The Revised Bloom’s Taxonomy: A Crash Course

Assessment questions need to be written using the guidelines outlined in the revised Bloom’s Taxonomy. According to the revised Bloom’s Taxonomy, there are different levels of cognitive behavior that can explain thinking skills and abilities used in learning. In order for our assessments to accurately measure skill level, we ask that you write questions from each of the following levels:

- Remember
- Understand
- Apply
- Analyze

Remember-level questions validate whether or not the learner has the ability to recall memorized facts, terms, and basic concepts.

- Language in these types of questions could include:
  - What is the definition of… ?
  - What is a characteristic of… ?
  - What is true of… ?

Remember-level Sample Question
Question: What is the AWS Simple Monthly Calculator? ● A tool that allows customers to estimate the monthly cost of running their applications in the Amazon cloud. ● An interface that lets you visualize and understand the costs of your prior AWS bills. ● A tool that calculates the resource usage that a customer's application will require every month. ● A tool that provides monthly suggestions in how to better manage your account's AWS environment.
This can be answered by having simply memorized the formula. Remember-level questions are best used for fundamental concepts, facts, or terms. Be careful to avoid trivial facts that most people would simply look up. Click for additional examples

Understand-level questions validate whether or not the learner has the ability to explain the meaning of facts, terms, and concepts.

- Language in these types of questions could include:
  - If A happens, what happens to B?
  - What is an example of… ?
  - Which procedure applies to… ?

Understand-level Sample Question

Question: What is the difference between CloudFront and ElastiCache?

- CloudFront delivers content from Edge Locations, whereas ElastiCache uses in-memory data stores.
- CloudFront provides durable storage, whereas ElastiCache is more ephemeral.
- CloudFront is best used for infrequently accessed data, whereas ElastiCache is for frequently accessed data.
- CloudFront is ideal for serving videos, whereas ElastiCache is more useful for serving images.

This requires the test-taker to know a bit more about the fundamental accounting equation. The test-taker doesn't necessarily have to be able to apply it, but they need a deeper understanding of how it works. Click for additional examples

Apply-level questions validate whether or not the learner has the ability to perform work tasks using facts, concepts, techniques, and rules.

- Language in these types of questions could include:
  - What would you do to…?
  - Given a scenario, what is the next step to the stated problem?
  - Given a problem, what is the solution?

Apply-level Sample Question

Question: An Azure admin needs to create a new role copied from the built-in role, Reader. As the first step in the process, what is one way this can be done?

- In Azure PowerShell, issue the command: `Get-AzureRmRoleDefinition -Name "Reader" | ConvertTo-Json | Out-File D:\role.json`
- In Azure CLI, issue the command: `az role definition list --name "Reader" --output json`
- In Azure PowerShell, issue the command: `Get-AzureRmRoleDefinition "Reader" | ConvertTo-Json |`
- In Azure CLI, issue the command: `az role definition --name "Reader" --output json`

This asks the test-taker to apply their knowledge to solve a problem that they may encounter on the job. The test-taker needs to recall the accounting formula, understand how it works, and then apply it to the given problem. Click for additional examples

Analyze-level questions validate whether or not the learner has the ability to diagnose situations and craft solutions using facts, concepts, techniques and rules.

- Language in these types of questions could include:
  - Given these facts, what would you do to…?
  - Given a scenario, what changes would you make to…?

- ○ Given a scenario, what is the most likely explanation for…?

Analyze-level Sample Question

Question: You are attempting to save `true` in `SharedPreferences` in an activity using the code snippet below:

```
val prefs = getSharedPreferences("pp", Context.MODE_PRIVATE)
prefs.edit().putBoolean("myBool", true)
```

However, the below code still returns `false`:

```
prefs.getBoolean("myBool", false)
```

What is a possible explanation of this behavior?

1. Neither `apply()` nor `commit()` are invoked on the editor after setting the `myBool` value, hence the default value (`false`) is returned.
2. The `putBoolean` operation is happening in the UI thread which defers the write operation until the app relaunches. False is returned before relaunch.
3. `Context.MODE_PRIVATE` restricts the read operation to authorized users only, and as a result the default value

This goes one step beyond application. Rather being given the specific problem, the test-taker must first diagnose the situation then identify the best solution. Analysis questions often require the test-taker to integrate knowledge from multiple sources. A well-crafted analysis question can tell you a lot about the test-taker. *[Click for additional examples]*

Additional Bloom's Taxonomy Examples

Remember-level

Which MongoDB server command removes a document from a collection?

- **remove**
- deleteOne
- drop
- the Upsert command

What is true about dependency injection?

- **Dependency injection is the mechanism whereby one object supplies the dependencies of another object.**
- Dependency injection is another name for the Spring framework.
- Using dependency injection, you can inject third-party libraries to an application.
- Dependency injection helps to build a highly coupled system.

Understand-level

What is the difference between writeErrors and writeConcernErrors fields?

- **writeConcernErrors only contain errors related to replicating the write to other nodes.**
- Because these are aliases, there is no difference between the writeErrors and writeConcernErrors fields.
- writeErrors are returned when you operate against a single node. writeConcernErrors is returned instead when you connect to a replica set.
- writeErrors only contains disk I/O errors, while writConcernErrors can contain networking errors.

What can using dependency injection allow you to do?

- **Provide support for different configurations at runtime.**
- Pass any number of parameters to a method.
- Know where to find dependencies.
- Define how an object can use another object.

Apply-level

(Technical Example)

Collection "demo" has a document { "_id" : 1, "name" : "wanda", "x":1}. You run an update db.demo.update({"_id":1}, {"_id": 2, "name": "wanda", "type": "fish"}); What will be the outcome?

- **The document with _id 1 will become {"_id": 1, "name": "wanda", "type": "fish"}.**
- The document with _id 1 will become {"_id": 2, "name": "wanda", "type": "fish"}.
- The document with _id 1 will become {"_id": 1, "name": "wanda", "x": 1, "type": "fish"}.
- The document with _id 1 will become {"_id": 2, "name": "wanda", "x": 1, "type": "fish"}.

(Non-Technical Example)

Your company has a large number of pdf files and several offices around the world. How would you set up MongoDB to help achieve fast access to all of these files from every office?

- **Create a replica set with nodes in each office, and use GridFS to store the files.**
- Use GridFS to store the files on a MongoDB server, then replicate the mongo data directory to the other offices as read-only.
- Store the pdf files in a field with the BSON Binary data type, and use a Replica Set cluster.
- Use GridFS, then shard the "chunks" collection only.

Analyze-level

(Technical Example)

In a replica set cluster, 2 separate clients are updating a single document. Client one adds the value 'yo1' to the array field named 'tags', using the $push operator. Client two adds the value 'yo2' to the same 'tags' field using the $push operator. Both clients issue the update at the exact same time to the Mongo server. If the 'tags' field value was an empty array to begin with, and both writes were acknowledged by all secondaries, what would be the in the array if read from a secondary node?

- **Either ["yo1","yo2"] or ["yo2","yo1"]**
- Either ["yo1"] or ["yo1","yo2"]
- ["yo1","yo2"]
- ["yo2"]

Consider the following test class:

```
public class UserTest {
private String username;
private String password;
private User user;
private ISerializer serializer;

void testValidUsername() {
// ...
}

void testValidPassword() {
// ...
}

void testUserCreation() {
// ...
}

void testUserConfigurationAfterRetrieval() {
// ...
}
}
```

Which attribute is the best candidate for dependency injection?

- **serializer**
- User
- Username
- password

(Non-Technical Example)

You enabled profiling on your main database, by issuing db.setProfilingLevel(1,40). However, your very busy application still seems slow, and the profile data contains only a few operations every minute, none of them slower than 80 milliseconds. Why might that be?

- **Since only operations slower than 40 milliseconds are captured, you may not see a huge number of queries that are faster, which cumulatively affect the performance negatively.**
- Because the second parameter - 40 - is the maximum number, not the minimum for an operation to be logged.
- Passing 1 as the first parameter captures queries only, so you may be missing write operations that are slowing the database.
- The profiler only measures timing of the operations themselves. If you have document validation enabled, then the profile doesn't capture that but it may be slowing queries down.