Question 1: Case Studies . . . . . [200 points]

(i) **[10 points] In-Memory Database Systems:**
Distinguish between logging and recovery in: (1) disk-oriented and (2) in-memory DBMSs.

(ii) **[10 points] In-Memory Database Systems:**
Why are in-memory DBMSs still stymied by the slow sync time of non-volatile storage?

(iii) **[10 points] Logical Logging:**
Explain why it is hard to implement recovery with logical logging if you have concurrent txns.

(iv) **[10 points] Azure:**
How does Azure SQL use the same data structure for MVCC and logging?

(v) **[10 points] Azure:**
How does Azure SQL support constant-time recovery?

(vi) **[10 points] Azure:**
How does Azure undo updates without having to process undo records in WAL?

(vii) **[10 points] Azure:**
Why does Azure store versions from **all** tables are stored in a single table?

(viii) **[10 points] Azure:**
Why is delta record space is not pre-allocated per tuple?

(ix) **[10 points] Azure:**
Distinguish between: (1) in-row and (2) off-row versioning.

(x) **[10 points] SiloR:**
Explain why SiloR uses OCC.

(xi) **[10 points] SiloR:**
Explain how SiloR parallelizes: (1) logging, (2) checkpointing, and (3) recovery.

(xii) **[10 points] SiloR:**
Explain the purpose of "epoch".

(xiii) **[10 points] SiloR:**
What is a persistent epoch?

(xiv) **[10 points] SiloR:**
Why does SiloR process log in reverse order?

(xv) **[10 points] Checkpoints:**
Distinguish between: (1) complete and (2) delta checkpoints.

(xvi) **[10 points] Checkpoints:**
Distinguish between: (1) time-based and (2) log file size-based checkpointing.

(xvii) **[10 points] Checkpoints:**
Why do copy-on-update checkpoints work well with MVCC?

Question 1 continues…

(xviii) **[10 points] Facebook Scuba:**
Explain how Scuba supports fast restarts.

(xix) **[10 points] Facebook Scuba:**
Explain how Scuba uses shared memory.

(xx) **[10 points] Facebook Scuba:**
Distinguish between: (1) heap memory and (2) shared memory.