Esercizi di BASI DI DATI

(Corso di Laurea in INFORMATICA, A. A. 2012/2013)

Linguaggi d’interrogazione

1. Si vuole progettare una base di dati che contenga i seguenti attributi.

A:	materia	attributo alfanumerico.
B:	codice	attributo numerico di tipo intero
C:	studente	alfanumerico,
D:	voto	attributo numerico di tipo intero

Dopo aver determinato le ovvie dipendenze funzionali esistenti tra gli attributi, si trovi uno schema della base di dati nella forma normale più alta che le dipendenze consentono.
Supponiamo di aver progettato la base di dati utilizzando due variabili relazionali: la prima di nome **`corso`** con schema {A, B} e la seconda di nome **`esame`** con schema {B, C, D}. Si vuole interrogare la base di dati per ottenere una relazione con schema {A, C, D} che riporti gli esami che sono stati superati con il voto più basso in assoluto. Si formuli tale richiesta con un’espressione del calcolo relazionale e la si valuti sulla seguente base di dati attendendosi quanti più possibile ad uno schema algoritmico.

relazione di nome **`corso`**

A	B
Programmazione	1
Algoritmi	2
Architetture	3
Linguaggi	4
Sistemi informativi	5

relazione di nome **`esame`**

B	C	D
1	Angelo	18
1	Maria	25
2	Angelo	30
2	Carlo	25
3	Valeria	28

2. Si considerino le due variabili relazionali di nome **`regno`** e schema {SOVRANO, INIZIO, FINE}, e di nome **`dinastia`** e schema {NOME, SESSO, NASCITA, MORTE}. Dare un’espressione del calcolo relazionale che fornisca i nomi dei membri di sesso maschile della dinastia nati nel XVII secolo che non furono sovrani, e la si valuti sulla seguente base di dati:

`regno`

SOVRANO	INIZIO	FINE
Giacomo I	1603	1625
Carlo I	1625	1648
Carlo II	1660	1685
Giacomo II	1685	1688
Maria II	1688	1694
Anna	1702	1714

`dinastia`

NOME	SESSO	NASCITA	MORTE
Giacomo I	M	1566	1625
Elisabetta	F	1590	1662
Carlo I	M	1600	1649
Carlo II	M	1630	1685
Maria	F	1631	1659
Giacomo II	M	1633	1701
Enrichetta A.	F	1640	1670
Maria II	F	1662	1694
Anna	F	1665	1714
Giacomo E.	M	1686	1766

3. Si consideri lo stato *D* della base di dati dell'Esempio 2. Si fornisca un'espressione del calcolo relazionale che fornisca il nome della prima regina, e la si valuti sulla base di dati *D*.

4. Si consideri lo stato *D* della base di dati dell'Esempio 2. Si dia un'espressione del calcolo relazionale il cui valore sulla seguente relazione fornisca i nomi del re e della regina che hanno regnato più a lungo.

5. Si consideri una base di dati con due variabili relazionali: la prima di nome **`studente`** che ha schema {NOME, MATRICOLA, NAZIONALITÀ}, la seconda di nome **`esame`** che ha schema {MATERIA, MATRICOLA, VOTO}. Si vuole sapere il nome e la nazionalità degli studenti che hanno preso 30 in tutti gli esami che hanno sostenuto. Si formuli la domanda nell'algebra relazionale o nel calcolo relazionale. La si valuti poi sullo stato della base di dati specificato dalle due relazioni:

`studente`

NOME	MATRICOLA	NAZIONALITÀ
Al-Khuwarizmi	1	araba
Gauss	2	tedesca
Galois	3	francese
Hilbert	4	tedesca

`esame`

MATERIA	MATRICOLA	VOTO
Matematica	1	30
Matematica	2	30
Fisica	2	30
Fisica	3	25
Matematica	4	30
Fisica	4	28

6. Si consideri la base di dati con due variabili relazionali: la prima di nome **`studente`** con schema {NOME, MEDIA}, la seconda di nome **`tandem`** con schema {COPPIA, S1, S2}. Si vuole sapere per ogni coppia di studenti la media (dei voti) dello studente più bravo. Si formuli la domanda nel Calcolo Relazionale. La si valuti poi sullo stato della base di dati specificato dalle due relazioni:

`studente`

NOME	MEDIA
Al-Khuwarizmi	25
Gauss	26
Galois	30
Wiles	24
Cantor	29

tandem	COPPIA	S1	S2
	alpha	Al-Khuwarizmi	Gauss
	beta	Galois	Hilbert
	gamma	Wiles	Cantor

7. Si consideri la base di dati dell'Esempio 1. Si vuole progettare una base di dati che contenga i seguenti attributi.

A:	materia	attributo alfanumerico
B:	docente	attributo alfanumerico
C:	studente	attributo alfanumerico
D:	voto	attributo numerico di tipo intero

Supponiamo di aver progettato la base di dati utilizzando due variabili relazionali: la prima di nome **corso** con schema {A, B} e la seconda di nome **esame** con schema {A, C, D}. Si vuole interrogare la base di dati per ottenere una relazione con schema {B, D} che riporti il voto più alto dato da ciascun docente. Si formuli tale richiesta con un'espressione del calcolo relazionale e la si valuti sulla seguente base di dati attenendosi quanto più possibile ad uno schema algoritmico.

relazione di nome **corso**

A	B
Programmazione	Knuth
Algoritmi	Knuth
Architetture	Tanenbaum
Linguaggi	Pascal
Basi di Dati	Ullman

relazione di nome **esame**

A	C	D
Algoritmi	Angelo	18
Algoritmi	Maria	25
Linguaggi	Angelo	30
Basi di Dati	Carlo	25
Basi di Dati	Valeria	30
Programmazione	Angelo	27

soluzione

La domanda può essere formulata nel Calcolo Relazionale usando l'espressione

$$E \quad = \quad \{x(\mathrm{B}, \mathrm{D}) \mid f(x)\}$$

dove

$$f(x) = \exists\, y(\mathrm{A}, \mathrm{B})\ (\mathbf{corso}(y) \wedge y(\mathrm{B}) = x(\mathrm{B}) \wedge g(y) \wedge h(y))$$

con

$$g(y) = \exists\, z(\mathrm{A}, \mathrm{C}, \mathrm{D})\ (\mathbf{esame}(z) \wedge z(\mathrm{A}) = y(\mathrm{A}) \wedge z(\mathrm{D}) = x(\mathrm{D}))$$

$$h(y) = \forall\, u(\mathrm{A}, \mathrm{B})\ (\neg\mathbf{corso}(u) \vee u(\mathrm{B}) \neq y(\mathrm{B}) \vee k(u))$$

dove

$$k(u) = \forall\, v(\mathrm{A}, \mathrm{C}, \mathrm{D})\ (\neg\mathbf{esame}(v) \vee v(\mathrm{A}) \neq u(\mathrm{A}) \vee v(\mathrm{D}) \leq x(\mathrm{D})).$$

8. Si consideri la base di dati dell'Esempio 7.
Si vuole interrogare la base di dati per ottenere una relazione con schema {C} che riporti gli studenti che hanno riportato il voto più alto dato da Knuth. Si formuli tale richiesta con un'espressione del calcolo relazionale e la si valuti sullo stato *D* della base di dati che contiene come relazione di nome **corso** la relazione *r*

A	B
Programmazione	Knuth
Agoritmi	Knuth
Architetture	Tanenbaum
Linguaggi	Pascal
Basi di Dati	Ullman

e come relazione di nome **esame** la relazione *s*

A	C	D
Algoritmi	Angelo	18
Algoritmi	Maria	27
Linguaggi	Angelo	30
Basi di Dati	Carlo	25
Basi di Dati	Valeria	30
Programmazione	Angelo	27

SOLUZIONE

La domanda può essere formulata nel Calcolo Relazionale usando l'espressione

$$E \quad = \quad \{x(\mathrm{C}) \mid f(x)\}$$

dove

$$f(x) = \exists\, y(\mathrm{A, C, D})\ (\mathbf{esame}(y) \wedge y(\mathrm{C}) = x(\mathrm{C}) \wedge g(y) \wedge h(y))$$

con

$$g(y) = \exists\, z(\mathrm{A, B})\ (\mathbf{corso}(z) \wedge z(\mathrm{A}) = y(\mathrm{A}) \wedge z(\mathrm{B}) = \mathrm{Knuth})$$

$$h(y) = \forall\, y'(\mathrm{A, C, D})\ (\neg\,\mathbf{esame}(y') \vee y'(\mathrm{D}) \leq y(\mathrm{D}) \vee k(y'))$$

dove

$$k(y') = \forall\, z'(\mathrm{A, B})\ (\neg\,\mathbf{corso}(z') \vee z'(\mathrm{A}) \neq y'(\mathrm{A}) \vee z'(\mathrm{B}) \neq \mathrm{Knuth}).$$

La valutazione dell'espressione *E* sullo stato *D* della base di dati, richiede la sostituzione di *x* con ogni elemento *c* del dominio dell'attributo C, e la successiva interpretazione della formula $f(x)|_{x=c}$. Ora, l'interpretazione di $f(x)|_{x=c}$ è pari a

$$\vee_{t \in s}\ I[t(\mathrm{C}) = c] \wedge I[g(y)|_{y=t}] \wedge I[h(y)|_{y=t}]$$

dove

$I[g(y)|_{y=t}] = \vee_{u \in r}\, I[u(\mathrm{A}) = t(\mathrm{A})] \wedge I[u(\mathrm{B}) = \mathrm{Knuth}]$

$I[h(y)|_{y=t}] = \wedge_{v \in s}\, I[v(\mathrm{D}) \leq t(\mathrm{D})] \vee I[k(y')|_{y'=v}]$

con

$I[k(y')|_{y'=v}] = \wedge_{w \in r}\, I[w(\mathrm{A}) \neq v(\mathrm{A})] \vee I[w(\mathrm{B}) \neq \mathrm{Knuth}]$.

Esplicitamente si ha che l'interpretazione di $f(x)|_{x=c}$ è vera se solo se $c \in$ {Angelo, Maria} e, dunque, la rispota alla domanda è la relazione

C
Angelo
Maria

9. Supponiamo di aver progettato una base di dati utilizzando due nomi di relazione **corso** ed **esame** i cui schemi siano rispettivamente {MATERIA, DOCENTE} e {MATERIA, STUDENTE, VOTO}. Si vuole interrogare la base di dati per ottenere una relazione con schema {DOCENTE, STUDENTE} che contenga gli studenti (STUDENTE) che abbiano superato tutti gli esami sostenuti con un voto non inferiore a 18 e contenga per ogni studente i docenti (DOCENTE) delle materie oggetto degli esami da lui sostenuti. Si formuli tale richiesta con un'espressione dell'algebra relazionale e la si valuti sullo stato della base di dati

corso

MATERIA	DOCENTE
Programmazione	Knuth
Algoritmi	Knuth
Architetture	Tanenbaum
Linguaggi	Pascal
Basi di Dati	Ullman

esame

MATERIA	STUDENTE	VOTO
Algoritmi	Angelo	10
Algoritmi	Valeria	12
Linguaggi	Angelo	30
Basi di Dati	Carlo	25
Basi di Dati	Valeria	30
Programmazione	Valeria	18
Programmazione	Angelo	27

dettagliando i passi del processo di valutazione.

SOLUZIONE

Siano

$E_1 = \sigma_{\mathrm{VOTO}<18}(\mathbf{esame})$ $\quad E_2 = \pi_{\mathrm{STUDENTE}}(E_1)$

$E_3 = \pi_{\mathrm{STUDENTE}}(\mathbf{esame})$ $\quad E_4 = E_3 - E_2$

$E_5 = \pi_{\mathrm{MATERIA,STUDENTE}}(\mathbf{esame})$ $\quad E_6 = E_4 \bowtie E_5$

$E_7 = \mathbf{corso} \bowtie E_6$ $\quad E_8 = \pi_{\mathrm{DOCENTE,STUDENTE}}(E_7)$

Allora E_8 è un'espressione dell'Algebra Relazionale per la domanda in esame. I valori delle otto espressioni sullo stato della base di dati sono rispettivamente

E_1

MATERIA	STUDENTE	VOTO
Algoritmi	Angelo	10
Algoritmi	Valeria	12

E_2

STUDENTE
Angelo
Valeria

E_3

STUDENTE
Angelo
Valeria
Carlo

E_4

STUDENTE
Carlo

E_5

MATERIA	STUDENTE
Algoritmi	Angelo
Algoritmi	Valeria
Linguaggi	Angelo
Basi di Dati	Carlo
Basi di Dati	Valeria
Programmazione	Valeria
Programmazione	Angelo

E_6

MATERIA	STUDENTE
Basi di Dati	Carlo

E_7

MATERIA	DOCENTE	STUDENTE
Basi di Dati	Ullman	Carlo

E_8

DOCENTE	STUDENTE
Ullman	Carlo

10. Sia **`rel`** il nome di una variabile relazionale con schema {A, B, C}, dove A e B sono rispettivamente il nome ed il cognome di un padre, e C è il nome di un suo figlio (maschio). Si vuole conoscere il nome ed il cognome di tutti i nonni che non hanno nipoti che portino il loro nome. Si formuli tale richiesta con un'espressione *E* del Calcolo Relazionale e la si valuti sulla relazione *r*

A	B	C
Aldo	Rosso	Andrea
Aldo	Rosso	Mario
Franco	Rossi	Andrea
Mario	Bianchi	Filippo
Andrea	Rosso	Filippo
Giulio	Bianchi	Mario
Filippo	Bianchi	Giulio
Andrea	Rossi	Franco
Mario	Bianchi	Giulio

Ottenuto il valore *E*(*r*) dell'espressione *E*, si esplicitino in dettaglio i passi dell'intepretazione per un'ennupla presente in *E*(*r*) e per un'ennupla che non appartiene ad *E*(*r*).

SOLUZIONE

La relazione richiesta deve contenere tutte le coppie (a, b) tali che

(i) (a, b) sia un nonno

(ii) per ogni figlio c di (a, b) si abbia che $c' \neq a$ per ogni figlio c' di c.

Introduciamo la variabile $x(\mathrm{A}, \mathrm{B})$ per indicare la coppia (a, b). Allora, la condizione (i) si esprime come

$$f(x) = \exists\, y(\mathrm{A}, \mathrm{B}, \mathrm{C})\ (\mathtt{rel}(y) \wedge y(\mathrm{A}) = x(\mathrm{A}) \wedge y(\mathrm{B}) = x(\mathrm{B}) \wedge g(x, y))$$

dove

$$g(x, y) = \exists\, z(\mathrm{A}, \mathrm{B}, \mathrm{C})\ (\mathtt{rel}(z) \wedge z(\mathrm{A}) = y(\mathrm{C}) \wedge z(\mathrm{B}) = x(\mathrm{B}))$$

La condizione (ii) si esprime come

$$h(x) = \forall\, u(\mathrm{A}, \mathrm{B}, \mathrm{C})\ (\neg\,\mathtt{rel}(u) \vee u(\mathrm{A}) \neq x(\mathrm{A}) \vee u(\mathrm{B}) \neq x(\mathrm{B}) \vee k(x, u))$$

dove

$$k(x, u) = \forall\, v(\mathrm{A}, \mathrm{B}, \mathrm{C})\ (\neg\,\mathtt{rel}(v) \vee v(\mathrm{A}) \neq u(\mathrm{C}) \vee v(\mathrm{B}) \neq x(\mathrm{B}) \vee v(\mathrm{C}) \neq x(\mathrm{A})).$$

Riassumendo abbiamo l'espressione

$$E = \{x(\mathrm{A}, \mathrm{B}) \mid f(x) \wedge h(x)\}.$$

Il valore di E su r è la relazione

A	B
Aldo	Rosso
Filippo	Bianchi

Per (a, b) = (Aldo, Rosso), abbiamo $I[f(a, b) \wedge h(a, b)]$ = VERO perché

— $I[f(a, b)]$ = VERO quando le variabili y e z vengono sostituite rispettivamente con (Aldo, Rosso, Andrea) e (Andrea, Rosso, Filippo), e

— $I[h(a, b)] = \wedge_{(a,\, b,\, c)\, \in r}\, I[\wedge_{(c,\, b,\, c')\, \in r}\, I[c' \neq a]] = I[\text{Filippo} \neq \text{Aldo}]$ = VERO.

Per (a, b) = (Andrea, Rosso), abbiamo $I[f(a, b) \wedge h(a, b)]$ = FALSO perché

— $I[f(a, b)]$ = FALSO perché Aldo Rosso non è un nonno in quanto suo figlio Filippo non risulta avere figli (cioè non esiste nessuna terna che può essere utilmente sostituita alla variabile z).

11. Sia **`rel`** il nome di una variabile relazionale con schema {A, B, C}. Interpretare (attenendosi ad uno schema algoritmico) la formula

$$f = \exists\, y(\text{A, B, C})\ (\mathbf{rel}(y) \wedge y(\text{A}) = \text{Aldo} \wedge y(\text{B}) = \text{Rosso} \wedge g(y))$$

dove

$$g(y) = \exists\, z(\text{A, B, C})\ (\mathbf{rel}(z) \wedge z(\text{A}) = y(\text{C}) \wedge z(\text{B}) = y(\text{B})),$$

sulla relazione *r*

A	B	C
Aldo	Rosso	Andrea
Aldo	Rosso	Mario
Franco	Rossi	Andrea
Mario	Bianchi	Filippo
Andrea	Rosso	Filippo
Giulio	Bianchi	Mario
Filippo	Bianchi	Giulio
Andrea	Rossi	Franco
Mario	Bianchi	Giulio

Il seguente algoritmo fornisce il valore dell'interpretazione di *f*.

1) *valore* := Falso.

2) Per ogni terna $(i, j, k) \in r$ ripetere

per ogni terna (i', j', k') ripetere

se (i = Aldo) e (j = Rosso) e ($i' = k$) e ($j' = j$)
allora *valore* := Vero ed Uscire.

L'applicazione dell'algoritmo dà valore Vero.

N.B. La formula *f* è interpretabile dal momento che è chiusa. Per lo stesso motivo, *f* non può essere utilizzata da sola per definire un'espressione del Calcolo Relazionale.

12. Siano A, B e C tre attributi binari e sia `rel` una variabile relazionale con schema R = { A, B, C}. Sia

$$E = \{x(R) \mid \mathbf{rel}(x) \wedge f\}$$

dove

$$f = \forall\, x(\text{A, B})\ (\exists\, y(\text{A})\ (x(\text{A}) = y(\text{A}) \wedge x(\text{B}) = 1))\ .$$

Dimostrare nei dettagli che E è un'espressione del Calcolo Relazionale.

Se n sono le ennuple della relazione di nome `rel` contenuta nello stato attuale D della base di dati, quante ennuple contiene la relazione che è il valore di E su D?

Qual è un'espressione dell'Algebra Relazionale equivalente ad E ?

Soluzione. La formula `rel`$(x) \wedge f$ è ben formata. La variabile x è è parzialmente strutturata nella formula $x(\text{B}) = 1$ e vi ha schema $\{\text{B}\}$

è parzialmente strutturata nella formula $(x(\text{A}) = y(\text{A}) \wedge x(\text{B}) = 1)$ e vi ha schema $\{\text{A}, \text{B}\}$

è parzialmente strutturata nella formula $\exists\, y(\text{A})\ (x(\text{A}) = y(\text{A}) \wedge x(\text{B}) = 1)$ e vi ha schema $\{\text{A}, \text{B}\}$

ha solo occorrenze legate nella formula f

è ben strutturata nella formula $\mathbf{rel}(x) \wedge f$ con schema R.

La variabile y

è parzialmente strutturata nella formula $(x(\text{A}) = y(\text{A}) \wedge x(\text{B}) = 1)$ e vi ha schema $\{\text{A}\}$

ha solo occorrenze legate nelle tre formule

$\exists\, y(\text{A})\ (x(\text{A}) = y(\text{A}) \wedge x(\text{B}) = 1) \qquad f \qquad \boldsymbol{r}(x) \wedge f$

Le variabili x ed y che compaiono nell formula f vi hanno solo occorrenze legate; pertanto, f è già interpretabile ed $I[f] = \text{FALSO}$ (esistendo coppie (a, b) con $b \neq 1$). Dunque, per ogni terna (a, b, c) si ha

$I[\mathbf{rel}(x)|_{x=(a,b,c)} \wedge f] = I[\mathbf{rel}(x)|_{x=(a,b,c)}] \wedge I[f] = I[\mathbf{rel}(x)|_{x=(a,b,c)}] \wedge \text{FALSO} = \text{FALSO}$

e quindi il valore E su ogni possibile stato della base di dati è una relazione vuota con schema $\{\text{A}, \text{B}, \text{C}\}$. Un'espressione E' dell'Algebra Relazionale equivalente ad E è

$E' = \mathbf{rel} - \mathbf{rel}$.

13. Si consideri una base di dati sul calcio definita dalle due variabili relazionali: **campione** con schema {ANNO, SQUADRA, ALLENATORE} e **capocannoniere** con schema {ANNO, SQUADRA, GIOCATORE}. Dell'ultima (in ordine di tempo) tra le squadre (se mai ve n'è stata una) che in uno stesso anno non solo ha vinto lo scudetto ma anche ospitava nelle proprie fila il capocannoniere, si vuole conoscere il nome e l'allenatore. Si formuli la domanda nel calcolo relazionale e la si valuti con le seguenti relazioni.

campione

ANNO	SQUADRA	ALLENATORE
1990	A	X
1991	B	Y
1992	A	X
1993	C	X
1994	D	Z
1995	B	U
1996	C	W
1997	A	X
1998	B	V
1999	D	V

Legenda: “X era l’allenatore della squadra A che vinse lo scudetto nel 1990”

capocannoniere

ANNO	SQUADRA	GIOCATORE
1990	C	P
1991	B	Q
1992	D	R
1993	A	R
1994	B	M
1995	B	P
1996	C	Q
1997	A	R
1998	C	S
1999	A	T

Legenda: “P fu il caponniere nel 1990 e militava nella squadra C”

soluzione

La relazione cercata contiene tutte le coppie *x*(SQUADRA, ALLENATORE) tali che esistano sia una terna *y*(ANNO, SQUADRA, ALLENATORE) nella relazione di nome **campione** che una terna *z*(ANNO, SQUADRA, GIOCATORE) nella relazione di nome **capocannoniere** tali che

1) *y*(SQUADRA) = *x*(SQUADRA)

2) *y*(ALLENATORE) = *x*(ALLENATORE)

3) *z*(SQUADRA) = *x*(SQUADRA)

4) *y*(ANNO) = *z*(ANNO)

5) per ogni terna *u*(ANNO, SQUADRA, ALLENATORE) nella relazione di nome **campione** se *u*(ANNO) > *y*(ANNO) allora “la” terna *v*(ANNO, SQUADRA, ALLENATORE) nella relazione di nome **capocannoniere** con *v*(ANNO) = *u*(ANNO) ha *v*(SQUADRA) ≠ *u*(SQUADRA).

Si noti che la condizione 5) assume l’ipotesi (verosimile) che ANNO sia una chiave per entrambi i modelli di **campione** e **capocannoniere**. Se non si volesse far uso di tale assunzione, la condizione (5) andrebbe sostituita da:

per ogni terna *u*(ANNO, SQUADRA, ALLENATORE) nella relazione di nome **campione** se *u*(ANNO) > *y*(ANNO) allora per ogni terna *v*(ANNO, SQUADRA, ALLENATORE) nella relazione di nome **capocannoniere** se *v*(ANNO) = *u*(ANNO) allora *v*(SQUADRA) ≠ *u*(SQUADRA).

Nel Calcolo Relazione abbiamo l’espressione {*x*(SQUADRA, ALLENATORE) | *f*(*x*)} dove *f*(*x*) è nel primo caso

∃ *y*(ANNO, SQUADRA, ALLENATORE)

(**campione**(*y*) ∧ *y*(SQUADRA) = *x*(SQUADRA) ∧
y(ALLENATORE) = *x*(ALLENATORE) ∧

$\exists\, z$(ANNO, SQUADRA, GIOCATORE)

(**capocannoniere**(z) $\wedge$ z(SQUADRA) = x(SQUADRA) $\wedge$
z(ANNO) = y(ANNO) $\wedge$

$\forall\, u$(ANNO, SQUADRA, ALLENATORE)

($\neg$**campione**(u) $\vee$ u(ANNO) $\leq$ y(ANNO) $\vee$

$\exists\, v$(ANNO, SQUADRA, GIOCATORE)

(**capocannoniere**(v) $\wedge$ v(ANNO) = u(ANNO) $\wedge$
v(SQUADRA) $\neq$ u(SQUADRA)))))

ed è nel secondo caso

$\exists\, y$(ANNO, SQUADRA, ALLENATORE)

(**campione**(y) $\wedge$ y(SQUADRA) = x(SQUADRA) $\wedge$
y(ALLENATORE) = x(ALLENATORE) $\wedge$

$\exists\, z$(ANNO, SQUADRA, GIOCATORE)

(**capocannoniere**(z) $\wedge$ z(SQUADRA) = x(SQUADRA) $\wedge$
z(ANNO) = y(ANNO) $\wedge$

$\forall\, u$(ANNO, SQUADRA, ALLENATORE)

($\neg$**campione**(u) $\vee$ u(ANNO) $\leq$ y(ANNO) $\vee$

$\forall\, v$(ANNO, SQUADRA, GIOCATORE)

($\neg$**capocannoniere**(v) $\vee$ v(ANNO) $\neq$ u(ANNO) $\vee$
v(SQUADRA) $\neq$ u(SQUADRA))))

14. Si consideri la base di dati dell'Esempio 2 e sia $\Delta 1$ che contenga due variabili relazionali: **regno1** con schema {SOVRANO, INIZIO, FINE, DURATA} e **dinastia1** con schema {NOME, SESSO, NASCITA, MORTE, VITA}. Sia $D1$ lo stato di $\Delta 1$ ottenuto da D aggiungendo i valori di DURATA = FINE – INIZIO e di VITA = MORTE – NASCITA.

Sia E la seguente espressione del Calcolo Relazione su $\Delta 1$

{x(NOME, DURATA, VITA) | $f(x)$}

dove

$f(x)$ = $\exists\, y$(SOVRANO, INIZIO, FINE, DURATA)
(**regno1**(y) $\wedge$ y(SOVRANO) = x(NOME) $\wedge$ y(DURATA) = x(DURATA)) $\wedge$

$(\exists\, z(\text{NOME, SESSO, NASCITA, MORTE, VITA})$

$(\mathbf{dinastia1}(z) \wedge z(\text{NOME}) = x(\text{NOME}) \wedge z(\text{VITA}) = x(\text{VITA}) \wedge$

$\forall\, u(\text{SOVRANO, INIZIO, FINE, DURATA})$
$(\neg\,\mathbf{regno1}(u) \vee u(\text{DURATA}) \geq x(\text{DURATA}) \vee$

$\exists\, v(\text{NOME, SESSO, NASCITA, MORTE, VITA})$
$(\mathbf{dinastia1}(v) \wedge v(\text{NOME}) = u(\text{SOVRANO}) \wedge$
$(v(\text{SESSO}) \neq z(\text{SESSO}) \vee v(\text{VITA}) \leq x(\text{VITA})))))$

Si dia in chiaro (cioè a parole) la proprietà che caratterizza i sovrani selezionati da *E*.

Si valuti *E* su *D*1.

soluzione

Lo stato *D*1 della base di dati Δ1 contiene le due relazioni riportate nelle due tabelle

SOVRANO	INIZIO	FINE	DURATA
Giacomo I	1603	1625	22
Carlo I	1625	1648	23
Carlo II	1660	1685	25
Giacomo II	1685	1688	3
Maria II	1688	1694	6
Anna	1702	1714	12

NOME	SESSO	NASCITA	MORTE	VITA
Giacomo I	M	1566	1625	59
Elisabetta	F	1590	1662	72
Carlo I	M	1600	1649	49
Carlo II	M	1630	1685	55
Maria	F	1631	1659	28
Giacomo II	M	1633	1701	68
Enrichetta A.	F	1640	1670	30
Maria II	F	1662	1694	32
Anna	F	1665	1714	49
Giacomo E.	M	1686	1766	80

L'espressione *E* richiede i sovrani che hanno regnato meno degli altri sovrani oppure sono vissuti più a lungo degli altri sovrani dello stesso sesso.

Il valore di *E* su *D*1 è dato dalla relazione

NOME	DURATA	VITA
Giacomo II	3	68
Anna	12	49

15. Siano *A*, *B* e *C* tre attributi binari. Si consideri lo stato *D* = {*r*, *s*}

r	A	B
	0	1
	1	0

s	B	C
	1	0
	1	1

della base di dati $\Delta = \{$**rel1**, **rel2**$\}$ dove **rel1** ha schema $\{A, B\}$ ed **rel2** ha schema $\{B, C\}$. Si consideri la seguente espressione "parametrica" del Calcolo Relazionale dove il "parametro" α sta per il quantificatore esistenziale $\exists$ o per quello universale $\forall$:

$$E_\alpha = \{x(A, C) \mid f_\alpha\}$$

dove f_α è la formula

$$\alpha\, y(A, B)\, (\mathbf{rel1}(y) \wedge\ \alpha\, z(B, C)\, (\mathbf{rel2}(z) \wedge\ (g)))$$

dove g è la formula

$$((x(A) = y(A) \wedge (x(C) = z(C)) \wedge (y(B) \neq z(B))).$$

a) Si verifichi che f_α è una formula ben formata e che E_α è un'espressione del Calcolo Relazionale.

b) Calcolare il valore di E_α su D seguendo la procedura di sostituzione ed intepretazione.

c) Qual è un'espressione dell'algebra relazionale equivalente ad E_α?

d) Sia D' lo stato di Δ che risulta dall'aggiunta dell'ennupla (0, 0) alla relazione s. Qual è il valore dell'espressione E_α su D'?

16. Si consideri la Base di Dati dell'Esempio 2. Si fornisca un'espressione del calcolo relazionale ed un'espressione dell'algebra relazionale che fornisca il nome e l'anno di nascita dei membri della dinastia di sesso femminile che non sono state regine, e la si valuti sullo stato D della Base di Dati Araldici.

17. Si consideri una Base di Dati Araldici formata da due variabili relazionali, una di nome **regno** con schema {SOVRANO, INIZIO, FINE} e l'altra di nome **vita** con schema {NOME, SESSO, ANNI}, e lo stato D della base di dati formato dalle due relazioni.

regno	SOVRANO	INIZIO	FINE
	Giacomo I	1603	1625
	Carlo I	1625	1648
	Carlo II	1660	1685
	Giacomo II	1685	1688
	Maria II	1688	1694
	Anna	1702	1714

vita	NOME	SESSO	ANNI
	Giacomo I	M	59
	Elisabetta	F	72

Carlo I	M	49
Carlo II	M	55
Maria	F	28
Giacomo II	M	68
Enrichetta A.	F	30
Maria II	F	32
Anna	F	49
Giacomo E.	M	80

Si fornisca un'espressione del calcolo relazionale E = {x(SOVRANO, INIZIO, FINE) | $f(x)$} che fornisca il nome, l'inizio e la fine del re e della regina che sono vissuti l'uno almeno tanto quanto ogni altro re, e l'altra almeno tanto quanto ogni altra regina. Utilizzando lo stato D, si calcolino le intepretazioni delle quattro formule chiuse $f(x)|_{x=t}$ per ciascuna delle seguenti ennuple t:

Giacomo I	1603	1625
Giacomo II	1685	1688
Maria II	1688	1694
Anna	1702	1714

Soluzione

$f(x)$ = **regno**(x) ∧ ∃ y(NOME, SESSO, ANNI) **vita**(y) ∧ y(NOME) = x(SOVRANO) ∧

∀ u(SOVRANO, INIZIO, FINE) ¬ **regno**(u) ∨
∃ v(NOME, SESSO, ANNI) **vita**(u) ∧ v(NOME) = u(SOVRANO) ∧
(v(SESSO) ≠ y(SESSO) ∨ v(ANNI) ≤ y(ANNI))

Le intepretazioni delle formule chiuse $f(x)|_{x=t}$ per le quattro ennuple sono rispettivamente: Falso, Vero, Falso, Vero.

18. Si consideri una base di dati formata da due variabili relazionali, una di nome **personale** con schema {NOME, PROFESSIONE} e l'altra di nome **parentela** con schema {PERSONA, PADRE, MADRE}, e lo stato D della base di dati formato dalle due relazioni.

personale

NOME	PROFESSIONE
Giacomo	avvocato
Carlo	operaio
Carla	casalinga
Mario	avvocato
Maria	attrice
Anna	casalinga
Elena	attrice

parentela

PERSONA	PADRE	MADRE
Giacomo	Antonio	Ida
Carla	Filippo	Matilde
Maria	Giuseppe	Eleonora
Carlo	Giacomo	Carla
Anna	Carlo	Maria
Mario	Carlo	Maria
Elena	Carlo	Maria

Si fornisca un'espressione del calcolo relazionale E = {x(NOME, PROFESSIONE) |

$f(x)\}$ che fornisca il nome e la professione delle persone che fanno la stessa professione di uno dei genitori o di uno dei nonni. Utilizzando lo stato D, si calcolino le intepretazioni delle tre formule chiuse che si ottengono sostituendo alla variabile x in $f(x)$ le tre ennuple:

Maria	attrice
Anna	casalinga
Elena	attrice

Soluzione

$f(x)$ $\quad$ **personale**(x) $\wedge$ $\exists$ y(PERSONA, PADRE, MADRE) **parentela**(y) $\wedge$ $y(\text{PERSONA}) = x(\text{NOME}) \wedge g(x, y)$

dove

$g(x, y)$ $\quad$ $\exists$ z(PERSONA, PADRE, MADRE) **parentela**(z) $\wedge$ $h(x, y, z)$) $\wedge$ $l(x, z)$

dove

$h(x, y, z)$ $\quad$ $z(\text{PERSONA}) = y(\text{PADRE}) \vee z(\text{PERSONA}) = y(\text{MADRE}) \vee$
($\exists$ v(PERSONA, PADRE, MADRE) **parentela**(v) $\wedge$
$(v(\text{PERSONA}) = y(\text{PADRE}) \vee v(\text{PERSONA}) = y(\text{MADRE})) \wedge$
$(z(\text{PERSONA}) = v(\text{PADRE}) \vee z(\text{PERSONA}) = v(\text{MADRE}))$

$l(x, z)$ $\quad$ $\exists$ u(NOME, PROFESSIONE) **personale**(u) $\wedge$
$u(\text{NOME}) = z(\text{PERSONA}) \wedge u(\text{PROFESSIONE}) = x(\text{PROFESSIONE})$

Le intepretazioni delle formule chiuse $f(x)|_{x=t}$ per le tre ennuple sono rispettivamente: FALSO per (Maria, attrice) e VERO sia per (Anna, attrice) che per (Elena, attrice).

19. Si considerino le seguenti espressioni dell'algebra relazionale sulla Base di Dati formata dalle due variabili relazionali: **regno** con schema {SOVRANO, INIZIO, FINE} e **dinastia** con schema {NOME, SESSO, NASCITA, MORTE}:

$E_1 = \pi_{\text{SOVRANO,INIZIO}}$ (**regno**)
$E_2 = \rho_{\text{SOVRANO}\leftarrow\text{S,INIZIO}\leftarrow\text{I}}\,(E_1)$
$E_3 = E_1 \bowtie_{\text{INIZIO}\geq\text{I}} E_2$
$E_4 = E_3 \div E_2$
$E_5 = \pi_{\text{NOME,NASCITA}}$ (**dinastia**)
$E_6 = (\pi_{\text{SOVRANO}}(E_4)) \bowtie_{\text{SOVRANO=NOME}} E_5$
$E_7 = \pi_{\text{SOVRANO,NASCITA}}\,(E_6)$.

Determinare gli schemi R_i ($i = 1, \ldots, 7$) delle sette espressioni.
Riscrivere (ove necessario) le sette espressioni in maniera che contengano solo gli operatori di selezione, proiezione, join, ridenominazione e differenza.
Per ciascuna delle sette espressioni E_i così ottenute, si dia l'espressione $\{x_i(R_i) \mid f_i(x_i)\}$ del calcolo relazionale equivalente ad E_i.

Soluzione

$E_1 = \pi_{\text{SOVRANO,INIZIO}}(\textbf{regno})$
$R_1 = \{\text{SOVRANO, INIZIO}\}$
$f_1 = \exists\, x(\text{SOVRANO, SOVRANO, FINE})\ \textbf{regno}(x) \wedge x(\text{SOVRANO}) = x_1(\text{SOVRANO})$
$\wedge\ x(\text{INIZIO}) = x_1(\text{INIZIO})$

$E_2 = \rho_{\text{SOVRANO}\leftarrow\text{S,INIZIO}\leftarrow\text{I}}(E_1)$
$R_2 = \{\text{S, I}\}$
$f_2 = \exists\, x_1(R_1)\, f_1(x_1) \wedge x_1(\text{SOVRANO}) = x_2(\text{S}) \wedge x_1(\text{INIZIO}) = x_2(\text{I})$

$E_3 = \sigma_{\text{INIZIO}\geq\text{I}}\ (E_1 \bowtie E_2)$
$R_3 = \{\text{SOVRANO, INIZIO, S, I}\}$
$f_3 = (\exists\, x_1(R_1)\, f_1(x_1) \wedge x_1(\text{SOVRANO}) = x_3(\text{SOVRANO}) \wedge x_1(\text{INIZIO}) = x_3(\text{INIZIO}))$
$\wedge\ (\exists\, x_2(R_2)\, f_2(x_2) \wedge x_2(\text{S}) = x_3(\text{S}) \wedge x_2(\text{I}) = x_3(\text{I})) \wedge$
$x_3(\text{INIZIO}) > x_3(\text{I})$

$E_4 = \pi_{\text{SOVRANO,INIZIO}}(E_3) - \pi_{\text{SOVRANO,INIZIO}}((\pi_{\text{SOVRANO,INIZIO}}(E_3) \bowtie E_2)$
$- E_3)$
$R_4 = \{\text{SOVRANO, INIZIO}\}$
$= \exists\, x_3(R_3)\, f_3(x_3) \wedge x_3(\text{SOVRANO}) = x_4(\text{SOVRANO}) \wedge x_3(\text{INIZIO}) = x_4(\text{INIZIO}) \wedge$
$(\forall\, x_2(R_2)\ \neg f_2(x_2) \vee \exists\, x(R_3)\, f_3(x) \wedge x(\text{SOVRANO}) = x_4(\text{SOVRANO}) \wedge$
$x(\text{INIZIO}) = x_4(\text{INIZIO}) \wedge x(\text{S}) = x_2(\text{S}) \wedge x(\text{I}) = x_2(\text{I}))$

$E_5 = \pi_{\text{NOME,NASCITA}}(\textbf{dinastia})$
$R_5 = \{\text{NOME, NASCITA}\}$
$f_5 = \exists\, x(\text{NOME, SESSO, NASCITA, MORTE})\ \textbf{dinastia}(x) \wedge$
$x(\text{NOME}) = x_5(\text{NOME}) \wedge x(\text{NASCITA}) = x_5(\text{NASCITA})$

$E_6 = \sigma_{\text{SOVRANO=NOME}}\ ((\pi_{\text{SOVRANO}}(E_4) \bowtie E_5)$
$R_6 = \{\text{SOVRANO, NOME, NASCITA}\}$
$f_6 = \exists\, x_4(R_4)\, f_4 \wedge x_4(\text{SOVRANO}) = x_6(\text{SOVRANO}) \wedge$
$\exists\, x_5(R_5)\, f_5 \wedge x_5(\text{NOME}) = x_6(\text{NOME}) \wedge x_5(\text{NASCITA}) = x_6(\text{NASCITA}) \wedge$
$x_6(\text{SOVRANO}) = x_6(\text{NOME})$

$E_7 = \pi_{\text{SOVRANO,NASCITA}}(E_6)$
$R_7 = \{\text{SOVRANO, NASCITA}\}$
$f_7 = \exists\, x_6(R_6)\, f_6 \wedge x_6(\text{SOVRANO}) = x_7(\text{SOVRANO}) \wedge x_6(\text{NASCITA}) = x_7(\text{NASCITA})$

20. Sia Δ una base di dati contenente la variabile relazionale ($\textbf{rel}$, R, d)

dove $R = \{A_1, \ldots, A_{2n}\}$ e $d(A_1) = d(A_2) = \ldots = d(A_{2n}) = \{0, 1\}$. Si consideri la seguente espressione del Calcolo Relazionale su Δ:

$E = \{x(A_{n+1}, \ldots, A_{2n}) \mid f\}$

dove f è la formula

$$f = \exists\, y(R)\ \mathbf{rel}(y) \wedge (y(A_{n+1}) = x(A_{n+1}) \wedge \ldots \wedge y(A_{2n}) = x(A_{2n})) \wedge$$

$$\wedge\ (\forall\, z(R)\ (\neg\, \mathbf{rel}(z) \vee (z(A_{n+1}) \leq y(A_{n+1}) \wedge \ldots \wedge z(A_{2n}) \leq y(A_{2n})))$$

a) Calcolare il valore di E sullo stato di Δ in cui il valore di ρ è relazione r

A_1	A_2	…	A_n	A_{n+1}	A_{n+2}	…	A_{2n}
0	0		0	0	0		0
0	0		0	1	1		1

.

b) Qual è un'espressione dell'Algebra Relazionale su Δ equivalente ad E?

21. Si vuole costruire una base di dati che contenga le seguenti informazioni:

– informazioni anagrafiche sugli studenti: matricola, nome e cognome, sesso, giorno mese ed anno di nascita, comune (o stato estero) di nascita, codice fiscale (*).

– informazioni anagrafiche sulle materie: codice, denominazione, docente, corso di laurea, anno accademico.

informazioni sugli esami: la matricola, il nome e cognome, giorno mese ed anno della seduta d'esame, il codice della materia, il voto d'esame, il docente della materia, anno accademico.

Si definisca una base di dati Δ che contenga tutte le informazioni suddette. Si scriva un'espressione del Calcolo Relazionale su Δ che trovi il cognome ed il sesso degli studenti che hanno superato l'esame di una data materia (si prenda ad es. Programmazione) in un dato anno accademico (si prenda ad es. 2004/05) con il voto più alto (in quella materia e in quell'anno accademico).

22. Si consideri la base di dati dell'Esempio 9. Si vuole interrogare la base di dati per ottenere una relazione con schema {STUDENTE} che riporti gli studenti che hanno riportato il voto più alto dato da Knuth. Si formuli tale richiesta con un'espressione del calcolo relazionale e la si valuti sulla seguente base di dati attenendosi quanto più possibile ad uno schema algoritmico.

corso

A	B
Programmazione	Knuth
Algoritmi	Knuth
Architetture	Tanenbaum
Linguaggi	Pascal
Basi di Dati	Ullman

esame

A	C	D
Algoritmi	Angelo	18
Algoritmi	Maria	27
Linguaggi	Angelo	30
Basi di Dati	Carlo	25
Basi di Dati	Valeria	30
Programmazione	Angelo	27

23. Un testimone di un assassinio dichiara che l'assassino era una persona non giovane di statura alta. Date due variabili relazionali:

indiziato (NUMERO, OCCHI, STATURA, ETÀ)

soggetto (NUMERO, NOME, AVVOCATO)

Si interroghi la base di dati formulando un'espressione *E* del Calcolo Relazionale per conoscere gli avvocati delle persone che rispondono alla descrizione fornita dal testimone.

Si valuti in maniera algoritmica l'espressione *E* sullo stato della base di dati che contiene le due relazioni

indiziato	NUMERO	OCCHI	STATURA	ETÀ
	1	marroni	alta	adulto
	2	verdi	media	giovane
	3	neri	bassa	giovane
	4	verdi	alta	vecchio
	5	verdi	alta	giovane

soggetto	NUMERO	NOME	AVVOCATO
	1	A	H
	2	B	G
	3	C	F
	4	D	H
	5	E	F

24. Si consideri una variabile relazionale di nome **personale** con attributi: NOME, TITOLO, ETÀ. Si voglia selezionare quelle persone che rispondano ai seguenti requisiti:

- hanno un'età minore di 30
- siano laureati ed abbiano conseguito il titolo di dottore di ricerca.

Trovare un'espressione dell'algebra relazionale.

25. Sia **rel** il nome di una variabile relazionale con schema {A, B, C} dove A e B sono rispettivamente il nome ed il cognome di un padre, e C è il nome di un suo figlio maschio. Si vuole conoscere il nome ed il cognome di tutti i nonni che non hanno nipoti che portino il loro nome. Si formuli tale richiesta con un'espressione dell'Algebra Relazionale e la si valuti sulla relazione

A	B	C
Aldo	Rosso	Andrea
Aldo	Rosso	Mario
Franco	Rossi	Andrea
Mario	Bianchi	Filippo
Andrea	Rosso	Filippo
Giulio	Bianchi	Mario
Filippo	Bianchi	Giulio
Andrea	Rossi	Franco
Mario	Bianchi	Giulio

soluzione

$E_1 = \rho_{A \leftarrow A^*, B \leftarrow B^*, C \leftarrow C^*}$ (**rel**)
$E_2 =$ **rel** $\bowtie_{A = C^*, B = B^*, C = A^*} E_1$
$E_3 = \pi_{A,B} (E_2)$
$E = \pi_{A,B}$ (**rel**) $- E_3$

Il valore di *E* è la relazione che contiene le tre coppie (Aldo, Rosso), (Andrea, Rosso), (Filippo, Bianchi): Aldo Rosso ha un solo nipote di nome Filippo, Andrea Rosso non ha nipoti, Filippo Bianchi ha un solo nipote di nome Mario.

Modelli

1. Si consideri il modello $M = [R, F]$, dove $R = abcdeg$ ed $F = \{a \rightarrow b), ae \rightarrow g, bc \rightarrow de\}$.

(*a*) Calcolare la chiusura logica di $\{a, c\}$ rispetto ad F.
(*b*) Dare una derivazione da F della dipendenza funzionale $\{a, c\} \rightarrow R$.

2. Si consideri il modello $M = [R, F]$, dove $R = \{\text{MATRICOLA, PROF, CORSO}\}$ ed $F = \{\{\text{MATRICOLA, CORSO}\} \rightarrow \text{PROF}, \text{PROF} \rightarrow \text{CORSO}\}$. Si provi che:

(*a*) M non è in forma normale di Boyce-Codd.
(*b*) Non esiste alcuna scomposizione nonbanale di M che conservi le dipendenze.

3. Si consideri la relazione

corso

S	M	D
Giacomo	Matematica	Gauss
Giacomo	Fisica	Newton
Carlo	Matematica	Leinitz
Carlo	Fisica	Galileo

avente la seguente semantica: per ogni materia (M), ogni studente (S) di quella materia ha un unico docente (D), ogni docente (D) insegna una sola materia (M). Fornire il modello della relazione **corso** dettato da questa semantica. Provare che tale modello non è in forma normale di Boyce-Codd e discutere le anomalie che possono sorgere in fase d'aggiornamento. Scomporre (in maniera algoritmica) il modello in sottomodelli che siano in forma normale di Boyce-Codd ed analizzare il risultato in termini di conservazione dei dati e delle dipendenze.

4. Si consideri un modello relazionale $M = [R, F]$ dove

$R = \{\text{MATRICOLA, DOCENTE, CORSO}\}$

$F = \{\{\text{MATRICOLA, Corso}\} \rightarrow \text{DOCENTE, DOCENTE,} \rightarrow \text{CORSO}\}$

In quali forme normali si trova il modello M? (Provare la correttezza della risposta che si fornisce)

Soluzione

Il modello M ha due chiavi: {MATRICOLA, CORSO} e {MATRICOLA, DOCENTE}. Dunque, i tre attributi in R sono tutti primari. Ne viene che, se i domini dei tre attributi sono atomici, allora M è sicuramente in terza forma normale (riportare la definizione) e, quindi in seconda e prima forma normale, ma non è in forma normale di Boyce-Codd (riportare la definizione) perché l'attributo (primario) CORSO dipende parzialmente (e quindi transitivamente) dalla chiave {MATRICOLA, DOCENTE}.

5. Si consideri lo schema formato dai quattro attributi SOVRANO, SESSO, INIZIO, FINE e DURATA. Si trovino le dipendenze funzionali e con queste si definisca un modello M. Dire se M è in terza forma normale. Se non lo è, trovare (algoritmicamente) una scomposizione di M in terza forma normale che conservi dati e dipendenze.

6. Il seguente algoritmo trova una chiave del modello $M = [R, F]$ dove $R = \{A_1, \ldots, A_i\}$.

(1) $X := R$;

(2) Per $i = 1, \ldots, n$

se la chiusura logica (rispetto ad F) di $X - \{A_i\}$ coincide con R allora $X := X - \{A_i\}$.

Dimostrare la correttezza dell'algoritmo.

7. Si consideri lo schema R = {MATRICOLA, STUDENTE, MATERIA, DOCENTE, VOTO} e si assumano le dipendenze funzionali

MATRICOLA → STUDENTE,
MATERIA → DOCENTE,
MATRICOLA, MATERIA → VOTO.

Supponendo che i domini degli attributi siano atomici, trovare una scomposizione dello schema R in III forma normale che conservi sia i dati che le dipendenze. (Riportare l'algoritmo di scomposizione ed applicarlo al caso in esame). Decidere se la somposizione trovata è anche in forma normale di Boyce-Codd.

Dopo aver introdotto delle opportune variabili relazionali, una per ciascuno dei sottoschemi di R prodotti dalla scomposizione, formulare nell'Algebra Relazionale e nel Calcolo Relazionale la seguente domanda: quali sono le coppie (a, b) su {STUDENTE, DOCENTE} tali che a abbia superato con 30 almeno un esame sulla materia insegnata da b?

8. Si vuole progettare una base di dati contenente i seguenti attributi:

A:	materia	attributo alfanumerico
B:	docente	attributo alfanumerico
C:	studente	alfanumerico
D:	voto	attributo numerico di tipo intero

Dopo aver individuato le dipendenze funzionali esistenti tra gli attributi (si assuma tra l'altro che A determina B) si trovi uno schema della base di dati nella forma normale più alta che le dipendenze consentono.

9. Si consideri un modello relazionale con schema R = {A, B, C, D, E, F} e dipendenze funzionali {A, C} → E, {A, E} → F, {B, C} → D, {B, D} → A.

(a) Trovare tutte le chiavi del modello.

(b) Applicare il Test di Normalità al modello. Se questo dà esito negativo, trovare una scomposizione conservativa in Terza Forma Normale. Applicare il Test di Normalità a ciascuna delle proiezioni così ottenute per sapere se queste sono anche in Forma Normale di Boyce-Codd.

soluzione

Per il teorema sull'unicità della chiave, la coppia {B, C} è l'unica chiave del modello. Inoltre il modello non è normale e l'applicazione dell'algoritmo per la scomposizione conservativa in Terza Forma Normale dà ***S*** = {{A, C, E}, {A, E, F}, {B, C, D}, {A, B, D}}.

La proiezione del modello su {A, C, E} ha per dipendenze funzionali

> {A, C} → E e tutte le altre dipendenze funzionali valide nel modello che siano applicabili ad {A, C, E}.

Visto che, per ogni sottoinsieme nonvuoto *X* di {A, C, E} con $X \neq$ {A, C}, si ha che *X* coincide con la sua chiusura logica, l'unica dipendenza funzionale nonbanale valida nella proiezione del modello su {A, C, E} è {A, C} → E. Pertanto, la proiezione del modello su {A, C, E} è un modello normale. Analogamente si prova che anche le proiezioni del modello su {A, E, F}, {B, C, D} ed {A, B, D} sono modelli normali.

10. Si consideri il modello *M* = [*R*, *F*] dove

> *R* = {A, B, C, D}
> *F* = {A→BC, B→BC, C→BD, D→AB, BC→AD}

(a) Dopo aver riportato l'Algoritmo di Semplificazione e l'Algoritmo di Riduzione, applicarli per ottenere

> un modello minimale *M′* equivalente ad *M*,
> un modello ridotto *M″* equivalente ad *M*,
> un modello regolare *N* equivalente ad *M*.

(b) Dopo aver riportato l'Algoritmo di Scomposizione in Terza Forma Normale, applicarlo ad *N*. Sia ***S*** il ricoprimento di *R* così ottenuto.

(c) Dopo aver riportato il Test di Conservazione dei Dati ed il Test di Conservazione delle Dipendenze, applicarli per verificare che la scomposizione di *N* indotta da ***S*** conserva sia i dati che le dipendenze.

soluzione

(a) Una soluzione è data da

M′ = [*R*, *F′*] dove *F′* = {A→BC, B→BC, C→BD, D→AB} (BC→AD è ridondante).

M″ = [*R*, *F″*] dove *F″* = {A→BC, B→BC, C→BD, D→AB, C→AD}: l'attributo B (o, alternativamente, l'attributo C) è eliminabile dal determinante della dipendenza funzionale BC→AD).

N = [*R*, *G*] dove *G* = {A→B, B→C, C→D, D→A}, cosicché ogni attributo forma una chiave.

(b) L'applicazione dell'Algoritmo di Scomposizione in Terza Forma Normale ad N genera il ricoprimento $\boldsymbol{S} = \{\text{AB, AD, BC, CD}\}$ di R.

(c) Si verifica facilmente che il Test di Conservazione dei Dati dà esito positivo. Quanto al Test di Conservazione delle Dipendenze si osservi che, siccome ogni attributo forma una chiave di N, per ogni dipendenza $A \rightarrow B$ in G e per ogni $R' = \{A, B\}$ in $\boldsymbol{S}$ si ha che

$$\langle(\{A\} \cap R'\rangle_F = \langle\{A\}\rangle_F = R$$

sicché $A \rightarrow B$ è sempre valida nel sottomodello di N generato da $\boldsymbol{S}$, il quale dunque risulta essere equivalente ad N.

11. Si consideri il modello con schema $R = \{\text{A, B, C, D, E, F, G}\}$ e dipendenze funzionali $F = \{\text{A} \rightarrow \text{D, AB} \rightarrow \text{DE, CE} \rightarrow \text{F, E} \rightarrow \text{G}\}$. Trovare gli attributi primari e secondari del modello. Verificare che il modello non è in seconda forma normale esibendo un attributo secondario A ed una chiave X tale che A dipenda parzialmente da X. Applicare l'algoritmo di scomposizione in terza forma normale. Verificare che i modelli ottenuti dalla scomposizione sono tutti in forma normale di Boyce-Codd.

soluzione

Per il teorema sull'unicità della chiave, il modello $M = [R, F]$ ha un'unica chiave che è data da $X = \{\text{A, B, C}\}$. Pertanto, A, B e C sono gli attributi primari di M, e D, E, F e G sono gli attributi secondari di M. La dipendenza funzionale A→D in F dimostra poi che l'attributo secondario D è parzialmente dipendente dalla chiave X; pertanto, M non è in seconda forma normale. Per applicare l'algoritmo di scomposizione in terza forma normale, occorre prima fornire un modello equivalente ad M che sia in forma canonica. A questo scopo, si eliminano le dipendenze funzionali in F che non sono semplici: $F' = \{\text{A} \rightarrow \text{D, AB} \rightarrow \text{D, AB} \rightarrow \text{E, CE} \rightarrow \text{F, E} \rightarrow \text{G}\}$. Quindi, si eliminano le dipendenze funzionali ridondanti: $F'' = \{\text{A} \rightarrow \text{D, AB} \rightarrow \text{E, CE} \rightarrow \text{F, E} \rightarrow \text{G}\}$. Infine, ove possibile si riducono i determinanti (ma non è questo il caso). Dunque il modello $[R, F'']$ è equivalente ad M ed è in forma canonica. L'applicazione ad $[R, F'']$ dell'algoritmo di scomposizione in terza forma normale genera il ricoprimentro $\boldsymbol{S} = \{R_1, R_2, R_3, R_4, R_5\}$ con $R_1 = \{\text{A, D}\}$, $R_2 = \{\text{A, B, E}\}$, $R_3 = \{\text{C, E, F}\}$, $R_4 = \{\text{E, G}\}$ ed $R_5 = X = \{\text{A, B, C}\}$. Per i modelli M_1, M_2, M_3, M_4, M_5 che si ottengono dalla scomposizione abbiamo le forme canoniche

$$[R_1, \{\text{A} \rightarrow \text{D}\}] \qquad [R_2, \{\text{AB} \rightarrow \text{E}\}] \qquad [R_3, \{\text{CE} \rightarrow \text{F}\}]$$

$$[R_4, \{\text{E} \rightarrow \text{G}\}] \qquad [R_5, \emptyset]$$

e, dunque, essi sono tutti in forma normale di Boyce-Codd.

12. Sia r la relazione con schema R = ABC che contiene le uniche due ennuple (0 0 0) e (1 0 1).

(a) Dato il ricoprimento $\boldsymbol{S} = \{\text{AB, BC}\}$, si elenchino tutte le relazioni appartententi alla classe di equivalenza $[r]$ generata da $\boldsymbol{S}$.

(b) Si calcoli $\psi(r)$ e si verifichi che

$\psi(r)$ è l'elemento massimale di $[r]$,

$\psi(r') \subseteq \psi(r)$ per ogni sottoinsieme r' di r

$\psi(\psi(r)) = \psi(r)$.

(*c*) Si determino tutte le dipendenze funzionali nonbanali semplici soddisfatte da r. Sia F tale insieme e sia $M = [R, F]$. Trovare un modello semplice M' equivalente ad M. Si applichi ad M' l'Algoritmo di Scomposizione in Terza Forma Normale e si verifichi che la scomposizione che se ne ottiene è conservativa.

soluzione

(*a*) $[r]$ contiene le seguenti sette relazioni con schema R = ABC

ABC	ABC	ABC	ABC	ABC	ABC	ABC
000	001	000	000	000	001	000
101	100	001	001	100	100	001
		100	101	101	101	100
						101

(*b*) $\pi_{AB}(r) = \{(0\ 0), (1\ 0)\}$, $\pi_{BC}(r) = \{(0\ 0), (0\ 1)\}$, $\psi(r)$ è l'elemento massimale di $[r]$. I sottoinsiemi di r sono le seguenti quattro relazioni con schema R = ABC

ABC	ABC	ABC	ABC
	000	101	000
			101
$(r_1 = \emptyset)$	(r_2)	(r_3)	$(r_4 = r)$

ed è immediato verificare che $\psi(r_i) \subseteq \psi(r)$, $i = 1, \ldots, 4$

(*c*) L'insieme delle dipendenze funzionali semplici non banali che sono soddisfatte da r è:

$F = \{\text{A} \rightarrow \text{B}, \text{A} \rightarrow \text{C}, \text{AB} \rightarrow \text{C}, \text{AC} \rightarrow \text{B}, \text{BC} \rightarrow \text{A}, \text{C} \rightarrow \text{A}, \text{C} \rightarrow \text{B}\}$.

Un modello semplice equivalente ad $M = [R, F]$ è $M' = [R, F']$ dove

$F' = \{\text{A} \rightarrow \text{C}, \text{C} \rightarrow \text{A}, \text{C} \rightarrow \text{B}\}$

oppure

$F' = \{\text{A} \rightarrow \text{C}, \text{C} \rightarrow \text{A}, \text{A} \rightarrow \text{B}\}$.

Avendo scelto $F' = \{\text{A} \rightarrow \text{C}, \text{C} \rightarrow \text{A}, \text{C} \rightarrow \text{B}\}$, la scomposizione di $M' = [R, F']$ in terza forma normale dà il ricoprimento $\boldsymbol{S} = \{\text{AC}, \text{BC}\}$. Applicando il test di conservazione dei dati ed il test di conservazione delle dipendenze si verifica facilmente che la scomposizione di M' indotta da $\boldsymbol{S}$ è conservativa.

13. Sia r la relazione con schema R = ABC che contiene le tre ennuple (0 0 0), (1 1 0) e (1 1 1).

(*a*) Si calcolino le proiezioni di r su AB e AC. Sia poi r^* la relazione ottenuta facendo il prodotto join delle proiezioni di r su AB e AC. Si vuole sapere se r^* soddisfa o meno la dipendenza funzionale A $\rightarrow$ B. Per questo, si dia l'algoritmo nel caso generale e si illustrino i passi compiuti dall'algoritmo con input r^* ed A $\rightarrow$ B.

(*b*) Si determino tutte le dipendenze funzionali nonbanali semplici soddisfatte dalla relazione r. Sia F tale insieme e sia $M = [R, F]$. Trovare un modello semplice M' equivalente ad M. Si applichi ad M' l'Algoritmo di Scomposizione in Terza Forma Normale e si verifichi che la scomposizione che se ne ottiene è conservativa.

14. Si consideri il modello $M = [R, F]$, dove

$$R = \{\mathrm{A, B, C, D, E}\}$$

$$F = \{\{\mathrm{A, B}\} \rightarrow \mathrm{C}, \mathrm{C} \rightarrow \mathrm{A}, \mathrm{D} \rightarrow \mathrm{A}\}.$$

(*a*) Trovare gli attributi primari e secondari del modello M.

(*b*) Dire (provandolo) in quali forme normali si trova M.

(*c*) Applicare l'algoritmo di scomposizione in terza forma normale in maniera da ottenere una scomposizione che conservi dati e dipendenze.

15. Si consideri il modello $M = [R, F]$, dove

$$R = \{\mathrm{A, B, C}\}$$
$$F = \{\mathrm{A} \rightarrow \mathrm{B}, \mathrm{B} \rightarrow \mathrm{C}\}.$$

(*a*) Utilizzare il teorema sull'unicità della chiave per provare che esiste un'unica chiave di M.

{A} è una sovrachiave ed è quindi l'unica chiave.

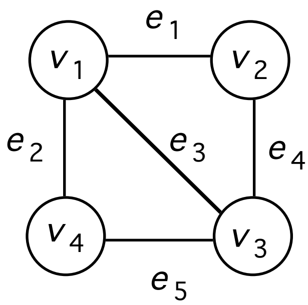
(*b*) Si consideri il ricoprimento $\boldsymbol{S}^* = \{\{\mathrm{A, B}\}, \{\mathrm{A, C}\}\}$ di R. Si trovino le dipendenze funzionali nonbanali presenti in ciascuna delle due proiezioni di M su $\boldsymbol{S}^*$. Si provi in maniera algoritmica che la scomposizione di M indotta da $\boldsymbol{S}^*$ conserva i dati ma non le dipendenze.

Le dipendenze funzionali nonbanali presenti nella proiezione di M su {A, B} sono A $\rightarrow$ B, A $\rightarrow$ {A, B}; quelle presenti nella proiezione di M su {A, C} sono A $\rightarrow$ C, A $\rightarrow$ {A, C}. L'applicazione del Test di Conservazione dei Dati dà esito positivo e l'applicazione del Test di Conservazione delle Dipendenze dà esito negativo per via della dipendenza funzionale B $\rightarrow$ C.

(*c*) Applicare ad M l'algoritmo di scomposizione in terza forma normale in maniera da ottenere un ricoprimento $\boldsymbol{S}$ di R che induca una scomposizione di M che conservi dati e dipendenze. Si trovino le dipendenze funzionali nonbanali presenti in ciascuna delle proiezioni di M su $\boldsymbol{S}$. Si provi che l'applicazione del Test di Normalità a ciascuna delle proiezioni di M su $\boldsymbol{S}$ dà esito positivo.

L'applicazione dell'Algoritmo di Scomposizione in Terza Forma Normale produce il ricoprimento $\boldsymbol{S} = \{\{\mathrm{A, B}\}, \{\mathrm{B, C}\}\}$. Le dipendenze funzionali nonbanali presenti nella proiezione di M su {A, B} sono A $\rightarrow$ B, A $\rightarrow$ {A, B}; quelle presenti nella proiezione di M su {B, C} sono B $\rightarrow$ C, B $\rightarrow$ {B, C}. L'applicazione del Test di Normalità alla proiezione di M su {A, B} ed alla proiezione di M su {B, C} dà entrambe le volte esito positivo, sicché la scomposizione ottenuta è in Forma Normale di Boyce-Codd.

16. Si consideri il grafo G mostrato in Figura.



Si costruisca poi il modello $M = [R, F]$ tale che le coperture di G corrispondano alle sovrachiavi di M. Trovare le coperture minimali di G e verificare che ciascuna di esse corrisponde ad una chiave di M.

Le coperture minimali di G sono $\{v_1, v_2, v_4\}$, $\{v_1, v_3\}$ e $\{v_2, v_3, v_4\}$.

Nel modello $M = [R, F]$, R è $\{A_1, A_2, A_3, A_4, A_5, B_1, B_2, B_3, B_4\}$ ed F contiene le dipendenze funzionali

$B_1 \to A_1$, $B_1 \to A_2$, $B_1 \to A_3$

$B_2 \to A_1$, $B_2 \to A_4$

$B_3 \to A_3$, $B_3 \to A_4$, $B_3 \to A_5$

$B_4 \to A_2$, $B_4 \to A_5$

$\{A_1, A_2, A_3, A_4, A_5\} \to \{B_1, B_2, B_3, B_4\}$

Così, è facile vedere che le chiusure logiche rispetto ad F dei tre insiemi $\{B_1, B_2, B_4\}$, $\{B_1, B_3\}$ e $\{B_2, B_3, B_4\}$ sono tutte uguali ad R. Inoltre, siccome la chiusura logica rispetto ad F di nessuno dei sottoinsiemi propri dei tre insiemi $\{B_1, B_2, B_4\}$, $\{B_1, B_3\}$ e $\{B_2, B_3, B_4\}$ coincide con R, essi sono tutti e tre chiavi di M.

17. Si ricordi che un modello $M = [R, F]$ è *regolare* se

— F è minimale,
— il determinante di ogni dipendenza funzionale in F è irriducibile,
— il dipendente di ogni dipendenza funzionale in F è formato da un singolo attributo.

Trovare in maniera algoritmica un modello regolare $M = [R, F]$ equivalente al modello $M^* = [R, F^*]$ dove

R è formato dai sette attributi: A, B, C, D, E, I, J

$F^* = \{A \rightarrow BCE, AB \rightarrow DE, BI \rightarrow J, I \rightarrow A\}$

Soluzione

$F = \{A \rightarrow B, A \rightarrow C, A \rightarrow D, A \rightarrow E, I \rightarrow A, I \rightarrow J\}$

18. Dato un insieme $R = \{A_1, \ldots, A_{2n}\}$ di attributi binari, $n \geq 1$, si consideri la seguente relazione r con schema R :

A_1	A_2	$\ldots$	A_n	A_{n+1}	A_{n+2}	$\ldots$	A_{2n}
0	0		0	0	0		0
0	0		0	1	1		1

Sia F l'insieme delle dipendenze funzionali (banali e non) soddisfatte da r.

a) Dare una caratterizzazione delle dipendenze funzionali in F.

b) Calcolare il numero di dipendenze funzionali nonbanali presenti in F.

c) Quali e quante sono le chiavi del modello $M = [R, F]$?

soluzione

(*a*) Una dipendenza funzionale $X \rightarrow Y$ è soddisfatta da r se e solo se

$$X \cap \{A_{n+1}, \ldots, A_{2n}\} \neq \emptyset \qquad \text{oppure} \qquad X \cup Y \subseteq \{A_1, \ldots, A_n\} .$$

(*b*) Siano $X' = X \cap \{A_1, \ldots, A_n\}$ ed $X'' = X \cap \{A_{n+1}, \ldots, A_{2n}\}$. Nel primo caso, possiamo scegliere X' in 2^n modi ed X'' in $2^n–1$ modi. Quindi, possiamo scegliere X in $2^n(2^n–1)$ modi, mentre possiamo scegliere Y arbitrariamente cioè in $2^{2n}–1$ modi. Pertanto, le dipendenze funzionali del primo tipo sono in numero di

$$2^n\,(2^n–1)\,(2^{2n}–1).$$

Nel secondo caso, X e Y sono arbitrari sottoinsiemi non vuoti di $\{A_1, \ldots, A_n\}$. Pertanto, le dipendenze funzionali del secondo tipo sono in numero di

$$(2^n–1)^2.$$

In totale dunque abbiamo

$$N = 2^n\,(2^n–1)\,(2^{2n}–1) + (2^n–1)^2$$

dipendenze funzionali soddisfatte da r. Un'espressione più semplice di N è

$$N = (2^n–1)\,(2^{3n}–1)$$

Ora da N dobbiano sottrarre le dipendenze funzionali banali che sono in numero di

$$3^{2n} – 2^{2n}$$

cosicché il numero di dipendenze funzionali nonbanali presenti in F è dato da

$$(2^n-1)\,(2^{3n}-1) - 3^{2n} + 2^{2n}\ .$$

Nel caso $n = 1$, tale numero è pari a 2 ed infatti F contiene le sole due dipendenze funzionali nonbanali $A_2 \rightarrow A_1$ e $A_2 \rightarrow \{A_1, A_2\}$.

(c) Ovviamente, ogni singolo attributo A_i con $n+1 \leq i \leq 2n$ forma da solo una chiave del modello. Pertanto, eventuali altre chiavi non possono che essere sottoinsiemi di $\{A_1, \ldots, A_n\}$. Sia ora X un tale sottoinsieme e calcoliamo $\langle X\rangle_F$ con l'algoritmo di Armstrong. Dopo aver esaminato le dipendenze funzionali in F del secondo tipo, avremo $\langle X\rangle_F = \{A_1, \ldots, A_n\}$. A questo punto, nessuna delle dipendenze funzionali in F del primo tipo potrà ulteriormente incrementare $\langle X\rangle_F$. Ne segue che nessun sottoinsieme di $\{A_1, \ldots, A_n\}$ può essere una sovrachiave e, tanto meno una chiave, del modello. Dunque le chiavi del modello sono gli n insiemi $\{A_{n+1}\}$, $\{A_{n+2}\}$, $\ldots$, $\{A_{2n}\}$. Si osservi che il modello è in forma normale di Boyce-Codd se e solo se $n = 1$ e che, per $n > 1$, ogni dipendenza della forma $A_i \rightarrow A_j$ $(1 \leq i < j \leq n)$ è critica per il modello.

9. Si vuole costruire una base di dati che contenga le seguenti informazioni:

– informazioni anagrafiche sugli studenti: matricola, nome e cognome, sesso, giorno mese ed anno di nascita, comune (o stato estero) di nascita, codice fiscale (*).

– informazioni anagrafiche sulle materi: codice, denominazione, docente, corso di laurea, anno accademico.

informazioni sugli esami: la matricola, il nome e cognome, giorno mese ed anno della seduta d'esame, il codice della materia, il voto d'esame, il docente della materia, anno accademico.

Si definisca una base di dati che contenga tutte le informazioni suddette e sia almeno in Terza Forma Normale.

(*) Si ricorda che il codice fiscale è una sequenza di 16 caratteri alfanumerici, così formata: tre caratteri alfabetici determinati dal cognome, tre caratteri alfabetici determinati dal nome, due caratteri numerici determinati dall'anno di nascita, un carattere alfabetico determinato dal mese di nascita, due caratteri numerici determinati dal giorno di nascita e dal sesso, quattro caratteri (uno alfabetico e tree numerici) determinati dal comune italiano o stato estero di nascita, un carattere alfabetico (di controllo) determinato dai primi quindici caratteri del codice fiscale.

20. Dato un modello $M = [R, F]$ e due ricoprimenti $\boldsymbol{S}$ ed $\boldsymbol{S}'$, si dimostri che

se $\boldsymbol{S}$ è più fine di $\boldsymbol{S}'$ ed $\boldsymbol{S}$ induce una scomposizione conservativa di M, allora anche $\boldsymbol{S}'$ induce una scomposizione conservativa di M,

se $\boldsymbol{S}$ è più fine di $\boldsymbol{S}'$ ed $\boldsymbol{S}'$ non induce una scomposizione conservativa di M, allora neppure $\boldsymbol{S}$ induce una scomposizione conservativa di M.

Siano $R = \{A, B, C, D\}$, $F = \{A \rightarrow C, B \rightarrow C, B \rightarrow D\}$ ed $\boldsymbol{S} = \{AC, AD, BCD\}$. Decidere in maniera algoritmica (riportando gli algoritmi) se $\boldsymbol{S}$ induce una scomposizione conservativa del modello $M = [R, F]$. In caso negativo, trovare un ricoprimento di R che induca una scomposizione conservativa di M.

21. Sia $M = [R, F]$ un modello in prima forma normale, dove $R = ABCD$ ed $F = \{A \rightarrow B, B \rightarrow CD, C \rightarrow A\}$.

(*a*) Utilizzando il criterio dell'unicità della chiave, provare che M non ha un'unica chiave

(*b*) Una chiave di M si puo' ottenere in maniera algoritmica partendo da R ed eliminando (se necessario) attributi finché non si trova una sovrachiave minimale (utilizzando l'algoritmo di Chiusura). Applicare questo metodo due volte: una volta esaminando gli attributi in R in ordine alfabetico, e l'altra al contrario.

(*c*) Si applichi il Test di Normalità per sapere se M è in forma normale di Boyce-Codd, e se è in terza forma normale.

(*d*) Utilizzando una delle due chiavi ottenute al punto (*b*), applicare ad M l'Algoritmo di Scomposizione in Terza Forma Normale, e verificare (facendo uso dei Test di Conservazione dei Dati e delle Dipendenze) che il ricoprimento di R generato dall'algoritmo induce una scomposizione conservativa di M.

22. Si consideri il modello $M = [R, F]$, dove $R = \{A, B, C, D, E\}$ ed $F = \{A \rightarrow B, B \rightarrow A, \{A, B\} \rightarrow C, C \rightarrow D\}$. Applicare il Test di Normalità e, se questo dà esito negativo, trovare una dipendenza critica in F. A questo punto, mostrare, con un esempio di relazione con schema R, le anomalie di aggiornamento e la ridondanza dei dati.

23. Sia $M = [R, F]$ un modello regolare che abbia un'unica chiave. Sia questa X. Dare una condizione su X ed F che sia necessaria e sufficiente perché M sia in terza forma normale.

Fornire due esempi di modelli regolari con un'unica chiave, dei quali solo uno sia in terza forma normale. Applicare ad entrambi l'algoritmo di Scomposizione in Terza Forma Normale.

soluzione

Sia X l'unica chiave di M cosicché gli attributi primari di M sono gli elementi di X.

Ora, se F è un insieme vuoto, allora M è banalmente in terza forma normale. Assumiamo che F non sia un insieme vuoto e sia $F = \{X_1 \rightarrow A_1, \ldots, X_n \rightarrow A_n\}$, $n > 0$. Siccome X è l'unica chiave di M, si ha $X = R \setminus \{A_1, \ldots, A_n\}$. Pertanto, gli attributi secondari di M sono $A_1, \ldots, A_n$. Distinguiamo due casi:

Caso 1: $X_i = X$ per ogni valore di i. Allora si vede facilmente che M è non solo in Terza Forma Normale ma anche in Forma Normale di Boyce-Codd.

Caso 2: $X_i \neq X$ per un qualche valore di i. Allora, siccome il determinante di ogni dipendenza funzionale in F è ridotto ai minimi termini, X_i non può contenere propriamente X e, quindi, la dipendenza funzionale $X_i \rightarrow X$ non è implicata da F. Pertanto, l'attributo secondario A_i dipende transitivamente da X (tramite X_i).

Riassumendo, condizione su X ed F che sia necessaria e sufficiente perché M sia in terza forma normale è che o $F = \emptyset$ oppure il determinante di ogni dipendenza funzionale in F coincida con la chiave di M.

24. Si consideri il modello $M = [R, F]$ dove

R = {A, B, C, D}
F = {A→BC, B→BC, C→BD, D→AB, BC →AD}.

Dopo aver riportato gli algoritmi di Riduzione e di Contrazione, applicarli per un modello regolare N equivalente ad M.

Dopo aver riportato l'Algoritmo di Scomposizione in Terza Forma Normale, applicarlo ad N. Sia $\boldsymbol{S}$ il ricoprimento di R così ottenuto.

Dopo aver riportato il Test di Conservazione dei Dati ed il Test di Conservazione delle Dipendenze, applicarli per verificare che la scomposizione di N indotta da $\boldsymbol{S}$ conserva sia i dati che le dipendenze.

soluzione

Prima di applicare gli algoritmi di Riduzione e di Contrazione, conviene rendere le dipendenze semplici e si ottiene:

A→B, A→C, B→B, B→C, C→B, C→D, D→A, D→B, BC→A, BC→D.

Dopo aver applicato l'algoritmo di Riduzione, si ottiene l'insieme minimale:

A→B, B→C, C→D, D→A.

Siccome il determinante di ogni dipendenza contiene un solo attributo, l'algoritmo di Contrazione non apporta modifiche. Pertanto un modello regolare equivalente ad M è modello $N = [R, G]$ dove

R = {A, B, C, D}
G = {A→B, B→C, C→D, D→A}.

L'Algoritmo di Scomposizione in Terza Forma Normale, applicato ad N, dà il ricoprimento $\boldsymbol{S}$ = {AB, BC, CD, AD} di R, e i Test di Conservazione dei Dati e di Conservazione delle Dipendenze danno entrambi esito positivo.

Organizzazione fisica

1. Sia T un B-albero in cui sia memorizzato un file "principale" di 1000 record. Sappiamo che un blocco di memoria ha una capacità massima di 19 record, tanto per i blocchi contenenti record del file "principale" (che si considerano parte di *T*) quanto per quelli contenenti record dell'indice. Sappiamo inoltre che ogni blocco di *T* contiene esattamente 10 record.
(a) Quanti livelli ha *T*?
(b) Che numero *minimo* di inserzioni è necessario effettuare in *T* perché esso cresca di un livello?
(c) Che numero *massimo* di inserzioni è possibile effettuare in *T* prima che esso cresca di un livello?

2. (a) Si descriva la procedura di ricerca per una interrogazione a intervallo su k-d-tree. (b) Si supponga che per un dato un file R(*X*, *Y*, *Z*) con *X*, *Y* e *Z* campi di tipo intero, sia stata adotttata una organizzazione di tipo k-d-tree. A partire dal k-d-tree iniziale vuoto, si inseriscano nell'ordine i seguenti record: (5, 5, 5), (7, 2, 0), (4, 3, 3), (3, 7, 5), (3, 7, 1), (4, 2, 10), (2, 7, 3), (3, 7, 7), (4, 1, 5). Disegnare il k-d-tree risultante. (c) Illustrare la procedura di ricerca sul k-d-tree ottenuto al punto b) nel caso dell'interrogazione [$3 \leq X < 5$] and [$Y \geq 4$].

3. Considerare un file in cui ciascun record contenga un intero *e* su cui sia definito un indice a B-albero con blocchi di capacità $2e-1 = 2d-1 = 9$. Sia T l'albero ottenuto inserendo i primi 54 interi positivi nell'ordine naturale: 1, 2, ... 54.
(*a*) Descrivere cosa avviene in T a seguito della cancellazione del record contenente 46.
(*b*) Descrivere cosa avviene in T a seguito della cancellazione del record contenente 12.

4. Qual è il massimo numero di blocchi che possono essere creati a seguito dell'inserzione di un record in un B-albero di profondità *n*?
Qual è il numero massimo di blocchi che possono essere eliminati da un B-albero di profondità *n* a seguito di una cancellazione?

5. Considerare due strategie alternative per la cancellazione di un record da un B-albero: nella prima le chiavi dei record (*k*, *p*) dell'indice vengono aggiornate in modo che *k* sia sempre uguale alla più piccola chiave del file principale presente nel sottoalbero puntato da *p*. Nella seconda no. Mostrare un B-albero sul quale una cancellazione seguita da una inserzione producano nei due casi una diversa distribuzione dei record del file principale.

6. Descrivere l'algoritmo di cancellazione di un record in un B-albero.

7. Considerare una base di dati in cui i record hanno come chiave un intero positivo. Le inserzioni nella base di dati si susseguono nel tempo rispettando sempre l'ordinamento naturale delle chiavi: 1, 2, …, *n*. Nessun intero positivo viene "saltato" e non sono previste cancellazioni di record. Quale (o quali) delle seguenti organizzazioni fisiche è da preferirsi alle altre? B-albero; indice ISAM con ricerca binaria; indice ISAM con ricerca per interpolazione; insieme di bucket con funzione di accesso hash.

8. Definire in dettaglio (incluso il formato dei blocchi e dei record) l'organizzazione fisica di una base di dati i cui record contengano parole di testo (una per ogni record). I record possono essere puntati e sono previste operazioni di *look-up*, inserzione e cancellazione. Mostrare lo stato della base di dati quando vi sia inserita una strofa della canzone *O' Sole mio*.

9. Si vuole memorizzare un file di record *non puntati* utilizzando un kd-tree. Ogni record contiene il nome e l'età di un bambino che frequenta la scuola materna del paese. Si assuma di utilizzare come chiavi sia il nome del bambino che l'età. Si mostri lo stato dettagliato del kd-tree dopo l'inserimento dei seguenti record: Pia (anni 2), Camillo (anni 4), Rosina (anni 2), Piero (anni 2), Enzo (anni 4), Amelia (anni 4), Mariagrazia (anni 5), Angelo (anni 4), Mario (anni 4), Teresa (anni 5), Nadia (anni 4), Ugo (anni 2), Claudio (anni 3), Alessandro (anni 2). Si descriva passo per passo la seguente operazione di ricerca nel kd-tree descritta informalmente: quali bambini hanno almeno 4 anni?

10. Si vuole memorizzare un file di record *non puntati* utilizzando un *B-tree*. Ogni record contiene un intero memorizzato in un campo che usiamo come chiave. Ogni blocco, che sia all'interno o che sia una foglia, può contenere al più 3 record. Si mostri lo stato dettagliato del B-tree dopo l'inserimento dei seguenti record:

149, 80, 122, 184, 78, 106, 47, 133, 64, 166, 85, 179.

Si mostri poi lo stato dettagliato del B-tree dopo l'inserimento di:

177, 176, 193, 48, 72, 115, 199, 93, 119, 107, 12, 1.

Infine, si mostri lo stato dettagliato del B-tree dopo la cancellazione di tutti i numeri compresi tra 170 e 179.
Si descriva passo passo la seguente operazione di ricerca nel B-tree descritta informalmente: quali record hanno un campo chiave maggiore o uguale a 100?

11. È dato un file di 250.000 record. I record sono a lunghezza fissa non puntati. Ogni record occupa 325 byte, di cui 75 per la chiave. Un puntatore a blocco occupa 5 byte. Un blocco di memoria contiene 2048 byte. Utilizziamo una organizzazione hash con una funzione che prende come argomento la chiave, e distribuisce uniformemente i record in 350 bucket.

- Quanti blocchi di memoria occupa la bucket directory?
- Quanti blocchi di memoria occupa ogni bucket?
- Qual è il numero medio di accessi a blocco richiesti per ricercare un record?

12. Supponiamo di avere un file di 700.000 record. Ogni record occupa 180 byte, di cui 60 per il campo chiave. Ogni blocco contiene 2048 byte. Un puntatore a blocco occupa 5 byte. Non è richiesto l'allineamento dei campi ad indirizzi multipli di 4 e i record non sono puntati. Usiamo una organizzazione B-tree. Qual è il numero minimo di blocchi che dobbiamo utilizzare complessivamente per il file indice (totale dei blocchi di tutti i livelli) e per il file principale (livello foglia)?

13. Supponiamo di avere un file di 1.850.000 record. Ogni record occupa 370 byte, di cui 55 per il campo chiave. Ogni blocco contiene 2048 byte. Un puntatore a blocco

occupa 5 byte. Non è richiesto l'allineamento dei campi ad indirizzi multipli di 4 e i record non sono puntati. Usiamo una organizzazione ISAM. Qual è il numero minimo di blocchi che dobbiamo utilizzare complessivamente per il file indice e per il file principale, supponendo di lasciare in ogni blocco il 20% (al massimo) di spazio libero?

14. Per un file R di 2.350.000 records vengono creati due indici densi I_1 ed I_2 su due distinte chiavi del file k_1 e k_2 rispettivamente. Il primo indice è un *file ordinato* ISAM su k_1 ed il secondo indice è una tabella hash con chiave k_2. Entrambe le chiavi individuano univocamente un record del file.
Individuare il numero di bucket della tabella hash (supponendo che la funzione di hash distribuisca uniformemente i records) tale che il numero medio di blocchi letti per una lookup sia pari al numero di blocchi letti per una lookup tramite l'indice I_1. I parametri del problema sono i seguenti

lunghezza blocco	2048 byte
lunghezza record	520 byte
lunghezza puntatore	4 byte
lunghezza k_1	22 byte
lunghezza k_2	53 byte

Calcolare il numero totale di blocchi per i due indici ed il file principale allocati su disco.

15. Un *file R* viene memorizzato tramite indice sparso *B-tree*. Il *B-tree* è costruito su di una chiave *k* il cui dominio è l'insieme degli interi. Viene sottoposta al sistema la seguente domanda

Q **select** * **from** *R*
where *k* **between** 1 **and** 13.250

ovvero la domanda *Q* seleziona 13.250 *record* utilizzando la chiave *k*. Indicare il numero complessivo di blocchi letti per eseguire la domanda *Q* nell'ipotesi che *R* abbia i seguenti parametri:

numero di record	5.460.000
lunghezza blocco	8096 byte
lunghezza record	945 byte
lunghezza puntatore	4 byte
lunghezza chiave *k*	98 byte

e che i blocchi foglia e blocchi indice del *B-tree* siano pieni al 70%. Si supponga che, per accedere ad un blocco foglia, si debba sempre passare per la radice dell'albero.

Il numero di record che possono stare in un blocco è pari a $\lfloor 8096/945 \rfloor = 8$. Poiché per ipotesi i blocchi sono pieni al 70%, un blocco foglia contiene $\lfloor 8 \times 0{,}7 \rfloor = 5$ record. Il numero di record indice N che possono essere in un blocco è determinato dal fatto che

$$N \times lp + (N-1) \times lk \leq lb$$

quindi N = $\lfloor(8096 + 98)/(98+4)\rfloor = 80$. Poiché per ipotesi i blocchi sono pieni al 70%, un blocco indice contiene $\lfloor 80 \times 0,7\rfloor = 56$ record. Abbiamo allora che

numero blocchi foglia = $\lceil 5.460.000/5\rceil = 1.092.000$
numero blocchi indice 1° livello = $\lceil 1.092.000/56\rceil = 19.500$
numero blocchi indice 2° livello = $\lceil 19.500/56\rceil = 349$
numero blocchi indice 3° livello = $\lceil 349/56\rceil = 6$
numero blocchi indice 4° livello = $\lceil 6/56\rceil = 1$

L'accesso ad un blocco foglia passando per la radice dell'albero richiede quindi la lettura di 5 blocchi. Poiché la domanda *Q* richiede 13.250 record che sono memorizzati consecutivamente secondo l'ordine dato dal *B-tree*, occorre leggere al più $\lceil 13.250/5\rceil = 2.650$ blocchi foglia. Quindi il numero complessivo di blocchi letti nell'ipotesi che occorra passare sempre per la radice è $2650 \times 5 = 13.250$.

16. Per un file R viene utilizzato un indice denso su chiave **k** con organizzazione a B^*-tree. La differenza con il B-tree è che ogni blocco del solo livello foglia contiene un puntatore al blocco successivo. Supponiamo che il dominio della chiave **k** sia l'insieme degli interi e che venga proposta al sistema la seguente query:

select *
from R
where **k** between 1.000 and 3.250

la quale legge 2.250 records. Individuare e descrivere un algoritmo efficiente per eseguire la query facendo uso dell'indice e calcolare il numero di blocchi letti dall'algoritmo nel caso peggiore. I parametri del problema sono i seguenti

numero di records	2.250.000
lunghezza blocco	4.096 byte
lunghezza puntatore	2 byte
lunghezza chiave **k**	49 byte

SOLUZIONE

Riportiamo i dati del problema

nr=numero di records	2.250.000
lb=lunghezza blocco	4.096 byte
lp=lunghezza puntatore	2 byte
lk=lunghezza chiave **k**	49 byte

Poiché abbiamo un indice denso, i blocchi foglia conterranno dei puntatori ai singoli records del file R, che supponiamo memorizzato come un heap. Nel caso peggiore l'indice avrà i blocchi pieni per metà. In un blocco foglia possono stare N puntatori dove N soddisfa il seguente vincolo

$$N * lp + (N-1) * lk \leq lb - 2 \qquad (1)$$

dove si è tolto 2 dalla lunghezza del blocco per tenere conto dello spazio del puntatore al blocco successivo. Pertanto

$$N = \lfloor (4096+49-2) / (49+2) \rfloor = 81$$

Nell'ipotesi che i blocchi siano pieni a metà abbiamo che il numero di puntatori in un blocco foglia è

$$\lceil 81 / 2 \rceil = 41$$

Avendo un indice denso dobbiamo avere al livello foglia tanti puntatori quanti records del file. Quindi il numero blocchi livello foglia dell'indice è

$$\lceil 2.250.000 / 41 \rceil = 54.879$$

per i livelli dell'indice abbiamo che il numero minimo è anch'esso 41 calcolato dal vincolo (1) tenendo presente però che non dobbiamo considerare lo spazio occupato dal puntatore al blocco successivo. Pertanto otteniamo

num blocchi primo livello	$\lceil 54.879 / 41 \rceil = 1.339$
num blocchi secondo livello	$\lceil 1.339 / 41 \rceil = 33$
num blocchi terzo livello	$\lceil 33 / 41 \rceil = 1$

L'altezza dell'albero è quindi di h=4 livelli. L'algoritmo per eseguire la query può sfruttare il fatto che i blocchi foglia dell'indice sono collegati tra di loro con un puntatore e che la query richiede 2.250 records consecutivi. Possiamo quindi 1) individuare il primo blocco foglia che contiene il puntatore al primo record da leggere e 2) scorrere i blocchi foglia utilizzando il puntatore che li lega per recuperare gli altri puntatori ai records da leggere. Il numero complessivo di blocchi foglia da leggere è quindi

$$nbl = \lceil 2.250 / 41 \rceil = 55$$

Se indichiamo con ns = 2.250 il numero di records selezionati dalla query e supponendo che nel caso peggiore, ogni record da leggere si trovi in un blocco diverso dell'heap allora il numero complessivo dei blocchi letti dall'algoritmo è pari $h + nbl + ns = 4 + 55 + 2.250 = 2.309$.

17. Per un *file* di 2.350.000 record vengono creati due indici densi I_1 ed I_2 su due distinte chiavi del file, rispettivamente k_1 e k_2. Il primo indice è un *b-tree* su k_1 ed il secondo indice è una tabella *hash* con chiave k_2. Entrambe le chiavi individuano univocamente un record del file.
Determinare il numero di *bucket* della tabella *hash* (supponendo che la funzione di *hash* distribuisca uniformemente le chiavi) tale che il numero medio di blocchi letti per una *lookup* sia pari al numero di blocchi letti per una *lookup* nel caso peggiore, tramite l'indice I_1. I parametri del problema sono i seguenti

lunghezza blocco	2048 byte
lunghezza puntatore	4 byte
lunghezza k_1	22 byte
lunghezza k_2	53 byte

Si assuma che il file principale sia memorizzato come un *heap* e che, come usuale, i record non siano spezzati tra i blocchi.

Soluzione

Riportiamo i dati del problema

$R = 2.350.000$
$B = 2048$
$lk_1 = 22$
$lk_2 = 53$
$lp = 4$

Cominciamo a calcolare l'altezza del b-tree relativo all'indice I_1. Essendo un indice denso questo contiene un numero records puntatori pari al numero di records del file principale. Il numero di records per blocco RB è al massimo

$$RB = \lfloor (B + lk_1) / (lk_1 + lp) \rfloor = \lfloor (2048 + 22) / (22 + 4 \rfloor = 79$$

Quindi dovendo considerare il caso peggiore supponiamo che i blocchi siano pieni a metà. Pertanto considerando un numero di records puntatori per blocco pari a 39 abbiamo che

num blocchi primo livello	$\lceil 2.350.000 / 39 \rceil = 60.257$
num blocchi secondo livello	$\lceil 60.257 / 39 \rceil = 1.546$
num blocchi terzo livello	$\lceil 1.546 / 39 \rceil = 40$
num blocchi quarto livello	$\lceil 40 / 39 \rceil = 2$
num blocchi quinto livello	$\lceil 2 / 39 \rceil = 1$

Quindi per accedere ad un records occorrono 5 + 1 accessi. Ne consegue che la struttura dell'indice I_2 deve consentire l'accesso al file con un numero medio di 6 accessi. Il singolo bucket deve contenere un numero di blocchi non maggiore di 10. Calcoliamo il numero di records puntatore per l'indice I_2 che possono stare in un blocco:

$$\lfloor B / (lk_2 + lp) \rfloor = \lfloor 2048 / (53 + 4) \rfloor = 35$$

Quindi se ogni bucket deve contenere 10 blocchi questo conterrà al più 350 records puntatori. Tenendo anche conto che l'indice I_2 è un indice denso, il numero di buckets deve essere $\lceil 2.350.000 / 350 \rceil = 6715$.

18. Per un file R di 2.350.000 records vengono creati due indici densi I_1 ed I_2 su due distinte chiavi del file k_1 e k_2 rispettivamente. Il primo indice è un *b-tree* su k_1 ed il secondo indice è una tabella hash con chiave k_2. Entrambe le chiavi individuano univocamente un record del file.
Individuare il numero di bucket della tabella hash (supponendo che la funzione di hash distribuisca uniformemente le chiavi) tale che il numero medio di blocchi letti per una lookup sia pari al numero di blocchi letti per una lookup, nel caso peggiore, tramite l'indice I_1. I parametri del problema sono i seguenti

lunghezza blocco	2048 byte

lunghezza puntatore	4 byte
lunghezza k_1	22 byte
lunghezza k_2	53 byte

Si assuma che il file principale sia memorizzato come un heap e che, come usuale, i record non sono spezzati tra i blocchi.

19. È dato un file di 3.725.000 record, ognuno di 256 byte. La chiave occupa 35 byte. Un blocco contiene 4096 byte. Un puntatore a blocco occupa 5 byte. Si vuole indicizzare il file tramite una organizzazione hash con 2000 bucket, supponendo di utilizzare una funzione che distribuisca uniformemente i record nei bucket. Trovare occupazione in blocchi della bucket directory, occupazione totale dei bucket con la distribuzione descritta e costo medio di una ricerca per valori non ripetuti. Dire inoltre cosa occorrerebbe fare per diminuire il costo medio di una ricerca della metà.

Gestione delle transazioni

1. Si considerino le seguenti transazioni:

T1: LOCK A; LOCK B; UNLOCK B; UNLOCK A
T2: LOCK B; LOCK C; UNLOCK C; UNLOCK B
T3: LOCK C; LOCK A; UNLOCK A; UNLOCK C

(a) Esistono scheduling non seriali di T1, T2 e T3 ma serializzabili?
(b) Esistono scheduling di T1, T2 e T3 non serializzabili?

2. Si dimostri che se il grafo di serializzabilità di uno scheduling S non ha cicli, allora S è serializzabile. (b) Si consideri lo scheduling S delle transazioni T1, T2, T3, T4 e T5 dato da:

1. T1: lock A	11. T4: unlock C
2. T1: lock B	12. T2: unlock B
3. T4: lock C	13. T2: lock C
4. T3: lock D	14. T3: lock B
5. T3: unlock D	15. T5: unlock D
6. T1: unlock B	16. T4: unlock A
7. T2: lock B	17. T5: lock A
8. T1: unlock A	18. T2: unlock C
9. T4: lock A	19. T3. unlock B
10. T5: lock D	20. T5: unlock A.

Si applichi l'algoritmo per il test di serializzabilità ad S.

3. Dimostrare che qualunque schedule di transazioni a due fasi è serializzabile.

4. Mostrare un insieme *T* di transazioni tali che: nessuna transazione in *T* segue il protocollo a due fasi, ma qualunque schedule di transazioni in *T* è seriale.

5. Chiamiamo "sicuro" un insieme S di transazioni tale che nessuna esecuzione concorrente di elementi di S può produrre una situazione di stallo. Sia A < B < ... < Z un ordinamento su un insieme di risorse e sia S1 l'insieme di tutte le transazioni T che rispettano il protocollo che consiste nel richiedere i Lock rispettando l'ordinamento, cioè per le quali, se Lock X precede Lock Y in T, allora la risorsa X precede la risorsa Y nell'ordinamento. Sia S2 l'insieme ottenuto aggiungendo ad S1 la transazione T1:

Lock A
Lock C
Unlock C
Lock B
Unlock B
Unlock A

(*a*) L'insieme S2 è sicuro?
(*b*) Se si è risposto negativamente ad (*a*) fornire un esempio di stallo. Se si è risposto affermativamente fornire un nuovo protocollo tale che l'insieme di tutte le transazioni che lo rispettano sia sicuro ed includa S2.

6. Considerare il seguente *schedule* di due transazioni T_1 e T_2:

T_1: Lock A

T_1: Lock B

T_1: Unlock B (α)

T_2: Lock B

T_2: Unlock B (β)

T_1: Unlock A (γ)

T_1: Lock C

T_1: Unlock C (δ)

Chiamiamo rispettivamente a, b e c i valori iniziali delle risorse A, B e C. Esprimere in termini delle funzioni f, g, h ed k, associate ripsettivamente alle azioni α, β, γ e δ, i valori delle risorse A, B e C.

7. Sia $R_1 < R_2 < \ldots < R_k$ un ordinamento totale delle risorse di un sistema concorrente. Si consideri il seguente protocollo P rispettato da tutte le transazioni del sistema: se una transazione effettua un lock su R_i e successivamente un lock su R_j, allora $i \leq j$. Dimostrare o confutare formalmente la seguente affermazione: il protocollo P garantisce che tutti gli scheduling sono serializzabili.

8. Sia dato il seguente scheduling per l'insieme di transazioni T1, T2, T3

T1	T2	T3
	READ(X)	
		READ(Y)
	X = X - 20	
READ(X)		
X = X + 10		
		Y = Y - 5
WRITE(X)		
		WRITE(Y)
	WRITE(X)	
READ(Y)		
Y = Y + 20		
WRITE(Y)		

Il controllo della concorrenza è basato su timestamp, e le transazioni hanno i seguenti timestamp:

TS(T1) = 110, TS(T2) = 100, TS(T3) = 105.

Descrivere come vengono modificate durante lo schedule le variabili READ_TIMESTAMP e WRITE_TIMESTAMP associate agli item X ed Y, e come procedono le tre transazioni. Fornire inoltre i valori finali dei due item.

9. Sia dato il seguente schedule S per l'insieme di transazioni T1, T2, T3, T4, T5:

T1	T2	T3	T4	T5
LOCK A				
	LOCK B			
UNLOCK A				
	UNLOCK B			
	LOCK A			
		LOCK B		
	UNLOCK A			
				LOCK A
		UNLOCK B		
				UNLOCK A
		LOCK A		
			LOCK B	
		UNLOCK A		
			UNLOCK B	
LOCK B				
			LOCK A	
			UNLOCK A	
UNLOCK B				
				LOCK B
				UNLOCK B

Verificare se lo schedule è serializzabile, e in questo caso fornire lo schedule seriale equivalente (o gli schedule seriali equivalenti se ne esiste più di uno).

10. Sia dato il seguente schedule S per l'insieme di transazioni T1, T2, T3, T4, T5:

T1	T2	T3	T4	T5
LOCK A				
		LOCK B		
	LOCK C			
		UNLOCK B		
				LOCK B
				UNLOCK B
UNLOCK A				
LOCK B				
			LOCK A	
	UNLOCK C			
		LOCK C		
			UNLOCK A	
				LOCK A
		UNLOCK C		
				LOCK C
UNLOCK B				
				UNLOCK C
			LOCK C	
			UNLOCK C	
				UNLOCK A

Verificare se lo schedule è serializzabile, e in questo caso fornire lo schedule seriale equivalente (o gli schedule seriali equivalenti se ne esiste più di uno). In caso contrario indicare i cicli contenuti nel grafo.

11. Sia dato il seguente scheduling S. Decidere se è serializzabile. Se si indicare almeno due scheduling seriali equivalenti, se no indicare il ciclo del grafo di serializzazione.

T1	T2	T3	T4	T5
			lock(g)	
	lock (y)			
				lock(x)
lock(z)				
	unlock(y)			
				unlock(x)
			unlock(g)	
lock(g)				
unlock(g)				
		lock(x)		
unlock(z)				
				lock(z)
		unlock(x)		
lock(y)				
	lock(x)			
unlock(y)				unlock(z)
	unlock(x)			

12. Sia dato il seguente *scheduling*

T1	T2	T3
read (X)		
	read (Y)	
		read(X)
X:=X+2		
write (X)		
	read (X)	
	X:=X+Y	
	write (X)	
		X:=X+10
		write (X)

dove i *timestamp* delle transazione sono i seguenti

TS(T1) = 200 TS(T2) = 100 TS(T3) = 150.

Descrivere il comportamento dello *scheduler* ed indicare il ReadTS e WriteTS delle variabili *X* ed *Y* ad ogni passo dello *scheduling*.

200	100	150	X		Y	
T1	T2	T3	RTS	WTS	RTS	WTS
read (X)			200			

	read (Y)			100	
		read(X)			
X:=X+2					
write (X)			200		
	read (X)				T2 abortisce poiché tenta di leggere un valore X con time stamp in scrittura maggiore di TS(T2)
	X:=X+Y				
	write (X)				
		X:=X+10			
		write (X)			T3 abortisce poiché tenta di scrivere un valore X con time stamp in lettura maggiore di TS(T3)

13. Sia dato il seguente scheduling. Si assuma che la gestione della concorrenza sia basata sui *timestamp*. I timestamp delle transazioni sono TS(T1) = 150, TS(T2) = 100 e TS(T3) = 200

T1	**T2**	**T3**
		read (X)
read (Y)		
		read (Y)
	read(X)	
		X:=X+Y*32
		write (X)
read (X)		
X:=X+Y		
write (X)		
	X:=X+20	
	write (X)	

Descrivere come vengono modificate durante lo schedule le variabili READ_TIMESTAMP e WRITE_TIMESTAMP associate agli item X ed Y, e come procedono le tre transazioni.

Soluzione

150	100	200		X		Y	Note
T1	T2	T3	Read TS	Write TS	Read TS	Write TS	
		read (X)	200	0	0	0	
read (Y)			200	0	150	0	
		read (Y)	200	0	200	0	
	read (X)		200	0	200	0	
		X:=X+Y*32	200	0	200	0	
		write (X)	200	200	200	0	
read (X)			200	200	200	0	T1 abortisce poiché nella lettura di X si ha TS(T1) < WriteTS(X)
X:=X+Y			200	200	200	0	
write (X)			200	200	200	0	
		write (Y)	200	200	200	200	
	X:=X+20		200	200	200	200	
	write (X)		200	200	200	200	T2 abortisce poiché nella scrittura di X si ha TS(T2) < ReadTS(X)

14. Sia dato il seguente *scheduling*. Creare il grafo di serializzazione e decidere se è serializzabile. Se sì, indicare almeno due scheduling seriali equivalenti; altrimenti, individuare il ciclo del grafo di serializzazione.

T1	T2	T3	T4	T5
	lock(g)			
			lock (y)	
				lock(x)
lock(z)				
			unlock(y)	
				unlock(x)
	unlock(g)			
lock(g)				
unlock(g)				
		lock(x)		
unlock(z)				
				lock(z)
		unlock(x)		
lock(y)				
			lock(x)	
				unlock(z)
unlock(y)				
			unlock(x)	

Soluzione

T1	T2	T3	T4	T5
	lock(g)			
			lock (y)	
				lock(x)
lock(z)				
			unlock(y)	
				unlock(x)
	unlock(g)			
lock(g)				
unlock(g)				
		lock(x)		
unlock(z)				
				lock(z)
		unlock(x)		
lock(y)				
			lock(x)	
				unlock(z)
unlock(y)				
			unlock(x)	

Il Grafo G = (V, E) di serializzazione è tale che V = {1, 2, 3, 4, 5} ed E = {(2, 1), (1, 5), (5, 3), (3, 4), (4, 1)}. Lo scheduling non è serializzabile poiché esiste un ciclo C = (1, 5 , 3, 4).

15. Sia dato il seguente scheduling S. Creare il grafo *G* di serializzazione e decidere se è serializzabile. Se sì, indicare almeno due scheduling seriali equivalenti; se no, indicare il ciclo del grafo di serializzazione.

T1	T2	T3	T4	T5
	lock(g)			
			lock (y)	
				lock(x)
lock(z)				
			unlock(y)	
				unlock(x)
	unlock(g)			
lock(g)				
unlock(g)				
		lock(x)		
unlock(z)				
				lock(z)
		unlock(x)		
lock(y)				
			lock(x)	
				unlock(z)
unlock(y)				
			unlock(x)	

Soluzione

T1	T2	T3	T4	T5
	lock(g)			
			lock (y)	
				lock(x)
lock(z)				
			unlock(y)	
				unlock(x)
	unlock(g)			
lock(g)				
unlock(g)				
		lock(x)		
unlock(z)				
				lock(z)
		unlock(x)		
lock(y)				
			lock(x)	
				unlock(z)
unlock(y)				
			unlock(x)	

Il grafo $G = (V, E)$ ha $V = \{1, 2, 3, 4, 5\}$ ed $E = \{(2, 1), (1, 5), (5, 3), (3, 4), (4 ,1)\}$. Lo schedule non è serializzabile poiché G contiene il ciclo (1, 5, 3, 4).

16. È dato il seguente schedule di 5 transazioni T1, T2, T3, T4, T5.

T1	T2	T3	T4	T5
		LOCK(B)		
LOCK(A)				
	LOCK(C)			

UNLOCK(A)				
		UNLOCK(B)		
	UNLOCK(C)			
				LOCK(B)
				UNLOCK(B)
LOCK(C)				
			LOCK(A)	
	LOCK(B)			
UNLOCK(C)				
	UNLOCK(B)			
			UNLOCK(A)	
			LOCK(C)	
		LOCK(A)		
			UNLOCK(C)	
		UNLOCK(A)		
				LOCK(C)
LOCK(B)				
				UNLOCK(C)
UNLOCK(B)				

Verificare se lo schedule è serializzabile. Nel caso lo sia, elencare tutti gli schedule seriali equivalenti. In caso contrario elencare tutti i cicli presenti.