

DP-300 Case Study – Advanced Monitoring & Query Optimization

Participant Reading Case Study

Participant Instructions

This case study is designed by the trainer to simulate a **real-world, complex Azure SQL production environment**.

You are expected to **analyze symptoms, identify root causes, and design solutions** based on DP-300 concepts.

No answers or hints are provided. Be prepared to justify every decision.

1. Organization Context

Aquila Logistics Systems (ALS) is a large supply-chain and logistics platform supporting:

- Real-time shipment tracking
- Warehouse inventory updates
- Partner-facing analytics dashboards

ALS operates across multiple regions in India and Southeast Asia.

- Average Daily Transactions: ~9 million
 - Peak Concurrency: Very high during end-of-day reconciliation
 - Data Pattern: Heavy writes with unpredictable read spikes
-

2. Azure SQL Environment

The platform uses the following Azure SQL configuration:

- Azure SQL Database (Elastic Pool)
- Pool Service Tier: General Purpose
- Total Pool vCores: 16
- Number of Databases in Pool: 6
- One primary OLTP database consumes the majority of workload
- Zone Redundancy: Enabled

Workloads

- Continuous OLTP write operations
 - Scheduled analytics queries every 30 minutes
 - Ad-hoc reporting by the operations team
-

3. Business Expectations

The business has defined the following expectations:

- No noticeable performance degradation for OLTP workloads
- Analytics queries must complete within 5 minutes
- Infrastructure cost must remain stable
- Platform should self-heal with minimal manual intervention

Some of these expectations may conflict with one another.

4. Situation After Go-Live

After approximately six weeks in production, multiple teams begin reporting performance issues. Observations vary across teams and time windows.

You are assigned as the **Senior Azure SQL Administrator** to assess and stabilize the platform.

PROBLEM AREA 1 – Confusing Resource Signals

Observed Symptoms

- Elastic pool CPU averages around **55–60%**
- Individual database CPU intermittently spikes to **95%**
- DTU/vCore alerts trigger inconsistently
- Some dashboards show normal behavior while applications experience timeouts

Participant Task

Analyze how resource utilization should be interpreted in an **elastic pool environment**, and identify what additional monitoring is required.

PROBLEM AREA 2 – Inconsistent Query Performance

Observed Symptoms

- The same reporting query runs quickly during morning hours but slows significantly at night
- Query Store shows multiple execution plans for the same query
- No application deployments or schema changes occurred during slow periods

Participant Task

Determine why query performance varies and how stability can be achieved.

PROBLEM AREA 3 – Scaling That Did Not Fix the Problem

Observed Symptoms

- The elastic pool was scaled from 16 to 24 vCores
- Performance improved briefly, then degraded again
- Monthly Azure costs increased noticeably

Participant Task

Explain why scaling did not permanently resolve the issue and what should have been addressed first.

PROBLEM AREA 4 – Blocking Without Obvious Locks

Observed Symptoms

- Blocking is reported during analytics execution windows
- No long-running transactions are visible
- Database uses default READ COMMITTED isolation

Participant Task

Identify possible causes of blocking in this scenario and how you would confirm them.

PROBLEM AREA 5 – Monitoring Blind Spots

Observed Symptoms

- Alerts are triggered only after users report issues
- Metrics are reviewed in isolation, without correlation
- No historical performance baseline exists

Participant Task

Design a monitoring approach that allows issues to be detected **before** user impact.

PROBLEM AREA 6 – Long-Term Optimization Strategy

Observed Symptoms

- Manual performance tuning is reactive
- DBA team struggles to keep up with incidents

Participant Task

Propose a long-term strategy for continuous monitoring and optimization aligned with Azure SQL best practices.

5. Constraints

You must work within the following constraints:

- No immediate application code changes are allowed
 - Downtime must be minimal
 - Azure-native tools only
 - Cost stability is a key requirement
-

6. Discussion Expectations

Be prepared to:

- Explain your reasoning clearly
- Map your decisions to DP-300 concepts
- Defend why alternative approaches were not chosen

Multiple designs may be acceptable if they meet the business requirements.

End of Case Study – Participant Edition