DP-300 Mega Case Study (Participant Edition)

Enterprise-Scale Monitoring, Performance, and Operational Optimization in Azure SQL

> **Participant Instructions**
> This is a **trainer-created, enterprise-grade mega case study**.
> It is intentionally **long, theory-heavy, and complex** and is designed to take **at least 60–90 minutes** to analyze and solve.
> There are **no answers, hints, or shortcuts**. You are expected to read carefully, correlate information across sections, and apply DP-300 concepts from the **final modules** of the course.

1. Organization Background

Helios One Commerce Platform (HOCP) is a multi-tenant B2C and B2B commerce ecosystem operating across India, the Middle East, and Southeast Asia.

HOCP supports: - Online retail transactions - Vendor onboarding and settlements - Real-time inventory synchronization - Internal analytics and forecasting

The platform operates **24×7** and experiences extreme load variability.

2. Business and Technical Expectations

The executive leadership has defined the following expectations:

- The platform must remain available during peak traffic events
- Performance degradation must be detected *before* customer impact
- Query performance must remain consistent regardless of data growth
- Operational overhead must reduce year-over-year
- Cloud costs must be predictable and justified

Failure to meet these expectations results in revenue loss and SLA penalties.

3. Data Characteristics

The primary transactional database has the following characteristics:

- Current size: ~8 TB
- Annual growth rate: ~35%
- Tables with billions of rows

- Mix of short OLTP transactions and long analytical queries
- Highly skewed data distribution (hot and cold data)

Historical data is rarely modified but frequently queried for reporting.

4. Azure SQL Architecture

The platform uses a **hybrid Azure SQL design**:

4.1 OLTP Layer

- Azure SQL Managed Instance
- Business Critical tier
- 16 vCores
- Read scale-out enabled
- Private endpoints only

4.2 Reporting Layer

- Azure SQL Database (Elastic Pool)
- General Purpose tier
- 24 vCores shared across 5 databases
- Used by analytics and reporting teams

4.3 Supporting Services

- Azure Monitor
- Log Analytics workspace
- Azure DevOps for database deployments

5. Operational Reality (After 3 Months)

Despite a well-planned migration, multiple issues begin to surface.

Different teams report **conflicting and sometimes misleading symptoms**.

You are brought in as a **Principal Azure SQL Architect** to perform a deep assessment.

SECTION A – Monitoring Complexity and Blind Spots

A.1 Observed Metrics

- Average CPU on Managed Instance: ~60%
- CPU spikes to 95% during flash sales

- Memory usage remains consistently high
- Storage latency increases without clear correlation to traffic
- Elastic pool metrics appear healthy at the pool level

A.2 Observed Problems

- Alerts fire only after application failures
- Metrics are reviewed in isolation
- No shared understanding of "normal" behavior

Participant Task A

Design a **comprehensive monitoring strategy** that: - Distinguishes between normal and abnormal behavior - Correlates metrics across services - Detects slow degradation, not just spikes

---

SECTION B – Query Performance at Scale

B.1 Observed Symptoms

- Some queries show exponential performance degradation over time
- Identical queries have drastically different execution times
- Query Store contains thousands of plans
- Plan forcing was attempted but later reverted

B.2 Additional Context

- Parameter values vary significantly by tenant
- Statistics updates are inconsistent
- Index fragmentation grows rapidly

Participant Task B

Analyze how **Query Store, execution plans, and statistics** should be used together to: - Identify root causes - Stabilize performance - Prevent future regressions

---

SECTION C – Read vs Write Contention

C.1 Observed Symptoms

- OLTP latency increases during reporting windows
- Blocking is reported, but long-running transactions are rare
- Read scale-out is enabled, yet contention persists

C.2 Additional Observations

- Some reports use SELECT * across large tables

- Index usage patterns vary significantly

Participant Task C

Determine why **read scale-out alone** did not eliminate contention and what architectural or operational changes are required.

SECTION D – Elastic Pool Paradoxes

D.1 Observed Symptoms

- Elastic pool CPU averages at ~50%
- Individual reporting databases experience throttling
- Scaling the pool improves performance temporarily
- Costs increase without proportional benefit

Participant Task D

Explain how **elastic pool resource governance** works and how noisy-neighbor effects should be identified and mitigated.

SECTION E – Automatic vs Manual Optimization

E.1 Observed Situation

- DBAs manually tune indexes weekly
- Automatic tuning is disabled due to fear of regressions
- Performance improvements are short-lived

Participant Task E

Evaluate the role of **Automatic Tuning and Intelligent Performance** features and design a safe adoption strategy.

SECTION F – Long-Term Operational Strategy

F.1 Strategic Challenges

- Data volume continues to grow rapidly
- DBA team size remains constant
- Performance incidents are becoming more frequent

Participant Task F

Design a **long-term operational model** that: - Reduces manual effort - Improves predictability - Scales with data growth

---

6. Constraints

- No major application rewrites allowed
- Downtime must be minimal
- Azure-native solutions only
- Cost optimization is mandatory

---

7. Final Discussion Expectations

You should be prepared to:

- Correlate symptoms across sections
- Justify trade-offs
- Explain why some “obvious” fixes are incorrect
- Map solutions explicitly to DP-300 exam objectives

This case study has **no single correct answer**, but weak designs will fail under scrutiny.

End of Mega Case Study – Participant Edition