PDF 4: Performance Monitoring & Optimization

Subjective Case Study Questions and Answers

Q1. Optimizing Query Performance in Synapse
Scenario: A retail analytics team reports slow queries on large sales tables.
Answer: Use clustered columnstore indexes for fact tables, partition tables by date, create materialized views for frequently queried aggregations, and ensure statistics are updated. Evaluate Dedicated SQL Pool workload distribution.

Q2. Monitoring Pipeline Performance in ADF
Scenario: Pipelines are taking longer than expected.
Answer: Use ADF monitoring dashboard, enable pipeline diagnostics, check activity run duration, and parallelism. Optimize copy activities by adjusting batch size, source/destination integration runtime, and use staging if needed.

Q3. Performance Issues with Semi-Structured Data
Scenario: JSON datasets are processed slowly in Synapse.
Answer: Flatten JSON at ingestion using Mapping Data Flow, avoid repeated cross-joins, pre-aggregate where possible, and partition tables properly. Consider using serverless SQL for ad-hoc queries to reduce load on Dedicated SQL Pools.

Q4. Cost vs Performance Trade-offs
Scenario: Query performance needs to improve without increasing costs.
Answer: Evaluate workload management (DWUs) in Synapse, scale-out only during peak loads, use materialized views selectively, leverage serverless SQL for non-critical ad-hoc workloads, and archive cold data.

MCQs

1. Best index type for large fact tables?
   Answer: B. Clustered columnstore index — optimized for analytics.
2. How to monitor ADF pipelines?
   Answer: A. ADF Monitoring Dashboard — activity runtime and failures.
3. Which technique improves JSON ingestion?
   Answer: C. Flatten transformation — simplifies semi-structured data.
4. Reducing query cost while maintaining performance?
   Answer: B. Scale DWUs dynamically — pay only during heavy workloads.
5. Frequent query optimization method?
   Answer: D. Materialized Views — precomputed results.
6. Detect pipeline bottlenecks?
   Answer: A. Activity run diagnostics + parallelism analysis.
7. When to use serverless SQL?
   Answer: B. Ad-hoc queries on external data — avoid Dedicated SQL Pool load.
8. Partitioning tables improves:
   Answer: C. Query performance — reduces scanned data.