BD, Seconda prova di verifica, del 20/12/2013

1. Si consideri il seguente schema

   R(ABCDE,{AC→D, CE→A, ABC→E, CD→E})

   (a) Si portino le dipendenze in forma canonica

   (b) Si trovino tutte le chiavi.

   (c) Si dica se l'insieme di dipendenze rispetta o meno la terza forma normale e la BCNF.

   (d) Si applichi allo schema l'algoritmo di sintesi, e si dica se la decomposizione risultante è in FNBC.

2. Si consideri lo schema relazionale:

   Studenti(<u>IdStudente</u>, Nome, Cognome, AnnoNascita, Sesso)
   Iscrizioni(IdStudente*, IdCorso*, DataIscrizione)
   Corsi(<u>IdCorso</u>, Materia, AnnoAccademico)

   Si consideri la seguente interrogazione

   ```
   SELECT    DISTINCT s.Cognome, c.AnnoAccademico, COUNT(*)
   FROM      Studenti s, Iscrizioni i, Corsi c
   WHERE     s.IdStudente = i.IdStudente  AND  i.IdCorso = c.IdCorso
             AND  i.DataIscrizione > '1/1/2012'
   GROUP BY  s.IdStudente, s.Cognome, c.AnnoAccademico
   HAVING    COUNT(*) > 2;
   ```

   (a) Si disegni un albero logico per tale interrogazione.

   (b) Si disegni un albero fisico efficiente per la stessa interrogazione che non faccia uso di indici. Si assuma che Studenti sia ordinata rispetto a IdStudente.

   (c) Si disegni un albero fisico efficiente per la stessa interrogazione che acceda a tutte e tre le tabelle usando l'operatore **IndexFilter**.

   Si assuma ora di sostiture la prima riga dell'interrogazione con una delle due scelte seguenti. Dire, per ciascuna delle due, se la clausola **DISTINCT** ha qualche effetto oppure può essere rimossa senza modificare il risultato. Si osservi che andiamo a modificare solo gli attributi proiettati, la clausola di raggruppamento rimane uguale.

   (a) **SELECT DISTINCT** s.Idstudente, c.AnnoAccademico, COUNT(*)
   FROM . . . (vedi sopra)

   (b) **SELECT DISTINCT** s.Idstudente, s.Cognome, COUNT(*)
   FROM . . . (vedi sopra)

3. Riportare la definizione delle proprietà delle transazioni.

4. Si riporti la definizione di $F \vDash X \rightarrow Y$. Poi si dimostri che la seguente affermazione è *falsa*, senza usare il teorema di correttezza e completezza degli assiomi di Armstrong, ma andando semplicemente a negare la definizione appena riportata con un esempio.

$$\{AB \rightarrow C\} \vDash A \rightarrow C$$

5. (Opzionale) Si dimostri la seguente affermazione, usando la definizione di $F \vdash X \rightarrow Y$ oppure un qualunque teorema o algoritmo visto a lezione:

$$\{(X \setminus \{A\}) \rightarrow Y\} \vdash X \rightarrow Y$$

BD, Seconda prova di verifica, del 20/12/2013: soluzioni

1. Si consideri il seguente schema

   R(ABCDE,{AC→D, CE→A, ABC→E, CD→E})

   (a) Si portino le dipendenze in forma canonica

   Per ogni dipendenza, calcolo la chiusura dei sottoinsiemi del determinante per verificare la presenza di attributi estranei.

   AC→D: C^+_F = C; A^+_F = A.

   CE→A: E^+_F = E; C^+_F = C.

   ABC→E: BC^+_F = BC; AC^+_F = ACDE: l'attributo B è estraneo, sostituisco ABC→E con AC→E; A^+_F = A.

   CD→E: D^+_F = D; C^+_F = C.

   Lo schema privo di attributi estranei è quindi

   AC→D, CE→A, AC→E, CD→E

   Verifico ora la presenza di dipendenze ridondanti, calcolando, ad esempio, per la prima:

   $AC^*_{F \setminus \{AC \to D\}}$ = ACE

   La sola dipendenza ridondante è

   AC→E

   lo schema risultante è quindi:

   R(ABCDE,{AC→D, CE→A, CD→E})

   (b) Si trovino tutte le chiavi.

   Gli attributi B e C non sono a destra di alcuna dipendenza, per cui appartengono a tutte le chiavi. Gli attributi {A, D, E} si trovano sia a sinistra che a destra per cui un qualunque loro sottoinsieme potrebbe appartenere ad una chiave.

   BC^+_F = BC, per cui BC non è chiave. Considero quindi BCA, BCD e BCE.

   BCA^+_F = BCADE, per cui BCA è chiave. BCD^+_F = BCDEA, per cui BCD è chiave. BCE^+_F = BCEAD, per cui BCE è chiave.

   Non ci sono altre chiavi.

   (c) Si dica se l'insieme di dipendenze rispetta o meno la terza forma normale e la BCNF.

   Tutti gli attributi sono primi, per cui lo schema rispetta la 3NF.

   Nessuna dipendenza ha una superchiave a sinistra, per cui lo schema non è in BCNF.

   (d) Si applichi allo schema l'algoritmo di sintesi e si dica se la decomposizione risultante preserva è in FNBC

   Raggruppiamo le dipendenze F e costruiamo le relazioni corrispondenti:

AC $\rightarrow$ D	R1(ACD, $\{$AC $\rightarrow$ D$\}$)
CE $\rightarrow$ A	R2(ACE, $\{$CE $\rightarrow$ A$\}$)
CD $\rightarrow$ E	R3(CDE, $\{$CD $\rightarrow$ E$\}$)

Nessuna tabella è contenuta in un'altra e si aggiunge una relazione con una chiave di R

R1(ACD, $\{$AC $\rightarrow$ D$\}$)
R2(ACE, $\{$CE $\rightarrow$ A$\}$)
R3(CDE, $\{$CD $\rightarrow$ E$\}$)
R4(BCE, $\{\ \}$)

Per stabilire se la decomposizione risultante preserva è in FNBC si proiettano le dipendenze F su ognuno dei cinque sottoschemi.

R1(ACD, $\{$AC $\rightarrow$ D, CD $\rightarrow$ A$\}$)
R2(ACE, $\{$CE $\rightarrow$ A, AC $\rightarrow$ E$\}$)
R3(CDE, $\{$CD $\rightarrow$ E, CE $\rightarrow$ D$\}$)
R4(BCE, $\{\ \}$)

La decomposizione è in FNBC perché i determinanti delle DF delle relazioni $R1$, $R2$ e $R3$ sono chiavi. Possiamo quindi concludere che la decomposizione prodotta dall'algoritmo di sintesi è in BCNF, oltre a preservare, per costruzione, dati e dipendenze.

2. Si consideri lo schema relazionale:

Studenti(IdStudente, Nome, Cognome, AnnoNascita, Sesso)
Iscrizioni(IdStudente*, IdCorso*, DataIscrizione)
Corsi(IdCorso, Materia, AnnoAccademico)

Si consideri la seguente interrogazione

```
SELECT    DISTINCT s.Cognome, c.AnnoAccademico, COUNT(*)
FROM      Studenti s, Iscrizioni i, Corsi c
WHERE     s.IdStudente = i.IdStudente  AND  i.IdCorso = c.IdCorso
           AND  i.DataIscrizione > '1/1/2012'
GROUP BY  s.IdStudente, s.Cognome, c.AnnoAccademico
HAVING    COUNT(*) > 2;
```

(a) Si disegni un albero logico per tale interrogazione.

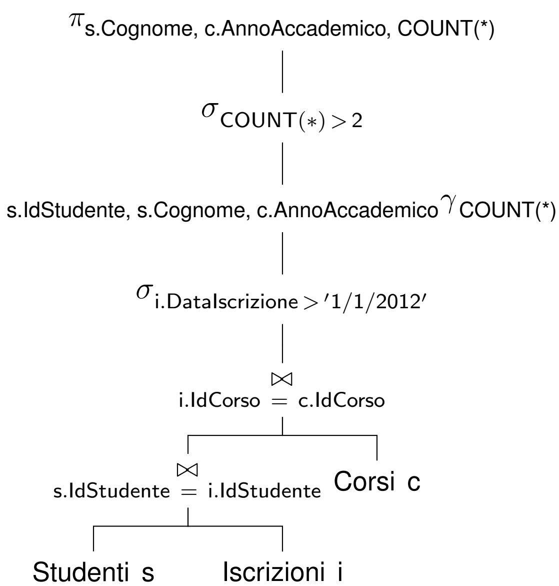

Figura 1: Albero logico

(b) Si disegni un albero fisico efficiente per la stessa interrogazione che non faccia uso di indici. Si assuma che Studenti sia ordinata rispetto a IdStudente.

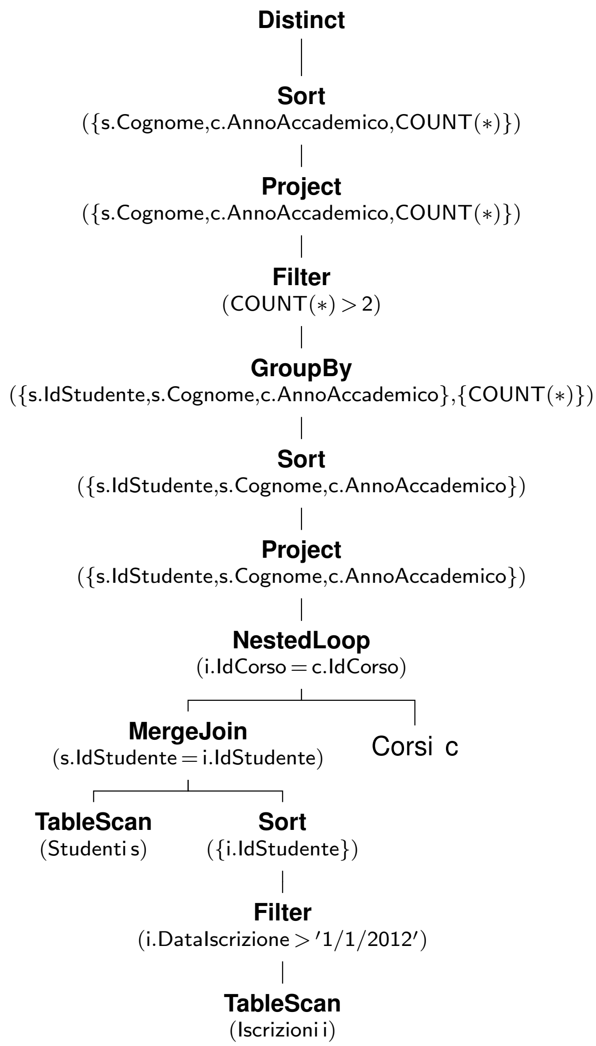

Figura 2: Albero fisico senza indici

(c) Si disegni un albero fisico efficiente per la stessa interrogazione che acceda a tutte e tre le tabelle usando l'operatore **IndexFilter**.

Si assuma ora di sostituire la prima riga dell'interrogazione con una delle due scelte seguenti. Dire, per ciascuna delle due, se la clausola **DISTINCT** ha qualche effetto oppure può essere rimossa senza modificare il risultato. Si osservi che andiamo a modificare solo gli attributi proiettati, la clausola di raggruppamento rimane uguale.

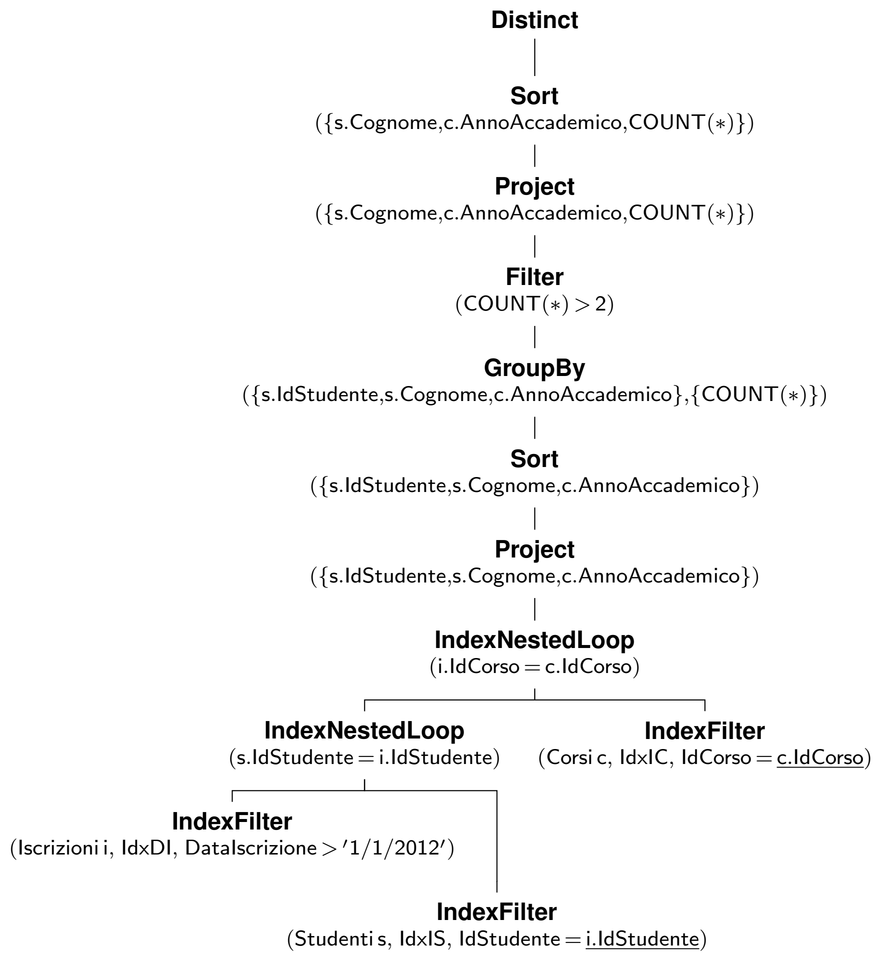

Figura 3: Albero fisico senza indici

Consideriamo una query

SELECT DISTINCT A_1, ..., A_n, f(B_1), ..., f(B_n)

...

GROUP BY A_1, ..., A_n, C_1,...C_m

...

dove f(B_1), ..., f(B_n) sono funzioni di aggregazione e C_1,...C_m sono gli attributi di raggruppamente non presenti nella clausola **SELECT**. Dato che il risultato del raggruppamento ha A_1, ..., A_n, C_1,...C_m come superchiave, la **DISTINCT** è ridondante se, e solo se, A_1, ..., A_n

determina funzionalmente C_1,...C_m.[1]

(a) **SELECT DISTINCT** s.Idstudente, c.AnnoAccademico, COUNT(*)
FROM ... (vedi sopra)

Dato che

$$\big(\text{s.IdStudente, c.AnnoAccademico}\big)$$

determina funzionalmente

$$\big(\text{s.Cognome}\big)$$

non possono prodursi duplicati, per cui la clausola **DISTINCT** è ridondante.

(b) **SELECT DISTINCT** s.Idstudente, s.Cognome, COUNT(*)
FROM ... (vedi sopra)

Dato che

$$\big(\text{s.IdStudente, s.Cognome}\big)$$

non determina funzionalmente

$$\big(\text{c.AnnoAccademico}\big)$$

possono prodursi duplicati (in dettaglio, nel caso in cui uno studente si sia iscritto allo stesso numero di corsi in anni accademici diversi, il fatto di aver cancellato l'anno accademico produce la duplicazione). Quindi, la clausola **DISTINCT** non è ridondante.

3. Riportare la definizione delle proprietà delle transazioni (vedi libro).

4. Si riporti la definizione di $F \models X \to Y$. Poi si dimostri che la seguente affermazione è *falsa*, senza usare il teorema di correttezza e completezza degli assiomi di Armstrong, ma andando semplicemente a negare la definizione appena riportata con un esempio.

$$\{AB \to C\} \models A \to C$$

Per definizione, $F \models X \to Y \Leftrightarrow \forall r.\ r \models F \Rightarrow r \models X \to Y$, ovvero
$F \not\models X \to Y \Leftrightarrow \exists r.\ r \models F \wedge r \not\models X \to Y$.

Dobbiamo quindi dimostrare che $\exists r.\ r \models AB \to C \wedge r \not\models A \to C$. Allo scopo, basta considerare la seguente tabella r:

[1] Per verificare se A_1, ..., A_n determina C_1,...C_m bisogna tenere conto non solo dei vincoli di chiave espressi nello schema ma anche delle uguaglianze imposte nella clausola di giunzione: un'uguaglianza a.pk=b.fk, ad esempio, genera le dipendenze a.pk$\to$b.fk e b.pk$\to$a.fk.

A	B	C
1	1	1
1	2	2

5. (Opzionale) Si dimostri la seguente affermazione, usando la definizione di $F \vdash X \rightarrow Y$ oppure un qualunque teorema o algoritmo visto a lezione:

$$\{(X \setminus \{A\}) \rightarrow Y\} \vdash X \rightarrow Y$$

$X \rightarrow (X \setminus \{A\})$ per riflessività. La tesi $X \rightarrow Y$ segue da quindi da $X \rightarrow (X \setminus \{A\})$ e dall'ipotesi $(X \setminus \{A\}) \rightarrow Y$ per transitività.