

## Basi di Dati – Secondo compito – 4/6/2019 – Compito 1

1) Si consideri il seguente schema relazionale

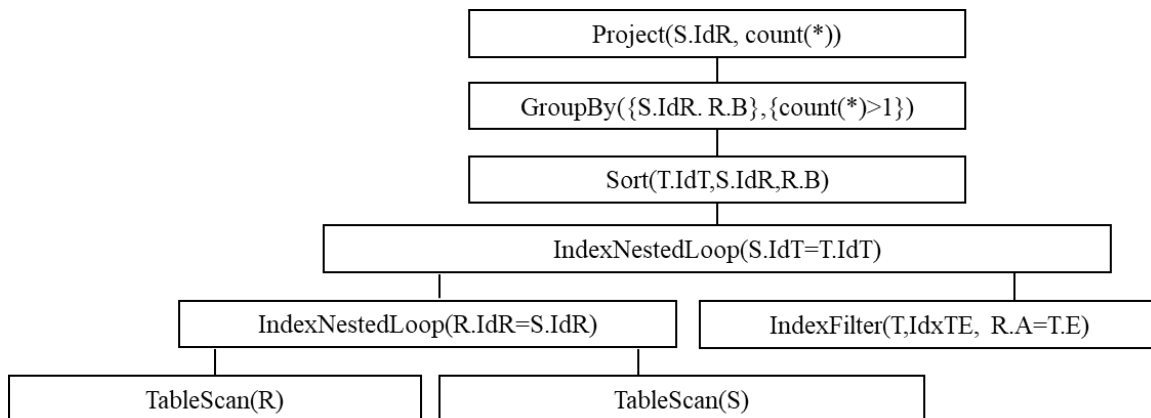
$R \langle ABCDE, F = \{ CE \rightarrow A, D \rightarrow E, CB \rightarrow E, CE \rightarrow B \} \rangle$

- Trovare almeno una chiave.
  - Sapendo che l'insieme degli attributi primi è  $\{ C, D \}$ , verificare quali forme normali sono rispettate dallo schema.
  - La dipendenza  $BD \rightarrow A$  sarebbe ridondante rispetto ad  $F$ ? La dipendenza  $BCD \rightarrow A$  conterrebbe attributi estranei?
  - Applicare l'algoritmo di sintesi e dire se dati e dipendenze sono stati preservati.
  - Applicare l'algoritmo di analisi e dire se dati e dipendenze sono stati preservati.
- 2) Si consideri il seguente schema relazionale  $R(\underline{IdR}, A, B)$ ,  $S(\underline{IdS}, IdR^*, IdT^*, C, D)$ ,  $T(\underline{IdT}, IdR^*, E)$  e la seguente interrogazione

```

SELECT DISTINCT S.IdR, count(*)
FROM R, S, T
WHERE R.IdR = S.IdR and S.IdT=T.IdT and R.A = T.E
GROUP BY S.IdR, R.B
HAVING count(*) > 1
    
```

- Opzionale: La clausola **DISTINCT** è ridondante?
- Si scriva un piano di accesso logico
- Si scriva un piano di accesso fisico efficiente che non fa uso di indici, e (opzionale) verificare se la sort prima della Group By può essere evitata
- Si scriva un piano di accesso fisico efficiente che fa uso di due indici, e (opzionale) verificare se la sort prima della Group By può essere evitata
- Si consideri il seguente piano di accesso e si elenchino gli errori in esso presenti



- 3) Si consideri un sistema gestito con protocollo disfare-rifare, si supponga che avvenga un fallimento di sistema, e si assuma che al momento della ripartenza questo sia il contenuto del log, dove un record (W,T1,A,1,10) sta a indicare una scrittura (Write) della transazione T1 sulla variabile A con vecchio valore 1 e con nuovo valore 10:

(begin,T1) (W,T1,A,0,20) (begin,T2) (W,T2,B,0,30) (begin-ckp,{T1,T2}) (W,T2,B,30,50)  
(W,T1,C,0,30) (end-ckp) (begin,T3) (commit,T2) (W,T3,B,50,70) (begin,T4)  
(W,T4,D,0,50) (commit,T4)

- a) Al momento in cui il sistema aveva iniziato a prendere il log, qual era il valore di A, B, C, D?
- b) Qual era il valore di A, B, C, D nel buffer al momento del checkpoint
- c) Al termine del checkpoint qual è il valore di A, B, C, D sul disco?
- d) Quali sono le transazioni che erano riuscite a dare il commit prima del fallimento di sistema e quali no?
- e) Al momento del fallimento di sistema qual è il valore di A, B, C, D nel buffer?
- f) Cosa possiamo dire riguardo al valore di A, B, C, D sul disco al momento del fallimento di sistema?
- g) Quali sono le transazioni che al momento della ripartenza dovranno essere rifatte e quali sono quelle che dovranno essere disfatte?
- h) Si elenchino tutte le operazioni che devono essere rifatte, nell'ordine in cui saranno rifatte
- i) Si elenchino tutte le operazioni che devono essere disfatte, nell'ordine in cui saranno disfatte

## Basi di Dati – Secondo compito – 4/6/2019 – Compito 2

1) Si consideri il seguente schema relazionale

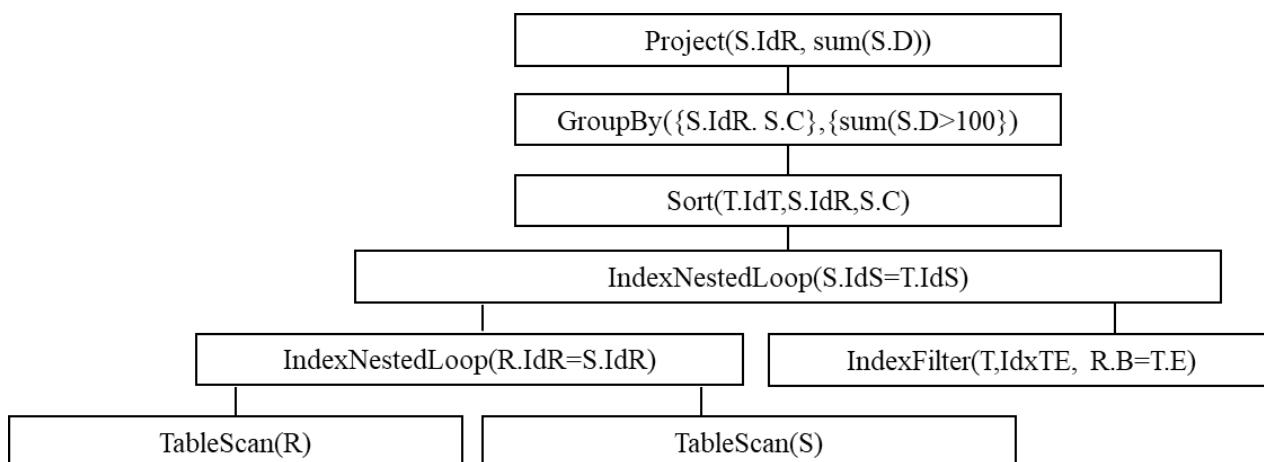
$R \langle ABCDE, F = \{ BD \rightarrow C, E \rightarrow B, AD \rightarrow B, BD \rightarrow A \} \rangle$

- Trovare almeno una chiave.
  - Sapendo che l'insieme degli attributi primi è  $\{ D, E \}$ , verificare quali forme normali sono rispettate dallo schema.
  - La dipendenza  $AE \rightarrow C$  sarebbe ridondante rispetto ad  $F$ ? La dipendenza  $ADE \rightarrow C$  conterrebbe attributi estranei?
  - Applicare l'algoritmo di sintesi e dire se dati e dipendenze sono stati preservati.
  - Applicare l'algoritmo di analisi e dire se dati e dipendenze sono stati preservati.
- 2) Si consideri il seguente schema relazionale  $R(\underline{IdR}, A, B)$ ,  $S(\underline{IdS}, IdR^*, IdT^*, C, D)$ ,  $T(\underline{IdT}, IdS^*, E)$  e la seguente interrogazione

```

SELECT DISTINCT S.IdR, sum(S.D)
FROM R, S, T
WHERE R.IdR = S.IdR and S.IdS=T.IdS and R.B = T.E
GROUP BY S.IdR, S.C
HAVING sum(S.D) > 100
    
```

- Opzionale: La clausola **DISTINCT** è ridondante?
- Si scriva un piano di accesso logico
- Si scriva un piano di accesso fisico efficiente che non fa uso di indici, e (opzionale) verificare se la sort prima della Group By può essere evitata
- Si scriva un piano di accesso fisico efficiente che fa uso di due indici, e (opzionale) verificare se la sort prima della Group By può essere evitata
- Si consideri il seguente piano di accesso e si elenchino gli errori in esso presenti



- 3) Si consideri un sistema gestito con protocollo disfare-rifare, si supponga che avvenga un fallimento di sistema, e si assuma che al momento della ripartenza questo sia il contenuto del log, dove un record (W,T1,A,1,10) sta a indicare una scrittura (Write) della transazione T1 sulla variabile A con vecchio valore 1 e con nuovo valore 10:

(begin,T1) (W,T1,A,0,20) (begin,T2) (W,T2,B,0,30) (begin-ckp,{T1,T2}) (W,T2,B,30,50)  
(W,T1,C,0,30) (end-ckp) (begin,T3) (commit,T2) (W,T3,B,50,70) (begin,T4)  
(W,T4,D,0,50) (commit,T4)

- a) Al momento in cui il sistema aveva iniziato a prendere il log, qual era il valore di A, B, C, D?
- b) Qual era il valore di A, B, C, D nel buffer al momento del checkpoint
- c) Al termine del checkpoint qual è il valore di A, B, C, D sul disco?
- d) Quali sono le transazioni che erano riuscite a dare il commit prima del fallimento di sistema e quali no?
- e) Al momento del fallimento di sistema qual è il valore di A, B, C, D nel buffer?
- f) Cosa possiamo dire riguardo al valore di A, B, C, D sul disco al momento del fallimento di sistema?
- g) Quali sono le transazioni che al momento della ripartenza dovranno essere rifatte e quali sono quelle che dovranno essere disfatte?
- h) Si elenchino tutte le operazioni che devono essere rifatte, nell'ordine in cui saranno rifatte
- i) Si elenchino tutte le operazioni che devono essere disfatte, nell'ordine in cui saranno disfatte

## Basi di Dati – Secondo compito – 4/6/2019 – Compito 1 – Soluzioni – V0.1

1) Si consideri il seguente schema relazionale

$R \langle ABCDE, F = \{ CE \rightarrow A, D \rightarrow E, CB \rightarrow E, CE \rightarrow B \} \rangle$

a) Trovare almeno una chiave.

L'insieme ABCDE è una superchiave. Proviamo a eliminare gli attributi partendo dal primo per arrivare a una superchiave senza attributi estranei, ovvero una chiave.

~~ABCDE~~: BCDE è ancora superchiave, eliminiamo A

~~AB~~CDE: CDE è ancora superchiave, eliminiamo B

~~ABC~~DE: DE non è superchiave, teniamo C

~~ABC~~~~D~~E: CE non è superchiave, teniamo D

~~ABC~~~~D~~E: CD è superchiave e nessun attributo può essere più eliminato, per cui CD è chiave

Soluzione alternativa: osserviamo che CD non appaiono mai a destra di alcuna dipendenza, e quindi appartengono ad ogni chiave. Osserviamo poi che  $CD^+ = ABCDE$ , quindi CD è chiave.

b) Sapendo che l'insieme degli attributi primi è  $\{ C, D \}$ , verificare quali forme normali sono rispettate dallo schema.

Consideriamo  $CE \rightarrow A$ : CE non è superchiave e A non è primo, quindi lo schema non è in 3NF e quindi neppure in BCNF.

c) La dipendenza  $BD \rightarrow A$  sarebbe ridondante rispetto ad F? Se aggiungessimo ad F la dipendenza  $BCD \rightarrow A$ , questa conterrebbe attributi estranei?

La chiusura di BD è uguale a BDE, per cui  $BD \rightarrow A$  non sarebbe ridondante.

Per ciò che riguarda  $BCD \rightarrow A$ , CD determina A per cui B sarebbe estraneo. Anche BC determina A, per cui anche A sarebbe estraneo. D'altra parte, la dipendenza stessa sarebbe ridondante.

d) Applicare l'algoritmo di sintesi e dire se dati e dipendenze sono stati preservati.

Raggruppiamo le dipendenze e definiamo le relazioni:

$CE \rightarrow A, CE \rightarrow B$  R1(ABCE)

$D \rightarrow E$  R2(DE)

$CB \rightarrow E$  R3(BCE)

Eliminiamo le relazioni contenute in altre: R1(ABCE), R2(DE)

Verifichiamo se una relazione è superchiave:  $ABCE^+ = ABCE$ ,  $DE^+ = DE$

Nessuna relazione è superchiave, aggiungiamo R3(CD), per cui la decomposizione è:

$\{ R1(ABCE), R2(DE), R3(CD) \}$

e) Applicare l'algoritmo di analisi e dire se dati e dipendenze sono stati preservati.

Consideriamo  $CE \rightarrow A$ .  $CE^+ = CEAB$ , per cui decomponiamo:

R1(CEAB), R2(CED)

Proiettiamo le dipendenze:

R1 < CEAB, { CE → A, CB → E, CE → B } >

R2 < CED, { D → E } >

CE<sup>+</sup>=CEAB e CB<sup>+</sup>=CBEA, per cui R1 è in BCNF

D<sup>+</sup>=DE, per cui R2 va ancora decomposta:

R2 < CED, { D → E } > -> R3 < DE, { D → E } >, R4 < DC, {} >

La decomposizione è quindi: { R1(CBEA), R3(DE), R4(DC) }.

La decomposizione preserva dati e dipendenze ed è in realtà la stessa prodotta dall'algoritmo di sintesi.

- 2) Si consideri il seguente schema relazionale R(IdR,A,B), S(IdS,IdR\*, IdT\*,C,D), T(IdT,IdR\*,E) e la seguente interrogazione

```

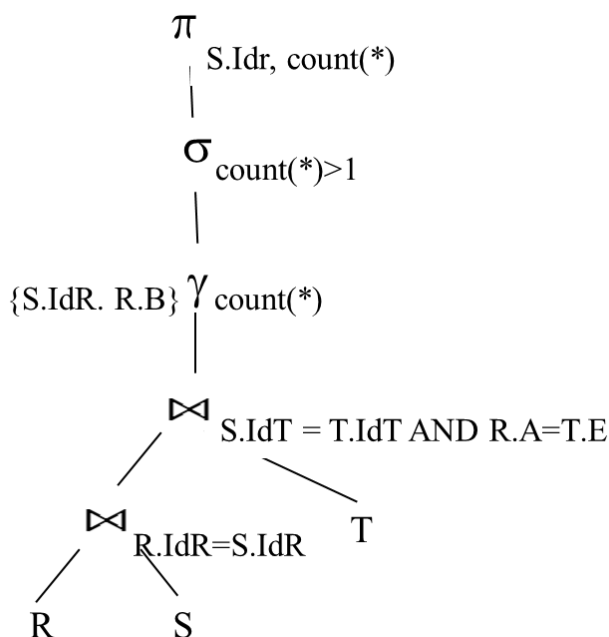
SELECT DISTINCT  S.IdR, count(*)
FROM             R, S, T
WHERE           R.IdR = S.IdR and S.IdT=T.IdT and R.A = T.E
GROUP BY       S.IdR, R.B
HAVING         count(*) > 1
    
```

- a) Opzionale: La clausola DISTINCT è ridondante?

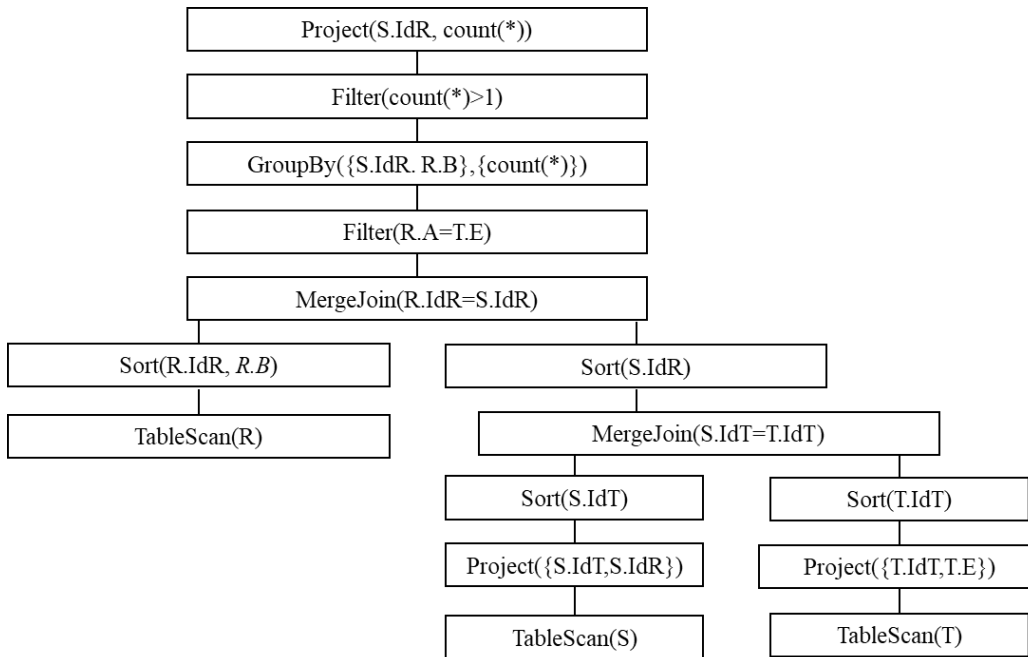
La clausola DISTINCT è ridondante: se due righe coincidono su S.IdR esse coincidevano anche su R.IdR, per via della condizione di giunzione, e quindi anche su R.B. Pertanto, non è possibile che due righe abbiano lo stesso valore di S.IdR e valori diversi di R.B, quindi S.IdR è una chiave per il risultato della GROUP BY.

(Attenzione: nel compito 2 invece la clausola DISTINCT non è ridondante!)

- b) Si scriva un piano di accesso logico



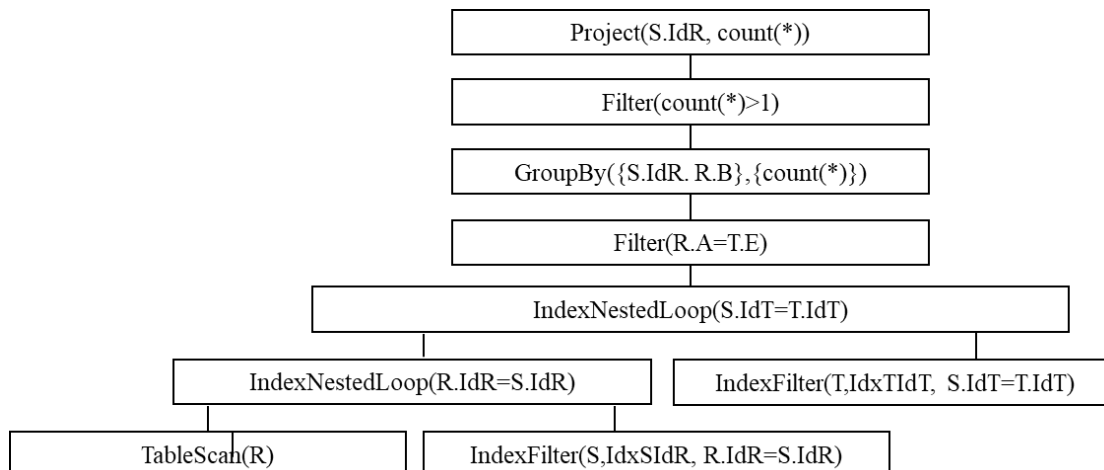
- c) Si scriva un piano di accesso fisico efficiente che non fa uso di indici, e (opzionale) verificare se la sort prima della Group By può essere evitata



Con questo piano la sort prima della Group By può essere evitata, i dati arrivano già ordinati su R.IdR e R.B.

In realtà, dato che IdR determina R.B, la Sort(R.IdR, R.B) può essere riscritta come Sort(R.IdR) (attenzione, questa osservazione non sarebbe vera nella soluzione del compito 2!)

- d) Si scriva un piano di accesso fisico efficiente che fa uso di due indici, e (opzionale) verificare se la sort prima della Group By può essere evitata



I dati sono raggruppati su IdR, e quindi su (IdR, R.B), all'uscita della TableScan dato che IdR è chiave per R, e restano tali attraverso le due INL e la Filter, per la Sort prima della GroupBy non serve. (Attenzione: questa linea di ragionamento non sarebbe valida nel compito 2!)

- 4) Si consideri un sistema gestito con protocollo disfare-rifare, si supponga che avvenga un fallimento di sistema, e si assuma che al momento della ripartenza questo sia il contenuto del log, dove un record (W,T1,A,1,10) sta a indicare una scrittura (Write) della transazione T1 sulla

variabile A con vecchio valore 1 e con nuovo valore 10:

(begin,T1) (W,T1,A,0,20) (begin,T2) (W,T2,B,0,30) (begin-ckp,{T1,T2}) (W,T2,B,30,50) (W,T1,C,0,30)  
(end-ckp) (begin,T3) (commit,T2) (W,T3,B,50,70) (begin,T4) (W,T4,D,0,50) (commit,T4)

a) Al momento in cui il sistema aveva iniziato a prendere il log, qual era il valore di A, B, C, D?  
A=0, B=0, C=0, D=0

b) Qual era il valore di A, B, C, D nel buffer al momento del checkpoint

Alla fine del checkpoint: A=20, B=50, C=30, D indefinito o 0

c) Al termine del checkpoint qual è il valore di A, B, C, D sul disco?

Alla fine del checkpoint: A=20, B=30 o 50, C=0 o 30, D=0

d) Quali sono le transazioni che erano riuscite a dare il commit prima del fallimento di sistema e quali no?

Hanno dato il commit: T2, T4. Non lo hanno dato: T1, T3

e) Al momento del fallimento di sistema qual è il valore di A, B, C, D nel buffer?

Nel buffer: A=20, B=70, C=30, D=50

f) Cosa possiamo dire riguardo al valore di A, B, C, D sul disco al momento del fallimento di sistema?

A=20, B=30 o 50 o 70, C=0 o 30, D=0 o 50

g) Quali sono le transazioni che al momento della ripartenza dovranno essere rifatte e quali sono quelle che dovranno essere disfatte?

Da rifare: T2, T4. Da disfare: T1, T3

h) Si elenchino tutte le operazioni che devono essere rifatte, nell'ordine in cui saranno rifatte

(W,T2,B,30,50) (W,T4,D,0,50)

i) Si elenchino tutte le operazioni che devono essere disfatte, nell'ordine in cui saranno disfatte

(W,T3,B,50,70) (W,T1,C,0,30) (W,T1,A,0,20)