

Matricola: _____

Cognome: _____

Nome: _____

Basi di Dati

Prova intermedia del 7 giugno 2016

SOLUZIONI

Durata 2h15min

DOMANDE TEORIA (è obbligatorio rispondere ad almeno una domanda delle quattro di seguito elencate)

- (3) Illustrare l'architettura di un DMBS descrivendo in particolare il modulo di gestione dei guasti (o gestore dell'affidabilità); si indichi inoltre, per ogni modulo dell'architettura, quali sono le proprietà delle transazioni che contribuisce a garantire.
- (3) Si presenti in dettaglio la politica di concessione dei lock applicata dal gestore dell'esecuzione concorrente secondo la tecnica detta "Locking a due fasi".
- (2) Lo studente illustri la struttura di accesso ai dati denominata indice primario sparso, si descrivano in particolare i seguenti punti: (i) le caratteristiche della struttura di accesso, (ii) l'algoritmo di ricerca di una tupla con chiave K usando l'indice.
- (2) Lo studente illustri le differenze tra i costrutti **elemento** e **attributo** del linguaggio XML, mostrando un esempio di documento XML dove vengono utilizzati entrambi.

ESERCIZI

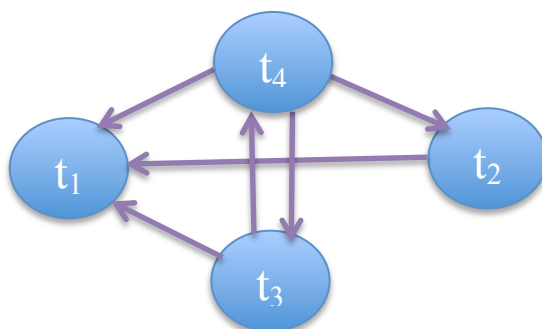
Esecuzione concorrente

Dato il seguente schedule S:

S: $r_4(t), w_2(t), r_1(t), r_4(y), r_3(y), w_4(y), w_4(z), w_3(z), w_1(t), w_2(x), w_1(z)$

- (2) indicare se è conflict-SR oppure no (calcolare l'insieme dei conflitti),

$conflitti(S) = \{ (r_4(t), w_2(t)), (r_4(t), w_1(t)), (w_2(t), r_1(t)), (w_2(t), w_1(t)), (r_3(y), w_4(y)), (w_4(z), w_3(z)), (w_4(z), w_1(z)), (w_3(z), w_1(z)) \}$



Poiché il grafo non è ACICLICO lo schedule non è CSR.

- (4) se non è CSR verificare se è view-SR oppure non-SR (giustificare dettagliatamente la risposta).

S: $r_4(t), w_2(t), r_1(t), r_4(y), r_3(y), w_4(y), w_4(z), w_3(z), w_1(t), w_2(x), w_1(z)$

$t_1 = r_1(t), w_1(t), w_1(z)$

$t_2 = w_2(t), w_2(x)$

$t_3 = r_3(y), w_3(z)$

$t_4 = r_4(t), r_4(y), w_4(y), w_4(z)$

$LeggeDa(S) = \{ (r_1(t), w_2(t)) \} \Rightarrow t_2 < t_1, t_4 < t_2, t_4 < t_1, t_3 < t_4$

$ScrittureFinali(S) = \{ w_1(z), w_2(x), w_1(t), w_4(y) \} \Rightarrow t_2 < t_1, t_3 < t_1, t_4 < t_1$

Schedule seriali da considerare:

t_3, t_4, t_2, t_1

S1: $r_3(y), w_3(z), r_4(t), r_4(y), w_4(y), w_4(z), w_2(t), w_2(x), r_1(t), w_1(t), w_1(z)$

$LeggeDa(S1) = \{ (r_1(t), w_2(t)) \}$

$ScrittureFinali(S1) = \{ w_1(z), w_1(t), w_2(x), w_4(y) \}$

Pertanto S è view-equivalente a S1 quindi è VSR.

Ottimizzazione

Si consideri il seguente schema relazionale contenente le visite svolte in un ospedale:

PAZIENTE(CodiceSSN, Nome, Cognome, Ntelefono, Indirizzo, Città);

VISITA(CodiceSSN, CFMedico, Data, Orainizio, Durata, PressioneArteriosaMin, PressioneArteriosaMax)

MEDICO(CFMedico, Cognome, Nome, Specialita)

Vincoli d'integrità referenziale: VISITA.CodiceSSN -> PAZIENTE, VISITA.CFMedico -> MEDICO

Formulare in SQL la seguente interrogazione:

(1) Trovare le visite eseguite da medici di specialità 'Cardiologia', riportando per ogni visita: il nome e il cognome del medico, il codice sanitario del paziente (CodiceSNN) e la data della visita.

```
SELECT V.CodiceSSN, V.Data, M.Nome, M.Cognome  
FROM VISITA V JOIN MEDICO C ON V.CFMedico = M.CFMedico  
WHERE M.Specialità = 'Cardiologia'
```

(3) Calcolare il costo dell'interrogazione in termini di numero di accessi a memoria secondaria sotto le seguenti ipotesi:

- la selezione dei medici di specialità 'Cardiologia' richiede una scansione sequenziale della tabella MEDICO
- l'ordine di esecuzione del join è MEDICO \bowtie VISITA
- le operazioni di join vengono eseguite con la tecnica "Nested Loop Join" con una pagina di buffer disponibile per ogni tabella
- NP(MEDICO) = 12, NP(VISITA) = 2000, NP(PAZIENTE) = 130
- NR(MEDICO) = 1200, NR(VISITA) = 240000, NR(PAZIENTE) = 12500
- VAL(Specialità, MEDICO) = 25, VAL(CFMedico, VISITA) = 1200

COSTO SELEZIONE su MEDICO = NP(MEDICO)

COSTO JOIN = NR(MEDICO con selezione Specialità = 'Cardiologia') * NP(VISITA) =
NR(MEDICO) / VAL(Specialità, MEDICO) * NP(VISITA)

Quindi:

COSTO SELEZIONE su MEDICO = 12

COSTO JOIN = 1200 / 25 * 2000 = 96000

Infine:

COSTO TOTALE = 96012

(2) Come cambia il costo se è disponibile un indice B⁺-tree sull'attributo CFMedico della tabella VISITA che ha profondità 3.

COSTO SELEZIONE su MEDICO = come al punto precedente

COSTO JOIN = NR(MEDICO con selezione Specialità = 'Cardiologia') *

(Profondità Indice + selettività di CFMedico in VISITA) =

NR(MEDICO) / VAL(Specialità, MEDICO) * (Profondità Indice + NR(VISITA) / VAL(CFMedico, VISITA))

Quindi:

COSTO SELEZIONE su MEDICO = 12

COSTO JOIN = 1200 / 25 * (3 + 240000 / 1200) = 48 * 203 = 9744

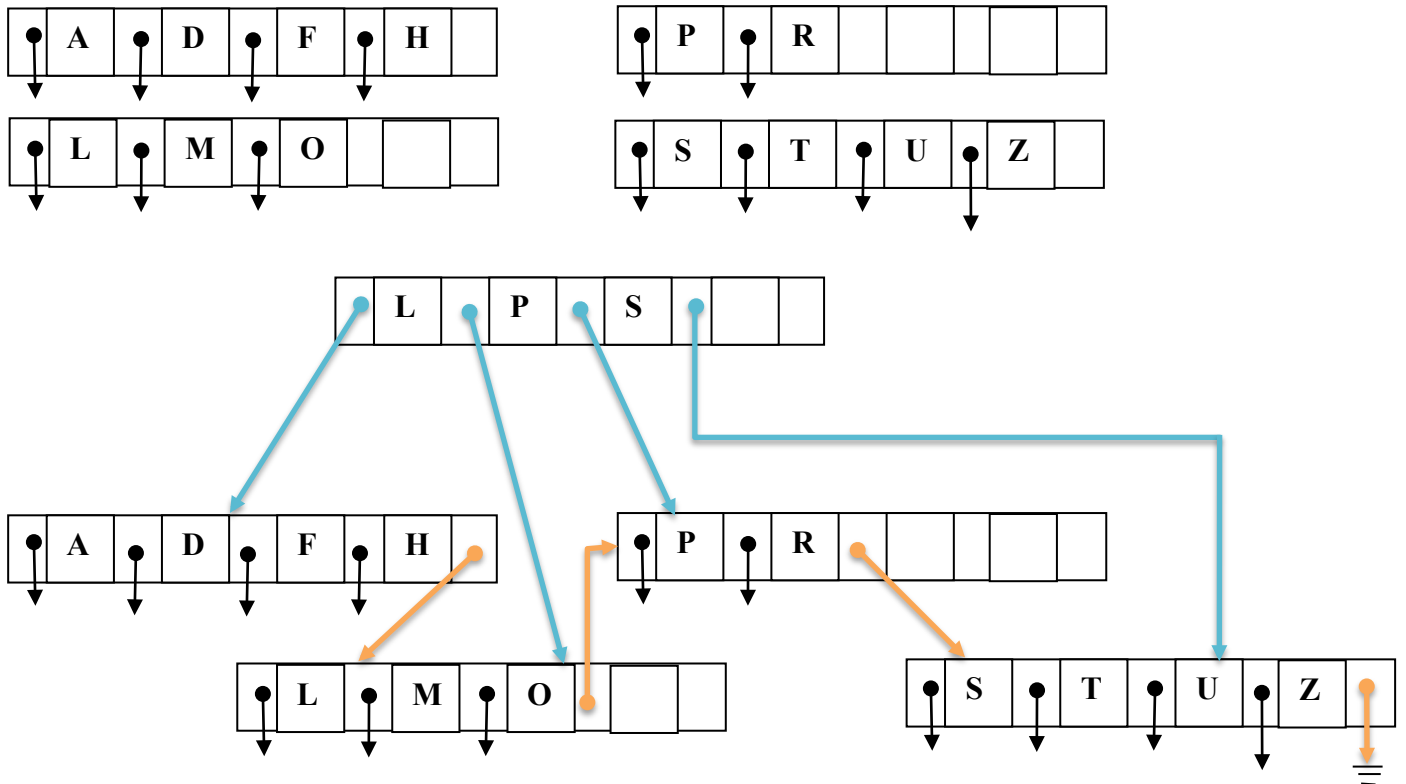
Infine:

COSTO TOTALE = 9756

B⁺-tree

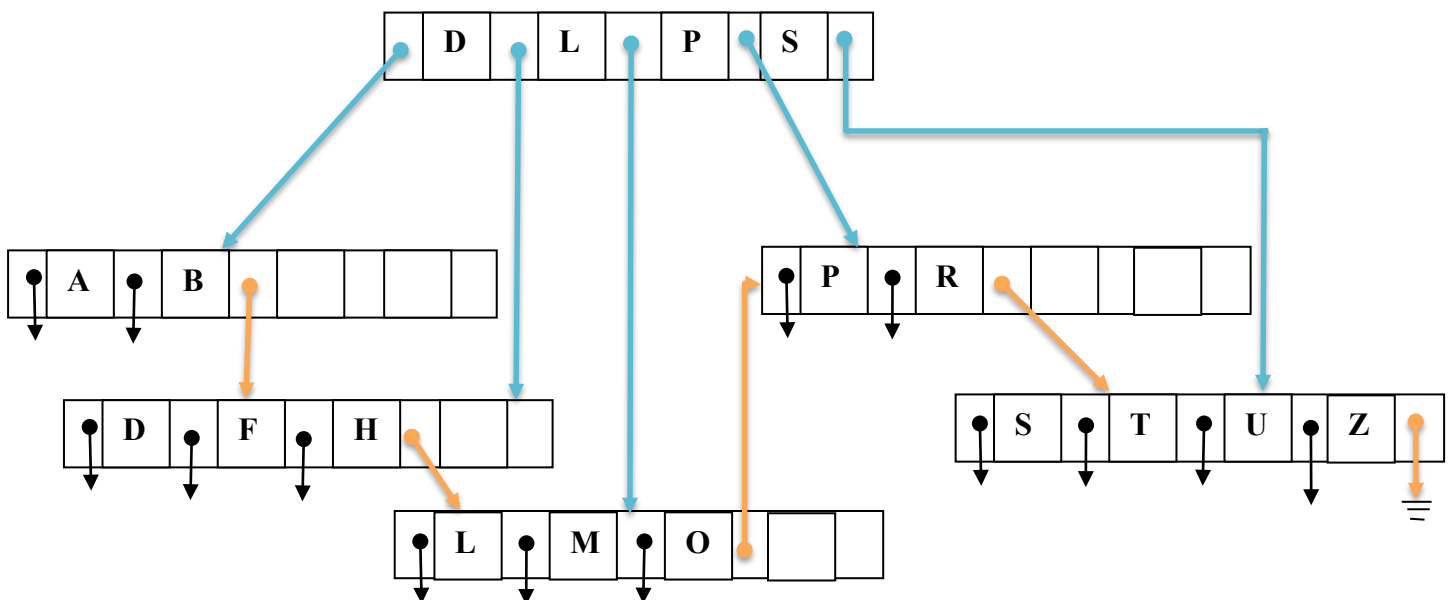
Data la seguente lista di possibili valori chiave $L = (A, B, C, D, E, F, G, H, I, L, M, N, O, P, Q, R, S, T, U, W, Z)$:

a) (2) costruire un B⁺-tree (fan-out=5) che contenga i seguenti nodi foglia:

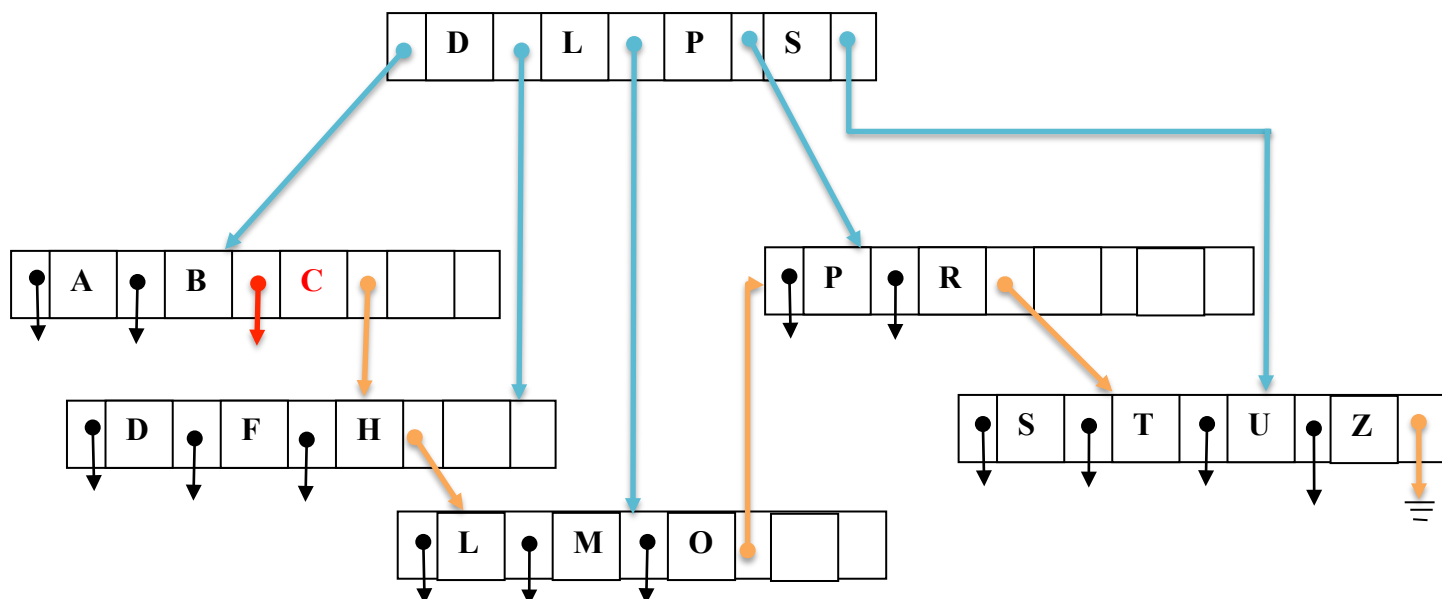


b) (4) mostrare l'albero dopo l'inserimento del valore chiave B, poi del valore C, poi di E, poi di Q.

Dopo l'inserimento di B: richiede lo split del primo nodo foglia



Dopo l'inserimento di C: non si richiede alcuno split in quanto C può essere inserito nel primo nodo foglia senza violare i vincoli di riempimento.



Lo stesso vale per l'inserimento di E e Q che non violano i vincoli di riempimento dei rispettivi nodi foglia.

XML

(6) Dato il seguente file XML e i seguenti requisiti si produca il file XML schema (XSD) che ne descrive la sintassi.

```
<?xml version="1.0"?>
<elezioni xmlns="http://www.proiezioniElettorali.it"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://www.proiezioniElettorali.it elezioni.xsd">
  <tornata anno="2016">
    <tipo>Amministrative</tipo>
  </tornata>
  <comune>
    <nome>Milano</nome>
    <proiezione timestamp=" 2016-06-06 T 09:45:22">
      <lista>
        <nome>LISTA A</nome> <voti>2345</voti>
      </lista>
      <lista>
        <nome>LISTA B</nome> <voti>245</voti>
      </lista>
    </proiezione>
    <proiezione timestamp=" 2016-06-06 T 11:05:13">
      <lista>
        <nome>LISTA A</nome> <voti>2845</voti>
      </lista>
      <lista>
        <nome>LISTA B</nome> <voti>845</voti>
      </lista>
    </proiezione>
  </comune>
  <comune>
    <nome>Roma</nome>
    <proiezione timestamp=" 2016-06-06 T 09:55:22">
      <lista>
        <nome>LISTA A</nome> <voti>1345</voti>
      </lista>
    </proiezione>
  </comune>
</elezioni>
```

Requisiti

L'attributo anno e l'attributo timestamp sono obbligatori. L'elemento tipo può assumere solo i valori: Amministrative, Politiche, Europee.

XMLSchema elezioni.xsd da completare

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.proiezioniElettorali.it"
  xmlns="http://www.proiezioniElettorali.it">
  <xsd:simpleType name="ST_TipoElezione">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Amministrative"/>
      <xsd:enumeration value="Politiche"/>
      <xsd:enumeration value="Europee"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:complexType name="CT_Tornata">
    <xsd:sequence>
      <xsd:element name="tipo" type="ST_TipoElezione"/>
    </xsd:sequence>
    <xsd:attribute name="anno" type="xsd:gYear" use="required"/>
  </xsd:complexType>
  <xsd:complexType name="CT_Proiezione">
    <xsd:sequence>
      <xsd:element name="lista" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="nome" type="xsd:string"/>
            <xsd:element name="voti" type="xsd:unsignedInt"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="timestamp" type="xsd:dateTime" use="required"/>
  </xsd:complexType>

  <xsd:element name="comune">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="nome" type="xsd:string"/>
        <xsd:element name="proiezione" type="CT_Proiezione"
          maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="elezioni">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="tornata" type="CT_Tornata"/>
        <xsd:element ref="comune" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```