- 1. (a) (8 points) Of the following statements, identify all that hold about XML keys:
 - A. A keyref implies that the selected tuples are unique but only for the fields specified
 - B. A keyref indicates that each selected tuple of a selected element corresponds to an element in the referenced key
 - C. The unique element has been deprecated and replaced by the key element
 - D. A key allows more than one field

Solution: B and D -2 per wrongly checked or wrongly unchecked choice

(b) (8 points) Of the following statements, identify all that hold about the data-centric view of XML

- A. The data-centric view works best when dealing with settings where documents have a regular structure
- B. The data-centric view gives prominence to XML documents as repositories of corporate knowledge
- C. The data-centric view reduces but may not fully eliminate the need for storing large data objects such as CLOBs
- D. The data-centric view requires that XML documents be shared through databases: thus there is no need for separate messaging

Solution: A and C

-2 per wrongly checked or wrongly unchecked choice

- (c) (8 points) Of the following statements, identify all that hold about the XPath syntax allowed in key selectors
 - A. It is possible to construct an expression referring to a grandchild element of the given element
 - B. If the schema states that a specific grandchild element of the given element may contain only text, it is possible to construct an expression referring to the text occurring within that grandchild element
 - C. It is possible to construct an expression referring to the first grandchild element of the given element
 - D. ../child is allowed, where child is a particular child element and exists in the appropriate context

Solution: A and B

-2 per wrongly checked or wrongly unchecked choice

(d) (6 points) List three possible interpretations of "null" values (in about 25 words total).

Solution:

- Not applicable
- Unknown: missing
- Questionable existence
- Absent (known but absent)

Listing 1: Unique songs nested in unique singers with no attributes

```
<SongList>
<listTitle>Classics</listTitle>
<Sgr name="Eagles">
<Song genre="rock" lg="en">Hotel California</Song>
</Sgr>
```

```
<Sgr name="H_Belafonte">

<Song genre="folk" lg="cpe">Day O</Song>

<Song genre="calypso" lg="en">Jamaica Farewell</Song>

</Sgr>

<Sgr name="J_Prasad">

<Song lg="pa" genre="folk">Mera Dil Darda</Song>

</Sgr>

</Sgr>
```

2. (20 points) Consider the XML schema underlying Listing 1 to a relational schema. Assume that SongList have unique listTitles, Sgrs have unique names within SongList, and within a Sgr, song titles (expressed as text fields) are unique.

Of the following statements, identify all that are true about mapping the above-mentioned XML schema to a relational schema using *the generalized TGE approach*

- A. Create a table SongList whose key is listTitle
- B. Create a table SongList with no columns
- C. Create a table Sgr with one column name, which is also its key
- D. Create a table Sgr with two columns name and listTitle; name is its key and listTitle is a foreign key to the SongList table
- E. The Song table has at least the columns genre and lg for the corresponding XML attributes
- F. The Song table has a column for the text
- G. The Song table has both the following foreign keys: name in Sgr and listTitle in SongList
- H. The key of the Song table is a composite of name and listTitle
- I. There is no need for a separate table for listTitle because it is promoted to help uniquify its parent, SongList
- J. Many rows of the Sgr table may have the same name, but each such row would correspond to a different song title in Song

Solution: A, E, F, G, I

D is false because the Sgr's key should be a composite of name and listTitle

H is false because Song's key should also include its title text

J is false because the songs may be repeated in different song lists

More on the solution itself:

- listTitle is promoted to uniquify SongList
- SongList(listTitle); key is listTitle
- Sgr(name, listTitle); key is (name, listTitle); foreign key on listTitle
- Song(name, listTitle, genre, lg, title-text); key is (name, listTitle, title-text); foreign keys on listTitle and name