

Feedback — Optional Week 7 Quiz

[Help Center](#)

You submitted this quiz on **Tue 17 Feb 2015 12:30 PM PST**. You got a score of **5.50** out of **6.00**. However, you will not get credit for it, since it was submitted past the deadline.

Question 1

Which of the following are true of NoSQL databases? (check all that apply)

Your Answer	Score	Explanation
<input type="checkbox"/> Most NoSQL databases are built on top of SQL databases	✓ 0.25	
<input checked="" type="checkbox"/> At a very basic level, they are similar to hash tables	✓ 0.25	
<input type="checkbox"/> NoSQL databases are fast enough that developers do not have to think about how to optimize for them	✓ 0.25	
<input checked="" type="checkbox"/> They do not support any query languages	✗ 0.00	
Total	0.75 / 1.00	

Question Explanation

See: Section 5 Module 2 Part 1: NoSQL Databases

Question 2

What is database normalization? (check all that apply)

Your Answer	Score	Explanation
<input type="checkbox"/> The process of applying standard interfaces to a database for interoperability	✓ 0.25	

<input type="checkbox"/> The process of ensuring that all data values in the database are based on the same scale	✓	0.25
<input type="checkbox"/> The process of calculating the normal value of the database vector for optimization	✓	0.25
<input checked="" type="checkbox"/> The process of ensuring that a piece of data is not duplicated multiple times in a database	✓	0.25
Total		1.00 / 1.00

Question Explanation

See: Section 5 Module 2 Part 1: NoSQL Databases

Question 3

What is the role of denormalization in NoSQL database optimization? (check all that apply)

Your Answer	Score	Explanation
<input type="checkbox"/> Denormalization is used to make reads faster by ensuring that all queries always go to the same host	✓ 0.25	
<input checked="" type="checkbox"/> Denormalization can be used to make reads faster by duplicating data and requiring fewer key-based lookups	✓ 0.25	
<input type="checkbox"/> Denormalization is used to make writes faster by ensuring that there is high write contention	✓ 0.25	
<input type="checkbox"/> Denormalization can be used to make writes faster by ensuring that data is always backed up	✓ 0.25	
Total		1.00 / 1.00

Question Explanation

See: Section 5 Module 2 Part 1: NoSQL Databases

Question 4

Which of the following are ways to optimize reads for specific queries? (check all that apply)

Your Answer	Score	Explanation
<input checked="" type="checkbox"/> Grouping all data needed for a specific query and attaching it to a single key	✓ 0.25	
<input type="checkbox"/> Not grouping data under the same key to avoid single key lookups	✓ 0.25	
<input type="checkbox"/> Grouping all needed data under the same namespace to improve temporal locality	✓ 0.25	
<input type="checkbox"/> Grouping all needed data under multiple keys to ensure that writes are parallelized	✓ 0.25	
Total	1.00 / 1.00	

Question Explanation

See: Section 5 Module 2 Part 2: Optimizing for Query Patterns

Question 5

How does a sharded counter work? (check all that apply)

Your Answer	Score	Explanation
<input type="checkbox"/> A counter is broken into multiple shards that are each stored under a different key. When the counter is read, each shard is fetched and the sum of the counter shards is returned. On writes, all shards are fetched and incremented. Write parallelism is proportional to the number of shards.	✓ 0.25	
<input checked="" type="checkbox"/> A counter is broken into multiple shards that are each stored under a different key. When the counter is read, each shard is fetched and the sum of the counter shards is returned. On writes, a counter shard is randomly chosen and incremented. Write parallelism is proportional to the number of shards.	✓ 0.25	

☐ A counter is broken into multiple shards that are each stored under the same exact key. When the counter is read, a random shard is fetched and the value is returned. On writes, a counter shard is randomly chosen and incremented. Write parallelism is proportional to the number of shards. ✓ 0.25

☐ A counter is broken into multiple shards that are each stored under a different key. When the counter is read, each shard is fetched and the sum of the counter shards is returned. On writes, at least $\log(N)$ shards are randomly chosen and incremented. Write parallelism is proportional to the log of the number of shards. ✓ 0.25

Total	1.00 /
	1.00

Question Explanation

See: Section 5 Module 2 Part 4: Write Contention & Sharding

Question 6

What is the relationship between read and write optimization in NoSQL databases? (check all that apply)

Your Answer	Score	Explanation
<input type="checkbox"/> Many write optimization strategies require denormalization, which always increases read throughput	✓ 0.25	
<input type="checkbox"/> Many read optimization strategies require denormalization, which always increases write throughput	✓ 0.25	
<input type="checkbox"/> Many write optimization strategies require denormalization, which can increase read complexity	✗ 0.00	
<input checked="" type="checkbox"/> Many read optimization strategies require denormalization, which can increase write complexity	✓ 0.25	
Total	0.75 /	
	1.00	

Question Explanation

See: Section 5 Module 2 Part 3: Optimizing for Reads vs. Writes