

**Complementi di Basi di dati – Esame del 9 aprile 2008**  
**Tempo a disposizione: 2 ore (libri chiusi)**

**Possibile Soluzione Domanda 1 (10%)**

(si noti il content model dell'elemento dichiarazione utilizzato per esprimere contenuto misto).

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT dichiarazioni (dichiarazione)+>
<!ELEMENT dichiarazione (#PCDATA|data|studente|corso|docente)*>
<!ELEMENT data (giorno, mese, anno)>
<!ELEMENT studente (nome, cognome)>
<!ELEMENT giorno (#PCDATA)>
<!ELEMENT mese (#PCDATA)>
<!ELEMENT anno (#PCDATA)>
<!ELEMENT nome (#PCDATA)>
<!ELEMENT cognome (#PCDATA)>
<!ELEMENT corso (#PCDATA)>
<!ELEMENT docente (#PCDATA)>
<!ATTLIST dichiarazione numero CDATA #REQUIRED>
```

**Possibile Soluzione domanda 2 (20%)**

documento XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<a:visite xmlns:a="http://uniroma3.it"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://uniroma3.it due.xsd">
  <visita numero="1234">
    <dataora>9 aprile 2008 - 10:30</dataora>
    <diagnosi>Tonsillite</diagnosi>
    <paziente CF="BELIVN74T12H501Q">
      <cognome>Belli</cognome>
      <nome>Ivana</nome>
      <datanascita>12/12/1974</datanascita>
      <indirizzo>Via dei Pini, 13 - Roma</indirizzo>
    </paziente>
    <medico codice="abd456">
      <cognome>Marchi</cognome>
      <nome>Andrea</nome>
      <telefono>0655214587</telefono>
      <specializzazione codice="OLG">Otorino</specializzazione>
    </medico>
  </visita>
</a:visite>
```

XML schema:

```
<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:a="http://uniroma3.it"
targetNamespace="http://uniroma3.it">
  <element name="visite">
    <complexType>
      <sequence>
        <element name="visita" type="a:tipo_visita" maxOccurs="unbounded">
          <key name="visita_id">
            <selector xpath="."/>
            <field xpath="@numero"/>
          </key>
        </element>
      </sequence>
    </complexType>
  </element>
```

```

<complexType name="tipo_visita">
  <sequence>
    <element name="dataora" type="string" minOccurs="1" />
    <element name="diagnosi" type="string" />
    <element name="paziente" type="a:tipo_paziente" />
    <element name="medico" type="a:tipo_medico" />
  </sequence>
  <attribute name="numero" type="integer" use="required"/>
</complexType>
<complexType name="tipo_paziente">
  <sequence>
    <element name="cognome" type="string" minOccurs="1" />
    <element name="nome" type="string" />
    <element name="datanascita" type="string" />
    <element name="indirizzo" type="string" />
  </sequence>
  <attribute name="CF" type="string" use="required"/>
</complexType>
<complexType name="tipo_medico">
  <sequence>
    <element name="cognome" type="string" minOccurs="1" />
    <element name="nome" type="string" />
    <element name="telefono" type="string" />
    <element name="specializzazione" type="a:tipo_spec" />
  </sequence>
  <attribute name="codice" type="string" use="required"/>
</complexType>
<complexType name="tipo_spec">
  <simpleContent>
    <extension base="string">
      <attribute name="codice" type="string" use="required"/>
    </extension>
  </simpleContent>
</complexType>
</schema>

```

### Possibile Soluzione Domanda 3 (30%)

(a)

```
doc("Scontrini.xml")/scontrini/scontrino[.//prodotto="35657" and .//totale>10]/data
```

(b)

```

<xsl:stylesheet ...>
  <xsl:template match="scontrini">
    <html>
      <head><title>Riepilogo</title></head>
      <body>
        <ul>
          <xsl:for-each-group select="scontrino" group-by="./data">
            <li>
              <b>Data</b> - <xsl:value-of select="data/giorno"/>/
                <xsl:value-of select="data/mese"/>/
                <xsl:value-of select="data/anno"/><br/>
              <b>Incasso totale: </b>
                <xsl:value-of select="sum(current-group()/totale)"/><br/>
              <b>Prodotti: </b>
                <xsl:value-of select="current-group()//prodotto"/><br/>
            </li>
          </xsl:for-each-group>
        </ul>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>

```

(c)

```
<spese-clienti>
{
  for $c in doc("Clienti.xml")//cliente
  order by $c/eta
  return
  <spese-cliente>
    <cliente>{ $c/nome } { $c/cognome }</cliente>
    <spese>
      {
        for $s in doc("Scontrini.xml")//scontrino
        where $s/cliente = $c/codice
        return <spesa>{$s/data,$s//prodotto,$s//totale}</spesa>
      }
    </spese>
  </spese-cliente>
}
</spese-clienti>
```

#### Possibile Soluzione domanda 4 (20%)

Per ciascuno dei due mapping, vediamo se sono soddisfatte le tre proprietà:

- mapping da S1 ad S2:
  - totale: sì, perché per ogni istanza lecita di S1 si ottiene una istanza lecita di S2
  - iniettivo: no, perché ci possono essere istanze diverse di S1 che portano alla stessa istanza di S2; ad esempio, le seguenti istanze di S1 portano alla stessa istanza di S2:

A	B	C
10	20	30

A	D
11	40

  

A	B	C
10	20	30

A	D

Se ci fosse un vincolo che impone ai valori di A di essere gli stessi nelle due relazioni, allora il mapping sarebbe iniettivo

- funzionale: sì, perché per ogni istanza di S1 si ottiene al più una istanza di S2
- mapping da S2 ad S1
  - totale: no, perché ad alcune istanze lecite di S2 non corrispondono istanze lecite di S1; esempio:

A	B
10	20

A	D	C
10	40	30
10	41	31

- iniettivo: no, per motivi analoghi a quelli visti per l'altro mapping; anche qui, se ci fosse un vincolo che impone ai valori di A di essere gli stessi nelle due relazioni, allora il mapping sarebbe iniettivo
- funzionale: sì, perché per ogni istanza di S1 si ottiene al più una istanza di S2

#### Possibile Soluzione domanda 5 (20%)

(usiamo abbreviazioni per i nomi degli attributi)

Chiave:

gli attributi P e S compaiono solo a primo membro e quindi partecipano a tutte le chiavi; la chiusura dell'insieme PS contiene tutti gli attributi e quindi PS è chiave (ed è l'unica chiave)

Copertura ridotta:

l'unica fd con due attributi a primo membro è  $SC \rightarrow FE$ ; calcoliamo la chiusura di S rispetto alle fd e otteniamo SCFE e quindi  $S \rightarrow FE$  è implicata e quindi possiamo eliminare C dal primo membro; se avessimo fatto lo stesso con S non avremmo potuto eliminarlo, perché la chiusura di C è CF (senza E);

fra le varie dipendenze (considerando i secondi membri un attributo alla volta), possiamo eliminare, perché ridondanti, le fd  $P \rightarrow I$  e  $S \rightarrow F$ ;

quindi una copertura ridotta (in effetti, l'unica) è:  $P \rightarrow D, D \rightarrow I, S \rightarrow C, S \rightarrow E, C \rightarrow F$

L'algoritmo di sintesi genera quindi lo schema :

$R_1$  (Professore, Dipartimento)  
 $R_2$  (Dipartimento, Indirizzo)  
 $R_3$  (Studiante ,CorsoDiLaurea, Età)  
 $R_4$  (CorsoDiLaurea, Facoltà)  
 $R_0$ (Professore, Studente)