

Basi di dati II
Esame — 27 giugno 2013 — Compito A

Rispondere su questo fascicolo.

Tempo a disposizione: due ore e quarantacinque minuti (prova lunga), un'ora e trenta minuti (prova breve).

Cognome _____ **Nome** _____ **Matricola** _____

Domanda 1 (10% per la prova lunga, 20% per la prova breve)

Considerare il seguente DTD

```
<!ELEMENT league (team+)>
<!ELEMENT team ((city,state?) | (city,state?,name) | name) >
<!ELEMENT city (#PCDATA) >
<!ELEMENT state (#PCDATA) >
<!ELEMENT name (#PCDATA) >
```

che si vorrebbe usare per validare documenti come il seguente

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE league SYSTEM "league.dtd">
<league>
  <team><city>Boston</city><state>MA</state><name>Red Sox</name></team>
  <team><name>Yankees</name></team>
  <team><city>Detroit</city></team>
</league>
```

Spiegare perché il DTD non è corretto.

Correggere il DTD (mostrare solo la riga o le righe da modificare)

Basi di dati II — 27 giugno 2013 — compito A

Domanda 2 (15% per la prova lunga, 30% per la prova breve)

Si supponga siano di interesse documenti che contengono informazioni su gruppi montuosi e singole montagne, come il seguente:

```
<montagne>
  <gruppoMontuoso>
    <nome>Gran Sasso</nome>
    <provincia>L'Aquila</provincia>
    <provincia>Teramo</provincia>
    <cima>Corno Grande</cima>
  </gruppoMontuoso>
  <gruppoMontuoso>
    <nome>Monte Rosa</nome>
    <cima>Punta Dufour</cima>
    <cima>Punta Gnifetti</cima>
  </gruppoMontuoso>
  <montagna>Vesuvio</montagna>
  <montagna>Etna</montagna>
</montagne>
```

Considerare lo schema XSD seguente con cui si è tentato erroneamente di modellare tali documenti e individuare (annotandole sullo schema stesso) le modifiche che a tale schema che rendano valido rispetto ad esso il documento sopra mostrato:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="montagne">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="gruppoMontuoso" type="TipoGruppo" />
        <xsd:element name="montagna" type="xsd:string" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:complexType name="TipoGruppo">
    <xsd:sequence>
      <xsd:element name="nome" type="xsd:string"/>
      <xsd:element name="provincia" type="xsd:string" />
      <xsd:element name="cima" type="xsd:string" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

Basi di dati II — 27 giugno 2013 — compito A

Mostrare come potrebbe essere modificato lo schema XSD mostrato in risposta al quesito precedente per rendere validi anche documenti in cui gli elementi `gruppoMontuoso` e `montagna` sono liberamente alternati, come ad esempio nel caso seguente

```
<montagne>
  <gruppoMontuoso>
    <nome>Gran Sasso</nome>
    <provincia>L'Aquila</provincia>
    <provincia>Teramo</provincia>
    <cima>Corno Grande</cima>
  </gruppoMontuoso>
  <montagna>Monviso</montagna>
  <montagna>Vesuvio</montagna>
  <gruppoMontuoso>
    <nome>Monte Rosa</nome>
    <cima>Punta Dufour</cima>
    <cima>Punta Gnifetti</cima>
  </gruppoMontuoso>
  <montagna>Etna</montagna>
</montagne>
```

Nota bene: mostrare solo la parte di schema che viene modificata, cioè quella relativa all'elemento `montagne`

Basi di dati II — 27 giugno 2013 — compito A

Domanda 3 (15% per la prova lunga, 30% per la prova breve)

Si supponga si abbiano documenti che contengono informazioni su film e interpreti, come il seguente:

```
<cineteca>
  <pellicola>
    <titolo>The Apartment</titolo>
    <cast>
      <interprete>Jack Lemmon</interprete>
      <interprete>Shirley MacLaine</interprete>
    </cast>
  </pellicola>
  <pellicola>
    <titolo>Some Like It Hot</titolo>
    <cast>
      <interprete>Marilyn Monroe</interprete>
      <interprete>Jack Lemmon</interprete>
    </cast>
  </pellicola>
  <pellicola>
    <titolo>Gentlemen Prefer Blondes</titolo>
    <cast>
      <interprete>Marilyn Monroe</interprete>
      <interprete>Jane Russel</interprete>
    </cast>
  </pellicola>
</cineteca>
```

Formulare le seguenti interrogazioni:

In XPath, trovare i titoli dei film interpretati da Jack Lemmon

In XPath, contare i film interpretati da Jack Lemmon presenti nell'elenco

Basi di dati II — 27 giugno 2013 — compito A

In XQuery, produrre un documento che contiene, per ogni interprete, la lista dei film. Con riferimento al documento mostrato in precedenza, si vuole ottenere:

```
<interprete>
  Jack Lemmon
  <titolo>The Apartment</titolo>
  <titolo>Some Like It Hot</titolo>
</interprete>
<interprete>
  Shirley MacLaine
  <titolo>The Apartment</titolo>
</interprete>
<interprete>
  Marilyn Monroe
  ..
```

Basi di dati II — 27 giugno 2013 — compito A

Domanda 4 (10% per la prova lunga, 20% per la prova breve)
Considerare un documento XML con il seguente DTD

```
<!ELEMENT document (record)>
<!ELEMENT record (A,B,C,D,E) >
<!ELEMENT A (#PCDATA) >
<!ELEMENT B (#PCDATA) >
<!ELEMENT C (#PCDATA) >
<!ELEMENT D (#PCDATA) >
<!ELEMENT E (#PCDATA) >
```

Si desidera memorizzare il documento in una base di dati relazionale e sono note le seguenti informazioni

- il documento contiene con $N = 400.000$ elementi `record`
- i valori dell'elemento `A`, che occupano $a = 10$ byte ciascuno, sono tutti diversi tra loro (e quindi possono essere usati per identificare gli elementi `record`)
- i valori degli elementi `B`, `C`, `D`, `E` occupano rispettivamente (in media, perché hanno lunghezza variabile, ma questo non è importante) $b = 30$ byte, $c = 20$ byte, $d = 10$ byte, $e = 80$ byte
- il sistema utilizzato ha blocchi di dimensione $P = 4$ kilobyte
- le operazioni più importanti sono le seguenti (vengono indicate in XQuery o XPath ma saranno poi eseguite in SQL):
 1. l'interrogazione XQuery che restituisce i valori di tutti gli elementi `record` su `A`, `B` e `C` (in una base di dati relazionale, questa sarebbe la proiezione dell'intera relazione sugli attributi `A`, `B` e `C`) con una frequenza $f_1 = 10$ esecuzioni nell'unità di tempo:

```
for $r in //document/record return <X>{$r/A},{r/B},{r/C}</X>
```
 2. l'interrogazione analoga su `A`, `D` ed `E`, con una frequenza $f_2 = 10$:

```
for $r in //document/record return <X>{$r/A},{r/D},{r/E}</X>
```
 3. l'interrogazione XPath che restituisce tutti i `record`, con una frequenza $f_3 = 20$:

```
//document/record
```

Indicare il numero di accessi a memoria secondaria nell'unità di tempo per le due possibili memorizzazioni, quella che utilizza una sola relazione e quella che ne utilizza due, con un partizionamento verticale (assumere che, grazie ad una opportuna memorizzazione, l'eventuale join possa essere eseguito con una singola scansione). Per ciascuna, indicare il costo in forma simbolica e numerica, nello spazio sottostante

Memorizzazione in una sola relazione, $R(\underline{A}, B, C, D, E)$:

Memorizzazione in due relazioni, $R_1(\underline{A}, B, C)$, $R_2(\underline{A}, D, E)$:

Basi di dati II — 27 giugno 2013 — compito A

Domanda 5 (20%)

Si abbiano due relazioni R_1 e R_2 che occupano rispettivamente $N_1 = 14.400$ e $N_2 = 216.000$ blocchi e siano disponibili $P = 150$ pagine di buffer.

1. Si supponga di effettuare un merge sort su R_1 , sfruttando nel modo migliore possibile i buffer disponibili.

Quanti passi di merge sono necessari, dopo il passo iniziale di ordinamento di porzioni del file (spiegare brevemente)?

Quanti blocchi sono occupati da ciascuna delle porzioni ordinate in cui viene inizialmente diviso il file (indicare formula e numero)?

Quanti blocchi sono occupati da ciascuna delle porzioni ordinate ottenute dopo il primo passo di merge?

2. Si supponga di effettuare un merge sort su R_2 , sfruttando nel modo migliore possibile i buffer disponibili.

Quanti passi di merge sono necessari, dopo il passo iniziale di ordinamento di porzioni del file (spiegare brevemente)?

Quanti blocchi sono occupati da ciascuna delle porzioni ordinate in cui viene inizialmente diviso il file (indicare formula e numero)?

Quanti blocchi sono occupati da ciascuna delle porzioni ordinate ottenute dopo il primo passo di merge?

3. Si supponga di effettuare un hash join di R_1 e R_2 , sfruttando nel modo migliore possibile i buffer disponibili.

Quanti bucket vengono prodotti per ciascuna delle due relazioni nel primo passo e quanti blocchi occupano, rispettivamente?

Nel secondo passo, quante pagine vengono dedicate a blocchi di R_1 ?

Nel secondo passo, quante pagine vengono dedicate a blocchi di R_2 ?

Basi di dati II — 27 giugno 2013 — compito A

Domanda 6 (15%)

In uno schema dimensionale relativo a chiamate telefoniche, si stanno considerando le seguenti tre alternative per la dimensione temporale:

1. una dimensione per la data e una per la fascia oraria
2. una dimensione per il mese (considerando non il nome ma l'effettiva identità, quindi ad esempio gennaio 2013 è diverso da gennaio 2012), una per il giorno della settimana e una per la fascia oraria
3. una dimensione per data e fascia oraria (insieme)

Confrontare le tre alternative in termini di granularità (indicando, con un breve commento, se sono uguali oppure quale o quali sono più fini)

Per ciascuna delle alternative, indicare una ragione per la quale essa può essere preferibile alle altre:

1.

2.

3.

Basi di dati II — 27 giugno 2013 — compito A

Domanda 7 (15%)

Per ciascuno degli schedule sotto riportati, indicare, scrivendo **sì** o **no** in ciascuna casella, a quali classi appartiene: S (seriale, rispetto a letture e scritture, ignorare commit e abort), CSR (conflict-serializzabile), S2PL (cioè generabile da uno scheduler basato su 2PL stretto), MV (cioè generabile da uno scheduler multiversion con controllo di serializzabilità: “a serializable transaction cannot modify or lock rows changed by other transactions after the serializable transaction began”). Negli schedule, s_i indica l’inizio della transazione i e c_i il suo commit.

	S	CSR	S2PL	MV
$s_2, s_1, r_2(x), w_2(x), r_1(x), c_2, w_1(x), c_1$				
$s_2, r_2(x), w_2(x), c_2, s_1, r_1(x), w_1(x), c_1$				
$s_1, r_1(x), s_2, r_2(x), w_1(x), c_1, w_2(x), c_2$				
$s_1, r_1(x), s_2, r_2(x), w_2(x), r_2(y), w_2(y), c_2, r_1(y), c_1$				

Basi di dati II
Esame — 27 giugno 2013 — Compito B

Rispondere su questo fascicolo.

Tempo a disposizione: due ore e quarantacinque minuti (prova lunga), un'ora e trenta minuti (prova breve).

Cognome _____ **Nome** _____ **Matricola** _____

Domanda 1 (10% per la prova lunga, 20% per la prova breve)

Considerare il seguente DTD

```
<!ELEMENT league (club+)>
<!ELEMENT club ((town,state?) | (town,state?,name) | name) >
<!ELEMENT town (#PCDATA) >
<!ELEMENT state (#PCDATA) >
<!ELEMENT name (#PCDATA) >
```

che si vorrebbe usare per validare documenti come il seguente

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE league SYSTEM "league.dtd">
<league>
  <club><town>Boston</town><state>MA</state><name>Red Sox</name></club>
  <club><name>Yankees</name></club>
  <club><town>Detroit</town></club>
</league>
```

Spiegare perché il DTD non è corretto.

Correggere il DTD (mostrare solo la riga o le righe da modificare)

Basi di dati II — 27 giugno 2013 — compito B

Domanda 2 (15% per la prova lunga, 30% per la prova breve)

Si supponga siano di interesse documenti che contengono informazioni su gruppi montuosi e singole montagne, come il seguente:

```
<montagne>
  <gruppoMontano>
    <nome>Gran Sasso</nome>
    <provincia>L'Aquila</provincia>
    <provincia>Teramo</provincia>
    <cima>Corno Grande</cima>
  </gruppoMontano>
  <gruppoMontano>
    <nome>Monte Rosa</nome>
    <cima>Punta Dufour</cima>
    <cima>Punta Gnifetti</cima>
  </gruppoMontano>
  <monte>Vesuvio</monte>
  <monte>Etna</monte>
</montagne>
```

Considerare lo schema XSD seguente con cui si è tentato erroneamente di modellare tali documenti e individuare (annotandole sullo schema stesso) le modifiche che a tale schema che rendano valido rispetto ad esso il documento sopra mostrato:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="montagne">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="gruppoMontano" type="TipoGruppo" maxOccurs="unbounded"/>
        <xsd:element name="monte" type="xsd:string" />
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:complexType name="TipoGruppo">
    <xsd:sequence>
      <xsd:element name="nome" type="xsd:string"/>
      <xsd:element name="provincia" type="xsd:string" maxOccurs="unbounded" />
      <xsd:element name="cima" type="xsd:string" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

Basi di dati II — 27 giugno 2013 — compito B

Mostrare come potrebbe essere modificato lo schema XSD mostrato in risposta al quesito precedente per rendere validi anche documenti in cui gli elementi `gruppoMontano` e `monte` sono liberamente alternati, come ad esempio nel caso seguente

```
<montagne>
  <gruppoMontano>
    <nome>Gran Sasso</nome>
    <provincia>L'Aquila</provincia>
    <provincia>Teramo</provincia>
    <cima>Corno Grande</cima>
  </gruppoMontano>
  <monte>Monviso</monte>
  <monte>Vesuvio</monte>
  <gruppoMontano>
    <nome>Monte Rosa</nome>
    <cima>Punta Dufour</cima>
    <cima>Punta Gnifetti</cima>
  </gruppoMontano>
  <monte>Etna</monte>
</montagne>
```

Nota bene: mostrare solo la parte di schema che viene modificata, cioè quella relativa all'elemento `montagne`

Basi di dati II — 27 giugno 2013 — compito B

Domanda 3 (15% per la prova lunga, 30% per la prova breve)

Si supponga si abbiano documenti che contengono informazioni su film e interpreti, come il seguente:

```
<cineteca>
  <movie>
    <titolo>The Apartment</titolo>
    <cast>
      <interprete>Jack Lemmon</interprete>
      <interprete>Shirley MacLaine</interprete>
    </cast>
  </movie>
  <movie>
    <titolo>Some Like It Hot</titolo>
    <cast>
      <interprete>Marilyn Monroe</interprete>
      <interprete>Jack Lemmon</interprete>
    </cast>
  </movie>
  <movie>
    <titolo>Gentlemen Prefer Blondes</titolo>
    <cast>
      <interprete>Marilyn Monroe</interprete>
      <interprete>Jane Russel</interprete>
    </cast>
  </movie>
</cineteca>
```

Formulare le seguenti interrogazioni:

In XPath, trovare i titoli dei film interpretati da Jack Lemmon

In XPath, contare i film interpretati da Jack Lemmon presenti nell'elenco

Basi di dati II — 27 giugno 2013 — compito B

In XQuery, produrre un documento che contiene, per ogni interprete, la lista dei film. Con riferimento al documento mostrato in precedenza, si vuole ottenere:

```
<interprete>
  Jack Lemmon
  <titolo>The Apartment</titolo>
  <titolo>Some Like It Hot</titolo>
</interprete>
<interprete>
  Shirley MacLaine
  <titolo>The Apartment</titolo>
</interprete>
<interprete>
  Marilyn Monroe
  ..
```

Basi di dati II — 27 giugno 2013 — compito B

Domanda 4 (10% per la prova lunga, 20% per la prova breve)
Considerare un documento XML con il seguente DTD

```
<!ELEMENT document (record)>
<!ELEMENT record (A,B,C,D,E) >
<!ELEMENT A (#PCDATA) >
<!ELEMENT B (#PCDATA) >
<!ELEMENT C (#PCDATA) >
<!ELEMENT D (#PCDATA) >
<!ELEMENT E (#PCDATA) >
```

Si desidera memorizzare il documento in una base di dati relazionale e sono note le seguenti informazioni

- il documento contiene con $L = 200.000$ elementi `record`
- i valori dell'elemento `A`, che occupano $a = 10$ byte ciascuno, sono tutti diversi tra loro (e quindi possono essere usati per identificare gli elementi `record`)
- i valori degli elementi `B`, `C`, `D`, `E` occupano rispettivamente (in media, perché hanno lunghezza variabile, ma questo non è importante) $b = 30$ byte, $c = 20$ byte, $d = 10$ byte, $e = 80$ byte
- il sistema utilizzato ha blocchi di dimensione $P = 2$ kilobyte
- le operazioni più importanti sono le seguenti (vengono indicate in XQuery o XPath ma saranno poi eseguite in SQL):
 1. l'interrogazione XQuery che restituisce i valori di tutti gli elementi `record` su `A`, `B` e `C` (in una base di dati relazionale, questa sarebbe la proiezione dell'intera relazione sugli attributi `A`, `B` e `C`) con una frequenza $f_1 = 10$ esecuzioni nell'unità di tempo:
$$\text{for } \$r \text{ in //document/record return } \langle X \rangle \{ \$r/A \}, \{ \$r/B \}, \{ \$r/C \} \langle /X \rangle$$
 2. l'interrogazione analoga su `A`, `D` ed `E`, con una frequenza $f_2 = 10$:
$$\text{for } \$r \text{ in //document/record return } \langle X \rangle \{ \$r/A \}, \{ \$r/D \}, \{ \$r/E \} \langle /X \rangle$$
 3. l'interrogazione XPath che restituisce tutti i `record`, con una frequenza $f_3 = 40$:
$$\text{//document/record}$$

Indicare il numero di accessi a memoria secondaria nell'unità di tempo per le due possibili memorizzazioni, quella che utilizza una sola relazione e quella che ne utilizza due, con un partizionamento verticale (assumere che, grazie ad una opportuna memorizzazione, l'eventuale join possa essere eseguito con una singola scansione). Per ciascuna, indicare il costo in forma simbolica e numerica, nello spazio sottostante

Memorizzazione in una sola relazione, $R(\underline{A}, B, C, D, E)$:

Memorizzazione in due relazioni, $R_1(\underline{A}, B, C)$, $R_2(\underline{A}, D, E)$:

Basi di dati II — 27 giugno 2013 — compito B

Domanda 5 (20%)

Si abbiano due relazioni R_1 e R_2 che occupano rispettivamente $N_1 = 125.000$ e $N_2 = 10.000$ blocchi e siano disponibili $P = 120$ pagine di buffer.

1. Si supponga di effettuare un merge sort su R_1 , sfruttando nel modo migliore possibile i buffer disponibili.

Quanti passi di merge sono necessari, dopo il passo iniziale di ordinamento di porzioni del file (spiegare brevemente)?

Quanti blocchi sono occupati da ciascuna delle porzioni ordinate in cui viene inizialmente diviso il file (indicare formula e numero)?

Quanti blocchi sono occupati da ciascuna delle porzioni ordinate ottenute dopo il primo passo di merge?

2. Si supponga di effettuare un merge sort su R_2 , sfruttando nel modo migliore possibile i buffer disponibili.

Quanti passi di merge sono necessari, dopo il passo iniziale di ordinamento di porzioni del file (spiegare brevemente)?

Quanti blocchi sono occupati da ciascuna delle porzioni ordinate in cui viene inizialmente diviso il file (indicare formula e numero)?

Quanti blocchi sono occupati da ciascuna delle porzioni ordinate ottenute dopo il primo passo di merge?

3. Si supponga di effettuare un hash join di R_1 e R_2 , sfruttando nel modo migliore possibile i buffer disponibili.

Quanti bucket vengono prodotti per ciascuna delle due relazioni nel primo passo e quanti blocchi occupano, rispettivamente?

Nel secondo passo, quante pagine vengono dedicate a blocchi di R_1 ?

Nel secondo passo, quante pagine vengono dedicate a blocchi di R_2 ?

Basi di dati II — 27 giugno 2013 — compito B

Domanda 6 (15%)

In uno schema dimensionale relativo a chiamate telefoniche, si stanno considerando le seguenti tre alternative per la dimensione temporale:

1. una dimensione per il mese (considerando non il nome ma l'effettiva identità, quindi ad esempio gennaio 2013 è diverso da gennaio 2012) e una per il giorno della settimana
2. una dimensione per la data
3. una dimensione per il mese (considerando non il nome ma l'effettiva identità, quindi ad esempio gennaio 2013 è diverso da gennaio 2012) e una per il giorno nel mese (cioè il numero da 1 a 31)

Confrontare le tre alternative in termini di granularità (indicando, con un breve commento, se sono uguali oppure quale o quali sono più fini)

Per ciascuna delle alternative, indicare una ragione per la quale essa può essere preferibile alle altre:

1.

2.

3.

Basi di dati II — 27 giugno 2013 — compito B

Domanda 7 (15%)

Per ciascuno degli schedule sotto riportati, indicare, scrivendo **sì** o **no** in ciascuna casella, a quali classi appartiene: S (seriale, rispetto a letture e scritture, ignorare commit e abort), CSR (conflict-serializzabile), S2PL (cioè generabile da uno scheduler basato su 2PL stretto), MV (cioè generabile da uno scheduler multiversion con controllo di serializzabilità: “a serializable transaction cannot modify or lock rows changed by other transactions after the serializable transaction began”). Negli schedule, s_i indica l’inizio della transazione i e c_i il suo commit.

	S	CSR	S2PL	MV
$s_2, r_2(x), w_2(x), c_2, s_1, r_1(x), w_1(x), c_1$				
$s_1, r_1(x), s_2, r_2(x), w_1(x), c_1, w_2(x), c_2$				
$s_2, s_1, r_2(x), w_2(x), r_1(x), c_2, w_1(x), c_1$				
$s_1, r_1(x), s_2, r_2(x), w_2(x), r_2(y), w_2(y), c_2, r_1(y), c_1$				

Basi di dati II
Esame — 27 giugno 2013 — Compito C

Rispondere su questo fascicolo.

Tempo a disposizione: due ore e quarantacinque minuti (prova lunga), un'ora e trenta minuti (prova breve).

Cognome _____ **Nome** _____ **Matricola** _____

Domanda 1 (10% per la prova lunga, 20% per la prova breve)

Considerare il seguente DTD

```
<!ELEMENT league (club+)>
<!ELEMENT club ((city,state?) | (city,state?,name) | name) >
<!ELEMENT city (#PCDATA) >
<!ELEMENT state (#PCDATA) >
<!ELEMENT name (#PCDATA) >
```

che si vorrebbe usare per validare documenti come il seguente

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE league SYSTEM "league.dtd">
<league>
  <club><city>Boston</city><state>MA</state><name>Red Sox</name></club>
  <club><name>Yankees</name></club>
  <club><city>Detroit</city></club>
</league>
```

Spiegare perché il DTD non è corretto.

Correggere il DTD (mostrare solo la riga o le righe da modificare)

Domanda 2 (15% per la prova lunga, 30% per la prova breve)

Si supponga siano di interesse documenti che contengono informazioni su gruppi montuosi e singole montagne, come il seguente:

```
<montagne>
  <gruppoMontano>
    <nome>Gran Sasso</nome>
    <provincia>L'Aquila</provincia>
    <provincia>Teramo</provincia>
    <cima>Corno Grande</cima>
  </gruppoMontano>
  <gruppoMontano>
    <nome>Monte Rosa</nome>
    <cima>Punta Dufour</cima>
    <cima>Punta Gnifetti</cima>
  </gruppoMontano>
  <montagna>Vesuvio</montagna>
  <montagna>Etna</montagna>
</montagne>
```

Considerare lo schema XSD seguente con cui si è tentato erroneamente di modellare tali documenti e individuare (annotandole sullo schema stesso) le modifiche che a tale schema che rendano valido rispetto ad esso il documento sopra mostrato:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="montagne">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="gruppoMontano" type="TipoGruppo" />
        <xsd:element name="montagna" type="xsd:string" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:complexType name="TipoGruppo">
    <xsd:sequence>
      <xsd:element name="nome" type="xsd:string"/>
      <xsd:element name="provincia" type="xsd:string" maxOccurs="unbounded" />
      <xsd:element name="cima" type="xsd:string" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

Basi di dati II — 27 giugno 2013 — compito C

Mostrare come potrebbe essere modificato lo schema XSD mostrato in risposta al quesito precedente per rendere validi anche documenti in cui gli elementi `gruppoMontano` e `montagna` sono liberamente alternati, come ad esempio nel caso seguente

```
<montagne>
  <gruppoMontano>
    <nome>Gran Sasso</nome>
    <provincia>L'Aquila</provincia>
    <provincia>Teramo</provincia>
    <cima>Corno Grande</cima>
  </gruppoMontano>
  <montagna>Monviso</montagna>
  <montagna>Vesuvio</montagna>
  <gruppoMontano>
    <nome>Monte Rosa</nome>
    <cima>Punta Dufour</cima>
    <cima>Punta Gnifetti</cima>
  </gruppoMontano>
  <montagna>Etna</montagna>
</montagne>
```

Nota bene: mostrare solo la parte di schema che viene modificata, cioè quella relativa all'elemento `montagne`

Basi di dati II — 27 giugno 2013 — compito C

Domanda 3 (15% per la prova lunga, 30% per la prova breve)

Si supponga si abbiano documenti che contengono informazioni su film e interpreti, come il seguente:

```
<cineteca>
  <pellicola>
    <titolo>The Apartment</titolo>
    <cast>
      <interprete>Jack Lemmon</interprete>
      <interprete>Shirley MacLaine</interprete>
    </cast>
  </pellicola>
  <pellicola>
    <titolo>Some Like It Hot</titolo>
    <cast>
      <interprete>Marilyn Monroe</interprete>
      <interprete>Jack Lemmon</interprete>
    </cast>
  </pellicola>
  <pellicola>
    <titolo>Gentlemen Prefer Blondes</titolo>
    <cast>
      <interprete>Marilyn Monroe</interprete>
      <interprete>Jane Russel</interprete>
    </cast>
  </pellicola>
</cineteca>
```

Formulare le seguenti interrogazioni:

In XPath, trovare i titoli dei film interpretati da Jack Lemmon

In XPath, contare i film interpretati da Jack Lemmon presenti nell'elenco

Basi di dati II — 27 giugno 2013 — compito C

In XQuery, produrre un documento che contiene, per ogni interprete, la lista dei film. Con riferimento al documento mostrato in precedenza, si vuole ottenere:

```
<interprete>
  Jack Lemmon
  <titolo>The Apartment</titolo>
  <titolo>Some Like It Hot</titolo>
</interprete>
<interprete>
  Shirley MacLaine
  <titolo>The Apartment</titolo>
</interprete>
<interprete>
  Marilyn Monroe
  ..
```

Domanda 4 (10% per la prova lunga, 20% per la prova breve)
 Considerare un documento XML con il seguente DTD

```
<!ELEMENT document (record)>
<!ELEMENT record (A,B,C,D,E) >
<!ELEMENT A (#PCDATA) >
<!ELEMENT B (#PCDATA) >
<!ELEMENT C (#PCDATA) >
<!ELEMENT D (#PCDATA) >
<!ELEMENT E (#PCDATA) >
```

Si desidera memorizzare il documento in una base di dati relazionale e sono note le seguenti informazioni

- il documento contiene con $N = 200.000$ elementi `record`
- i valori dell'elemento `A`, che occupano $a = 10$ byte ciascuno, sono tutti diversi tra loro (e quindi possono essere usati per identificare gli elementi `record`)
- i valori degli elementi `B`, `C`, `D`, `E` occupano rispettivamente (in media, perché hanno lunghezza variabile, ma questo non è importante) $b = 30$ byte, $c = 20$ byte, $d = 10$ byte, $e = 80$ byte
- il sistema utilizzato ha blocchi di dimensione $P = 2$ kilobyte
- le operazioni più importanti sono le seguenti (vengono indicate in XQuery o XPath ma saranno poi eseguite in SQL):
 1. l'interrogazione XQuery che restituisce i valori di tutti gli elementi `record` su `A`, `B` e `C` (in una base di dati relazionale, questa sarebbe la proiezione dell'intera relazione sugli attributi `A`, `B` e `C`) con una frequenza $f_1 = 10$ esecuzioni nell'unità di tempo:


```
for $r in //document/record return <X>{$r/A},{r/B},{r/C}</X>
```
 2. l'interrogazione analoga su `A`, `D` ed `E`, con una frequenza $f_2 = 10$:


```
for $r in //document/record return <X>{$r/A},{r/D},{r/E}</X>
```
 3. l'interrogazione XPath che restituisce tutti i `record`, con una frequenza $f_3 = 20$:


```
//document/record
```

Indicare il numero di accessi a memoria secondaria nell'unità di tempo per le due possibili memorizzazioni, quella che utilizza una sola relazione e quella che ne utilizza due, con un partizionamento verticale (assumere che, grazie ad una opportuna memorizzazione, l'eventuale join possa essere eseguito con una singola scansione). Per ciascuna, indicare il costo in forma simbolica e numerica, nello spazio sottostante

Memorizzazione in una sola relazione, $R(\underline{A}, B, C, D, E)$:

Memorizzazione in due relazioni, $R_1(\underline{A}, B, C)$, $R_2(\underline{A}, D, E)$:

Basi di dati II — 27 giugno 2013 — compito C

Domanda 5 (20%)

Si abbiano due relazioni R_1 e R_2 che occupano rispettivamente $N_1 = 14.400$ e $N_2 = 216.000$ blocchi e siano disponibili $P = 150$ pagine di buffer.

1. Si supponga di effettuare un merge sort su R_1 , sfruttando nel modo migliore possibile i buffer disponibili.

Quanti passi di merge sono necessari, dopo il passo iniziale di ordinamento di porzioni del file (spiegare brevemente)?

Quanti blocchi sono occupati da ciascuna delle porzioni ordinate in cui viene inizialmente diviso il file (indicare formula e numero)?

Quanti blocchi sono occupati da ciascuna delle porzioni ordinate ottenute dopo il primo passo di merge?

2. Si supponga di effettuare un merge sort su R_2 , sfruttando nel modo migliore possibile i buffer disponibili.

Quanti passi di merge sono necessari, dopo il passo iniziale di ordinamento di porzioni del file (spiegare brevemente)?

Quanti blocchi sono occupati da ciascuna delle porzioni ordinate in cui viene inizialmente diviso il file (indicare formula e numero)?

Quanti blocchi sono occupati da ciascuna delle porzioni ordinate ottenute dopo il primo passo di merge?

3. Si supponga di effettuare un hash join di R_1 e R_2 , sfruttando nel modo migliore possibile i buffer disponibili.

Quanti bucket vengono prodotti per ciascuna delle due relazioni nel primo passo e quanti blocchi occupano, rispettivamente?

Nel secondo passo, quante pagine vengono dedicate a blocchi di R_1 ?

Nel secondo passo, quante pagine vengono dedicate a blocchi di R_2 ?

Basi di dati II — 27 giugno 2013 — compito C

Domanda 6 (15%)

In uno schema dimensionale relativo a chiamate telefoniche, si stanno considerando le seguenti tre alternative per la dimensione temporale:

1. una dimensione per il mese (considerando non il nome ma l'effettiva identità, quindi ad esempio gennaio 2013 è diverso da gennaio 2012), una per il giorno della settimana e una per la fascia oraria
2. una dimensione per data e fascia oraria (insieme)
3. una dimensione per la data e una per la fascia oraria

Confrontare le tre alternative in termini di granularità (indicando, con un breve commento, se sono uguali oppure quale o quali sono più fini)

Per ciascuna delle alternative, indicare una ragione per la quale essa può essere preferibile alle altre:

1.

2.

3.

Basi di dati II — 27 giugno 2013 — compito C

Domanda 7 (15%)

Per ciascuno degli schedule sotto riportati, indicare, scrivendo **sì** o **no** in ciascuna casella, a quali classi appartiene: S (seriale, rispetto a letture e scritture, ignorare commit e abort), CSR (conflict-serializzabile), S2PL (cioè generabile da uno scheduler basato su 2PL stretto), MV (cioè generabile da uno scheduler multiversion con controllo di serializzabilità: “a serializable transaction cannot modify or lock rows changed by other transactions after the serializable transaction began”). Negli schedule, s_i indica l’inizio della transazione i e c_i il suo commit.

	S	CSR	S2PL	MV
$s_2, s_1, r_2(x), w_2(x), r_1(x), c_2, w_1(x), c_1$				
$s_2, r_2(x), w_2(x), c_2, s_1, r_1(x), w_1(x), c_1$				
$s_1, r_1(x), s_2, r_2(x), w_2(x), r_2(y), w_2(y), c_2, r_1(y), c_1$				
$s_1, r_1(x), s_2, r_2(x), w_1(x), c_1, w_2(x), c_2$				

Basi di dati II
Esame — 27 giugno 2013 — Compito D

Rispondere su questo fascicolo.

Tempo a disposizione: due ore e quarantacinque minuti (prova lunga), un'ora e trenta minuti (prova breve).

Cognome _____ **Nome** _____ **Matricola** _____

Domanda 1 (10% per la prova lunga, 20% per la prova breve)

Considerare il seguente DTD

```
<!ELEMENT league (team+)>
<!ELEMENT team ((town,state?) | (town,state?,name) | name) >
<!ELEMENT town (#PCDATA) >
<!ELEMENT state (#PCDATA) >
<!ELEMENT name (#PCDATA) >
```

che si vorrebbe usare per validare documenti come il seguente

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE league SYSTEM "league.dtd">
<league>
  <team><town>Boston</town><state>MA</state><name>Red Sox</name></team>
  <team><name>Yankees</name></team>
  <team><town>Detroit</town></team>
</league>
```

Spiegare perché il DTD non è corretto.

Correggere il DTD (mostrare solo la riga o le righe da modificare)

Basi di dati II — 27 giugno 2013 — compito D

Domanda 2 (15% per la prova lunga, 30% per la prova breve)

Si supponga siano di interesse documenti che contengono informazioni su gruppi montuosi e singole montagne, come il seguente:

```
<montagne>
  <gruppoMontuoso>
    <nome>Gran Sasso</nome>
    <provincia>L'Aquila</provincia>
    <provincia>Teramo</provincia>
    <cima>Corno Grande</cima>
  </gruppoMontuoso>
  <gruppoMontuoso>
    <nome>Monte Rosa</nome>
    <cima>Punta Dufour</cima>
    <cima>Punta Gnifetti</cima>
  </gruppoMontuoso>
  <monte>Vesuvio</monte>
  <monte>Etna</monte>
</montagne>
```

Considerare lo schema XSD seguente con cui si è tentato erroneamente di modellare tali documenti e individuare (annotandole sullo schema stesso) le modifiche che a tale schema che rendano valido rispetto ad esso il documento sopra mostrato:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="montagne">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="gruppoMontuoso" type="TipoGruppo" maxOccurs="unbounded"/>
        <xsd:element name="monte" type="xsd:string" />
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:complexType name="TipoGruppo">
    <xsd:sequence>
      <xsd:element name="nome" type="xsd:string"/>
      <xsd:element name="provincia" type="xsd:string" />
      <xsd:element name="cima" type="xsd:string" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

Basi di dati II — 27 giugno 2013 — compito D

Mostrare come potrebbe essere modificato lo schema XSD mostrato in risposta al quesito precedente per rendere validi anche documenti in cui gli elementi `gruppoMontuoso` e `monte` sono liberamente alternati, come ad esempio nel caso seguente

```
<montagne>
  <gruppoMontuoso>
    <nome>Gran Sasso</nome>
    <provincia>L'Aquila</provincia>
    <provincia>Teramo</provincia>
    <cima>Corno Grande</cima>
  </gruppoMontuoso>
  <monte>Monviso</monte>
  <monte>Vesuvio</monte>
  <gruppoMontuoso>
    <nome>Monte Rosa</nome>
    <cima>Punta Dufour</cima>
    <cima>Punta Gnifetti</cima>
  </gruppoMontuoso>
  <monte>Etna</monte>
</montagne>
```

Nota bene: mostrare solo la parte di schema che viene modificata, cioè quella relativa all'elemento `montagne`

Basi di dati II — 27 giugno 2013 — compito D

Domanda 3 (15% per la prova lunga, 30% per la prova breve)

Si supponga si abbiano documenti che contengono informazioni su film e interpreti, come il seguente:

```
<cineteca>
  <movie>
    <titolo>The Apartment</titolo>
    <cast>
      <interprete>Jack Lemmon</interprete>
      <interprete>Shirley MacLaine</interprete>
    </cast>
  </movie>
  <movie>
    <titolo>Some Like It Hot</titolo>
    <cast>
      <interprete>Marilyn Monroe</interprete>
      <interprete>Jack Lemmon</interprete>
    </cast>
  </movie>
  <movie>
    <titolo>Gentlemen Prefer Blondes</titolo>
    <cast>
      <interprete>Marilyn Monroe</interprete>
      <interprete>Jane Russel</interprete>
    </cast>
  </movie>
</cineteca>
```

Formulare le seguenti interrogazioni:

In XPath, trovare i titoli dei film interpretati da Jack Lemmon

In XPath, contare i film interpretati da Jack Lemmon presenti nell'elenco

Basi di dati II — 27 giugno 2013 — compito D

In XQuery, produrre un documento che contiene, per ogni interprete, la lista dei film. Con riferimento al documento mostrato in precedenza, si vuole ottenere:

```
<interprete>
  Jack Lemmon
  <titolo>The Apartment</titolo>
  <titolo>Some Like It Hot</titolo>
</interprete>
<interprete>
  Shirley MacLaine
  <titolo>The Apartment</titolo>
</interprete>
<interprete>
  Marilyn Monroe
  ..
```

Basi di dati II — 27 giugno 2013 — compito D

Domanda 4 (10% per la prova lunga, 20% per la prova breve)
Considerare un documento XML con il seguente DTD

```
<!ELEMENT document (record)>
<!ELEMENT record (A,B,C,D,E) >
<!ELEMENT A (#PCDATA) >
<!ELEMENT B (#PCDATA) >
<!ELEMENT C (#PCDATA) >
<!ELEMENT D (#PCDATA) >
<!ELEMENT E (#PCDATA) >
```

Si desidera memorizzare il documento in una base di dati relazionale e sono note le seguenti informazioni

- il documento contiene con $L = 400.000$ elementi `record`
- i valori dell'elemento `A`, che occupano $a = 10$ byte ciascuno, sono tutti diversi tra loro (e quindi possono essere usati per identificare gli elementi `record`)
- i valori degli elementi `B`, `C`, `D`, `E` occupano rispettivamente (in media, perché hanno lunghezza variabile, ma questo non è importante) $b = 30$ byte, $c = 20$ byte, $d = 10$ byte, $e = 80$ byte
- il sistema utilizzato ha blocchi di dimensione $P = 4$ kilobyte
- le operazioni più importanti sono le seguenti (vengono indicate in XQuery o XPath ma saranno poi eseguite in SQL):
 1. l'interrogazione XQuery che restituisce i valori di tutti gli elementi `record` su `A`, `B` e `C` (in una base di dati relazionale, questa sarebbe la proiezione dell'intera relazione sugli attributi `A`, `B` e `C`) con una frequenza $f_1 = 10$ esecuzioni nell'unità di tempo:
$$\text{for } \$r \text{ in //document/record return } \langle X \rangle \{ \$r/A \}, \{ \$r/B \}, \{ \$r/C \} \langle /X \rangle$$
 2. l'interrogazione analoga su `A`, `D` ed `E`, con una frequenza $f_2 = 10$:
$$\text{for } \$r \text{ in //document/record return } \langle X \rangle \{ \$r/A \}, \{ \$r/D \}, \{ \$r/E \} \langle /X \rangle$$
 3. l'interrogazione XPath che restituisce tutti i `record`, con una frequenza $f_3 = 40$:
$$\text{//document/record}$$

Indicare il numero di accessi a memoria secondaria nell'unità di tempo per le due possibili memorizzazioni, quella che utilizza una sola relazione e quella che ne utilizza due, con un partizionamento verticale (assumere che, grazie ad una opportuna memorizzazione, l'eventuale join possa essere eseguito con una singola scansione). Per ciascuna, indicare il costo in forma simbolica e numerica, nello spazio sottostante

Memorizzazione in una sola relazione, $R(\underline{A}, B, C, D, E)$:

Memorizzazione in due relazioni, $R_1(\underline{A}, B, C)$, $R_2(\underline{A}, D, E)$:

Basi di dati II — 27 giugno 2013 — compito D

Domanda 5 (20%)

Si abbiano due relazioni R_1 e R_2 che occupano rispettivamente $N_1 = 125.000$ e $N_2 = 10.000$ blocchi e siano disponibili $P = 120$ pagine di buffer.

1. Si supponga di effettuare un merge sort su R_1 , sfruttando nel modo migliore possibile i buffer disponibili.

Quanti passi di merge sono necessari, dopo il passo iniziale di ordinamento di porzioni del file (spiegare brevemente)?

Quanti blocchi sono occupati da ciascuna delle porzioni ordinate in cui viene inizialmente diviso il file (indicare formula e numero)?

Quanti blocchi sono occupati da ciascuna delle porzioni ordinate ottenute dopo il primo passo di merge?

2. Si supponga di effettuare un merge sort su R_2 , sfruttando nel modo migliore possibile i buffer disponibili.

Quanti passi di merge sono necessari, dopo il passo iniziale di ordinamento di porzioni del file (spiegare brevemente)?

Quanti blocchi sono occupati da ciascuna delle porzioni ordinate in cui viene inizialmente diviso il file (indicare formula e numero)?

Quanti blocchi sono occupati da ciascuna delle porzioni ordinate ottenute dopo il primo passo di merge?

3. Si supponga di effettuare un hash join di R_1 e R_2 , sfruttando nel modo migliore possibile i buffer disponibili.

Quanti bucket vengono prodotti per ciascuna delle due relazioni nel primo passo e quanti blocchi occupano, rispettivamente?

Nel secondo passo, quante pagine vengono dedicate a blocchi di R_1 ?

Nel secondo passo, quante pagine vengono dedicate a blocchi di R_2 ?

Basi di dati II — 27 giugno 2013 — compito D

Domanda 6 (15%)

In uno schema dimensionale relativo a chiamate telefoniche, si stanno considerando le seguenti tre alternative per la dimensione temporale:

1. una dimensione per la data
2. una dimensione per il mese (considerando non il nome ma l'effettiva identità, quindi ad esempio gennaio 2013 è diverso da gennaio 2012) e una per il giorno nel mese (cioè il numero da 1 a 31)
3. una dimensione per il mese (considerando non il nome ma l'effettiva identità, quindi ad esempio gennaio 2013 è diverso da gennaio 2012) e una per il giorno della settimana

Confrontare le tre alternative in termini di granularità (indicando, con un breve commento, se sono uguali oppure quale o quali sono più fini)

Per ciascuna delle alternative, indicare una ragione per la quale essa può essere preferibile alle altre:

1.

2.

3.

Basi di dati II — 27 giugno 2013 — compito D

Domanda 7 (15%)

Per ciascuno degli schedule sotto riportati, indicare, scrivendo **sì** o **no** in ciascuna casella, a quali classi appartiene: S (seriale, rispetto a letture e scritture, ignorare commit e abort), CSR (conflict-serializzabile), S2PL (cioè generabile da uno scheduler basato su 2PL stretto), MV (cioè generabile da uno scheduler multiversion con controllo di serializzabilità: “a serializable transaction cannot modify or lock rows changed by other transactions after the serializable transaction began”). Negli schedule, s_i indica l’inizio della transazione i e c_i il suo commit.

	S	CSR	S2PL	MV
$s_1, r_1(x), s_2, r_2(x), w_2(x), r_2(y), w_2(y), c_2, r_1(y), c_1$				
$s_1, r_1(x), s_2, r_2(x), w_1(x), c_1, w_2(x), c_2$				
$s_2, s_1, r_2(x), w_2(x), r_1(x), c_2, w_1(x), c_1$				
$s_2, r_2(x), w_2(x), c_2, s_1, r_1(x), w_1(x), c_1$				

Basi di dati II
Esame — 27 giugno 2013 — Compito A
Cenni sulle soluzioni

Rispondere su questo fascicolo.

Tempo a disposizione: **due ore e quarantacinque minuti (prova lunga), un'ora e trenta minuti (prova breve).**

Cognome _____ Nome _____ Matricola _____

Domanda 1 (10% per la prova lunga, 20% per la prova breve)

Considerare il seguente DTD

```
<!ELEMENT league (team+)>
<!ELEMENT team ((city,state?) | (city,state?,name) | name) >
<!ELEMENT city (#PCDATA) >
<!ELEMENT state (#PCDATA) >
<!ELEMENT name (#PCDATA) >
```

che si vorrebbe usare per validare documenti come il seguente

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE league SYSTEM "league.dtd">
<league>
  <team><city>Boston</city><state>MA</state><name>Red Sox</name></team>
  <team><name>Yankees</name></team>
  <team><city>Detroit</city></team>
</league>
```

Spiegare perché il DTD non è corretto.

Possibile risposta

La specifica è ambigua in quanto l'espressione regolare è non deterministica: leggendo un documento, il parser, trovato un elemento **city** non saprebbe se proseguire con la prima sottoespressione o la seconda

Correggere il DTD (mostrare solo la riga o le righe da modificare)

Possibile risposta, l'intero DTD

```
<!ELEMENT league (team+)>
<!ELEMENT team ( (city,state?,name?) | name) >
<!ELEMENT city (#PCDATA) >
<!ELEMENT state (#PCDATA) >
<!ELEMENT name (#PCDATA) >
```

Basi di dati II — 27 giugno 2013 — compito A

Domanda 2 (15% per la prova lunga, 30% per la prova breve)

Si supponga siano di interesse documenti che contengono informazioni su gruppi montuosi e singole montagne, come il seguente:

```
<montagne>
  <gruppoMontuoso>
    <nome>Gran Sasso</nome>
    <provincia>L'Aquila</provincia>
    <provincia>Teramo</provincia>
    <cima>Corno Grande</cima>
  </gruppoMontuoso>
  <gruppoMontuoso>
    <nome>Monte Rosa</nome>
    <cima>Punta Dufour</cima>
    <cima>Punta Gnifetti</cima>
  </gruppoMontuoso>
  <montagna>Vesuvio</montagna>
  <montagna>Etna</montagna>
</montagne>
```

Considerare lo schema XSD seguente con cui si è tentato erroneamente di modellare tali documenti e individuare (annotandole sullo schema stesso) le modifiche che a tale schema che rendano valido rispetto ad esso il documento sopra mostrato:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="montagne">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="gruppoMontuoso" type="TipoGruppo" maxOccurs="unbounded"/>
        <xsd:element name="montagna" type="xsd:string" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:complexType name="TipoGruppo">
    <xsd:sequence>
      <xsd:element name="nome" type="xsd:string"/>
      <xsd:element name="provincia" type="xsd:string" minOccurs="0" maxOccurs="unbounded" />
      <xsd:element name="cima" type="xsd:string" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

Basi di dati II — 27 giugno 2013 — compito A

Mostrare come potrebbe essere modificato lo schema XSD mostrato in risposta al quesito precedente per rendere validi anche documenti in cui gli elementi `gruppoMontuoso` e `montagna` sono liberamente alternati, come ad esempio nel caso seguente

```
<montagne>
  <gruppoMontuoso>
    <nome>Gran Sasso</nome>
    <provincia>L'Aquila</provincia>
    <provincia>Teramo</provincia>
    <cima>Corno Grande</cima>
  </gruppoMontuoso>
  <montagna>Monviso</montagna>
  <montagna>Vesuvio</montagna>
  <gruppoMontuoso>
    <nome>Monte Rosa</nome>
    <cima>Punta Dufour</cima>
    <cima>Punta Gnifetti</cima>
  </gruppoMontuoso>
  <montagna>Etna</montagna>
</montagne>
```

Nota bene: mostrare solo la parte di schema che viene modificata, cioè quella relativa all'elemento `montagne`

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="montagne">
    <xsd:complexType>
      <xsd:choice maxOccurs="unbounded">
        <xsd:element name="gruppoMontuoso" type="TipoGruppoMontuoso"/>
        <xsd:element name="montagna" type="xsd:string" />
      </xsd:choice>
    </xsd:complexType>
  </xsd:element>
  <xsd:complexType name="TipoGruppoMontuoso">
    <xsd:sequence>
      <xsd:element name="nome" type="xsd:string"/>
      <xsd:element name="provincia" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="cima" type="xsd:string" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

Basi di dati II — 27 giugno 2013 — compito A

Domanda 3 (15% per la prova lunga, 30% per la prova breve)

Si supponga si abbiano documenti che contengono informazioni su film e interpreti, come il seguente:

```
<cineteca>
  <pellicola>
    <titolo>The Apartment</titolo>
    <cast>
      <interprete>Jack Lemmon</interprete>
      <interprete>Shirley MacLaine</interprete>
    </cast>
  </pellicola>
  <pellicola>
    <titolo>Some Like It Hot</titolo>
    <cast>
      <interprete>Marilyn Monroe</interprete>
      <interprete>Jack Lemmon</interprete>
    </cast>
  </pellicola>
  <pellicola>
    <titolo>Gentlemen Prefer Blondes</titolo>
    <cast>
      <interprete>Marilyn Monroe</interprete>
      <interprete>Jane Russel</interprete>
    </cast>
  </pellicola>
</cineteca>
```

Formulare le seguenti interrogazioni:

In XPath, trovare i titoli dei film interpretati da Jack Lemmon

```
//pellicola[cast/interprete="Jack Lemmon"]/titolo oppure
//pellicola[./interprete="Jack Lemmon"]/titolo
```

notare che

```
//pellicola[./interprete="Jack Lemmon"]/titolo è sbagliato
```

In XPath, contare i film interpretati da Jack Lemmon presenti nell'elenco

```
count(//pellicola[cast/interprete="Jack Lemmon"]) oppure
count(//pellicola[./interprete="Jack Lemmon"])
```

notare che

```
//pellicola[./interprete="Jack Lemmon"] è sbagliato
```

Basi di dati II — 27 giugno 2013 — compito A

In XQuery, produrre un documento che contiene, per ogni interprete, la lista dei film. Con riferimento al documento mostrato in precedenza, si vuole ottenere:

```
<interprete>
  Jack Lemmon
  <titolo>The Apartment</titolo>
  <titolo>Some Like It Hot</titolo>
</interprete>
<interprete>
  Shirley MacLaine
  <titolo>The Apartment</titolo>
</interprete>
<interprete>
  Marilyn Monroe
  ..
```

Possibile risposta:

```
let $nl := "
"
for $i in distinct-values(//interprete)
return
<interprete>
  {" ",$i}{for $f in //pellicola
  where $f/cast/interprete = $i
  return ($nl," ",$f/titolo)}
</interprete>
```

Domanda 4 (10% per la prova lunga, 20% per la prova breve)
 Considerare un documento XML con il seguente DTD

```
<!ELEMENT document (record)>
<!ELEMENT record (A,B,C,D,E) >
<!ELEMENT A (#PCDATA) >
<!ELEMENT B (#PCDATA) >
<!ELEMENT C (#PCDATA) >
<!ELEMENT D (#PCDATA) >
<!ELEMENT E (#PCDATA) >
```

Si desidera memorizzare il documento in una base di dati relazionale e sono note le seguenti informazioni

- il documento contiene con $N = 400.000$ elementi `record`
- i valori dell'elemento `A`, che occupano $a = 10$ byte ciascuno, sono tutti diversi tra loro (e quindi possono essere usati per identificare gli elementi `record`)
- i valori degli elementi `B`, `C`, `D`, `E` occupano rispettivamente (in media, perché hanno lunghezza variabile, ma questo non è importante) $b = 30$ byte, $c = 20$ byte, $d = 10$ byte, $e = 80$ byte
- il sistema utilizzato ha blocchi di dimensione $P = 4$ kilobyte
- le operazioni più importanti sono le seguenti (vengono indicate in XQuery o XPath ma saranno poi eseguite in SQL):
 1. l'interrogazione XQuery che restituisce i valori di tutti gli elementi `record` su `A`, `B` e `C` (in una base di dati relazionale, questa sarebbe la proiezione dell'intera relazione sugli attributi `A`, `B` e `C`) con una frequenza $f_1 = 10$ esecuzioni nell'unità di tempo:

$$\text{for } \$r \text{ in //document/record return } \langle X \rangle \{ \$r/A \}, \{ \$r/B \}, \{ \$r/C \} \langle /X \rangle$$
 2. l'interrogazione analoga su `A`, `D` ed `E`, con una frequenza $f_2 = 10$:

$$\text{for } \$r \text{ in //document/record return } \langle X \rangle \{ \$r/A \}, \{ \$r/D \}, \{ \$r/E \} \langle /X \rangle$$
 3. l'interrogazione XPath che restituisce tutti i `record`, con una frequenza $f_3 = 20$:

$$\text{//document/record}$$

Indicare il numero di accessi a memoria secondaria nell'unità di tempo per le due possibili memorizzazioni, quella che utilizza una sola relazione e quella che ne utilizza due, con un partizionamento verticale (assumere che, grazie ad una opportuna memorizzazione, l'eventuale join possa essere eseguito con una singola scansione). Per ciascuna, indicare il costo in forma simbolica e numerica, nello spazio sottostante

Memorizzazione in una sola relazione, $R(\underline{A}, B, C, D, E)$:

$$c_1 = N \times (a + b + c + d + e) / P = 400.000 \times 150 / 4.000 = 15.000$$

$$c_2 = N \times (a + b + c + d + e) / P = 15.000$$

$$c_3 = N \times (a + b + c + d + e) / P = 15.000$$

$$c_T = c_1 \times f_1 + c_2 \times f_2 + c_3 \times f_3 = 600.000$$

Memorizzazione in due relazioni, $R_1(\underline{A}, B, C)$, $R_2(\underline{A}, D, E)$:

$$c_1 = N \times (a + b + c) / P = 6.000$$

$$c_2 = N \times (a + d + e) / P = 10.000$$

$$c_3 = N \times ((a + b + c) + (a + d + e)) / P = 16.000$$

$$c_T = c_1 \times f_1 + c_2 \times f_2 + c_3 \times f_3 = 480.000$$

Basi di dati II — 27 giugno 2013 — compito A

Domanda 5 (20%)

Si abbiano due relazioni R_1 e R_2 che occupano rispettivamente $N_1 = 14.400$ e $N_2 = 216.000$ blocchi e siano disponibili $P = 150$ pagine di buffer.

1. Si supponga di effettuare un merge sort su R_1 , sfruttando nel modo migliore possibile i buffer disponibili.

Quanti passi di merge sono necessari, dopo il passo iniziale di ordinamento di porzioni del file (spiegare brevemente)?

1

Quanti blocchi sono occupati da ciascuna delle porzioni ordinate in cui viene inizialmente diviso il file (indicare formula e numero)?

$$\sqrt[1+1]{N_1} = 120$$

Quanti blocchi sono occupati da ciascuna delle porzioni ordinate ottenute dopo il primo passo di merge?

14.400

2. Si supponga di effettuare un merge sort su R_2 , sfruttando nel modo migliore possibile i buffer disponibili.

Quanti passi di merge sono necessari, dopo il passo iniziale di ordinamento di porzioni del file (spiegare brevemente)?

2

Quanti blocchi sono occupati da ciascuna delle porzioni ordinate in cui viene inizialmente diviso il file (indicare formula e numero)?

$$\sqrt[2+1]{N_2} = 60$$

Quanti blocchi sono occupati da ciascuna delle porzioni ordinate ottenute dopo il primo passo di merge?

3.600

3. Si supponga di effettuare un hash join di R_1 e R_2 , sfruttando nel modo migliore possibile i buffer disponibili.

Quanti bucket vengono prodotti per ciascuna delle due relazioni nel primo passo e quanti blocchi occupano, rispettivamente?

120 bucket, 120 e 1800 blocchi

Nel secondo passo, quante pagine vengono dedicate a blocchi di R_1 ?

120

Nel secondo passo, quante pagine vengono dedicate a blocchi di R_2 ?

1

Basi di dati II — 27 giugno 2013 — compito A

Domanda 6 (15%)

In uno schema dimensionale relativo a chiamate telefoniche, si stanno considerando le seguenti tre alternative per la dimensione temporale:

1. una dimensione per la data e una per la fascia oraria
2. una dimensione per il mese (considerando non il nome ma l'effettiva identità, quindi ad esempio gennaio 2013 è diverso da gennaio 2012), una per il giorno della settimana e una per la fascia oraria
3. una dimensione per data e fascia oraria (insieme)

Confrontare le tre alternative in termini di granularità (indicando, con un breve commento, se sono uguali oppure quale o quali sono più fini) In tutti i compiti l'alternativa con il giorno della settimana ha grana più grossa, le altre due più fine

Per ciascuna delle alternative, indicare una ragione per la quale essa può essere preferibile alle altre:

Basi di dati II — 27 giugno 2013 — compito A

Domanda 7 (15%)

Per ciascuno degli schedule sotto riportati, indicare, scrivendo **sì** o **no** in ciascuna casella, a quali classi appartiene: S (seriale, rispetto a letture e scritture, ignorare commit e abort), CSR (conflict-serializzabile), S2PL (cioè generabile da uno scheduler basato su 2PL stretto), MV (cioè generabile da uno scheduler multiversion con controllo di serializzabilità: “a serializable transaction cannot modify or lock rows changed by other transactions after the serializable transaction began”). Negli schedule, s_i indica l’inizio della transazione i e c_i il suo commit.

Soluzioni per il compito A

	S	CSR	S2PL	MV
$s_2, s_1, r_2(x), w_2(x), r_1(x), c_2, w_1(x), c_1$	sì	sì	no	no
$s_2, r_2(x), w_2(x), c_2, s_1, r_1(x), w_1(x), c_1$	sì	sì	sì	sì
$s_1, r_1(x), s_2, r_2(x), w_1(x), c_1, w_2(x), c_2$	no	no	no	no
$s_1, r_1(x), s_2, r_2(x), w_2(x), r_2(y), w_2(y), c_2, r_1(y), c_1$	no	no	no	sì *

(*) in questo caso $r_1(y)$ legge il valore di y all’inizio della transazione 1, cioè prima della scrittura $w_2(y)$