

Cognome Nome Matricola

Sistemi Informativi

07-09-2017

Sbarramento							
1 (4pt)	2 (4pt)	3 (5pt)	4 (6pt)	5 (4pt)	6 (5pt)	7 (5pt)	Somma (33pt)

Il compito ha la durata di 3 ore, per la compilazione attenersi alle seguenti istruzioni:

- Nome e cognome devono essere scritti in **STAMPATELLO**.
- É sola responsabilità dello studente scrivere in modo **LEGGIBILE**.
- Solo le soluzioni scritte all'interno dei riquadri verranno corrette.
- Non sono ammessi appunti o altro materiale, a chiunque copia verrà ritirato l'esame.
- Affinché l'esame venga corretto e quindi valutato, negli esercizi marchiat **con **** si dovrà raggiungere almeno 5 punti come somma totale.
- L'esame si ritiene superato se si raggiunge il punteggio di 18. La lode si ottiene con punteggio ≥ 31 .
- Qualsiasi soluzione ragionevole è accettata ma verranno premiate soluzioni complete in termini di sintassi e valutata positivamente la padronanza dei concetti presentati durante il corso.
- Sono ammesse le seguenti abbreviazioni nell'XML schema e nel DTD:
 - `<e ...> = <!ELEMENT ... >` e `<xs:element ... >`
 - `<a ...> = <!ATTLIST ... >` e `<xs:attribute ... >`
 - `<ct ...> = <xs:complexType ... >`
 - `<st ...> = <xs:simpleType ... >`

Esercizio 1 (4 punti). Dato lo Schedule:

S: r1(A); r1(C); r2(C); w3(D); r1(A); r2(B); w1(C); w2(C); r3(A); r2(A); w3(A); w3(D)

1pt) ** Dimostrare se S è **conflict serializzabile**.

1pt) Aggiornare lo schedule inserendo le operazioni per shared, exclusive e upgradable lock.

2pt) Usare shared, exclusive, e upgradable locks. Descrivere cosa fa ogni parte di un scheduler 2PL con le prime 5 operazioni di S

Sia dato il seguente XML contenuto nel file “doc.xml”:

```
<?xml version="1.0" encoding="UTF-8"?>
<libro>
  <ricetta nome="R1"><!-- almeno una -->
    <tempo unita="minuti" durata="120" /><!-- secondi, minuti oppure ore, default minuti -->
    <difficolta>1</difficolta><!-- Da 1 a 5 -->
    <porzioni>12</porzioni>
    <r_ingredienti> <!-- almeno un ingrediente -->
      <r_ingrediente iid="F1" unita="kg">1</r_ingrediente> <!-- kg oppure lt, no default -->
      <r_ingrediente iid="F2" unita="kg">2</r_ingrediente>
      <r_ingrediente iid="L1" unita="kg">3</r_ingrediente>
      <r_ingrediente iid="A1" unita="kg">0.01</r_ingrediente>
    </r_ingredienti>
  </ricetta>
  <ricetta nome="B2" >
    <tempo unita="ore" durata="2" /><!-- secondi, minuti oppure ore, default minuti -->
    <difficolta>5</difficolta><!-- Da 1 a 5 -->
    <porzioni>1</porzioni>
    <r_ingredienti> <!-- almeno un ingrediente -->
      <r_ingrediente iid="F1" unita="kg">5</r_ingrediente> <!-- kg oppure lt, no default -->
      <r_ingrediente iid="F1" unita="kg">0.5</r_ingrediente>
    </r_ingredienti>
  </ricetta>
  <ingredienti>
    <farina id="F1">
      <tipo>0</tipo>
      <produttore>Produttore1</produttore>
    </farina>
    <farina id="F2">
      <tipo>00</tipo> <!-- 0, 00, oppure integrale -->
      <produttore>Produttore2</produttore>
      <origine>Scozia</origine>
    </farina>
    <lievito id="L1">
      <nome>Lievito1</nome>
      <produttore>Produttore3</produttore>
      <origine>Italia</origine>
    </lievito>
    <lievito id="L2">
      <nome>Lievito1</nome>
      <produttore>Produttore4</produttore>
    </lievito>
    <altro id="A1"><nome>Cannella</nome>
      <produttore>Produttore5</produttore>
      <origine>India</origine>
    </altro>
  </ingredienti></libro>
```

Esercizio 2 (4 punti). Rispondere alle seguenti domande

2pt) Scrivere un possibile schema DTD valido

2pt) Scrivere il frammento di un possibile XML schema degli elementi <Ingredienti>

Esercizio 3 (5 punti). Si consideri il contenuto XML precedente.

3pt) Scrivere l'interrogazione XQuery per restituire la ricetta (compreso tutto il tag) che ha meno ingredienti. `<ricetta>...</ricetta>`

2pt) Scrivere il risultato dell'esecuzione della seguente query (si assuma che `$doc` sia ben definito).

```
let $resp := (  
  for $t in $doc//tempo, $p in $doc//porzioni  
  where data($t/@durata) = ''120''  
  return <i>{$p}</i>  
)  
return <list>{$resp}</list>
```


Esercizio 4 (6 punti). Data la tabella con nome persona, colore preferito e città di provenienza.

2pt) ** Definisci in una frase il concetto di *Edit-Distance*. Calcolare Edit distance tra i nomi delle tuple x_1 , x_6 e x_7

CC	Nome	Colore	Città
x1	Alice	Rosso	TN
x2	Bob	Rosso	MI
x3	Carol	Blu	VR
x4	Bilbo	Giallo	TN
x5	Bobby	Verde	MI
x6	Carol	Blu	VR
x7	Alex	Rosso	TN

2pt) Applicare l'algoritmo di entity resolution utilizzando *Set-valued attributes*. Si considerano due tuple simili, se il valore della colonna *Nome* ha le stesse due lettere iniziali.

2pt) Considerare come funzione di distanza tra due tuple il numero di valori diversi (Es. tupla x_1 e x_2 distanza 3, tupla x_3 e x_6 distanza 1). Mostrare i passaggi di clustering gerarchico per ottenere 4 cluster. **Eslicitare eventuali assunzioni**

Esercizio 5 (4 punti). Considerare le seguenti 3 matrici.

2pt) ** Motivare se e quali di esse siano valide matrici di transizione per il calcolo del Page Rank.

M1

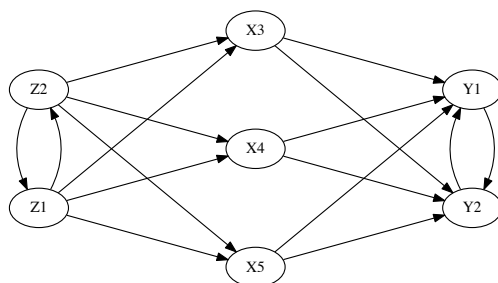
	x1	x2	x3	x4	x5
x1					1
x2				1	
x3			1		
x4		1			
x5	1				

M2

	x1	x2	x3	x4	x5
x1	1/5	1/3			
x2	1/5	1/3	1/3		
x3	1/5	1/3	1/3		
x4	1/5		1/3	2/3	1/2
x5	1/5			1/3	1/2

M3

	x1	x2	x3	x4	x5
x1	1/3			1/3	
x2		1/2			1/2
x3		1/2	1/2		
x4	1/3		1/2	1/3	1/3
x5	1/3				1/3



Teleport Set = {Y1, Y2}

2pt) Dato il grafo in figura, calcolare pagerank con $\beta = 0.8$ utilizzando le prime 2 iterazioni del metodo con relaxation

Esercizio 6 (5 punti). Si considerino 3 aziende Az1, Az2, Az3. Az1 vende prodotti di categoria “profumi” e “cosmetici”, Az2 vende “cosmetici” “biologici”, mentre Az3 “cibi” “biologici” . Ogni azienda ha un budget di 50\$ offerto dal motore di ricerca Altavista, e ogni *inserzione* costa 15\$.

Si consideri il seguente ordine di ricerche:

“biologici”; “cosmetici”, “profumi”; “biologici”; “cibi”; “profumi” ;“profumi” ; “cibi”;

2pt) ** Si trovi l’assegnamento ottimo mostrando brevemente perchè è ottimo.

3pt) Si trovi l’assegnamento dato dall’algoritmo *balance greedy* mostrando brevemente il processo

Esercizio 7 (5 punti). Sia dato un database per il noleggio di auto. Per ogni auto si ha marca, colore, e anno di immatricolazione. Per ogni cliente si ha professione, età e città. Si vuole analizzare per ogni cliente, l'auto noleggiata, la data, il numero di giorni, e costo totale.

2pt) ** Definire fact table e dimension tables per questo scenario.

3pt) Scrivere la query per definire il **formal data cube** con schema

`NCube(data, colore, marca, citta, numero_auto, somma_costo, massimo_costo, minimo_costo)`