# SDE Quiz - 3

1. When comparing file systems and DBMS, which factors best explain *why DBMS consumes more storage space* for the same data size?
   A. It pre-allocates schema and metadata
   B. It stores data redundantly across multiple drives
   C. It maintains statistical and optimization data
   D. It compresses data to improve access time
   **Answer:** A, C

2. A system designer decides to store medical records in plain files for space efficiency but finds query performance too low. Which trade-off is evident here?
   A. Time efficiency traded for space efficiency
   B. Consistency traded for concurrency
   C. Reliability traded for heterogeneity
   D. Persistence traded for scalability
   **Answer:** A

3. Why can't file systems completely replace DBMS even though both use B+ trees?
   A. File systems lack query optimization layers
   B. DBMS supports multiple index types
   C. File systems cannot handle transactional integrity
   D. File systems store data only in row-oriented format
   **Answer:** A, B, C

4. Which principle best summarizes why DBMS trades *space for time*?
   A. It stores redundant metadata for faster query optimization
   B. It reduces I/O by removing indexes
   C. It allocates more disk space to reduce computation cost during execution
   D. It compresses data to reduce query execution time
   **Answer:** C

5. During query execution, which DBMS components collaborate to ensure both correctness and efficiency?
   A. Query parser
   B. Scheduler
   C. Recovery manager

D. File management layer
**Answer:** A, B, C, D

6. The “Empty Table Phenomenon” demonstrates which architectural design trade-off?
   A. Overhead from pre-allocating schema and statistics
   B. Inefficient memory paging
   C. Lack of metadata optimization
   D. Poor concurrency control
   **Answer:** A

7. In which scenarios would a **hybrid database** (SQL + NoSQL) be most beneficial?
   A. E-commerce platforms with product metadata and real-time transactions
   B. Static government record archives
   C. Scientific data pipelines with schema evolution
   D. Systems requiring both ACID and BASE guarantees
   **Answer:** A, C, D

8. Which aspect of UML parallels the concept of “entities” in ER models?
   A. Classes
   B. Methods
   C. Namespaces
   D. Interfaces
   **Answer:** A

9. How does transaction management differ between relational and NoSQL databases?
   A. RDBMS supports multi-table atomicity
   B. NoSQL ensures ACID compliance across collections
   C. NoSQL typically supports document-level transactions
   D. RDBMS transactions guarantee isolation and durability
   **Answer:** A, C, D

10. Which scenario violates ACID but satisfies BASE?
    A. A social media post appearing before final consistency
    B. A banking transfer rolling back successfully

C. A file write operation with journaling
D. A transactional invoice update
**Answer:** A

11. Which database choices align best with these use cases?
    - (i) Real-time bidding system
    - (ii) Student information system
    - (iii) Log analytics platform

A. Redis → (i)
B. PostgreSQL → (ii)
C. Hadoop-based Key-Value → (iii)
D. XML DB → (i)
**Answer:** A, B, C

12. The “Curse of Dimensionality” primarily impacts which design dimensions?
A. Index efficiency
B. Query discrimination
C. Schema normalization
D. Data sparsity
**Answer:** A, B, D

13. What design patterns enable reliability in DBMS ecosystems?
A. Recovery manager redundancy
B. Predictable query optimization strategies
C. Static schema enforcement
D. Error detection through metadata auditing
**Answer:** A, B, D

14. Why are **B+ trees** preferred for file system and DBMS indexing over **B-trees**?
A. Fixed node depth ensures predictable performance
B. Data stored only in leaf nodes
C. Higher branching factor for faster traversal
D. Reduced best-case access time
**Answer:** A, B, C

15. Which type of data system is best suited for applications needing *flexibility* and *semi-structured* storage?
    A. Graph Database
    B. Relational DB
    C. Key-Value Store
    D. XML Database
    **Answer:** D

16. Which statements about Key-Value stores are true in analytical workloads?
    A. Excellent for high-volume writes
    B. Suitable for single-record ACID transactions
    C. Efficient for distributed analytical reads
    D. Inefficient for value-based searches
    **Answer:** A, C, D

17. Which database types best match their **strengths**?
    A. XML DB → Schema flexibility
    B. Key-Value → Write optimization
    C. RDBMS → Complex relationship handling
    D. Graph DB → Transactional reliability
    **Answer:** A, B, C

18. What limitations make relational databases unsuitable for rapidly changing schemas?
    A. Schema must be predefined
    B. High restructuring overhead
    C. Rigid constraint enforcement
    D. Lack of support for polymorphic records
    **Answer:** A, B, C, D

19. Which database selection would best fit a financial transaction system?
    A. ACID-compliant RDBMS
    B. BASE-oriented NoSQL
    C. Key-Value store
    D. Graph database
    **Answer:** A

20. When comparing **XML** and **Key-Value** databases, which distinctions are correct?
    A. XML supports semi-structured hierarchy
    B. Key-Value stores are schema-less
    C. XML supports path-based navigation
    D. Key-Value supports relational joins
    **Answer:** A, B, C

21. Which property of B+ trees ensures consistent access time?
    A. Data stored in all nodes
    B. Uniform leaf depth
    C. Variable branching factor
    D. Reduced pointer count
    **Answer:** B

22. Which indicators help detect relational schema self-descriptiveness issues?
    A. Missing timeline/audit data
    B. Inability to distinguish derived tables
    C. Lack of metadata for indexes
    D. Redundant key duplication
    **Answer:** A, B

23. In MapReduce, which factors influence whether a workload is write-heavy or read-heavy?
    A. Sorting requirements
    B. Data shuffling strategy
    C. Key distribution
    D. Output aggregation frequency
    **Answer:** A, B, C, D

24. A log analytics system needs fast writes and occasional aggregate reads. Which DB type is best?
    A. XML Database
    B. Graph Database
    C. RDBMS
    D. Key-Value Store
    **Answer:** D

25. Which key challenges cause scalability failure in RDBMS systems?
    A. Non-linear storage growth
    B. Metadata overhead
    C. Index explosion
    D. Weak consistency mechanisms
    **Answer:** A, B, C

26. Which of these represent *trade-offs* in database technology selection?
    A. ACID vs. BASE
    B. Scalability vs. Consistency
    C. Space vs. Time
    D. Declarative vs. Imperative programming
    **Answer:** A, B, C, D

27. In a DBMS, which component is responsible for interpreting user queries into execution plans?
    A. Query Parser and Processor
    B. File Manager
    C. Buffer Manager
    D. Recovery Manager
    **Answer:** A

28. You create hundreds of empty tables for a multi-tenant prototype and storage spikes. Which design choices explain the spike and help avoid it?
    A. Each empty table preallocates index structures and stats.
    B. Use a shared table with tenant_id column instead of many tables.
    C. Turn off query optimization statistics globally.
    D. Use partitioning or a single JSON/document column per tenant.
    **Answer:** A, B, D

29. You must design a system that handles banking transactions (strict consistency), user profiles with dynamic fields, and high-throughput event logs. Which hybrid combination is most appropriate?
    A. RDBMS (ACID) for transaction processing
    B. Document store (schema-flexible) for user profiles
    C. Key-value store for event logs
    D. Graph DB for event logs
    **Answer:** A, B, C

30. Which locking strategy in an XML DB reduces contention for updates to deep nodes while preserving parent-child integrity?
    A. Path locking entire root-to-leaf path for every update
    B. Node-level locking with subtree extension for parents
    C. Global database lock during each update
    D. No locking, rely on optimistic concurrency without checks
    **Answer:** B

31. You need to design a session store with very fast reads and occasional bulk analytics on sessions. Which design decisions balance performance and analytics needs?
    A. Keep sessions in an in-memory key-value store with TTLs.
    B. Periodically snapshot sessions to a distributed key-value store with time-partitioning for analytics.
    C. Always store sessions directly in RDBMS to simplify analytics.
    D. Emit session events to a log system (MapReduce/Spark) for batch aggregation.
    **Answer:** A, B, D

32. You must design a remote island data center (limited power, intermittent connectivity, hurricane risk) for a mixed workload. Which single strategy best mitigates both availability and data safety risks?
    A. Use a hybrid cloud architecture with local DB for fast ops and remote replicas when connectivity allows.
    B. Centralize everything in a single powerful server on the island.
    C. Use in-memory databases only to save power.
    D. Rely solely on NoSQL eventual-consistency stores to reduce replication overhead.
    **Answer:** A

33. A database optimized for *read-heavy workloads* would likely adopt which design decisions?
    A. Denormalized tables
    B. Column-oriented storage
    C. Frequent index rebuilding
    D. Precomputed materialized views
    **Answer:** A, B, D

34. Which database property most negatively impacts write performance in a traditional RDBMS?
    A. Normalization

B. Lack of indexing
C. ACID transaction enforcement
D. Row-oriented storage
**Answer:** C

35. You're designing a database for IoT sensor logs with **billions of daily inserts** but infrequent reads. Which architectural patterns fit best?
A. Append-only log architecture
B. In-memory cache for recent writes
C. Extensive secondary indexing
D. NoSQL key-value model
**Answer:** A, B, D

36. A "both read and write not friendly" system is often a result of:
A. Poor schema design
B. Over-normalization
C. Over-indexing with high maintenance overhead
D. Balanced indexing and normalization
**Answer:** A, B, C

37. You're asked to justify constraints for a *write-heavy, read-light* system. Which constraints are likely to be **relaxed or modified**?
A. Immediate foreign key enforcement
B. Strict ACID durability
C. Index creation on non-primary attributes
D. Eventual consistency adoption
**Answer:** A, B, C, D

38. Which real-world system is **read-heavy but write-light**?
A. E-commerce order system
B. Social media feed viewer
C. Stock trading order book
D. IoT sensor data stream
**Answer:** B

39. To achieve *write efficiency* while maintaining acceptable query performance, a schema designer might:
    A. Use partitioning or sharding
    B. Avoid multi-table joins
    C. Normalize all attributes into separate tables
    D. Use batch writes instead of individual commits
    **Answer:** A, B, D

40. Match these real-world systems with their likely database category:
    A. Banking core system → Both read/write friendly
    B. Search engine index → Read friendly, write unfriendly
    C. Log collection system → Write friendly, read unfriendly
    D. Historical archive → Both read/write unfriendly
    **Answer:** A, B, C, D