BD: seconda prova di verifica del 19/12/2011 — Soluzioni

1. Si consideri lo schema relazionale

   R<(A, B, C, D), {B $\rightarrow$ CD, C $\rightarrow$ D, B $\rightarrow$ C, BC $\rightarrow$ D, BD $\rightarrow$ A}>.

   (a) Sia F un insieme di dipendenze funzionali e $X \rightarrow Y \in F$. Come si controlla se $A \in X$ è estraneo, e se $X \rightarrow Y$ è ridondante?

   $A \in X$ è estraneo sse $F \vdash (X - \{A\}) \rightarrow Y$
   $X \rightarrow Y$ è ridondante sse $F - \{X \rightarrow Y\} \vdash X \rightarrow Y$

   (b) Si porti l'insieme delle dipendenze funzionali di R in forma canonica.

   Elimino attributi estranei: B in $BC \rightarrow D$, D in $BD \rightarrow A$
   $F = \{$B $\rightarrow$ C, B $\rightarrow$ D, C $\rightarrow$ D, B $\rightarrow$ A$\}$
   Elimino dipendenze ridondanti: $B \rightarrow D$
   $F = \{$B $\rightarrow$ C, C $\rightarrow$ D, B $\rightarrow$ A$\}$

   (c) Si trovino le chiavi di R.

   B non sta mai a destra e $B^+ = BCDA$, quindi la sola chiave è B.

   (d) Dire se lo schema è in 3FN o in FNBC.

   Non è in 3NF, quindi neppure in BCNF.

   (e) Si applichi allo schema l'algoritmo di sintesi per ottenere una decomposizione in 3FN che preserva i dati.

   Algoritmo di sintesi produce: $R1(B, C, A), R2(C, D)$

   (f) Gli schemi relazionali della decomposizione del passo precedente sono in FNBC?

   Proietto le dipendenze: $\pi_{BCA}(F) = \{B \rightarrow CA\}; \pi_{CD}(F) = \{C \rightarrow D\}$. *I due schemi sono in BCNF.*

   (g) Si applichi l'algoritmo di analisi allo schema R(A, B, C, D), con le dipendenze in forma canonica, e si dica se il risultato preserva dati e dipendenze.

   Risultato algoritmo di analisi, partendo da $B \rightarrow C$, o da $B \rightarrow A$:
   R1<(B, C, A), {B $\rightarrow$ CA}>, R2<(B, D), {B $\rightarrow$ D}>.
   La dipendenza $C \rightarrow D$ è andata perduta.

   Risultato algoritmo di analisi, partendo da $C \rightarrow D$:
   R1<(C, D), {C $\rightarrow$ D}>, R2<(A, B, C), {B $\rightarrow$ A, B $\rightarrow$ C}>.
   Le dipendenze sono preservate.

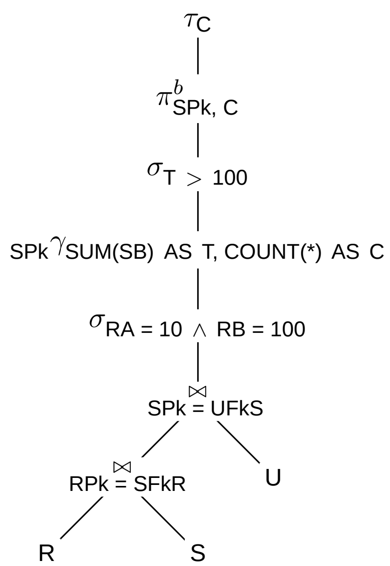
2. Si consideri lo schema relazionale:

R(RPk: int, RA: int, RB: int)
S(SPk: int, SA: string, SB: int, SFkR: int)
U(UFkS: int, UA: string)

Si supponga che la relazione S sia *ordinata* sulla chiave primaria SPk. Si consideri l'interrogazione

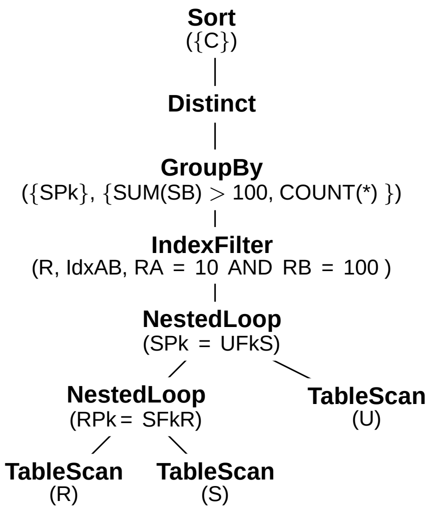
```
SELECT      DISTINCT SPk, COUNT(*) AS C
FROM        R, S, U
WHERE       RPk = SFkR AND SPk = UFkS
            AND RA = 10 AND RB = 100
GROUP BY    SPk
HAVING      SUM(SB) > 100
ORDER BY    C;
```

(a) Disegnare l'albero di sintassi astratta di un'espressione algebrica (albero logico) per l'interrogazione.



Tipo risultato: {{(SPk :int, C :int)}}

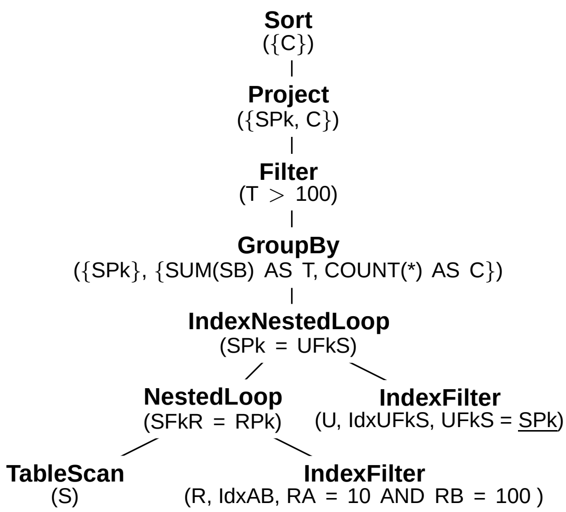
(b) Si dica se il seguente piano d'accesso è corretto e produce il risultato cercato. Se non va bene si dica perché e lo si renda corretto facendo solo le modifiche necessarie.



- *È scorretto l'uso di un operatore IndexFilter che abbia un altro operatore come parametro, l'operatore IndexFilter cerca i dati in memoria persistente per cui può apparire nell'albero solo come foglia. Tale operatore va qui sostituito con Filter.*
- *L'operatore GroupBy non può avere un confronto come parametro (SUM(SB) >100); tale confronto deve essere effettuato con un operatore Filter; inoltre, l'operatore deve ridenominare COUNT(*) come C, usata dall'operatore Sort.*
- *Occorre un Sort(SPk) prima dell'operatore GroupBy.*
- *L'operatore Distinct è ridondante. Occorre un Project(SPk, C)*

(c) Si riscriva il piano usando indici a vostra scelta.

Si assume un indice sugli attributi (RA, RB) di R, e sulle chiavi primarie ed esterne.



(d) Si dica che cosa è un indice e perché i DBMS ne prevedono l'uso.

Un indice su di un attributo A di una tabella R è una tabella ordinata $I(A, \mathrm{RID})$ con le seguenti proprietà: (a) i record di I hanno campi con valori (k_i, r_j), con k_i un valore di A in $\pi_A(R)$ e r_j un riferimento al record di R con $A = k_i$; (b) I record di I son ordinati sui valori di A; (c) $|I| = |R|$.

Un indice su A è utilizzato per accedere rapidamente ai record di R che soddisfano una condizione $A = k$ oppure $k_1 \leq A \leq k_2$. Un indice può essere anche multiattributo.