

Esercitazione per la II prova intermedia 2013

DOMANDE di Teoria sui sistemi DBMS

- a) (3) Lo studente illustri la politica di concessione dei lock adottata dalla tecnica di locking a due fasi (2PL), mostrando anche un esempio di schedule 2PL (non seriale) e uno di schedule non 2PL.

Illustrare la politica del gestore dei lock presentando le strutture dati (stato della risorsa: $s(x)$) e le transazioni in r_lock : $c(x)$) e la tabella di concessione dei lock

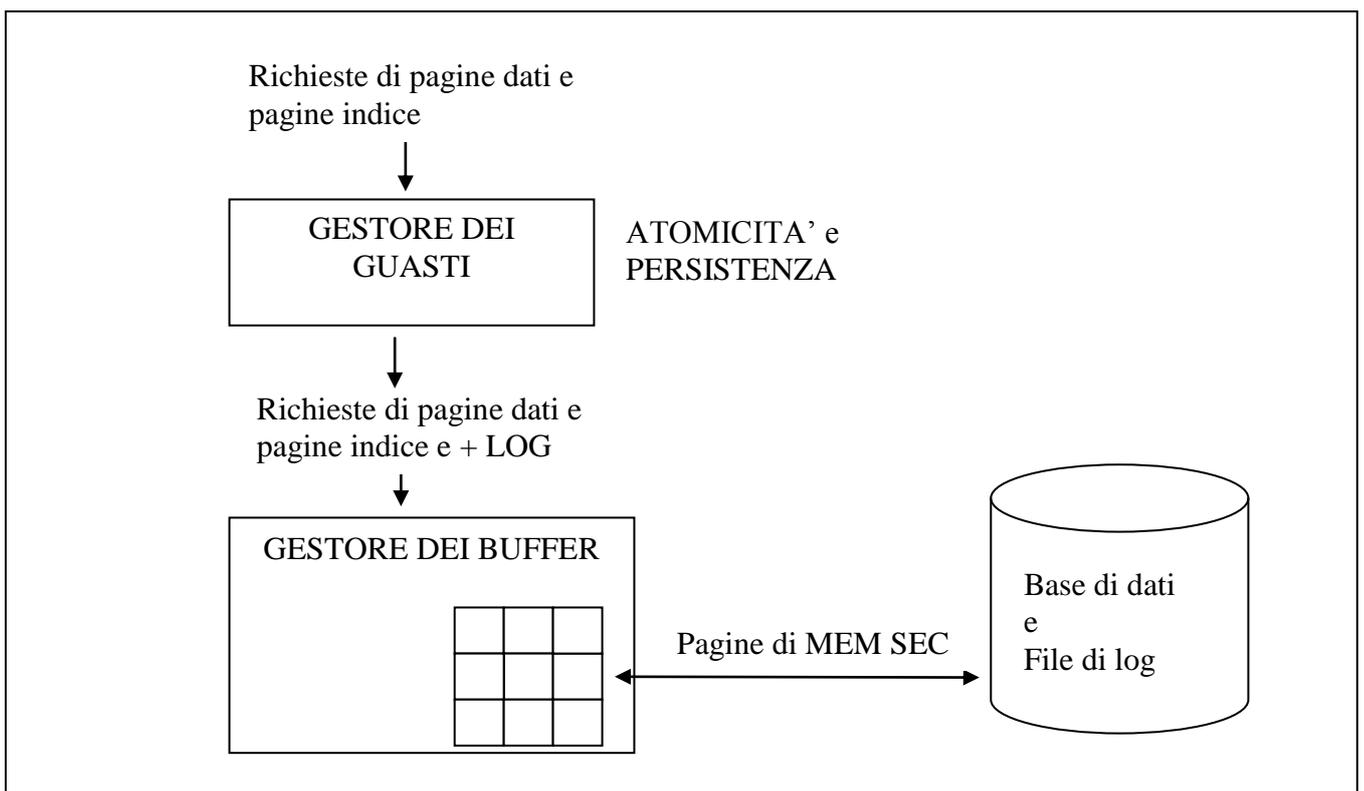
2PL non seriale

non 2PL

$r_1(x)w_1(x) r_2(y)w_2(y) r_1(y)w_1(y)$

$r_1(y)w_1(y) r_2(y)w_2(y) r_1(x)w_1(x)$

- b) (3) Illustrare nello spazio seguente come interagiscono tra loro i seguenti moduli di un DBMS: Gestore dell'affidabilità (o Gestore dei guasti) e Gestore dei buffer. Si indichino inoltre quali proprietà delle transazioni vengono garantite dal Gestore dell'affidabilità.



- c) (3) Lo studente illustri la struttura di un nodo intermedio di un indice B^+ -tree caratterizzato da un fan-out=N.

- d) (2) Lo studente descriva la struttura di un indice secondario costruito su una struttura sequenziale ordinata.

DOMANDE di TEORIA su Applicazioni Web e Dati Multimediali

- e) (4) Lo studente illustri le caratteristiche fondamentali della tecnologia Java Server Pages (JSP) e il suo ruolo nell'architettura MVC-2 servlet centric.

- f) (4) Lo studente illustri le caratteristiche della tecnica di compressione Huffman.

ESERCIZI sulle Strutture di accesso ai dati (indici) e su XML

1. (4) Si generi la struttura di un B+-tree con fan-out=4 e contenente 5 nodi foglia con i seguenti valori della chiave di ricerca: {A,B,C,L,M,N,O,P,Q,S,T,U,W,Z}
2. (3) Si mostri la struttura dell'albero ottenuto al primo esercizio dopo l'inserimento del valore E e dopo la cancellazione del valore L.
3. (6) Dato il seguente file XML e i seguenti requisiti si produca il file XML schema (XSD) che ne descrive la struttura.

XML

```
<?xml version="1.0"?>
<RubricaTelefonica xmlns="http://www.pagineBianche.org"
                    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                    xsi:schemaLocation="http://www.pagineBianche.org
                                        pagineBianche.xsd">

  <Utente id="UT00032">
    <tipo>privato</tipo>
    <numero>0458027777</numero>
    <indirizzo>
      <via civico="23">Lombardia</via>
      <città>Milano</città>
    </indirizzo>
  </Utente>
  <Utente id="UT00123">
    <tipo>attività commerciale</tipo>
    <numero>048372839</numero>
    <indirizzo>
      <piazza civico="8">Duomo</piazza>
      <città>Brescia</città>
    </indirizzo>
  </Utente>
  <Utente id="UT0003">
    <tipo>attività professionale</tipo>
    <numero>04832222839</numero>
    <indirizzo>
      <piazzale civico="2">Stazione</piazzale>
      <città>Brescia</città>
    </indirizzo>
  </Utente>
  ...
</RubricaTelefonica>
```

Requisiti

Il tipo può assumere solo uno dei seguenti valori: {privato, attività commerciale, attività professionale, servizio pubblico}. L'attributo id è obbligatorio. Supponiamo per semplicità che il civico sia sempre un intero positivo.

XMLSchema pagineBianche.xsd

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             targetNamespace="http://www.pagineBianche.org"
             xmlns="http://www.pagineBianche.org">
```

```

<xsd:element name="RubricaTelefonica">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="Utente" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:simpleType name="utenteTipo">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="privato"/>
    <xsd:enumeration value="attività commerciale"/>
    <xsd:enumeration value="attività professionale"/>
    <xsd:enumeration value="servizio pubblico"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="numeroTipo">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="\d{9}"/>
    <xsd:pattern value="\d{11}"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="viaTipo">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="civico" type="xsd:unsignedShort" use="required"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<xsd:element name="Utente">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="tipo" type="utenteTipo"/>
      <xsd:element name="numero" type="numeroTipo"/>
      <xsd:element name="indirizzo">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:choice>
              <xsd:element name="via" type="viaTipo"/>
              <xsd:element name="piazza" type="viaTipo"/>
              <xsd:element name="piazzale" type="viaTipo"/>
            </xsd:choice>
            <xsd:element name="città" type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:ID" use="required"/>
  </xsd:complexType>
</xsd:element>
</xsd:schema>

```

ESERCIZI su SQL

4. (6) Data la seguente base di dati "Tornei", contenente informazioni sulle partite di una serie di tornei di pallavolo:

SQUADRA (Codice, Nome, Provenienza, Sponsor*)

PARTECIPAZIONE(Squadra, Torneo, PuntiClassifica)

PARTITA (CodGara, Squadra1, Squadra2, Giornata, Torneo, Ora, Set1*, Set2*)

GIORNATA(Numero, Torneo, Data)

TORNEO(Nome, DataInizio, DataFine, Luogo, Sponsor*)

TORNEO.Luogo indica il luogo dove il torneo si svolge.

Vincoli di integrità referenziale:

PARTITA.Squadra1 → SQUADRA; PARTITA.Squadra2 → SQUADRA

PARTITA.(Giornata,Torneo) → GIORNATA

GIORNATA.Torneo → TORNEO

PARTECIPAZIONE.Squadra → SQUADRA; PARTECIPAZIONE.Torneo → TORNEO

Ulteriori vincoli di identificazione:

PARTITA.(Squadra1, Squadra2, Giornata,Torneo) è chiave candidata

Scrivere in SQL le seguenti interrogazioni:

- a. Trovare i tornei dove si svolgono nella stessa giornata almeno una partita prima delle 12.00 e almeno una partita dopo le 22.00 riportando il nome e la data d'inizio del torneo.

```
SELECT T.Nome, T.DataInizio
```

```
FROM Torneo T, Partita P1, Partita P2
```

```
WHERE T.Nome=P1.Torneo AND T.Nome=P2.Torneo AND T1.Giornata=P2.Giornata
```

```
AND P1.Ora < '12.00' AND P2.Ora > '22.00'
```

- b. Trovare il nome e la provenienza delle squadre che non hanno mai giocato tornei a Verona.

```
SELECT S.Nome, S.Provenienza
```

```
FROM Squadra S
```

```
WHERE NOT EXISTS (SELECT 1 FROM Partita P1 JOIN Torneo T ON P1.Torneo=T.Nome
```

```
WHERE (P1.Squadra1=S.Codice OR P1.Squadra2=S.Codice)
```

```
AND T.Luogo = 'Verona')
```

- c. Trovare per ogni torneo il numero di giornate e il numero di partite in cui è organizzato riportando il nome del torneo, il luogo dove si svolge e i conteggi richiesti.

```
SELECT T.Nome, T.Luogo,
```

```
count(DISTINCT P.Giornata) as NumGiornate, count(*) as NumPartite
```

```
FROM Torneo T JOIN Partita P ON T.Nome=P.Torneo
```

```
GROUP BY T.Nome, T.Luogo
```

- d. Trovare il nome e lo sponsor della squadra che nel torneo "T" non ha mai perso una partita.

```
SELECT S.Nome, S.Sponsor
```

```
FROM Squadra S
```

```
WHERE NOT EXISTS (SELECT 1 FROM Partita P
```

```
WHERE P.Torneo='T' AND
```

```
((P.Squadra1=S.Codice AND P.Set1 < P.Set2)
```

```
OR
```

```
(P.Squadra2=S.Codice AND P.Set2 < P.Set1))
```

ESERCIZI sui Dati Multimediali

5. (2,5) Si descrivano le caratteristiche strutturali di un k-d-tree per $k=2$. Lo studente illustri come scambia la struttura dell'indice per $k=3$.

6. (3,5) Data il seguente messaggio ZWYYWYYWWZWW lo si codifichi su 4 bit secondo l'algoritmo LZW con il seguente dizionario iniziale ($W=0000$, $Y=0001$, $Z=0010$). Si riporti anche il contenuto del dizionario dopo la codifica.