UNIVERSITY OF MSILA
COMPUTER SCIENCE DEPARTMENT

MASTER 1 – IDO
BIG DATA
EXAM CORRECTION

11/01/2026
2:00 H

Student Information

First Name: .....

Last Name: .....

Group: .....

Final Mark

_____

Exam Instructions

- Smartphones and smart devices must be **OFF**.
- Select **one correct answer only**.
- from 1 to 40 = 0.25 / from 41 to 60 = 0.5.

1. Which condition is sufficient but not necessary for a system to be considered Big Data?
   - ☐ High ingestion velocity
   - ■ **Distributed storage**
   - ☐ Complex analytics
   - ☐ Machine learning usage

2. Data volume is large, but processing is infrequent and centralized. Which conclusion is most accurate?
   - ☐ The system is not scalable
   - ■ **Volume alone does not imply Big Data complexity**
   - ☐ Velocity is absent
   - ☐ Distributed processing is mandatory

3. Velocity in Big Data primarily refers to:
   - ☐ Query execution speed
   - ■ **Speed of data generation and arrival**
   - ☐ Visualization latency
   - ☐ Model convergence time

4. Which factor most directly increases schema complexity?
   - ☐ High data velocity
   - ■ **Data variety**
   - ☐ Replication
   - ☐ Parallelism

5. Which statement best captures the role of Value among the 5Vs?
   - ☐ It measures data size
   - ☐ It reflects ingestion speed
   - ■ **It depends on how data is exploited analytically**
   - ☐ It guarantees correctness

6. A distributed architecture is primarily adopted in Big Data systems to:
   - ☐ Improve data quality
   - ■ **Scale storage and processing**
   - ☐ Reduce schema complexity
   - ☐ Eliminate failures

7. Which workload aligns best with HDFS design?
   - ☐ Frequent record updates

- ☐ Interactive dashboards
- ■ **Large sequential scans**
- ☐ Low-latency queries

8. Why does HDFS discourage in-place file modification?
   - ☐ Replication overhead
   - ■ **Append-only file semantics**
   - ☐ Network bandwidth limits
   - ☐ Centralized scheduling
9. The primary consequence of storing many small files in HDFS is:
   - ☐ Reduced fault tolerance
   - ☐ Disk fragmentation
   - ■ **Metadata overload at the NameNode**
   - ☐ Increased replication delay
10. Random access in HDFS is:
    - ☐ Unsupported
    - ■ **Supported but inefficient**
    - ☐ Faster than sequential access
    - ☐ Required for fault tolerance
11. Which property distinguishes a Data Lake from a traditional Data Warehouse?
    - ☐ Strong ACID guarantees
    - ■ **Schema applied at read time**
    - ☐ Structured data only
    - ☐ Low-latency queries
12. Without governance, a Data Lake primarily suffers from:
    - ☐ Storage inefficiency
    - ■ **Loss of trust and usability**
    - ☐ Reduced scalability
    - ☐ Increased latency
13. Schema-on-read increases flexibility but introduces risk because:
    - ☐ Data cannot be reused
    - ■ **Interpretation depends on query logic**
    - ☐ Storage cost increases
    - ☐ Streaming becomes impossible
14. The Lakehouse architecture emerged mainly to:
    - ☐ Replace NoSQL databases
    - ■ **Bridge flexibility and reliability**
    - ☐ Eliminate ETL
    - ☐ Support unstructured data only
15. Which system is fundamentally incompatible with OLTP (Online Transactional Processing) workloads?
    - ☐ Relational DBMS
    - ■ **HDFS**
    - ☐ NewSQL engines
    - ☐ Document stores
16. MapReduce performs poorly for iterative workloads because it:
    - ☐ Is not distributed
    - ■ **Persists intermediate results to disk**
    - ☐ Uses weak consistency
    - ☐ Lacks fault tolerance
17. Spark improves execution speed mainly by:
    - ■ **Retaining intermediate data in memory**
    - ☐ Eliminating parallelism
    - ☐ Reducing input size
    - ☐ Avoiding failures
18. RDD lineage in Spark is used primarily for:
    - ☐ Load balancing
    - ☐ Scheduling
    - ■ **Fault recovery**
    - ☐ Data encryption
19. Which platform natively treats streaming and batch uniformly?
    - ☐ Hadoop
    - ☐ Spark
    - ■ **Flink**
    - ☐ Hive
20. Data ingestion focuses on:
    - ☐ Transforming data
    - ■ **Moving data into a platform**
    - ☐ Visualizing data
    - ☐ Enforcing governance
21. A data pipeline extends ingestion by:
    - ☐ Enforcing schemas only

■ **Including transformation and coordination**

☐ Replacing storage systems

☐ Guaranteeing correctness

22. ETL is preferred when:

☐ Raw data must be preserved

■ **Target systems require validated structured data**

☐ Storage is cheap

☐ Data arrives continuously

23. In ELT, transformations are postponed because:

☐ Data quality is ignored

■ **Storage scalability allows flexibility**

☐ Governance is unnecessary

☐ Streaming is unsupported

24. Sqoop achieves scalability mainly through:

☐ Compression

■ **Parallel imports**

☐ Streaming ingestion

☐ Schema inference

25. NiFi FlowFiles encapsulate:

☐ Only metadata

☐ Only content

■ **Content and attributes**

☐ Temporary tables

26. A NiFi Processor is responsible for:

☐ User authentication

■ **Acting on FlowFiles**

☐ Metadata storage

☐ Cluster coordination

27. Kafka differs from classical queues because it:

☐ Discards messages after consumption

■ **Retains data independently of consumers**

☐ Breaks ordering

☐ Requires batch ingestion

28. Kafka partitions mainly provide:

☐ Security

☐ Consistency

■ **Parallelism**

☐ Compression

29. Inconsistent analytics over clean ingested data often indicate:

☐ Replication failure

■ **Schema-on-read ambiguity**

☐ High throughput

☐ Parallel execution

30. Big Data systems do NOT inherently guarantee:

☐ Fault tolerance

☐ Scalability

■ **Correct decisions**

☐ Distributed processing

31. A system filters raw data aggressively during ingestion. Which Big Data property is most reduced?

☐ Velocity

☐ Veracity

■ **Value**

☐ Variety

32. Horizontal scaling without parallel execution implies that:

☐ The system is not distributed

■ **Scalability does not ensure performance**

☐ Fault tolerance is impossible

☐ Big Data requirements are met

33. Schema-on-read requires governance mainly to:

☐ Reduce storage cost

■ **Prevent conflicting analytical interpretations**

☐ Improve throughput

☐ Enable streaming

34. Increasing HDFS block size improves performance by:

☐ Reducing metadata size

■ **Optimizing sequential I/O**

☐ Improving random access

☐ Increasing concurrency

35. Which design choice makes HDFS unsuitable for transactional updates?

☐ Replication

☐ Rack awareness

■ **Write-once semantics**

☐ Heartbeats

36. A Data Lake without lineage information leads to:
    - ☐ Faster queries
    - ■ **Loss of auditability**
    - ☐ Stronger consistency
    - ☐ Reduced storage

37. Regulatory compliance most strongly motivates:
    - ☐ ELT
    - ■ **ETL**
    - ☐ Streaming analytics
    - ☐ Data lakes

38. ELT delays transformations mainly because:
    - ☐ Engines are slower
    - ■ **Storage is cheaper than compute**
    - ☐ Governance is ignored
    - ☐ Streaming is mandatory

39. Sqoop is ineffective for real-time ingestion because it:
    - ☐ Lacks scalability
    - ■ **Operates in batch mode**
    - ☐ Does not support SQL
    - ☐ Cannot handle failures

40. NiFi back-pressure exists to:
    - ☐ Increase latency
    - ☐ Encrypt data
    - ■ **Protect downstream components**
    - ☐ Enforce schemas

41. NiFi avoids data loss between processors by:
    - ☐ Bulletins
    - ☐ Controller services
    - ■ **Persistent queues**
    - ☐ Clustering

42. Kafka ordering is guaranteed:
    - ☐ Globally
    - ☐ Per topic
    - ■ **Per partition**
    - ☐ Across consumers

43. Kafka retention enables:
    - ☐ Automatic deletion
    - ■ **Event replay**
    - ☐ Exactly-once delivery
    - ☐ Low latency

44. Kafka tolerates which failure without data loss?
    - ☐ All brokers failing
    - ■ **Consumer crash after fetch**
    - ☐ Network partition without replicas
    - ☐ Producer crash before send

45. Streaming is preferred when:
    - ☐ Data is small
    - ☐ Storage is limited
    - ■ **Immediate response is required**
    - ☐ Data is structured

46. Treating batch as streaming implies:
    - ☐ Batch systems are obsolete
    - ■ **Unified execution semantics**
    - ☐ Lower cost
    - ☐ Stronger consistency

47. Governance enforcement belongs primarily to:
    - ☐ Ingestion
    - ☐ Storage
    - ■ **Management layer**
    - ☐ Visualization

48. An auditable pipeline requires:
    - ☐ High throughput
    - ☐ Parallelism
    - ■ **Lineage and metadata**
    - ☐ Streaming

49. Which belief is a Big Data fallacy?
    - ☐ Architecture matters
    - ☐ Data quality matters
    - ■ **More data guarantees better outcomes**
    - ☐ Governance is necessary

50. High throughput ingestion with weak governance primarily increases the risk of:
    - ☐ Data loss
    - ■ **Semantic inconsistency**
    - ☐ Low availability
    - ☐ High latency

51. Exactly-once semantics are hardest to guarantee in:

- ☐ Batch processing
- ■ **Distributed streaming systems**
- ☐ Centralized databases
- ☐ Offline analytics

52. Replication in distributed storage primarily improves:
    - ☐ Latency
    - ■ **Fault tolerance**
    - ☐ Schema flexibility
    - ☐ Throughput predictability

53. A system prioritizing availability over consistency follows:
    - ☐ ACID
    - ■ **BASE**
    - ☐ OLTP
    - ☐ ETL

54. Centralized metadata services become bottlenecks mainly because they:
    - ☐ Store data blocks
    - ■ **Handle namespace and coordination**
    - ☐ Execute computations
    - ☐ Enforce schemas

55. Late-arriving data in streaming pipelines mainly challenges:
    - ☐ Throughput
    - ■ **Correctness of time-based analytics**
    - ☐ Fault tolerance
    - ☐ Storage scalability

56. Event-time processing differs from processing-time because it:
    - ☐ Reduces latency
    - ■ **Accounts for data arrival delays**
    - ☐ Eliminates watermarks
    - ☐ Requires batch execution

57. A unified batch–streaming engine simplifies architectures by:
    - ☐ Removing storage layers
    - ■ **Sharing execution semantics**
    - ☐ Eliminating governance
    - ☐ Avoiding failures

58. Data lineage is most critical when:
    - ☐ Data volume increases
    - ☐ Storage is cheap
    - ■ **Results must be explainable and auditable**
    - ☐ Ingestion is fast

59. Which layer should detect policy violations?
    - ☐ Compute layer
    - ☐ Storage layer
    - ■ **Governance layer**
    - ☐ Visualization layer

60. A Big Data platform optimized only for throughput risks:
    - ☐ Underutilization
    - ■ **Loss of interpretability and control**
    - ☐ Reduced scalability
    - ☐ Increased consistency