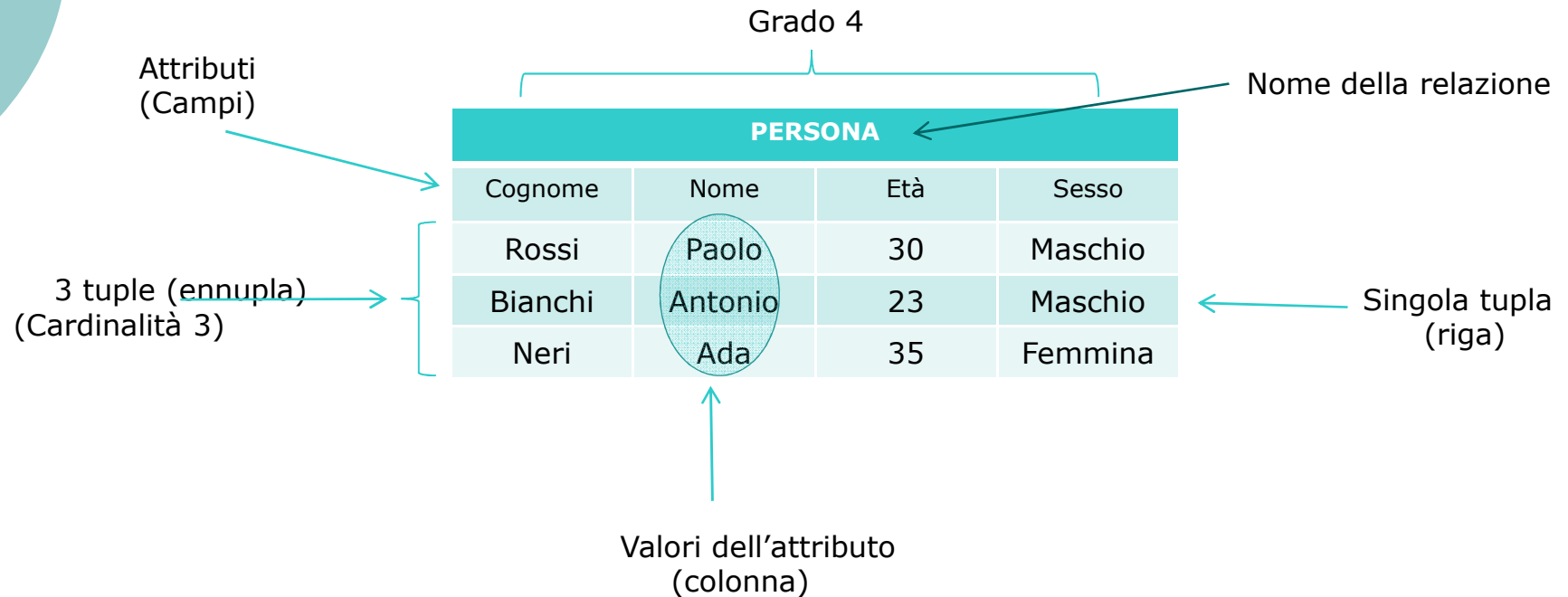
2. In forma Insiemistica:

PERSONA

- (Rossi, Paolo, 30, Maschio)
- (Bianchi, Antonio, 23, Maschio)
- (Neri, Ada, 35, Femmina)

1 Le relazioni

3. In forma Tabellare:





6 Dal Diagramma ER allo schema relazionale (fase di traduzione)

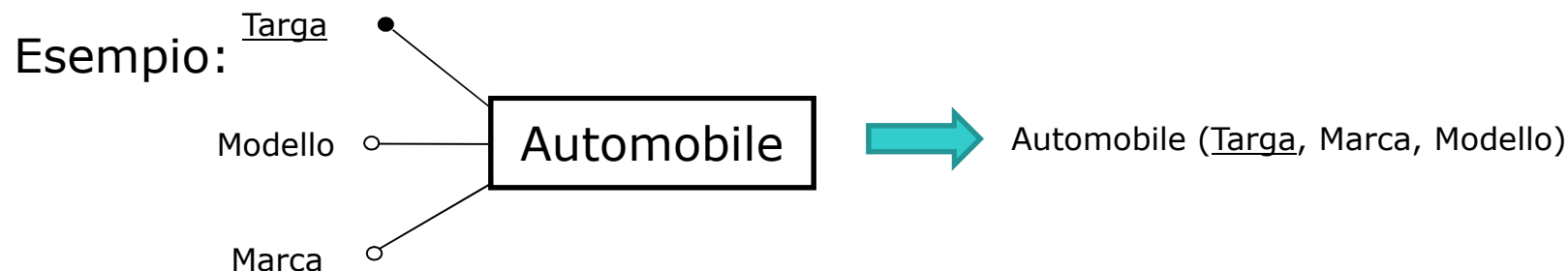
- Il modello relazionale mette a disposizione del progettista solo **le relazioni** per i vari aspetti della realtà
- Dal diagramma ER ristrutturato, il progettista deve effettuare un "mapping" delle entità e delle associazioni trasformandoli in relazioni del modello relazionale
- Con opportune **regole di trasformazione** vengono rappresentate:
 - Entità e attributi;
 - Associazioni di tipo 1:1;
 - Associazioni di tipo 1:N;
 - Associazioni di tipo N:N;

7 Rappresentazione delle entità e degli attributi

- Una Entità E con i suoi attributi elementari A1, A2,...,An è direttamente rappresentabile attraverso una relazione:

$$E(A1, A2, \dots, An)$$

- Ogni tipo di entità diventa una relazione (rappresentabile mediante una tabella)
- Ogni attributo dell'entità diventa un attributo della relazione (rappresentabile mediante una colonna della tabella)
- L'attributo chiave dell'entità diventa attributo chiave della relazione





8 Rappresentazione delle associazioni

- La traduzione delle Associazioni dipende:
 - Dal grado dell'associazione (numero di entità partecipanti)
 - Dagli eventuali vincoli di cardinalità

- Si hanno due alternative:
 - l'associazione viene rappresentata inserendo opportuni attributi (chiavi esterne) in una delle relazioni rappresentanti le entità partecipanti.
 - l'associazione stessa viene modellata con una relazione

- Vediamo ora per ogni tipo di associazione come si procede

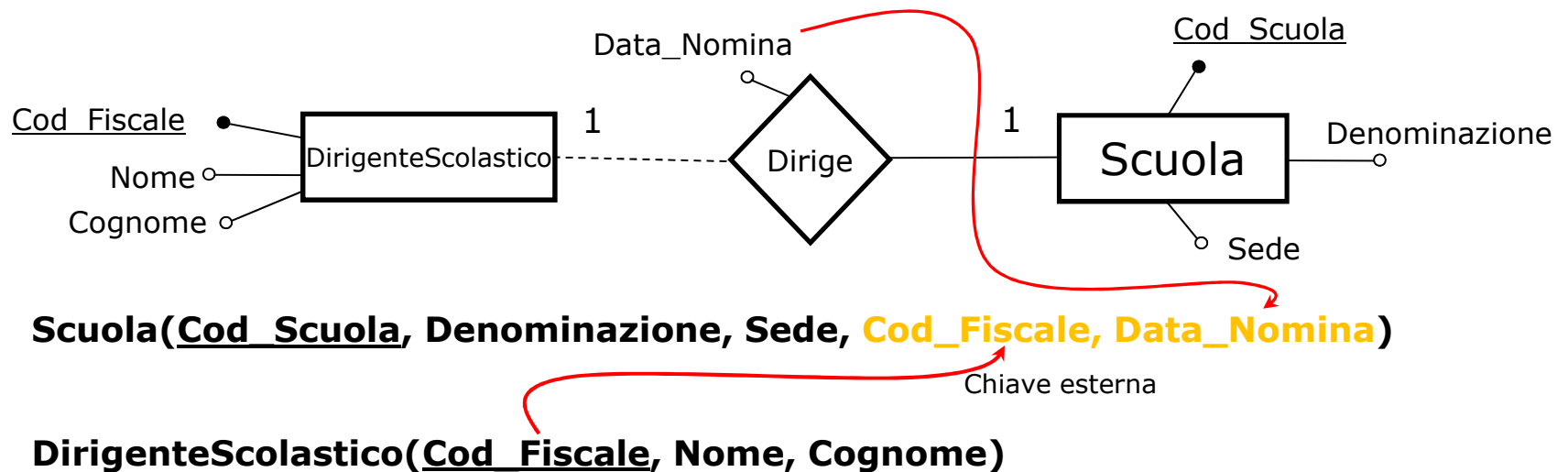


8.2 Rappresentazione delle associazioni binarie 1:1

- L'associazione viene modellata mediante attributi aggiuntivi inseriti nelle relazioni che modellano le entità partecipanti
- Si hanno due casi:
 - a) partecipazione obbligatoria (totale) di una sola delle due entità
 - b) partecipazione opzionale (parziale) od obbligatoria (totale) di entrambe le entità

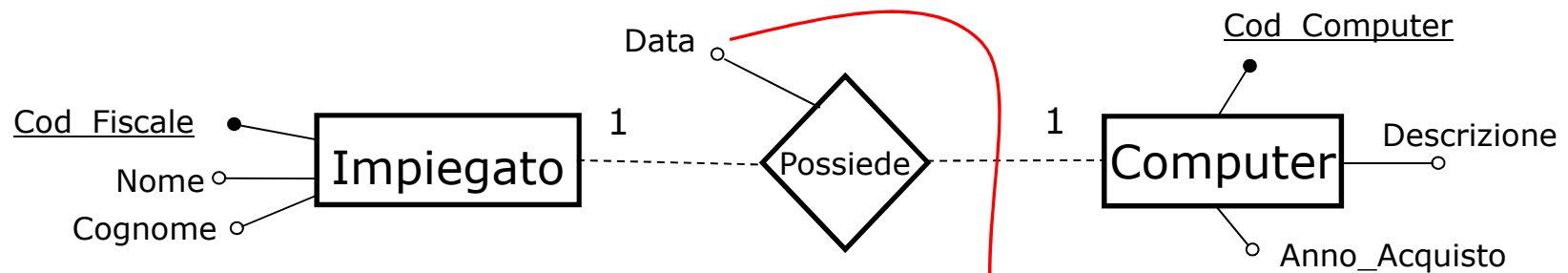
8.2 Rappresentazione delle associazioni binarie 1:1

- **Caso a:** partecipazione obbligatoria (totale) di una sola delle due entità
- La relazione che rappresenta **l'entità per cui l'associazione è obbligatoria (totale)** contiene come **chiave esterna** la chiave della relazione che rappresenta l'altra entità e come attributi anche **gli attributi dell'associazione**



8.2 Rappresentazione delle associazioni binarie 1:1

- **Caso b:** partecipazione opzionale (parziale) od obbligatoria (totale) di entrambe le entità
- La traduzione avviene come già visto per il caso (a), ma la relazione da ampliare **può essere scelta indifferentemente**



Una delle due possibili soluzioni:

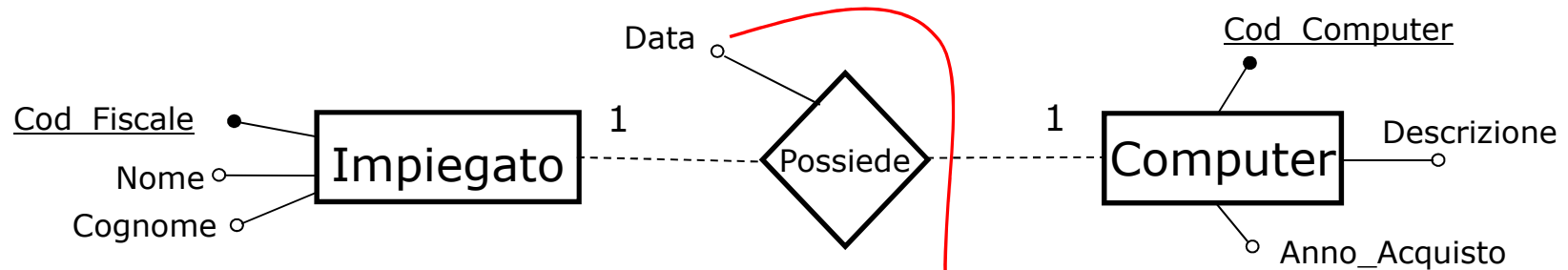
Computer(Cod Computer, Descrizione, Anno_Acquisto, Cod_Fiscale, Data)

Impiegato(Cod Fiscale, Nome, Cognome)

8.2 Rappresentazione delle associazioni binarie 1:1

- **Caso b:**

-



L'altra possibile soluzione:

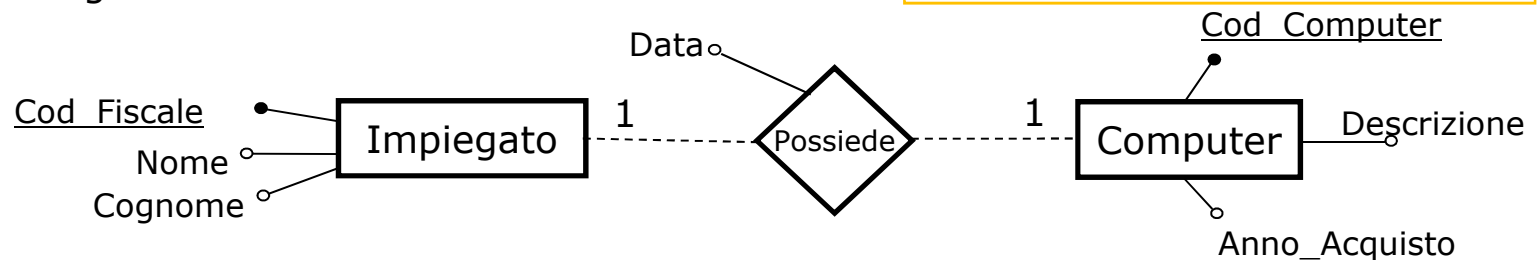
Impiegato(Cod Fiscale, Nome, Cognome, **Cod_Computer, **Data**)**

Chiave esterna

Computer(Cod Computer, Descrizione, Anno_Acquisto,)

8.2 Rappresentazione delle associazioni binarie 1:1

- **Approfondimento:**
nel caso particolare di partecipazione opzionale (parziale) di entrambe le entità (come nell'esempio appena visto) si può decidere, in alternativa alla soluzione presentata, di introdurre una relazione nuova per modellare l'associazione
- Tale nuova relazione conterrà:
 - le chiavi delle entità partecipanti
 - gli attributi dell'associazione



Impiegato(Cod Fiscale, Nome, Cognome)

Computer(Cod Computer, Descrizione, Anno_Acquisto,)

Possiede(Cod Fiscale, Cod Computer, Data)

Nota: (la chiave primaria della nuova relazione inserita sarà o una o l'altra chiave esterna)

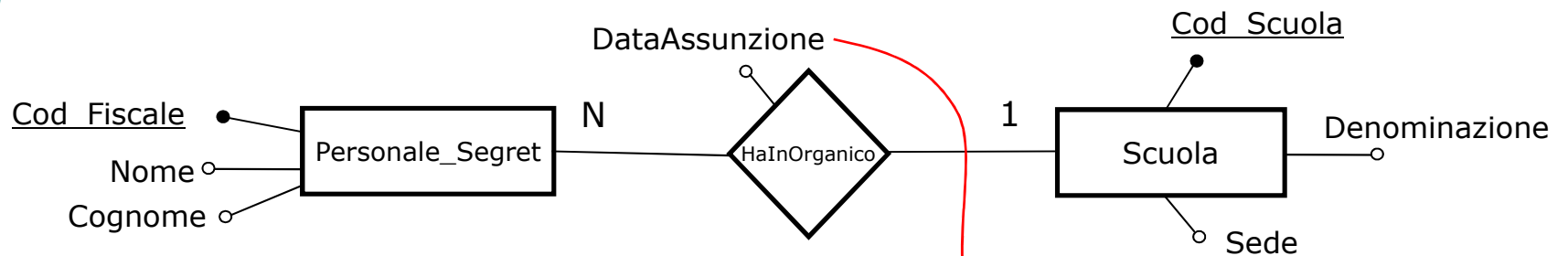


8.1 Rappresentazione delle associazioni binarie 1:N (uno a molti)

- L'associazione viene modellata mediante attributi aggiuntivi inseriti nelle relazioni che modellano le entità partecipanti
- Si inseriscono nella relazione **dell'entità dal lato N** ulteriori attributi:
 - La chiave della relazione corrispondente all'entità del lato 1 come **chiave esterna**
 - Gli attributi dell'associazione come **attributi aggiuntivi**

8.1 Rappresentazione delle associazioni binarie 1:N (uno a molti)

○ Esempio1



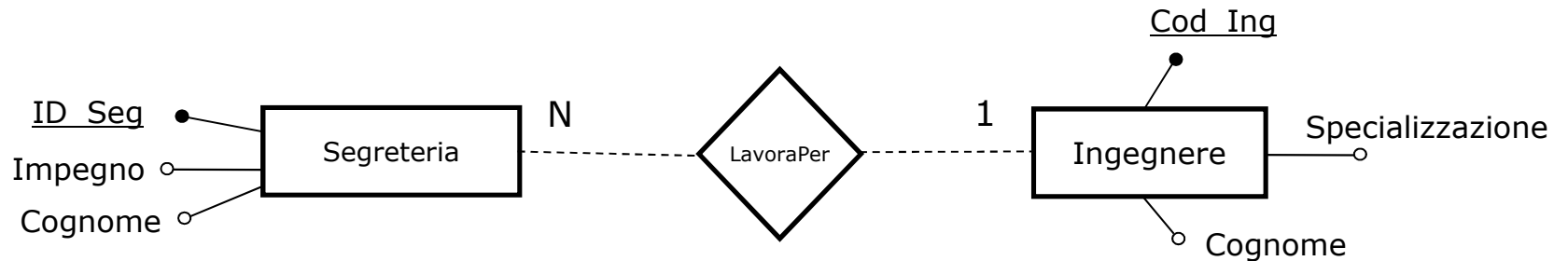
Personale_Segret(Cod_Fiscale, Nome, Cognome, **Cod_Scuola, **DataAssunzione**)**

Scuola(Cod_Scuola, Denominazione, Sede)

Chiave esterna

8.1 Rappresentazione delle associazioni binarie 1:N (uno a molti)

○ Esempio2



Segretaria(ID_Seg, Impegno, Cognome, **Cod_Ing)**

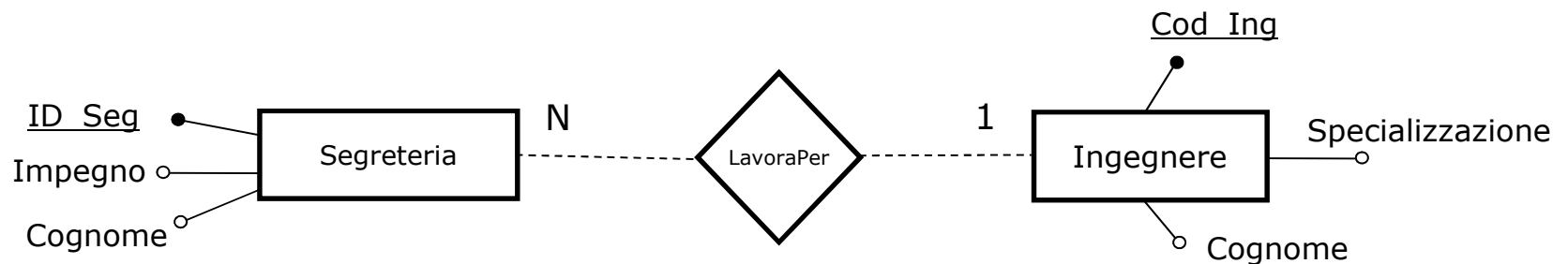
Chiave esterna

Ingegnere(Cod_Ing, Cognome, Specializzazione)

8.1 Rappresentazione delle associazioni binarie 1:N (uno a molti)

- **Approfondimento:**
nel caso particolare di partecipazione opzionale dell'entità dal lato uno si può decidere di introdurre una relazione nuova per modellare l'associazione
- Tale nuova relazione conterrà:
 - Le chiavi delle entità partecipanti
 - Gli attributi dell'associazione

Vantaggi: mai valori nulli
Svantaggi: una relazione in più



Segreteria(ID_Seg, Cognome, Impegno)

Ingegnere(Cod_Ing, Cognome, Specializzazione)

LavoraPer(Cod_Ing, ID_Seg)

Nota: (la chiave primaria della nuova relazione inserita sarà l'insieme delle chiavi esterne)

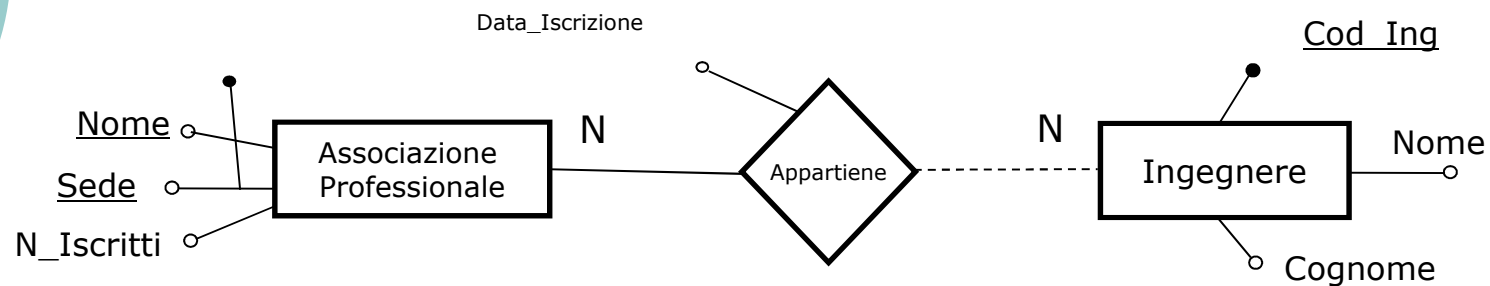


8.3 Rappresentazione delle associazioni binarie N:N (Molti a Molti)

- L'associazione viene modellata mediante una **nuova relazione** che prenderà il nome dell'associazione ed avrà opportuni attributi.
- Si inseriscono, nella nuova relazione che modella l'associazione, come attributi:
 - **Le chiavi di entrambe le entità** che partecipano all'associazione **come chiavi esterne**
 - Gli eventuali **attributi dell'associazione**
- La nuova associazione avrà come **chiave primaria l'insieme delle chiavi esterne.**

8.3 Rappresentazione delle associazioni binarie N:N (Molti a Molti)

- Esempio:



AssociazioneProfessionale(Nome, Sede, N_Iscritti)

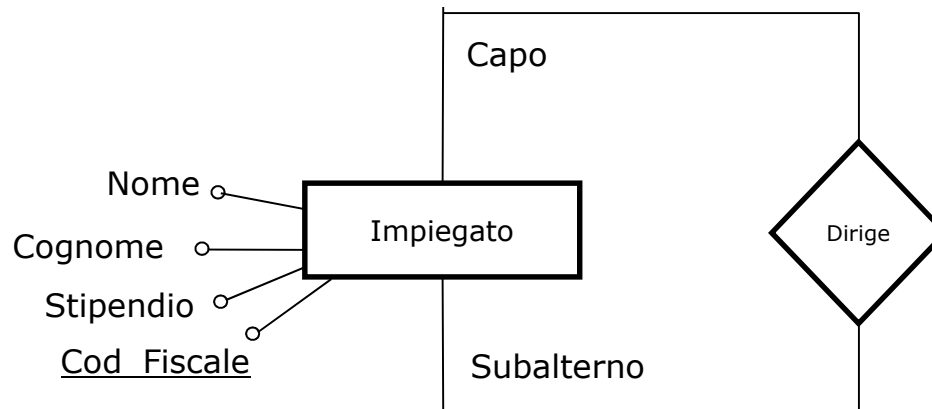
Ingegnere(Cod_Ing, Cognome, Nome)

Appartiene(Nome, Sede, ID_Ing , Data_Iscrizione)

Nota: (Come già detto, la chiave primaria della nuova relazione inserita sarà l'insieme delle chiavi esterne)

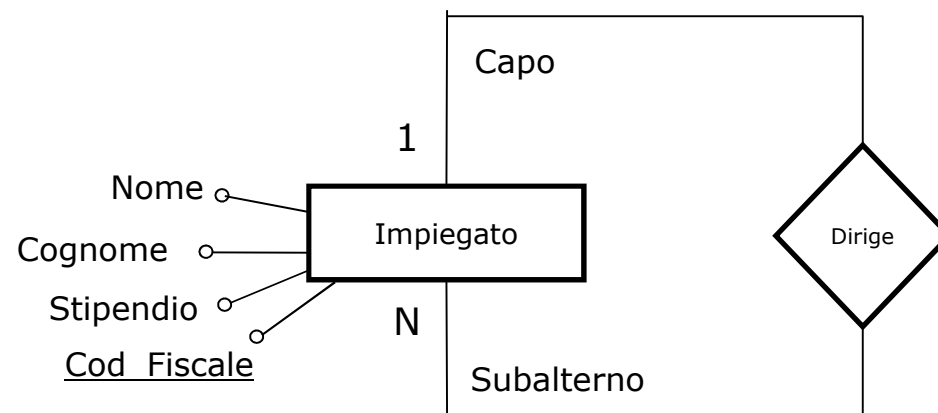
8.4 Rappresentazione delle associazioni unarie 1:1, 1:N, N:N

- Per l'associazione unaria si procede come già visto per le associazioni binarie distinguono due casi:
 - L'associazione unaria ha cardinalità 1:1 o 1:N
 - L'associazione unaria ha cardinalità N:N



8.4 Rappresentazione delle associazioni unarie 1:1, 1:N, N:N

- L'associazione unaria con cardinalità 1:1 o 1:N

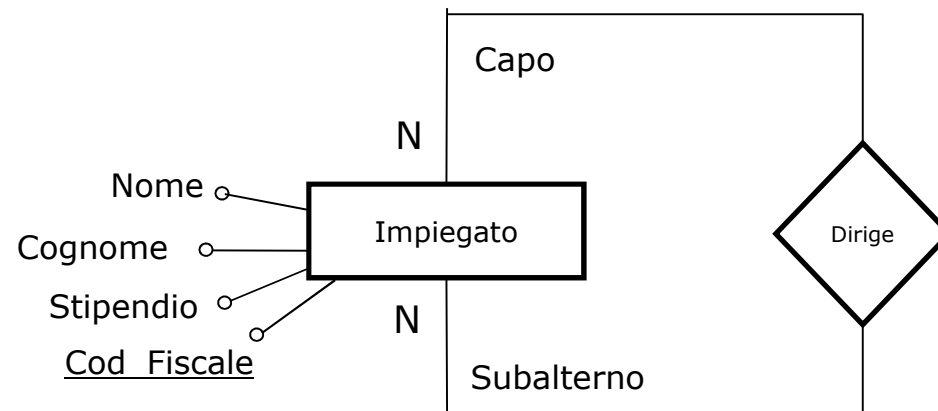


Impiegato(Cod Fiscale, Nome, Cognome, Stipendio, Cod_Fiscale_Capo)

Nota: Eventuali attributi dell'associazione diventano attributi dell'entità

8.4 Rappresentazione delle associazioni unarie 1:1, 1:N, N:N


- L'associazione unaria con cardinalità N:N



Impiegato(Cod Fiscale, Nome, Cognome, Stipendio)

Dirige(Cod Fiscale Subalterno, Cod Fiscale Capo)

Nota: Eventuali attributi dell'associazione diventano attributi della nuova entità

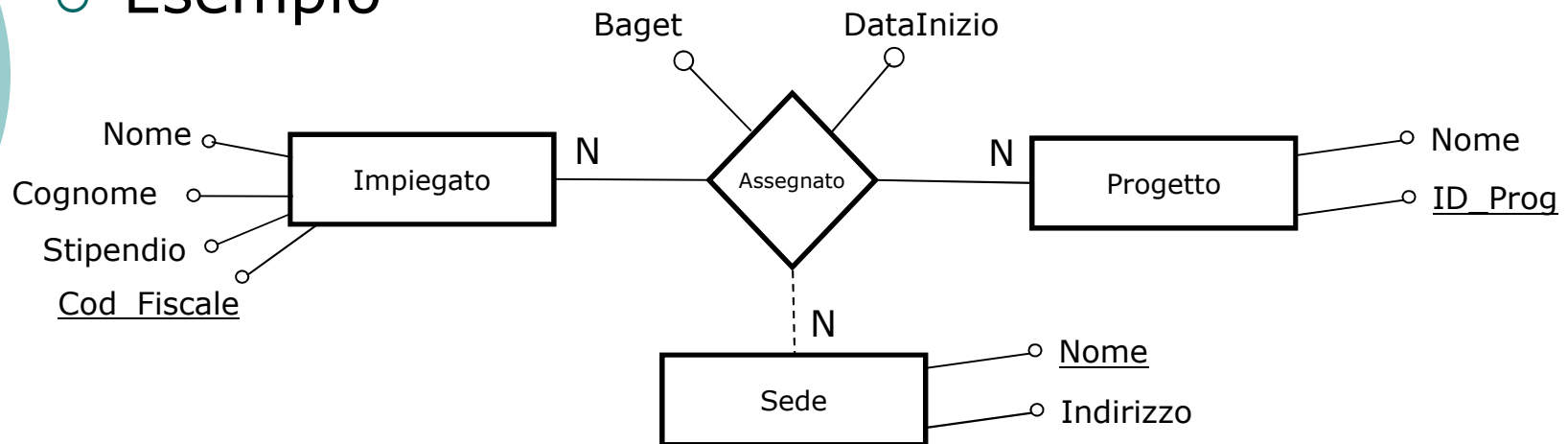


8.5 Rappresentazione delle associazioni non binarie (n-aria)

- Per l'associazione non binarie (n-aria) viene modellata mediante una **nuova relazione** che prenderà il nome dell'associazione ed avrà opportuni attributi.
- Si inseriscono, nella nuova relazione che modella l'associazione n-aria, come attributi:
 - **Le chiavi di tutte le entità** che partecipano all'associazione **come chiavi esterne**
 - Gli eventuali **attributi dell'associazione**
- La nuova associazione avrà come **chiave primaria l'insieme delle chiavi esterne.**

8.5 Rappresentazione delle associazioni non binarie (n-aria)

○ Esempio



Impiegato(Cod Fiscale, Nome, Cognome, Stipendio)

Progetto (Nome, ID Prog)

Sede(Nome, Indirizzo)

Assegnato(Cod Fiscale, ID Progetto, Nome Sede , Bage, DatInizio)

Nota: Eventuali attributi dell'associazione diventano attributi della nuova relazione



11 Le operazioni relazionali

- Ci focalizziamo ora sulle operazioni che ci consentono di interrogare una base di dati relazionale.
- Definizione:
 - Interrogare una base di dati significa estrarre o combinare una o più tabelle per generare nuove relazioni contenenti i dati e le informazioni che ci interessano.
- I linguaggi utilizzati sono di tipo non procedurale e utilizzano due tipi di approccio:
 - Basato sull'algebra relazionale (che useremo)
 - Basato sul calcolo relazionale



11 Le operazioni relazionali

- Con ambedue gli approcci si formulano interrogazioni alla base di dati
- Approccio basato sull'**algebra relazionale**
 - Utilizza alcuni **operatori di algebra relazionale**
- Approccio basato sul **calcolo relazionale**
 - Utilizza il **calcolo dei predicati del primo ordine**
- I due approcci sono considerati equivalenti

11 Le operazioni relazionali

- Approccio relazionale:

Utilizza **5 operatori relazionali fondamentali** con i quali è possibile formulare qualsiasi tipo di interrogazione sulla base di dati:

- Unione U
 - Differenza $-$
 - Prodotto Cartesiano X
 - Restrizione (Selezione) σ
 - Proiezione π
- } Operazioni Binarie
- } Operazioni Unarie

- Vi sono inoltre altri **2 operatori relazionali derivati** che corrispondono ad una composizione di quelli base:

- Intersezione \cap
- Giunzione naturale (join) \bowtie

11.1 Unione di relazioni

- Definizione di Compatibilità
 - Due relazioni R ed S vengono dette compatibili se:
 - Hanno lo stesso numero di attributi
 - Ogni attributo che si trova nella stessa posizione all'interno delle due relazioni è dello stesso tipo.
- Esempio:

Persona(Nome:Stringa, Stipendio:Interro, DataNascita:Date)

Dipendente(Nominativo:Stringa, Stip:Interro, DNascita:Date)

```
graph TD; P1[Persona(Nome:Stringa, Stipendio:Interro, DataNascita:Date)] --> D1[Dipendente(Nominativo:Stringa, Stip:Interro, DNascita:Date)]; P2[Persona(Nome:Stringa, Stipendio:Interro, DataNascita:Date)] --> D2[Dipendente(Nominativo:Stringa, Stip:Interro, DNascita:Date)]; P3[Persona(Nome:Stringa, Stipendio:Interro, DataNascita:Date)] --> D3[Dipendente(Nominativo:Stringa, Stip:Interro, DNascita:Date)];
```

11.1 Unione di relazioni

- Unione di relazioni (U)
 - Date due relazioni **compatibili** R e S, l'unione di R con S è la relazione ottenuta dall'unione insiemistica delle due relazioni:

$$R \cup S = \{t \mid t \in R \text{ OR } t \in S\}$$

R

- (Rossi, 2000, 2001/11/08)
- (Bianchi, 2200, 1996/10/23)
- (Neri, 2560, 1998/09/25)

S

- (Rossi, 2000, 2001/11/08)
- (Giovanni, 2500, 1998/10/26)
- (Luigi, 2700, 1995/03/28)

- (Bianchi, 2200, 1996/10/23)
- (Neri, 2560, 1998/09/25)

- (Rossi, 2000, 2001/11/08)

- (Giovanni, 2500, 1998/10/26)
- (Luigi, 2700, 1995/03/28)

RUS

11.1 Unione di relazioni

○ Unione di relazioni (U)

$$R \cup S = \{t \mid t \in R \text{ OR } t \in S\}$$

R		
Nome	Stipendio	DatNascita
Rossi	2000	2001/11/08
Giovanni	2500	1998/10/26
Luigi	2700	1995/03/28

S		
Nominativo	Stip	DNascita
Rossi	2000	2001/11/08
Bianchi	2200	1996/10/23
Neri	2560	1998/09/25

RUS		
Nome	Stipendio	DatNascita
Rossi	2000	2001/11/08
Giovanni	2500	1998/10/26
Luigi	2700	1995/03/28
Bianchi	2200	1996/10/23
Neri	2560	1998/09/25

Si ha che:

$$\text{Grado}(R \cup S) = \text{Grado}(R) = \text{Grado}(S)$$

$$\text{Card}(R \cup S) = \text{Card}(R) + \text{Card}(S) - \text{numero di tuple ripetute}$$

11.2 Differenza di relazioni

- Differenza di relazioni (-)
 - Date due relazioni **compatibili** R e S, la differenza di R con S è la relazione ottenuta dalla differenza insiemistica delle due relazioni:

$$R - S = \{t \mid t \in R \text{ AND } t \notin S\}$$

R

- (Rossi, 2000, 2001/11/08)
- (Bianchi, 2200, 1996/10/23)
- (Neri, 2560, 1998/09/25)

S

- (Rossi, 2000, 2001/11/08)
- (Giovanni, 2500, 1998/10/26)
- (Luigi, 2700, 1995/03/28)

- (Bianchi, 2200, 1996/10/23)
- (Neri, 2560, 1998/09/25)

$R - S$

11.2 Differenza di relazioni

○ Differenza di relazioni (-)

$$R - S = \{t \mid t \in R \text{ AND } t \notin S\}$$

R		
Nome	Stipendio	DatNascita
Rossi	2000	2001/11/08
Giovanni	2500	1998/10/26
Luigi	2700	1995/03/28

S		
Nominativo	Stip	DNascita
Rossi	2000	2001/11/08
Bianchi	2200	1996/10/23
Neri	2560	1998/09/25

R - S		
Nome	Stipendio	DatNascita
Giovanni	2500	1998/10/26
Luigi	2700	1995/03/28

Si ha che:

$\text{Grado}(R - S) = \text{Grado}(R) = \text{Grado}(S)$

$\text{Card}(R - S) = \text{Card}(R) - \text{numero di tuple presenti anche in } S$



11.5 Prodotto cartesiano di relazioni

- Prodotto cartesiano di relazioni (x)
 - Date due relazioni R e S rispettivamente di grado g1 e g2 e cardinalità c1 e c2, il prodotto cartesiano è la relazione di **grado g1+g2** e **cardinalità c1 x c2**, le cui tuple si ottengono concatenando **ogni tupla di R con ogni tupla di S**

$$R \times S = \{t \mid t = r \text{ conc } s, r \in R, s \in S\}$$

r ed s sono due tuple rispettivamente della relazione R e della relazione S

11.5 Prodotto cartesiano di relazioni

○ Prodotto cartesiano di relazioni (x)

Alunni		
Matricola	Nominativo	DatNascita
1234	Bianchi	2001/11/08
1235	Rossi	1998/10/26
1236	Neri	1995/03/28

Testi		
CodTesto	Titolo	Materia
T1	L'italia oggi	Italiano
T2	La Matematica	Matematica
T3	Informatica Teorica	Informatica

Alunni x Testi					
Matricola	Nominativo	DatNascita	CodTesto	Titolo	Materia
1234	Bianchi	2001/11/08	T1	L'italia oggi	Italiano
1234	Bianchi	2001/11/08	T2	La Matematica	Matematica
1234	Bianchi	2001/11/08	T3	Informatica Teorica	Informatica
1235	Rossi	1998/10/26	T1	L'italia oggi	Italiano
1235	Rossi	1998/10/26	T2	La Matematica	Matematica
1235	Rossi	1998/10/26	T3	Informatica Teorica	Informatica
1236	Neri	1995/03/28	T1	L'italia oggi	Italiano
1236	Neri	1995/03/28	T2	La Matematica	Matematica
1236	Neri	1995/03/28	T3	Informatica Teorica	Informatica

Si ha che:

$$\text{Grado}(\text{Alunni x Testi}) = \text{Grado}(\text{Alunni}) + \text{Grado}(\text{Testi}) = g_1 + g_2$$

$$\text{Card}(\text{Alunni x Testi}) = \text{Card}(\text{Alunni}) \times \text{Card}(\text{Testi}) = c_1 \times c_2$$



11.3 Proiezione di una relazione π

- Proiezione di una relazione (π)
 - Data una relazione R e un sottoinsieme $A\{A_1, A_2, \dots, A_k\}$ dei suoi attributi, si definisce **proiezione di R su A** la relazione di grado K che si ottiene considerando solo le colonne di R relative agli attributi contenuti in A ed eliminando le eventuali tuple duplicate.

$$\pi_{A_1, A_2, \dots, A_k}(R) = \{t[A_1, A_2, \dots, A_k] \mid t \in R\}$$

L'effetto di una proiezione è quello di selezionare un certo numero di colonne dalla tabella della relazione sulla quale viene applicata l'operazione

11.3 Proiezione di una relazione π

○ Proiezione di una relazione (π)

$$Rif_Ditta = \pi_{Ditta, Indirizzo}(Cliente) = \{t[Ditta, Indirizzo] \mid t \in Cliente\}$$

Cliente			
<u>Id Cliente</u>	Nome	Ditta	Indirizzo
C001	Bianchi	Polis	Via Po, 15
C002	Rossi	Tai&Ti	Via Firenze,45
C003	Neri	B&B	Piazza Principe,1
C004	Verdi	Polis	Via Po, 15

<i>Rif _ Ditta</i>	
Ditta	Indirizzo
Polis	Via Po, 15
Tai&Ti	Via Firenze,45
B&B	Piazza Principe,1

Si ha che:

$\text{Grado}(S) \leq \text{Grado}(R) \rightarrow \text{Grado}(Rif_Ditta) \leq \text{Grado}(Cliente)$

$\text{Card}(S) \leq \text{Card}(R) \rightarrow \text{Card}(Rif_Ditta) \leq \text{Card}(Cliente)$



11.4 Restrizione di una relazione σ

- Restrizione (o selezione) di una relazione (σ)
 - Data una relazione R e un Predicato P (semplice o composto: AND, OR, NOT, >, <, >=, <=, =) sui suoi attributi, la restrizione (o selezione) di R a P è la relazione costituita dalle tuple di R che soddisfano il predicato P

$$\sigma_P(R) = \{t \mid t \in R \text{ AND } P(t)\}$$

L'effetto di una restrizione è quello di selezionare un certo numero di righe (tuple) della tabella secondo le indicazioni definite dal Predicato

11.4 Restrizione di una relazione σ

○ Selezione di una relazione (σ)

Predicato: Selezionare dalla relazione Clienti tutte le informazioni dei clienti della stessa ditta Polis.

Predicato: $P(t) = \{Ditta = \text{"Polis"}\}$

$$S = \sigma_{Ditta=Polis}(Cliente) = \{t \mid t \in Cliente \text{ AND } Ditta = \text{"Polis"}\}$$

Cliente				
<u>Id Cliente</u>	Nome	Ditta	Indirizzo	Provincia
C001	Bianchi	Polis	Via Po, 15	MI
C002	Rossi	Tai&Ti	Via Firenze,45	TO
C003	Neri	B&B	Piazza Principe,1	MI
C004	Verdi	Polis	Via Po, 15	MI

S				
<u>Id Cliente</u>	Nome	Ditta	Indirizzo	Provincia
C001	Bianchi	Polis	Via Po, 15	MI
C004	Verdi	Polis	Via Po, 15	MI

Si ha che:

$\text{Grado}(S) = \text{Grado}(R)$

$\text{Card}(S) \leq \text{Card}(R)$

11.4 Restrizione di una relazione σ

Esempio1

A	B	C
a	b	c
d	a	f
c	b	d

R

A	B	C
a	b	c
c	b	d

$$\sigma_{B=b}(R)$$

A	B	C
d	a	f

$$\sigma_{NOTB=b}(R)$$

A	B	C
a	b	c
c	b	d

$$\sigma_{B=b \text{ OR } A=C}(R)$$

A	B	C
---	---	---

$$\sigma_{B=b \text{ AND } A=C}(R)$$

13 Interrogazine sullo schema relazionale

Esempio2

Imp#	Nome	Mansione	Data_A	Stipendio	Premio_P	Dip#
7369	Rossi	ingegnere	17-Dic-80	1600,00	500,00	20
7499	Andrei	tecnico	20-Feb-81	800,00	?	30
7521	Bianchi	tecnico	20-Feb-81	800,00	100,00	30
7566	Rosi	dirigente	02-Apr-81	2975,00	?	20
7654	Martini	segretaria	28-Set-81	800,00	?	30
7698	Blacchi	dirigente	01-Mag-81	2850,00	?	30
7782	Neri	ingegnere	01-Giu-81	2450,00	200,00	10
7788	Scotti	segretaria	09-Nov-81	800,00	?	20
7839	Dare	ingegnere	17-Nov-81	2600,00	300,00	10
7844	Turni	tecnico	08-Set-81	1500,00	?	30
7876	Adami	ingegnere	28-Set-81	1100,00	500,00	20
7900	Gianni	ingegnere	03-Dic-81	1950,00	?	30
7902	Fordi	segretaria	03-Dic-81	1000,00	?	20
7934	Milli	ingegnere	23-Gen-82	1300,00	150,00	10
7977	Verdi	dirigente	10-Dic-80	3000,00	?	10

$\Pi_{\text{Nome}} (\sigma_{\text{Stipendio} > 2000} (\text{Impiegati}))$

Nome
Rosi
Blacchi
Neri
Dare
Verdi

$\Pi_{\text{Nome, Dip\#}} (\sigma_{\text{Stipendio} > 2000 \text{ AND } \text{Mansione} = \text{'ingegnere'}} (\text{Impiegati}))$

Nome	Dip#
Neri	10
Dare	10

Altro esempio p.71 con fig. A3.31

11.6 Intersezione di due relazioni \cap

○ Intersezione di relazioni (\cap)

$$R \cap S = \{t \mid t \in R \text{ AND } t \in S\}$$

R		
<u>Nome</u>	<u>Stipendio</u>	<u>DatNascita</u>
Rossi	2000	2001/11/08
Giovanni	2500	1998/10/26
Luigi	2700	1995/03/28

$R \cap S$		
<u>Nome</u>	<u>Stipendio</u>	<u>DatNascita</u>
Rossi	2000	2001/11/08

S		
<u>Nominativo</u>	<u>Stip</u>	<u>DNascita</u>
Rossi	2000	2001/11/08
Bianchi	2200	1996/10/23
Neri	2560	1998/09/25

Si ha che:

$$\text{Grado}(R \cap S) = \text{Grado}(R) = \text{Grado}(S)$$

$$\text{Card}(R \cap S) \leq \text{Alla Cardinalità minore fra le Card}(R) \text{ e Card}(S)$$

Ulteriore esercizio sul libro a pag.66

11.7 Giunzione di due relazioni




- Giunzione (join) di due relazioni
- Vi sono vari tipi di Join, noi ne analizzeremo 3:
 - Join con predicato generico P
 - Join con predicato di uguaglianza (equi-join)
 - **Join naturale** (equi-join in cui vengono eliminate le colonne ridondanti)

11.7 Giunzione di due relazioni



- Join con predicato generico
 - Date due relazioni R ed S ed un predicato P (semplice o composto), l'operazione di giunzione (join) di R e S equivale al prodotto cartesiano R x S seguito da una selezione con predicato P
 1. Si effettua il prodotto cartesiano di R e S;
 2. Sulla relazione ottenuta si opera una restrizione (selezione) sulle tuple che rispondono al predicato P

$$R \underset{P}{\bowtie} S = \sigma_P (R \times S)$$

Predicato di join 

11.7 Giunzione di due relazioni



R

A	B	C
1	2	3
4	5	6
7	8	9

$R \bowtie_{B < D} S$

A	B	C	D	E
1	2	3	3	1
1	2	3	6	2
4	5	6	6	2

$$R \bowtie_{B < D} S = \sigma_{B < D} (R \times S)$$

S

D	E
3	1
6	2

Join

$R \times S$

A	B	C	D	E
1	2	3	3	1
1	2	3	6	2
4	5	6	3	1
4	5	6	6	2
7	8	9	3	1
7	8	9	6	2

11.7 Giunzione di due relazioni



- Equi-Join con predicato di uguaglianza

- Date due relazioni R ed S ed un predicato P di uguaglianza, l'operazione di giunzione (equi-join) di R e S equivale al prodotto cartesiano R x S seguito da una selezione **basata sul predicato di uguaglianza dei valori di colonne**

1. Si effettua il prodotto cartesiano di R e S;
2. Sulla relazione ottenuta si opera una restrizione (selezione) sulle tuple che rispondono al predicato P

$$R \underset{P}{\bowtie} S = \sigma_P (R \times S)$$

Predicato di uguaglianza join

11.7 Giunzione di due relazioni



R

A	B	C
1	2	3
4	5	6
7	8	9

S

D	E
3	1
6	2
5	7

$R \bowtie_{A=E} S$

A	B	C	D	E
1	2	3	3	1
7	8	9	5	7

$$R \bowtie_{A=E} S = \sigma_{A=E} (R \times S)$$

R x S

A	B	C	D	E
1	2	3	3	1
1	2	3	6	2
1	2	3	5	7
4	5	6	3	1
4	5	6	6	2
4	5	6	5	7
7	8	9	3	1
7	8	9	6	2
7	8	9	5	7

Equi-Join

11.7 Giunzione di due relazioni



- Giunzione naturale (join) di due relazioni
 - Date due relazioni R ed S, l'operazione di giunzione naturale (join) di R e S sugli attributi A di R e B di S equivale ad un equi-join in cui vengono eliminate le colonne ridondanti le cui tuple si ottengono con il seguente procedimento:
 1. Si effettua il prodotto cartesiano di R e S;
 2. Sulla relazione ottenuta si opera una restrizione (selezione) sulle tuple aventi gli stessi valori degli attributi di A e B
 3. La relazione che si ottiene avrà le colonne A e B uguali, per cui eliminiamo una delle due (Proiezione di tutte le colonne esclusa la colonna A o B).

$$R \bowtie_{A=B} S = \pi_{(\text{tutti i campi escluso A o B})} (\sigma_{A=B} (R \times S))$$

Predicato di equi-join

11.7 Giunzione di due relazioni



R

A	B	C
1	2	3
4	5	6
7	8	9

$R \bowtie_{A=E} S$

A	B	C	D
1	2	3	3
7	8	9	5

$$R \bowtie_{A=E} S = \pi_{A,B,C,D} (\sigma_{A=E} (R \times S))$$

S

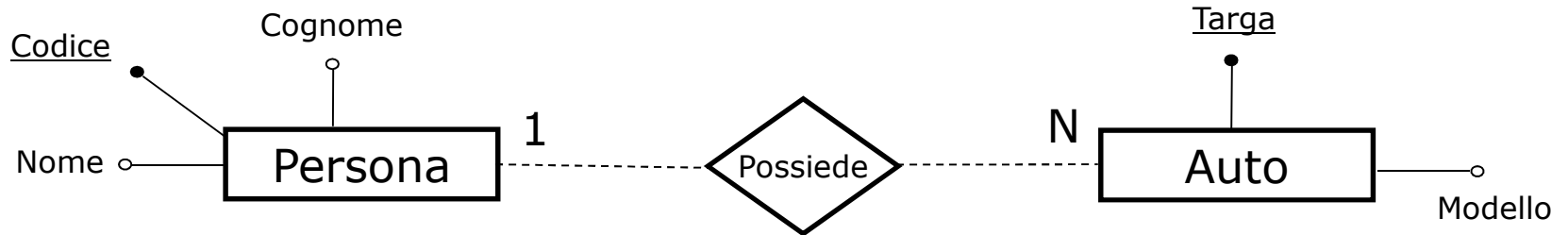
D	E
3	1
6	2
5	7

R x S

A	B	C	D	E
1	2	3	3	1
1	2	3	6	2
1	2	3	5	7
4	5	6	3	1
4	5	6	6	2
4	5	6	5	7
7	8	9	3	1
7	8	9	6	2
7	8	9	5	7

Join Naturale

11.7 Giunzione di due relazioni



Persona

<u>Codice</u>	Nome	Cognome
001	Mario	Rossi
002	Luigi	Bianchi
003	Giuseppe	Neri
004	Franco	Neri

Auto

<u>Targa</u>	Modello	Cod_Persona
PI45BCD	Tipo1	002
MN67HGD	Tipo2	003
PR42YUE	Tipo3	005
MI12LMN	Tipo4	001
RE89KLE	Tipo5	002

Codice	Nome	Cognome	Targa	Modello
001	Mario	Rossi	MI12LMN	Tipo4
002	Luigi	Bianchi	PI45BCD	Tipo1
002	Luigi	Bianchi	RE89KLE	Tipo5
003	Giuseppe	Neri	MN67HGD	Tipo2

$$R \bowtie_{\text{Codice=Cod_Persona}} S = \pi_{\text{Codice, Nome, Cognome, Targa, Modello}} \left(\sigma_{\text{Codice=Cod_Persona}} (R \times S) \right)$$



14. Normalizzazione

- Processo di eliminazione delle “anomalie” che potrebbero essere presenti in una base di dati.
- Ripetizioni di dati con conseguente spreco di tempo (nelle fasi di inserimento e/o aggiornamento) e spazio di memoria



14. Normalizzazione

- Consideriamo la relazione (vedi testo p. 73)
- Anomalie che si riscontrano:
 - Di inserimento
 - Per inserire un nuovo cliente è necessario inserire un articolo ordinato.
 - Per inserire un nuovo articolo è necessario inserire un acquirente.
 - Anomalia dovuta alla chiave primaria composta: CodArticolo, CodCliente



14. Normalizzazione

- Consideriamo la relazione (vedi testo p. 73)
- Anomalie che si riscontrano:
 - Di cancellazione
 - Cancellando una tupla relativa ad un acquisto si corre il rischio di cancellare anche i dati relativi al cliente. Se si cancella l'articolo (Neri, Art5) perdiamo anche i dati del cliente (nel nostro caso Citta di residenza)



14. Normalizzazione

- Consideriamo la relazione (vedi testo p. 73)
- Anomalie che si riscontrano:
 - In Aggiornamento
 - Se vogliamo modificare l'indirizzo di un cliente (nel nostro caso Città) occorre aggiornare ogni tupla in cui è presente il cliente.



14. Normalizzazione

- Le anomalie che bisogna verificare sono dunque tre:
 - Inserimento
 - Cancellazione
 - Aggiornamento
- In particolare quelle di Cancellazione e Aggiornamento sono diretta conseguenza della Ridondanza dei dati senza apporto di nuove informazioni



14. Normalizzazione

- Teoria della Normalizzazione:
 - Metodo per creare tabelle senza anomalie cioè creare tabelle "corrette"
- **Normalizzazione:**

Procedimento che consente di verificare se lo schema relazionale corrisponde ai "canoni standard" di correttezza della base di dati avvalendosi di un preciso insieme di regole, riportare le tabelle in quelle che sono definite "forme normali" delle tabelle relazionali.



14. Normalizzazione

- Regole basi a cui attenersi per adeguarsi alla teoria della normalizzazione:
 - Ogni tabella deve avere sempre una chiave primaria;
 - Ogni campo deve contenere “normalmente” un solo valore;
 - I campi di una tabella non devono dipendere da altri campi che non siano la chiave primaria;
 - Bisogna evitare le ripetizioni e la ridondanza dei dati.



14. Normalizzazione

- Livelli di Normalizzazione:
 - Forma normale zero "0FN" (a cui di solito non si fa cenno perché ritenuta regola base di uno schema relazionale)
 - Forma normale uno "1FN"
 - Forma normale due "2FN"
 - Forma normale tre "3FN"
 - Forma normale BCNF (Forma normale di Boyce-Codd)
- Se una relazione risponde ai canoni di un certo livello di normalizzazione, allora sicuramente risponde ai canoni delle forme normali inferiori



14. Normalizzazione

- Forma normale zero "0FN"
 - Ogni tabella deve avere una chiave primaria composta da un campo o da un insieme di campi.
 - Qualora in una tabella non vi sono campi in cui i dati non si ripetono, bisogna inserire un nuovo campo da considerare chiave primaria per adeguare la relazione alla forma "0FN" (vedi esempio p. 74)



14.1 Prima forma normale (1FN)

- Forma normale uno "1FN" anche detta forma atomica
 - Una volta che si è sicuro che le tabelle contengano le relative chiavi primarie ("0FN") occorre accertarsi che queste rispondano ai requisiti imposti dalle regole per la forma normale 1 ("1FN")
 - Ogni campo abbia un solo valore atomico deve essere cioè un campo semplice, quindi non composto e non multiplo.
 - Se nelle fasi precedenti si è provveduto ad una corretta "ristrutturazione dello schema relazionale" allora questa regola è automaticamente rispettata.



14.1 Prima forma normale (1FN)

- Forma normale uno "1FN" anche detta forma atomica
 - Un campo che prevede dati composti deve essere scomposto in più campi. L'indirizzo di una persona, normalmente, deve essere scomposto in più campi (vedi esempio p. 74)
 - Nota: La regola è cogente in funzione del tipo di utilizzo e interrogazioni previste dalle specifiche. Ad esempio un indirizzo (Via e numero civico) non sempre deve essere considerato dato composto ma dipende dal tipo di interrogazioni previste.



14.1 Prima forma normale (1FN)

- Forma normale uno "1FN" anche detta forma atomica
 - Un campo che debba contenere i numeri di telefono di una persona (campo multiplo) deve essere scorporato in un'altra tabella opportunamente legata alla prima. (vedi esempio p. 75)
 - Nota: questo passaggio coincide con la ristrutturazione in presenza di un campo multiplo, dove avevamo già previsto la creazione di un'altra entità legata alla prima con associazione 1:N.



14.2 Seconda forma normale (2FN)

- **Dipendenza funzionale**
- Prima di parlare di "2FN" chiariamo il concetto di dipendenza funzionale
- Sia R una relazione che contiene almeno due attributi X e Y.
Se il valore di Y varia al variare del valore di X, si dice che Y ha una **dipendenza funzionale** da X e si indica con $X \rightarrow Y$
(X determina Y; X è determinante per Y)
- Analoga definizione viene data se la dipendenza funzionale riguarda due o più attributi.



14.2 Seconda forma normale (2FN)

- **Dipendenza funzionale**
- Es. Consideriamo la relazione:

Studente(Matricola, Nome, Telefono, Corso, Voto)

Possiamo individuare le seguenti dipendenze funzionali:

Matricola → Nome

Matricola → Telefono

Matricola, Corso → Voto

Ad ogni nome corrisponde una Matricola.

Ad ogni telefono corrisponde una matricola.

Ad ogni voto corrisponde l'insieme di attributi Matricola, Corso.

Ulteriore Es. vedi p.76 testo)



14.2 Seconda forma normale (2FN)

- Forma normale due "2FN" anche detta "della dipendenza totale dalla chiave"
- Una volta che si è sicuro che le tabelle siano in 1FN occorre accertarsi che queste rispondano ai requisiti imposti dalle regole per la forma normale 2 ("2FN")
- La seconda forma normale si applica alle tabelle che hanno la **chiave primaria composta da più attributi**
 - Una relazione R è in seconda forma normale (2FN) se è in prima forma normale e ogni attributo, non chiave, dipende funzionalmente e completamente (cioè non parzialmente) dalla chiave primaria; non possono esistere attributi che dipendono solamente da una parte della chiave.



14.2 Seconda forma normale (2FN)

○ Es. vedi fotocopie pag. 52 e testo pag. 77

○ In generale:

Data una relazione

$R(\underline{A}, \underline{B}, C, D, E)$

Con le seguenti dipendenze

$A, B \rightarrow C$

$A \rightarrow D$

$B \rightarrow E$

Per rendere il DB conforme alle norme 2FN si dovrà trasformare la relazione R nelle seguenti tre relazioni conformi 2FN

$R1(\underline{A}, D)$

$R2(\underline{B}, E)$

$R3(\underline{A}, \underline{B}, C)$



14.3 Terza forma normale (3FN)

- Forma normale due "3FN" anche detta "della dipendenza transitiva"
- Una volta che si è sicuro che le tabelle siano in 2FN occorre accertarsi che queste rispondano ai requisiti imposti dalle regole per la forma normale 3 ("3FN")
- Eliminare la dipendenza transitiva degli attributi dalla chiave.
 - Una relazione R è in terza forma normale (3FN) se è in seconda forma normale e ogni attributo, non chiave, dipende direttamente dalla chiave primaria. In altri termini, la relazione R non deve possedere attributi non chiave che dipendono funzionalmente da altri attributi non chiave.



14.3 Terza forma normale (3FN)

○ Es. vedi fotocopie pag. 52,53 e testo pag. 78

○ In generale:

Data una relazione

$R(\underline{A}, B, C, D, E)$

Con le seguenti dipendenze

$A \rightarrow B$

$A \rightarrow E$

$B \rightarrow C$

$B \rightarrow D$

Si evince che sussiste una dipendenza transitiva fra

$A \rightarrow C$ e $A \rightarrow D$

Per rendere il DB conforme alle norme 3FN si dovrà decomporre trasformando la relazione R nelle seguenti tre relazioni conformi 3FN

$R1(\underline{B}, C)$

$R2(\underline{B}, D)$

$R3(\underline{A}, B, E)$



14.4 Forma normale di Boyce-Codd(BCNF)

- Una relazione R è in forma normale di Boyce-Codd(BCNF, Boyce-Codd Normal Form) quando rispetta le caratteristiche fondamentali del modello relazionale (1FN) e se, per ogni dipendenza $X \rightarrow Y$ definita su di essa, X (**il determinante**) è una **chiave candidata**.
Cioè ogni attributo dal quale dipendono altri attributi può essere una chiave.
- Se una relazione è in forma normale BCNF allora vuol dire che i campi dipendono da un determinante che è chiave candidata, quindi può essere presa come chiave primaria.
- Una relazione che rispetta la forma normale di Boyce-Codd è quindi anche 2FN e 3FN, ma non è vero il contrario.