Quiz 4: DSC 208 Data Management for Analytics

Questions and Explanations

Question 1. Which of the following is not a syntactic feature of XML documents?

a) Middle tag
b) None of the three
c) Start tag
d) End Tag

Answer: a) Middle tag

- *Explanation:* XML documents are structured using a hierarchy of elements, defined by ‘start tags‘ (e.g., ‘¡element¿‘) and ‘end tags‘ (e.g., ‘¡/element¿‘). There is no concept of a ”middle tag” in XML syntax. Well-formed XML requires every start tag to have a matching end tag (or be an empty-element tag, e.g., ‘¡element/¿‘).

Question 2. Which capability of semi-structured data models enables storing arbitrary non-normalized data unlike the relational data model?

a) Schema-later approach
b) None of the three
c) Different attributes across records
d) Nesting of records

Answer: d) Nesting of records

- *Explanation:* Semi-structured data models (like JSON or XML) allow for hierarchical or ‘nested‘ structures within a single record. This means that a single field can contain an entire sub-document or array of values, enabling the representation of complex, non-normalized data without requiring the flattening of structures into multiple tables, which is characteristic of the relational model. While ”different attributes across records” is also a feature of semi-structured data flexibility, ‘nesting of records‘ is the primary mechanism that directly allows for the arbitrary non-normalized, hierarchical data storage that contrasts sharply with the flat, normalized tables of relational databases.

Question 3. Which of these paradigms of parallelism is most common in data systems?

a) Shared disk
b) Shared CPU
c) Shared nothing
d) Shared memory

Answer: c) Shared nothing

- *Explanation:* The ‘shared nothing‘ architecture is the most prevalent paradigm for scalable, distributed data systems (like Hadoop, Spark, NoSQL databases). In this architecture, each node in the cluster has its own dedicated CPU, memory, and storage, and they communicate only by passing messages over a network. This design minimizes contention, provides high fault tolerance, and allows for near-linear scalability, as adding more nodes directly increases total resources.

Question 4. Which component of Dask divides up the work to different nodes?

a) Worker
b) Client
c) Scheduler
d) Dispatcher

Answer: c) Scheduler

- *Explanation:* In Dask’s distributed architecture, the ‘Scheduler‘ is the central component responsible for coordinating computation. It receives tasks from the ‘Client‘, builds the task graph, optimizes it, and then dispatches these tasks to the ‘Workers‘ for execution. It also monitors the workers and manages data transfer between them.

Question 5. Which data partitioning strategy enables full scalability along both the number of rows and number of columns of a matrix?

a) Row-oriented
b) Tile-oriented
c) Column-oriented
d) All of the three

Answer: b) Tile-oriented

- *Explanation:*
  - ‘Row-oriented‘ partitioning divides a matrix by rows, allowing scalability along the number of rows but not inherently along columns.

  - ‘Column-oriented‘ partitioning divides a matrix by columns, allowing scalability along the number of columns but not inherently along rows.
  - ‘Tile-oriented‘ partitioning (also known as block partitioning) divides the matrix into smaller, rectangular blocks or ”tiles.” This strategy allows for independent processing of these blocks and enables scalability in both row and column dimensions, as new rows or columns can be added, and the matrix can continue to be processed in these smaller, manageable tiles across a distributed system.

Question 6. Which capability of semi-structured data models enables storing arbitrary non-normalied data unlike the relational data model?

a) scheme-later approach

b) none of these

c) different attributes across records

d) nesting of records

Answer: d) nesting of records

- *Explanation:* This is a repeat of Question 2. As explained, the ability to embed complex structures within a single field (like lists within a field or sub-documents) directly allows for non-normalized, hierarchical data storage, which is a key distinction from relational models.

Question 7. Which data partitioning strategy enables full scalability along both the number of rows and number of columns of a matrix?

a) row-oriented

b) tile-oriented

c) column-oriented

d) all three

Answer: b) tile-oriented

- *Explanation:* This is a repeat of Question 5. As explained, ‘tile-oriented‘ partitioning divides a matrix into rectangular blocks, offering scalability in both row and column dimensions for distributed processing.