

## 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 **relazionale puro**.

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 o modello 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.

#### **PREMESSA IMPORTANTE**

*Prima di procedere con le definizioni di relazione e di tutte le sue caratteristiche secondo il modello relazionale, al fine di rendere lo studio e l'assimilazione di tali contenuti quanto più agevoli possibile si raccomanda la visione della seguente presentazione powerpoint presente sul sito del docente:*

**Presentazione powerpoint (sezione TEORIA file PDF 15-A)**

<http://www.riochierego.it/mobile/docs/quinta/teo/EXTRA-Teoria-degli-Insiemi.pdf>

#### **ORA E' POSSIBILE CONTINUARE.....**

**DEF:** Secondo la teoria degli insiemi una **relazione R** è un **sottoinsieme finito del prodotto cartesiano** di una sequenza di **n** insiemi  $D_1 \times D_2 \times \dots \times D_n$  (non necessariamente distinti).

Una **relazione R** è costituita da tutte le enuple, appartenenti al prodotto cartesiano

$D_1 \times D_2 \times \dots \times D_n$ , che soddisfano il suo **predicato** (enunciato semplice o composto)

Quindi, utilizzando il concetto di "inclusione" tra insiemi, è del tutto evidente che risulterà:

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

dove:

- **n** (con  $n > 1$ ) è detto **GRADO** della relazione **R** ed è indicato con la scritta **Grado(R)**;

- gli insiemi  $D_1, D_2, \dots, D_n$  sono detti **DOMINI** della relazione **R** 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 che le sarà assegnato seguito dalla 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> identificano i tipi di dato da assegnare agli attributi come ad esempio carattere, stringa, intero, reale, booleano, data, tempo, timestamp.

**N.B. In questa fase di mapping relazionale, per ragioni di semplificazione, all'interno dello schema di una relazione ometteremo l'indicazione dei tipi di dato per gli attributi.**

Per **convenzione** scriveremo sia i nomi delle relazioni sia i nome degli attributi **con la sola iniziale in maiuscolo.**

*Esempio: consideriamo il seguente schema della relazione "Dipendente" utilizzato per rappresentare alcune caratteristiche di un lavoratore di una certa azienda:*

*Dipendente (CodFisc: Stringa(16), Cognome: Stringa(30), Nome: Stringa(20), Livello: Intero(1), Stipendio: Reale(8,2))*

Spesso per brevità di esposizione scriveremo:

*Dipendente (CodFisc, Cognome, Nome, Livello, Stipendio)*

**DEF:** Secondo la teoria degli insiemi gli elementi (le istanze) appartenenti alla relazione **R** sono chiamati **ennuple** o **n-ple** e vengono indicati in modo generale con:

$(d_1, d_2, \dots, d_n)$

dove  $d_1 \in D_1, d_2 \in D_2, \dots, d_n \in D_n$

Chiameremo **istanza di una relazione R** l'insieme delle sue ennuple in un determinato istante di tempo.

**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)**.

**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.**

**N.B. Coerentemente con la definizione di insieme, una relazione non può mai contenere ennuple uguali.**

**Modalità di rappresentazione di una relazione**

Poiché è definita come sottoinsieme del prodotto cartesiano dei suoi domini, una **relazione** (così come accade per ogni tipo di **insieme** matematico o non) può essere rappresentata:

- a) per elencazione;**
- b) in forma insiemistica** (diagramma di **Eulero-Venn**);
- c) tramite una proprietà caratteristica**

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

*Dipendente (CodFisc, Cognome, Nome, Livello, Stipendio)*

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:

*Dipendente (CodFisc, Cognome, Nome, Livello, Stipendio) =*

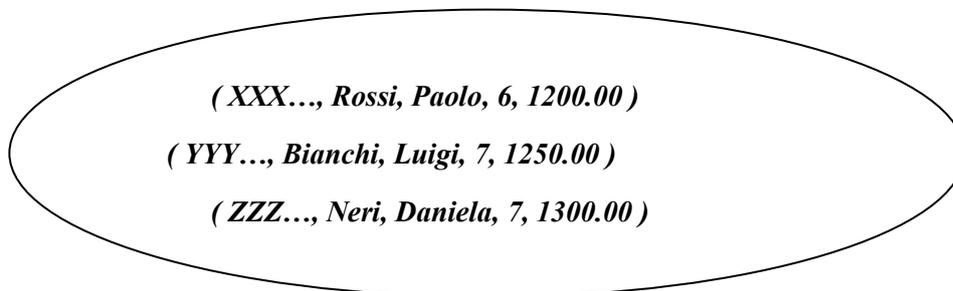
*ennupla* → {  
 (XXXXXXXXXXXXXXXXXX, Rossi, Paolo, 6, 1200.00),  
 (YYYYYYYYYYYYYYYYYYYY, Bianchi, Luigi, 7, 1250.00),  
 (ZZZZZZZZZZZZZZZZZZZZ, Neri, Daniela, 7, 1300.00)  
 }

← istanza della relazione composta da tre ennuple (o istanze) con **Card (Dipendente) = 3**

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

Ritornando al nostro esempio scriveremo:

**Dipendente (CodFisc, Cognome, Nome, Livello, Stipendio)**



**c) rappresentazione tramite una proprietà caratteristica:** è possibile rappresentare un'istanza di una relazione indicando una determinata caratteristica che contraddistingue le sue ennuple.

Ritornando al nostro esempio scriveremo:

**Dipendente** = { $x$  |  $x$  è un dipendente di una certa azienda di cui considero solo le "qualità"  
**CodFisc, Cognome, Nome, Livello, Stipendio**}

Per le relazioni tuttavia esiste una forma di rappresentazione "speciale" molto più intuitiva e potente rispetto alle precedenti:

**d) rappresentazione in forma tabellare:** è possibile rappresentare un'istanza di una relazione utilizzando una tabella costituita da **m righe** ed **n colonne** dove ovviamente il valore **m** rappresenta la **cardinalità** ed il valore **n** rappresenta il **grado** della relazione (rispettivamente **Card(R)** e **Grado(R)**)

Ogni **riga** rappresenta una **ennupla** ed ogni **colonna** rappresenta la **sequenza dei valori** assunti dal corrispondente attributo.

Ritornando al nostro esempio scriveremo relativamente alla relazione **Dipendente**:

$$n = \text{Grado (Dipendente)} = 5$$

<b>CodFisc</b>	<b>Cognome</b>	<b>Nome</b>	<b>Livello</b>	<b>Stipendio</b>
XXX...	Rossi	Paolo	6	1200.00
YYY...	Bianchi	Luigi	7	1250.00
ZZZ...	Neri	Daniela	7	1300.00

} **m = Card (Dipendente) = 3**

### Chiavi di una relazione

Anche nel modello relazionale occorre specificare come vincolo quello relativo alla presenza di una **chiave primaria** per ciascuna relazione.

**DEF:** Si definisce **chiave candidata o superchiave** di una relazione **R** un insieme non vuoto **K** di **attributi di R i cui valori** individuano univocamente ogni ennupla per ciascuna possibile istanza della relazione **R**.

**DEF:** Si definisce **chiave primaria o PRIMARY KEY (o PK)** 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.** In ogni relazione **R** esiste sempre almeno una *chiave candidata* (quella formata da tutti gli attributi di **R**) tenendo presente che non potranno mai esistere due ennuple uguali (ossia con gli stessi valori per tutti gli attributi previsti).

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

**DEF** (alternativa diretta): Si definisce **chiave primaria o PRIMARY KEY (o PK)** di una relazione **R** un insieme **minimo (o non ridondante) K** non vuoto di **attributi di R**, **i cui valori** individuano **univocamente** ogni ennupla per ciascuna possibile istanza della relazione **R**.

*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, NumDocumento, TipoDocumento, DataDocumento)*

Sono possibili chiavi candidate:

1. (NumProgressivo)
2. (Cognome, Nome, DataNascita)
3. (NumDocumento, TipoDocumento, DataDocumento)
4. (NumProgressivo, Cognome, Nome, DataNascita, NumDocumento, TipoDocumento, DataDocumento)

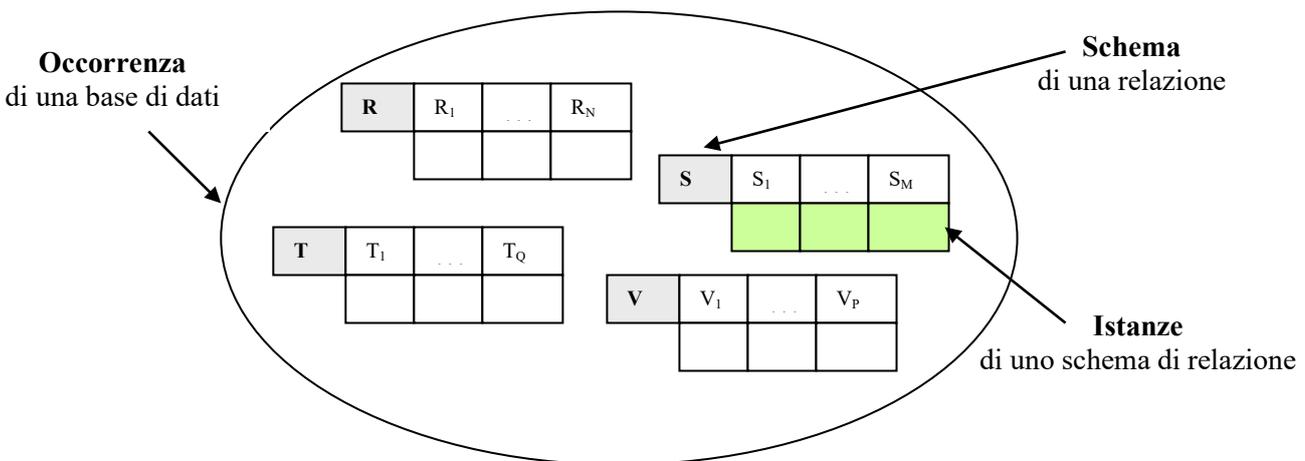
*Fra le possibili chiavi candidate è stata scelta come chiave primaria (NumProgressivo) perché è costituita dal minor numero di attributi.*

**N.B.** Spesso nella progettazione di basi dati quando non si è in grado di determinare un attributo tra quelli scelti 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*.

## **DAL DIAGRAMMA ER ALLO SCHEMA LOGICO 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” (o derivazione) delle *entità* e delle *associazioni* individuate comprese di *attributi*, trasformandole opportunamente in *relazioni* del *modello relazionale* (output della fase di progettazione logica) che tutte insieme costituiranno il cosiddetto **schema logico relazionale** o più semplicemente **schema relazionale**.

Lo **schema relazionale** si ricava dunque dal diagramma ER applicando alcune semplici **regole di “derivazione”** o **“mapping”** per rappresentarne tutti gli oggetti in esso rappresentabili ossia:

- **entità ed attributi** (elementari);
- **associazioni binarie di molteplicità 1:N oppure N:1** (eventualmente ricorsive);
- **associazioni binarie di molteplicità 1:1** (eventualmente ricorsive);
- **associazioni binarie di molteplicità N:N** (eventualmente ricorsive);
- **vincoli di integrità** (impliciti ed espliciti).

**N.B. Ricordiamo che nel Diagramma ER l'eventuale presenza di attributi aggregati (o composti) e multipli è stata evitata utilizzando opportune tecniche di trasformazione illustrate in precedenza**

**Anche la presenza all'interno di un Diagramma ER di eventuali associazioni per generalizzazione o ISA è stata evitata attraverso l'utilizzo di una delle tre differenti strategie di risoluzione previste a seconda della sua tipologia**

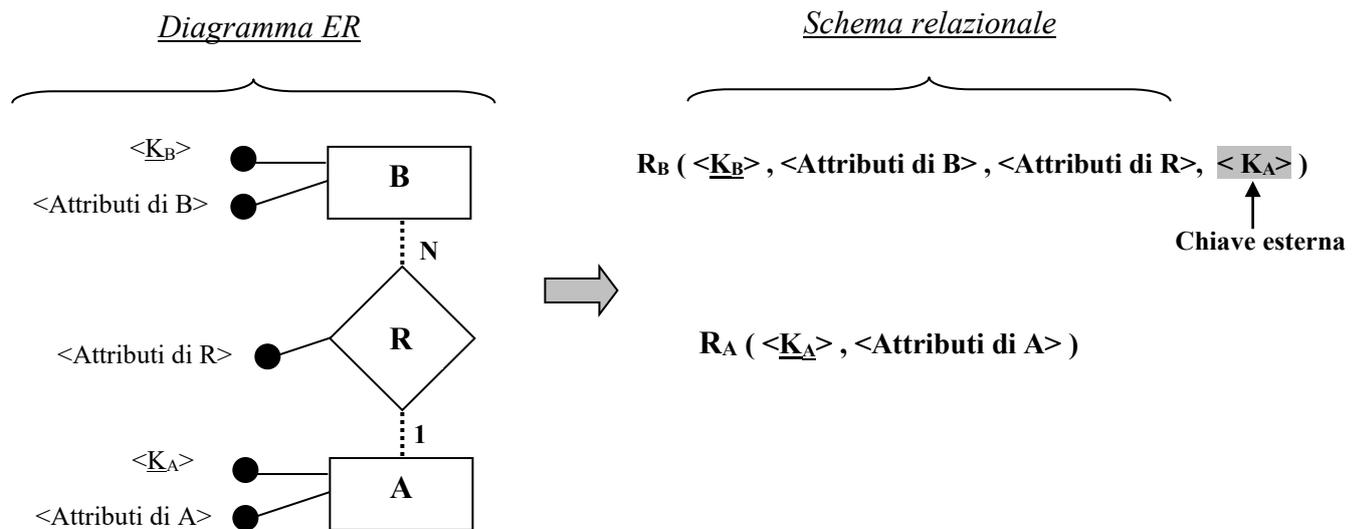
### ***PREMESSA IMPORTANTE***

***Prima di procedere dettagliatamente ad illustrare le regole di derivazione o mapping si consiglia di consultare il manuale completo contenente tali regole presente sul sito del docente***

**MANUALE REGOLE DI DERIVAZIONE o MAPPING COMPLETO DI DATI DI ESEMPIO (sezione LABORATORIO file PDF 15-A)**

<http://www.riochierego.it/mobile/docs/quinta/lab/MAPPING-RELAZIONALE-ASSOCIAZIONI.pdf>



**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 imponga l'esistenza nella relazione  $R_B$  di una chiave esterna uguale alla chiave primaria  $K_A$ .

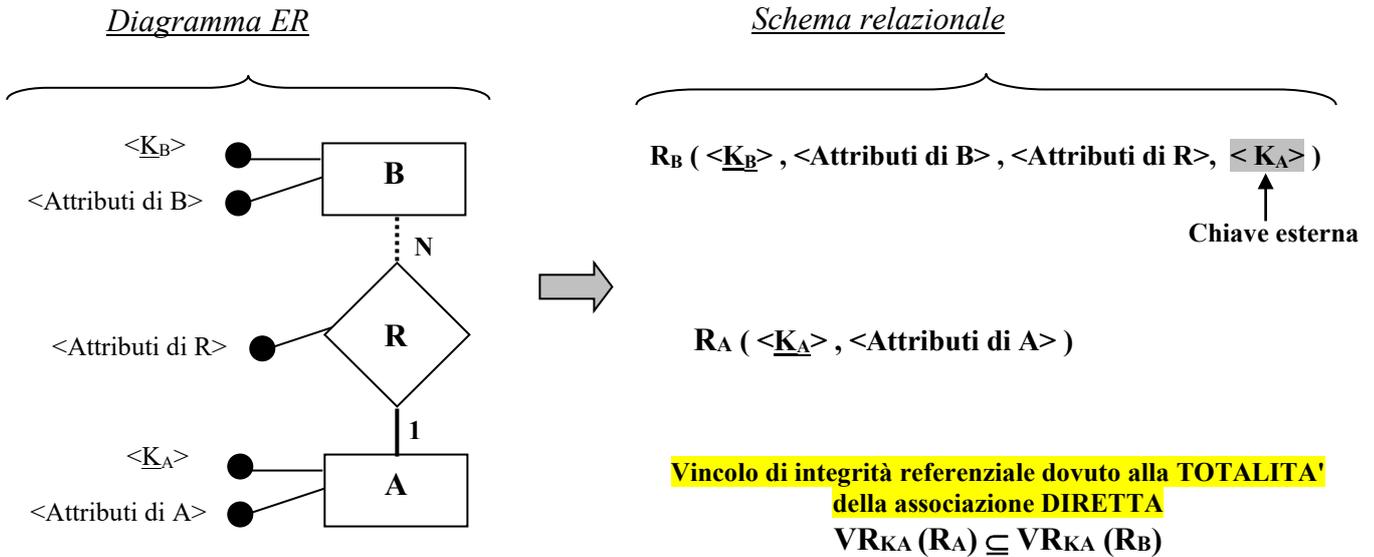
In altre parole il vincolo di integrità da imporre è che.

$$VR_{K_A}(R_A) \subseteq VR_{K_A}(R_B)$$

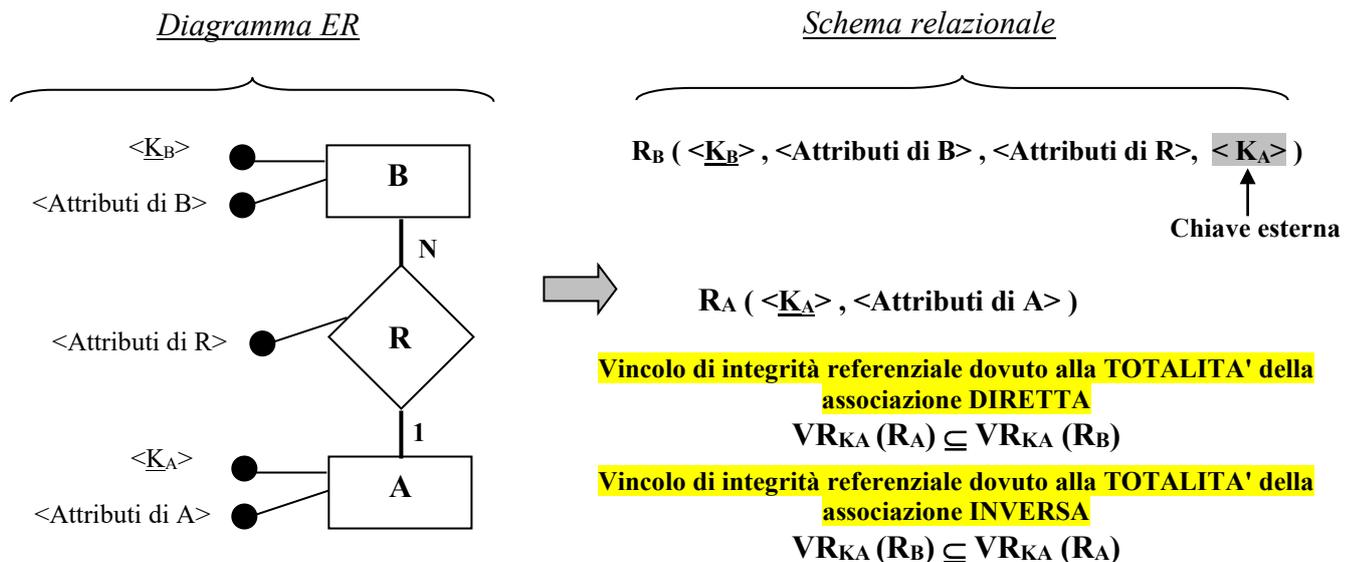
Esso significa che tutti i valori dell'attributo chiave primaria  $K_A$  della relazione  $R_A$  devono essere presenti nell'attributo chiave esterna  $K_A$  della relazione  $R_B$ .

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**



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



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.

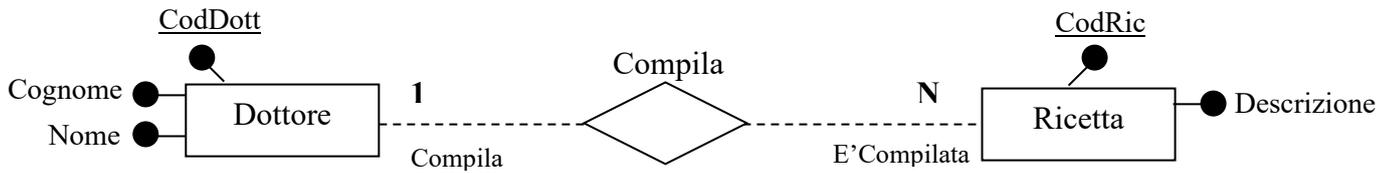
**Esso significa che tutti i valori dell'attributo chiave esterna  $K_A$  della relazione  $R_B$  devono essere presenti nell'attributo chiave primaria  $K_A$  della relazione  $R_A$**

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** (CodDott, Cognome, Nome)

Chiave primaria

<u>CodDott</u>	Cognome	Nome
D01	ROSSI	MARIO
D02	BIANCHI	GIULIO
D03	VERDI	LUIGI
<b>D04</b>	<b>NERI</b>	<b>DANIELA</b>

**Ricetta** (CodRic, Descrizione, CodDott)

Chiave primaria

<u>CodRic</u>	Descrizione	<u>CodDott</u>
R01	Ves	D01
R02	Radiografia	D02
R03	Sangue	D03
<b>R04</b>	<b>Urina</b>	<b>NULL</b>
R05	Tiroide	D02

Chiave esterna su relazione Dottore

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 lo stesso nome (eventualmente con l’aggiunta di un progressivo numerico) 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 con codice "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.

**Mapping relazionale delle associazioni binarie di molteplicità 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

$$VR_{KA}(R_B) = VR_{KA}(R_A)$$

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 (CodDir, Cognome, Nome)*

Chiave primaria

CodDir	Cognome	Nome
D01	SESSA	DARIO
D02	BIANCHI	GIULIO

*Scuola (CodS, Denominazione, CodDir)*

Chiave primaria

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

Chiave esterna su relazione DS

Vi sono anche i seguenti due vincoli di integrità referenziali

(1)  $VR_{CodDir}(\text{DirigenteScolastico}) \subseteq VR_{CodDir}(\text{Scuola})$  **Vincolo di integrità referenziale (in breve VR) dovuto alla TOTALITA' dell'ASS. DIRETTA "Dirige"**

(2)  $VR_{CodDir}(\text{Scuola}) \subseteq VR_{CodDir}(\text{DirigenteScolastico})$  **Vincolo di integrità referenziale (in breve VR) dovuto alla TOTALITA' dell'ASS. INVERSA "E' Diretta"**

**b) Unica relazione**

*DirigenteScolastico-Scuola (CodDir, Cognome, Nome, CodS, Denominazione)*

N.B. Questo attributo perde il ruolo di chiave e potrebbe essere omissso

Chiave primaria

CodDir	Cognome	Nome	CodS	Denominazione
D01	MARTINO	TERESA	S02	ISIS "TASSINARI"
D02	BIANCHI	GIULIO	S01	ITI "GALVANI"

**Mapping relazionale delle associazioni binarie di molteplicità N:N**

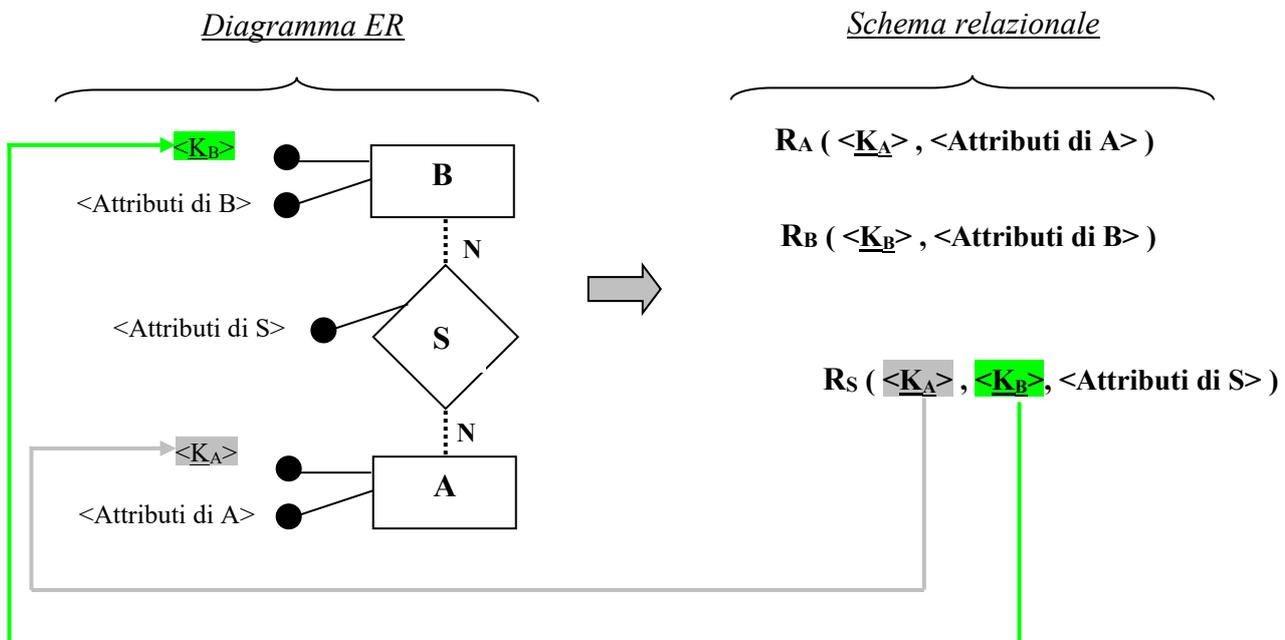
**REGOLA GENERALE:** Sia data una associazione **R** di tipo **N:N** tra due entità **A** e **B** con diretta ed inversa entrambe parziali.

Per “mappare” tale associazione nel modello relazionale occorre introdurre **tre relazioni** e **due vincolo di integrità referenziale** così costituiti:

- una relazione **R<sub>A</sub>** avente gli attributi di **A**;
- una relazione **R<sub>B</sub>** avente gli attributi di **B**;
- una relazione **R<sub>S</sub>** avente gli attributi chiave **K<sub>A</sub>** di **R<sub>A</sub>** e gli attributi chiave **K<sub>B</sub>** di **R<sub>B</sub>** come chiave primaria (quindi come minimo la relazione **R<sub>S</sub>** avrà due attributi);
- un vincolo di integrità che assicuri che ad ogni valore della chiave esterna **K<sub>A</sub>** presente in **R<sub>S</sub>** corrisponda un valore della chiave primaria **K<sub>A</sub>** della relazione **R<sub>A</sub>**;
- un vincolo di integrità che assicuri che ad ogni valore della chiave esterna **K<sub>B</sub>** presente in **R<sub>S</sub>** corrisponda un valore della chiave primaria **K<sub>B</sub>** della relazione **R<sub>B</sub>**.

Quindi ogni ennupla di **R<sub>S</sub>** 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**  
 (Nota Bene: Senza gestione dello “storico”)

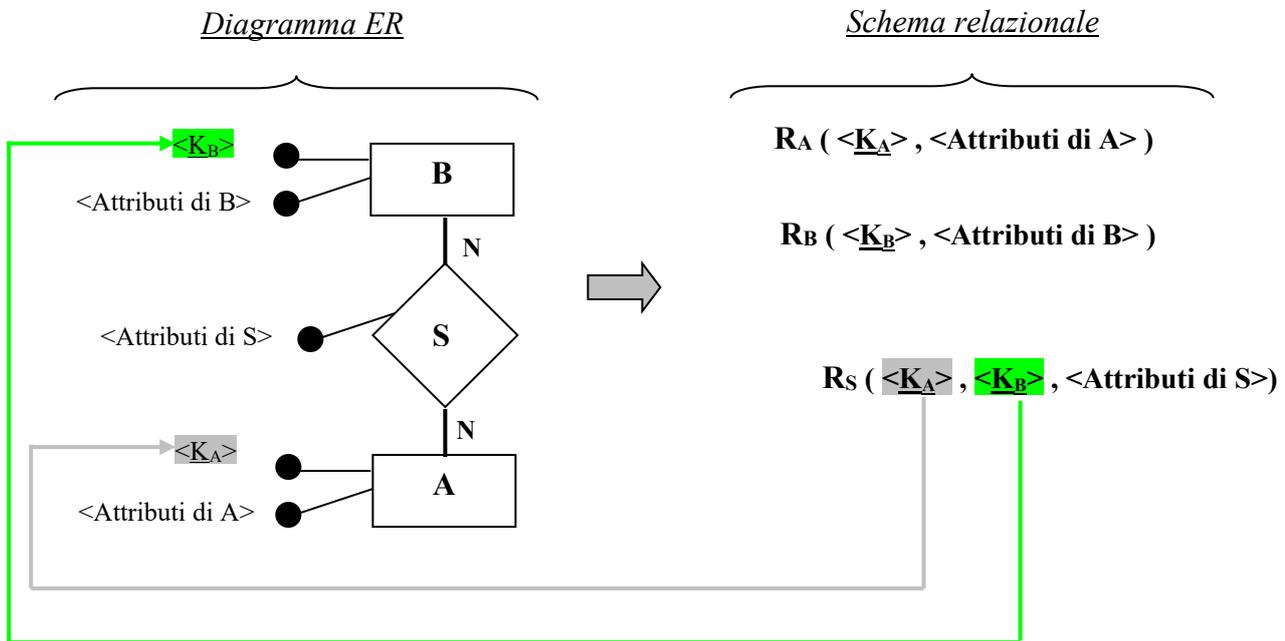


- $$\left\{ \begin{array}{l} (1) \text{ VR}_{K_A} (R_S) \subseteq \text{VR}_{K_A} (R_A) \\ (2) \text{ VR}_{K_B} (R_S) \subseteq \text{VR}_{K_B} (R_B) \end{array} \right. \begin{array}{l} \text{Vincoli di integrità referenziali (VR)} \\ \text{causati dal generico mapping relazionale} \\ \text{di un'associazione di molteplicità N:N} \end{array}$$

N.B. **IN ALTRE PAROLE** tali vincoli di integrità referenziale esprimono il fatto che **ogni chiave esterna** presente nella relazione **R<sub>S</sub>** appositamente creata, **DEVE FARE RIFERIMENTO** ossia **ASSUMERE NECESSARIAMENTE UNO** dei valori **GIA' PRESENTI** tra le chiavi primarie rispettivamente delle relazioni **R<sub>A</sub>** e **R<sub>B</sub>**

**IN QUESTO CASO** occorre notare che le chiavi esterne (FOREIGN KEY) **K<sub>A</sub>** e **K<sub>B</sub>** della relazione **R<sub>S</sub>** **SONO STATE SOTTOLINEATE** in quanto risultano entrambe far parte della **PRIMARY KEY** della relazione **R<sub>S</sub>** appositamente creata.

**MAPPING Associazione binaria N:N con associazione diretta (da A ad B) ed inversa (da B a A) TOTALI**  
 (Nota Bene: Senza gestione dello “storico”)



- (1)  $VR_{KA}(R_S) \subseteq VR_{KA}(R_A)$  **Vincoli di integrità referenziali (VR)**
- (2)  $VR_{KB}(R_S) \subseteq VR_{KB}(R_B)$  **causati dal generico mapping relazionale di un'associazione di molteplicità N:N**
- (3)  $VR_{KA}(R_A) \subseteq VR_{KA}(R_S)$  **Vincolo di integrità referenziale dovuto alla TOTALITA' della associazione DIRETTA (da A verso B)**
- (4)  $VR_{KB}(R_B) \subseteq VR_{KB}(R_S)$  **Vincolo di integrità referenziale dovuto alla TOTALITA' della associazione INVERSA (da B verso A)**

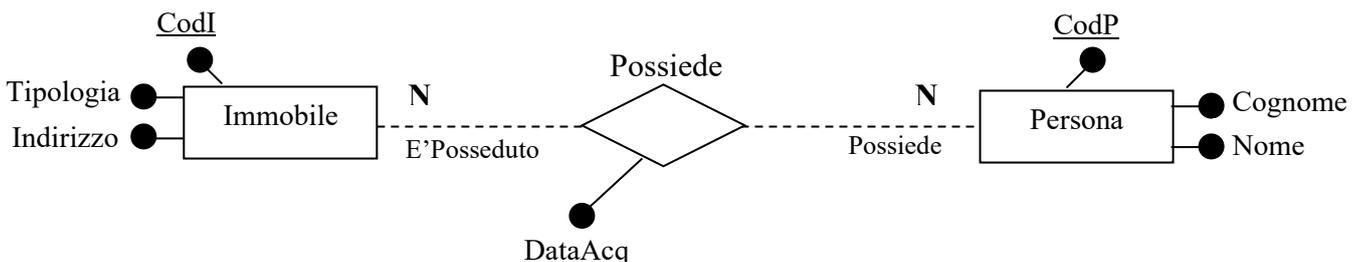
**IN ALTRE PAROLE.....**

Il vincolo di integrità referenziale (3) afferma che ad ogni ennupla presente nella relazione  $R_A$  deve corrispondere una ennupla presente nella relazione  $R_S$ .

Il vincolo di integrità referenziale (4) afferma che ad ogni ennupla presente nella relazione  $R_B$  deve corrispondere una ennupla presente nella relazione  $R_S$ .

*Esempio: Supponiamo che una persona può possedere nessuno o più immobili e che viceversa un immobile può essere posseduto da nessuna o più persone.*

*La situazione descritta può essere schematizzata utilizzando il seguente diagramma ER:*



Operando il mapping illustrato come visto si ha:

**Persona** (CodP, Cognome, Nome)

Chiave primaria

<u>CodP</u>	Cognome	Nome
P01	BIANCHI	MARIO
P05	VERDI	GIANNI
P13	ROSSI	MARIA

**Immobile** (CodI, Tipologia, Superficie)

Chiave primaria

<u>CodI</u>	Tipologia	Superficie
I01	Villa	120,00
I12	Attico	75,00
I24	Rudere	56,00

Per mantenere la correlazione tra Persona ed Immobile secondo il mapping previsto dobbiamo creare una nuova relazione che chiameremo **Possiede** utilizzando le chiavi primarie delle due relazioni **Persona** ed **Immobile** che diventano **INSIEME** chiave primaria della nuova relazione e che risultano **chiavi esterne** ciascuna su una delle due relazioni di partenza.

Chiave esterna su Persona

Chiave esterna su Immobile

<u>CodP</u>	<u>CodI</u>	DataAcq
P01	I01	01-12-2000
P05	I12	21-03-1998
P05	I01	01-12-2000
P01	I12	21-03-1998

**Possiede** (CodP, CodI, DataAcq)

Per convenzione nella nuova relazione per gli attributi chiavi esterne è meglio utilizzare lo stesso nome (eventualmente con l'aggiunta di un progressivo numerico) dell'attributo relativo alla chiave primaria nella relazione riferita

Vincoli di integrità referenziale per esprimere che ogni chiave esterna della relazione "Possiede" corrisponde ad una chiave primaria rispettivamente della relazione "Persona" e della relazione "Immobile"

- (1)  $VR_{CodP}(Possiede) \subseteq VR_{CodP}(Persona)$  **Vincoli di integrità referenziali (VR) causati dal generico mapping relazionale di un'associazione di molteplicità N:N**
- (2)  $VR_{CodI}(Immobile) \subseteq VR_{CodI}(Immobile)$  **Vincoli di integrità referenziali (VR) causati dal generico mapping relazionale di un'associazione di molteplicità N:N**
- (3)  $VR_{CodP}(Persona) \subseteq VR_{CodP}(Possiede)$  **Vincolo di integrità referenziale dovuto alla TOTALITA' della associazione DIRETTA "Possiede"**
- (4)  $VR_{CodI}(Immobile) \subseteq VR_{CodI}(Possiede)$  **Vincolo di integrità referenziale dovuto alla TOTALITA' della associazione INVERSA "E'Posseduto"**

In questo esempio si ha una inconsistenza dei dati

a) se dalla relazione **Persona** si cancella l'istanza relativa alla persona **P01**. Infatti in tal caso nella relazione **Possiede** continueremmo ad avere la chiave esterna **P01** alla quale non corrisponderebbe alcuna persona

b) se dalla relazione **Persona** si modifica il valore della chiave **P01** (ad esempio cambiandolo in **PXX**). Infatti in tal caso nella relazione **Possiede** continueremmo ad avere la chiave esterna **P01** alla quale non corrisponderebbe alcuna persona

Si avrà ancora un'inconsistenza dei dati

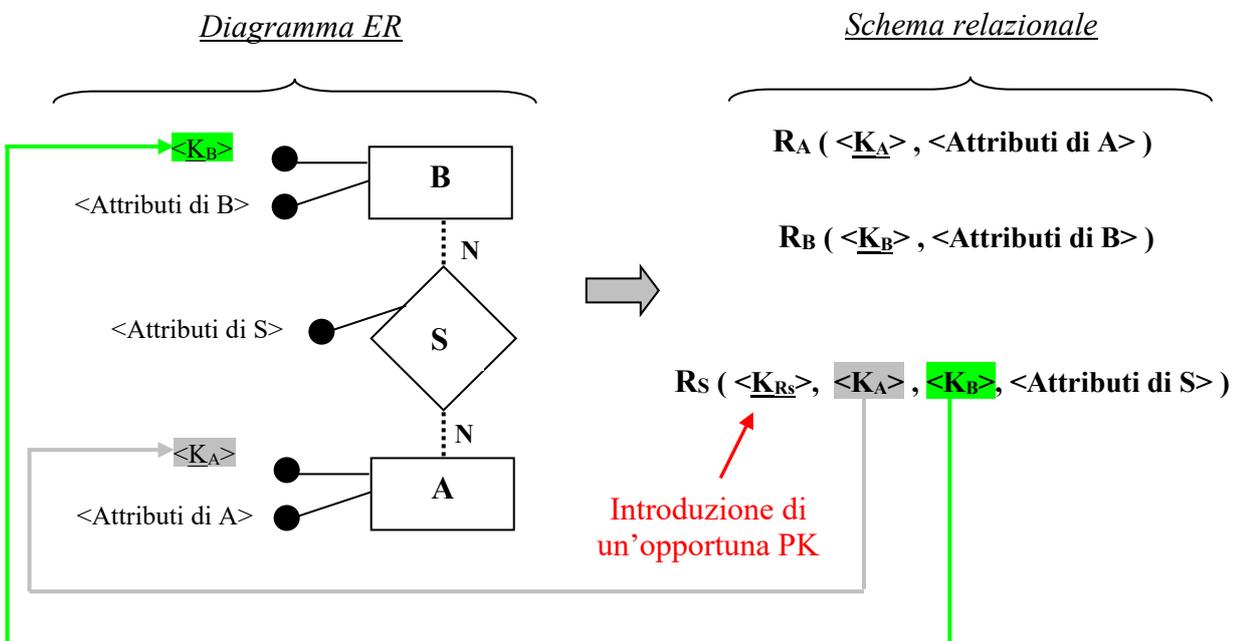
c) se dalla relazione **Immobil**e si cancella l'istanza relativa all'immobile **I12**. Infatti in tal caso nella relazione **Possiede** continueremmo ad avere la chiave esterna **I12** alla quale non corrisponderebbe alcun immobile

d) se dalla relazione **Immobil**e si modifica il valore della chiave **I12** (ad esempio cambiandolo in **PXX**). Infatti in tal caso nella relazione **Possiede** continueremmo ad avere la chiave esterna **I12** alla quale non corrisponderebbe alcun immobile

N.B. Per assicurare l'integrità referenziale prima di cancellare (o modificare) 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 il controllo dell'integrità referenziale è assicurata direttamente dal **DBMS** che prevede la possibilità di enunciare attraverso appositi linguaggi dichiarativi delle **regole di validazione**. Tali regole vengono conservate in appositi archivi detti **cataloghi delle regole**.

**MAPPING Associazione binaria N:N con associazione diretta (da A ad B) ed inversa (da B a A) PARZIALI**  
(Nota Bene: CON gestione dello "storico")



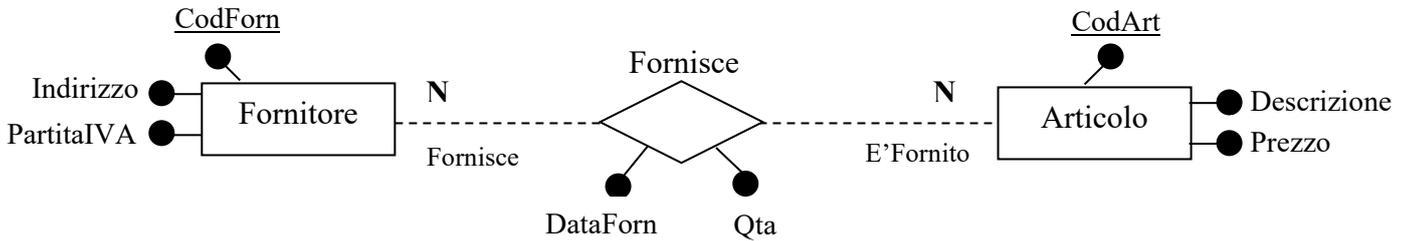
- (1)  $VR_{KA}(R_S) \subseteq VR_{KA}(R_A)$  **Vincoli di integrità referenziali (VR)**
- (2)  $VR_{KB}(R_S) \subseteq VR_{KB}(R_B)$  **causati dal generico mapping relazionale di un'associazione di molteplicità N:N**

N.B. ANCHE IN QUESTO CASO tali vincoli di integrità referenziale esprimono il fatto che ogni chiave esterna presente nella relazione **Rs** appositamente creata, **DEVE FARE RIFERIMENTO** ossia **ASSUMERE NECESSARIAMENTE UNO** dei valori **GIA' PRESENTI** tra le chiavi primarie rispettivamente delle relazioni **RA** e **RB**

N.B. Occorre notare che le chiavi esterne (FOREIGN KEY) **KA** e **KB** della relazione **Rs** **NON SONO PIU' sottolineate** in quanto **NON SONO** più **PRIMARY KEY** della relazione **Rs** in quanto con la gestione dello storico è ovviamente possibile che esistano più coppie di valori uguali

Esempio: Supponiamo che un certo fornitore possa fornire uno o più articoli (in date e quantità differenti) e che viceversa un certo articolo possa essere fornito da uno o più fornitori (in date e quantità differenti)

La situazione descritta può essere schematizzata utilizzando il seguente diagramma ER:



**N.B. Operando il mapping SENZA GESTIONE DELLO STORICO come visto finora si avrebbe la situazione già illustrata in precedenza:**

**Fornitore** (CodForn, Indirizzo, PartitaIVA)

Chiave primaria

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

**Articolo** (CodArt, Descrizione, Prezzo)

Chiave primaria

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

**Fornisce** (CodForn1, CodArt1, DataForn, Qta)

Chiavi esterne

CodForn1	CodArt1	DataForn	Qta
F03	A01	01-01-2019	10
F03	A04	11-02-2019	25
F16	A04	13-02-2019	20
F03	A01	14-04-2019	30

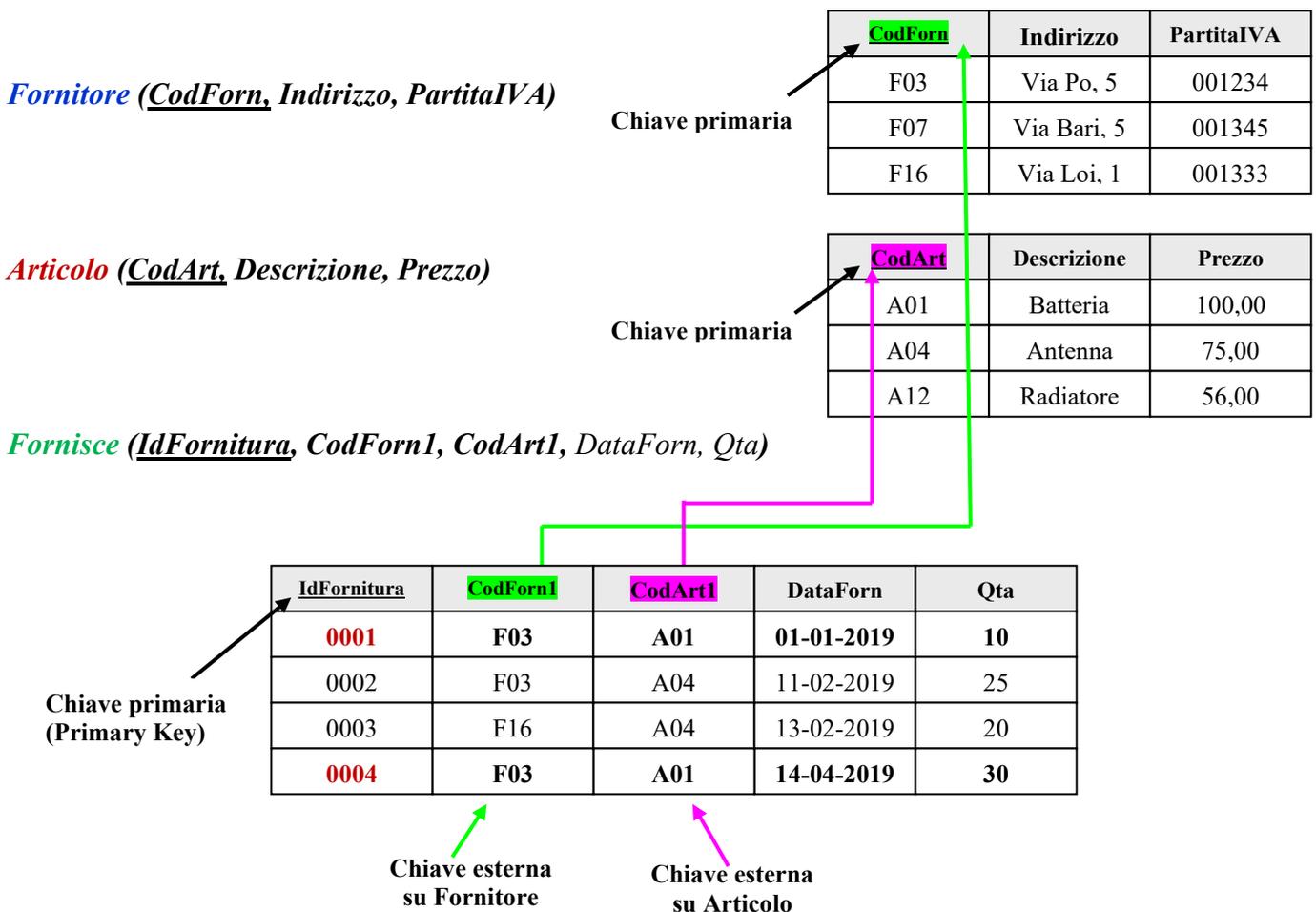
**!!! ERRORE !!!**

Il mapping relazionale applicato in precedenza (ossia SENZA GESTIONE DELLO STORICO) renderebbe impossibile l'evento (richiesto dal diagramma ER) che un fornitore (chiave **F03**) fornisca uno stesso prodotto (chiave **A01**) in una data differente ed in quantità differenti dalle consegne già effettuate senza incorrere in una **violazione del vincolo di chiave primaria (chiave primaria duplicata)**

Quindi per mantenere la correlazione tra Fornitore ed Articolo gestendo lo "storico" delle consegne effettuate, la nuova relazione non potrà **usare le chiavi esterne** alle due relazioni **Fornitore** ed **Articolo** per costruire la propria **chiave primaria**.

Queste ultime ovviamente devono comunque obbligatoriamente essere presenti **nella nuova relazione da creare** svolgendo il proprio ruolo (ossia avere i propri valori presenti in ciascuna delle due relazioni di partenza) **ma occorrerà inserire in essa una chiave primaria "indipendente" da esse** per distinguere le ennuple.

## Mapping relazionale alternativo soluzione per la "gestione dello storico"



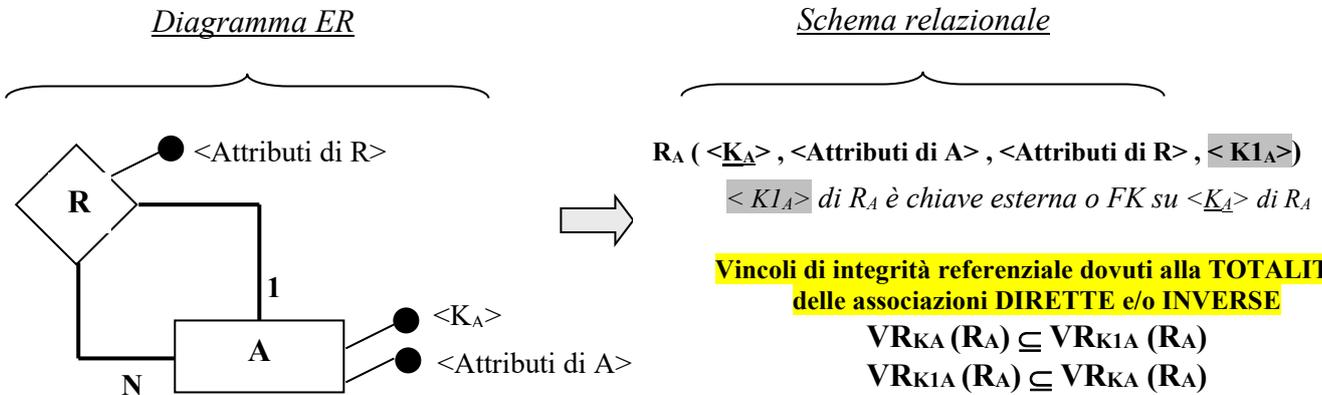
IN QUESTO CASO il mapping relazionale applicato (ossia CON GESTIONE DELLO STORICO) **rende possibile** che un fornitore (chiave **F03**) fornisca uno stesso prodotto (chiave **A01**) in una data differente ed in quantità differenti dalle consegne già effettuate senza incorrere in una **violazione del vincolo di chiave primaria** in quanto le sue forniture saranno contraddistinte da **identificativi univoci** (in questo caso chiavi 0001 e 0004)

**Mapping relazionale delle associazioni su di una stessa entità (associazioni binarie RICORSIVE).**

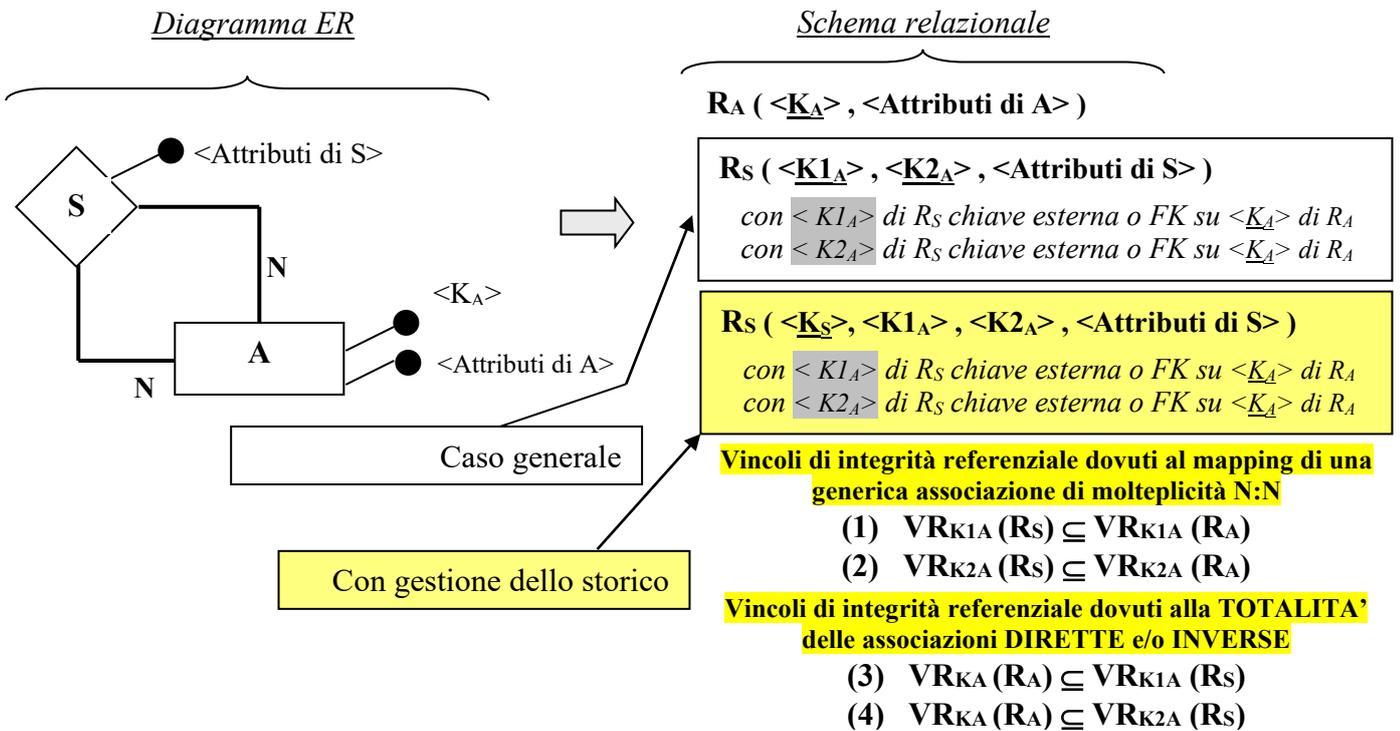
Un caso particolare di associazioni 1:N (o N:1), 1:1 e N:N (con e senza “storico”) è quello in cui l’entità di partenza è uguale a quella di arrivo (associazioni binarie RICORSIVE)

In tutti i casi vanno utilizzati, con apposite SEMPLIFICAZIONI, le regole di mapping illustrate nel caso delle associazioni binarie di tipo 1:N (o N:1), 1:1 e N:N (con e senza “storico”).

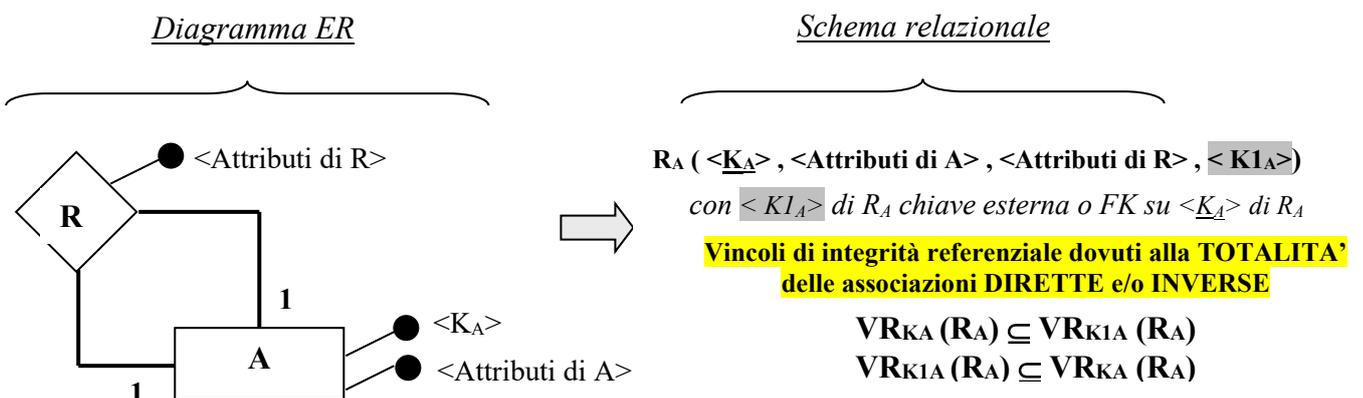
**MAPPING Associazione binaria 1:N (o N:1) con associazione diretta ed inversa TOTALE**



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



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



## Vincoli di integrità nel modello logico relazionale

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 affinché le informazioni contenute nella base dati restino corrette e significative (ossia consistenti) per qualsiasi utente/applicazione le utilizzi.

*Esempio: consideriamo la relazione **Dipendente** che contiene informazioni anagrafiche relative ad un lavoratore di una certa azienda espressa in forma tabellare:*

<b>Matricola</b>	<b>Cognome</b>	<b>Nome</b>	<b>Stipendio</b>	<b>DataNascita</b>	<b>DataAssunzione</b>
2345	Rossi	Gianni	1200.00	01/01/1987	04/09/1986
4667	Neri	Alfonso	-1250.00	08/07/1971	05/09/2006
4667	Bianchi	Adele	1300.00	12/08/1988	06/09/2006

**Oss. 1:** L'attributo "Stipendio" del dipendente "Neri Alfonso" ha un valore pari a **-1250.00** che pur essendo corretta dal punto di vista del tipo del dominio intero (-1250.00 è un numero intero negativo), rappresenta un dato inverosimile per uno stipendio reale;

**Oss. 2:** Nella relazione "Dipendente" 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 lavoratori abbiano lo stesso numero di matricola;

**Oss. 3:** Nella relazione "Dipendente" vi è una ennupla che ha nel campo DataNascita e DataAssunzione 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 un dato lavoratore abbia data di nascita successiva alla data di assunzione.

**Conseguenze:** in questo caso dovremmo poter specificare:

- un vincolo di integrità che ci assicuri che lo stipendio sia un numero maggiore di zero;
- un vincolo di integrità che affermi che non possano esserci due dipendenti con lo stesso numero di matricola;
- un vincolo di integrità che affermi che per ciascun dipendente occorre controllare che la data di assunzione 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 per ogni istanza di una certa relazione possa essere 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 le quali TUTTI I VINCOLI DI INTEGRITA' 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** (in breve **V.d.I.**) del modello relazionale possono essere classificati in:

- **vincoli di integrità INTRA-RELAZIONALI** 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 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 **DOMINIO DI PIU' 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.

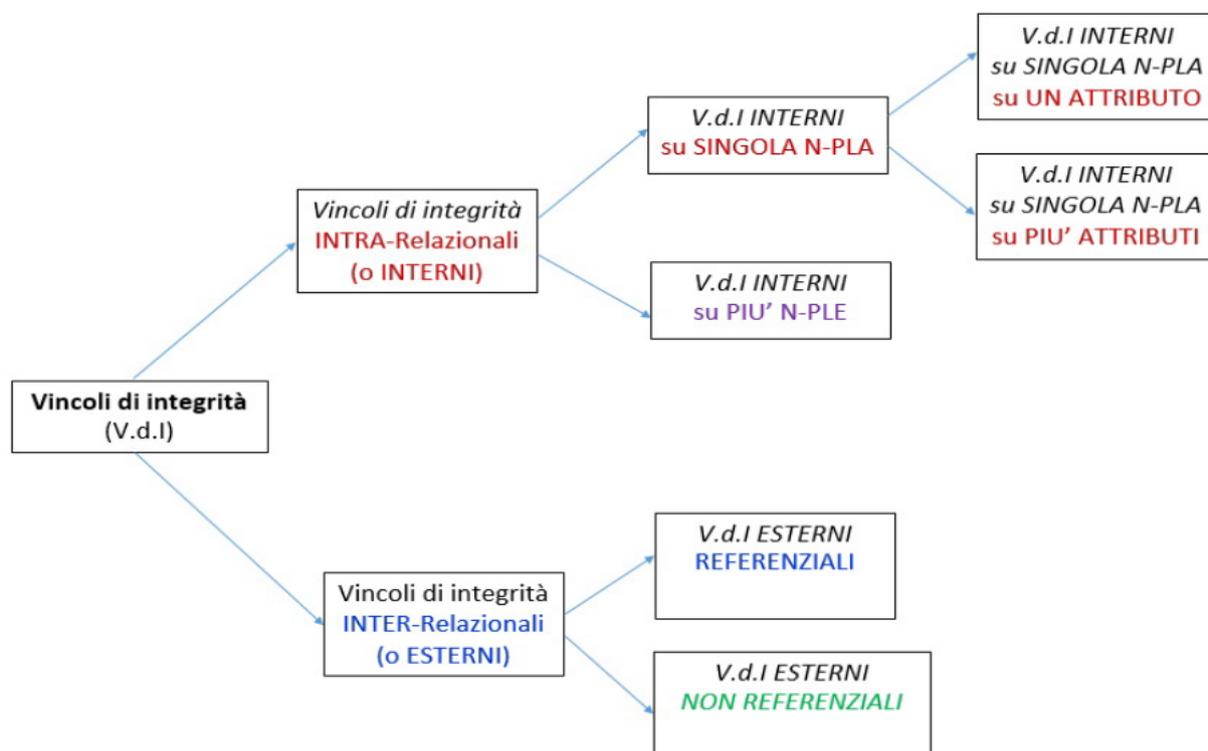
- vincoli **su PIU' ENNUPLE**: sono quei vincoli di integrità che coinvolgono i valori di più ennuple. Rientrano in questo tipo i *vincoli (impliciti) di chiave primaria* (ossia le ennuple di una stessa relazione devono essere tutte diverse tra loro).

- **vincoli di integrità INTER-RELAZIONALI** 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 effettuare il mapping relazionale di un'associazione di molteplicità N:N
- vincoli **NON REFERENZIALI**: Possono rientrare alcuni vincoli espliciti trovati in fase di progettazione concettuale.

### Schematizzazione teorica



**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 **Stipendio > 0** è un vincolo di integrità **intrarelazionale** o **interno**, su **singola** **ennupla** che esprime una condizione sul **dominio di un unico attributo**;

**OSS 2:** il vincolo che afferma che un dipendente **non può avere lo stesso numero di matricola di un altro dipendente** 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 dipendente **DataAssunzione > DataNascita** è un vincolo di integrità **intrarelazionale** o **interno**, su **singola ennupla** che esprime una condizione sul **dominio di più attributi**

### Rappresentazione dei Vincoli di integrità nel modello relazionale

Per rappresentare i **vincoli di integrità di chiave primaria** sottolineiamo i relativi attributi (così come abbiamo fatto nel diagramma ER).

Per rappresentare **tutti gli altri vincoli** utilizzeremo un nostro pseudolinguaggio (così come abbiamo fatto nel diagramma ER per i vincoli di integrità espliciti).

#### a) vincoli di integrità **INTRA-RELAZIONALI** o **INTERNI**

<b>V</b> <NumProgr> (<NomeRelazione>) : (<Espressione>)
---

dove:

- <NumProgr> è il numero progressivo del vincolo relativo alla relazione <NomeRelazione>.

**N.B. Il progressivo dovrà essere lo stesso del vincolo esplicito indicato nel diagramma ER**

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

Esempi:

V1(Dipendente): (Stipendio > 0)

**N.B. L'uso della DOT NOTATION non è necessario**

V2(Dipendente): (DataAssunzione > DataNascita)

#### b) vincoli di integrità **INTER-RELAZIONALI** O **ESTERNI**

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

**VR**<Attributo1> (<Relazione1>)  $\subseteq$  **VR**<Attributo2> (<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>.

<b>N.B. Fanno parte di questi i vincoli di integrità impliciti dovuti alla TOTALITA' delle associazioni dirette e/o inverse</b>
---

Ricordiamo che per semplicità e convenienza <Attributo1> ed <Attributo2> **dovranno avere lo stesso identico nome oppure due nomi sostanzialmente simili** (ad esempio a meno di un progressivo).

Esempio:

**VR** CodAzienda (Dipendente)  $\subseteq$  **VR** CodAzienda (Azienda)

Significa che tutti i valori dell'attributo "**CodAzienda**" presenti nelle ennuple della relazione "**Dipendente**" devono anche essere presenti tra i valori dell'attributo "**CodAzienda**" delle ennuple della relazione "**Azienda**".

In altre parole nella relazione “**Dipendente**” non possono essere presenti codici di aziende non presenti nella relazione “**Azienda**”.

**VR**  $\text{CodRegione1 (Comune)} \subseteq \text{VR CodRegione (Regione)}$

Significa che tutti i valori dell’attributo “**CodRegione1**” presenti nelle ennuple della relazione “**Comune**” devono anche essere presenti tra i valori dell’attributo “**CodRegione**” delle ennuple della relazione “**Regione**”.

In altre parole nella relazione “**Comune**” non possono essere presenti codici di regioni non presenti nella relazione “**Regione**”.

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. Il progressivo dovrà essere lo 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 due relazioni:*

*Azienda (CodAzienda, Denominazione, DataFondazione, Indirizzo, Città, Cap)*

*Dipendente (Matricola, Cognome, Nome, DataNascita, DataAssunzione, Livello, Stipendio, CodAzienda)*

Per esprimere il vincolo di integrità che impone che la data di nascita di ogni dipendente sia successiva alla data di fondazione dell’azienda per cui lavora, scriveremo:

**V3 (Dipendente, Azienda) : (Dipendente.DataNascita > Azienda.DataFondazione)**

**N.B.** Come per gli attributi di entità e/o associazioni in un diagramma ER, per riferirci all’attributo di una certa relazione utilizzeremo la seguente *dot-notation*

<NomeRelazione>.<NomeAttributo>

### **Mapping relazionale delle ISA ossia delle associazioni per generalizzazione**

Abbiamo già visto in precedenza le tre strategie possibili per effettuare la sostituzione di un’ISA all’interno di un diagramma ER (**vedi capitolo 14 Il diagramma ER**).

Pertanto il mapping relazionale in tal caso seguirà tutte le regole di derivazione sin qua illustrate.

Unico caso particolare è quello della **sostituzione della generalizzazione con tante associazioni binarie di molteplicità 1:1 quante sono le entità di specializzazione** individuate.

In pratica in questa tipologia di traduzione si trasforma la generalizzazione in tante associazioni di molteplicità 1:1 quante sono le entità figlie che verranno mappate utilizzando questa accortezza: tutte le relazioni derivate dalle entità figlie saranno **identificate esternamente** ossia **la chiave della relazione derivata dall’entità padre** verrà utilizzata nelle n relazioni figlie sia come chiave primaria che come chiave esterna (quindi non si aggiunge alcun attributo a nessuna delle due)

**N.B. VEDI ESEMPI DI MAPPING DI UNA ISA COMPLETO PRESENTI SUL SITO (sezione LABORATORIO file PDF 15-E1 oppure 15-E2)**

<https://www.riochierego.it/mobile/docs/quinta/lab/Esempio-Completo-3-Mapping-ISA.pdf>

<https://www.riochierego.it/mobile/docs/quinta/lab/Esempio-Completo-4-Mapping-ISA.pdf>

### **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 diagramma ER ci permette di calcolare le relazioni che ne derivano:

$$\text{Numero RELAZIONI} = \text{Numero Entità} + \text{Numero Associazioni N:N}$$

N.B. Rispetto alla formula (che vale anche in caso di associazioni RICORSIVE) dobbiamo specificare che non compaiono le associazioni di tipo 1:N in quanto vengono inglobate nelle entità che sono legate lato N;

In caso di associazioni di tipo 1:1 mappate utilizzando la tecnica della fusione in una sola relazione avremo;

$$\text{Numero RELAZIONI} = \text{Numero Entità} + \text{Numero Associazioni N:N} - \text{Numero Associazioni 1:1}$$

N.B. ricordiamo che le gerarchie ISA vengono "risolte" utilizzando una delle strategie viste in precedenza:

### **N.B. VEDI ESEMPI DI MAPPING COMPLETI SUL SITO**

(sezione LABORATORIO file PDF 3-D)

<http://www.riochierego.it/mobile/docs/quinta/lab/Schema-Di-Esempio-Svolto.pdf>

(sezione TRAINING 2° PROVA file PDF 4)

<https://www.riochierego.it/mobile/docs/quinta/lab/BIBLIOTECA-Esercizio-Svolto.pdf>

## 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:

<u>CodCli</u>	<u>Indirizzo</u>	<u>Città</u>	<u>Cap</u>	<u>CodAcc</u>	<u>Descrizione</u>	<u>Prezzo</u>	<u>Quantità</u>
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:

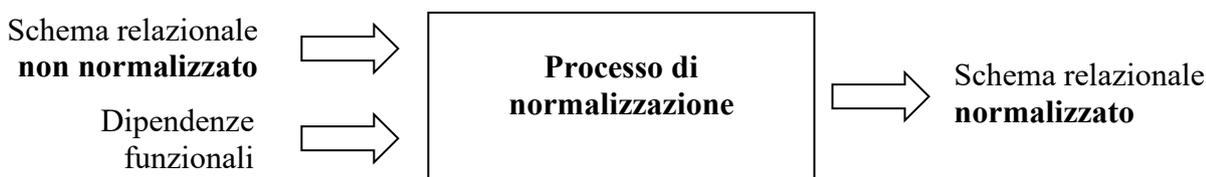
- gli **accessori** presenti in magazzino;
- i dati anagrafici dei **clienti**;
- gli **ordini** dei clienti relativi a determinati accessori.

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.



### 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 dello stesso tipo ovvero appartengono allo stesso dominio;
- i **valori di una ennupla** (di una riga) sono diversi da quelli delle altre ennuple ovvero non possono esistere due ennuple uguali;
- l'**ordine delle ennuple** è irrilevante;
- gli **attributi** sono di tipo **elementare** ossia:
  - non possono *essere composti* da gruppi di attributi ripetuti (no attributi composti o aggregati);
  - non possono *prevedere* una lista variabile di valori (no attributi multipli).

### Dipendenze funzionali

**DEF:** Data una relazione **R** ed un insieme  $X = \{ X_1, X_2, \dots, X_N \}$  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:

- **R** è in **1FN**;
- tutti i suoi attributi dipendono funzionalmente dall'intera chiave primaria e non solo da una parte di essa (ossia non *dipendono parzialmente* dalla chiave primaria).

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.

Le relazioni in 2FN possono ancora essere esposte ad anomalie in quanto possono presentare delle ridondanze

Esempio: nella seguente relazione:

Codice	Titolo	Voto	Matricola
INF1	Informatica	7	0988
SIS1	Sistemi	7	0988
INF1	Informatica	8	0325
ITA1	Italiano	8	0546
SIS1	Sistemi	6	0325

l'attributo **Titolo** dipende solo da **Codice** e non da **Matricola** mentre **Voto** dipende da entrambi.

**La normalizzazione 2NF** darà origine alle seguenti due tabelle:

Codice	Voto	Matricola	Codice	Titolo
INF1	7	0988	INF1	Informatica
SIS1	7	0988	SIS1	Sistemi
INF1	8	0325	INF1	Informatica
ITA1	8	0546	ITA1	Italiano
SIS1	6	0325	SIS1	Sistemi

### 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**;

- tutti gli attributi che non formano una chiave non dipendono in modo transitivo dalla chiave di R in altre parole essi dipendono esclusivamente dalla chiave stessa e non da altri attributi non chiave)(eliminazione delle dipendenze transitive). dipendente dalla chiave primaria

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*.

Esempio: nella seguente relazione:

<u>Matricola</u>	Nome	Comune	CODICE ISTAT
0988	Gianni	Palermo	G273
0989	Franco	Palermo	G273
0325	Isabella	Trapani	H456
0546	Nicoletta	Milano	I123
0328	Maurizio	Palermo	G273

l'attributo **Comune** dipende dall'attributo **CodiceIstat** che non appartiene alla chiave primaria (**Matricola**).

La normalizzazione 3NF darà origine alle seguenti due tabelle:

<u>Matricola</u>	Nome	CODICE ISTAT	Comune	<u>CODICE ISTAT</u>
0988	Gianni	G273	Palermo	G273
0989	Franco	G273	Palermo	G273
0325	Isabella	H456	Trapani	H456
0546	Nicoletta	I123	Milano	I123
0328	Maurizio	G273	Palermo	G273

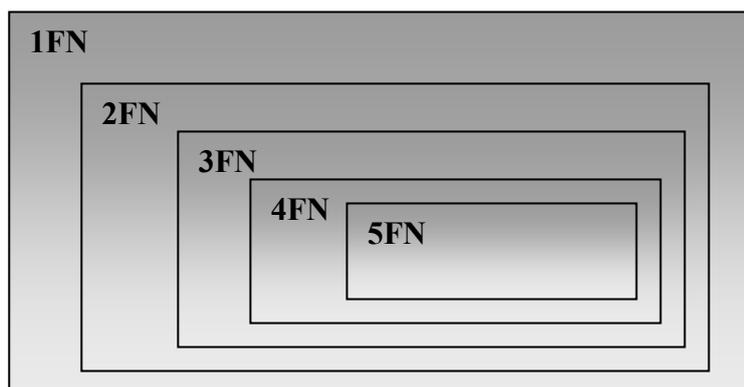
Questa definizione di 3FN può essere data in forma equivalente con la definizione di forma normale di Boyce-Codd o BCNF..

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 annidamento delle forme normali.



## GLI OPERATORI 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 dell'algebra relazionale vista la maggiore familiarità con gli operatori algebrici rispetto al calcolo dei predicati.**

### Algebra relazionale

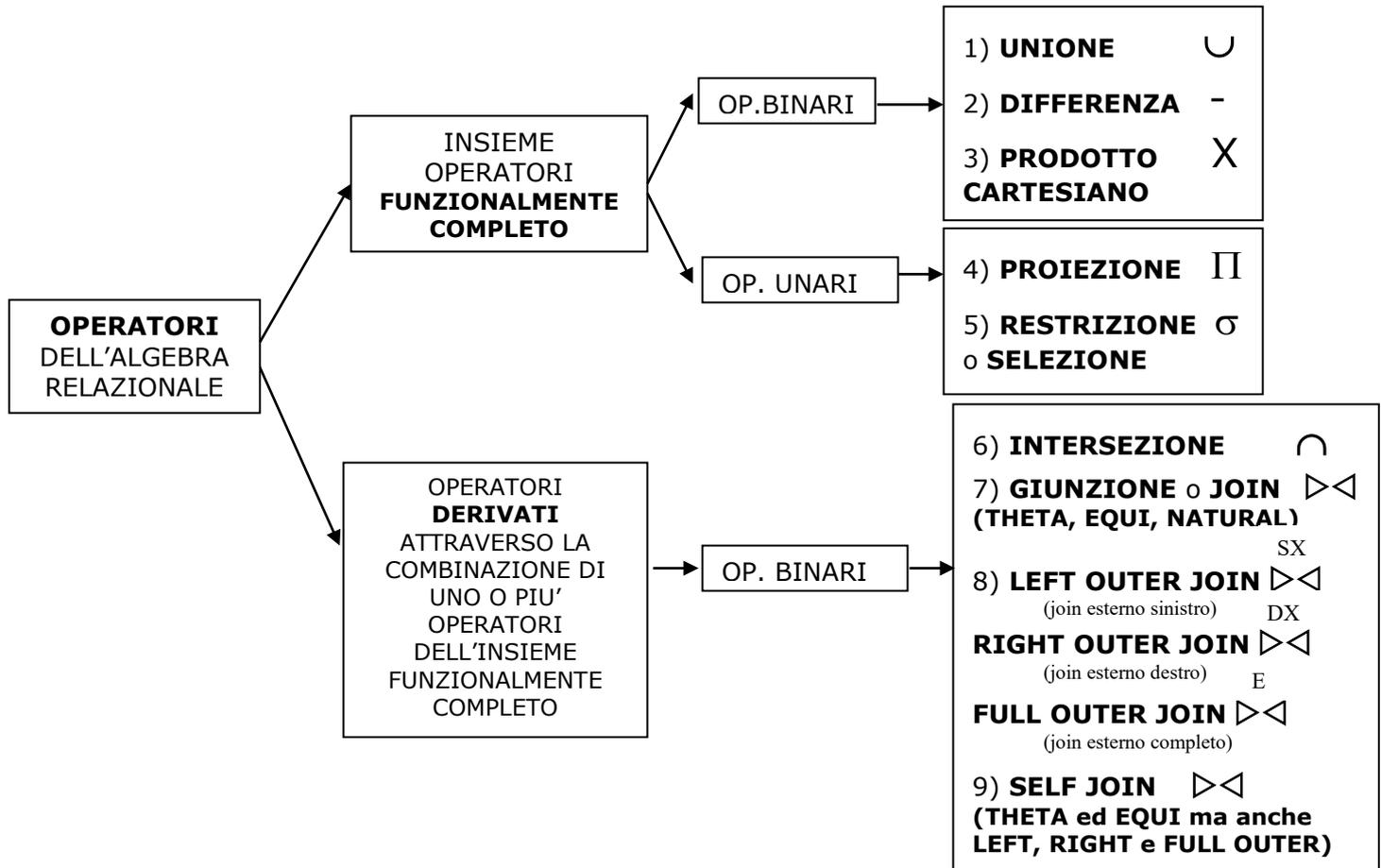
Occorre scegliere innanzitutto un insieme **funzionalmente completo** di operatori (ossia un insieme di operatori che ha la caratteristica di essere sufficiente per rappresentare tutte le funzioni) da utilizzare nelle *interrogazioni o query*.

**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 introdurre 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**

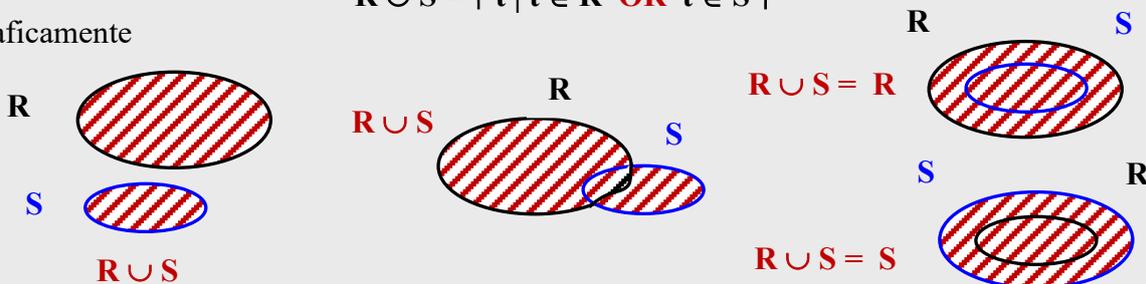
**DEF:** Due relazioni **R** ed **S** vengono chiamate **compatibili** se:  
 - hanno lo stesso *numero* di attributi;  
 - ogni attributo nella stessa posizione all'interno delle due relazioni è dello *stesso tipo*.;

**1) UNIONE di due relazioni (operatore  $\cup$ )**

**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:

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

Graficamente



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

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

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

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

**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  $\cup$  si ha che:

$R \cup S$	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
	C004	Verdi Giuseppe	Via Pia, 11	Lecce

Grado (Cliente-2004  $\cup$  Cliente-2005) = Grado (Cliente-2004) = Grado (Cliente-2005) = 4

Card (Cliente-2004  $\cup$  Cliente-2005) = Card (Cliente-2004) + Card (Cliente-2005) – numero di ennuple ripetute = (3 + 2) – 1 = 4

**2) DIFFERENZA di due relazioni (operatore - )**

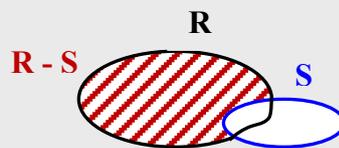
**DEF:** Date due relazioni **compatibili** **R** ed **S** la **differenza di R con S** è la relazione ottenuta dalla differenza insiemistica delle due relazioni ossia:

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

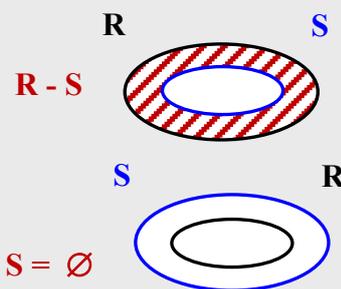
Graficamente



$R - S = R$



$R - S = \emptyset$



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  $\neq$  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**

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

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

<b>S</b>	<b>CodCliente</b>	<b>Nominativo</b>	<b>Indirizzo</b>	<b>Provincia</b>
	<b>C002</b>	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:

<b>R - S</b>	<b>CodCliente</b>	<b>Nominativo</b>	<b>Indirizzo</b>	<b>Provincia</b>
	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 **r = (a1, a2, ..., ag1)** ed una qualsiasi ennupla della seconda relazione **s = (b1, b2, ..., bg2)** e definiamo l'operazione **conc** come

$$r \text{ conc } s = (a_1, a_2, \dots, a_{g_1}, b_1, b_2, \dots, b_{g_2})$$

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

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

**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:

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

$$\text{Card } (R \times S) = \text{Card } (R) * \text{Card } (S)$$

**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.

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

**R = Alunno**

**S = LibroDiTesto**

R	Matricola	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	CodTesto	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	Matricola	Nominativo	Data	CodTesto	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 LibroDiTesto) = Grado (Alunno) + Grado (LibroDiTesto) = (3 + 3) = 6

Card (Alunno X LibroDiTesto) = Card (Alunno) \* Card (LibroDiTesto) = (3 \* 2) = 6

#### 4) PROIEZIONE di una relazione (operatore $\Pi$ )

**DEF:** Data una relazione **R** ed un sottoinsieme  $A = \{A_1, A_2, \dots, A_k\}$  non vuoto dei suoi attributi, si definisce proiezione di R su A la relazione di grado **K** che si ottiene da **R** considerando solo le colonne relative agli attributi contenuti in **A** ed eliminando le eventuali ennuple duplicate. Pertanto scriveremo :

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

oppure in maniera più compatta

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

**N.B.** L'effetto di tale operazione è quello di selezionare un certo numero di colonne della relazione cui si applica (**TAGLIO VERTICALE**).

E' possibile, in teoria, scegliere di proiettare anche tutti gli attributi di **R**.

In questo caso la proiezione di **R** su tutti i suoi attributi coinciderebbe ovviamente con la stessa **R** (**PROIEZIONE BANALE**)

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

**Grado** ( $\Pi_A(R)$ ) = **k** (k è il numero di attributi scelti presenti nell'insieme A)

**Card** ( $\Pi_A(R)$ ) non è prevedibile a priori ma sicuramente minore o uguale a **Card (R)**

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

**R = Cliente**

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**

e supponiamo si vogliano effettuare le seguenti query o interrogazioni:

**Q1: Prelevare dalla base dati il codice e la provincia di tutti i clienti**

Per la risoluzione di tale query occorre effettuare una **proiezione** rispetto ai soli **DUE** attributi **CodCliente** e **Provincia** della relazione **Cliente** che non prevederà eliminazione di ennuple duplicate in quanto evenienza non possibile se perché tra gli attributi previsti è presente anche la chiave primaria della relazione stessa.

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**

$\Pi_A(R)$	<u>CodCliente</u>	Provincia
	C001	Napoli
	C002	Milano
	C003	Napoli

**R = Cliente**  
**A = {CodCliente, Provincia}**

**Grado ( $\Pi_A(\text{Cliente})$ ) = 2** (ossia pari al numero di attributi presente nell'insieme A)

**Card ( $\Pi_A(\text{Cliente})$ ) = 3** (valore comunque  $\leq \text{Card}(\text{Cliente})$ )

**Q2: Prelevare dalla base dati la provincia di tutti i clienti**

Per la risoluzione di tale query occorre effettuare una **proiezione** rispetto al solo attributo **Provincia** della relazione **Cliente** effettuando eventuali eliminazioni di ennuple duplicate.

Operando analogamente a quanto visto in precedenza la relazione ottenuta sarà (con eliminazione di duplicati):

$\Pi_A(R)$	Provincia
	Napoli
	Milano
	<del>Napoli</del>

**R = Cliente**  
**A = {Provincia}**

**Grado ( $\Pi_A(\text{Cliente})$ ) = 1** (ossia pari al numero di attributi scelto)

**Card ( $\Pi_A(\text{Cliente})$ ) = 2** (valore comunque  $\leq \text{Card}(\text{Cliente})$ )

**5) RESTRIZIONE O SELEZIONE di una relazione (operatore  $\sigma$ )**

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

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-ple (ossia avrebbe cardinalità pari a zero).

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

**Grado** (  $\sigma_P(R)$  ) = **Grado** ( **R** )

**Card** (  $\sigma_P(R)$  ) non è prevedibile a priori ma sicuramente minore o uguale a **Card**(**R**)

**Esempio:** Sia data la **STESSA** relazione **Cliente** definita in precedenza utilizzando la rappresentazione tabellare:

**R= Cliente**

<b>R</b>	<b>CodCliente</b>	<b>Nominativo</b>	<b>Indirizzo</b>	<b>Provincia</b>
	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

e supponiamo si vogliono effettuare le seguenti query o interrogazioni:

**Q3: Prelevare dalla base dati tutti i clienti di Napoli**

In questo caso abbiamo

**R= Cliente** e **P** = {Provincia = "Napoli"}

e quindi dalla tabella **Cliente** si "sfileranno" solo le ennuple che verificano il predicato **P**

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

**R** = Cliente  
**P** = {Provincia = "Napoli"}

ottenendo la seguente tabella:

$\sigma_P(R)$	<b>CodCliente</b>	<b>Nominativo</b>	<b>Indirizzo</b>	<b>Provincia</b>
	C001	Neri Mario	Via Po, 5	Napoli
	C003	Rossi Antonio	Via Riga, 9	Napoli

**Grado** (  $\sigma_P(\text{Cliente})$  ) = 4 (ossia pari al numero di attributi della relazione **Cliente**)

**Card** (  $\sigma_P(\text{Cliente})$  ) = 2 (comunque sicuramente  $\leq$  **Card** ( **Cliente** ) )

**Q4: Prelevare dalla base i clienti che si chiamano Mario Neri e sono di Napoli**

In questo caso avendo

$$R = \text{Cliente} \quad e \quad P = \{(Nominativo = "Neri Mario") \text{ AND } (Provincia = "Napoli")\}$$

verrà prelevata solo una ennupla che verifica il predicato P

$\sigma_P(R)$	<u>CodCliente</u>	Nominativo	Indirizzo	Provincia
	C001	Neri Mario	Via Po, 5	Napoli

**Grado** ( $\sigma_P(\text{Cliente})$ ) = 4 (ossia pari al numero di attributi della relazione **Cliente**)

**Card** ( $\sigma_P(\text{Cliente})$ ) = 1 (comunque sicuramente  $\leq \text{Card}(\text{Cliente})$ )

**!!! ATTENZIONE !!!**

Le query proposte fino a questo momento sono state risolte utilizzando di volta in volta **UN SOLO OPERATORE** dell'algebra relazionale. Questa è un'evenienza sicuramente possibile.

In **GENERALE** però una query può richiedere per la sua risoluzione l'utilizzo di **PIU' DI UN'OPERATORE DELL'ALGEBRA RELAZIONALE**.

In questo caso per ottenere la **TABELLA "risultato"** dovremo "svolgere" l'ESPRESSIONE relazionale progettata applicando uno dopo l'altro gli operatori relazionali indicati, in modo del tutto analogo al concetto matematico di "funzione composta" (ossia procedendo dall'operatore più "interno" verso l'operatore più "esterno" rispettando eventuali parentesi tonde)

**Esempio:** Sia data la **STESSA** relazione **Cliente** definita in precedenza utilizzando la rappresentazione tabellare:

**R= Cliente**

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**

e supponiamo si vogliono effettuare le seguenti query o interrogazioni:

**Q5: Prelevare i nominativi dei clienti di Napoli**

Innanzitutto devo effettuare un'attenta lettura della traccia proposta rispondendo alle seguenti domande in grado di aiutarmi nel formulare la soluzione:

**1) Quali sono gli attributi "citati" o "sottintesi" nel testo della query ed a quali relazioni appartengono?**

Sono gli attributi "Nominativo" e "Provincia" entrambi appartenenti alla relazione "Cliente".

**2) Quali sono gli attributi i cui valori si vogliono ottenere come "output" (visualizzazione) della query ed a quali relazioni appartengono?**

E' l'attributo "Nominativo" della relazione "Cliente"..... quindi **A = {Nominativo}**

**3) Quali sono gli attributi sui cui valori si richiede la verifica di un determinato predicato ed a quali relazioni appartengono? Qual è questo predicato?**

E' l'attributo "Provincia" della relazione "Cliente" e quindi **P = {Provincia = "Napoli"}**

N.B. In questo caso abbiamo evitato di usare la **DOT NOTATION** per indicare a quale relazione appartengono gli attributi coinvolti nella query, in quanto non vi è alcuna possibilità di confusione considerato il coinvolgimento di **UN'UNICA RELAZIONE**.

**In caso di due o più relazioni la DOT NOTATION andrà obbligatoriamente UTILIZZATA.**

Quindi riepilogando:

- dalla risposta alla **domanda 2** deriva l'utilizzo dell'operatore relazionale **PROIEZIONE** della relazione **Cliente** sull'insieme di attributi **A = {Nominativo}**;
- dalla risposta alla **domanda 3** deriva l'utilizzo dell'operatore relazionale **SELEZIONE** o **RESTRIZIONE** della relazione **Cliente** sul predicato **P = {Provincia = "Napoli"}**;

Quindi per risolvere la nostra query occorrerà applicare due operatori relazionali (**PROIEZIONE** e **SELEZIONE** o **RESTRIZIONE**) .....

.... **si ma in quale ordine?**

.....**e su quale relazione (tabella)?**

**Ricordiamo che l'esecuzione di UN QUALSIASI OPERATORE relazionale darà origine ad una NUOVA RELAZIONE (tabella) in generale DIFFERENTE dalla RELAZIONE (tabella) di PARTENZA**

**A) prima ipotesi:**

$$Q5 = \sigma_P(\Pi_A(\text{Cliente}))$$

**A = {Nominativo}**  
**P = {Provincia = "Napoli"}**

**1. Effettuiamo prima l'operazione relazionale  $\Pi_A$  sulla relazione **Cliente****

Cliente	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

**A = {Nominativo}**



$\Pi_A(\text{Cliente})$	Nominativo
	Neri Mario
	Bianchi Gianni
	Rossi Antonio

**2. Effettuiamo ora l'operazione relazionale  $\sigma_P$  sulla relazione  $\Pi_A(\text{Cliente})$**

???

$\Pi_A(\text{Cliente})$	Nominativo
	Neri Mario
	Bianchi Gianni
	Rossi Antonio

Tale operazione appare evidentemente **IMPOSSIBILE** poiché è sparito l'attributo "Provincia" sul quale applicare il predicato **P**

**P = {Provincia = "Napoli"}**

**B) seconda ipotesi:**

$$Q5 = \Pi_A (\underbrace{\sigma_P}_{1} (\underbrace{\text{Cliente}}_{2}))$$

$A = \{\text{Nominativo}\}$   
 $P = \{\text{Provincia} = \text{"Napoli"}\}$

1. Effettuiamo prima l'operazione relazionale  $\sigma_P$  sulla relazione **Cliente**

Cliente	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

$P = \{\text{Provincia} = \text{"Napoli"}\}$



$\sigma_P$ (Cliente)	CodCliente	Nominativo	Indirizzo	Provincia
	C001	Neri Mario	Via Po, 5	Napoli
	C003	Rossi Antonio	Via Riga, 9	Napoli

2. Effettuiamo ora l'operazione relazionale  $\Pi_A$  sulla relazione  $\sigma_P$  (Cliente)



$\Pi_A$ ( $\sigma_P$ (Cliente))	Nominativo
	Neri Mario
	Rossi Antonio

$A = \{\text{Nominativo}\}$

**!!! NOTA BENE !!!**  
 Occorre sempre effettuare l'operazione relazionale di **PROIEZIONE** come ultima in quanto altera la struttura della **RELAZIONE** (tabella) sulla quale, viene applicata.  
 Quindi, in questo caso, **SEMPRE DOPO** l'operazione di **SELEZIONE** o **RESTRIZIONE**

### OPERATORI RELAZIONALI DERIVATI

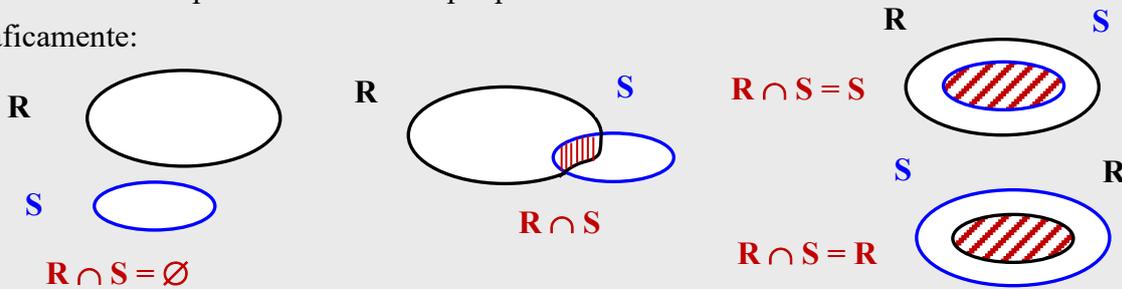
#### 6) INTERSEZIONE di due relazioni (operatore $\cap$ )

**DEF:** Date due relazioni **compatibili** **R** ed **S**, l'intersezione di R ed S restituisce la relazione ottenuta dall'intersezione insiemistica tra le relazioni ossia

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

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

Graficamente:

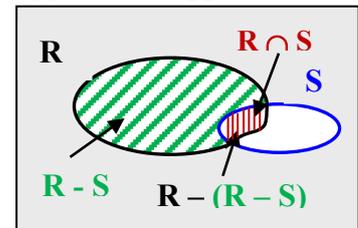


E' possibile verificare grazie con l'utilizzo dei diagrammi di Eulero-Venn per gli insiemi che vale la seguente uguaglianza (e ciò dimostra che tale operatore è derivabile applicando opportunamente alla relazione uno o più operatori fondamentali)

$$R \cap S = R - (R - S)$$

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

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



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

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

**R= Clienti-2004**

**S= Clienti-2005**

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

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

S	CodCliente	Nominativo	Indirizzo	Provincia
	<b>C002</b>	<b>Bianchi Gianni</b>	<b>Via Lima, 7</b>	<b>Milano</b>
	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:

$R \cap S$	CodCliente	Nominativo	Indirizzo	Provincia
	<b>C002</b>	<b>Bianchi Gianni</b>	<b>Via Lima, 7</b>	<b>Milano</b>

$$\text{Grado} ( \text{Clienti-2004} \cap \text{Clienti-2005} ) = \text{Grado} ( \text{Clienti-2004} ) = \text{Grado} ( \text{Clienti-2005} ) = 4$$

$$\text{Card}(\text{Clienti-2004} \cap \text{Clienti-2005}) = 1 \text{ ossia } \leq \min \{ \text{Card}(\text{Clienti-2004}), \text{Card}(\text{Clienti-2005}) \}$$

**N.B. Casi particolari:**

Se  $R \subseteq S$  allora è facile dimostrare che  $\text{Card}(R \cap S) = \text{Card}(R)$

Se  $S \subseteq R$  allora è facile dimostrare che  $\text{Card}(R \cap S) = \text{Card}(S)$

**7) THETA-JOIN, EQUI JOIN e NATURAL JOIN tra due relazioni (operatore  $\triangleright \triangleleft$ )**

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

**DEF:** Date due relazioni **qualsiasi**  $R$  ed  $S$  rispettivamente di **grado**  $g_1$  e  $g_2$  e **cardinalità**  $c_1$  e  $c_2$  e scelto un **predicato**  $P$  qualunque su uno o più attributi delle due relazioni, si definisce **join condizionale** o **THETA-JOIN** tra  $R$  ed  $S$  la relazione di **grado**  $(g_1 + g_2)$  e **cardinalità** non prevedibile a priori ma certamente minore o uguale a  $c_1 * c_2$  le cui ennuple si ottengono con il seguente procedimento:

- 1) prima si effettua il **prodotto cartesiano** di  $R$  ed  $S$  ( $R \times S$ );
- 2) poi si effettua una **restrizione (o selezione)** sulla relazione  $R \times S$  che seleziona le ennuple che verificano il predicato  $P$  prefissato.

**Quindi in sintesi si tratta della seguente operazione relazionale composta  $\sigma_P(R \times S)$  che è possibile sintetizzare con il seguente formalismo:**

$$R \triangleright \triangleleft S \\ P$$

Il predicato  $P$  può contenere un enunciato semplice o composto che utilizzi uno qualsiasi dei **sei operatori di confronto fondamentali** ( $\leq, <, >, \geq, =, \neq$ )

**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  $S$  rispettivamente di **grado**  $g_1$  e  $g_2$  e **cardinalità**  $c_1$  e  $c_2$  e scelto un **attributo**  $A$  di  $R$  ed un **attributo**  $B$  di  $S$ , *aventi lo stesso tipo*, si definisce **equi giunzione** o **EQUI-JOIN** tra  $R$  ed  $S$  la relazione di **grado**  $(g_1 + g_2)$  e **cardinalità** non prevedibile a priori ma certamente minore o uguale a  $c_1 * c_2$  le cui ennuple si ottengono con il seguente procedimento:

- 1) prima si effettua il **prodotto cartesiano** di  $R$  ed  $S$  (ossia  $R \times S$ );
- 2) poi si effettua una **restrizione (o selezione)** sulla relazione  $R \times S$  che seleziona le ennuple aventi **lo stesso valore** degli attributi  $A$  di  $R$  e  $B$  di  $S$  ossia applicando il predicato (enunciato semplice)  $P = \{ R.A = S.B \}$

Si ottiene così una relazione al cui interno saranno presenti ovviamente anche le colonne  $R.A$  e  $S.B$  che risulteranno possedere valori uguali.

**Anche in questo caso si tratta della seguente operazione relazionale composta  $\sigma_P(R \times S)$  che è possibile sintetizzare con il seguente formalismo:**

$$R \triangleright \triangleleft S \\ R.A = S.B$$

**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 vi è un'ovvia ridondanza in quanto le due colonne che avevano già i valori uguali (perché frutto di un equi-join) ora hanno anche il nome uguale. Si ottiene in questo modo una relazione uguale alla precedente, ma con grado  $(g_1 + g_2 - 1)$  che prende il nome di **giunzione naturale** o **NATURAL-JOIN** tra  $R$  ed  $S$

**DEF:** Date due relazioni **qualsiasi R** ed **S** rispettivamente di **grado g1** e **g2** e **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 **NATURAL-JOIN** tra **R** ed **S** la relazione di **grado (g1 + g2 -1)** e **cardinalità non prevedibile a priori ma minore o uguale a c1\*c2** le cui ennuple si ottengono con il seguente procedimento:

- 1) prima si effettua il **prodotto cartesiano** di R ed S (**R x S**);
- 2) poi si effettua una **restrizione (o selezione)** sulla relazione **R x S** che seleziona le ennuple aventi **lo stesso valore** degli attributi R.A e S.A (ossia predicato  $P = \{R.A = S.A\}$ ).  
Si ottiene così una relazione una relazione al cui interno sarebbero presenti le colonne R.A e S.A con gli stessi valori e con lo stesso nome;
- 3) infine si elimina dalla relazione ottenuta, una di queste due colonne perché ridondante.

**In questo caso si tratta della seguente operazione relazionale composta**

$$\Pi_K(\sigma_P(R \times S))$$

con **K** che risulta essere l'insieme di tutti gli attributi della relazione **R x S** eccetto una tra le due colonne uguali.

**N.B.** Per quest'ultimo operatore relazionale non è prevista una notazione ad hoc

Lo scopo della **giunzione naturale** o **NATURAL-JOIN** in sintesi è quello di combinare due relazioni aventi uno o più attributi in comune omonimi 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 (equigiunzione o EQUI-JOIN):** Siano date le seguenti due relazioni **R** ed **S** i cui **schemi**, dopo il mapping relazionale, siano i seguenti:

**R = Cliente** (CodCliente, NomeCliente, Indirizzo, **CodAg**)

*con l'attributo "CodAg" della relazione "Cliente" che risulta chiave esterna (o FK) sull'attributo "CodAgente" della relazione "Agente"*

**S = Agente** (CodAgente, NomeAgente, Telefono)

Utilizzando la rappresentazione tabellare, supponiamo che tali relazioni contengano, in un certo istante di tempo, le seguenti ennuple:

R	<u>CodCliente</u>	NomeCliente	Indirizzo	<b>CodAg</b>
	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	<b>CodAgente</b>	NomeAgente	Telefono
	A001	Verdi Luca	081-123456
	A002	Gialli Matteo	081-654321
	A003	Vito Andrea	081-456345

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

Proviamo adesso a far vedere come si ricava la relazione risultato dell'operazione relazionale:

**Cliente**  $\bowtie$  **Agente**  
CodAg = CodAgente

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

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

<b>R X S</b>	<b>CodCliente</b>	<b>NomeCliente</b>	<b>Indirizzo</b>	<b>CodAg</b>	<b>CodAgente</b>	<b>NomeAgente</b>	<b>Telefono</b>
	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

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

<b><math>\sigma_P(\mathbf{R X S})</math></b>	<b>CodCliente</b>	<b>NomeCliente</b>	<b>Indirizzo</b>	<b>CodAg</b>	<b>CodAgente</b>	<b>NomeAgente</b>	<b>Telefono</b>
	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**  $\text{Cliente} \bowtie_{\text{CodAg} = \text{CodAgente}} \text{Agente}$  proposto è la seguente tabella:

<b><math>\text{Cliente} \bowtie_{\text{CodAg} = \text{CodAgente}} \text{Agente}</math></b>	<b>CodCliente</b>	<b>NomeCliente</b>	<b>Indirizzo</b>	<b>CodAg</b>	<b>CodAgente</b>	<b>NomeAgente</b>	<b>Telefono</b>
	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

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

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

N.B. Se invece proviamo a ricavare la relazione risultato di quest'altra operazione relazionale:

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

che ha come scopo quello di mostrare, in un'unica tabella, per ogni agente, anche le informazioni relative ai clienti gestiti, ci accorgeremmo ovviamente che essa differirà dalla tabella ottenuta in precedenza.

Ricordiamo infatti che, qualunque sia il tipo di join eseguito, esso conterrà sempre anche il **prodotto cartesiano tra due relazioni**, che, come sappiamo, è un operatore relazionale che **non gode della proprietà commutativa**.

**Esempio (giunzione naturale o NATURAL-JOIN):** Siano date le medesime due relazioni **R** ed **S** definite nell'esercizio precedente, ed immaginiamo di modificare esclusivamente il nome di un attributo della relazione Cliente (quello che svolge il ruolo di chiave esterna), in modo che esso coincida con il nome di un attributo della relazione agente (quello che svolge il ruolo di chiave primaria):

**R= Cliente**

**S= Agente**

R	CodCliente	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	CodAgente	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 seguente **NATURAL-JOIN**

**Cliente  $\triangleright\triangleleft$  Agente**

R.CodAgente = S.CodAgente

dopo tutti i passi previsti, può essere vista come la tabella ottenuta effettuando l'EQUI-JOIN come spiegato in precedenza, ma eliminando la colonna R.CodAgente perché ridondante, ossia:

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

**Grado (Cliente  $\triangleright\triangleleft$  Agente) = Grado (Agente) + Grado (Cliente) - 1 = (4 + 3) - 1 = 6**  
 Cliente.CodAgente = Agente.CodAgente

**Card (Cliente  $\triangleright\triangleleft$  Agente) = 3** che risulta essere  $\leq$  **Card (Agente) \* Card (Cliente) = 12**  
 Cliente.CodAgente = Agente.CodAgente

**ALTRI TIPI DI JOIN**

**8.1) Join esterno sinistro (LEFT-OUTER-JOIN ossia Left join**  $\triangleright\triangleleft^{\wedge}$  <sup>SX</sup>  
<sub>DX</sub>

**8.2) Join esterno destro (RIGHT-OUTER-JOIN ossia Right join**  $\triangleright\triangleleft^{\wedge}$  <sup>E</sup>

**8.3) Join esterno completo (FULL-OUTER-JOIN ossia Outer join**  $\triangleright\triangleleft^{\wedge}$ )

**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:** A tutte le n-ple risultanti dall’applicazione dell’operatore relazionale **EQUI-JOIN**, si uniscono (sommano) 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.

$R \triangleright\triangleleft^{\wedge} S = R \triangleright\triangleleft S \cup$  <sup>SX</sup>  
<sub>R.A=S.B R.A=S.B</sub> 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:** A tutte le n-ple risultanti dall’applicazione dell’operatore relazionale **EQUI-JOIN**, si uniscono (sommano) 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.

$R \triangleright\triangleleft^{\wedge} S = R \triangleright\triangleleft S \cup$  <sup>DX</sup>  
<sub>R.A=S.B R.A=S.B</sub> 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**.

$R \triangleright\triangleleft^{\wedge} S = R \triangleright\triangleleft^{\wedge} S \cup R \triangleright\triangleleft^{\wedge} S$  <sup>E SX DX</sup>  
<sub>R.A=S.B R.A=S.B R.A=S.B</sub>

**9) SELF-(EQUI)-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.

**Nota bene: Esempio classico di utilizzo si ha quando occorre eseguire un EQUI-JOIN su una relazione ottenuta dal mapping relazionale di una associazione binaria ricorsiva di molteplicità 1:N (o N:1 o 1:1)**

**Esempio di join esterni:**

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

**R= Cliente**

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= Agente**

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** (detto anche **INNER JOIN**) si ottiene la seguente tabella

**Cliente**  $\bowtie$  **Agente**  
CodAg = CodAgente

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

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

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

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

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

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

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



Ossia dalla tabella:

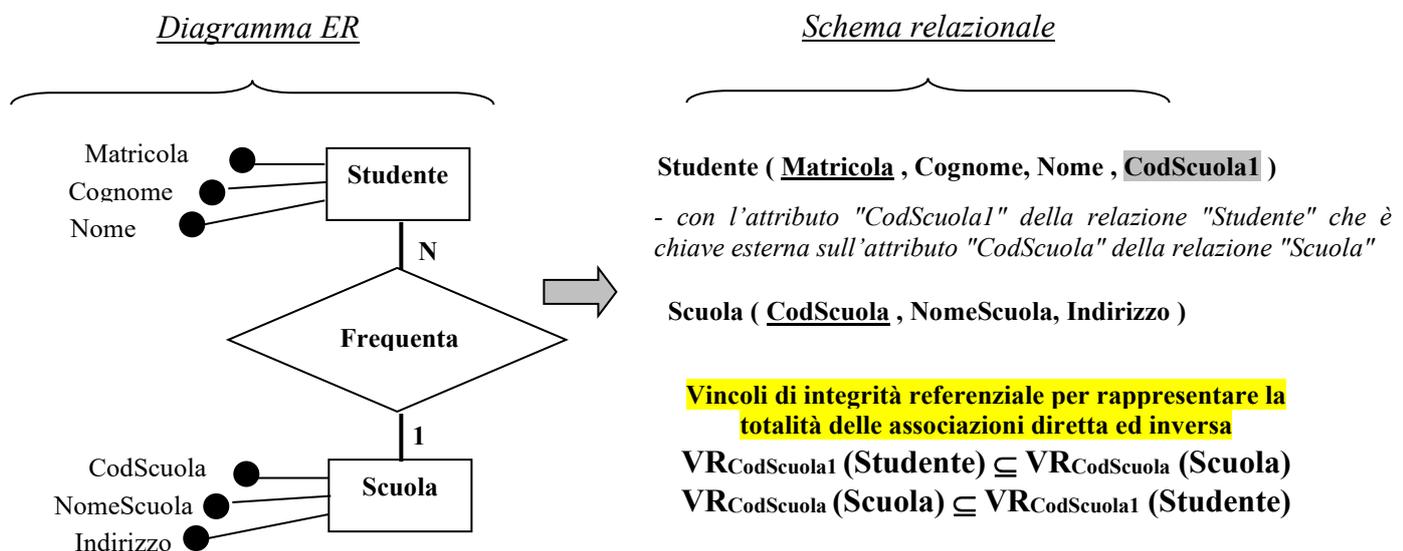
$\sigma_P (R \times R)$	<u>CodDip</u>	Cognome	Nome	Indirizzo	CodDir	<u>CodDip</u>	Cognome	Nome	Indirizzo	CodDir
$\uparrow$ $(R \triangleright \triangleleft R)$ $R.A = R.B$	D001	Baracco	Michele	Via Po, 5	D002	D002	Battistina	Marina	Via Adda, 7	NULL
	D003	Bessone	Fabia	Via Adige, 8	D002	D002	Battistina	Marina	Via Adda, 7	NULL
	D004	Genchi	Mario	Via Tevere, 4	D003	D003	Bessone	Fabia	Via Adige, 8	D002

## QUERY SULLO SCHEMA RELAZIONALE UTILIZZANDO CON L'ALGEBRA RELAZIONALE

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

### A) Interrogazione di RELAZIONI derivanti da associazioni di molteplicità N:1 (oppure 1:N oppure 1:1)

B) Aiutiamoci con un esempio



**QUERY n. 1: Vogliamo conoscere i cognomi ed i nomi degli studenti che frequentano la scuola con codice meccanografico "NATF091011"**

Questa interrogazione richiede informazioni reperibili dalla sola relazione "Studente". Occorre dapprima eseguire una **restrizione** sulla relazione **Studente** volta a selezionare le n-ple per le quali **CodScuola1 = "NATF091011"** per poi operare una **proiezione** (taglio verticale) rispetto agli attributi scelti **Studente.Cognome** e **Studente.Nome**.

$$\Pi_{Cognome, Nome} \left( \overset{2^\circ}{\underbrace{\sigma_{CodScuola1 = "NATF091011"} (Studente)}} \right)$$

Questa interrogazione può essere vista come una interrogazione composta da **due interrogazioni elementari** una più interna chiamata **sottointerrogazione** ed una più esterna:

quella più interna:  $T1 = \sigma_{CodScuola1 = "NATF091011"} (Studente)$

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

Utilizziamo il modello grafico per le relazioni (forma tabellare) supponendo che la tabella **Studente** sia, in un certo istante di tempo, così popolata e calcoliamo il risultato passo per passo:

Tabella **Studente**

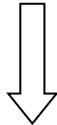
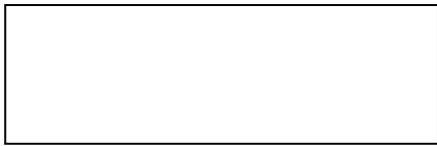
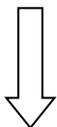
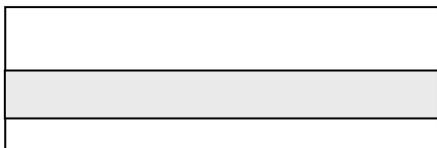
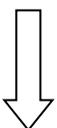


Tabella **Studente**



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



**T2** = Tabella **T1** "proiettata"

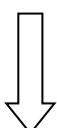
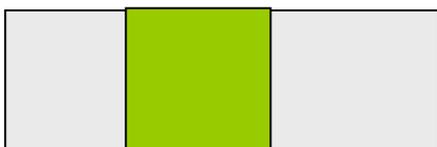


Tabella **risultato finale**



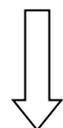
<u>Matricola</u>	Cognome	Nome	CodScuola1
M001	Rossi	Paolo	NATF091011
M002	Bianchi	Aldo	NATF091011
M003	Verdi	Ada	NAXX123456
M004	Neri	Maria	NAXX123456



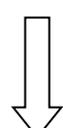
<u>Matricola</u>	Cognome	Nome	CodScuola1
M001	Rossi	Paolo	NATF091011
M002	Bianchi	Aldo	NATF091011
M003	Verdi	Ada	NAXX123456
M004	Neri	Maria	NAXX123456



<u>Matricola</u>	Cognome	Nome	CodScuola1
M001	Rossi	Paolo	NATF091011
M002	Bianchi	Aldo	NATF091011



<u>Matricola</u>	Cognome	Nome	CodScuola1
M001	<b>Rossi</b>	<b>Paolo</b>	NATF091011
M002	<b>Bianchi</b>	<b>Aldo</b>	NATF091011



**Risultato finale**  
**T2**

Cognome	Nome
<b>Rossi</b>	<b>Paolo</b>
<b>Bianchi</b>	<b>Aldo</b>

**QUERY n. 2: Vogliamo conoscere i cognomi ed i nomi degli studenti che frequentano la scuola "ISIS G. Tassinari"**

Questa interrogazione rispetto alla precedente richiede informazioni reperibili sia dalla relazione "Studente" sia dalla relazione "Scuola".

1) Occorre dunque eseguire un **EQUI-JOIN** tra le relazioni **Studente** e **Scuola** uguagliando i valori della chiave esterna **CodScuola1** (sulla relazione **Studente**) con quelli della chiave primaria **CodScuola** (della relazione **Scuola**).

2) Sulla relazione così ottenuta si eseguirà poi una **selezione** con condizione **NomeScuola = "ISIS G. Tassinari"**.

3) Successivamente si opererà una **proiezione** (taglio verticale) rispetto agli attributi scelti **Studente.Cognome, Studente.Nome**.

$$\Pi_{\text{Cognome, Nome}} \left( \underbrace{\left( \underbrace{\left( \underbrace{\text{Studente} \bowtie \text{Scuola}}_{1^\circ} \right)}_{2^\circ} \right)}_{3^\circ} \right)_{\text{CodScuola1} = \text{CodScuola}}$$

Questa interrogazione può essere quindi vista come una interrogazione composta da tre interrogazioni (una derivata e due elementari) da eseguire una dopo l'altra, ossia:

la prima interrogazione:  $T1 = (\text{Studente} \bowtie \text{Scuola})_{\text{CodScuola1} = \text{CodScuola}}$

la seconda interrogazione:  $T2 = \sigma_{\text{NomeScuola} = \text{"ISIS G. Tassinari"}} (T1)$

la terza interrogazione:  $T3 = \Pi_{\text{Cognome, Nome}} (T2)$

Supponiamo che le tabelle **Studente** e **Scuola** siano, in un certo istante di tempo, così popolate:

<b>Studente</b>	<u>Matricola</u>	Cognome	Nome	CodScuola1
	M001	Rossi	Paolo	NATF091011
	M002	Bianchi	Aldo	NATF091011
	M003	Verdi	Ada	NAXX123456
	M004	Neri	Maria	NAXX123456

<b>Scuola</b>	<u>CodScuola</u>	NomeScuola	Indirizzo
	NATF091011	ISIS G. Tassinari	Via Fasano, 78
	NAXX123456	ITIS A. Volta	Via Roma, 144

Allora eseguiamo innanzitutto la prima operazione relazionale (**equi-join**) prevista:

$$T1 = (\text{Studente} \bowtie \text{Scuola})_{\text{CodScuola1} = \text{CodScuola}}$$

Tale operazione come ricordiamo, risulta essere un'operazione derivata che da come risultato una tabella le cui n-ple si ottengono dal seguente procedimento:

a) prima di tutto dobbiamo eseguire il prodotto cartesiano delle relazioni **Studente** e **Scuola**:

<b>Studente X Scuola</b>	<u>Matricola</u>	Cognome	Nome	CodScuola1	<u>CodScuola</u>	NomeScuola	Indirizzo
	M001	Rossi	Paolo	NATF091011	NATF091011	ISIS G. Tassinari	Via Fasano, 78
	M001	Rossi	Paolo	NATF091011	NAXX123456	ITIS A. Volta	Via Roma, 144
	M002	Bianchi	Aldo	NATF091011	NATF091011	ISIS G. Tassinari	Via Fasano, 78
	M002	Bianchi	Aldo	NATF091011	NAXX123456	ITIS A. Volta	Via Roma, 144
	M003	Verdi	Ada	NAXX123456	NATF091011	ISIS G. Tassinari	Via Fasano, 78
	M003	Verdi	Ada	NAXX123456	NAXX123456	ITIS A. Volta	Via Roma, 144
	M004	Neri	Maria	NAXX123456	NATF091011	ISIS G. Tassinari	Via Fasano, 78
	M004	Neri	Maria	NAXX123456	NAXX123456	ITIS A. Volta	Via Roma, 144

b) poi sulla relazione **Studente X Scuola** così ottenuta, dobbiamo eseguire una **selezione** ponendo la condizione di equi-join **Studente.CodScuola1 = Scuola.CodScuola**.

Avremo come risultato la seguente relazione **T1**

<b>T1</b>	<u>Matricola</u>	Cognome	Nome	CodScuola1	<u>CodScuola</u>	NomeScuola	Indirizzo
	M001	Rossi	Paolo	NATF091011	NATF091011	ISIS G. Tassinari	Via Fasano, 78
	M002	Bianchi	Aldo	NATF091011	NATF091011	ISIS G. Tassinari	Via Fasano, 78
	M003	Verdi	Ada	NAXX123456	NAXX123456	ITIS A. Volta	Via Roma, 144
	M004	Neri	Maria	NAXX123456	NAXX123456	ITIS A. Volta	Via Roma, 144

Eseguiamo poi la seconda operazione relazionale prevista ossia

$$\mathbf{T2} = \sigma_{\text{NomeScuola} = \text{"ISIS G. Tassinari"}}(\mathbf{T1})$$

ed otterremo la seguente relazione:

<b>T2</b>	<u>Matricola</u>	Cognome	Nome	CodScuola1	<u>CodScuola</u>	NomeScuola	Indirizzo
	M001	Rossi	Paolo	NATF091011	NATF091011	ISIS G. Tassinari	Via Fasano, 78
	M002	Bianchi	Aldo	NATF091011	NATF091011	ISIS G. Tassinari	Via Fasano, 78

Infine eseguendo la terza operazione relazionale prevista ossia

$$\mathbf{T3} = \Pi_{\text{Cognome, Nome}}(\mathbf{T2})$$

otterremo la relazione finale (risultato):

<b>T3</b>	Cognome	Nome
	Rossi	Paolo
	Bianchi	Aldo

### Nota Bene

Le due query appena eseguite risultano equivalenti ossia danno origine agli stessi risultati, pur utilizzando operazioni relazionali differenti (stessi risultati ma “procedimento” diverso).

Quindi, prima di progettare una query, occorre analizzare attentamente il testo che la descrive ricavando da esso tutte le informazioni richieste e le condizioni imposte sui vari attributi, facendo ben attenzione a coinvolgere esclusivamente le relazioni strettamente necessarie (al fine di evitare possibili inefficienze)

**Domanda: sarebbe stato possibile eseguire la query 2 in un modo più efficiente, ossia minimizzando il numero di n-ple coinvolte per la sua risoluzione?**

Questo quesito si pone ogni qualvolta ci troviamo ad eseguire **PRIMA un EQUI-JOIN e POI una SELEZIONE** sul risultato ottenuto.

**Cosa avverrebbe se eseguiamo uno scambio di posizione tra le due operazioni da effettuare ossia eseguiamo PRIMA la SELEZIONE per POI eseguire l'EQUI-JOIN?** Ossia:

- 1) Eseguiamo una **SELEZIONE** sulla relazione **Scuola** con condizione **NomeScuola = "ISIS G. Tassinari"**.
- 2) Poi eseguiamo un **EQUI-JOIN** tra la relazione **Studente** e quella appena ottenuta uguagliando i valori della chiave esterna **CodScuola1** (sulla relazione **Studente**) con quelli della chiave primaria **CodScuola** (della relazione **Scuola**).
- 3) Successivamente operiamo una **proiezione** (taglio verticale) rispetto agli attributi scelti **Studente.Cognome, Studente.Nome**.

$$\underbrace{\Pi_{\text{Cognome, Nome}} \left( \underbrace{\text{Studente} \bowtie \left( \underbrace{\sigma_{\text{NomeScuola} = \text{"ISIS G. Tassinari"}} (\text{Scuola})}_{1^\circ} \right)}_{2^\circ} \right)}_{3^\circ}$$

CodScuola1 = CodScuola

Anche questa interrogazione può essere quindi vista come una interrogazione composta da tre interrogazioni (due elementari ed una derivata) ossia:

la prima interrogazione:  $T1 = \sigma_{\text{NomeScuola} = \text{"ISIS G. Tassinari"}} (\text{Scuola})$

la seconda interrogazione:  $T2 = (\text{Studente} \bowtie T1)$   
CodScuola1 = CodScuola

la terza interrogazione:  $T3 = \Pi_{\text{Cognome, Nome}} (T2)$

Allora eseguiamo innanzitutto **la prima operazione relazionale** prevista:

$$T1 = \sigma_{\text{NomeScuola} = \text{"ISIS G. Tassinari"}} (\text{Scuola})$$

Otterremo la seguente tabella

<b>T1</b>	<u>CodScuola</u>	NomeScuola	Indirizzo
	NATF091011	ISIS G. Tassinari	Via Fasano, 78

**Eseguiamo poi la seconda operazione relazionale** prevista

$$T2 = (\text{Studente} \bowtie T1)$$

CodScuola1 = CodScuola

Tale operazione come ricordiamo, risulta essere un'operazione derivata che da come risultato una tabella le cui n-ple si ottengono dal seguente procedimento:

- a) prima di tutto dobbiamo eseguire il prodotto cartesiano delle relazioni **Studente** e **T1**

<b>Studente</b> <b>x</b> <b>T1</b>	<u>Matricola</u>	Cognome	Nome	CodScuola1	<u>CodScuola</u>	NomeScuola	Indirizzo
	M001	Rossi	Paolo	NATF091011	NATF091011	ISIS G. Tassinari	Via Fasano, 78
M002	Bianchi	Aldo	NATF091011	NATF091011	ISIS G. Tassinari	Via Fasano, 78	
M003	Verdi	Ada	NAXX123456	NATF091011	ISIS G. Tassinari	Via Fasano, 78	
M004	Neri	Maria	NAXX123456	NATF091011	ISIS G. Tassinari	Via Roma, 144	

b) poi sulla relazione **Studente X T1** così ottenuta, dobbiamo eseguire una **selezione** ponendo la condizione di equi-join **Studente.CodScuola1 = Scuola.CodScuola**.

Avremo come risultato la seguente relazione **T2**

<b>T2</b>	<u>Matricola</u>	Cognome	Nome	CodScuola1	<u>CodScuola</u>	NomeScuola	Indirizzo
	M001	Rossi	Paolo	NATF091011	NATF091011	ISIS G. Tassinari	Via Fasano, 78
	M002	Bianchi	Aldo	NATF091011	NATF091011	ISIS G. Tassinari	Via Fasano, 78

Infine eseguendo **la terza operazione relazionale** prevista ossia

$$\mathbf{T3} = \Pi_{\text{Cognome, Nome}}(\mathbf{T2})$$

otterremo la relazione finale (risultato):

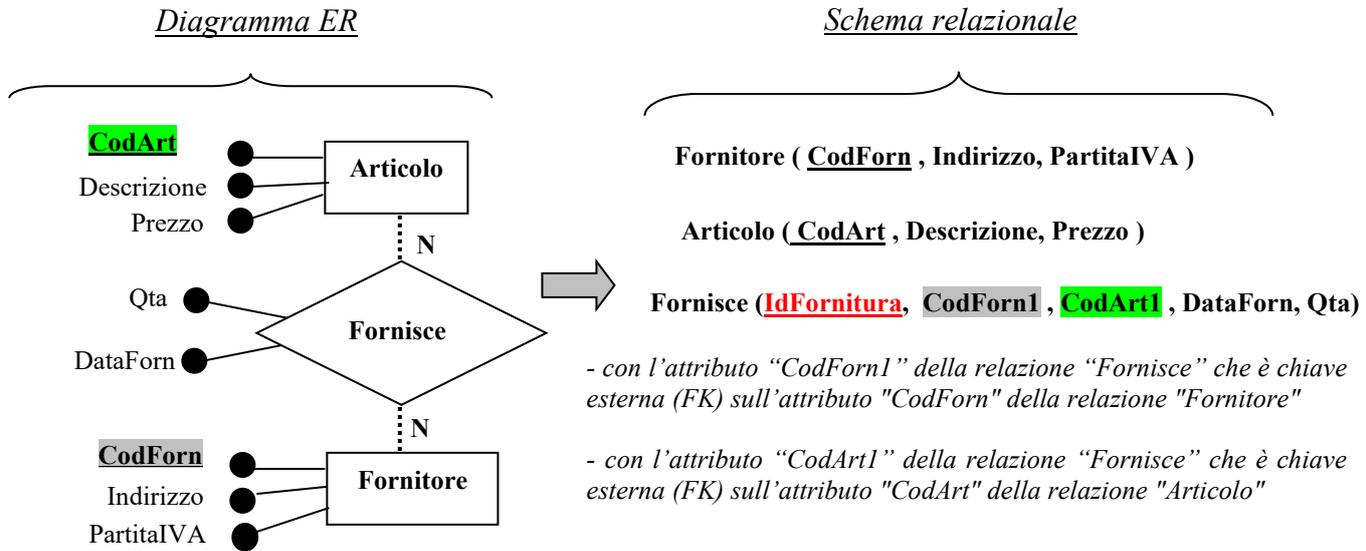
<b>T3</b>	Cognome	Nome
	Rossi	Paolo
	Bianchi	Aldo

### Conclusioni

**Confrontando le due versioni della query n. 2 se ne ricava che l'ultima proposta è senza dubbio più efficiente poiché minimizza il numero di ennuple coinvolte dal prodotto cartesiano comunque contenuto nell'operazione di EQUI-JOIN.**

**B) Interrogazione di RELAZIONI derivanti da associazioni di molteplicità N:N**

Aiutiamoci con un esempio



**QUERY n. 3: Elencare le partite IVA dei fornitori dell'articolo avente CodArt = "A04"**

Per quanto detto prima, utilizzando gli operatori dell'algebra relazionale, una possibile soluzione (sicuramente meno efficiente) è rappresentata dalla seguente espressione:

$$Q3_{-eff} = \Pi_{PartitaIVA} \left( \sigma_{CodArt1 = "A04"} \left( \text{Fornitore} \bowtie_{\text{CodForn} = \text{CodForn1}} \text{Fornisce} \right) \right)$$

La soluzione più efficiente si ottiene, come osservato in precedenza, eseguendo PRIMA la SELEZIONE e POI l'EQUI-JOIN proposto lasciando in entrambi i casi la proiezione come operazione finale ossia:

$$Q3_{+eff} = \Pi_{PartitaIVA} \left( \left( \text{Fornitore} \bowtie_{\text{CodForn} = \text{CodForn1}} \sigma_{CodArt1 = "A04"} \left( \text{Fornisce} \right) \right) \right)$$

Analizziamo in dettaglio la prima interrogazione proposta ossia Q3<sub>-eff</sub>

$$\Pi_{PartitaIVA} \left( \sigma_{CodArt1 = "A04"} \left( \text{Fornitore} \bowtie_{\text{CodForn} = \text{CodForn1}} \text{Fornisce} \right) \right)$$

Questa query risulta composta dalle seguenti tre sottointerrogazioni:

$$T1 = \text{Fornitore} \bowtie_{\text{CodForn} = \text{CodForn1}} \text{Fornisce}$$

$$T2 = \sigma_{CodArt1 = "A04"} (T1)$$

$$T3 = \Pi_{PartitaIVA} (T2)$$

Applichiamo questa interrogazione alla seguente istanza di base di dati che risulta così popolata in un certo istante t:

**Fornitore** (CodForn, Indirizzo, PartitaIVA)

<u>CodForn</u>	Indirizzo	PartitaIVA
F03	Via Po, 5	001234
F07	Via Bari, 5	001345
F16	Via Loi, 1	001333

Chiave primaria

**Articolo** (CodArt, Descrizione, Prezzo)

<u>CodArt</u>	Descrizione	Prezzo
A01	Batteria	100,00
A04	Antenna	75,00
A12	Radiatore	56,00

Chiave primaria

**Fornisce** (IdFornitura, CodForn1, CodArt1, DataForn, Qta)

<u>IdFornitura</u>	<u>CodForn1</u>	<u>CodArt1</u>	DataForn	Qta
0001	F03	A01	01-01-2019	10
0002	F03	A04	11-02-2019	25
0003	F16	A04	13-02-2019	20
0004	F03	A01	14-04-2019	30

Chiave primaria

Chiave esterna su relazione "Articolo"

Chiave esterna su relazione "Fornitore"

**La prima operazione relazionale** da eseguire è

$$T1 = \text{Fornitore} \bowtie_{\text{CodForn} = \text{CodForn1}} \text{Fornisce} \quad \text{che equivale a} \quad \sigma_{\text{CodForn} = \text{CodForn1}} (\text{Fornitore} \times \text{Fornisce})$$

a) Tale operazione derivata comporta dapprima l'esecuzione del **prodotto cartesiano** delle due relazioni ossia **Fornitore X Fornisce**

Fornitore X Fornisce	<u>CodForn</u>	Indirizzo	PartitaIVA	<u>IdFornitura</u>	<u>CodForn1</u>	<u>CodArt1</u>	DataForn	Qta
	<b>F03</b>	Via Po, 5	001234	0001	<b>F03</b>	A01	01-01-2019	10
	<b>F03</b>	Via Po, 5	001234	0002	<b>F03</b>	A04	11-02-2019	25
	F03	Via Po, 5	001234	0003	F16	A04	13-02-2019	20
	<b>F03</b>	Via Po, 5	001234	0004	<b>F03</b>	A01	14-04-2019	30
	F07	Via Bari, 5	001345	0001	F03	A01	01-01-2019	10
	F07	Via Bari, 5	001345	0002	F03	A04	11-02-2019	25
	F07	Via Bari, 5	001345	0003	F16	A04	13-02-2019	20
	F07	Via Bari, 5	001345	0004	F03	A01	14-04-2019	30
	F16	Via Loi, 1	001333	0001	F03	A01	01-01-2019	10
	F16	Via Loi, 1	001333	0002	F03	A04	11-02-2019	25
	<b>F16</b>	Via Loi, 1	001333	0003	<b>F16</b>	A04	13-02-2019	20
	F16	Via Loi, 1	001333	0004	F03	A01	14-04-2019	30

b) Poi sulla tabella ottenuta, viene eseguita la **selezione** contenuta nell'**equi-join** proposto:

<b>T1</b>	<b>CodForn</b>	<b>Indirizzo</b>	<b>PartitaIVA</b>	<b>IdFornitura</b>	<b>CodForn1</b>	<b>CodArt1</b>	<b>DataForn</b>	<b>Qta</b>
	<b>F03</b>	Via Po, 5	001234	0001	<b>F03</b>	A01	01-01-2019	10
	<b>F03</b>	Via Po, 5	001234	0002	<b>F03</b>	A04	11-02-2019	25
	<b>F03</b>	Via Po, 5	001234	0004	<b>F03</b>	A01	14-04-2019	30
	<b>F16</b>	Via Loi, 1	001333	0003	<b>F16</b>	A04	13-02-2019	20

**La seconda operazione relazionale** da eseguire è

$$\mathbf{T2} = \sigma_{\text{CodArt1} = \text{"A04"}}(\mathbf{T1})$$

che una volta eseguita darà origine alla relazione

<b>T2</b>	<b>CodForn</b>	<b>Indirizzo</b>	<b>PartitaIVA</b>	<b>IdFornitura</b>	<b>CodForn1</b>	<b>CodArt1</b>	<b>DataForn</b>	<b>Qta</b>
	F03	Via Po, 5	001234	0002	F03	<b>A04</b>	11-02-2019	25
	F16	Via Loi, 1	001333	0003	F16	<b>A04</b>	13-02-2019	20

**La terza (ultima) operazione relazionale** da eseguire è

$$\mathbf{T3} = \Pi_{\text{PartitaIVA}}(\mathbf{T2})$$

che una volta eseguita darà origine alla relazione (finale)

<b>T3</b>	<b>PartitaIVA</b>
	<b>001234</b>
	<b>001333</b>

<b>QUERY n. 4: Elencare le partite IVA dei fornitori dell'articolo con descrizione "Antenna"</b>
--

Per quanto detto prima, utilizzando gli operatori dell'algebra relazionale, una possibile soluzione (sicuramente meno efficiente) è rappresentata dalla seguente espressione:

$$Q4_{-eff} = \Pi_{PartitaIVA} \left( \underbrace{\left( \underbrace{\left( \underbrace{\text{Fornitore} \bowtie \text{Fornisce}}_{1^\circ} \right) \bowtie \text{Articolo}}_{2^\circ} \right)}_{3^\circ} \right)_{4^\circ}$$

CodForn = CodForn1      CodArt1 = CodArt

La soluzione più efficiente in questo caso si ottiene effettuando l'ultimo EQUI-JOIN dopo la selezione:

$$Q4_{+eff} = \Pi_{PartitaIVA} \left( \underbrace{\left( \underbrace{\text{Fornitore} \bowtie \text{Fornisce}}_{1^\circ} \right) \bowtie \underbrace{\left( \sigma_{Descrizione = "Antenna"}(\text{Articolo}) \right)}_{2^\circ} \right)_{3^\circ}$$

CodForn = CodForn1      CodArt1 = CodArt

Analizziamo in dettaglio la prima interrogazione proposta ossia **Q4<sub>-eff</sub>**

Questa query risulta composta dalle seguenti quattro sottointerrogazioni, da eseguire una dopo l'altra:

$$T1 = \text{Fornitore} \bowtie \text{Fornisce}$$

CodForn = CodForn1

$$T2 = T1 \bowtie \text{Articolo}$$

CodArt1 = CodArt

$$T3 = \sigma_{Descrizione = "Antenna"}(T2)$$

$$T4 = \Pi_{PartitaIVA}(T3)$$

a) La prima operazione relazionale da eseguire è il seguente **EQUI-JOIN** :

**T1** = **Fornitore**  $\bowtie$  **Fornisce** che equivale a  $\sigma_{\text{CodForn} = \text{CodForn1}}$  (**Fornitore X Fornisce**)  
CodForn = CodForn1

Fornitore X Fornisce	<u>CodForn</u>	<u>Indirizzo</u>	<u>PartitaIVA</u>	<u>IdFornitura</u>	<u>CodForn1</u>	<u>CodArt1</u>	<u>DataForn</u>	<u>Qta</u>
		<b>F03</b>	Via Po, 5	001234	0001	<b>F03</b>	A01	01-01-2019
	<b>F03</b>	Via Po, 5	001234	0002	<b>F03</b>	A04	11-02-2019	25
	F03	Via Po, 5	001234	0003	F16	A04	13-02-2019	20
	<b>F03</b>	Via Po, 5	001234	0004	<b>F03</b>	A01	14-04-2019	30
	F07	Via Bari, 5	001345	0001	F03	A01	01-01-2019	10
	F07	Via Bari, 5	001345	0002	F03	A04	11-02-2019	25
	F07	Via Bari, 5	001345	0003	F16	A04	13-02-2019	20
	F07	Via Bari, 5	001345	0004	F03	A01	14-04-2019	30
	F16	Via Loi, 1	001333	0001	F03	A01	01-01-2019	10
	F16	Via Loi, 1	001333	0002	F03	A04	11-02-2019	25
	<b>F16</b>	Via Loi, 1	001333	0003	<b>F16</b>	A04	13-02-2019	20
	F16	Via Loi, 1	001333	0004	F03	A01	14-04-2019	30

<b>T1</b>	<u>CodForn</u>	<u>Indirizzo</u>	<u>PartitaIVA</u>	<u>IdFornitura</u>	<u>CodForn1</u>	<u>CodArt1</u>	<u>DataForn</u>	<u>Qta</u>
	<b>F03</b>	Via Po, 5	001234	0001	<b>F03</b>	A01	01-01-2019	10
	<b>F03</b>	Via Po, 5	001234	0002	<b>F03</b>	A04	11-02-2019	25
	<b>F03</b>	Via Po, 5	001234	0004	<b>F03</b>	A01	14-04-2019	30
	<b>F16</b>	Via Loi, 1	001333	0003	<b>F16</b>	A04	13-02-2019	20

b) La seconda operazione relazionale da eseguire è il secondo EQUI-JOIN :

**T2** = **T1**  $\bowtie$  **Articolo**      che, ricordiamo, equivale a  $\sigma_{\text{CodArt1} = \text{CodArt}}$  (**T1** X **Articolo**)  
 CodArt1 = CodArt

<b>T1</b> X <b>Articolo</b>	<u>CodForn</u>	<u>Indirizzo</u>	<u>PartitaIVA</u>	<u>IdFornitura</u>	<u>CodForn1</u>	<u>CodArt1</u>	<u>DataForn</u>	<u>Qta</u>	<u>CodArt</u>	<u>Descrizione</u>	<u>Prezzo</u>
	F03	Via Po, 5	001234	0001	F03	A01	01-01-2019	10	A01	Batteria	100.00
	F03	Via Po, 5	001234	0001	F03	A01	01-01-2019	10	A04	Antenna	75.00
	F03	Via Po, 5	001234	0001	F03	A01	01-01-2019	10	A01	Radiatore	56.00
	F03	Via Po, 5	001234	0002	F03	A04	11-02-2019	25	A01	Batteria	100.00
	F03	Via Po, 5	001234	0002	F03	A04	11-02-2019	25	A04	Antenna	75.00
	F03	Via Po, 5	001234	0002	F03	A04	11-02-2019	25	A01	Radiatore	56.00
	F03	Via Po, 5	001234	0004	F03	A01	14-04-2019	30	A01	Batteria	100.00
	F03	Via Po, 5	001234	0004	F03	A01	14-04-2019	30	A04	Antenna	75.00
	F03	Via Po, 5	001234	0004	F03	A01	14-04-2019	30	A01	Radiatore	56.00
	F16	Via Loi, 1	001333	0003	F16	A04	13-02-2019	20	A01	Batteria	100.00
	F16	Via Loi, 1	001333	0003	F16	A04	13-02-2019	20	A04	Antenna	75.00
	F16	Via Loi, 1	001333	0003	F16	A04	13-02-2019	20	A01	Radiatore	56.00

<b>T2</b>	<u>CodForn</u>	<u>Indirizzo</u>	<u>PartitaIVA</u>	<u>IdFornitura</u>	<u>CodForn1</u>	<u>CodArt1</u>	<u>DataForn</u>	<u>Qta</u>	<u>CodArt</u>	<u>Descrizione</u>	<u>Prezzo</u>
	F03	Via Po, 5	001234	0001	F03	A01	01-01-2019	10	A01	Batteria	100.00
	F03	Via Po, 5	001234	0001	F03	A01	01-01-2019	10	A01	Radiatore	56.00
	F03	Via Po, 5	001234	0002	F03	A04	11-02-2019	25	A04	Antenna	75.00
	F03	Via Po, 5	001234	0004	F03	A01	14-04-2019	30	A01	Batteria	100.00
	F03	Via Po, 5	001234	0004	F03	A01	14-04-2019	30	A01	Radiatore	56.00
	F16	Via Loi, 1	001333	0003	F16	A04	13-02-2019	20	A04	Antenna	75.00

c) La terza operazione relazionale da eseguire è la seguente SELEZIONE :

$$T3 = \sigma_{\text{Descrizione} = \text{"Antenna"}}(T2)$$

<b>T2</b>	<u>CodForn</u>	<u>Indirizzo</u>	<u>PartitaIVA</u>	<u>IdFornitura</u>	<u>CodForn1</u>	<u>CodArt1</u>	<u>DataForn</u>	<u>Qta</u>	<u>CodArt</u>	<u>Descrizione</u>	<u>Prezzo</u>
	F03	Via Po, 5	001234	0001	F03	A01	01-01-2019	10	A01	Batteria	100.00
	F03	Via Po, 5	001234	0001	F03	A01	01-01-2019	10	A01	Radiatore	56.00
⇒	F03	Via Po, 5	001234	0002	F03	A04	11-02-2019	25	A04	Antenna	75.00
	F03	Via Po, 5	001234	0004	F03	A01	14-04-2019	30	A01	Batteria	100.00
	F03	Via Po, 5	001234	0004	F03	A01	14-04-2019	30	A01	Radiatore	56.00
⇒	F16	Via Loi, 1	001333	0003	F16	A04	13-02-2019	20	A04	Antenna	75.00



<b>T3</b>	<u>CodForn</u>	<u>Indirizzo</u>	<u>PartitaIVA</u>	<u>IdFornitura</u>	<u>CodForn1</u>	<u>CodArt1</u>	<u>DataForn</u>	<u>Qta</u>	<u>CodArt</u>	<u>Descrizione</u>	<u>Prezzo</u>
	F03	Via Po, 5	001234	0002	F03	A04	11-02-2019	25	A04	Antenna	75.00
	F16	Via Loi, 1	001333	0003	F16	A04	13-02-2019	20	A04	Antenna	75.00

d) La quarta operazione relazionale da eseguire è la seguente PROIEZIONE :

$$T4 = \Pi_{\text{PartitaIVA}}(T3)$$

<b>T4</b>	<u>PartitaIVA</u>
	001234
	001333

**Esercizi di approfondimento:**

**Query A) Utilizzando i medesimi dati iniziali forniti per le tre tabelle “Fornitore”, “Fornisce” ed “Articolo” eseguire passo passo la seguente query:**

$$Q_{A+eff} = \Pi_{PartitaIVA} \left( \text{(Fornitore} \bowtie \sigma_{\substack{\text{CodArt1} = \text{“A04”} \\ \text{CodForn} = \text{CodForn1}}} \text{(Fornisce))} \right)$$

**Query B) Vogliamo conoscere la descrizione degli articoli forniti dal fornitore avente CodForn = "F07"**

**Query C) Vogliamo conoscere la partita IVA e la descrizione degli articoli forniti da tutti i fornitori**

**Query D) Elencare le partite IVA dei fornitori dell'articolo avente come descrizione "Batteria"**

**Query E) Vogliamo conoscere la partita IVA, la descrizione e la quantità degli articoli forniti dal fornitore con codice "F03"**