A. Data Formats (35 points)

1. (10 points) Explain the following two advantages of storing data in the form of XML.

   **XML is flexible**
   If an object’s property is expressed as a certain child tag (rather than a column), different objects can have different number of instances of that property (but we cannot have variable number of columns in the relational model). The tags under a certain tag can also change is the schema allows.

   **XML is self-describing**
   A tag name describes its content, therefore, a data collection in XML can be understood without an external description (schema). Note we may still have DTDs for validating XML content, but that’s just for validation.
2. **(10 points)** Use DTD to describe a collection of books that satisfies the following conditions:
   i. The collection contains zero or more books.
   ii. Every book in such a collection is required to have a *title* and one or more *author(s)*.
   iii. Each book is required to have a unique-identifier called *ISBN*.
   iv. Each book has an *index* and one or more *section(s)*.
   v. Some books have a *preface*.

   ```
   <!ELEMENT books (book*)> /// 3 points
   <!ELEMENT book (index, title, section+, author+, preface*)> /// 4 points
   <!ATTLIST book ISBN CDATA #REQUIRED> /// 3 points ****
   ISBN can also be defined as a tag inside book ****
   ```

3. **(5 points)** Consider the SAX and DOM methods used in parsing XML. Which one is more suitable for parsing a large XML file? Why?
   SAX - Because DOM loads the entire file into the physical memory to build its XML tree completely, and if the data collection is too large in size, it does not fit in memory.

4. **(10 points)** Explain a disadvantage in transferring a data collection in the format of JSON.
   The redundancy of attribute names that are repeated in every record.
B. Performance Optimization (35 points)

1. **(30 points)** Choose three of the following four optimization techniques. For each one of your choices, explain (1) how it is useful in making our JDBC programs faster, and (2) a challenge that one can face as a side effect of using that technique.

**Batch Inserting, Disabling Auto-commit, Caching, and Using Prepared Statements**

- **Batch Inserting:** insertion is performed as one transaction rather than several ones (so for example query processing tasks such as syntax-checking are saved). One challenge may be that the entire batch can be rejected because only one of the records is problematic.

- **Disabling Auto-commit:** database log statements are not forced to disk until commit() is called explicitly. Challenge: consistency must be handled manually by (for example) rolling back if a JDBC statement fails.

- **Caching:** if some records are already fetched from the database, if they exist in cache, we can save some database operations. Challenge: cache content may go out-of-date, or, the cache container must be managed in the case of concurrent usage.

- **Using prepared statements:** saves some query pre-processing. No specific challenge (in fact even if prepared statement is not used, most databases cache the prepared version of past statements and use them if available).
2. **(5 points)** Choose one of the four ACID transactional properties and describe what it guarantees for processing a transaction.
   
   **Atomicity**: a transaction is either performed completely, or it leaves no trace on the database at all.
   
   **Concurrency**: A transaction takes the database from a consistent state to another consistent state.
   
   **Isolation**: the execution of a transaction is independent of other transactions running concurrently.
   
   **Durability**: once a transaction is committed, its results will stay in the database permanently.

C. **Using AJAX (30 points)**

1. **(10 points)** Explain how AJAX works and why it enables us to have more responsive web pages.
   
   Using AJAX, we can send HTTP requests and process their response using a thread that works asynchronously of the main thread. The page isn’t blocked when an AJAX request is being sent. The page is not reloaded completely, and therefore it’s more responsive.

2. **(6 points)** How are the `readystate` and `onreadystatechange` properties of the `XMLHttpRequest` object used?
   
   `onreadystatechange` is set to a callback function, and this function is called whenever the `readystate` variable changes. The `readystate` property is updated on different stages of the AJAX lifecycle: for example when the request is sent, when the response arrives, when the response is loaded completely, etc.

3. **(14 points)** Discuss two reasons why using AJAX results in having less “overhead” in our websites.
   
   1- Less network overhead, because using AJAX, only the changing content needs to be transferred from the server.
   
   2- Less processing overhead in the browser, because it doesn’t need
to interpret the entire HTML page again and again.

Good luck,
Jamshid