Basi di Dati
Introduzione al Calcolo relazionale

Corso B

Base dati di esempio

PAZIENTI

<u>COD</u>	Cognome	Nome	Residenza	AnnoNascita
A102	Necchi	Luca	TO	1950
B372	Rossigni	Piero	NO	1940
B543	Missoni	Nadia	TO	1960
B444	Missoni	Luigi	VC	2000
S555	Rossetti	Gino	AT	2010

REPARTI

<u>COD</u>	Nome-Rep	Primario
A	Chirurgia	203
B	Pediatria	574
C	Medicina	530
L	Lab-Analisi	530
R	Radiologia	405

RICOVERI

PAZ	Inizio	Fine	Reparto
A102	2/05/2014	9/05/2014	A
A102	2/12/2004	2/01/2005	A
S555	5/10/2014	3/12/2014	B
B444	1/12/2004	2/01/2005	B
S555	6/09/2015	1/11/2015	A

MEDICI

<u>MATR</u>	Cognome	Nome	Residenza	Reparto
203	Neri	Piero	AL	A
574	Bisi	Mario	MI	B
461	Bargio	Sergio	TO	B
530	Belli	Nicola	TO	C
405	Mizzi	Nicola	AT	R
501	Monti	Mario	VC	A

Base dati di esempio

Schema relazionale con vincoli di integrità referenziali:

pazienti(COD, Cognome, Nome, Residenza, AnnoNascita)

medici(MATR, Cognome, Nome, Residenza, Reparto)

reparti(COD, Nome-Rep, Primario)

ricoveri(PAZ, Inizio, Fine, Reparto)

Base di Dati "Impiegati"

IMPIEGATI

MATR	Cognome	Nome	Età	Stipendio
203	Neri	Piero	50	40
574	Bisi	Mario	60	60
461	Bargio	Sergio	30	61
530	Belli	Nicola	40	38
405	Mizzi	Nicola	55	60
501	Monti	Mario	25	35

ORGANIGRAMMA

Capo	Impiegato
203	405
203	501
574	203
574	530
405	461

IMPIEGATI(MATR, Cognome, Nome, Età, Stipendio)

ORGANIGRAMMA(Capo,Impiegato)

Altri linguaggi di interrogazione

- Dal punto di vista dell'utente, l'algebra relazionale è un linguaggio di tipo procedurale
  - L'utente indica le operazioni da compiere per arrivare al risultato, il DBMS sceglie poi la strategia ottimale
- E' possibile impostare la tematica delle interrogazioni sulle basi di dati con un approccio dichiarativo fondato sulla logica
- Noi tratteremo solo il **calcolo relazionale su tuple con dichiarazione di range**
- Non tratteremo invece il **calcolo sui domini**

Calcolo su tuple
con dichiarazione di range

L'interrogazione è composta da tre parti:

$$\{ T \mid L \mid F \}$$

- Target (T)
- Range list (L)
- Formula (F)

Esempio

Elencare i pazienti residenti in Torino

{ p.Nome, p.Cognome | p(PAZIENTI) | p.Residenza='TO'}

Formula

La formula è un predicato del primo ordine su predicati di base (atomi) del tipo

- $x.A_i \, \Theta \, costante$
- $x.A_i \, \Theta \, y.A_j$

x e *y* sono delle variabili e A_i e A_j sono degli attributi

Il predicato è costruito con i soliti operatori

- $\wedge$ (AND), $\vee$ (OR), $\neg$ (NOT), $\Rightarrow$ (implicazione)

In più abbiamo i quantificatori universali

- $\forall$(per ogni), $\exists$ (esiste)

Range list

La range list non è altro che l'introduzione di variabili abbinate a tavole di base, con la seguente sintassi

nome_variabile(Nome_tavola_di_base)

Target list

La target list è la lista di informazioni che voglio in uscita

Sintassi possibilii:

- *variabile.Attributo1, variabile.Attributo2…*
- *variabile.(Attributo1,Attributo2)*
- *variabile.** (visualizza tutti gli attributi)
- *Nome: variabile.Attributo*

Le variabili usate nella target list devono essere dichiarate nella Range list

Esempio

Elencare i pazienti residenti in Torino

- ***L***: *p(PAZIENTI)*
- ***F***: *p.Residenza='TO'*
- ***T***: *p.Nome, p.Cognome*

Alternative a T

- p.(Nome,Cognome)
- p.*
- NomePaziente: p.Nome

Esempio

Elencare i pazienti residenti in Torino

{ p.Nome, p.Cognome | p(PAZIENTI) | p.Residenza='TO'}

Esempio

Elencare i dati dei pazienti ricoverati nel reparto A (si vuole anche la data del ricovero)

- Devo usare due variabili:
  - **L**: *p(PAZIENTI),r(RICOVERI)*
- Mi serve il join tra pazienti e ricoveri
  - **F**: *p.COD=r.PAZ*
- Per finire ho bisogno di aggiungere la condizione sui reparti
  - **F**: *p.COD=r.PAZ* $\wedge$ *r.Reparto='A'*

Esempio

Elencare i dati dei pazienti ricoverati nel reparto A (si vuole anche la data del ricovero)

$$\{p.Cognome, p.Nome, r.Inizio \mid p(PAZIENTI), r(RICOVERI) \mid p.COD = r.PAZ \wedge r.Reparto = 'A' \}$$

Join in calcolo relazionale

Il join si costruisce semplicemente inserendo nella formula la condizione di confronto delle variabili (relazioni) coinvolte (preliminarmente dichiarate nella range list)

Variante con quantificazione esistenziale

Supponiamo che nella interrogazione non sia richiesta nessuna informazione sul ricovero (solo su paziente)

Se l'interrogazione non richiede nessuna informazione sul ricovero, cancello dalla target list il riferimento a ricovero

$$\{p.Cognome,p.Nome \mid p(PAZIENTI),r(RICOVERI) \mid p.COD=r.PAZ \wedge r.Reparto='A' \}$$

Variante con quantificazione esistenziale

Quando una variabile non viene espressa nella target list possiamo riformulare l'interrogazione utilizzando la quantificazione esistenziale:

$$\{p.Cognome, p.Nome \mid p(PAZIENTI) \mid \exists\, r(RICOVERI)(p.COD=r.PAZ \wedge r.Reparto='A')\}$$

Sintassi della quantificazione universale ed esistenziale

- *$\exists$ variabile(Relazione)(formula)*

- *$\forall$ variabile(Relazione)(formula)*

N.B.: la formula è un predicato del primo ordine che può contenere sia variabili libere che quantificate (vincolate), tuttavia le uniche variabili libere presenti nella formula devono essere dichiarate nella range list

La target list, quindi, utilizza solo variabili libere

Esempio

Elencare i medici curanti del paziente A102

L: m(MEDICI)

T: m.Cognome,m.Nome

F: $\exists$ *r(RICOVERI)(r.Reparto=m.Reparto$\wedge$r.PAZ='A102')*

$$\{m.Cognome,m.Nome \mid m(MEDICI) \mid \exists\, r(RICOVERI)(r.Reparto=m.Reparto \wedge r.PAZ='A102')\}$$

Esempio

Elencare i medici curanti del paziente Rossini Piero

L: m(MEDICI)

T: m.Cognome,m.Nome

F: $\exists$ *p(PAZIENTI)(*$\exists$ *r(RICOVERI)(p.Nome='Piero'* $\wedge$ *p.Cognome='Rossini'* $\wedge$ *p.COD=r.PAZ* $\wedge$ *r.Reparto=m.Reparto))*

{m.Cognome,m.Nome | m(MEDICI) | $\exists$ *p(PAZIENTI)(*$\exists$ *r(RICOVERI)(p.Nome='Piero'* $\wedge$ *p.Cognome='Rossini'* $\wedge$ *p.COD=r.PAZ* $\wedge$ *r.Reparto=m.Reparto))}*

Esempio

Elencare cognome e nome comune a pazienti e medici

$$\{p.Cognome, p.Nome \mid p(PAZIENTI) \mid \exists\, m(MEDICI)(m.Cognome = p.Cognome \wedge m.Nome = p.Nome)\}$$

oppure

$$\{m.Cognome, m.Nome \mid m(MEDICI) \mid \exists\, p(PAZIENTI)(p.Cognome = m.Cognome \wedge p.Nome = m.Nome)\}$$

Esempio con la negazione

Elencare i cognomi e nomi dei pazienti non medici

E' sufficiente anteporre il NOT alla quantificazione esistenziale

$$\{p.Cognome, p.Nome \mid p(PAZIENTI) \mid \neg \exists\, m(MEDICI)(m.Cognome=p.Cognome \wedge m.Nome=p.Nome)\}$$

Esempio con la negazione

Elencare i cognomi e nomi dei pazienti non medici

Alternativa applicando De Morgan

$$\{p.Cognome, p.Nome \mid p(PAZIENTI) \mid \forall m(MEDICI)(m.Cognome \neq p.Cognome \vee m.Nome \neq p.Nome)\}$$

Esempio

Elencare i pazienti con almeno un ricovero in ogni reparto

$$\{p.* \mid p(PAZIENTI) \mid \forall r(REPARTI)(\exists r'(RICOVERI)(r.COD=r'.Reparto \wedge p.COD=r'.PAZ))\}$$

Spiegazione: la variabile *p* prende, come valori, le tuple di *pazienti*. Se istanzio la variabile *p* su una ben precisa tupla, la devo restituire in output se è stato ricoverato **in ogni** reparto. Per ogni istanza di reparto **deve esistere** un ricovero di quel paziente nel reparto generico in *r*

Esempio

Elencare i pazienti con almeno un ricovero in ogni reparto

$$\{p.* \mid p(PAZIENTI) \mid \forall r(REPARTI)(\exists r'(RICOVERI)(r.COD=r'.Reparto \wedge p.COD=r'.PAZ))\}$$

Spiegazione: se un paziente non è stato ricoverato in un reparto, quando *r* è istanziato con la tupla corrispondente a quel reparto, la quantificazione esistenziale fallisce e di conseguenza fallisce anche la quantificazione universale (la condizione deve essere vera **per ogni** reparto)

Esempio con doppia negazione

Neghiamo due volte la quantificazione universale

$$\{p.* \mid p(PAZIENTI) \mid \neg\neg\forall r(REPARTI)(\exists r'(RICOVERI)(r.COD=r'.Reparto \wedge p.COD=r'.PAZ))\}$$

$$\{p.* \mid p(PAZIENTI) \mid \neg\exists r(REPARTI)(\neg\exists r'(RICOVERI)(r.COD=r'.Reparto \wedge p.COD=r'.PAZ))\}$$

In SQL l'interrogazione si esprime esattamente così!

Calcolo relazionale $\leftrightarrow$ SQL

La **target list** corrisponde alla **SELECT**

La **range list** corrisponde alla **FROM**

La **formula** corrisponde alla **WHERE**

Esempio

Elencare i pazienti ricoverati due o più volte

$$\{p.* \mid p(\mathit{PAZIENTI}) \mid \exists r'(\mathit{RICOVERI})(\exists r''(\mathit{RICOVERI})(p.\mathit{COD}=r'.\mathit{PAZ} \wedge p.\mathit{COD}=r''.\mathit{PAZ} \wedge r'.\mathit{Inizio} \neq r''.\mathit{Inizio}))\}$$

Esempio

Elencare i pazienti ricoverati una sola volta

equivalente a: elencare i pazienti ricoverati ma non ricoverati due o più volte

$$\{p.* \mid p(PAZIENTI) \mid \exists r(RICOVERI)\ (p.COD=r.PAZ \land \neg \exists r'(RICOVERI)\ (\exists r''(RICOVERI)(p.COD=r'.PAZ \land p.COD=r''.PAZ \land r'.Inizio \neq r''.Inizio)))\}$$

Pattern di soluzione

Elencare i pazienti ricoverati una sola volta

equivalente a: elencare i pazienti ricoverati ma non ricoverati due o più volte

$$\{p.* \mid p(PAZIENTI) \mid formula\_U \wedge \neg formula\_P\}$$

In algebra relazionale: $R = U - P$

U: pazienti ricoverati **P**: pazienti ricoverati due o più volte

Esempio

Elencare i medici non primari

$$\{m.* \mid m(MEDICI) \mid \neg\exists r(REPARTI)(r.Primario=m.MATR)\}$$

- In questo esempio, identico al precedente, pare manchi l'universo del discorso *U*
- In realtà la parte *U* è definita implicitamente dalla variabile libera *m*

Prodotto cartesiano

Si rende semplicemente in questo modo

{x., y.* | x(R),y(S) }*

Unione

L'unione insiemistica dell'algebra relazionale **non è esprimibile** col calcolo relazionale con dichiarazione di range

Esempio: elenco (cognome e nome) di tutti i pazienti e tutti medici

Limiti del calcolo relazionale

Il problema dell'inesprimibilità dell'unione insiemistica è una limitazione del calcolo relazionale su tuple con dichiarazione di range o degli approcci dichiarativi in generale?

- E' un problema specifico del calcolo relazionale con dichiarazione di range
- Il **calcolo relazionale su domini** non comporta questa limitazione

Calcolo relazionale sui domini

- Le variabili sono introdotte sui domini degli attributi
- Questo calcolo è universale, è cioè "completo" rispetto agli operatori dell'algebra relazionale
- Perché, allora, abbiamo introdotto il calcolo relazionale su tuple con dichiarazione di range?

Motivazioni

1. Il calcolo relazionale con dichiarazione di range ha ispirato direttamente l'SQL
   – { T | L | F } → `select… from… where…`
   – l'unione in SQL non è contenuta nel costrutto "`select… from… where`", ma è fuori

2. Per poter trattare il calcolo relazionale sui domini in modo adeguato richiede sviluppare un impianto logico che richiederebbe troppo tempo
   – formule safe

Esempio

Riconsideriamo questo esempio

S= studenti, E = esami, P = Piano di studi

S

MATR	Nome
1	Rossi
2	Verdi
3	Bianchi

E

MATR	Corso
2	Programmazione
3	Algebra
2	Basi di dati
3	Programmazione
2	Algebra

P

Corso
Programmazione
Basi di dati
Algebra

Elencare gli studenti che hanno sostenuto tutti gli esami

Esempio

Elencare gli studenti che hanno sostenuto tutti gli esami

$$\{\, s.* \mid s(S) \mid \forall p(P)(\exists e(E)(e.MATR = s.MATR \wedge e.Corso = p.Corso))\}$$

Variante all'esempio

Variamo leggermente l'esempio

S= studenti, E = esami, O = Offerta formativa

S

<u>MATR</u>	Nome	Indirizzo
1	Rossi	Reti
2	Verdi	Sistemi
3	Bianchi	Reti

E

MATR	Corso	Indirizzo
2	Programmazione	Sistemi
3	Algebra	Sistemi
2	Basi di dati	Sistemi
3	Programmazione	Reti
2	Algebra	Sistemi

O

Corso	Indirizzo
Programmazione	Sistemi
Basi di dati	Sistemi
Programmazione	Reti
Basi di dati	Reti
Algebra	Sistemi

Pattern di soluzione

Elencare gli studenti che hanno superato tutti gli esami del loro indirizzo

Devo introdurre l'implicazione $\alpha \Rightarrow \beta$, equivalente a $(\neg\alpha\vee\beta)$

$$\{s.* \mid s(S) \mid \forall o(O)(o.Indirizzo=s.Indirizzo \Rightarrow \exists e(E)(e.MATR=s.MATR \wedge e.Corso=o.Corso \wedge e.Indirizzo=o.Indirizzo))\}$$

Pattern di soluzione

Elencare gli studenti che hanno superato tutti gli esami del loro indirizzo

$$\{s.* \mid s(S) \mid \forall o(O)(o.Indirizzo=s.Indirizzo \Rightarrow \exists e(E)(e.MATR=s.MATR \wedge e.Corso=o.Corso \wedge e.Indirizzo=o.Indirizzo))\}$$

Spiegazione:

- con la quantificazione universale scorro tutti i corsi dell'offerta formativa
- con l'antecedente mi soffermo sui corsi dell'indirizzo dello studente
- con il conseguente verifico che lo studente abbia superato l'esame

Pattern di soluzione

Elencare gli studenti che hanno superato tutti gli esami del loro indirizzo

$$\{s.* \mid s(S) \mid \forall o(O)(o.Indirizzo=s.Indirizzo \Rightarrow \exists e(E)(e.MATR=s.MATR \wedge e.Corso=o.Corso \wedge e.Indirizzo=o.Indirizzo))\}$$

Spiegazione:

- Se scorro corsi che non sono dell'indirizzo dello studente, l'antecedente è falso e l'implicazione rimane quindi vera (il quantificatore universale non viene falsificato)

Esercizio

Abbiamo già provato a rispondere alla stessa interrogazione in algebra relazionale

Ripasso:

- L'interrogazione **non è** meccanicamente riconducibile al quoziente
- Il modello astratto a cui si ispira il quoziente è quello giusto

Esempio

IMPIEGATI

<u>MATR</u>	Cognome	Nome	Età	Stipendio
203	Neri	Piero	50	40
574	Bisi	Mario	60	60
461	Bargio	Sergio	30	61
530	Belli	Nicola	40	38
405	Mizzi	Nicola	55	60
501	Monti	Mario	25	35

ORGANIGRAMMA

Capo	Impiegato
203	405
203	501
574	203
574	530
405	461

IMPIEGATI(<u>MATR</u>, Cognome, Nome, Età, Stipendio)

ORGANIGRAMMA(<u>Capo,Impiegato</u>)

Esempio

Elencare i capi i cui subalterni guadagnano tutti più del capo

Possiamo usare la quantificazione universale e l'implicazione

$\{i.* \mid i(IMPIEGATI) \mid \forall o(ORGANIGRAMMA)(o.Capo=i.MATR \Rightarrow \exists i'(IMPIEGATI)(i'.MATR=o.Impiegato \wedge i'.Stipendio>i.Stipendio))\}$

Esempio

Elencare i capi i cui subalterni guadagnano tutti più del capo

In algebra relazionale:

$$\Pi_{MATR}(impiegati) - \Pi_{I1.Capo}(\sigma_{I2.Stipendio \leq I1.Stipendio}(\rho_{I1 \leftarrow impiegati}(impiegati) \bowtie_{I1.MATR=Capo} organigramma \bowtie_{I2.MATR=Impiegato} \rho_{I2 \leftarrow impiegati}(impiegati)))$$

Esempio

Autori (A), Pubblicazioni (P), Firma (F)

Con schemi:

A(COD,Cognome,Nome,...)

P(ID,Titolo,Anno)

F(A,P)

Esempio

Trovare gli autori che hanno pubblicato solo in collaborazione (autori che non hanno mai pubblicato una pubblicazione in cui sono gli unici firmatari)

$$\{a.* \mid a(A) \mid \forall f(F)(f.A=a.COD \Rightarrow \exists f'(F)(f'.P=f.P \wedge f'.A \neq a.COD))\}$$

Esempio

Trovare gli autori che hanno pubblicato solo in collaborazione (autori che non hanno mai pubblicato una pubblicazione in cui sono gli unici firmatari)

In algebra relazionale:

$$\Pi_A U - \Pi_A(U - P) =$$

$$\Pi_A(f) - \Pi_A(f - (\Pi_{f1.A,\, f1.P}(\rho_{F1 \leftarrow F}(f) \bowtie_{f1.P=f2.P \wedge f1.A \neq f2.A} \rho_{F2 \leftarrow F}(f))))$$

Esempio

Trovare gli autori che hanno pubblicato solo in collaborazione (autori che non hanno mai pubblicato una pubblicazione in cui sono gli unici firmatari)

In algebra relazionale:

$$\Pi_A U - \Pi_A(U - P) =$$

$$\Pi_A(f) - \Pi_A(f - (\boldsymbol{\Pi_{f1.A,\, f1.P}(\rho_{F1 \leftarrow F}(f) \bowtie_{f1.P=f2.P \wedge f1.A \neq f2.A} \rho_{F2 \leftarrow F}(f))}))$$

P: Elenco degli autori con pubblicazioni firmate in collaborazione

Esempio

Trovare gli autori che hanno pubblicato solo in collaborazione (autori che non hanno mai pubblicato una pubblicazione in cui sono gli unici firmatari)

In algebra relazionale:

$$\Pi_A U - \Pi_A(U - P) =$$

$$\Pi_A(f) - \Pi_A(\boldsymbol{f - (\Pi_{f1.A,\, f1.P}(\rho_{F1 \leftarrow F}(f) \bowtie_{f1.P=f2.P \wedge f1.A \neq f2.A} \rho_{F2 \leftarrow F}(f)))})$$

P: Elenco degli autori con pubblicazioni firmate in collaborazione

U – P: Elenco degli autori con pubblicazioni a firma **unica**

Esempio

Trovare gli autori che hanno pubblicato solo in collaborazione (autori che non hanno mai pubblicato una pubblicazione in cui sono gli unici firmatari)

In algebra relazionale:

$$\Pi_A U - \Pi_A(U - P) =$$

$$\Pi_A (f) - \boldsymbol{\Pi_A(f - (\Pi_{f1.A,\, f1.P}(\rho_{F1 \leftarrow F}(f) \bowtie_{f1.P=f2.P \wedge f1.A \neq f2.A} \rho_{F2 \leftarrow F}(f))))}$$

P: Elenco degli autori con pubblicazioni firmate in collaborazione

U – P: Elenco degli autori con pubblicazioni a firma **unica**

$\Pi_A(U - P)$: Elenco degli autori che hanno pubblicato **anche** da soli

Esempio

Trovare gli autori che hanno pubblicato solo in collaborazione (autori che non hanno mai pubblicato una pubblicazione in cui sono gli unici firmatari)

In algebra relazionale:

$$\Pi_A U - \Pi_A(U - P) =$$

$$\boldsymbol{\Pi_A (f) - \Pi_A(f - (\Pi_{f1.A,\, f1.P}(\rho_{F1 \leftarrow F}(f) \bowtie_{f1.P=f2.P \wedge f1.A \neq f2.A} \rho_{F2 \leftarrow F}(f))))}$$

P: Elenco degli autori con pubblicazioni firmate in collaborazione

$U - P$: Elenco degli autori con pubblicazioni a firma **unica**

$\Pi_A(U - P)$: Elenco degli autori che hanno pubblicato **anche** da soli

$U - \Pi_A(U - P)$: Elenco degli autori che hanno pubblicato **solo** in collaborazione

Considerazione finale

Con i formalismi dell'algebra relazionale e del calcolo relazionale, sono in grado di esprimere tutte le interrogazioni potenzialmente immaginabili?

Considerazione finale

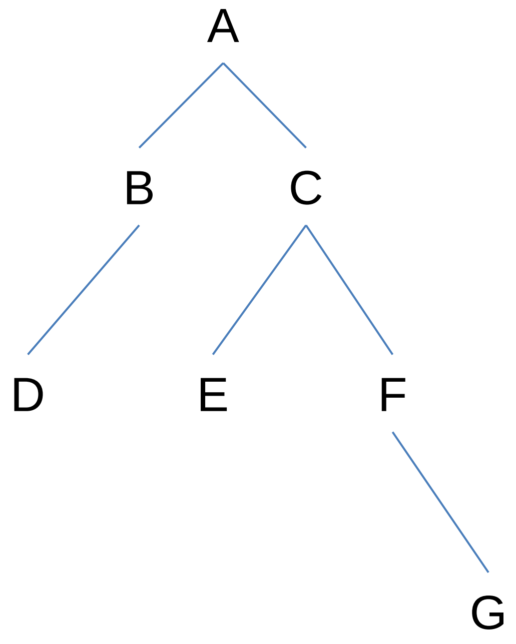
Con i formalismi dell'algebra relazionale e del calcolo relazionale, sono in grado di esprimere tutte le interrogazioni potenzialmente immaginabili?

No, perché manca la ricorsione!

Considerazione finale

Esempio: albero genealogico AG(G,F)

G	F
A	C
A	B
B	D
C	E
C	F
F	G



Considerazione finale

Relazione **nonno-nipote** facile da catturare:

$$\Pi_{AG1.P,AG2.F}(\rho_{AG1 \leftarrow AG}(ag) \bowtie_{ag1.F=ag2.P} \rho_{AG2 \leftarrow AG}(ag))$$

- Con un altro join possiamo trovare i trisavoli
- In generale possiamo trovare un grado di parentela profondo a piacere **purché finito**!
- Le espressioni algebriche hanno quindi un numero finito di join

Considerazione finale

Estrarre una tavola discendenti di questo tipo

discendente(Persona1,Persona2)

in cui voglio tutti i rapporti di discendenza, ovvero:

G	F
A	C
A	B
B	D
C	E
C	F
F	G

Persona1	Persona2
A	C
A	B
A	D
A	E
A	F
A	G
B	D
C	E
C	F
C	G
F	G

Considerazione finale

Estrarre una tavola discendenti di questo tipo

discendente(Persona1,Persona2)

in cui voglio tutti i rapporti di discendenza

- Questa interrogazione non è esprimibile in algebra/calcolo relazionale in quanto richiede un impianto ricorsivo
- Si tratta di una chiusura transitiva

Considerazione finale

Estrarre una tavola discendenti di questo tipo

discendente(Persona1,Persona2)

in cui voglio tutti i rapporti di discendenza

- Si risolve nell'ambito delle **basi dati deduttive**
- Le ultime versioni di SQL implementano una forma di ricorsione (attraverso le chiusure transitive)

Considerazione finale

Estrarre una tavola discendenti di questo tipo

discendente(Persona1,Persona2)

in cui voglio tutti i rapporti di discendenza

- Rimane, anche nelle dati basi deduttive, il problema della negazione