DUMPSARENA

Advanced Analytics Specialist Exam for Data Scientists

EMC E20-065

Version Demo

Total Demo Questions: 9

Total Premium Questions: 94

Buy Premium PDF

https://dumpsarena.co

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

Topic Break Down

Topic	No. of Questions
Topic 1, Data Analytics Lifecycle	3
Topic 2, Discovery	10
Topic 3, Data Preparation	23
Topic 4, Model Planning	21
Topic 5, Model Building	28
Topic 6, Communicate Results	7
Topic 7, Operationalize	1
Topic 8, Mix Questions	1
Total	94

QUESTION NO: 1

Which scenario would be ideal for processing Hadoop data with Hive?

A. Structured data, real-time processing

B. Unstructured data; batch processing

C. Unstructured data; real-time processing

D. Structured data; batch processing

ANSWER: D

Explanation:

Structured data; batch processing is the ideal scenario for Hive. Apache Hive is a data-warehouse system built for querying, summarizing, and analyzing large datasets stored in Hadoop-compatible storage. It exposes data through tables and schemas and lets users use HiveQL, a SQL-like language, to perform operations such as filtering, joining, aggregation, and reporting. Although Hive can apply schema-on-read to files in formats such as text, ORC, and Parquet, the analytical workload is fundamentally based on representing data in structured, tabular form for query processing.

Hive was designed primarily for high-throughput analytical workloads rather than low-latency interactive or event-by-event processing. Queries are executed through distributed processing engines, including MapReduce historically and Tez or Spark in many deployments, which makes it well suited to scheduled ETL, data preparation, periodic reporting, and large-scale batch analytics. For example, an organization can run a nightly Hive query to aggregate structured transaction records into daily sales totals. Apache Hive describes itself as a distributed, fault-tolerant data warehouse system for reading, writing, and managing large datasets using SQL, and its documentation details the table-oriented data model and query language: Apache Hive and Hive Language Manual.

QUESTION NO: 2

Consider dataset that resides in HDFS. Which tool natively provides the capability to run a Random Forests model against this data?

A. Mahout

B. Pig

C. Hive

D. HBase

ANSWER: A

Explanation:

Mahout is correct because Apache Mahout is a machine-learning library designed for scalable, distributed data processing in the Hadoop ecosystem. Its decision-forest implementation includes Random Forest classification and is designed to execute as a Hadoop MapReduce job. Consequently, training data stored in HDFS can be consumed directly by the distributed job, allowing the model-building workload to be partitioned and processed across cluster nodes rather than requiring the data to be moved to a separate local analytics environment.

This capability fits the question's combination of requirements: a Random Forest model and data that already resides in HDFS. Mahout's Hadoop-oriented algorithms and execution model were specifically intended to support machine-learning workloads at scale on data held in Hadoop storage. Apache Mahout describes itself as a scalable machine-learning framework, and its project materials document its Hadoop-based heritage and distributed algorithm support. See the Apache Mahout project site and the Apache Mahout documentation.

QUESTION NO: 3

What is a typical use of a UDF in Pig?

A. Creating functionality outside of what is provided by the built-in functions

B. Providing Functional access to user-defined data in HDFS

C. Providing advanced analytics to Hadoop

D. Providing an interface from Pig to Microsoft Excel for easier data manipulation

ANSWER: A

Explanation:

Creating functionality outside of what is provided by the built-in functions is a typical use of a user-defined function (UDF) in Apache Pig. Pig Latin includes a set of built-in functions and operators for common data-processing tasks, but real-world transformations frequently require domain-specific parsing, validation, enrichment, scoring, or calculations that are not available out of the box. A UDF lets a developer extend Pig Latin with that custom behavior while continuing to use Pig's declarative data-flow language. Pig UDFs can be written in several supported languages, including Java, Python, JavaScript, Ruby, Groovy, and others, then registered for use in a Pig script. Depending on the required operation, a UDF may act on individual values or tuples, filter records, load or store data in a custom format, or perform aggregate processing. This extensibility is central to Pig's role in Hadoop-based data preparation: users can combine concise Pig Latin pipelines with reusable custom code when built-in capabilities do not meet a processing requirement. See the Apache Pig documentation on user-defined functions and the Pig Latin language.

QUESTION NO: 4 - (SIMULATION)

Which problem type is best suited for simulation?

ANSWER: See the explanation for the answer

Explanation:

Answer: A complex, stochastic (random/uncertain) *what-if* problem is best suited for simulation.

Use simulation when a system has uncertain inputs, many interacting variables, or dynamic behavior and it is impractical to derive an exact analytical solution. For example, estimating queue wait times, inventory needs, financial risk, or the probability of failure under varying conditions.

QUESTION NO: 5

Why would a company decide to use HBase to replace an existing relational database?

A. It is required for performing ad-hoc queries.

B. Varying formats of input data requires columns to be added in real time.

C. The company's employees are already fluent in SQL.

D. Existing SQL code will run unchanged on HBase.

ANSWER: B

Explanation:

Varying formats of input data requires columns to be added in real time. HBase is a distributed, column-family-oriented NoSQL database that is well suited to very large, sparse datasets whose records do not all share the same attributes. Within an HBase column family, column qualifiers can be created dynamically as data is written; a predefined, fixed table-wide set of columns is not required. This allows an application to accommodate evolving or heterogeneous input records without continually altering a relational schema and coordinating schema migrations. HBase therefore provides useful flexibility when

new fields appear over time, different entities contain different attributes, or many possible attributes would make a relational design sparse and difficult to manage. The design should still select column families carefully because they are part of the physical storage and access design, but qualifiers within those families support the needed variability. Apache's HBase documentation describes the data model as tables composed of rows and column families, with columns represented by qualifiers and values, and its schema-design guidance discusses the importance of column-family design for access patterns. See the Apache HBase data model documentation and the Apache HBase schema design guidance.

QUESTION NO: 6

In a social network, what does it mean for a node to have a high degree but low betweenness?

A. The node is adjacent to a few nodes, each of each has high Page Ranks.

B. The node has the only edge connecting its community to the rest of the graph.

C. The node can be easily bypassed by communications taking other shorter paths.

D. The node acts as the hub of the graph.

ANSWER: C

Explanation:

The node can be easily bypassed by communications taking other shorter paths. A high degree means that the node has many direct connections to other nodes. However, low betweenness means that the node is included in relatively few of the shortest paths between pairs of other nodes. In practical social-network terms, the node may be well connected within a local circle or community, but it is not a critical intermediary through which information, relationships, or communication between other people must pass.

This combination commonly occurs where the node's neighbors have many connections among themselves or have alternative routes to reach the rest of the network. Even though the node has numerous contacts, those contacts can communicate through other nodes using paths that are shorter than, or equivalent to, paths through this node. Consequently, removing or avoiding the node would not substantially disrupt shortest-path communication across the graph. Degree and betweenness therefore measure different structural roles: direct popularity or connectivity versus brokerage and control over paths. NetworkX defines degree centrality as a measure based on the fraction of nodes adjacent to a node, while betweenness centrality measures the fraction of shortest paths that pass through a node. See the NetworkX degree centrality documentation and NetworkX betweenness centrality documentation.

QUESTION NO: 7

What best describes the meaning behind the phrase "Six Degrees of Separation'"?

A. Ability to use about six hops to reach any other node in an extremely large social network

B. Erdos number of all scholars having written papers with Paul Erdos

C. Maximum number of edges between nodes in a graph with a diameter of six

D. Typical distance between nodes that are connected by triadic closure

ANSWER: A

Explanation:

The correct description is **"Ability to use about six hops to reach any other node in an extremely large social network."** Six degrees of separation is the social-network concept that any two people can generally be connected through a short chain of acquaintances, conventionally expressed as no more than six intermediary social links or "hops." In graph terms, people are nodes and acquaintance relationships are edges; the phrase describes the surprisingly small path length that often exists between otherwise distant individuals in a large, highly connected social graph. The idea became widely known through Stanley Milgram's small-world experiments, which examined how messages could be passed from person to

person through personal acquaintances. Later network-science research formalized the broader “small-world” phenomenon: networks can contain enormous numbers of nodes while still having relatively short average path lengths because a limited number of bridging connections link communities together. Thus, “six” is best understood as a popular approximation for short social paths rather than a fixed technical guarantee for every pair of individuals. See the Encyclopaedia Britannica overview of six degrees of separation and Duncan Watts and Steven Strogatz’s foundational small-world network research.

QUESTION NO: 8

What advantage does replication provide while storing a file in HDFS?

A. Data protection and scheduling flexibility

B. Elimination of requirement for a combiner process

C. Elimination of requirement for Shuffle and Sort process

D. Memory optimization and minimizing tasks to run

ANSWER: A

Explanation:

Data protection and scheduling flexibility is correct because HDFS stores multiple replicas of each file block on separate DataNodes, typically with placement across different racks. This redundancy protects data availability when a DataNode, disk, or network path fails: HDFS can continue serving the block from another healthy replica and can later restore the intended replication level. Replication also improves scheduling flexibility for distributed processing. Because the same block is available on more than one node, the cluster scheduler has multiple possible locations at which it can place work that reads that block. It can preferentially schedule a task on a node holding a local replica, or at least on the same rack, reducing network traffic and improving throughput. This combination of fault tolerance, availability, and data locality is a foundational reason HDFS is designed around replicated blocks rather than a single stored copy. Hadoop’s HDFS architecture documentation describes replicated block storage and rack-aware replica placement, while its MapReduce tutorial explains scheduling map tasks near the data whenever possible. See the Apache Hadoop HDFS Architecture Guide and the Apache Hadoop MapReduce Tutorial.

QUESTION NO: 9

Which problem type is best suited for simulation?

A. One with a few. non-random input variables

B. One that has a closed-form solution

C. One with numerous, non-random Input-variables

D. One that compares "what-if" scenarios

ANSWER: D

Explanation:

One that compares “what-if” scenarios is best suited for simulation because simulation models let analysts represent a system, vary assumptions, and observe the resulting outcomes without changing the real-world system. This is especially valuable when decisions depend on uncertain inputs, interacting processes, resource constraints, timing, or complex operational rules. A model can be run repeatedly with different parameter values or operating policies, enabling a data scientist to compare the likely consequences of alternative decisions in a controlled and reproducible way.

For example, a simulation can compare staffing levels, inventory policies, routing rules, or service-capacity plans under different demand conditions. Instead of requiring a single fixed answer, it produces evidence about how outcomes such as cost, utilization, wait time, risk, or throughput change across scenarios. Repeated runs can also characterize the range and distribution of possible results, which supports decisions made under uncertainty. This scenario-based experimentation is a

core use of simulation modeling. IBM describes simulation as a technique for creating a virtual representation of a system to analyze its behavior and test changes before implementation; see IBM: What is simulation?. For additional background on discrete-event simulation and scenario analysis, see Oracle: What is discrete event simulation?.