15. IL MODELLO RELAZIONALE

Il *diagramma ER*, come abbiamo già visto, è un *modello concettuale* (ossia adatto a descrivere la realtà di interesse) indipendentemente da come saranno poi implementati i dati e le associazioni relative.
Abbiamo visto che per raggiungere l'obiettivo di progettare e realizzare una base di dati che soddisfi le esigenze dell'utente, è possibile seguire strade diverse in base al **modello** che si intende adottare. Il modello che abbiamo scelto di illustrare dettagliatamente è quello **puro relazionale.**
Abbiamo già detto inoltre che dopo la *progettazione concettuale* il passo successivo nella progettazione di una base di dati è la *progettazione logica* passo durante il quale si trasforma lo schema concettuale (nel nostro caso *il diagramma ER)* in una rappresentazione più efficiente rispetto al DBMS scelto detta *schema logico* (nel nostro caso *schema logico-relazionale*).

La **progettazione logica relazionale** consiste quindi nell'effettuare il "mapping" (ossia la *conversione, la traduzione*) di tutti gli oggetti rappresentati in un diagramma ER in un **insieme di relazioni** (rappresentate come **tabelle logiche**) che prendono il nome di **schema logico relazionale** (in quanto ottenute in accordo dell'omonimo approccio).

1) Le relazioni

Il **modello relazionale** dei dati, introdotto fin dal 1970 da **E.F. Codd**, prevede l'utilizzo del concetto matematico di **relazione tra insiemi** per strutturare i dati.

DEF: Una **relazione R** su una sequenza di insiemi $\mathbf{D_1}$, $\mathbf{D_2}$, … $\mathbf{D_n}$ (non necessariamente distinti) è un **sottoinsieme finito del prodotto cartesiano $\mathbf{D_1 \times D_2 \times \ldots \times D_n}$** che può essere indicato, secondo la teoria degli insiemi, con la scritta:

$$\mathbf{R \subseteq D_1 \times D_2 \times \ldots \times D_n}$$

dove:
- **n (con n > 1)** è detto **grado** della relazione ed è indicato con la scritta **Grado (R);**
- gli insiemi $\mathbf{D_1}$, $\mathbf{D_2}$, … $\mathbf{D_n}$ sono detti **domini** della relazione ed ognuno di essi può essere di un tipo di dato elementare *(ad esempio carattere, stringa, intero, reale, booleano, data).* Ad ogni dominio è associato un nome detto **attributo** che lo identifica univocamente all'interno della relazione.

DEF: Chiameremo **schema di una relazione R** il nome della relazione e la lista dei suoi attributi racchiusi tra parentesi tonde e separate da virgole che rappresenteremo con la seguente sintassi

<NomeRelazione> (<Attributo1>: <Tipo1>, <Attributo2>: <Tipo2>, … , <Attributon>: <Tipon>)

dove:
- **<Tipo1>**, **<Tipo2>**, … , **<Tipon>** sono i tipi elementari degli attributi (spesso in questa fase omessi per semplicità) come ad esempio carattere, stringa, intero, reale, booleano, data;

N.B. Per **convenzione** scriveremo sia i nomi delle relazioni sia i nome degli attributi **con la sola iniziale in maiuscolo**.
Lo schema di una relazione indica il **significato intensionale** di quest'ultima.

Esempio: consideriamo il seguente schema della relazione Persona utilizzato per rappresentare le caratteristiche di un essere umano:

Persona (CodFisc: Stringa(16), Cognome: Stringa(30), Nome: Stringa(20), Età: Intero, Sesso: Carattere)

Spesso per brevità di esposizione scriveremo:

Persona (CodFisc, Cognome, Nome, Età, Sesso)

DEF: Gli elementi di **R** sono detti **ennuple** o **n-ple** e vengono indicati con:

$$(\mathbf{d_1, d_2, \ldots, d_n})$$

dove $\mathbf{d_1 \in D_1, d_2 \in D_2, \ldots, d_n \in D_n}$
Chiameremo **istanza di una relazione R** l'insieme delle sue ennuple in un determinato istante di tempo.
L'istanza di una relazione rappresenta il suo significato estensionale.

N.B. A differenza delle relazioni matematiche le relazioni del modello relazionale sono variabili nel tempo in quanto le ennuple possono essere inserite, cancellate, aggiornate.

Coerentemente con la definizione di insieme una relazione non può mai contenere ennuple uguali.

DEF: Il numero **m** di ennuple presenti *in un dato istante di tempo* in una relazione **R** viene detto **cardinalità** (corrente) della relazione e viene indicato con **Card (R)**.

Riepilogo: ELEMENTI di TEORIA DEGLI INSIEMI

Definizione di insieme

Un **insieme** è un ente costituito da oggetti.
Il concetto di **insieme** e di **oggetto** si assumono in matematica come primitivi.

Appartenenza ad un insieme

Se un oggetto ***a*** fa parte di un insieme A si dice che esso *è* un suo ***elemento*** o che ***a*** appartiene ad A; in simboli

$$a \in A$$

Se un oggetto ***b*** *non* fa parte di un insieme A si dice che esso non *è* un suo ***elemento*** o che ***b*** non appartiene ad A; in simboli

$$b \notin A$$

Insieme vuoto

Un insieme privo di elementi si dice ***insieme vuoto*** e si denota con il simbolo ϕ

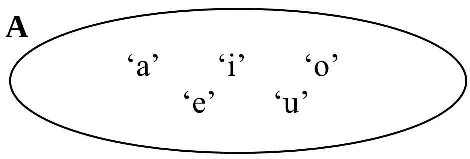
Insieme finito ed infinito

Un insieme A si dice ***finito*** se ha un numero finito di elementi; si dice ***ordine*** dell'insieme il numero dei suoi elementi.
Un insieme che non è *finito* si dice ***infinito***

Rappresentazione di un insieme

1. PER **ELENCAZIONE**: Scrivendone esplicitamente gli elementi separati da virgola e racchiusi tra parentesi graffe:
Nel nostro esempio:
A = { 'a', 'e', 'i', 'o', 'u' }

2. **GRAFICAMENTE**: Usando il diagramma di Eulero-Venn



3. TRAMITE **PROPRIETA' CARATTERISTICA**: Enunciando una proprietà che è soddisfatta da tutti e soli gli elementi dell'insieme:

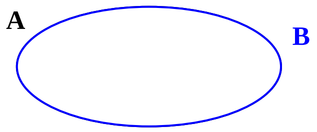
A = {x | x possiede la proprietà P }

Nel nostro esempio:
A = { x | x è una vocale }

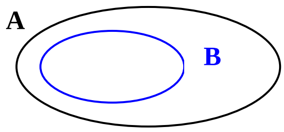
Sottoinsieme di un insieme

Siano A, B due insiemi.
Se ogni elemento di B appartiene anche ad A si dice che B ***è incluso*** in A o che B ***è sottoinsieme*** di A e si indica
$B \subseteq A$

Se ogni elemento di A appartiene a B e ogni elemento di B appartiene ad A allora A ***è uguale*** a B; in simboli
A = B



Se $B \subseteq A$ e $A \not\subset B$, allora vuol dire che *esiste almeno un elemento di* A *che non appartiene ad* B, pertanto si dice che B ***è incluso propriamente*** in A e si indica
$B \subset A$

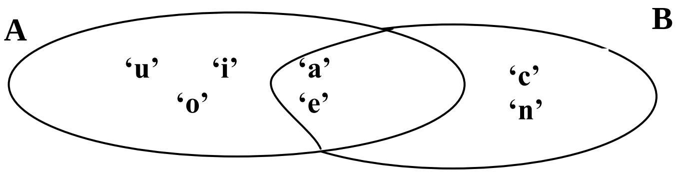


Operazioni con gli insiemi

Definizione: Siano A e B due insiemi. Dicesi ***unione*** di A e B l'insieme costituito dagli elementi che appartengono o ad A o a B;
in simboli
$A \cup B = \{x \mid x \in A \text{ o } x \in B\}$

N.B. Se $B \subseteq A \Rightarrow A \cup B = A$ e se $A \subseteq B \Rightarrow A \cup B = B$

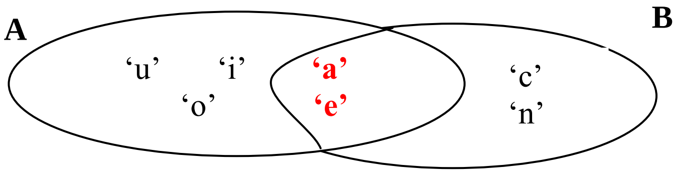
Esempio:
Sia A = { x | x è una vocale } e B = { x | x è una vocale della parola "cane"}



allora ***A ∪ B = { 'a', 'e', 'i', 'o', 'u', 'c', 'n' }***

Definizione**:** Siano A, B due insiemi. Dicesi **intersezione** di A e B l'insieme costituito dagli elementi che appartengono sia ad A che a B;
in simboli
$A \cap B = \{x \mid x \in A \text{ e } x \in B\}$

Esempio:
Sia A = { x | x è una vocale } e B = { x | x è una vocale della parola "cane"}



allora ***A ∩ B = { 'a', 'e' }***

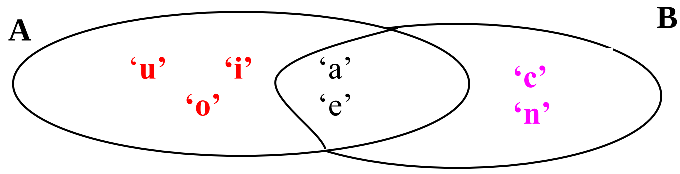
N.B. Se $B \subseteq A \Rightarrow A \cap B = B$ e se $A \subseteq B \Rightarrow A \cap B = A$

Se A e B non hanno elementi comuni allora $A \cap B = \phi$; essi si dicono **disgiunti** e la loro intersezione è l'insieme vuoto.

<u>Definizione</u> Siano A e B due insiemi. Si dice **differenza** tra gli insiemi A e B l'insieme costituito dagli elementi di A che non appartengono a B;
in simboli
$A - B = \{x \mid x \in A \text{ e } x \notin B\}$
N.B. Se $B \subset A$, A - B si dice *complementare* di B rispetto ad A.

Si dice **differenza** tra gli insiemi B e A l'insieme costituito dagli elementi di B che non appartengono a A;
in simboli
$B - A = \{x \mid x \in B \text{ e } x \notin A\}$
N.B. Se $A \subset B$, B - A si dice *complementare* di A rispetto ad B.

Esempio:
Sia A = { x | x è una vocale } e B = { x | x è una vocale della parola "cane"}



allora ***A - B = { 'i', 'o', 'u' }*** *e allora* ***B - A = { 'c', 'n' }***
<u>Quindi la differenza di A-B non gode della proprietà commutativa ossia A-B ≠ B - A</u>

Siano A, B e C tre insiemi.
Per le operazioni appena definite valgono le seguenti ***proprietà***:

	Intersezione		**Unione**
Idempotenza	$A \cap A = A$	**Idempotenza**	$A \cup A = A$
Commutativa	$A \cap B = B \cap A$	**Commutativa**	$A \cup B = B \cup A$
Associativa	$(A \cap B) \cap C = A \cap (B \cap C)$	**Associativa**	$(A \cup B) \cup C = A \cup (B \cup C)$
Distributiva	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$	**Distributiva**	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
	$A - (B \cap C) = (A - B) \cup (A - C)$		$A - (B \cup C) = (A - B) \cap (A - C)$

$$A - B \subseteq A$$

$$(A - B) \cap (A \cap B) = \varnothing$$

$$(A - B) \cap (B - A) = \varnothing$$

<u>Definizione</u> Dati due insiemi A e B non vuoti si definisce **prodotto cartesiano** di A e B, in simboli $A \times B$, l'insieme delle coppie ordinate (x,y) in cui $x \in A$ $y \in B$,
$A \times B = \{(x,y) \mid x \in A \text{ e } y \in B\}$
All'insieme $A \times B$ appartengono quindi tutte le possibili coppie in cui il primo elemento è sempre nel primo insieme e il secondo nel secondo insieme.
E' importante ricordare che le coppie del prodotto cartesiano sono <u>coppie ordinate</u>, cioè la coppia (x, y) risulta diversa dalla coppia (y, x), cioè $A \times B \neq B \times A$.
<u>Quindi il prodotto cartesiano di $A \times B$ non gode della proprietà commutativa.</u>
Se risulta B = A, allora si ha il prodotto cartesiano di un insieme per se stesso, in simboli $A \times A$, a volte indicato anche con . A^2

Esempio
Sia $A = \{0, 1\}$ e $B = \{a, b, c\}$. Il prodotto cartesiano $A \times B$ è l'insieme
$A \times B = \{(0, a); (0, b); (0, c); (1, a); (1, b); (1, c)\}$
Se $A = \{a, b, c\}$ allora risulta
$A\times A = \{(a, a); (a, b); (a,c); (b, a); (b, b); (b, c); (c, a); (c, b); (c, c)\}$.

Per il prodotto cartesiano $\mathbf{A \times B}$ valgono le seguenti ***proprietà***:

distributiva rispetto all'unione $\mathbf{A \times (B \cup C) = (A \times C) \cup (A \times C)}$

distributiva rispetto all'intersezione $\mathbf{A \times (B \cap C) = (A \times C) \cap (A \times C)}$

Se $\mathbf{C \subseteq A}$ e $\mathbf{D \subseteq B \Rightarrow C \times D \subseteq A \times B}$

<u>Definizione</u> Una proposizione (enunciato semplice o composto) riferita a due argomenti viene chiamata **relazione binaria**.
Le relazioni binarie determinano **coppie ordinate** di elementi (x, y), di cui il primo appartiene ad un insieme assegnato A e il secondo ad un altro insieme, anch'esso assegnato, B; in tal caso si dirà che "a è in relazione $\mathcal{R}$ con b", in simboli
$x \, \mathcal{R} \, y$ oppure $\mathcal{R}(x, y)$ con $x \in A$ e $y \in B$,
Le coppie ordinate che verificano la relazione assegnata $\mathcal{R}$ appartengono ad un insieme che risulta essere <u>sottoinsieme del prodotto cartesiano</u> A×B.

Esempio
Sia $A = \{2, 4, 7\}$ e $B = \{2, 5, 8\}$ e sia $\mathcal{R}$ = "essere minore di";
$x \, \mathcal{R} y \Leftrightarrow x \leq y$ con $x \in A$ e $y \in B$
Le coppie che verificano la $\mathcal{R}$ costituiscono l'insieme
$\{(2, 5); (2, 8); (4, 5); (4, 8); (7, 8)\} \subset A\times B$ (sottoinsieme del prodotto cartesiano).

Una relazione binaria $\Re$ può godere delle seguenti ***proprietà***:

riflessiva $\Leftrightarrow a \Re a \quad \forall a \in A$ cioè se ogni elemento di **A** è in relazione $\Re$ con se stesso

simmetrica $\Leftrightarrow a \Re b \Rightarrow b \Re a \qquad \forall a.b \in A$

antisimmetrica $\Leftrightarrow a \Re b \;\; e \;\; b \Re a \Rightarrow a = b$
cioè $\forall a.b \in A$ con $a \neq b \;\; a \Re b \Rightarrow b \; non \; \Re \; a$

Transitiva $\Leftrightarrow a \Re b \;\; e \;\; b \Re c \Rightarrow a \Re c \qquad \forall a.b.c \in A$

<u>Definizione</u> Una proposizione riferita ad **n** argomenti viene chiamata **relazione n-aria**.
Le relazioni n-arie determinano **n-ple ordinate** di elementi $(x_1, x_2, \dots, x_n)$, di cui il primo appartiene al primo insieme assegnato A_1, il secondo al secondo insieme assegnato A_2, … l'n-esimo appartiene all'n-esimo insieme assegnato A_n.
In tal caso si dirà che vale $\mathcal{R}(x_1, x_2, \dots, x_n)$ con $x_1 \in A_1, x_2 \in A_2, \dots, x_n \in A_n$

Differenti modi di rappresentare una relazione

Poiché è definita come sottoinsieme del prodotto cartesiano dei suoi domini, una **relazione** (così come accade per le relazioni matematiche) può essere rappresentata.
- per **elencazione;**
- in forma **tabellare;**
- in forma **insiemistica.**

Esempio: consideriamo ancora una volta il seguente schema della relazione Persona utilizzato per rappresentare le caratteristiche di un essere umano:
Persona (CodFisc:Stringa(16), Cognome: Stringa(30), Nome: Stringa(20), Età: Intero, Sesso: Carattere)

Più in dettaglio
a) **rappresentazione per elencazione**: è possibile rappresentare un'istanza di una relazione elencando tutte le sue ennuple (così come si fa per gli elementi di un insieme).

Ritornando al nostro esempio scriveremo:
Persona (<u>CodFisc</u>, Cognome, Nome, Età, Sesso) =

{
(XXXXXXXXXXXXXXXX, Rossi, Paolo, 30, M),
ennupla → ***YYYYYYYYYYYYYYYYY, Bianchi, Luigi, 23, M),***
(ZZZZZZZZZZZZZZZZ, Neri, Daniela, 35, F)
}

istanza composta da tre ennuple con ***Card (Persona) = 3***

b) **rappresentazione in forma insiemistica**: è possibile rappresentare un'istanza di una relazione disegnando una ellisse che racchiude le ennuple (diagramma di Eulero-Venn)

Ritornando al nostro esempio scriveremo:

Persona

(XXX…, Rossi, Paolo, 30, Maschio)
(YYY…, Bianchi, Luigi, 23, Maschio)
(ZZZ…, Neri, Daniela, 35, Femmina)

c) **rappresentazione in forma tabellare** è possibile rappresentare un'istanza di una relazione utilizzando **una tabella di n righe ed m colonne** dove ovviamente m è il **grado** ed n è la **cardinalità** della relazione.
Ogni **riga** rappresenta una **ennupla** ed ogni **colonna** rappresenta la **sequenza dei valori** assunti dal corrispondente attributo.

Ritornando al nostro esempio scriveremo:

Grado (Persona) = 5

Persona	<u>CodFisc</u>	Cognome	Nome	Età	Sesso
	XXX…	Rossi	Paolo	30	M
	YYY…	Bianchi	Luigi	23	M
	ZZZ…	Neri	Daniela	35	F

Card (Persona) = 3

Chiavi di una relazione

Abbiamo appena visto che per esprimere la *struttura* dei dati nel modello relazionale si utilizzano le relazioni in senso matematico.
Nel modello relazionale occorre specificare come vincolo quello relativo alla presenza di una **chiave primaria** per ciascuna relazione.

DEF: Si dice **chiave candidata o superchiave** di una relazione **R** un insieme non vuoto **K** di **attributi** di **R** attraverso i quali è possibile individuare univocamente ogni ennupla per ciascuna possibile istanza della relazione **R**.

N.B. Una *chiave candidata* **esiste sempre** essendo al limite costituita da tutti gli attributi di R considerato il fatto che non possono mai esistere due ennuple uguali.

DEF: Si dice **chiave primaria** di una relazione **R** la **superchiave minimale** ossia quella costituita dal minor numero di attributi tra tutte le diverse possibili chiavi candidate individuate per quella relazione.

N.B. Nello schema di una relazione (analogamente al diagramma ER) si sottolineano gli attributi che costituiscono una chiave.

Esempio: I dati personali dei clienti di un albergo sono trascritti su di un registro che può essere assimilato alla seguente relazione:

Ospite (NumProgressivo, Cognome, Nome, DataNascita, LuogoNascita, TipoDocumento, DataDocumento)

Sono possibili chiavi candidate:
1. (NumProgressivo)
2. (Cognome, Nome,, DataNascita, LuogoNascita)
3. (TipoDocumento, DataDocumento)
4. (NumProgressivo, Cognome, Nome, DataNascita, LuogoNascita, TipoDocumento, DataDocumento)

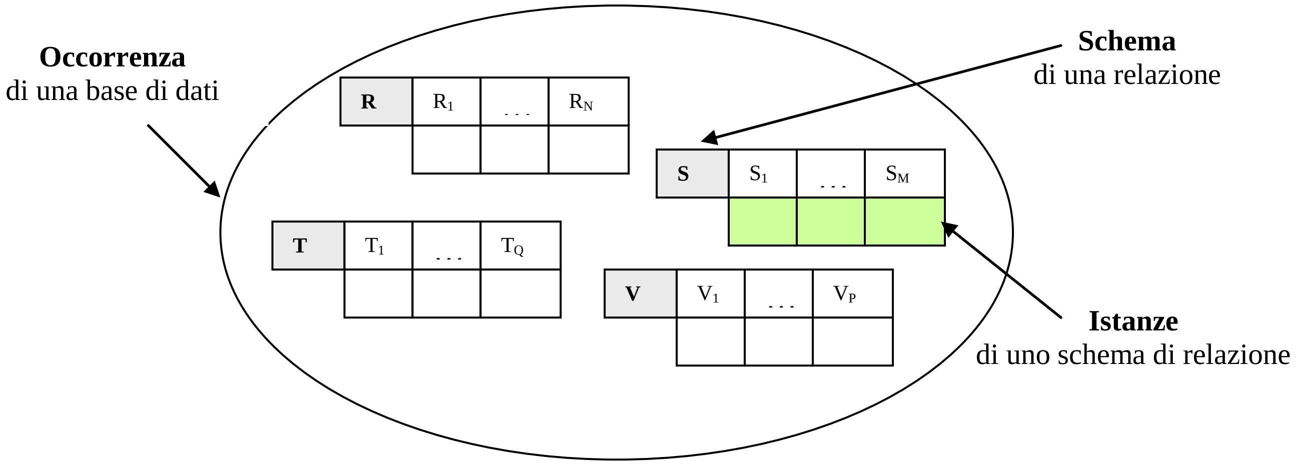
Fra le possibili ***chiavi candidate*** *è stata scelta come* ***chiave primaria (NumProgressivo)*** *perché è costituita dal minor numero di attributi.*

N.B. Spesso nella progettazione di basi dati quando non è possibile utilizzare un numero limitato di attributi come chiave primaria, si ricorre all'aggiunta di un nuovo attributo (come ad esempio un progressivo univoco) in grado di identificare univocamente le ennuple.

Schema ed occorrenza di una base di dati

DEF: Si definisce **schema di una base di dati relazionale** l'insieme di tutti gli *schemi* di relazione. Quindi si definisce **occorrenza o istanza di una base di dati relazionale** l'insieme delle *istanze* degli schemi di tutte le relazioni.

Possibile rappresentazione di uno schema di base di dati in un determinato istante di tempo:



I **legami** tra le relazioni si realizzano utilizzando le loro **chiavi**.

N.B. In seguito vedremo come rappresentare nel modello relazionale tali *legami* che corrispondono alle *associazioni* del *diagramma ER*.

Vincoli di integrità

Sappiamo che una base di dati è un *insieme di relazioni* che varia nel tempo e che è soggetto a continue modifiche, cancellazioni ed inserimenti di nuovi dati.
Abbiamo già detto che tali operazioni su una base di dati devono essere eseguite rispettando un **insieme di regole** che servono per mantenere l'integrità dei dati.

Quindi per specificare il fatto che alcune relazioni sono corrette dal punto di vista di chi sviluppa l'applicazione ed altre non lo sono, viene introdotto il concetto di **vincolo di integrità.**

DEF: Si definisce **vincolo di integrità (**nel modello relazionale) una **proprietà** che deve essere soddisfatta da tutte le istanze di una o più relazioni affinchè le informazioni contenute nella base dati restino corrette e significative per qualsiasi utente/applicazione le utilizzi..

*Esempio: consideriamo la relazione **Studente** che contiene informazioni anagrafiche relative ad uno studente espressa in forma tabellare:*

Studente	**Matricola**	**Cognome**	**Nome**	**Età**	**DataNascita**	**DataIscrizione**
	2345	*Rossi*	*Gianni*	*19*	***01/01/1987***	***04/09/1986***
	4667	*Neri*	*Alfonso*	***350***	*08/07/1971*	*05/09/2006*
	4667	*Bianchi*	*Adele*	*20*	*12/08/1988*	*06/09/2006*

Oss. 1: *L'attributo Età del nominativo "Neri Alfonso" ha un valore pari a 350 che pur essendo corretta dal punto di vista del tipo del dominio intero (350 è un numero intero), rappresenta un dato inverosimile per un età reale;*
Oss. 2: *Nella relazione Studente vi sono due ennuple che hanno lo stesso valore per l'attributo Matricola che pur essendo del tutto lecito dal punto di vista strutturale, porta ad una incongruenza nella rappresentazione della realtà che si vuole modellare in quanto non è possibile che due studenti abbiano lo stesso numero di matricola.*

Oss. 3: *Nella relazione Studente vi è una ennupla che ha nel campo DataNascita e DataIscrizione due valori del tutto leciti dal punto di vista del tipo del dominio data ma che portano ad una incongruenza nella rappresentazione della realtà che si vuole modellare in quanto non è possibile che per un dato studente abbia data di nascita successiva alla data di iscrizione.*

Conseguenze*: in questo caso dovremmo poter specificare:*
- un vincolo di integrità che ci assicuri che l'età sia un numero inferiore a 120;
- un vincolo di integrità che affermi che non possano esserci due studenti con lo stesso numero di matricola;
- un vincolo di integrità che affermi che per ciascuno studente occorre controllare che la data di iscrizione sia successiva alla data di nascita.

Alla luce dell'esempio fatto possiamo considerare ogni **vincolo** come **un'asserzione** (ricorda la preposizione dell'algebra di **Boole**) che può essere rispetto ad una certa istanza solamente vera oppure falsa.
Ciò significa che riguardo a tutte le varie istanze possibili sulla base dei *domini*, degli *schemi di relazione* e degli *schemi di base di dati* verranno considerate **accettabili** esclusivamente quelle istanze di base di dati per i quali i vincoli di integrità risultino **veri** (ossia possono verificarsi nella realtà di interesse)
Se arriva un dato che non rispetta un vincolo di integrità, diremo che quel dato (quell'istanza) **viola** un vincolo qualsiasi di integrità.

I **vincoli di integrità del modello relazionale** possono essere classificati in:

- **vincoli di integrità intrarelazionali** o **interni**: sono quei vincoli di integrità definiti all'interno di una stessa relazione. Possono essere a loro volta suddivisi in:

  - vincoli **su singola ennupla**: che esprimono una condizione
    - sul **dominio di un solo attributo**: sono quei vincoli di integrità che coinvolgono i valori assunti da un solo attributo il cui soddisfacimento può essere verificato facendo riferimento ad un singolo valore alla volta;
    - sui **domini di più attributi**: sono quei vincoli di integrità che coinvolgono i valori assunti tra più attributi ma sempre di ciascuna ennupla indipendentemente dalle altre ennuple.

  N.B. Possono rientrare qui alcuni vincoli espliciti trovati in fase di progettazione concettuale.

  - vincoli **su più ennuple**: sono quei vincoli di integrità che coinvolgono i valori di più ennuple. Rientrano in questo tipo i *vincoli (impliciti) di chiave primaria* (ossia le ennuple di una stessa relazione devono essere tutte diverse tra loro).

- **vincoli di integrità interrelazionali** o **esterni**: sono quei vincoli di integrità definiti tra più relazioni. Possono essere a loro volta suddivisi in:
  - vincoli **referenziali** Rientrano in questo tipo quelli (impliciti) trovati in fase di progettazione concettuale che esprimono la totalità di un'associazione ma anche quelli appositamente creati per effetuare il mapping relazionale di un'associazione di molteplicità N:N

  - vincoli **non referenziali:** Possono rientrare alcuni vincoli espliciti trovati in fase di progettazione concettuale.

N.B. Come è possibile riscontrare sono stati considerati in questo modo tutti i vincoli *impliciti* ed *espliciti* che abbiamo descritto nel *diagramma ER*.

Nell'esempio mostrato in precedenza:
OSS 1: *il vincolo* ***Età < 120*** *è un vincolo di integrità intrarelazionale o interno, su singola ennupla che esprime una condizione su di un unico attributo*
OSS 2: *il vincolo che afferma che uno studente* ***non può avere lo stesso numero di matricola*** *di un altro studente è un vincolo di integrità intrarelazionale o interno, su più ennuple (relativo al vincolo implicito vincolo di chiave primaria)*
OSS 3: *il vincolo che afferma che per qualsiasi studente* ***DataIscrizione > DataNascita*** *è un vincolo di integrità intrarelazionale o interno, su singola ennupla che esprime una condizione su più attributi*

Rappresentazione dei Vincoli di integrità

Per rappresentare i **vincoli di chiave primaria** sottolineiamo i relativi attributi (così come abbiamo fatto nel diagramma ER).
Per rappresentare i **gli altri vincoli** utilizziamo un nostro pseudolinguagio (così come abbiamo fatto nel diagramma ER).

a) **vincoli di integrità intrarelazionali o interni**

a1) **V <NumProgr> (<NomeRelazione>) : (<Espressione>)**

dove:
- **<NumProgr>** è il numero progressivo del vincolo relativo alla relazione <NomeRelazione>.
N.B. Sarà ovviamente o stesso del vincolo esplicito indicato nel diagramma ER;

- **<Espressione>** è una qualsiasi espressione in pseudolinguaggio naturale che serve a specificare il vincolo.

Esempio:
V1(Dipendente): (Dipendente.StipendioLordo > 0)
V2(Dipendente): (Dipendente.DataAssunzione> Data Nascita)
V3(Dipendente): (Dipendente.Trattenute> 0)

b) **vincoli di integrità interrelazionali o esterni**

b.1) **i vincoli di integrità interrelazionali referenziali**: per rappresentare i vincoli di integrità referenziale utilizzeremo la seguente sintassi:

$$\mathbf{VR}_{<Attributo1>}\ \mathbf{(<Relazione1>)} \subseteq \mathbf{VR}_{<Attributo2>}\ \mathbf{(<Relazione2>)}$$

con il significato che tutti i valori dell'attributo <Attributo1> presenti nelle ennuple della relazione <Relazione1> devono essere presenti tra i valori dell'attributo <Attributo2> delle ennuple della relazione <Relazione2>
N.B. Ricordiamo che per convenzione <Attributo1> ed <Attributo2> avranno lo stesso nome.

Esempio:

$$\mathbf{VR}_{CodForn}\ \mathbf{(Fornisce)} \subseteq \mathbf{VR}_{CodForn>}\ \mathbf{(Fornitore)}$$

Significa che tutti i valori dell'attributo ***CodForn*** *presenti nelle ennuple della relazione* ***Fornisce*** *devono anche essere presenti tra i valori dell'attributo* ***CodForn*** *delle ennuple della relazione* ***Fornitore****.*
In altre parole nella relazione ***Fornisce*** *non possono essere presenti codici di fornitori non presenti nella relazione* ***Fornitore****.*

VR $_{\text{CodArt}}$ (Fornisce) ⊆ VR $_{\text{CodArt}}$ (Articolo)

*Significa che tutti i valori dell'attributo **CodArt** presenti nelle ennuple della relazione **Fornisce** devono anche essere presenti tra i valori dell'attributo **CodArt** delle ennuple della relazione **Articolo**.*
*In altre parole nella relazione **Fornisce** non possono essere presenti codici di articoli non presenti nella relazione **Articolo**.*

b.2) **i vincoli di integrità interrelazionali NON referenziali**: per rappresentare i vincoli di integrità interrelazionali che non siano referenziali utilizzeremo la seguente sintassi:

V <NumProgr> (<Relazione1>, … <RelazioneN>) : (<Espressione>)

dove:
- **<NumProgr>** è il numero progressivo del vincolo;
N.B. Sarà ovviamente o stesso del vincolo esplicito indicato nel diagramma ER;
- **<Relazione1>, …, <RelazioneN>** sono i nomi delle relazioni che sono legate da vincoli esterni;
- **<Espressione>** è una qualsiasi espressione in pseudolinguaggio naturale che serve a specificare il vincolo.

Esempio: consideriamo le seguenti relazioni

Azienda** **(CodAzienda, RagioneSociale, CodAttività, StipendioLordoMedio, MatricolaDip)

Dipendente** **(Matricola, Nominativo, DataAssunzione, Livello, StipendioLordo, Trattenute)

Per esprimere il vincolo che impone che lo stipendio lordo di un dipendente di ottavo livello sia maggiore dello stipendio medio dell'azienda cui appartiene scriviamo

***V1** (**Dipendente**, **Azienda**) : (**SE Dipendente**.Livello ≥ 8 **ALLORA Dipendente**.StipendioLordo > **Azienda**.StipendioLordoMedio)*

N.B. Per riferirci all'attributo di una certa relazione utilizziamo la seguente ***dot-notation***

<NomeRelazione>.<NomeAttributo>

DAL DIAGRAMMA ER ALLO SCHEMA RELAZIONALE

Il modello relazionale mette a disposizione del progettista solo *le relazioni* per modellare i vari aspetti della realtà.
Partendo dal **diagramma ER** (output della fase della progettazione concettuale) il progettista deve effettuare un "mapping" delle *entità* e delle *associazioni* individuate, trasformandole opportunamente in *relazioni* del *modello relazionale* (output della fase di progettazione logica) che tutte insieme costituiranno il cosiddetto **schema relazionale**.

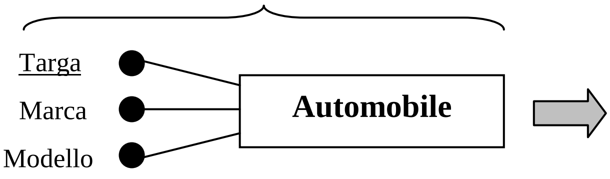
Lo **schema relazionale** si ricava dunque dal diagramma ER applicando alcune semplici **regole di derivazione** per rappresentarne tutti gli oggetti ossia:
- **entità** ed **attributi**;
- **associazioni di tipo 1:N (oppure N:1)**;
- **associazioni di tipo 1:1**;
- **associazioni di tipo N:N**;
- **generalizzazioni**;

Rappresentazione delle entità e degli attributi

Una entità **E** con **attributi elementari** $A_1, A_2, \dots, A_N$ del **diagramma ER** è immediatamente rappresentata attraverso una relazione **R** ($A_1, A_2, \dots, A_N$) dello **schema relazionale** dove:
- ogni **entità** diventa una **relazione** rappresentata mediante **tabella**;
- ogni **attributo** *dell'entità* diventa un **attributo** della *relazione* rappresentato mediante una **colonna** della *tabella*;
- l'**attributo chiave** dell'*entità* diventa **attributo chiave** della *relazione* rappresentato dai **campi chiave** nella *tabella*.

Esempio

Diagramma ER



Entità con attributi elementari

Schema relazionale

Automobile (Targa, Marca, Modello)

Relazione

rappresentabile graficamente in uno dei tre modi descritti ad esempio in forma **tabellare**

Automobile	Targa	Marca	Modello

Eventuale **attributi composti** vengono sostituiti con gli **attributi elementari componenti.**

*Esempio: L'attributo composto **Indirizzo** si sostituisce con gli attributi elementari componenti **Via**, **Città**, **Cap***

N.B. Per quanto riguarda eventuali **attributi multipli** si procederà alla **normalizzazione** della relazione le cui modalità operative verranno presentate in seguito.

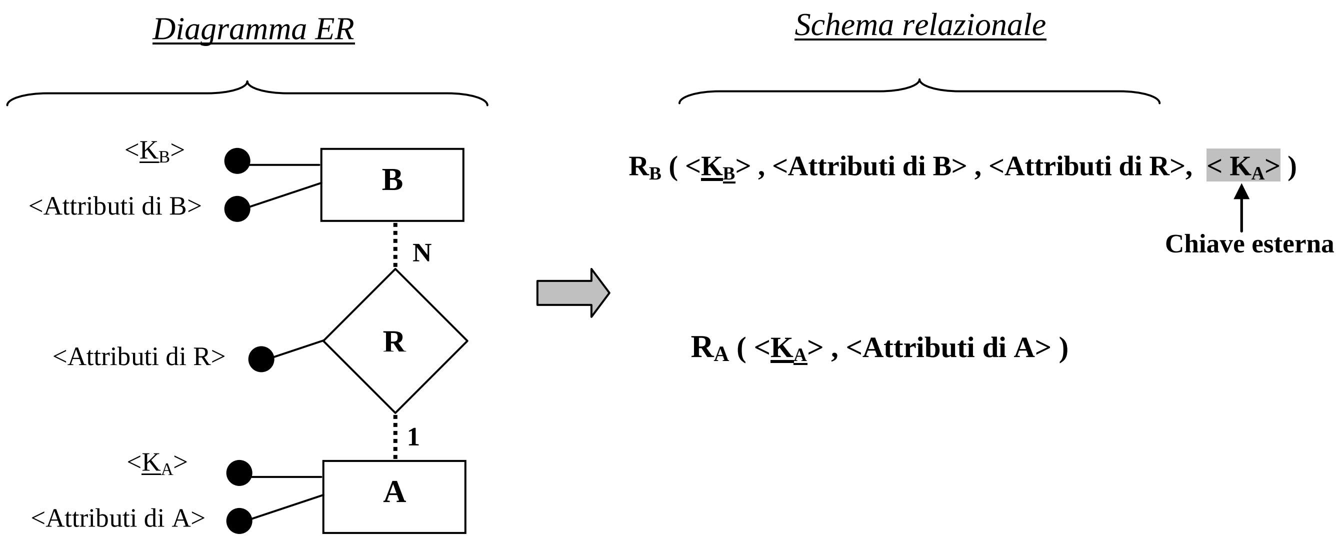
Rappresentazione delle associazioni binarie 1:N (oppure N:1)

Sia data una associazione **R** di tipo **1:N** tra due entità **A** e **B** con diretta ed inversa entrambe parziali.
Per “mappare” tale associazione nel modello relazionale occorre introdurre **due relazioni** costituite nel seguente modo:
- una relazione $\mathbf{R_A}$ avente tutti gli attributi di **A**;
- una relazione $\mathbf{R_B}$ avente tutti gli attributi di **B**, gli attributi di **R** e gli attributi chiave $\mathbf{K_A}$ di **A**.

Nella relazione $\mathbf{R_B}$ occorrerà dunque inserire, oltre agli attributi eventuali dell’associazione gli attributi chiave di **A** che costituiscono una cosiddetta **chiave esterna** per la relazione $\mathbf{R_B}$.
N.B. Il valore di una chiave esterna rappresenta un puntatore logico alla ennupla della relazione esterna.

N.B. Le chiavi esterne non devono essere sottolineate poiché non fanno parte della chiave primaria della relazione

MAPPING Associazione binaria 1:N con associazione diretta (da A ad B) ed inversa (da B a A) PARZIALI



OSSERVAZIONE: Per trasformare un diagramma ER con associazione diretta totale (linea continua da A al rombo dell’associazione) occorrerà specificare, oltre le due relazioni, **un vincolo di integrità referenziale** che forzi l’esistenza nella relazione R_B di una chiave esterna uguale alla chiave primaria K_A.
In altre parole il vincolo di integrità da imporre è che.

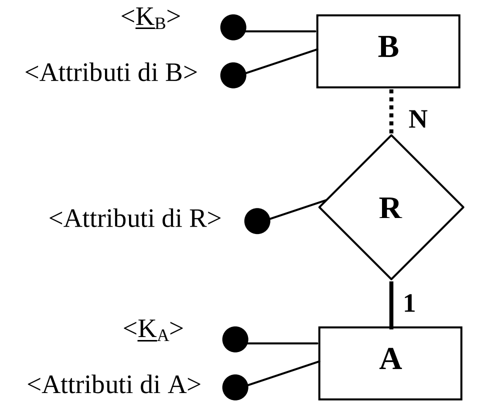
$$\mathbf{VR_{KA}\,(R_A) \subseteq VR_{KA}\,(R_B)}$$

Esso significa che tutti i valori dell’attributo chiave primaria K_A di R_A devono essere presenti nell’attributo chiave esterna K_A di R_B.

In altre parole possiamo dire che **“non si può inserire una ennupla nella relazione R_A se non si inserisce contemporaneamente la sua correlazione con qualche elemento della relazione R_B valorizzando opportunamente la chiave esterna“**

MAPPING Associazione binaria 1:N con associazione diretta (da A ad B) TOTALE ed inversa (da B a A) PARZIALE

Diagramma ER | *Schema relazionale*



R_B (<$\underline{K_B}$> , <Attributi di B> , <Attributi di R>, < K_A>)

Chiave esterna

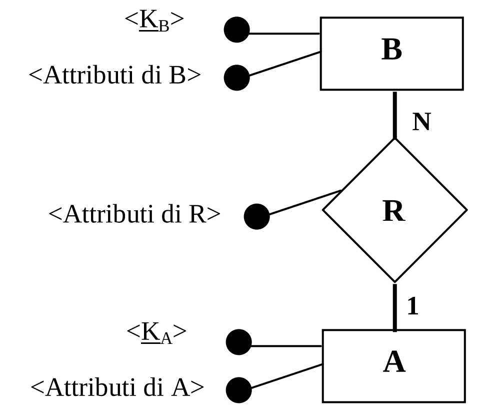
R_A (<$\underline{K_A}$> , <Attributi di A>)

Vincolo di integrità referenziale per rappresentare la totalità della associazione diretta

$$VR_{KA}\,(R_A) \subseteq VR_{KA}\,(R_B)$$

MAPPING Associazione binaria 1:N con associazione diretta (da A ad B) ed inversa (da B a A) TOTALI

Diagramma ER | *Schema relazionale*



R_B (<$\underline{K_B}$> , <Attributi di B> , <Attributi di R>, < K_A>)

Chiave esterna

R_A (<$\underline{K_A}$> , <Attributi di A>)

Vincoli di integrità referenziale per rappresentare la totalità delle associazioni diretta ed inversa

$$VR_{KA}\,(R_A) \subseteq VR_{KA}\,(R_B)$$
$$VR_{KA}\,(R_B) \subseteq VR_{KA}\,(R_A)$$

In questo caso deve essere imposto anche il seguente vincolo referenziale

$$VR_{KA}\,(R_B) \subseteq VR_{KA}\,(R_A)$$

che assicura la totalità dell'associazione inversa.

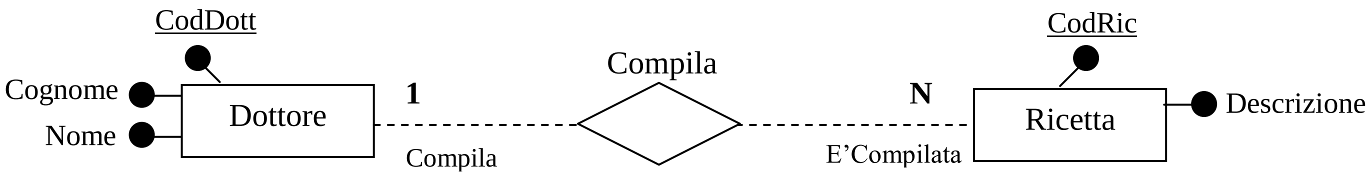
Esso significa che tutti i valori dell'attributo chiave esterna K_A di R_B devono essere presenti nell'attributo chiave primaria K_A di R_A

In altre parole possiamo dire che **"non si può inserire una ennupla nella relazione R_B senza che non venga messa in correlazione con almeno una ennupla della relazione R_A valorizzando opportunamente la chiave esterna".**

N.B. per le associazioni N:1 si segue esattamente quanto visto finora scambiando semplicemente i ruoli tra le due relazioni R_A e R_B

Esempio: Supponiamo che un certo Dottore può compilare nessuna o più ricette e che viceversa una ricetta può essere compilata da nessuno o un solo Dottore (essendo possibile che la segretaria le compili per lui)

Utilizzando il diagramma ER la situazione è la seguente



Operando il mapping come visto si ha:

***Dottore** (<u>CodDott</u>, Cognome, Nome)*

Chiave primaria

Dottore	**CodDott**	**Cognome**	**Nome**
	D01	ROSSI	MARIO
	D02	BIANCHI	GIULIO
	D03	VERDI	LUIGI
	D04	**NERI**	**DANIELA**

Chiave primaria

Chiave esterna su relazione Dottore

***Ricetta** (<u>CodRic</u>, Descrizione, CodDott)*

Ricetta	**CodRic**	**Descrizione**	**CodDott**
	R01	Ves	D01
	R02	Radiografia	D02
	R03	Sangue	D03
	R04	**Urina**	**NULL**
	R05	Tiroide	D02

*Per mantenere la correlazione tra Dottore e Ricetta secondo il mapping dobbiamo aggiungere alla relazione Ricetta – quella lato N - **la chiave esterna** della relazione Dottore.*
La parzialità dell'associazione diretta "Compila" si esplica nel fatto che vi è il Dottore con CodDott = D04 che non ha compilato alcuna ricetta (quindi non partecipa alla relazione).
La parzialità dell'associazione inversa "E'Compilata" si esplica nel fatto che vi è la Ricetta con CodRic = R04 che non è stata compilata da alcun dottore (valore NULL della chiave esterna) (quindi non partecipa alla relazione).

Per convenzione nella nuova relazione per gli attributi chiavi esterne è meglio utilizzare <u>lo stesso nome</u> dell'attributo relativo alla chiave primaria

N.B. NON DEVONO ESSERE IMPOSTI VINCOLI DI INTEGRITA' REFERENZIALI

*In questo esempio si ha una inconsistenza dei dati se dalla relazione **Dottore** si cancella l'istanza relativa al dottore **D01**. Infatti in tal caso nella relazione **Ricetta** avremo ben tre ricette che farebbero riferimento ad un'istanza non più esistente all'interno del database.*

*Per assicurare **l'integrità referenziale** prima di cancellare una qualsiasi ennupla, occorre verificare che non vi siano ennuple in altre relazioni che facciano riferimento alla ennupla da cancellare.*

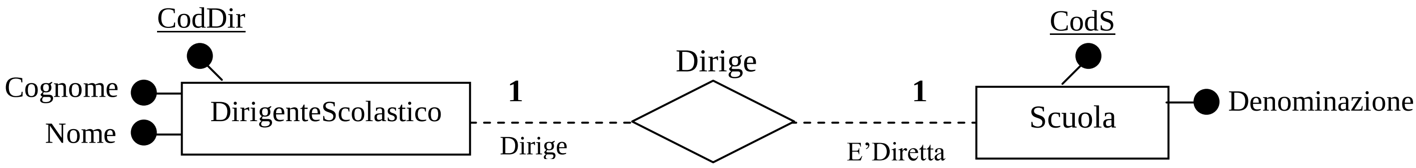
Rappresentazione delle associazioni binarie 1:1

Le **associazioni binarie 1:1** sono un caso particolare delle *associazioni 1:N* e quindi seguono le stesse regole viste finora.
Il **vincolo di integrità nelle associazioni 1:1** significa che *solo una chiave esterna deve corrispondere alla chiave primaria* ossia

$$\mathbf{VR_{KA}\,(R_B) = VR_{KA}\,(R_A)}$$

Si tende a trasformare a volte le due entità con associazione binaria 1:1 in un'unica relazione che si ottiene dalla fusione delle due e che possiede gli attributi dell'una e dell'altra (tabellone unico)
Altre volte si conservano le entità in relazioni separate per motivi di efficienza (infatti se si accede ad una entità più frequentemente dell'altra conviene avere relazioni più snelle ossia con il minor numero di attributi possibile).

Esempio: Supponiamo che un dirigente scolastico deve dirigere una sola scuola e che viceversa una scuola deve essere diretta da un solo dirigente scolastico.
Utilizzando il diagramma ER la situazione è la seguente



Sono possibili i seguenti mapping:

a) Relazioni separate

***DirigenteScolastico** (<u>CodDir</u>, Cognome, Nome)*

Chiave primaria

DirigenteScolastico	CodDir	Cognome	Nome
	D01	SESSA	DARIO
	D02	BIANCHI	GIULIO

***Scuola** (<u>CodS</u>, Denominazione, CodDir)*

Chiave primaria

Chiave esterna su relazione DS

Scuola	CodS	Denominazione	CodDir
	S01	ITI "GALVANI"	D02
	S02	ISIS"TASSINARI"	D01

Vi sono anche i seguenti due vincoli di integrità referenziali

(1) $\mathbf{VR_{CodDir}(DirigenteScolastico) \subseteq VR_{CodDir}\,(Scuola)}$ **TOTALITA' dell'a. diretta "Dirige"**

(2) $\mathbf{VR_{CodDir}\,(Scuola) \subseteq VR_{CodDir}\,(DirigenteScolastico)}$ **TOTALITA' dell'a. inversa "E'Diretta"**

b) Unica relazione

***DirigenteScolastico-Scuola** (<u>CodDir</u>, Cognome, Nome, CodS, Denominazione)*

Chiave primaria

N.B. Questo attributo Potrebbe essere omesso

Dirigente Scolastico Scuola	CodDir	Cognome	Nome	CodS	Denominazione
	D01	SESSA	DARIO	S02	ISIS"TASSINARI"
	D02	BIANCHI	GIULIO	S01	ITI "GALVANI"

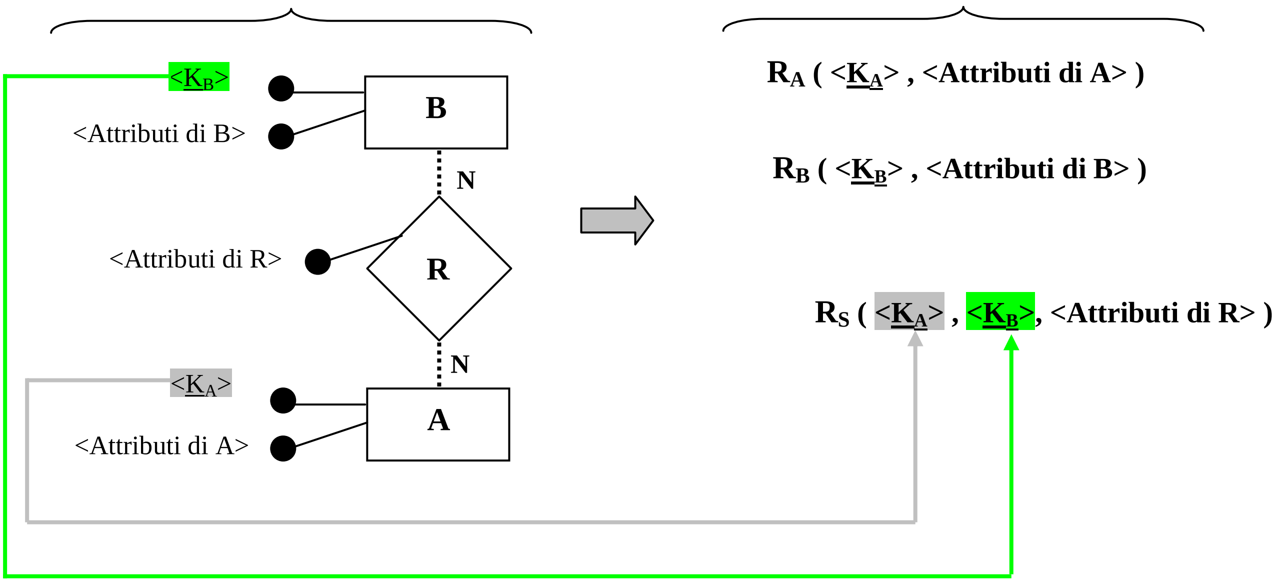
Rappresentazione delle associazioni binarie N:N

Sia data una associazione **R** di tipo **N:N** tra due entità **A** e **B** con diretta ed inversa entrambe parziali.
Per "mappare" tale associazione nel modello relazionale occorre introdurre **tre relazioni** e **due vincolo di integrità referenziale** così costituiti:
- una relazione $\mathbf{R_A}$ avente gli attributi di **A**;
- una relazione $\mathbf{R_B}$ avente gli attributi di **B**;
- una relazione $\mathbf{R_S}$ avente gli attributi chiave $\mathbf{K_A}$ di $\mathbf{R_A}$ e gli attributi chiave $\mathbf{K_B}$ di $\mathbf{R_B}$ come chiave primaria(quindi come minimo la relazione $\mathbf{R_S}$ avrà due attributi);
- un vincolo di integrità che assicuri che ad ogni chiave esterna $\mathbf{K_A}$ presente in $\mathbf{R_S}$ corrisponda una chiave primaria $\mathbf{K_A}$ della relazione $\mathbf{R_A}$;
- un vincolo di integrità che assicuri che ad ogni chiave esterna $\mathbf{K_B}$ presente in $\mathbf{R_S}$ corrisponda una chiave primaria $\mathbf{K_B}$ della relazione $\mathbf{R_B}$.

Quindi ogni ennupla di $\mathbf{R_S}$ rappresenta una coppia dell'associazione binaria **S**.

MAPPING Associazione binaria N:N con associazione diretta (da A ad B) ed inversa (da B a A) PARZIALI

Diagramma ER *Schema relazionale*



Vincoli di integrità referenziale per esprimere che ogni chiave esterna in $\mathbf{R_S}$ corrisponde una chiave primaria rispettivamente in $\mathbf{R_A}$ ed in $\mathbf{R_B}$

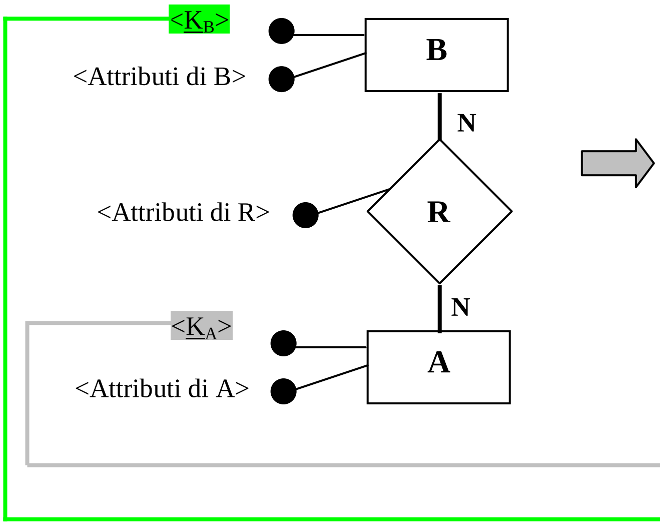
(1) $\mathbf{VR_{KA}(R_S) \subseteq VR_{KA}(R_A)}$

(2) $\mathbf{VR_{KB}(R_S) \subseteq VR_{KB}(R_B)}$

N.B. Occorre notare che le chiavi esterne K_A e K_B della relazione R_S sono sottolineate in quanto sono anche chiavi interne della relazione R_S

MAPPING Associazione binaria N:N con associazione diretta (da A ad B) ed inversa (da B a A) TOTALI

Diagramma ER | *Schema relazionale*



$\mathbf{R_A\ (\ \langle \underline{K_A} \rangle\ ,\ \langle \text{Attributi di A} \rangle\ )}$

$\mathbf{R_B\ (\ \langle \underline{K_B} \rangle\ ,\ \langle \text{Attributi di B} \rangle\ )}$

$\mathbf{R_S\ (\ \langle \underline{K_A} \rangle\ ,\ \langle \underline{K_B} \rangle\ ,\ \langle \text{Attributi di R} \rangle)}$

Vincoli di integrità referenziale per esprimere che ogni chiave esterna in R_S corrisponde una chiave primaria rispettivamente in R_A ed in R_B

$$\textbf{(1)}\quad VR_{KA}(R_S) \subseteq VR_{KA}(R_A)$$

$$\textbf{(2)}\quad VR_{KB}(R_S) \subseteq VR_{KB}(R_B)$$

Vincoli di integrità referenziale sulla TOTALITA' della diretta (da A verso B)

$$\textbf{(3)}\quad VR_{KA}(R_A) \subseteq VR_{KA}(R_S)$$

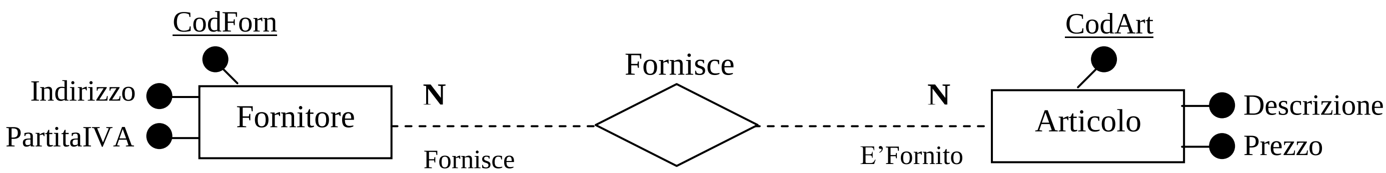
Vincoli di integrità referenziale sulla TOTALITA' della inversa (da B verso A)

$$\textbf{(4)}\quad VR_{KB}(R_B) \subseteq VR_{KB}(R_S)$$

Il vincolo di integrità referenziale **(3)** dice che ad ogni ennupla presente in $\mathbf{R_A}$ deve corrispondere una ennupla in $\mathbf{R_S}$.
Il vincolo di integrità referenziale **(4)** dice che ad ogni ennupla presente in $\mathbf{R_B}$ deve corrispondere una ennupla in $\mathbf{R_S}$.

Esempio: Supponiamo che un certo fornitore deve fornire uno o più articoli e che viceversa un certo articolo deve essere fornito da uno o più fornitori.

Utilizzando il diagramma ER la situazione è la seguente



Operando il mapping come visto si ha:

Fornitore (<u>CodForn,</u> Indirizzo, PartitaIVA)

Chiave primaria

Fornitore	CodForn	Indirizzo	PartitaIVA
	F03	Via Po, 5	001234
	F07	Via Bari, 5	001345
	F16	Via Loi, 1	001333

Articolo (<u>CodArt,</u> Descrizione, Prezzo)

Chiave primaria

Articolo	CodArt	Descrizione	Prezzo
	A01	Batteria	100,00
	A04	Antenna	75,00
	A12	Radiatore	56,00

*Per mantenere la correlazione tra Fornitore ed Articolo secondo il mapping dobbiamo creare una nuova relazione che chiameremo **Fornisce** utilizzando le chiavi primarie delle due relazioni **Fornitore** ed **Articolo** che diventano **INSIEME chiavi primarie** della nuova relazione e sono **chiavi esterne** ciascuna su una delle due relazioni di partenza.*

Fornisce (<u>CodForn</u>, <u>CodArt</u>)

Chiave esterna su Fornitore

Chiave esterna su Articolo

Fornisce	CodForn	CodArt
	F03	A01
	F03	A04
	F16	A04
	F07	A12

Per convenzione nella nuova relazione per gli attributi chiavi esterne è meglio utilizzare <u>lo stesso nome</u> dell'attributo relativo alla chiave primaria

Vincoli di integrità referenziale per esprimere che ogni chiave esterna della relazione "Fornisce" corrisponde ad una chiave primaria rispettivamente della relazione "Fornitore"e e della relazione "Articolo"

(1) $\mathbf{VR_{CodForn}}$ **(Fornisce)** $\subseteq$ $\mathbf{VR_{CodForn}}$ **(Fornitore)**

(2) $\mathbf{VR_{CodArt}}$ **(Fornisce)** $\subseteq$ $\mathbf{VR_{CodArt}}$ **(Articolo)**

Vincoli di integrità referenziale per esprimere la TOTALITA' dell'associazione diretta "Fornisce"

(3) $\mathbf{VR_{CodForn}}$ **(Fornitore)**$\subseteq$ $\mathbf{VR_{CodForn}}$ **(Fornisce)**

Vincoli di integrità referenziale per esprimere la TOTALITA' dell'associazione inversa "E'Fornito"

(4) $\mathbf{VR_{CodArt}}$ **(Articolo)**$\subseteq$ $\mathbf{VR_{CodArt}}$ **(Fornisce)**

*In questo esempio si ha una inconsistenza dei dati se dalla relazione **Fornitore** si cancella l'istanza relativa al fornitore **F03**. Infatti in tal caso nella relazione **Fornisce** avremo la chiave esterna F03 alla quale non corrisponde alcun fornitore.*
*Stesso discorso se si cancella ad esempio dalla relazione **Articolo** l'istanza relativa all'articolo **A01**: avremmo così nella relazione **Fornisce** la chiave esterna A01 alla quale non corrisponde alcun articolo.*

*Per assicurare **l'integrità referenziale** prima di cancellare una qualsiasi ennupla, occorre verificare che non vi siano ennuple in altre relazioni che facciano riferimento alla ennupla da cancellare.*

N.B. Nel modello relazionale l'integrità referenziale è assicurata direttamente dal **DBMS** che prevede la possibilità di dichiarare attraverso appositi linguaggi dichiarativi delle **regole di validazione**. Tali regole vengono conservate in appositi archivi detti **cataloghi delle regole**.

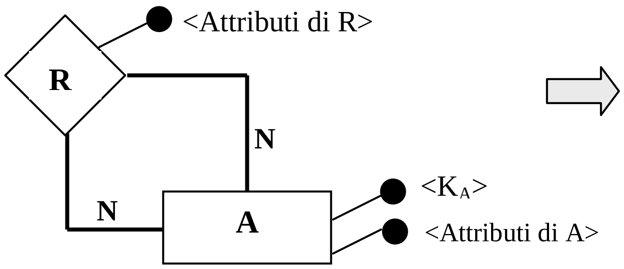
Rappresentazione delle associazioni su di una stessa entità (associazioni binarie ricorsive).

Un caso particolare di associazioni 1:N oppure N:N è quello in cui l'entità di partenza è uguale a quella di arrivo.
In entrambi i casi vanno utilizzati, con apposite modificazioni, le regole di rappresentazione illustrate nel caso delle associazioni binarie di tipo 1:N oppure N:N.

La chiave $<\underline{K_A}>$ comparirà in due colonne $<\underline{K1_A}>$ ed $<\underline{K2_A}>$ della tabella che rappresenta la relazione S.

MAPPING Associazione binaria N:N con associazione diretta ed inversa TOTALE

Diagramma ER



Schema relazionale

$\mathbf{R_A\ (<\underline{K_A}>\ ,\ <\text{Attributi di A}>\ )}$

$\mathbf{R_S\ (<\underline{K1_A}>\ ,\ <\underline{K2_A}>\ ,\ <\text{Attributi di R}>\ )}$

Vincoli di integrità referenziale

(1) $VR_{K1A}(R_S) \subseteq VR_{K1A}(R_A)$

(2) $VR_{K2A}(R_S) \subseteq VR_{K2A}(R_A)$

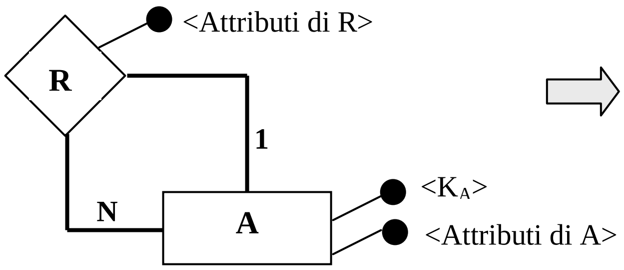
Vincoli di integrità referenziale per laTOTALITA'

(3) $VR_{KA}(R_A) \subseteq VR_{K1A}(R_S)$

(4) $VR_{KA}(R_A) \subseteq VR_{K2A}(R_S)$

MAPPING Associazione binaria 1:N con associazione diretta ed inversa TOTALE

Diagramma ER



Schema relazionale

$\mathbf{R_A\ (<\underline{K1_A}>\ ,\ <\text{Attributi di A}>\ ,\ <\text{Attributi di R}>\ ,\ <K2_A>)}$

Vincoli di integrità referenziale

$VR_{K1A}(R_A) \subseteq VR_{K2A}(R_A)$

$VR_{K2A}(R_A) \subseteq VR_{K1A}(R_A)$

Rappresentazione delle associazioni di generalizzazione

Per rappresentare una associazione per generalizzazione nel modello relazionale possiamo seguire tre strade diverse:
- **accorpamento** delle entità figlie nell'entità padre (**Valida qualunque sia l'ISA**);
- **accorpamento** dell'entità padre nelle entità figlie (**Valida solo se ISA TOTALE**) ;
- **sostituzione** della generalizzazione con associazioni binarie di tipo 1:1 (**Valida solo se ISA ESCLUSIVA**);

Nell'**accorpamento delle figlie nel padre**: le entità $\mathbf{S_1, S_2, \ldots, S_N}$ vengono eliminate ed i loro attributi e le associazioni cui partecipano, vengono aggiunti all'entità padre **G**.
Inoltre all'entità padre viene aggiunto un altro attributo che serve per distinguere **il tipo di ogni ennupla** del padre ossia per distinguere se ciascuna ennupla appartiene a $\mathbf{S_1, S_2, \ldots, S_N}$

OSS: *Tale traduzione conviene quando le operazioni sulla base dati non fanno molta distinzione tra ennuple di una figlia o di un'altra e tra gli attributi di una figlia ed un'altra.*
In questo caso avremmo minimizzato gli accessi alla memoria anche se con uno spreco maggiore della stessa visto che vi saranno valori nulli per alcuni attributi di volta in volta

N.B.: Tale traduzione è possibile solo quando la generalizzazione è **TOTALE** ossia se ogni ennupla di G è una ennupla di almeno figlia tra $\mathbf{S_1}$ o di $\mathbf{S_2}$ o di $\mathbf{S_N}$

Nell'**accorpamento del padre nelle figlie**: l'entità padre **G** viene eliminata ed i suoi attributi e le associazioni alle quali il padre partecipa, vengono aggiunti alle figlie $\mathbf{S_1, S_2, \ldots, S_N}$
Viene inoltre definito un **vincolo di integrità referenziale** che dica *"ogni ennupla di G corrisponde ad almeno una delle entità figlie"* (totalità della generalizzazione).

OSS: *Tale traduzione conviene quando ci sono operazioni sulla base dati che coinvolgono solo ennuple di una figlia o solo ennuple di un'altra figlia, distinguendo tra le entità figlie.*

N.B.: Tale traduzione è possibile solo quando la generalizzazione è **ESCLUSIVA** ossia se ogni ennupla di G è una ennupla di al massimo una figlia tra $\mathbf{S_1}$ o di $\mathbf{S_2}$ o di $\mathbf{S_N}$

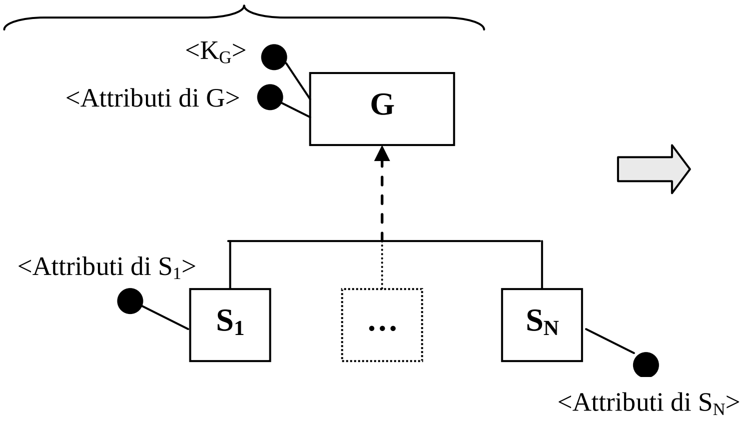
Nella **sostituzione della generalizzazione con associazioni 1:1**: in questa traduzione si trasforma la generalizzazione in tante associazioni di tipo 1.1 quante sono le entità figlie.

Le entità figlie $\mathbf{S_1, S_2, \ldots, S_N}$ sono **identificate esternamente** o meglio la chiave dell'entità padre **G** viene utilizzata nelle figlie sia come chiave primaria che come chiave esterna.

Quindi occorre introdurre:
- una relazione $\mathbf{R_G}$ avente gli attributi dell'entità padre **G** e gli attributi chiave primaria $\mathbf{K_G}$;
- una relazione $\mathbf{R_{S1}}$ avente gli attributi chiave $\mathbf{K_G}$ di $\mathbf{R_G}$ come attributi chiave primaria, poi gli altri attributi di $\mathbf{S_1}$ ed infine gli eventuali attributi per le chiavi esterne;
- una relazione $\mathbf{R_{S2}}$ avente gli attributi chiave $\mathbf{K_G}$ di $\mathbf{R_G}$ come attributi chiave primaria, poi gli altri attributi di $\mathbf{S_2}$ ed infine gli eventuali attributi per le chiavi esterne;
- ………………………………………………………………………………
- una relazione $\mathbf{R_{SN}}$ avente gli attributi chiave $\mathbf{K_G}$ di $\mathbf{R_G}$ come attributi chiave primaria, poi gli altri attributi di $\mathbf{S_N}$ ed infine gli eventuali attributi per le chiavi esterne;
- un **vincolo di integrità referenziale** che dica *"ad ogni ennupla di G non possono corrispondere contemporaneamente più ennuple in entità figlie distinte"* (esclusività della generalizzazione).

Se la Generalizzazione ESCLUSIVA

Diagramma ER



Schema relazionale

$$R_G\ (\ \langle \underline{K_G} \rangle\ ,\ \langle \text{Attributi di } G \rangle\ )$$

$$R_{S1}\ (\ \langle \underline{K_G} \rangle\ ,\ \langle \text{Attributi di } S_1 \rangle\ )$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$R_{SN}\ (\ \langle \underline{K_G} \rangle\ ,\ \langle \text{Attributi di } S_N \rangle\ )$$

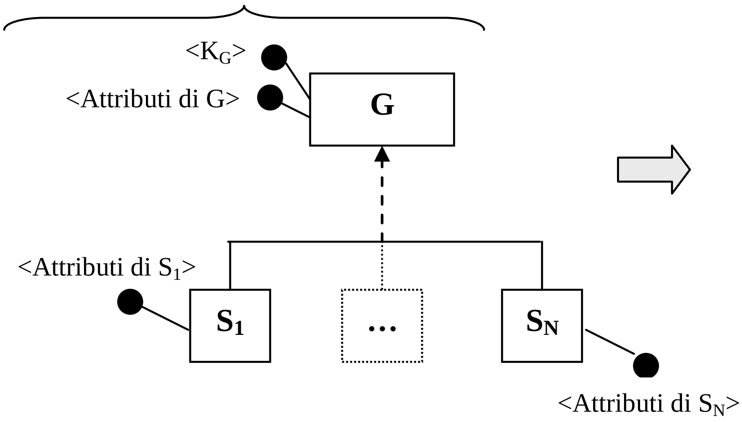
(A) Vincoli di integrità referenziale

$$VR_{KG}(R_{S1}) \subseteq VR_{KG}(R_G)$$
XOR
$$\ldots\ldots\ldots$$
XOR
$$VR_{KG}(R_{SN}) \subseteq VR_{KG}(R_G)$$

(A) Il vincolo referenziale esprime il fatto che "*ad ogni n-pla di G non possono corrispondere contemporaneamente più n-ple in entità figlie distinte*"

Se la Generalizzazione è TOTALE

Diagramma ER



Schema relazionale

$$R_G\ (\ \langle \underline{K_G} \rangle\ ,\ \langle \text{Attributi di } G \rangle\ )$$

$$R_{S1}\ (\ \langle \underline{K_G} \rangle\ ,\ \langle \text{Attributi di } S_1 \rangle\ )$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$R_{SN}\ (\ \langle \underline{K_G} \rangle\ ,\ \langle \text{Attributi di } S_N \rangle\ )$$

(B) Vincoli di integrità referenziale

$$VR_{KG}(R_G) \subseteq VR_{KG}(R_{S1})$$
OR
$$\ldots\ldots\ldots$$
OR
$$VR_{KG}(R_G) \subseteq VR_{KG}(R_{SN})$$

(B) Il vincolo referenziale esprime il fatto *"che ad ogni n-pla di G deve corrispondere almeno una n-pla in una delle entità figlie"*

Se la Generalizzazione è TOTALE ed ESCLUSIVA
Lo schema relazionale è lo stesso. Inoltre valgono entrambi i vincoli referenziali (A) e (B)

N.B. VEDI ESEMPIO DI MAPPING COMPLETO SUL SITO

UNA SEMPLICE FORMULA

Per avere un immediato riscontro alla “bontà” del nostro mapping del diagramma ER nello schema relazionale possiamo utilizzare la seguente formula che dato un diagramm aER ci permette di calcolare le relazioni che ne derivano:

Numero di relazioni = Numero di Entità – Numero di associazioni 1:1 + Numero di associazioni N:N

OSS: Rispetto alla formula dobbiamo specificare che
1) non compaiono associazioni di tipo 1:N in quanto vengono inglobate nelle entità che sono legate;
2) viene sottratto il numero di associazioni di tipo 1:N in quanto le due entità legate vengono fuse in una sola relazione;
3) per le generalizzazioni non occorre aggiungere altro in quanto vengono implementate con associazioni di tipo 1:1 interne alle entità figlie;

UN ESEMPIO COMPLETO DI DERIVAZIONE DAL DIAGRAMMA ER

Vedi esempio pagina 60 e vedi esempi forniti a lezione

LE OPERAZIONI RELAZIONALI

Ora che sappiamo che la nostra base di dati può essere rappresentata con un insieme di relazioni concentriamo la nostra attenzione sulle operazioni che consentono di **interrogare** una base dati relazionale appena creata.

Nel tempo sono stati proposti diversi **linguaggi per l'interrogazione** di una base di dati relazionali quasi tutti di tipo **non procedurale.**
Tali linguaggi di interrogazione utilizzano uno dei due seguenti approcci:

a) approccio **basato sull'algebra relazionale**
In questo approccio il risultato di una **interrogazione o query** è una *relazione*.. Per ottenere tale relazione si formula una interrogazione utilizzando alcuni operatori di ***algebra relazionale*** (ad esempio operatori di *unione*, *intersezione* e *differenza tra relazioni* in senso insiemistico) che vengono composti tra loro ed applicati alle relazioni della base dati;

b) approccio **basato sul calcolo relazionale**:
Anche in questo approccio il risultato di una **interrogazione o query** è una *relazione*.. Per ottenere tale relazione si formula una interrogazione utilizzando ***il calcolo dei predicati del primo ordine*** sulle relazioni della base dati.

I due approcci possono considerarsi **equivalenti sia dal punto di vista espressivo** (ossia la relazione ottenuta utilizzando l'approccio dell'algebra relazionale può essere espressa con un equivalente predicato del primo ordine), **sia dal punto di vista implementativo** . (ossia non vi è alcuna differenza tra i due approcci riguardo alla velocità di scrittura dell'interrogazione).

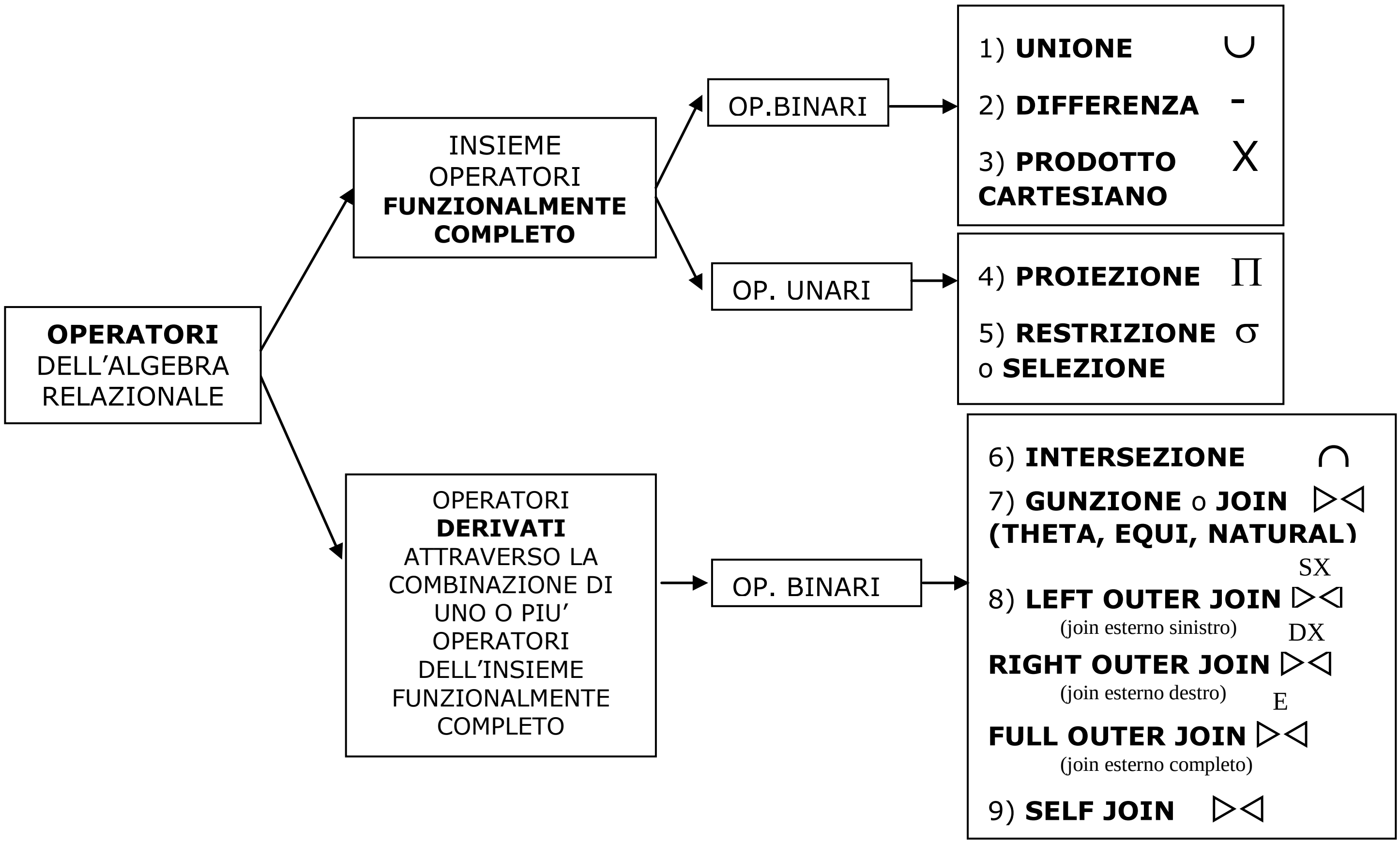
Scegliamo di utilizzare l'approccio <u>dell'algebra relazionale</u> vista la maggiore familiarità con gli operatori algebrici rispetto al calcolo dei predicati.

<u>Algebra relazionale</u>

Occorre scegliere innanzitutto un insieme **funzionalmente completo** di operatori (ossia un insieme di operatori che ha la caratteristica di essere sufficiente per rappresentare tutte le funzioni) da utilizzare nelle *interrogazioni o query*.

DEF: Un insieme **funzionalmente completo di operatori** è quello formato dai cinque operatori relazionali:
- 1) **unione** di due relazioni;
- 2) **differenza** di due relazioni;
- 3) **prodotto cartesiano** di due relazioni;
- 4) **proiezione** di una relazione;
- 5) **restrizione** di una relazione;

Oltre a questi cinque operatori è opportuno introdurne altri quattro operatori che possono essere ricavati naturalmente dai primi cinque (funzionalmente completo) ma il cui utilizzo permette di scrivere formule di interrogazioni più semplici e sintetiche.
- 6) **intersezione** di due relazioni;
- 7) **giunzione (THETA-JOIN, EQUI-JOIN, NATURAL JOIN)** di due relazioni;
- 8) **giunzione (sinistra|destra|completa) esterna** (**LEFT|RIGHT|FULL| OUTER JOIN**) di due relazioni;
- 9) **auto-giunzione** di una relazione (**SELF JOIN**) (N.B. è un equi join sulla stessa relazione);



INSIEME DEI 5 OPERATORI RELAZIONALI FUNZIONALMENTE COMPLETI

DEF: Due relazioni **R** ed **S** vengono chiamate **compatibili** se:
- hanno lo stesso *numero* di attributi;
- ogni attributo nella stessa posizione all'interno delle due relazioni è dello *stesso tipo*.;

1) UNIONE di due relazioni (operatore ∪)

DEF: Date due relazioni **compatibili R** ed **S** l'unione di **R** con **S** è la relazione ottenuta dall'unione insiemistica delle due relazioni ossia:

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

Graficamente

Per come è stata definita l'operazione di unione abbiamo che:

Grado (R ∪ S) = Grado (R) = Grado (S)

Card (R ∪ S) = Card (R) + Card (S) – numero di ennuple ripetute

N.B E' una operazione commutativa in quanto è facile dimostrare che **R ∪ S = S ∪ R**

<u>Esempio:</u> Siano date le seguenti due relazioni **R** ed **S compatibili** così definite utilizzando la rappresentazione tabellare:

R= Cliente-2004 **S**= Cliente-2005

R	**<u>CodCliente</u>**	**Nominativo**	**Indirizzo**	**Provincia**
	C001	Neri Mario	Via Po, 5	Napoli
	C002	Bianchi Gianni	Via Lima, 7	Milano
	C003	Rossi Antonio	Via Riga, 9	Napoli

Grado (R) = **4**
Card (R) = **3**

S	**<u>CodCliente</u>**	**Nominativo**	**Indirizzo**	**Provincia**
	C002	Bianchi Gianni	Via Lima, 7	Milano
	C004	Verdi Giuseppe	Via Pia, 11	Lecce

Grado (S) = **4**
Card (S) = **2**

Allora per come è stato definito l'operatore relazionale $\cup$ si ha che:

R ∪ S	**<u>CodCliente</u>**	**Nominativo**	**Indirizzo**	**Provincia**
	C001	Neri Mario	Via Po, 5	Napoli
	C002	Bianchi Gianni	Via Lima, 7	Milano
	C003	Rossi Antonio	Via Riga, 9	Napoli
	C004	Verdi Giuseppe	Via Pia, 11	Lecce

Grado (**Cliente-2004** ∪ **Cliente-2005**) = **Grado** (**Cliente-2004**) = **Grado** (**Cliente-2005**) = **4**

Card (**Cliente-2004**∪ **Cliente-2005**) = **Card** (**Cliente-2004**) + **Card** (**Cliente-2005**) – numero di ennuple ripetute = (3 + 2) – 1 = **4**

<u>2) DIFFERENZA di due relazioni</u> <u>(operatore -)</u>

DEF: Date due relazioni **compatibili R** ed **S** <u>la differenza di **R** con **S**</u> è la relazione ottenuta dalla differenza insiemistica delle due relazioni ossia:

$$\mathbf{R - S = \{ \; t \mid t \in R \text{ and } t \notin S \; \}}$$

Graficamente

Per come è stata definita l'operazione di differenza abbiamo che:

Grado (R - S) = Grado (R) = Grado (S)

Card (R - S) = Card (R) – numero di ennuple in comune tra R ed S

N.B Non è una operazione commutativa in quanto è facile dimostrare che **R – S ≠ S - R**

Esempio: Siano date le seguenti due relazioni **R** ed **S compatibili** così definite utilizzando la rappresentazione tabellare:

R= Cliente-2004 **S**= Cliente-2005

R	CodCliente	Nominativo	Indirizzo	Provincia
	C001	Neri Mario	Via Po, 5	Napoli
	C002	Bianchi Gianni	Via Lima, 7	Milano
	C003	Rossi Antonio	Via Riga, 9	Napoli

Grado (R) = 4
Card (R) = 3

S	CodCliente	Nominativo	Indirizzo	Provincia
	C002	Bianchi Gianni	Via Lima, 7	Milano
	C004	Verdi Giuseppe	Via Pia, 11	Lecce

Grado (S) = 4
Card (S) = 2

Allora per come è stato definito l'operatore relazionale **-** si ha che:

R - S	CodCliente	Nominativo	Indirizzo	Provincia
	C001	Neri Mario	Via Po, 5	Napoli
	C003	Rossi Antonio	Via Riga, 9	Napoli

Grado (**Cliente-2004 - Cliente-2005**) = **Grado** (**Cliente-2004**) = **Grado** (**Cliente-2005**) = **4**

Card (**Cliente-2004 - Cliente-2005**) = **Card** (**Cliente-2004**) – numero di ennuple ripetute = (3 - 1) = **2**

3) PRODOTTO CARTESIANO di due relazioni (operatore X)

DEF: Date due relazioni **qualsiasi R** ed **S** rispettivamente di **grado g1** e **g2** e **cardinalità c1** e **c2**, il prodotto cartesiano di **R** ed **S** è la relazione di **grado g1 + g2** e **cardinalità c1 x c2,** le cui ennuple si ottengono concatenando ogni ennupla di **R** con ogni ennupla di **S**.

Quindi se consideriamo una qualsiasi ennupla della prima relazione $\mathbf{r = (a_1, a_2, \dots , a_{g1})}$ ed una qualsiasi ennupla della seconda relazione $\mathbf{s = (b_1, b_2, \dots , b_{g2})}$ e definiamo l'operazione **conc** come

$$\mathbf{r \ conc \ s = (a_1, a_2, \dots , a_{g1}, b_1, b_2, \dots , b_{g2})}$$

allora il prodotto cartesiano delle relazioni **R** ed **S** viene allora definito come

$$\mathbf{R} \ \mathrm{X} \ \mathbf{S} = \{ \mathbf{t} \mid \mathbf{t = r \ conc \ s, r \in R , s \in S} \}$$

N.B. Per evitare ambiguità nei nomi degli attributi di R X S occorre che i nome degli attributi di R e di S siano diversi tra loro (eventualmente occorre rinominarli opportunamente prima di procedere all'operazione).

Per come è stata definita l'operazione di prodotto cartesiano abbiamo che:

Grado (R X S) = Grado (R) + Grado (S)

Card (R X S) = Card (R) * Card (S)

N.B Non è una operazione commutativa in quanto è facile dimostrare che **R X S ≠ S X R**

N.B. Vista la natura dell'operazione meramente algebrica il risultato di un prodotto cartesiano potrebbe non avere un significato ben chiaro anche in contesti semplici: generalmente questa operazione è un passo intermedio di una elaborazione più complessa.

<u>Esempio:</u> Siano date le seguenti due relazioni **R** ed **S** così definite utilizzando la rappresentazione tabellare:

R= Alunno S= Testo

R	**<u>Matricola</u>**	**Nominativo**	**Data**
	C001	Neri Mario	01/06/1978
	C002	Bianchi Gianni	02/07/1979
	C003	Rossi Antonio	05/08/1977

Grado (R) = **3**
Card (R) = **3**

S	**<u>CodTesto</u>**	**Titolo**	**Materia**
	T001	L'italiano oggi	Italiano
	T002	ITC	Informatica

Grado (S) = **3**
Card (S) = **2**

Allora per come è stato definito l'operatore relazionale **X** si ha che:

R X S	**<u>Matricola</u>**	**Nominativo**	**Data**	**<u>CodTesto</u>**	**Titolo**	**Materia**
	C001	Neri Mario	01/06/1978	T001	L'italiano oggi	Italiano
	C001	Neri Mario	01/06/1978	T002	ITC	Informatica
	C002	Bianchi Gianni	02/07/1979	T001	L'italiano oggi	Italiano
	C002	Bianchi Gianni	02/07/1979	T002	ITC	Informatica
	C003	Rossi Antonio	05/08/1977	T001	L'italiano oggi	Italiano
	C003	Rossi Antonio	05/08/1977	T002	ITC	Informatica

Grado (**Alunno X Testo**) = **Grado** (**Alunno**) + **Grado** (**Testo**) = (3 + 3) = **6**
Card (**Alunno X Testo**) = **Card** (**Alunno**) * **Card** (**Testo**) = (3 * 2) = **6**

<u>4) PROIEZIONE di una relazione (operatore Π)</u>

DEF: Data una relazione **R** ed un sottoinsieme **A** = $\{A_1, A_2, \ldots, A_k\}$ dei suoi attributi si definisce <u>proiezione di **R** su **A**</u> la relazione di grado **K** che si ottiene da **R** ignorando le colonne relative agli attributi non contenuti in **A** ed eliminando le eventuali ennuple duplicate.

Pertanto scriveremo

$$\Pi_{A_1,A_2,\ldots A_K}(R) = \{\ t\,[A_1,A_2,\ldots A_k]\ |\ t \in R\ \}$$

N.B. L'effetto di tale operazione è quello di selezionare *<u>un certo numero di colonne</u>* della tabella relazione. E' possibile, in teoria, scegliere tutti gli attributi di R. In questo caso la proiezione di R su tutti i suoi attributi coinciderebbe ovviamente con la stessa R.

Per come è stata definita l'operazione di proiezione abbiamo che:

Grado ($\Pi_{A_1,A_2,\ldots A_K}(R)$ **) = K**

Card ($\Pi_{A_1,A_2,\ldots A_K}(R)$ **)** non è prevedibile a priori ma sicuramente <u>minore o uguale</u> a **Card (R)**

Esempio: Sia data la seguenti relazione **R** così definita utilizzando la rappresentazione tabellare:

R= Cliente

R	CodCliente	Nominativo	Indirizzo	Provincia
	C001	Neri Mario	Via Po, 5	Napoli
	C002	Bianchi Gianni	Via Lima, 7	Milano
	C003	Rossi Antonio	Via Riga, 9	Napoli

Grado (R) = 4
Card (R) = 3

Allora per come è stato definito l'operatore relazionale Π si ha che nel caso si volesse effettuare una proiezione rispetto ai soli 2 attributi di R CodCliente e Provincia si otterrebbe la seguente relazione (senza eliminazione di duplicati perché ovviamente è presente la chiave primaria):

Π CodCliente, Provincia (**R**)	CodCliente	Provincia
	C001	Napoli
	C002	Milano
	C003	Napoli

Grado (Π CodCliente, Provincia (**Cliente**)) = **2** (ossia pari al numero di attributi scelto)

Card (Π CodCliente, Provincia (**Cliente**)) = **3** e comunque sicuramente ≤ **Card** (**Cliente**)

Qualora si volesse effettuare una proiezione rispetto al solo attributo di R Provincia si otterrebbe la seguente relazione (con eliminazione di duplicati):

Π Provincia (**R**)	Provincia
	Napoli
	Milano

Grado (Π CodCliente, Provincia (**Cliente**)) = **1** (ossia pari al numero di attributi scelto)

Card (Π CodCliente, Provincia (**Cliente**)) = **2** e comunque sicuramente ≤ **Card** (**Cliente**)

5) RESTRIZIONE O SELEZIONE di una relazione (operatore σ)

DEF: Data una relazione **R** ed un predicato **P** (semplice o composto) sui suoi attributi, si definisce restrizione di **R** a **P** la relazione costituita dalle ennuple di **R** che soddisfano **P.**

Pertanto scriveremo

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

N.B. L'effetto di tale operazione è quello di selezionare *un certo numero di righe* della tabella relazione.

Se il predicato P risultasse vero per tutte le n-ple di R, allora la restrizione di R a P coinciderebbe ovviamente con la stessa R (ossia avrebbe la stessa cardinalità di R).
Se il predicato P risultasse falso per tutte le n-ple di R, allora la restrizione di R a P non selezionerebbe alcuna n-pla (ossia avrebbe cardinalità pari a zero).

Per come è stata definita l'operazione di proiezione abbiamo che:

Grado (σ_P (R)) = Grado (R)

Card (σ_P (R)) non è prevedibile a priori ma sicuramente minore o uguale a **Card(R)**

Esempio: Sia data la seguente relazione **R** così definita utilizzando la rappresentazione tabellare:

R= Cliente e sia **P** il seguente predicato semplice **Provincia = "Napoli"** ossia
P = {Provincia = "Napoli"}

R	CodCliente	Nominativo	Indirizzo	Provincia
	C001	Neri Mario	Via Po, 5	Napoli
	C002	Bianchi Gianni	Via Lima, 7	Milano
	C003	Rossi Antonio	Via Riga, 9	Napoli

Grado (R) = **4**
Card (R) = **3**

Allora per come è stato definito l'operatore relazionale σ e tenendo conto di come è definito **P** si ha:

$\sigma_{Provincia="Napoli"}$ (R)	CodCliente	Nominativo	Indirizzo	Provincia
	C001	Neri Mario	Via Po, 5	Napoli
	C003	Rossi Antonio	Via Riga, 9	Napoli

Grado ($\sigma_{Provincia="Napoli"}$ **(Cliente)**) = **4** (ossia pari al numero di attributi scelto)

Card ($\sigma_{Provincia="Napoli"}$ **(Cliente)**) = **2** e comunque sicuramente $\leq$ **Card** (**Cliente**)

Se invece **P** il seguente predicato composto **Provincia = "Napoli" e CodCliente = "C001"** ossia
P = {Provincia = "Napoli" AND CodCliente="C001"} allora per come è stato definito l'operatore relazionale σ si ha:

$\sigma_{(Provincia="Napoli") \text{ AND } (CodCliente="C001")}$ (R)	CodCliente	Nominativo	Indirizzo	Provincia
	C001	Neri Mario	Via Po, 5	Napoli

OPERATORI RELAZIONALI DERIVATI

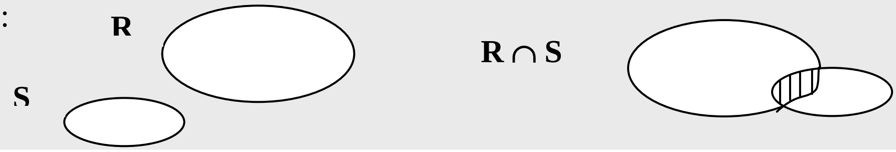
<u>**6) INTERSEZIONE di due relazioni** (operatore</u> $\cap$)

DEF: Data una relazione **compatibili R** ed **S**, <u>l'intersezione di **R** ed **S**</u> restituisce la relazione ottenuta dall'intersezione insiemistica tra le relazioni ossia

$$\mathbf{R \cap S = \{ \ t \mid t \in R \ and \ t \in S \ \}}$$

Essa restituisce in pratica tutte le ennuple presenti sia in **R** che in **S**.

Graficamente:



E' possibile verificare grazie con l'utilizzo dei diagrammi di Eulero-Venn per gli insiemi che vale la seguente uguaglianza (e ciò dimostra che tale operatore è derivabile applicando opportunamente alla relazione uno o più operatori fondamentali)

$$\mathbf{R \cap S = R - (R - S)}$$

Per come è stata definita l'operazione di proiezione abbiamo che:

Grado (R ∩ S) = Grado (R)

Card (R ∩ S) non è prevedibile a priori ma sicuramente <u>minore o uguale</u> del valore più piccolo tra **Card(R) e Card(S).**

<u>**Esempio:**</u> Siano date le seguenti due relazioni **R** ed **S compatibili** così definite utilizzando la rappresentazione tabellare:

R= Clienti-2004 **S**= Clienti-2005

R	<u>**CodCliente**</u>	**Nominativo**	**Indirizzo**	**Provincia**
	C001	Neri Mario	Via Po, 5	Napoli
	C002	Bianchi Gianni	Via Lima, 7	Milano
	C003	Rossi Antonio	Via Riga, 9	Napoli

Grado (R) = 4
Card (R) = 3

S	<u>**CodCliente**</u>	**Nominativo**	**Indirizzo**	**Provincia**
	C002	Bianchi Gianni	Via Lima, 7	Milano
	C004	Verdi Giuseppe	Via Pia, 11	Lecce

Grado (S) = 4
Card (S) = 2

Allora per come è stato definito l'operatore relazionale $\cap$ si ha che:

R ∩ S	<u>**CodCliente**</u>	**Nominativo**	**Indirizzo**	**Provincia**
	C002	Bianchi Gianni	Via Lima, 7	Milano

Grado (**Clienti-2004 ∩ Clienti-2005**) = **Grado** (**Clienti-2004**) = **Grado** (**Clienti-2005**) = **4**

Card(**Clienti-2004∩ Clienti-2005**) = **1** ossia ≤ **min** { **Card(Clienti-2004), Card**(**Clienti-2005**) }

N.B. Casi particolari:

Se $\mathbf{R} \subseteq \mathbf{S}$ allora è facile dimostrare che **Card (R⋂S) = Card (R)**

Se $\mathbf{S} \subseteq \mathbf{R}$ allora è facile dimostrare che **Card (R⋂S) = Card (S)**

<u>7) EQUI GIUNZIONE tra due relazioni</u> <u>(operatore **EQUI-JOIN** </u>$\triangleright\triangleleft$)

Innanzitutto il **join** è uno degli operatori relazionali più importanti, come vedremo, per **combinare informazioni tra due o più relazioni**

DEF: Date due relazioni **qualsiasi R** ed **S** di **grado g1** e **g2** e **cardinalità c1** e **c2,** si definisce **join condizionale o <u>THETA-JOIN</u>** <u>tra **R** ed **S**</u> la relazione di **grado (g1 + g2)** e **cardinalità** <u>non prevedibile a priori ma minore o uguale a</u> **c1*c2** le cui ennuple si ottengono effettuando una **selezione o restrizione** secondo un certo predicato **P** sul loro **prodotto cartesiano**.
(in sintesi σ_P **(R X S))**
Il predicato **P** può contenere un enunciato semplice o composto che utilizzi uno qualsiasi dei 6 operatori di confronto fondamentali (≤ , <, > , ≥ , =, ≠)

N.B. Se in un THETA-JOIN si utilizza, come operatore di confronto, esclusivamente l'operatore di uguaglianza si parla di **EQUI-JOIN**

DEF: Date due relazioni **qualsiasi R** ed **S** di **grado g1** e **g2 cardinalità c1** e **c2** ed un *attributo* **A** di **R** ed un *attributo* **B** di **S**, *aventi lo stesso tipo*, si definisce **equi giunzione o <u>EQUI-JOIN</u>** <u>tra **R** ed **S**</u> la relazione di **grado (g1 + g2)** e **cardinalità** <u>non prevedibile a priori ma minore o uguale a</u> **c1*c2** le cui ennuple si ottengono con il seguente procedimento:
a) si effettua il **prodotto cartesiano** di R ed S;
b) sulla relazione così ottenuta si effettua una **restrizione (o selezione)** volta a selezionare le ennuple aventi **lo stesso valore** degli attributi **A** e **B (** ossia predicato P **=** { R.A = S.B }),
ottenendo così una relazione con le colonne **A** e **B** uguali
Indicheremo questa operazione con

$$\mathbf{R} \underset{R.A\,=\,S.B}{\triangleright\triangleleft} \mathbf{S}$$

N.B. Se in un EQUI-JOIN i nomi degli attributi delle due relazioni che si utilizzano nella condizione di uguaglianza rispetto valori posseduti dalle ennuple SONO UGUALI, è possibile eliminare una delle due colonne (perché informazioni ridondanti) ed ottenere una relazione uguale alla precedente ma con grado **(g1 + g2 - 1)** che prende il nome di giunzione naturale **o <u>NATURAL-JOIN</u>** <u>tra **R** ed **S**</u>

DEF: Date due relazioni **qualsiasi R** ed **S** di **grado g1** e **g2 cardinalità c1** e **c2** ed un *attributo* **A** di **R** ed un *attributo* **A** di **S**, *aventi lo stesso tipo e lo stesso nome*, si definisce **giunzione naturale o <u>NATURAL-JOIN</u>** <u>tra **R** ed **S**</u> la relazione di **grado (g1 + g2 -1)** e **cardinalità** <u>non prevedibile a priori ma minore o uguale a</u> **c1*c2** le cui ennuple si ottengono con il seguente procedimento:
a) si effettua il **prodotto cartesiano** di R ed S;
b) sulla relazione così ottenuta si effettua una **restrizione (o selezione)** volta a selezionare le ennuple aventi **lo stesso valore** degli attributi R.**A** e S.**A (** ossia predicato P **=** { R.A = S.A }),
ottenendo così una relazione con le colonne R.**A** e S.**A** con gli stessi valori e con lo stesso nome;
c) si elimina una di queste due colonne.
Indicheremo l'operazione con

$$\mathbf{R} \underset{R.A\,=\,S.A}{\triangleright\triangleleft} \mathbf{S}$$

N.B. Lo scopo della giunzione naturale o NATURAL-JOIN in sintesi è quello di combinare due relazioni aventi uno o più attributi in comune generando una nuova relazione che contiene:
- le *colonne* della prima e della seconda meno gli attributi in comune;
- le *righe* della prima e della seconda combinate secondo i valori uguali dell'attributo in comune.

Esempio (EQUI-JOIN): Siano date le seguenti due relazioni **R** ed **S** così definite utilizzando la rappresentazione tabellare:

R= Cliente **S**= Agente

R	CodCliente	NomeCliente	Indirizzo	CodAg
	C001	Neri Mario	Via Po, 5	A001
	C002	Bianchi Gianni	Via Lima, 7	A002
	C003	Rossi Antonio	Via Riga, 9	A001
	C004	Russo Mario	Via Roma, 8	NULL

Grado (R) = 4
Card (R) = 4

S	CodAgente	NomeAgente	Telefono
	A001	Verdi Luca	081-123456
	A002	Gialli Matteo	081-654321
	A003	Vito Andrea	081-456345

Grado (S) = 3
Card (S) = 3

Proviamo a far vedere come si ricava la relazione indicata dall'operazione relazionale

$$\textbf{Cliente} \underset{\text{CodAg = CodAgente}}{\rhd\!\lhd} \textbf{Agente}$$

che ha come scopo quello di mostrare in un'unica tabella per ogni cliente anche le informazioni relative agli agenti che li servono.

Come da definizione per prima cosa costruiamo la tabella **R X S**

R X S	CodCliente	NomeCliente	Indirizzo	CodAg	CodAgente	NomeAgente	Telefono
	C001	Neri Mario	Via Po, 5	**A001**	**A001**	Verdi Luca	081-123456
	C001	Neri Mario	Via Po, 5	A001	A002	Gialli Matteo	081-654321
	C001	Neri Mario	Via Po, 5	A001	A003	Vito Andrea	081-456345
	C002	Bianchi Gianni	Via Lima, 7	A002	A001	Verdi Luca	081-123456
	C002	Bianchi Gianni	Via Lima, 7	**A002**	**A002**	Gialli Matteo	081-654321
	C002	Bianchi Gianni	Via Lima, 7	A002	A003	Vito Andrea	081-456345
	C003	Rossi Antonio	Via Riga, 9	**A001**	**A001**	Verdi Luca	081-123456
	C003	Rossi Antonio	Via Riga, 9	A001	A002	Gialli Matteo	081-654321
	C003	Rossi Antonio	Via Riga, 9	A001	A003	Vito Andrea	081-456345
	C004	Russo Mario	Via Roma, 8	NULL	A001	Verdi Luca	081-123456
	C004	Russo Mario	Via Roma, 8	NULL	A002	Gialli Matteo	081-654321
	C004	Russo Mario	Via Roma, 8	NULL	A003	Vito Andrea	081-456345

Proseguiamo effettuando la restrizione o selezione sulla relazione ottenuta ossia

$$\sigma_P \textbf{ (R X S)} \text{ con } \mathbf{P} = \{ \text{Cliente.CodAg = Agente.CodAgente} \}$$

σ_P (R X S)	CodCliente	NomeCliente	Indirizzo	CodAg	CodAgente	NomeAgente	Telefono
	C001	Neri Mario	Via Po, 5	A001	A001	Verdi Luca	081-123456
	C002	Bianchi Gianni	Via Lima, 7	A002	A002	Gialli Matteo	081-654321
	C003	Rossi Antonio	Via Riga, 9	A001	A001	Verdi Luca	081-123456

Risultato finale dell' EQUI-JOIN proposto è la seguente tabella

Cliente $\bowtie_{CodAg\ =\ CodAgente}$ **Agente**

<u>CodCliente</u>	**NomeCliente**	**Indirizzo**	**R.CodAgente**	**<u>S.CodAgente</u>**	**NomeAgente**	**Telefono**
C001	Neri Mario	Via Po, 5	A001	A001	Verdi Luca	081-123456
C002	Bianchi Gianni	Via Lima, 7	A002	A002	Gialli Matteo	081-654321
C003	Rossi Antonio	Via Riga, 9	A001	A001	Verdi Luca	081-123456

Grado (**Cliente** $\bowtie_{CodAg\ =\ CodAgente}$ **Agente**) = **Grado** (**Agente**) + **Grado** (**Cliente**) **=** (4 + 3) = **7**

Card (**Cliente** $\bowtie_{CodAg\ =\ CodAgente}$ **Agente**) = **3** che risulta ≤ **Card** (**Agente**) * **Card** (**Cliente**) = 12

<u>Esempio (NATURAL-JOIN):</u> Siano date le seguenti due relazioni **R** ed **S** definite a partire dalla rappresentazione tabellare vista nell'esercizio precedente, modificando esclusivamente il nome di un attributo della relazione Cliente in modo che esso coincida con il nome di un attributo della relazione agente (CodAgente):

R= Cliente   **S**= Agente

R	**<u>CodCliente</u>**	**NomeCliente**	**Indirizzo**	**CodAgente**
	C001	Neri Mario	Via Po, 5	A001
	C002	Bianchi Gianni	Via Lima, 7	A002
	C003	Rossi Antonio	Via Riga, 9	A001
	C004	Russo Mario	Via Roma, 8	NULL

Grado (R) = **4**
Card (R) = **4**

S	**<u>CodAgente</u>**	**NomeAgente**	**Telefono**
	A001	Verdi Luca	081-123456
	A002	Gialli Matteo	081-654321
	A003	Vito Andrea	081-456345

Grado (S) = **3**
Card (S) = **3**

Il risultato finale del NATURAL-JOIN proposto, dopo tutti i passi previsti, può essere ottenuto eliminando la colonna Cliente.CodAgente dalla tabella dell'EQUI-JOIN ottenuta al passo precedente

<u>CodCliente</u>	**NomeCliente**	**Indirizzo**	~~**R.CodAgente**~~	**<u>S.CodAgente</u>**	**NomeAgente**	**Telefono**
C001	Neri Mario	Via Po, 5	~~A001~~	A001	Verdi Luca	081-123456
C002	Bianchi Gianni	Via Lima, 7	~~A002~~	A002	Gialli Matteo	081-654321
C003	Rossi Antonio	Via Riga, 9	~~A001~~	A001	Verdi Luca	081-123456

Grado (**Cliente** $\bowtie_{Cliente.CodAgente\ =\ Agente.CodAgente}$ **Agente**) = **Grado** (**Agente**) + **Grado** (**Cliente**) - 1 **=** (4 + 3) -1 = **6**

Card (**Cliente** $\bowtie_{Cliente.CodAgente\ =\ Agente.CodAgente}$ **Agente**) = **3** che risulta ≤ **Card** (**Agente**) * **Card** (**Cliente**) = 12

ALTRI TIPI DI JOIN

8.1) Join esterno sinistro (Left outer join ossia Left join $\overset{SX}{\rhd\lhd}$)

8.2) Join esterno destro (Right outer join ossia Right join $\overset{DX}{\rhd\lhd}$)

8.3) Join esterno completo (Full outer join ossia Outer join $\overset{E}{\rhd\lhd}$)

9) Self Join

Oltre all'operatore di equi-giunzione o **equi-join (chiamato anche INNER-JOIN o join interno)** visto in precedenza, esistono altri tipi di join che restituiscono non solo le n-ple risultanti da questa specifica operazione di congiunzione e che rispettano la condizione (o clausola) specificata, ma anche le n-ple delle singole relazioni che non la soddisfano e pertanto non sono state congiunte.

8.1) Left outer join o left join: Aggiunge alle n-ple risultanti dall'applicazione dell'operatore relazionale equi-join (o inner join) anche tutte le eventuali n-ple presenti nella relazione a sinistra della condizione (o clausola) specificata che non hanno trovato corrispondenti, completate concatenando opportuni valori NULL per i restanti campi relativi alla relazione a destra della condizione stessa.

$\mathbf{R} \underset{R.A = S.B}{\overset{SX}{\rhd\lhd}} \mathbf{S} = \mathbf{R} \underset{R.A = S.B}{\rhd\lhd} \mathbf{S}$ **U** Insieme, se esiste, delle ennuple della relazione **R** che non sono state congiunte completate con opportuni valori **NULL** per gli attributi della relazione **S**

8.2) Right outer join o right join: Aggiunge alle n-ple risultanti dall'applicazione dell'operatore relazionale equi-join (o inner join) anche tutte le eventuali n-ple presenti nella relazione a destra della condizione (o clausola) specificata che non hanno trovato corrispondenti, completate concatenando opportuni valori NULL per i restanti campi relativi alla relazione a sinistra della condizione stessa.
per i restanti campi relativi alla relazione a destra della condizione stessa.

$\mathbf{R} \underset{R.A = S.B}{\overset{DX}{\rhd\lhd}} \mathbf{S} = \mathbf{R} \underset{R.A = S.B}{\rhd\lhd} \mathbf{S}$ **U** Insieme, se esiste, delle ennuple della relazione **S** che non sono state congiunte completate con opportuni valori **NULL** per gli attributi della relazione **R**

8.3) Full outer join o outer join: E' la combinazione delle n-ple ottenute dall'operazione di unione tra join esterno sinistro e join esterno destro.
I dati presenti in una delle relazioni origine che non hanno corrispondenti nell'altra in accordo alla condizione imposta, conterranno valori NULL.
per i restanti campi relativi alla relazione a destra della condizione stessa.

$\mathbf{R} \underset{R.A = S.B}{\overset{E}{\rhd\lhd}} \mathbf{S} = \mathbf{R} \underset{R.A = S.B}{\overset{SX}{\rhd\lhd}} \mathbf{S} \quad \mathbf{U} \quad \mathbf{R} \underset{R.A = S.B}{\overset{DX}{\rhd\lhd}} \mathbf{S}$

9) Self join: E' l'applicazione dell'operatore algebrico relazionale di equi join applicato alla medesima relazione in ingresso (o in altre parole trattasi di equi giunzione applicata a due relazioni coincidenti) che realizza la congiunzione delle n-ple della relazione in ingresso con se stessa e che restituisce solo quelle che soddisfano la condizione (o clausola) specificata.
Esempio classico di utilizzo si ha quando occorre eseguire una giunzione naturale su una relazione ottenuta dal mapping relazionale di una associazione binaria ricorsiva.

Esempio join esterni: Siano date le seguenti due relazioni **R** ed **S** così definite utilizzando la rappresentazione tabellare:

R= Cliente **S**= Agente

R	CodCliente	NomeCliente	Indirizzo	CodAg
	C001	Neri Mario	Via Po, 5	A001
	C002	Bianchi Gianni	Via Lima, 7	A002
	C003	Rossi Antonio	Via Riga, 9	A001
	C004	Russo Mario	Via Roma, 8	NULL

Grado (R) = **4**
Card (R) = **4**

S	CodAgente	NomeAgente	Telefono
	A001	Verdi Luca	081-123456
	A002	Gialli Matteo	081-654321
	A003	Vito Andrea	081-456345

Grado (S) = **3**
Card (S) = **3**

Applicando l'operatore di EQUI-JOIN o INNER JOIN si ottiene la seguente tabella

Cliente $\bowtie_{CodAg = CodAgente}$ **Agente**

CodCliente	NomeCliente	Indirizzo	R.CodAgente	S.CodAgente	NomeAgente	Telefono
C001	Neri Mario	Via Po, 5	A001	A001	Verdi Luca	081-123456
C002	Bianchi Gianni	Via Lima, 7	A002	A002	Gialli Matteo	081-654321
C003	Rossi Antonio	Via Riga, 9	A001	A001	Verdi Luca	081-123456

Applicando l'operatore di LEFT OUTER JOIN o Left join si ottiene la seguente tabella

Cliente $\overset{SX}{\bowtie}_{CodAg = CodAgente}$ **Agente**

CodCliente	NomeCliente	Indirizzo	R.CodAgente	S.CodAgente	NomeAgente	Telefono
C001	Neri Mario	Via Po, 5	A001	A001	Verdi Luca	081-123456
C002	Bianchi Gianni	Via Lima, 7	A002	A002	Gialli Matteo	081-654321
C003	Rossi Antonio	Via Riga, 9	A001	A001	Verdi Luca	081-123456
C004	**Russo Mario**	**Via Roma, 8**	**NULL**	**NULL**	**NULL**	**NULL**

Applicando l'operatore di RIGHT OUTER JOIN o Right join si ottiene la seguente tabella

Cliente $\overset{DX}{\bowtie}_{CodAg = CodAgente}$ **Agente**

CodCliente	NomeCliente	Indirizzo	R.CodAgente	S.CodAgente	NomeAgente	Telefono
C001	Neri Mario	Via Po, 5	A001	A001	Verdi Luca	081-123456
C002	Bianchi Gianni	Via Lima, 7	A002	A002	Gialli Matteo	081-654321
C003	Rossi Antonio	Via Riga, 9	A001	A001	Verdi Luca	081-123456
NULL	**NULL**	**NULL**	**NULL**	**A003**	**Vito Andrea**	**081-456345**

Applicando l’operatore di FULL OUTER JOIN o Full join si ottiene la seguente tabella

$$\textbf{Cliente} \; \overset{F}{\underset{CodAg\,=\,CodAgente}{\triangleright\!\triangleleft}} \; \textbf{Agente}$$

<u>CodCliente</u>	NomeCliente	Indirizzo	R.CodAgente	<u>S.CodAgente</u>	NomeAgente	Telefono
C001	Neri Mario	Via Po, 5	A001	A001	Verdi Luca	081-123456
C002	Bianchi Gianni	Via Lima, 7	A002	A002	Gialli Matteo	081-654321
C003	Rossi Antonio	Via Riga, 9	A001	A001	Verdi Luca	081-123456
C004	**Russo Mario**	**Via Roma, 8**	**NULL**	**NULL**	**NULL**	**NULL**
NULL	**NULL**	**NULL**	**NULL**	**A003**	**Vito Andrea**	**081-456345**

Esempio: SELF JOIN

Consideriamo invece la seguente relazione:

R = Dipendente

R	<u>CodDip</u>	Cognome	Nome	Indirizzo	CodDir
	D001	Baracco	Michele	Via Po, 5	D002
	D002	Battistin	Marina	Via Adda, 7	NULL
	D003	Bessone	Fabia	Via Adige, 8	D002
	D004	Genchi	Mario	Via Tevere, 4	D003

Grado (R) = 5
Card (R) = 4

Supponendo di voler ottenere l’elenco dei dipendenti con associati i dati del rispettivo dirigente, è possibile impostare l’interrogazione mediante l’operazione di self join ossia una equi join sulla medesima relazione:

$$S = (\mathbf{R} \underset{A\ =B}{\triangleright\!\triangleleft} \mathbf{R}) \text{ ossia } (\textbf{Dipendente} \underset{CodDip\,=\,CodDir}{\triangleright\!\triangleleft} \textbf{Dipendente})$$

La sintesi della procedura è rappresentata dalla seguente relazione:

S	<u>CodDip</u>	Cognome	Nome	Indirizzo	CodDir	<u>CodDip</u>	Cognome	Nome	Indirizzo	CodDir
	D001	Baracco	Michele	Via Po, 5	D002	D001	Baracco	Michele	Via Po, 5	D002
	D001	Baracco	Michele	Via Po, 5	**D002**	**D002**	Battistin	Marina	Via Adda, 7	NULL
	D001	Baracco	Michele	Via Po, 5	D002	D003	Bessone	Fabia	Via Adige, 8	D002
	D001	Baracco	Michele	Via Po, 5	D002	D004	Genchi	Mario	Via Tevere, 4	D003
	D002	Battistin	Marina	Via Adda, 7	NULL	D001	Baracco	Michele	Via Po, 5	D002
	D002	Battistin	Marina	Via Adda, 7	NULL	D002	Battistin	Marina	Via Adda, 7	NULL
	D002	Battistin	Marina	Via Adda, 7	NULL	D003	Bessone	Fabia	Via Adige, 8	D002
	D002	Battistin	Marina	Via Adda, 7	NULL	D004	Genchi	Mario	Via Tevere, 4	D003
	D003	Bessone	Fabia	Via Adige, 8	D002	D001	Baracco	Michele	Via Po, 5	D002
	D003	Bessone	Fabia	Via Adige, 8	**D002**	**D002**	Battistin	Marina	Via Adda, 7	NULL
	D003	Bessone	Fabia	Via Adige, 8	D002	D003	Bessone	Fabia	Via Adige, 8	D002
	D003	Bessone	Fabia	Via Adige, 8	D002	D004	Genchi	Mario	Via Tevere, 4	D003
	D004	Genchi	Mario	Via Tevere, 4	D003	D001	Baracco	Michele	Via Po, 5	D002
	D004	Genchi	Mario	Via Tevere, 4	D003	D002	Battistin	Marina	Via Adda, 7	NULL
	D004	Genchi	Mario	Via Tevere, 4	**D003**	**D003**	Bessone	Fabia	Via Adige, 8	D002
	D004	Genchi	Mario	Via Tevere, 4	D003	D004	Genchi	Mario	Via Tevere, 4	D003

Ossia la tabella

S	<u>CodDip</u>	Cognome	Nome	Indirizzo	<u>CodDir</u>	<u>CodDip</u>	Cognome	Nome	Indirizzo	CodDir
	D001	Baracco	Michele	Via Po, 5	**D002**	**D002**	Battistin	Marina	Via Adda, 7	NULL
	D003	Bessone	Fabia	Via Adige, 8	**D002**	**D002**	Battistin	Marina	Via Adda, 7	NULL
	D004	Genchi	Mario	Via Tevere, 4	**D003**	**D003**	Bessone	Fabia	Via Adige, 8	D002

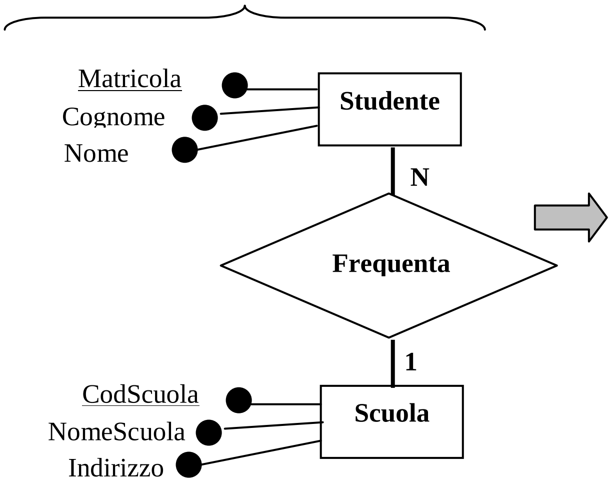
5) Le interrogazioni o query sullo schema relazionale eseguite con l'algebra relazionale

Applichiamo l'algebra relazionale per effettuare alcune semplici interrogazioni sulla nostra base di dati.

<u>Interrogazione di associazioni 1:N con diretta ed inversa TOTALI</u>

Aiutiamoci con un esempio

<u>Diagramma ER</u>



<u>Schema relazionale</u>

Studente (<u>Matricola</u> , Cognome, Nome , CodScuola1)

Chiave esterna

Scuola (<u>CodScuola</u> , NomeScuola, Indirizzo)

Vincoli di integrità referenziale per rappresentare la totalità delle associazioni diretta ed inversa

$$\mathbf{VR_{CodScuola1}\ (Studente) \subseteq VR_{CodScuola}\ (Scuola)}$$

$$\mathbf{VR_{CodScuola}\ (Scuola) \subseteq VR_{CodScuola1}\ (Studente)}$$

QUERY: Vogliamo conoscere i cognomi ed i nomi degli alunni che frequentano l'"ITIS Pozzuoli" (ipotizzando che abbia un CodScuola1 pari ad "NATF12000P")

Per quanto detto prima dovrà essere

$$\Pi_{\text{Cognome, Nome}}\left(\sigma_{\text{CodScuola1 = "NATF12000P"}}\ \mathbf{(Studente)}\right)$$

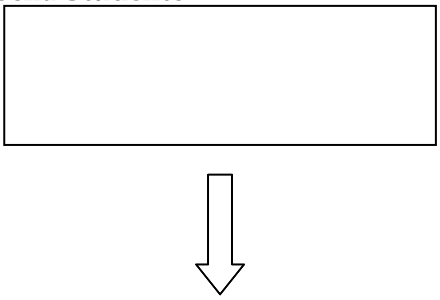
Questa interrogazione può essere vista, come abbiamo già detto, come una interrogazione composta dadue interrogazioni elementari una più interna chiamata **sottointerrogazione** ed una più esterna:

quella più interna: $\mathbf{T1} = \sigma_{\text{CodScuola1 = "NATF12000P"}}\ \mathbf{(Studente)}$

e quella più esterna: $\mathbf{T2} = \Pi_{\text{Cognome, Nome}}\ (\ \mathbf{T1}\ )$

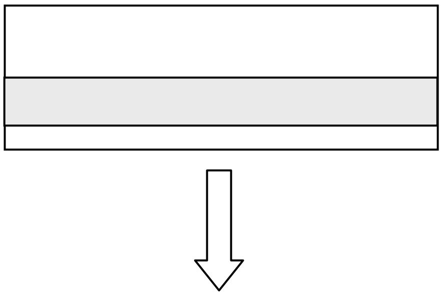
Utilizzando il modello grafico per le relazioni proposto dall'utilizzo della forma gabellare abbiamo avuto che

Tabella **Studente**



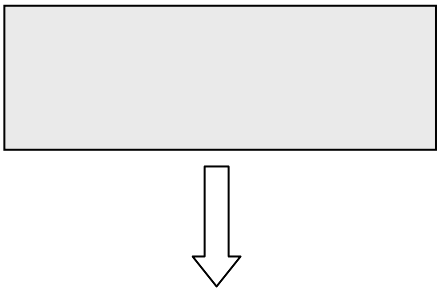
Studente	Matricola	Cognome	Nome	CodScuola1
	M001	Rossi	Paolo	NATF12000P
	M002	Bianchi	Aldo	NATF12000P
	M003	Verdi	Ada	NAEE12345Q
	M004	Neri	Maria	NAEE12345Q

Tabella **Studente**



Matricola	Cognome	Nome	CodScuola1
M001	Rossi	Paolo	NATF12000P
M002	Bianchi	Aldo	NATF12000P
M003	Verdi	Ada	NAEE12345Q
M004	Neri	Maria	NAEE12345Q

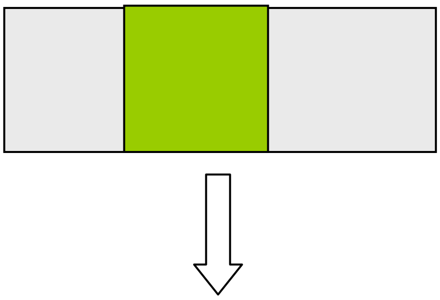
T1 = Tabella **Studente "ristretta" o "selezionata"**



T1

Matricola	Cognome	Nome	CodScuola1
M001	Rossi	Paolo	NATF12000P
M002	Bianchi	Aldo	NATF12000P

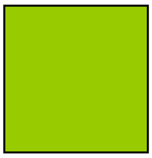
T2 = Tabella **T1 "proiettata"**



T1

Matricola	Cognome	Nome	CodScuola1
M001	**Rossi**	**Paolo**	NATF12000P
M002	**Bianchi**	**Aldo**	NATF12000P

Tabella **risultato finale**



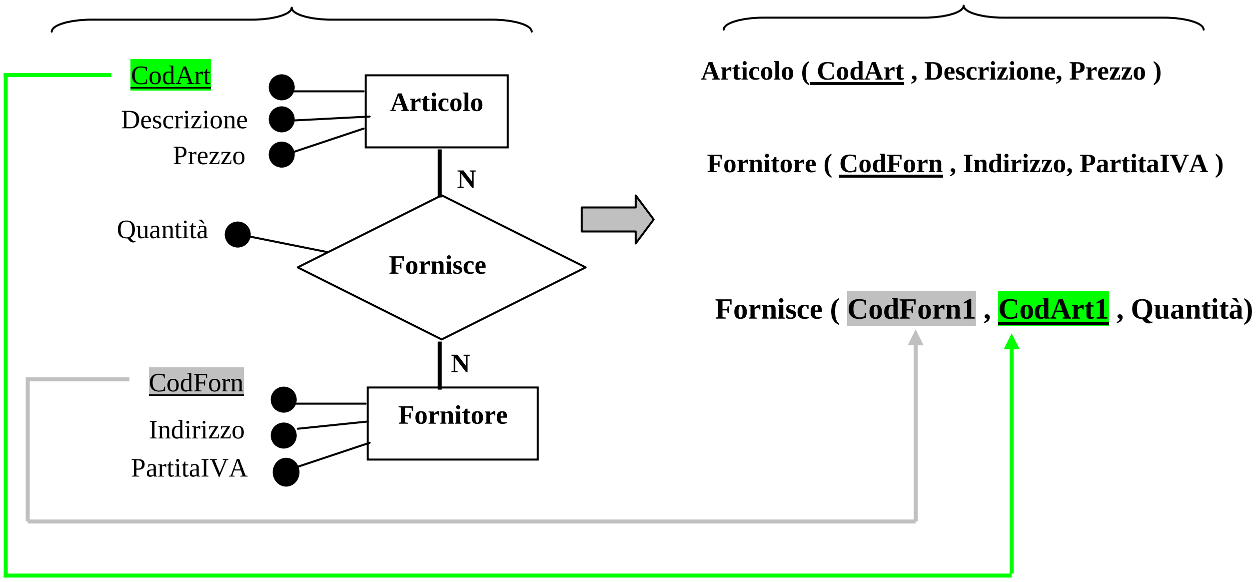
Risultato finale

Cognome	Nome
Rossi	**Paolo**
Bianchi	**Aldo**

Interrogazione di associazioni N.N senza considerare i vincoli di integrità

Aiutiamoci con un esempio

Diagramma ER *Schema relazionale*



QUERY1: Vogliamo conoscere i fornitori dell'articolo avente CodArt = "A04"

Per quanto detto prima, utilizzando gli operatori dell'algebra relazionale, dovrà essere

$$\textbf{Fornitore} \underset{\text{CodForn = CodForn1}}{\rhd\lhd} \Pi_{\text{CodForn1}} \left(\sigma_{\text{CodArt1 = "A04"}} (\textbf{Fornisce}) \right)$$

QUERY2: Vogliamo conoscere gli articoli forniti dal fornitore avente CodForn = "F07"

Per quanto detto prima, utilizzando gli operatori dell'algebra relazionale, dovrà essere

$$\textbf{Articolo} \underset{\text{CodArt = CodArt1}}{\rhd\lhd} \Pi_{\text{CodArt1}} \left(\sigma_{\text{CodForn1 = "F07"}} (\textbf{Fornisce}) \right)$$

Analizziamo in dettaglio la prima interrogazione proposta

$$\textbf{Fornitore} \underset{CodForn\,=\,CodForn1}{\bowtie} \Pi_{CodForn1} \left(\sigma_{CodArt1\,=\,\text{“A04”}} \left(\textbf{Fornisce}\right) \right)$$

Questa risulta composta dalle seguenti sottointerrogazioni:

$\textbf{T1} = \sigma_{CodArt1\,=\,\text{“A04”}} \left(\textbf{Fornisce}\right)$

$\textbf{T2} = \Pi_{CodForn1} \left(\textbf{T1} \right)$

$\textbf{T3} = \textbf{Fornitore} \underset{CodForn\,=\,CodForn1}{\bowtie} \textbf{T2}$

Applichiamo questa interrogazione alla seguente istanza di base di dati:

Fornitore (<u>CodForn</u>, Indirizzo, PartitaIVA)

Chiave primaria

Fornitore	**CodForn**	**Indirizzo**	**PartitaIVA**
	F03	Via Po, 5	001234
	F07	Via Bari, 5	001345
	F16	Via Loi, 1	001333

Articolo (<u>CodArt</u>, Descrizione, Prezzo)

Chiave primaria

Articolo	**CodArt**	**Descrizione**	**Prezzo**
	A01	Batteria	100,00
	A04	Antenna	75,00
	A12	Radiatore	56,00

Fornisce (<u>CodForn1</u>, <u>CodArt1</u>, Quantità)

Chiave esterna **Chiave esterna**

fornisce	**CodForn1**	**CodArt1**	**Quantità**
	F03	A01	200
	F03	A04	150
	F16	A04	250
	F07	A12	100

$$\textbf{T1} = \sigma_{\text{CodArt1} = \text{“A04”}}\ (\textbf{Fornisce})$$

Fornisce	CodForn1	CodArt1	Quantità
	F03	A01	200
	F03	**A04**	150
	F16	**A04**	250
	F07	A12	100

T1	CodForn1	CodArt1	Quantità
	F03	**A04**	150
	F16	**A04**	250

$$\textbf{T2} = \Pi_{\text{CodForn1}}\ (\textbf{T1})$$

T2	CodForn1
	F03
	F16

Fornitore	CodForn	Indirizzo	PartitaIVA
	F03	Via Po, 5	001234
	F07	Via Bari, 5	001345
	F16	Via Loi, 1	001333

$$\textbf{T3} = \textbf{Fornitore} \bowtie_{\text{CodForn} = \text{CodForn1}} \textbf{T2}$$

Dal prodotto cartesiano delle relazioni Fornitore e T2 ossia

Fornitore X T2

si devono prelevano solo le righe aventi valori uguali in quel campo ossia in questo caso la prima e l'ultima.

CodForn	Indirizzo	PartitaIVA	CodForn1
F03	**Via Po, 5**	**001234**	**F03**
F03	Via Po, 5	001234	F16
F07	Via Bari, 5	001345	F03
F07	Via Bari, 5	001345	F16
F16	Via Loi, 1	001333	F03
F16	**Via Loi, 1**	**001333**	**F16**

Il risultato finale è la seguente tabella:

CodForn	Indirizzo	PartitaIVA	CodForn1
F03	**Via Po, 5**	**001234**	**F03**
F16	**Via Loi, 1**	**001333**	**F16**

Si provi a titolo di esempio a fare l'analogo discorso per l'interrogazione

$$\textbf{Articolo} \bowtie_{\text{CodArt} = \text{CodArt1}} \Pi_{\text{CodArt1}}\ \left(\sigma_{\text{CodForn1} = \text{“F07”}}\ (\textbf{Fornisce})\right)$$

6) La normalizzazione delle relazioni

DEF: Una **forma normale** è una proprietà di uno *schema relazionale* che ne garantisce la qualità misurata in assenza di determinati difetti.
DEF: La **normalizzazione** è un procedimento che serve a trasformare uno schema che presenta **anomalie** (*schema non normalizzato*) in uno **equivalente** (con lo stesso contenuto informativo) in cui tali anomalie sono state eliminate (*schema normalizzato*).

Quando uno schema relazionale non è normalizzato può comportare **comportamenti non desiderati** che possono compromettere le operazioni di congruenza durante le operazioni di:
- *inserimento* dei dati;
-*aggiornamento* dei dati;
-*cancellazione* dei dati;

Vediamo quali anomalie si possono avere considerando il seguente esempio esplicativo.

Esempio: Consideriamo la seguente relazione ***Magazzino*** *che rappresenta lo schema di un magazzino di accessori per l'auto. In questo schema un cliente può ordinare più accessori ed ogni accessorio può essere ordinato da più clienti.*
La relazione è rappresentata dalla seguente tabella:

Magazzino	**CodCli**	**Indirizzo**	**Città**	**Cap**	**CodAcc**	**Descrizione**	**Prezzo**	**Quantità**
	C01	Via Po, 23	Pisa	56100	**M03**	Batteria	100,00	3
	C01	Via Po, 23	Pisa	56100	**M12**	Radiatore	75,00	1
	C01	Via Po, 23	Pisa	56100	**M04**	Antenna	25,00	3
	C02	Via Mori, 1	Napoli	80100	**M03**	Batteria	100,00	2
	C02	Via Mori, 1	Napoli	80100	**M12**	Radiatore	75,00	1
	C03	Via Ugo, 8	Roma	00100	**M03**	Batteria	100,00	2

La chiave della relazione è rappresentata dalla coppia di attributi (CodCli, CodAcc)

a) anomalie di inserimento:
- non è possibile inserire un nuovo *cliente* senza inserire i dati relativi agli accessori ordinati
- non è possibile inserire un nuovo *accessorio* senza inserire i dati relativi al cliente

b) anomalie di aggiornamento:
- per modificare l'indirizzo di un *cliente* occorre modificare tutte le ennuple in cui compare
- per modificare la descrizione di un *accessorio* occorre modificare tutte le ennuple in cui compare

Se le modifiche fossero parziali si lascerebbe la base dei dati in uno stato detto **inconsistente**

c) anomalie di cancellazione:
- cancellando la ennupla (C01, M04) si perdono le informazioni relative all'accessorio M04 (Batteria)
- cancellando la ennupla (C03, M03) si perdono le informazioni relative al cliente C03

Queste anomalie si verificano perché abbiamo rappresentato informazioni eterogenee tra loro con un'unica relazione.

Abbiamo infatti raggruppato in un'unica relazione informazioni relative ad:
- gli **accessori** presenti in magazzino;
- i dati anagrafici dei **clienti**;
- gli **ordini** dei clienti relativi a determinati accessori.

Il **processo di normalizzazione** elimina tali **anomalie** effettuando una serie di trasformazioni successive delle relazioni di partenza di uno schema relazionale ottenendo altre relazioni che a seconda del tipo di trasformazione applicata, possono rispondere a diversi livelli di *"bontà"* dette **forme normali**.
Esistono molte forme normali per uno schema relazionale:
- prima forma normale o **1FN**;
- seconda forma normale o **2FN**;
- terza forma normale o **3FN** con definizione alternativa di di Boyce-Codd o **BCFN**;
- quarta forma normale o **4FN**;
- quinta forma normale o **5FN**.

Per i nostri scopi è sufficiente applicare il processo di normalizzazione per ottenere uno schema relazionale in **terza forma normale o 3FN** e vedremo come ottenerlo attraverso lo studio delle dipendenze funzionali.

Schema relazionale **non normalizzato** ⇨
Dipendenze funzionali ⇨
Processo di normalizzazione ⇨ Schema relazionale **normalizzato**

PRIMA FORMA NORMALE o 1FN (o forma ATOMICA)

DEF: Diremo che una relazione **R** è in **prima forma normale** o **1FN** quando rispetta i **requisiti fondamentali** del modello relazionale che sono:
- i **valori di un attributo** (di una colonna) sono <u>dello stesso tipo</u> ovvero appartengono allo stesso dominio;
- i **valori di una ennupla** (di una riga) <u>sono diversi</u> da quelli delle altre ennuple ovvero non possono esistere due ennuple uguali;
- l'**ordine delle ennuple** <u>è irrilevante</u>;
- gli **attributi** sono di tipo **elementare** ossia:
  non possono essere ulteriormente *scomposti* in attributi più semplici (no attributi multipli);
  non possono essere *composti* da gruppi di attributi ripetuti (no attributi composti o aggregati).

Vedi esempio pag 74-75-76

Dipendenze funzionali

DEF: Data una relazione **R** ed un insieme $\mathbf{X} = \{ X_1, X_2, \dots, X_N \}$ di **R** si dice che un attributo **Y** di **R dipende funzionalmente da X** e si scrive:

$$X_1, X_2, \dots, X_N \rightarrow Y$$

se e solo se i valori degli attributi di **X** determinano univocamente il valore dell'attributo **Y** per ogni istanza della relazione **R**.
Si dice anche che **X determina Y**.

Nell'esempio precedente della relazione Magazzino possiamo individuare le seguenti dipendenze funzionali:

1) ***CodCli*** **→** ***Indirizzo:*** *L'indirizzo dipende funzionalmente da quel determinato cliente*
2) ***CodCli*** **→** ***Città:*** *La città dipende funzionalmente da quel determinato cliente*
3) ***CodCli*** **→** ***Cap:*** *Il Cap dipende funzionalmente da quel determinato cliente*

   Riassumendo ***i breve possiamo dire che CodCli → Indirizzo, Città, Cap***

4) ***CodAcc*** **→** ***Descrizione:*** *La descrizione dipende funzionalmente da quel determinato accessorio*
5) ***CodAcc*** **→** ***Prezzo:*** *Il prezzo dipende funzionalmente da quel determinato accessorio*

   Riassumendo ***i breve possiamo dire che CodAcc → Descrizione, Prezzo***

6) ***CodCli, CodAcc*** **→** ***Quantità:*** *La quantità ordinata di un determinato accessorio da parte di un cliente dipende funzionalmente dal cliente e dall'accessorio*

N.B.

a) Le *dipendenze funzionali* generalizzano il concetto di **chiave** ossia **tutti gli attributi non chiave dipendono funzionalmente dagli attributi chiave**.

b) Le **anomalie** sono deducibili dalle **dipendenze funzionali**: infatti un cliente ha un unico indirizzo *(CodCli → Indirizzo)* un accessorio *ha un'unica descrizione (CodAcc→ Descrizione)* e questo causa come abbiamo visto anomalie, mentre una quantità ha un unico cliente ed un unico accessorio *(CodCli, CodAcc→ Quantità)* e questo non causa alcuna anomalia.
La differenza si ha perché l'attributo *Quantità* **dipende funzionalmente da tutti gli attributi che formano la chiave, mentre gli attributi** *Descrizione* **e** *Prezzo* **dipendono funzionalmente solo da una parte della chiave.**

SECONDA FORMA NORMALE o 2FN

DEF: Diremo che una relazione **R** è in **seconda forma normale** o **2FN** se non esistono attributi dipendenti solo da una parte della chiave ossia *dipendenti parzialmente* dalla chiave.

Questa forma normale richiede che tutti gli attributi della relazione siano **omogenei** nel senso che devono essere tutte proprietà associate direttamente alla chiave.
Per trasformare una relazione in 2FN si procede **decomponendola** sulla base delle dipendenze funzionali al fine di separare le proprietà eterogenee.
Vedi esempio pag 74-75-76
Le relazioni in 2FN possono ancora essere esposte ad anomalie in quanto possono presentare delle ridondanze

TERZA FORMA NORMALE o 3FN

DEF: Diremo che una relazione **R** è in **terza forma normale** o **3FN** se per ogni possibile chiave di **R** accade che:
- **R è in 2FN** ossia non esistono attributi non chiave che dipendono solo da una parte della chiave;
- non esistono attributi non chiave che dipendono **transitivamente** dalla chiave ossia non esistono attributi non chiave che dipendono da altri attributi non chiave.

Vedi esempio pag 74-75-76

Per trasformare una relazione in **3FN** si crea una nuova relazione per ogni gruppo di *attributi non chiave* coinvolti nella dipendenza funzionale *con attributi non chiave*.

Questa definizione di **3FN** può essere data in forma equivalente con la definizione di forma normale di Boyce-Codd o BCNF..

Sebbene esistano la 4FN e la 5FN non vengono considerate in questo contesto.
La forma normale obbligatoria è la prima ossia la 1FN mentre le altre servono per separare meglio concetti eterogenei racchiudendoli in tabelle che al loro interno siano il più possibile omogenee.
Come si è potuto vedere passando dalla 1FN alle altre si eliminano le anomalie ma si introducono ridondanze (ripetizioni) nella base di dati.

Vale il seguente annodamento delle forme normali.