I just want to introduce the theory of two of the more advanced features of Aurora and those are **parallel query and Aurora Global Database.** Now these are things which I don't expect to be immediately on the exam, but it is useful knowledge to understand either for new exam questions or for real world usage and again, I'm going to stick to high level architecture only because the implementation of these things is beyond what you need at an associate level.

**So parallel query is something that you can configure when you create an Aurora cluster.** So if you recall from the previous lesson, when I'm creating an Aurora cluster and assuming I pick a compatible addition and version I get the ability to pick the parallel query option. And essentially, **the way this works is that it's a feature that once you enable it, it allows queries to be executed across all nodes in an Aurora cluster at the same time.** **So instead of a sequel query just running on a single node. For larger queries which need access to larger data sets or more compute power, you can have those queries executing across all of the nodes in your cluster at the same time, and that offers massive performance benefits. It's a great idea if you're querying large data sets that are across the entire cluster shared storage, then you're going to get some significant benefit.** **It's not something you'd use for smaller queries because they're not going to see much in the way of advantages but for larger queries, that is something that would give you massive performance benefit.** So that's all I'm going to cover. I just want you to be aware of the fact that it exists, and it's something you have to enable when you create an Aurora cluster. I'll make sure I include a link in the description with additional information if you are interested in reading more about it. But just for the exam, I want you to be aware that it exists.

Now the other thing I wanted to cover in this lesson is the concept of **the Aurora Global Database, and again, this is something that needs to be configured when you make the database cluster, and it's only supported with certain editions and certain versions.** So in this particular case, I've got mySQL compatibility selected and I've got 5.6.10a as the version. And if I've got that selected, I can take this box for global databases. Now, **when you create a global database, it consists of one primary AWS region, which is the master for your data and one read only secondary region. So when you select it, Aurora replicates data to the secondary AWS region typically in under a second. So all write operations have to occur on the primary database instance in the primary region. So Aurora benefits from dedicated infrastructure that AWS has got in place, which essentially allows this rapid replication to occur with really low latency on a global basis. So you generally utilize an Aurora Global Database when you want to add that global level of resilience or when you want to provide a second region which could be used for read workloads.**

So why don't I quickly look at stepping through how to create an Aurora Global Database. Well, I'm not going to actually use it for anything, so I don't need to be focused on what values I put. But I'm going to go ahead and create a global database. I'll use this default database identifier. I'm going to automatically generate a password and leave encryption set to enabled. When you're using a global database, you can't use burstable instances, so you can't use T2 or T3 so it is much more expensive to run a global database. I wouldn't recommend doing this in your own environment because these are fairly large, expensive sizes of instances. But let's say that I wanted to do it. I'm going to pick db.r5.large. So I'll select that as a memory optimized instance class. I want to create an Aurora replica reader node in a different AZ. So I want to make sure this is a highly durable solution. I'm going to scroll down further still, expand additional connectivity configuration, I don't want to make it publicly accessible. I'm going to accept the rest of the defaults, go to additional configuration, accept all of the defaults for the naming and they just go ahead and hit "Create Database." Okay if I minimize the regional cluster just to make it a little bit easier to see, we got this global database object so the role is global. Inside there we've got a primary database cluster, which has got one writer instance and one reader instance. Now this part is what a traditional Aurora cluster would look like if we want to add an additional region to configure this global level of resilience then we'd select the global object, select action, add region, and then we'd pick the specific region to use. So let me drop down this box and let's say that I've selected Asia Pacific Sydney. Now it'll run some checks to make sure that region is suitable for this deployment. Once it's done that I'll select the same instance size as I've used in the primary cluster. I'll pick to not make an Aurora replica. I'll get to pick the VPC to use. I could drop down the additional configuration. I'll get to select a subnet group. I can specify whether I want it to be publicly accessible. I'll leave all these as the defaults because I just want to illustrate the process and the architecture. I can expand the additional configuration. If I want to customize it I can specify a database instance name and a cluster identifier but I won't want to set any of these. I'll just select "Add Region" and once I've selected "Add Region." The process of creating the secondary cluster will begin. It will create the cluster inside the Asia Pacific Southeast 2 region, which is obviously the Sydney region. Because I've elected not to have any additional replicas, it's going to be one instance only. It will begin creating this reader role instance. So remember, **because this secondary cluster only has the ability to read it won't have this writer role inside it. The only cluster which will have the writer role is the primary cluster, so the primary cluster is in US East 1. The secondary cluster is in AP Southeast 2 and so the only instance types or replicas that I'll be able to create in this second cluster are going to be reader roles so that process will begin**. It will create and provisions that instance, it'll begin the replication from the primary cluster, and at that point I'll have the ability to perform these globally resilient reads from this secondary region. So that's all I wanted to cover on global databases. I just want to introduce you to the architecture and make you aware that **there are some limitations specifically on instance sizes, and you have to create the secondary cluster in a different region than the primary. So it is a fairly expensive implementation to perform.** So I don't recommend that you do this in your own environment. You need to use fairly large instances. So it is going to cost significant amounts of money to provision this. It's not something you'd use for testing. It's not something you generally use for developmental staging. You'd only utilize the global database functionality of Aurora if you had a specific reason to do so. But with that being said, that is everything I wanted to cover in this lesson. I just wanted to introduce both Aurora Global Databases and parallel queries just so you understand the concepts for the exam. So at this point, go ahead and mark this lesson as complete and when you're ready, you can join me in the next lesson where I'm going to be talking about Aurora Serverless.