

Basi di Dati
Algebra relazionale
- seconda parte -
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

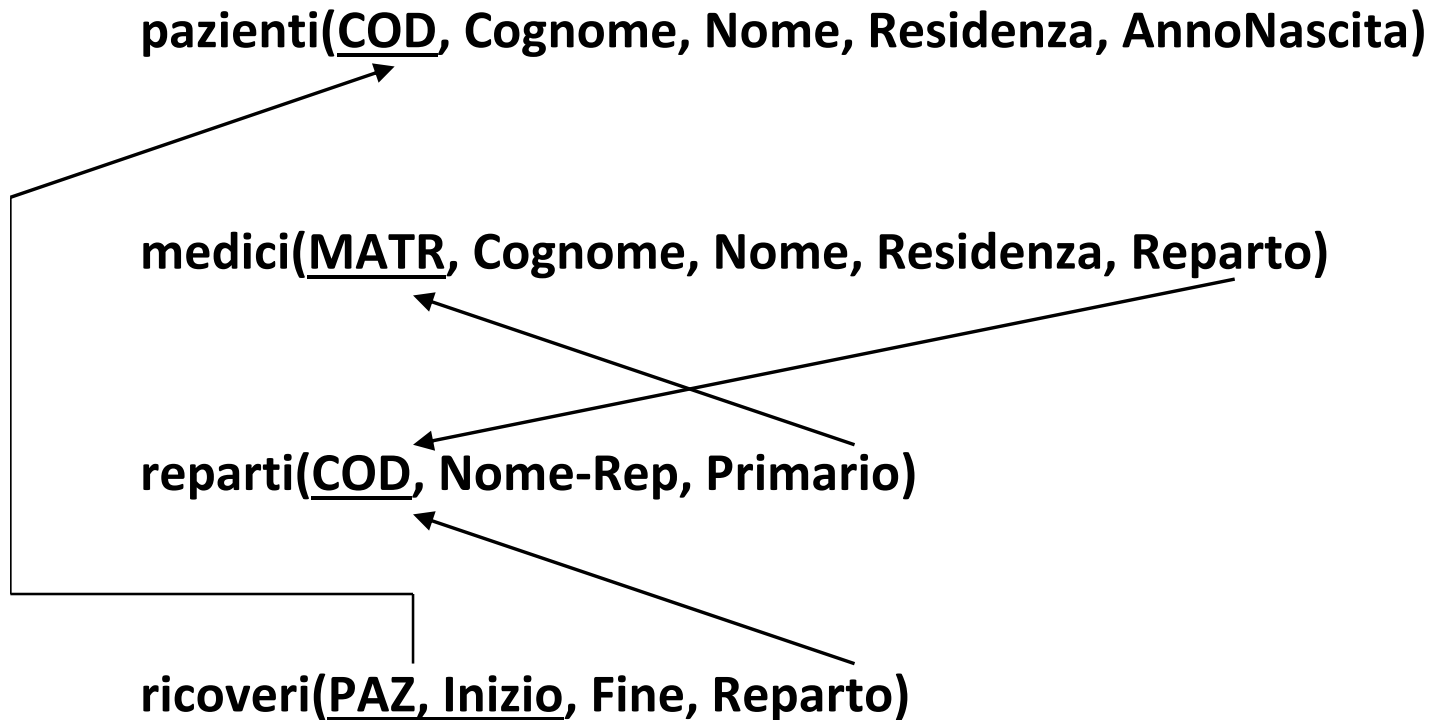
<u>PAZ</u>	Inizio	Fine	Reparto
A102	2/05/2014	9/05/2014	A
A102	2/12/2004	2/01/2005	A
S555	5/10/2014	3/12/2014	B
B444	1/12/2004	2/01/2005	B
S555	6/09/2015	1/11/2015	A

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:



Base di Dati "Impiegati"

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(MATR, Cognome, Nome, Età, Stipendio)

ORGANIGRAMMA(Capo, Impiegato)



Interrogazioni con negazione

Vediamo i seguenti esempi

- Esempio 1: elencare i pazienti **non residenti** in Torino
- Esempio 2: elencare i medici **non primari**

Interrogazioni con negazione

Esempio 1: elencare i pazienti **non residenti** in Torino

Si tratta di un'interrogazione con negazione non essenziale (inessenziale)

- Il contesto del sistema informativo mi suggerisce che è possibile risolvere l'interrogazione con un'interrogazione affermativa, ad esempio:

Elencare i pazienti con residenza **diversa** da Torino

Negazione non essenziale

"elencare i pazienti **non residenti** in Torino"

=

"elencare i pazienti con residenza **diversa** da Torino"

$\sigma_{\text{Residenza} \neq \text{'Torino'}}(\text{pazienti})$

Interrogazioni con negazione

Esempio 2: elencare i medici **non primari**

Non riesco a scrivere un'espressione algebrica con appropriati confronti che risolva quest'interrogazione

Si tratta di un'interrogazione con negazione essenziale

Negazione essenziale

Elencare i medici **non primari**

Esiste **un solo modo** per risolvere un'interrogazione essenziale.

Premessa

Nelle basi di dati si lavora con la **logica a mondo chiuso**, ovvero quello che è scritto nella base di dati è tutto ciò di cui disponiamo, e tutto va descritto all'interno di tale istanza di base di dati

La risposta alle interrogazioni va cercata all'interno delle sole informazioni presenti nella base di dati

Posso cercare i medici non primari **solo all'interno della relazione *medici***, non al di fuori

Negazione essenziale

Elencare i medici **non primari**

Schema generale di soluzione:

1. Si definisce l'universo del discorso U
 - Nel nostro esempio: $U = \text{medici}$

Negazione essenziale

Elencare i medici **non primari**

Schema generale di soluzione:

1. Si definisce l'universo del discorso U
 - Nel nostro esempio: $U = \text{medici}$
2. Rispondere alla domanda in forma positiva P
 - Nel nostro esempio: *elencare i medici primari*

$$P = \prod_{MATR, Cognome, Nome, Residenza, Reparto} (\text{medici} \bowtie_{MATR=Primario} \text{reparti})$$

Negazione essenziale

Elencare i medici **non primari**

Schema generale di soluzione:

1. Si definisce l'universo del discorso U
 - Nel nostro esempio: $U = \text{medici}$
2. Rispondere alla domanda in forma positiva P
 - Nel nostro esempio: *elencare i medici primari*

$$P = \prod_{MATR, Cognome, Nome, Residenza, Reparto} (\text{medici} \bowtie_{MATR=Primario} \text{reparti})$$

3. Trovo la risposta all'interrogazione $R = U - P$
 - Nel nostro esempio:

$$\text{medici} - \prod_{MATR, Cognome, Nome, Residenza, Reparto} (\text{medici} \bowtie_{MATR=Primario} \text{reparti})$$

Requisito

Gli schemi di U e P devono essere **uguali** (e ovviamente compatibili)

Alternativa

1. $U = \Pi_{MATR}(medici)$
2. $P = \Pi_{MATR}(medici \bowtie_{MATR=Primario} reparti)$
3. $R = U - P =$
 $\Pi_{MATR}(medici) - \Pi_{MATR}(medici \bowtie_{MATR=Primario} reparti)$
4. $R \bowtie medici$
 - per ottenere le informazioni dei medici

Esempio

Elencare i pazienti **non residenti** in città in cui risiede qualche medico

Esempio

Elencare i pazienti **non residenti** in città in cui risiede qualche medico

1. $U = \text{pazienti}$
2. $P = \prod_{\text{COD}, \text{pazienti.Cognome}, \text{pazienti.Nome}, \text{pazienti.Residenza}, \text{AnnoNascita}} (\text{pazienti} \bowtie_{\text{pazienti.Residenza} = \text{medici.Residenza}} \text{medici})$
3. $R = U - P =$
 $\text{pazienti} - \prod_{\text{COD}, \text{pazienti.Cognome}, \text{pazienti.Nome}, \text{pazienti.Residenza}, \text{AnnoNascita}} (\text{pazienti} \bowtie_{\text{pazienti.Residenza} = \text{medici.Residenza}} \text{medici})$

Esercizio

Elencare i reparti in cui non avvengono ricoveri

Generalità dell'approccio

Esempio 1: elencare i pazienti **non residenti** in Torino

Posso comunque usare la procedura studiata:

1. $U = \text{pazienti}$
2. $P = \sigma_{\text{Residenza}='Torino'}(\text{pazienti})$
3. $R = U - P = \text{pazienti} - \sigma_{\text{Residenza}='Torino'}(\text{pazienti})$

E' però una risposta ridondante!

Esempio più complesso

Elencare i pazienti di Torino mai curati dal primario 203

Esempio più complesso

Elencare i pazienti di Torino mai curati dal primario 203

1. $U = \sigma_{Residenza='Torino'}(pazienti)$

Esempio più complesso

Elencare i pazienti di Torino mai curati dal primario 203

1. $U = \sigma_{Residenza='Torino'}(pazienti)$
2. $P = \Pi_{pazienti.COD, Cognome, Nome, Residenza, AnnoNascita}$
 $(pazienti \bowtie_{COD=PAZ} ricoveri \bowtie_{Reparto=reparti.COD} \sigma_{Primario='203'}(reparti))$

Esempio più complesso

Elencare i pazienti di Torino mai curati dal primario 203

1. $U = \sigma_{Residenza='Torino'}(pazienti)$
2. $P = \Pi_{pazienti.COD, Cognome, Nome, Residenza, AnnoNascita}$
 $(pazienti \bowtie_{COD=PAZ} ricoveri \bowtie_{Reparto=reparti.COD} \sigma_{Primario='203'}(reparti))$
3. $R = U - P = \sigma_{Residenza='Torino'}(pazienti) -$
 $\Pi_{pazienti.COD, Cognome, Nome, Residenza, AnnoNascita}$
 $(pazienti \bowtie_{COD=PAZ} ricoveri \bowtie_{Reparto=reparti.COD} \sigma_{Primario='203'}(reparti))$

Ambiguità nelle interrogazioni

"Elencare i pazienti di Torino mai curati dal primario 203"

può essere letta come:

"Elencare i pazienti di Torino ricoverati ma mai presi in cura dal primario 203"

Cambia l'universo del discorso

$$U = \prod_{\text{pazienti.COD,Cognome,Nome,Residenza,AnnoNascita}} \\ (\text{pazienti} \bowtie_{\text{COD=PAZ}} \text{ricoveri})$$

Esempio (negazione nascosta)

Elencare i pazienti con un solo ricovero

Esempio (negazione nascosta)

"Elencare i pazienti con un solo ricovero"

equivalente a:

"Elencare i pazienti ricoverati almeno una volta ma che non hanno subito due o più ricoveri"

Esempio (negazione nascosta)

Elencare i pazienti ricoverati almeno una volta ma che non hanno subito due o più ricoveri

1. $U = \prod_{PAZ}(ricoveri)$
2. $P = \prod_{ricoveri1.PAZ}(\rho_{RICOVERI1 \leftarrow RICOVERI}(ricoveri)$
 $\bowtie_{ricoveri1.PAZ=ricoveri2.PAZ \wedge ricoveri1.Inizio \neq ricoveri2.Inizio}$
 $\rho_{RICOVERI2 \leftarrow RICOVERI}(ricoveri))$
– *interrogazione già vista*
3. $R = U - P = \dots$

Esempio (negazione nascosta)

Elencare i pazienti con un solo ricovero

$$\begin{aligned} & \Pi_{PAZ}(ricoveri) - \Pi_{ricoveri1.PAZ}(\rho_{RICOVERI1 \leftarrow RICOVERI}(ricoveri)) \\ & \bowtie_{ricoveri1.PAZ=ricoveri2.PAZ \wedge ricoveri1.Inizio \neq ricoveri2.Inizio} \\ & \rho_{RICOVERI2 \leftarrow RICOVERI}(ricoveri)) \end{aligned}$$

Uso "creativo" dell'algebra relazionale

Consideriamo una relazione $r(A)$ composta da un unico attributo di tipo intero *NUM*

<u>NUM</u>
3
5
2

Pur non disponendo degli operatori aritmetici, in algebra relazionale è possibile estrarre il massimo!

Uso "creativo" dell'algebra relazionale

Qui la forma con negazione è ancora più nascosta

Possiamo però ricordare che il massimo di una sequenza di numeri è tale se il numero è maggiore uguale di tutti gli altri

Purtroppo non posso esprimere direttamente questa definizione in algebra relazionale

Uso "creativo" dell'algebra relazionale

So però decidere se un numero risulta essere minore di qualcun altro

Calcolo il prodotto cartesiano:

$$\rho_{\text{NUM1} \leftarrow \text{NUM}}(r) \times \rho_{\text{NUM2} \leftarrow \text{NUM}}(r)$$

NUM1	NUM2
3	3
3	5
3	2
5	3
5	5
5	2
2	3
2	5
2	2

Uso "creativo" dell'algebra relazionale

So però decidere se un numero risulta essere minore di qualcun altro

Calcolo il prodotto cartesiano:

$$\rho_{NUM1 \leftarrow NUM}(r) \times \rho_{NUM2 \leftarrow NUM}(r)$$

NUM1	NUM2
3	3
3	5
3	2
5	3
5	5
5	2
2	3
2	5
2	2

Uso "creativo" dell'algebra relazionale

So però decidere se un numero risulta essere minore di qualcun altro

Seleziono le tuple per cui $NUM1 < NUM2$:

$$\sigma_{NUM1 < NUM2}(\rho_{NUM1 \leftarrow NUM}(r) \times \rho_{NUM2 \leftarrow NUM}(r))$$

NUM1	NUM2
3	5
2	3
2	5

Cos'è la proiezione su NUM1?

$$\Pi_{NUM1}(\sigma_{NUM1 < NUM2}(\rho_{NUM1 \leftarrow NUM}(r) \times \rho_{NUM2 \leftarrow NUM}(r)))$$

Uso "creativo" dell'algebra relazionale

Cos'è la proiezione su NUM1?

$$\Pi_{NUM1} (\sigma_{NUM1 < NUM2} (\rho_{NUM1 \leftarrow NUM}(r) \times \rho_{NUM2 \leftarrow NUM}(r)))$$

E' l'elenco dei numeri (2 e 3) ognuno dei quali risulta essere minore di **qualche** altro numero

NUM1
3
2

Uso "creativo" dell'algebra relazionale

Quindi se chiamo

- $U = \rho_{NUM1 \leftarrow NUM}(r)$
- $P = \prod_{NUM1} (\sigma_{NUM1 < NUM2} (\rho_{NUM1 \leftarrow NUM}(r) \times \rho_{NUM2 \leftarrow NUM}(r)))$

$R = U - P$ è l'elenco dei numeri **che non sono minori** di qualche altro numero, cioè dei numeri maggiori o uguali a tutti gli altri!

NUM1
5

Interrogazioni con quantificazione universale

Tutti gli esempi visti finora, tranne l'ultimo, ricadono nella casistica della **quantificazione esistenziale**

- **Esiste** un paziente ricoverato nel reparto di chirurgia
- **Esiste** un medico che è anche primario del reparto di chirurgia

Invece l'ultimo esempio ("estrarre il massimo di un insieme di numeri"), equivalente a "estrarre il numero maggiore o uguale **di tutti** i numeri" è un esempio di **quantificazione universale**!

Quantificazione universale

Quando troviamo "**tutti**" "**ogni**" "**sempre**" nell'interrogazione, siamo di fronte a casi di quantificazione universale.

Quantificazione universale

Possiamo ricondurci alla quantificazione esistenziale ricordandoci che la quantificazione universale è la negazione di una quantificazione esistenziale

Esempio

Consideriamo questo esempio

E = esami, P = Piano di studi

E

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

P

<u>Corso</u>
Programmazione
Basi di dati
Algebra

Elencare le matricole abbinate a tutti i corsi previsti dal piano di studi

Procedimento di soluzione

1. Individuo l'universo di riferimento

$$U = \Pi_{MATR}(e)$$

MATR
2
3

Procedimento di soluzione

1. Individuo l'universo di riferimento

$$U = \Pi_{MATR}(e)$$

2. Analizzo tutte le combinazioni possibili di matricola con i corsi previsti:

$$\Pi_{MATR}(e) \times p$$

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

Procedimento di soluzione

2. Analizzo tutte le combinazioni possibili di matricola con i corsi previsti:

$$\Pi_{MATR}(e) \times p$$

3. Trovo la differenza con e (hanno lo stesso schema)

$$(\Pi_{MATR}(e) \times p) - e$$

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

—

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

=

MATR	Corso
3	Basi di dati

Procedimento di soluzione

3. Trovo la differenza con e (hanno lo stesso schema)

$$(\Pi_{MATR}(e) \times p) - e$$

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

—

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

=

MATR	Corso
3	Basi di dati

Ho trovato l'elenco dei corsi non superati

4. Proietto su $MATR$

MATR
3

$$\Pi_{MATR}((\Pi_{MATR}(e) \times p) - e)$$

Procedimento di soluzione

4. Proietto su *MATR*

$$\Pi_{MATR}((\Pi_{MATR}(e) \times p) - e)$$

MATR
3

Ho ottenuto l'elenco delle matricole che non hanno ancora superato **qualche** corso

Ora riconsidero l'universo di partenza $U = \Pi_{MATR}(e)$

Procedimento di soluzione

5. Sottraggo il risultato all'universo di partenza

$$\Pi_{MATR}(e) - \Pi_{MATR}((\Pi_{MATR}(e) \times p) - e)$$

MATR
2

Ho finalmente l'elenco di studenti che hanno superato **tutti** i corsi!

Non è altro che la definizione dell'**operatore quoziente**

Operatore derivato quoziente

Dati due insiemi disgiunti di attributi A e B , l'operatore derivato quoziente si definisce come:

$$r(A,B) \div s(B) = \Pi_A(r) - \Pi_A((\Pi_A(r) \times s) - r)$$

La cardinalità del quoziente è

$$|r(A,B) \div s(B)| \leq |\Pi_A(r)| \leq |r|$$

Esempio

Elencare i pazienti che hanno subito ricovero **in ogni** reparto

Esempio

Elencare i pazienti che hanno subito ricovero **in ogni** reparto

Applichiamo il quoziente:

$$r(A,B) \div s(B)$$

- $r(A,B) = \prod_{PAZ, Reparto}(ricoveri)$
- $S(B) = \prod_{Reparto}(ricoveri)$

Esempio

Elencare i pazienti che hanno subito ricovero **in ogni** reparto

Applichiamo il quoziente:

$$\Pi_{PAZ,Reparto}(ricoveri) \div \Pi_{Reparto}(ricoveri)$$

Esempio

Elencare i pazienti che hanno subito ricovero **in ogni** reparto

Espandiamo il quoziente:

$$r(A,B) \div s(B) = \prod_A(r) - \prod_A ((\prod_A(r) \times s) - r)$$

$$\begin{aligned} & \prod_{PAZ,Reparto}(ricoveri) \div \prod_{Reparto}(ricoveri) = \\ & \quad \prod_{PAZ}(\prod_{PAZ,Reparto}(ricoveri)) - \\ & \quad \prod_{PAZ} ((\prod_{PAZ}(\prod_{PAZ,Reparto}(ricoveri)) \times \prod_{Reparto}(ricoveri)) - \\ & \quad \quad \prod_{PAZ,Reparto}(ricoveri)) \end{aligned}$$

Esercizio

Se avessi scelto come tavola s la relazione

$$\rho_{\text{Reparto} \leftarrow \text{COD}} \Pi_{\text{COD}}(\text{reparti})$$

nell'ipotesi che ci siano reparti non adibiti a ricoveri,
quale sarebbe stata la risposta del quoziente?

E' equivalente alla prima formulazione?

Semantica di Codd del valore nullo

Abbiamo fin qui ipotizzato l'assenza di valori nulli

Un valore nullo (**non specificato**) in realtà nella relazione c'è un valore vero e proprio, appartenente ad ogni dominio: il valore nullo

Significati:

- Informazione esistente, ma mancante
- Informazione inesistente
- Assenza di informazione

Semantica di Codd del valore nullo

Le interpretazioni date del valore nullo sono tutte esterne, non interna al **modello relazionale** che non distingue i tre casi

Semantica di Codd del valore nullo

La semantica di Codd del valore nullo è il comportamento formale di una interrogazione a fronte di tuple che contengono valori nulli

Ci riferiamo in particolare alla selezione σ_p con predicato p che è una combinazione di confronti del tipo:

- $A_i \theta \text{costante}$
- $A_i \theta A_j$

Semantica di Codd del valore nullo

$A_i \theta \text{ costante}$

Che valore di verità assume il confronto quando una tupla ha valore nullo in corrispondenza dell'attributo oggetto del confronto?

Quando il valore della tupla, in corrispondenza dell'attributo A_i è nullo, il valore di verità del confronto di un attributo con la costante è **sconosciuto**

Logica a tre valori

Codd propone di collocarsi in una logica a tre valori: *Falso (F)*, *Sconosciuto (U)*, *Vero (T)*

Quindi se $t[A_i]$ è nullo, il valore di verità del confronto

$t[A_i] \theta \text{ costante}$

sarà sconosciuto (*U*)

Logica a tre valori

Caso del confronto tra attributi $A_i \theta A_j$

- Ad esempio, join tra due tabelle

Ci sono diversi casi possibili:

1. $v_i \theta v_j$ (*assenza di valori nulli*)
2. $v_i \theta \text{valore nullo}$
3. $\text{valore nullo} \theta v_j$
4. $\text{valore nullo} \theta \text{valore nullo}$

Logica a tre valori

Caso del confronto tra attributi $A_i \theta A_j$

1. $v_i \theta v_j$ (*assenza di valori nulli*)
2. $v_i \theta$ *valore nullo*
3. *valore nullo* θv_j
4. *valore nullo* θ *valore nullo*

Codd decide di assegnare **sempre**, negli ultimi tre casi, il valore di verità *Sconosciuto* (U)

Esempio

Trovare le coppie di clienti residenti nella stessa città

CLIENTI

COD	Residenza
100	TO
200	NULL
300	TO
400	NULL

Esempio

Trovare le coppie di clienti residenti nella stessa città

CLIENTI	
COD	Residenza
100	TO
200	NULL
300	TO
400	NULL

$$\rho_{CLIENTI1} \leftarrow_{CLIENTI}(clienti)$$

$$\bowtie_{clienti1.COD \neq clienti2.COD \wedge clienti1.Residenza = clienti2.Residenza}$$

$$\rho_{CLIENTI2} \leftarrow_{CLIENTI}(clienti)$$

Esempio

Trovare le coppie di clienti residenti nella stessa città

COD	Residenza	COD	Residenza
100	TO	200	NULL
100	TO	300	TO
100	TO	400	NULL
200	NULL	100	TO
200	NULL	300	TO
200	NULL	400	NULL
300	TO	100	TO
300	TO	200	NULL
300	TO	400	NULL
400	NULL	100	TO
400	NULL	200	NULL
400	NULL	300	TO

Esempio

Trovare le coppie di clienti residenti nella stessa città

COD	Residenza	COD	Residenza
100	TO	200	NULL
100	TO	300	TO
100	TO	400	NULL
200	NULL	100	TO
200	NULL	300	TO
200	NULL	400	NULL
300	TO	100	TO
300	TO	200	NULL
300	TO	400	NULL
400	NULL	100	TO
400	NULL	200	NULL
400	NULL	300	TO

Non posso concludere che due clienti che non hanno dichiarato la città, sono residenti nella stessa città!

Valori nulli ed espressioni algebriche

Le espressioni algebriche contengono spesso congiunzioni in AND, OR e NOT di predicati semplici

Dobbiamo estendere le tabelle di verità dei tre operatori

Operatori e logica a tre valori

AND	F	U	T
F	F	F	F
U	F	U	U
T	F	U	T

OR	F	U	T
F	F	U	T
U	U	U	T
T	T	T	T

NOT	
F	T
U	U
T	F

Se consideriamo $F=0$, $U=1$ e $T=2$, l'AND si calcola facilmente con il **minimo**, l'OR con il **massimo**

Espressioni e logica a tre valori

Date le tabelle di verità a tre valori, ogni tupla presa in esame genera un valore di verità dell'intera espressione che può essere F, T o U

Ma quando eseguo una selezione e trovo una tupla con valore di verità **sconosciuto**, seleziono o no la tupla?

Funzione di collasso

Considero una tupla t , presa in esame da un predicato p di una selezione/join che può rispondere F , U o T

Per sapere se la tupla t è selezionata o meno, useremo la **funzione di collasso** del predicato $p(t)$

$p(t)$	$c(p(t))$
F	F
U	F
T	T

Seleziono la tupla solo se la funzione collasso è TRUE

Ricerca di valori nulli

Tutte le volte che una tupla presenta valore nullo, in corrispondenza della selezione dà sempre valore di verità sconosciuto

Per poter cercare una tupla che ha (o non ha) un valore nullo in corrispondenza dell'attributo A_i , scrivo un predicato con le clausole

- A_i *IS NULL* (se mi interessa il valore nullo)
- A_i *IS NOT NULL* (se mi interessano i valori non nulli)

Esempi

- $\sigma_{\text{Residenza IS NULL}}(\text{clienti})$

COD	Residenza
200	NULL
400	NULL

- $\sigma_{\text{Residenza IS NOT NULL}}(\text{clienti})$

COD	Residenza
100	TO
300	TO

Limiti della logica dei tre valori

La logica a tre valori viola il principio di **non contraddizione** per cui vale

$$(p \wedge \neg p) = F$$

Ma se p è U , abbiamo $(p \wedge \neg p) = U$

La logica a tre valori viola il principio del **terzo escluso**

$$(p \vee \neg p) = T$$

Ma se p è U , abbiamo $(p \vee \neg p) = U$

Esempio

Ipotizziamo che *pazienti* ammetta valore nullo in corrispondenza dell'attributo *AnnoNascita*, la selezione

$$\sigma_{AnnoNascita < 1980 \vee AnnoNascita \geq 1980} (pazienti)$$

contiene un predicato della forma $(p \vee \neg p)$, ma la funzione collasso dà *Falso*, quindi:

$$|\sigma_{AnnoNascita < 1980 \vee AnnoNascita > 1980} (pazienti)| \leq |pazienti|$$

Esempio

L'interrogazione

$$\sigma_{AnnoNascita < 1980 \vee AnnoNascita \geq 1980} (pazienti)$$

è equivalente

$$\sigma_{AnnoNascita \text{ IS NOT NULL}} (pazienti)$$

Perché la logica a tre valori?

Immaginiamo un confronto $\neg(A_i = \text{costante})$

- Quando arriva una tupla che ha valore nullo in corrispondenza di A_i , l'uguaglianza con la costante genera sconosciuto
- La funzione NOT applicata ad A_i , genera sconosciuto
- La funzione collasso non fa selezionare la tupla

Se avessi dato il valore F al confronto di un valore nullo con costante, **la negazione avrebbe dato T!**

Esempio: agenzie

AGENZIE

CF	Nome	Città	Numero	Capitale
OD	Odo	TO	1	10
PN	Ponto	TO	2	15
MN	Mungo	AT	1	20
GY	Gilly	NO	1	25

CLIENTI

Indirizzo	CF	Nome
TO	LN	Longo
AT	FS	Fosco
NO	OD	Odo
TO	CM	Camelia
VC	PN	Ponto

Join esterni (outer join)

Distinguiamo tre tipi di outer join

- Left join
- Right join
- Full join

L'outer-join è un theta-join qualsiasi, ma per gli esempi useremo il natural-join

Left join: \bowtie_{θ}

Left join tra *clienti* e *agenzie*:

clienti \bowtie *agenzie*

- Vengono prese in considerazione tutte le tuple della tavola *clienti*, nessuna esclusa
- Si verifica se le tuple della tabella *clienti* sono in join con tuple della tabella *agenzie*
- Qualora una tupla di *clienti* non faccia join con nessuna tupla della tavola *agenzia*, si inseriscono valori nulli in corrispondenza degli attributi della seconda tavola

Left join

Left join tra *clienti* e *agenzie*

clienti ⋈ *agenzie*

Indirizzo	CF	Nome	Città	Numero	Capitale
TO	LN	Longo	NULL	NULL	NULL
AT	FS	Fosco	NULL	NULL	NULL
NO	OD	Odo	TO	1	10
TO	CM	Camelia	NULL	NULL	NULL
VC	PN	Ponto	TO	2	15

AGENZIE

CF	Nome	Città	Numero	Capitale
OD	Odo	TO	1	10
PN	Ponto	TO	2	15
MN	Mungo	AT	1	20
GY	Gilly	NO	1	25

Right join: \bowtie_{θ}

Right join tra *clienti* e *agenzie*:

clienti \bowtie *agenzie*

- Vengono prese in considerazione tutte le tuple della tavola *agenzie*, nessuna esclusa
- Si verifica se le tuple della tabella *agenzie* sono in join con tuple della tabella *clienti*
- Qualora una tupla di *agenzie* non faccia join con nessuna tupla della tavola *clienti*, si inseriscono valori nulli in corrispondenza degli attributi della prima tavola

Right join

Right join tra *clienti* e *agenzie*

clienti ⋈ *agenzie*

Indirizzo	CF	Nome	Città	Numero	Capitale
NO	OD	Odo	TO	1	10
VC	PN	Ponto	TO	2	15
NULL	MN	Mungo	AT	1	20
NULL	GY	Gilly	NO	1	25

CLIENTI

Indirizzo	CF	Nome
TO	LN	Longo
AT	FS	Fosco
NO	OD	Odo
TO	CM	Camelia
VC	PN	Ponto

Full join: \bowtie

Il full join è l'unione del left join e del right join

clienti \bowtie agenzie

equivale a

(clienti \bowtie agenzie) \cup (clienti \bowtie agenzie)

Full join

Right-join tra *clienti* e *agenzie*

clienti ~~⋈~~ *agenzie*

Indirizzo	CF	Nome	Città	Numero	Capitale
TO	LN	Longo	NULL	NULL	NULL
AT	FS	Fosco	NULL	NULL	NULL
NO	OD	Odo	TO	1	10
TO	CM	Camelia	NULL	NULL	NULL
VC	PN	Ponto	TO	2	15
NULL	MN	Mungo	AT	1	20
NULL	GY	Gilly	NO	1	25

Utilità dell'outer join

Sono dei join che mettono ben in evidenza alcune informazioni (esempio, visualizzare contemporaneamente clienti che sono anche direttori e clienti che non lo sono)

Proprietà degli operatori algebrici

Obiettivi:

- Conoscere bene le proprietà permette di districarsi nelle espressioni algebriche complesse per manipolarle (ad esempio per semplificarle, per sapere se due espressioni sono equivalenti, ecc.)
- E' molto importante conoscere le proprietà perché sono effettivamente utilizzate dal DBMS per ottimizzare le interrogazioni

Gruppi di proprietà

- Proprietà commutative
 - del prodotto cartesiano
 - del theta-join
- Proprietà associative
 - del prodotto cartesiano
 - del theta-join
- Proprietà della selezione multipla
- Proprietà della sostituzione di operatori

Gruppi di proprietà

- Proprietà distributive della selezione (!)
 - rispetto alla proiezione
 - rispetto al prodotto cartesiano
 - rispetto al join
 - rispetto a unione
 - rispetto alla differenza
 - rispetto all'intersezione
- Proprietà della proiezione multipla
- Proprietà distributive della proiezione
 - rispetto al prodotto cartesiano
 - rispetto al join
 - rispetto all'unione

Proprietà commutativa del prodotto cartesiano

Ricordandoci che

- lo schema risultato del prodotto cartesiano è l'unione dello schema
- l'ordine degli attributi in una relazione di Codd è irrilevante,

possiamo concludere che:

$$r(A) \times s(B) = s(B) \times r(A)$$

Proprietà commutativa del Θ -join

Ricordandoci che

$$r(A) \bowtie_{\Theta} s(B) = \sigma_{\Theta}(r(A) \times s(B))$$

e che, come abbiamo dimostrato, vale la commutatività del prodotto cartesiano, possiamo scrivere

$$\sigma_{\Theta}(r(A) \times s(B)) = \sigma_{\Theta}(s(B) \times r(A))$$

quindi

$$\sigma_{\Theta}(s(B) \times r(A)) = s(B) \bowtie_{\Theta} r(A)$$

cioè

$$r(A) \bowtie_{\Theta} s(B) = s(B) \bowtie_{\Theta} r(A)$$

Proprietà associativa del prodotto cartesiano

$$(r(A) \times s(B)) \times u(C) = r(A) \times (s(B) \times u(C))$$

- lo schema di $r(A) \times s(B)$ è $A \cup B$
- lo schema di $(r(A) \times s(B)) \times u(C)$ è $(A \cup B) \cup C$
- lo schema di $(A \cup B) \cup C$ è uguale a $A \cup (B \cup C)$

di conseguenza i due schemi coincidono, quindi:

$$(r(A) \times s(B)) \times u(C) = r(A) \times (s(B) \times u(C))$$

Proprietà associativa del Θ -join

$$(r(A) \bowtie_{\Theta(A,B)} s(B)) \bowtie_{\Theta(A \cup B, C)} u(C)$$

dove

- $\Theta(A,B)$ = congiunzione di confronti $A_i \varphi B_j$
- $\Theta(A \cup B, C) = ?$

specifichiamo meglio i Θ , raggruppando i confronti

- $\Theta(A \cup B, C) = \Theta(A, C) \wedge \Theta(B, C)$

con

- $\Theta(A, C)$ = congiunzione di confronti $A_i \varphi C_k$
- $\Theta(B, C)$ = congiunzione di confronti $B_j \varphi C_k$

Proprietà associativa del Θ -join

$$\begin{aligned} (r(A) \bowtie_{\Theta(A,B)} s(B)) \bowtie_{\Theta(A \cup B, C)} u(C) \\ = \\ r(A) \bowtie_{\Theta(A, B \cup C)} (s(B) \bowtie_{\Theta(B, C)} u(C)) \end{aligned}$$

dove

- $\Theta(A, B \cup C) = \Theta(A, B) \wedge \Theta(A, C)$

Quindi ritroviamo le stesse componenti di confronto $\Theta(A, B)$, $\Theta(A, C)$, $\Theta(B, C)$, e si applica la proprietà associativa del prodotto cartesiano, con le giuste selezioni

Conseguenza della proprietà associativa

Non è ambiguo scrivere

$$r(A) \times s(B) \times u(C)$$

senza parentesizzazione

Mentre, nel caso del join la parentesizzazione è necessaria a leggere correttamente i confronti

$$r(A) \bowtie_{\theta(A,B)} s(B) \bowtie_{\theta(A \cup B, C)} u(C)$$

oppure

$$r(A) \bowtie_{\theta(A, B \cup C)} s(B) \bowtie_{\theta(B, C)} u(C)$$

Esempio tipico di uso di join

Tipicamente, quando abbiamo 3 o più join, ci sono delle tavole "ponte"

$$r(A) \bowtie_{\theta(A,B)} \mathbf{s(B)} \bowtie_{\theta(B,C)} u(C)$$

In questi casi la parentesizzazione non è necessaria

Esempio:

$$\textit{pazienti} \bowtie_{\textit{pazienti.COD=PAZ}} \mathbf{\textit{ricoveri}} \bowtie_{\textit{Reparto=reparti.COD}} \textit{reparti}$$

Proprietà della selezione multipla

$$\sigma_{p \wedge q}(r(A)) = \sigma_p(\sigma_q(r(A)))$$

La selezione $p \wedge q$ sceglie le tuple che soddisfano sia il predicato p che il predicato q

La selezione di p applicata alle tuple selezionate da q , seleziona le tuple che soddisfano contemporaneamente p e q

Proprietà della sostituzione di operatori

$$\sigma_{p \wedge q}(r(A)) = \sigma_p(r(A)) \cap \sigma_q(r(A))$$

Dimostrazione simile alla precedente derivata dalla semantica degli operatori

(al DBMS costa molto meno la forma a sinistra)

Proprietà della sostituzione di operatori

$$\sigma_{p \vee q}(r(A)) = \sigma_p(r(A)) \cup \sigma_q(r(A))$$

La dimostrazione è immediata (simile alle precedenti)

Proprietà della sostituzione di operatori

$$\sigma_{p \wedge \neg q}(r(A)) = \sigma_p(r(A)) - \sigma_q(r(A))$$

La dimostrazione è immediata (simile alle precedenti)

Proprietà distributiva della selezione rispetto alla proiezione

$$\sigma_p \Pi_X(r(A)) = \Pi_X \sigma_p(r(A))$$

Vale la condizione per cui il predicato p è definito solo su attributi di X

La dimostrazione è quindi immediata

Proprietà distributiva della selezione rispetto alla proiezione

Tutte le proprietà viste finora (distributiva esclusa) valgono come regole di riscrittura

Esempio:

- se trovo $\sigma_{p \wedge q}(r(A))$
- posso scrivere $\sigma_p(r(A)) \cap \sigma_q(r(A))$

ma vale anche il viceversa:

- se trovo $\sigma_p(r(A)) \cap \sigma_q(r(A))$
- posso scrivere $\sigma_{p \wedge q}(r(A))$

Proprietà distributiva della selezione rispetto alla proiezione

Nel caso della proprietà distributiva della selezione

- *se trovo: $\sigma_p \Pi_X(r(A))$*
- *posso sempre riscrivere $\Pi_X \sigma_p(r(A))$*

ma il viceversa **non è sempre valido**, ovvero

- *se trovo: $\Pi_X \sigma_p(r(A))$*
- *affinché possa scrivere $\sigma_p \Pi_X(r(A))$ è necessario che il predicato p sia definito solo su attributi di X*

Esempi

$$\Pi_{COD, AnnoNascita} \sigma_{AnnoNascita > '1980'}(pazienti)$$

posso riscriverla come

$$\sigma_{AnnoNascita > '1980'} \Pi_{COD, AnnoNascita} (pazienti)$$

Invece:

$$\Pi_{COD} \sigma_{AnnoNascita > '1980'}(pazienti)$$

non può essere riscritta come

$$\sigma_{AnnoNascita > '1980'} \Pi_{COD} (pazienti)$$

Proprietà distributiva della selezione rispetto al prodotto cartesiano

$$\sigma_p(r(A) \times s(B))$$

Quando p coinvolge sia attributi di A che attributi di B non c'è nessuna possibilità di applicare proprietà distributive

Se invece p coinvolge solo attributi contenuti nello schema di una delle due relazioni (ad esempio, A), è possibile scrivere

$$\sigma_p(r(A)) \times s(B)$$

Proprietà distributiva della selezione rispetto al prodotto cartesiano

- Una volta applicato il prodotto cartesiano, applichiamo la selezione (per ipotesi) solo alle parti A delle giustapposizione
- Selezionare le tuple giustapposte prendendo solo quelle che soddisfano p è equivalente a selezionare le tuple da r che soddisfano p e giustapparle in seguito alle tuple di s

Proprietà distributiva della selezione rispetto al join

$$\sigma_p(r(A) \bowtie_{\theta} s(B))$$

Quando p coinvolge sia attributi di A che attributi di B non c'è nessuna possibilità di applicare proprietà distributive

Se invece p coinvolge solo attributi contenuti nello schema di una delle due relazioni (ad esempio, A), è possibile scrivere

$$\sigma_p(r(A)) \bowtie_{\theta} s(B)$$

Proprietà distributiva della selezione rispetto al join

Sappiamo che

$$\sigma_p(r(A) \bowtie_{\theta} s(B)) = \sigma_p(\sigma_{\theta}(r(A) \times s(B)))$$

Applichiamo la proprietà della selezione multipla:

$$\sigma_p(\sigma_{\theta}(r(A) \times s(B))) = \sigma_{p \wedge \theta}(r(A) \times s(B))$$

Ma, dato che l'AND è commutativo, posso riapplicare la selezione multipla:

$$\sigma_{p \wedge \theta}(r(A) \times s(B)) = \sigma_{\theta}(\sigma_p(r(A) \times s(B)))$$

Dato che p vede solo attributi di r(A) e avendo già dimostrato la proprietà distributiva rispetto al prodotto cartesiano

$$\sigma_{\theta}(\sigma_p(r(A) \times s(B))) = \sigma_{\theta}(\sigma_p(r(A)) \times s(B)) = \sigma_p(r(A)) \bowtie_{\theta} s(B)$$

Proprietà distributiva della selezione rispetto all'unione e differenza

- $\sigma_p(r(A) \cup s(A)) = \sigma_p(r(A)) \cup \sigma_p(s(A))$
- $\sigma_p(r(A) - s(A)) = \sigma_p(r(A)) - \sigma_p(s(A))$
- $\sigma_p(r(A) \cap s(A)) = \sigma_p(r(A)) \cap \sigma_p(s(A))$

La dimostrazione è immediata (si vede con i diagrammi di Venn)

Per la differenza (e intersezione) vale anche:

$$\sigma_p(r(A) - s(A)) = \sigma_p(r(A)) - s(A)$$
$$\sigma_p(r(A) \cap s(A)) = \sigma_p(r(A)) \cap s(A)$$

Proprietà della proiezione multipla

$$\Pi_X \Pi_{X,Y} r(A) = \Pi_X r(A)$$

dove X e Y sono sottoinsiemi di A

La dimostrazione è immediata dalla definizione di proiezione

Proprietà distributiva della proiezione rispetto al prodotto cartesiano

$$\Pi_X(r(A) \times s(B)) = \Pi_{X_A}(r(A)) \times \Pi_{X_B}(s(B))$$

dove $X_A \subseteq A$ e $X_B \subseteq B$, ovvero

- $X_A = X \cap A$
- $X_B = X \cap B$

La dimostrazione è immediata

Proprietà distributiva della proiezione rispetto al join

$$\Pi_X(r(A) \bowtie_{\theta} s(B)) = \Pi_{X_A}(r(A)) \bowtie_{\theta} \Pi_{X_B}(s(B))$$

dove $X_A \subseteq A$ e $X_B \subseteq B$, ovvero

- $X_A = X \cap A$
- $X_B = X \cap B$

Inoltre, è necessario che gli attributi coinvolti nel predicato θ siano tutti contenuti in $X_A \cup X_B$

La dimostrazione è immediata

Proprietà distributiva della proiezione rispetto all'unione

- $\Pi_X(r(A) \cup s(A)) = \Pi_X(r(A)) \cup \Pi_X(s(A))$

La dimostrazione è immediata

N.B.: le proprietà distributive della proiezione rispetto alla differenza e all'intersezione **non esistono**

Proprietà non valida su differenza

Immaginiamo due relazioni con due soli attributi A e B

R

A	B
a	b_1

S

A	B
a	b_2

con $b_1 \neq b_2$

$$r - s = \begin{array}{|c|c|} \hline \mathbf{A} & \mathbf{B} \\ \hline a & b_1 \\ \hline \end{array} \quad \Pi_A(r - s) = \begin{array}{|c|} \hline \mathbf{A} \\ \hline a \\ \hline \end{array}$$

$$\Pi_A(r) - \Pi_A(s) = \begin{array}{|c|} \hline \mathbf{A} \\ \hline a \\ \hline \end{array} - \begin{array}{|c|} \hline \mathbf{A} \\ \hline a \\ \hline \end{array} = \emptyset$$