La normalizzazione

Per normalizzazione intendiamo quel procedimento che consente di verificare se la definizione dello schema corrisponde ai canoni standard di correttezza della base dei dati e, in caso, avvalendosi di un preciso insieme di regole, di riportare le tabelle in quelle che sono definite le forme normali.

La teoria della normalizzazione avviene rispettando le seguenti regole:

- Ogni tabella deve avere una chiave primaria;
- Ogni campo deve contenere un solo valore;
- I campi di una tabella non devono dipendere da altri campi che non siano la chiave primaria;
- Bisogna evitare le ripetizioni e la ridondanza dei dati;

In fase di progettazione occorre verificare che ogni tabella abbia una chiave primaria.

Data	Cliente	Quantità	Articolo	Prezzo	Prezzotot	Tipopagam
20/04/06	Rossi	25	AXY	50,25	1.256,25	Contanti
20/04/06	Neri	120	BCV	25,80	3.096,00	Contanti
20/04/06	Rossi	100	GTR	22,14	2.214,00	Bonifico
20/04/06	Rossi	45	FFT	50,25	2.261,25	Assegni
27/04/06	Verdi	100	AXY	50,25	5.025,00	Contanti

La tabella non contiene chiavi candidate: ogni colonna, infatti, contiene ripetizioni. È necessario aggiungere un nuovo campo ad esempio codice ordine.

Codord	Data	Cliente	Quantità	Articolo	Prezzo	Prezzotot	Tipopagam
1	20/04/06	Rossi	25	AXY	50,25	1.256,25	Contanti
2	20/04/06	Neri	120	BCV	25,80	3.096,00	Contanti
3	20/04/06	Rossi	100	GTR	22,14	2.214,00	Bonifico
4	20/04/06	Rossi	45	FFT	50,25	2.261,25	Assegni
5	27/04/06	Verdi	100	AXY	50,25	5.025,00	Contanti

Una volta sicuri che le tabelle contengano le chiavi primarie occorre accertarsi che:
ogni campo abbia un solo valore atomico (deve essere un campo semplice, quindi non composto e non multiplo): è questa la prima forma normale (1FN) chiamata anche forma atomica.

codpersona	Cognome	Nome	indirizzo
1	Alberti	Giovanni	Via Mantova 5 Brescia
2	Ughi	Amilcare	Viale Vittorio Veneto 67 Milano
3	Tramezzi	Gesualdo	Via Torquato Tasso 45 Roma

La tabella ha una chiave primaria quindi la prima regola è confermata. Il campo indirizzo contiene più valori: la via e la città nello stesso campo. Per normalizzare la relazione si suddivide il campo indirizzo in modo che ogni informazione sia rappresentata con un apposito attributo.

Per sapere fino a dove suddividere il campo si deve consultare l'analisi dei requisiti. In generale tutti i dati che servono per confronti o ricerche, nel normale utilizzo del database, devono essere isolati.

codpersona	Cognome	Nome	indirizzo	città
1	Alberti	Giovanni	Via Mantova 5	Brescia
2	Ughi	Amilcare	Viale Vittorio Veneto 67	Milano
3	Tramezzi	Gesualdo	Via Torquato Tasso 45	Roma

Altro esempio di relazione non normalizzata è quello in cui sono presenti degli attributi multipli.

codpersona	Cognome	Nome	recapititel		
1	Alberti	Giovanni	030123456	340123456	
2	Ughi	Amilcare	026543210	029876543	3331231231
3	Tramezzi	Gesualdo	0613579012		

Analizzando la seguente relazione è come se a 1 persona corrispondessero N numeri telefonici.
Per avere uno schema in 1FN si sostituisce la relazione non normalizzata con due relazioni:

1. *una simile a quella di origine ma senza l'attributo multiplo;*
2. *l'altra contenente la chiave primaria della prima relazione e un attributo semplice che contiene ogni singolo valore della sequenza originale.*

codpersona	Cognome	Nome
1	Alberti	Giovanni
2	Ughi	Amilcare
3	Tramezzi	Gesualdo

codpersona	recapititel
1	030123456
1	340123456
2	026543210
2	029876543
2	3331231231
3	0613579012

Una relazione è in seconda forma normale (2NF):

1. se è in 1NF
2. ogni attributo non chiave dipende funzionalmente e completamente dalla chiave primaria.

Questo significa che tutti i campi diversi dalla chiave primaria, devono dipendere dall'intera chiave primaria e non da una sua parte.

Prendiamo in considerazione il seguente schema di relazione:

ORDINI(codordini, codcliente, codprodotto, dataordine, quantità, prezzounit, descrizione)

Leggendo lo schema ci accorgiamo che non è in 2NF perché al suo interno sono memorizzate informazioni riguardanti più di un oggetto, cioè più elementi distinti.

Codordini	Codcliente	codprodotto	Dataordine	quantità	Prezzounit	descrizione
1	C1	AXY	20/04/06	25	50,25	Alfa
1	C1	BCV	20/04/06	120	25,80	Beta
1	C1	GTR	20/04/06	100	22,14	Delta
2	C24	FFT	20/04/06	45	50,25	Gamma
3	C56	AXY	27/04/06	120	50,25	Alfa

Lo schema, verificando la relazione, presenta alcune anomalie:

- inserimento: non è possibile inserire un nuovo articolo in magazzino fino a quando non viene ordinato.
- Cancellazione: se si cancellano la seconda e terza tupla si perdono informazioni sugli articoli che compaiono in esse;
- Aggiornamento: se varia il prezzo di un articolo occorre aggiornare tutte le tuple in cui compare compreso il relativo totale.

Per risolvere questi problemi, si deve scomporre la relazione in relazioni più semplici, ciascuna relativa ad una data categoria: gli ordini, i prodotti e i prodotti ordinati. Le relazioni devono essere collegate tramite le chiavi primarie.
Il nuovo schema relazionale sarà il seguente:

PRODOTTIORDINATI(Codordine, Codprodotto, quantità)
PRODOTTI(codprodotto, prezzounit, descrizione)
ORDINI(codordine, dataordine, codcliente)

Le anomalie non sono più presenti e attraverso le chiavi primarie è possibile reperire tutte le informazioni. Bisogna fare in modo che tutti gli attributi non chiave in uno schema di relazione dipendano funzionalmente dall'intera chiave primaria. *Il procedimento usato è il seguente:*

1. *nello schema originario rimane la chiave primaria e tutti gli attributi non chiave, se ci sono, che dipendono completamente da essa.*
2. *si crea un nuovo schema di relazione per ogni parte di chiave primaria da cui dipendono completamente altri attributi non chiave.*

Una relazione è in terza forma normale (3NF):

1. se è in 2NF;
2. ogni attributo non chiave dipende direttamente dalla chiave e non dipenda da altri attributi non chiave;

PRODOTTIORDINATI(Codordine, Codprodotto, quantità, percsconto)

Codordine	Codprodotto	quantità	percsconto
1	AXY	25	1,2
1	BCV	120	2
1	GTR	100	1,8
2	FFT	45	0,5
3	AXY	120	2

In questo caso, pur rispettando la 2NF osserviamo delle anomalie:

- inserimento: non è possibile inserire la percentuale di sconto relativa ad una certa quantità, sino a quando quest'ultima non compare in un ordine;
- cancellazione: se si cancella la prima tupla si perde l'informazione che per 25 pezzi si ha uno sconto del 1,2%;
- aggiornamento: se varia la percentuale di sconto per una certa quantità occorre aggiornare tutte le tuple interessate.

Siamo di fronte al caso che un attributo non chiave (percsconto) dipende da un altro attributo non chiave (quantità). Lo schema pertanto sarà così trasformato:

PRODOTTIORDINATI(Codordine, Codprodotto, quantità)
SCONTI(quantità, percsconto)

Quindi tutti gli attributi non chiave dipendono direttamente e non transitivamente dall'intera chiave primaria. *Per normalizzare si procede in questo modo:*

1. *nello schema originario rimane la chiave primaria e tutti gli attributi non chiave che dipendono direttamente da essa.*
2. *si crea un nuovo schema di relazione per ogni attributo da cui dipendono altri attributi non chiave.*

A questo punto possiamo dire che in uno schema in seconda forma normale in cui c'è un solo attributo non chiave è sicuramente in terza forma normale.

PRODOTTIORDINATI(Codordine, Codprodotto, quantità, prezzotot)
PRODOTTI(codprodotto, descrizione, prezzounit)

L'attributo prezzotot dipende dal campo quantità della relazione PRODOTTIORDINATI e dal campo prezzounit della relazione PRODOTTI. Se venisse il modificato il valore del campo quantità o del campo prezzounit dovrebbe essere modificato anche il valore del campo prezzotot. Il campo prezzotot, essendo un campo calcolato, dovrebbe essere eliminato dalla tabella ed il totale calcolato in fase di esecuzione con opportune interrogazioni.

Esiste anche una quarta e quinta forma normale ma non viene affrontata in questo corso di studio. Mentre la forma normale di Boyce-Codd (BCNF) dice: *una relazione è in BCNF quando è in prima forma normale e in essa ogni determinante è una chiave candidata, cioè ogni attributo dal quale dipendono altri attributi può svolgere la funzione di chiave.*

Vincoli d'integrità

Le operazioni che avvengono su una base di dati devono essere eseguite rispettando un insieme di regole che servono a non compromettere l'integrità dei dati. Il vincolo d'integrità è una proprietà che deve essere soddisfatta dalle istanze che rappresentano informazioni corrette per l'applicazione che utilizza la base dei dati. Possiamo classificare i vincoli in:

- **vincoli intrarelazionali o interni**, che sono quei vincoli definiti all'interno di una singola relazione. Si suddividono in:
  - **vincoli su singola ennupla** che esprimono una condizione:
    - **sul dominio degli attributi** sono quei vincoli che coinvolgono un solo attributo il cui soddisfacimento può essere verificato facendo riferimento a un singolo valore alla volta;
    - **su più attributi** sono quei vincoli che coinvolgono più attributi, ma sempre di ciascuna ennupla, indipendente dalle altre.

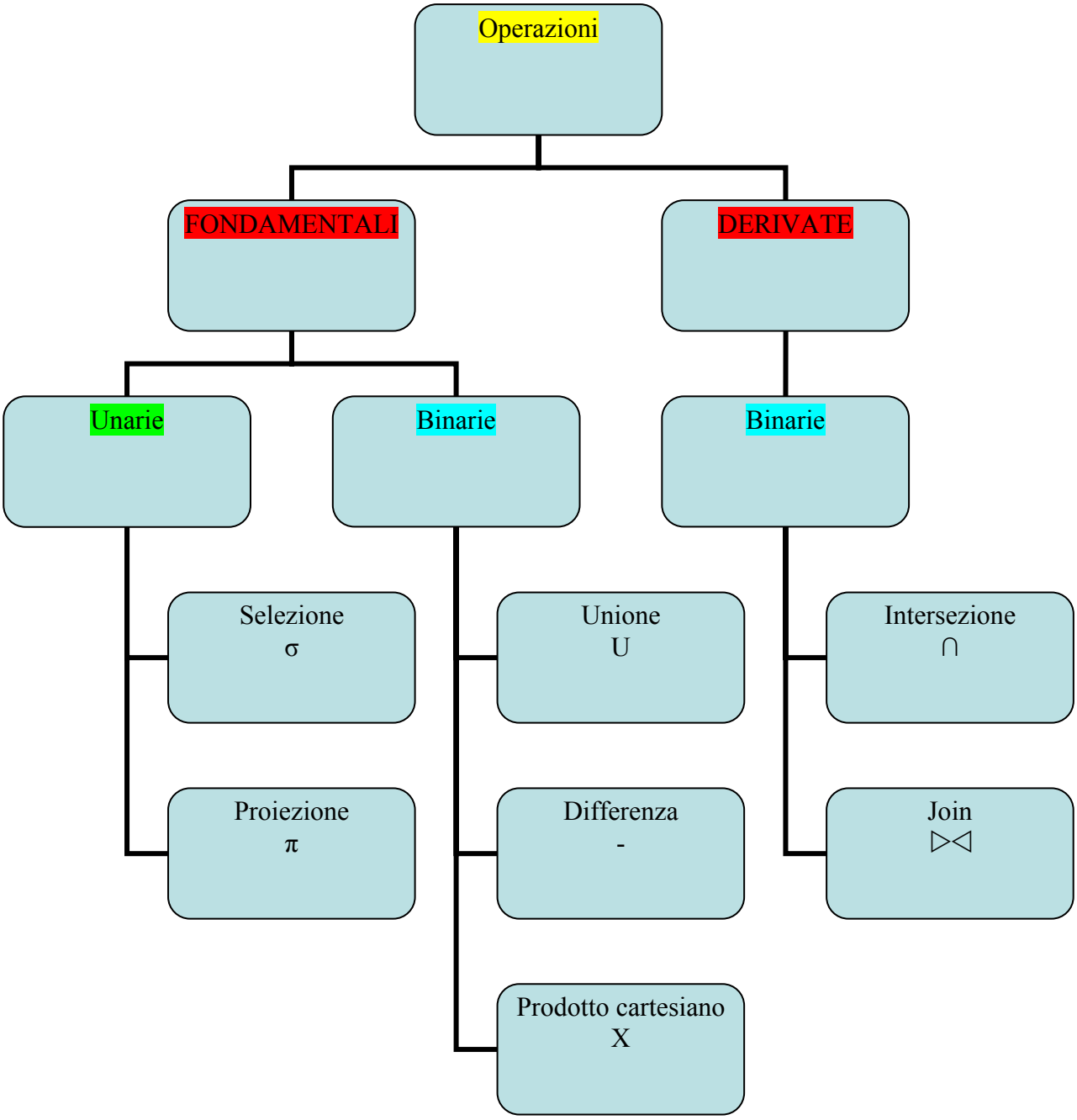
consideriamo la relazione:
DIPENDENTE(matricola,stipendioloro,trattenute,dataassunzione,datanascita)
V1(DIPENDENTE): stipendiolordo>0
V2(DIPENDENTE): dataassunzione>datanascita
V3(DIPENDENTE): trattenute>0
Una istanza della relazione soddisfa questo vincolo solo quando tutte le tuple lo soddisfano. I vincoli V1e V3 sono vincoli di dominio, V2 è un vincolo su più attributi.

  - **Vincoli su più ennuple** che coinvolgono i valori di più ennuple. Sono i vincoli di chiave primaria. (*esempio: uno studente non può avere lo stesso numero di matricola di un altro studente. Questo è un vincolo di chiave primaria*)
- **Vincoli interrelazionali o esterni** che sono definiti tra più relazioni. Tra questi rientrano i vincoli referenziali.

I vincoli di integrità referenziale rivestono particolare importanza. Abbiamo visto che per stabilire un legame tra due o più relazioni, utilizziamo le chiavi primarie delle due relazioni (creando altre relazioni che contengono i valori di queste chiavi), le quali prendono il nome di **chiavi esterne**. I vincoli di integrità referenziale riguardano i valori assunti dalle chiavi esterne nelle relazioni. Per assicurare l'integrità referenziale, prima di cancellare una tupla occorre verificare che non ci siano tuple in altre relazioni che facciano riferimento alla tupla da cancellare. L'integrità referenziale è assicurata dal DBMS che prevede la possibilità di dichiarare le regole di validazione attraverso appositi linguaggi dichiarativi. Tali regole sono mantenute su speciali archivi detti **cataloghi delle regole**.

Le operazioni relazionali

Tutte le operazioni che consentono di interrogare una base di dati relazionale utilizzando un linguaggio per l'interrogazione di tipo non procedurale ed utilizzando un approccio basato sull'algebra relazionale vengono definite operazioni relazionali.
Il risultato di una interrogazione è una relazione.



Due relazioni R e 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 relazioni U**

Date due relazioni compatibili R e S, l'unione di R con S è la relazione ottenuta dall'unione insiemistica delle due relazioni: **R U S = {t | t ε R OR t ε S }**

Grado(R U S) = Grado(R) = Grado(S)
Card(R U S)= Card(R) + Card(S) – numero tuple ripetute

R=cli1sem06

NOME	A	B
Rossi		
Bianchi		
Verdi		

S=cli2sem06

NOME	A	B
Gialli		
Bianchi		
Neri		

R U S

NOME	A	B
Rossi		
Bianchi		
Verdi		
Gialli		
Neri		

Si verifica aggiungendo in coda le righe di una tabella a quelle di una seconda tabella per produrre una terza tabella. Le righe duplicate vengono eliminate.

2 – **Differenza di relazioni -**

Date due relazioni compatibili R e S, la differenza di R con S è la relazione data dalla differenza insiemistica delle due relazioni: **R - S = {t | t ε R AND t not ε S }**
La differenza non gode della proprietà commutativa quindi **R – S ≠ S – R**

Grado(R - S) = Grado(R) = Grado(S)
Card(R - S)= Card(R) – numero tuple presenti anche in S

R=cli2006

NOME	A	B
Rossi		
Bianchi		
Verdi		

S=cli20005

NOME	A	B
Gialli		
Bianchi		
Neri		

R - S

NOME	A	B
Rossi		
Verdi		

Il risultato è una terza tabella che contiene le righe che esistono nella prima tabella ma che non compaiono nella seconda.

3 – **Proiezione di una relazione π**

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

$$\pi_{A1,A2,\ldots,Ak}(R) = \{t[A_1,A_2,\ldots,A_k] \mid t \in R\}$$

Grado(S) ≤ Grado(R)
Card(S) ≤ Card(R) perchè le tuple presenti nella proiezione possono essere di numero inferiore a quelle di R perché le tuple duplicate vengono scartate.

R= clienti

Codcli	Nome	Agente	Indirizzocli
C001	Bianchi	Alben	Via Po 13
C002	Neri	Reitano	Via Londra 11
C003	Verdi	Alben	Via Bologna 1

$$S = \pi_{agente}(R)$$

Agente
Alben
Reitano

Recupera un sottoinsieme di colonne da una tabella

4 – **Restrizione di una relazione σ**

Data una relazione R e un predicato P sui suoi attributi, si definisce restrizione o selezione di R a P la relazione costituita dalle tuple di R che soddisfano P.

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

Grado(S) = Grado(R)
Card(S) ≤ Card(R) è uguale solo quando tutte le tuple soddisfano P

R= clienti

Codcli	Nome	città	Indirizzocli
C001	Bianchi	Milano	Via Po 13
C002	Neri	Milano	Via Londra 11
C003	Verdi	Roma	Via Bologna 1

$$S = \sigma_{città = \text{“Milano”}}(R)$$

Codcli	Nome	città	Indirizzocli
C001	Bianchi	Milano	Via Po 13
C002	Neri	Milano	Via Londra 11

Recupera un sottoinsieme di righe da una tabella basandosi su una condizione imposta ai valori contenuti in una o più colonne

5 – **Prodotto cartesiano di due relazioni X**

Date due relazioni R e S, rispettivamente di grado g1 e g2 e cardinalità c1 e c2, il prodotto di R e S è la relazione di grado g1+g2 e cardinalità c1xc2, le cui tuple si ottengono concatenando ogni tupla di R con ogni tupla di S.

R X S = {t | t = r conc s, r ε R, s ε S }

Per evitare ambiguità nei nomi degli attribute di R X S, occorre che I nomi degli attribute di R e S siano diversi tra loro. La chiave primaria della relazione prodotto è data dall'unione delle chavi primarie delle due relazioni di partenza.

Grado(R X S) = g1 + g2
Card(R X S)= c1 x c2

R=alunni

matr	nome	classe
111111	Rossi	3
222222	Bianchi	3
333333	Verdi	3

S=testi

codtesto	titolo	materia
1	Italiano oggi	italiano
2	algebra	matematica
3	Dio e l'uomo	religione

R X S

matr	nome	classe	codtesto	titolo	materia
111111	Rossi	3	1	Italiano oggi	italiano
111111	Rossi	3	2	algebra	matematica
111111	Rossi	3	3	Dio e l'uomo	religione
222222	Bianchi	3	1	Italiano oggi	italiano
222222	Bianchi	3	2	algebra	matematica
222222	Bianchi	3	3	Dio e l'uomo	religione
333333	Verdi	3	1	Italiano oggi	italiano
333333	Verdi	3	2	algebra	matematica
333333	Verdi	3	3	Dio e l'uomo	religione

E' la concatenazione di ogni riga di una tabella con ogni riga della seconda tabella

6 – **Intersezione di due relazioni** ∩

Date due relazioni compatibili R e S, l'intersezione di R e S restituisce la relazione composta da tutte le tuple presenti sia in R sia in S: **R ∩ S ={t | t ε R AND t ε S }**

Grado(R∩S) = Grado(R)=Grado(S)
Card(R∩S) ≤ alla cardinalità minore tra Card(R) e Card(S)

R= clienti2004

Codcli	Nome	città	Indirizzocli
C001	Bianchi	Milano	Via Po 13
C002	Neri	Milano	Via Londra 11
C003	Rossi	Roma	Via Bologna 1

S=clienti2005

Codcli	Nome	città	Indirizzocli
C001	Verdi	Como	Via Lodi 130
C002	Neri	Milano	Via Londra 11
C003	Rossi	Roma	Via Bologna 1

R ∩ S

Codcli	Nome	città	Indirizzocli
C002	Neri	Milano	Via Londra 11
C003	Rossi	Roma	Via Bologna 1

E' una terza tabella che contiene le righe comuni ad entrambe. Le tabelle devono essere compatibili.

7 – **Giunzione di due relazioni** $\rhd\lhd$

Date due relazioni R e S rispettivamente di grado g1 e g2, l'operazione di giunzione naturale (join) di R e S su un attributo A di R e un attributo B di S, aventi lo stesso tipo, restituisce una relazione di grado (g1+g2-1) le cui tuple si ottengono con il seguente procedimento:

1. si effettua il prodotto cartesiano di R e S
2. sulla relazione così ottenuta si effettua una restrizione sulle tuple aventi gli attributi A e B, dello stesso valore;
3. la relazione così ottenuta ha le colonne A e B uguali, per cui si elimina una di esse.

$$R \rhd\lhd_{A=B} S$$

R=clienti

Codcli	Nome	città	codagenti
C001	Bianchi	Milano	1122
C002	Neri	Milano	2020
C003	Rossi	Roma	1122

S=agenti

codag	nominativo	telefono
1122	Rinaldi	6666666
2020	Torini	7777777

1) R X S

Codcli	Nome	città	codagenti	codag	nominativo	telefono
C001	Bianchi	Milano	1122	1122	Rinaldi	6666666
C001	Bianchi	Milano	1122	2020	Torini	7777777
C002	Neri	Milano	2020	1122	Rinaldi	6666666
C002	Neri	Milano	2020	2020	Torini	7777777
C003	Rossi	Roma	1122	1122	Rinaldi	6666666
C003	Rossi	Roma	1122	2020	Torini	7777777

2) $\sigma_{\text{codagenti = codag}}$ **(R X S)**

Codcli	Nome	città	codagenti	codag	nominativo	telefono
C001	Bianchi	Milano	1122	1122	Rinaldi	6666666
C002	Neri	Milano	2020	2020	Torini	7777777
C003	Rossi	Roma	1122	1122	Rinaldi	6666666

3) $R \underset{A=B}{\bowtie} S$

Codcli	Nome	città	codag	nominativo	telefono
C001	Bianchi	Milano	1122	Rinaldi	6666666
C002	Neri	Milano	2020	Torini	7777777
C003	Rossi	Roma	1122	Rinaldi	6666666

Grado $(R \underset{A=B}{\bowtie} S)$= (g1+g2-1) poiché l'attributo comune compare una volta sola

Card $(R \underset{A=B}{\bowtie} S)$ = non è prevedibile a priori in quanto dipende da quante righe nelle due tabelle hanno valori uguali per gli attributi su cui si effettua il join.