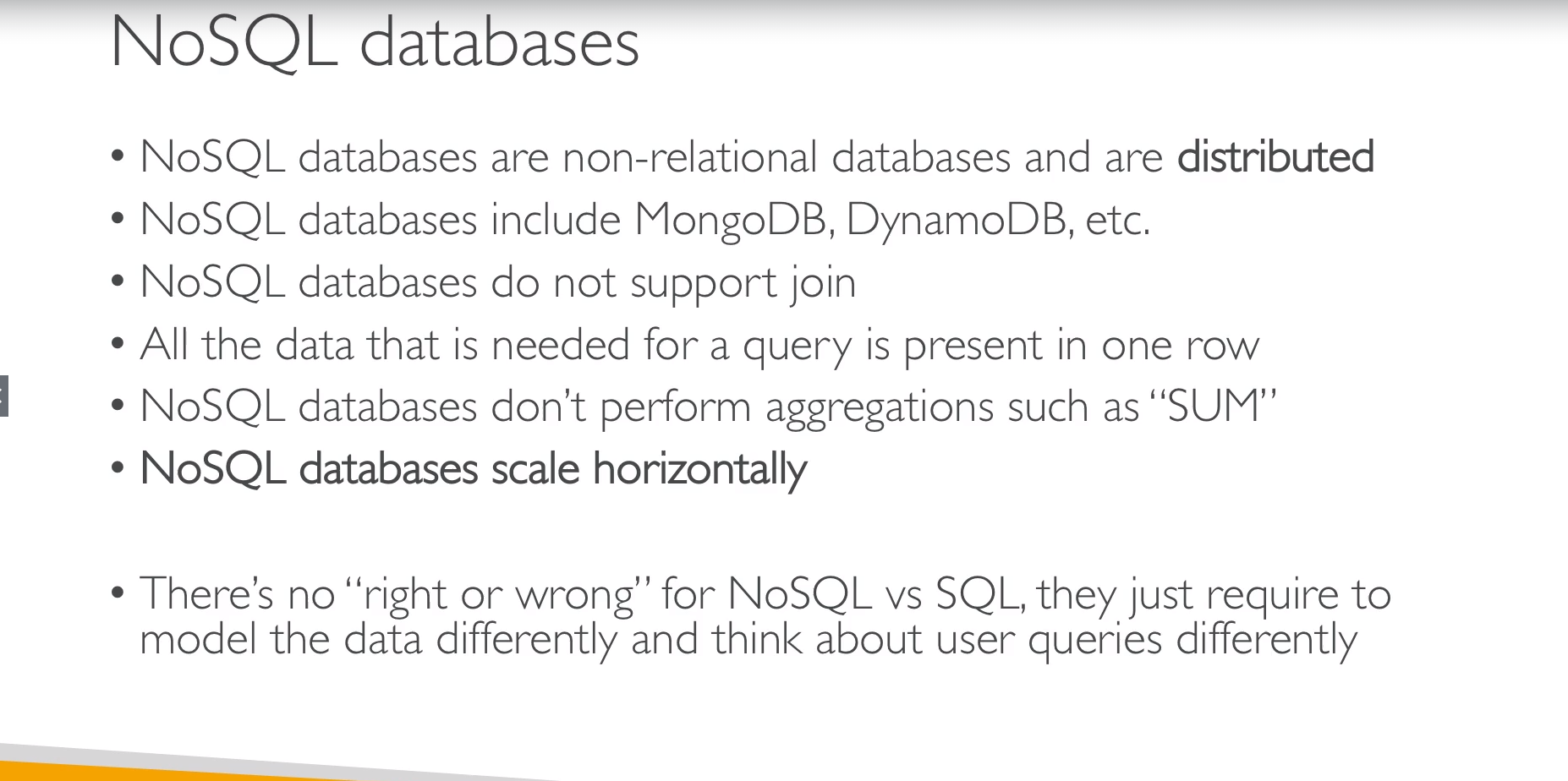
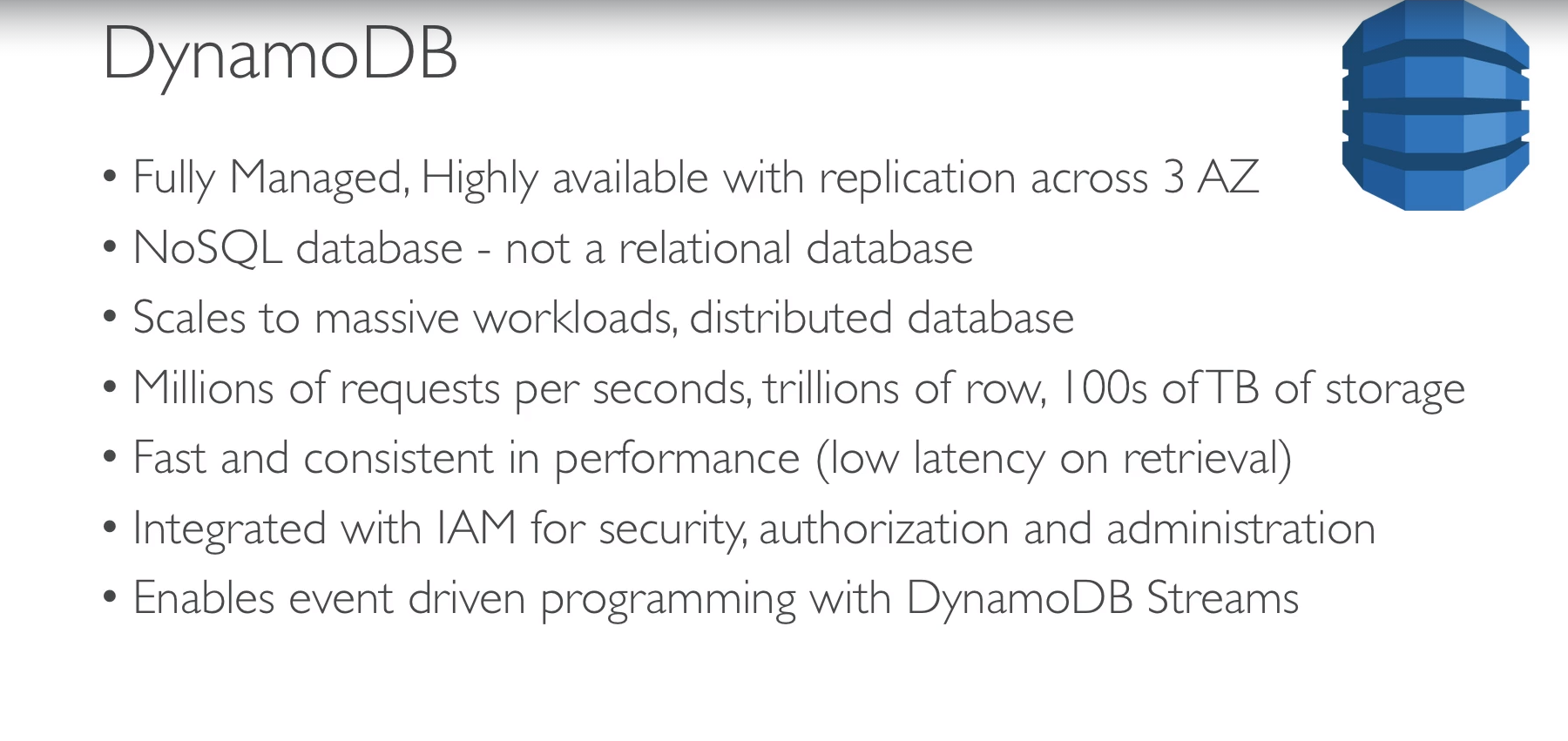
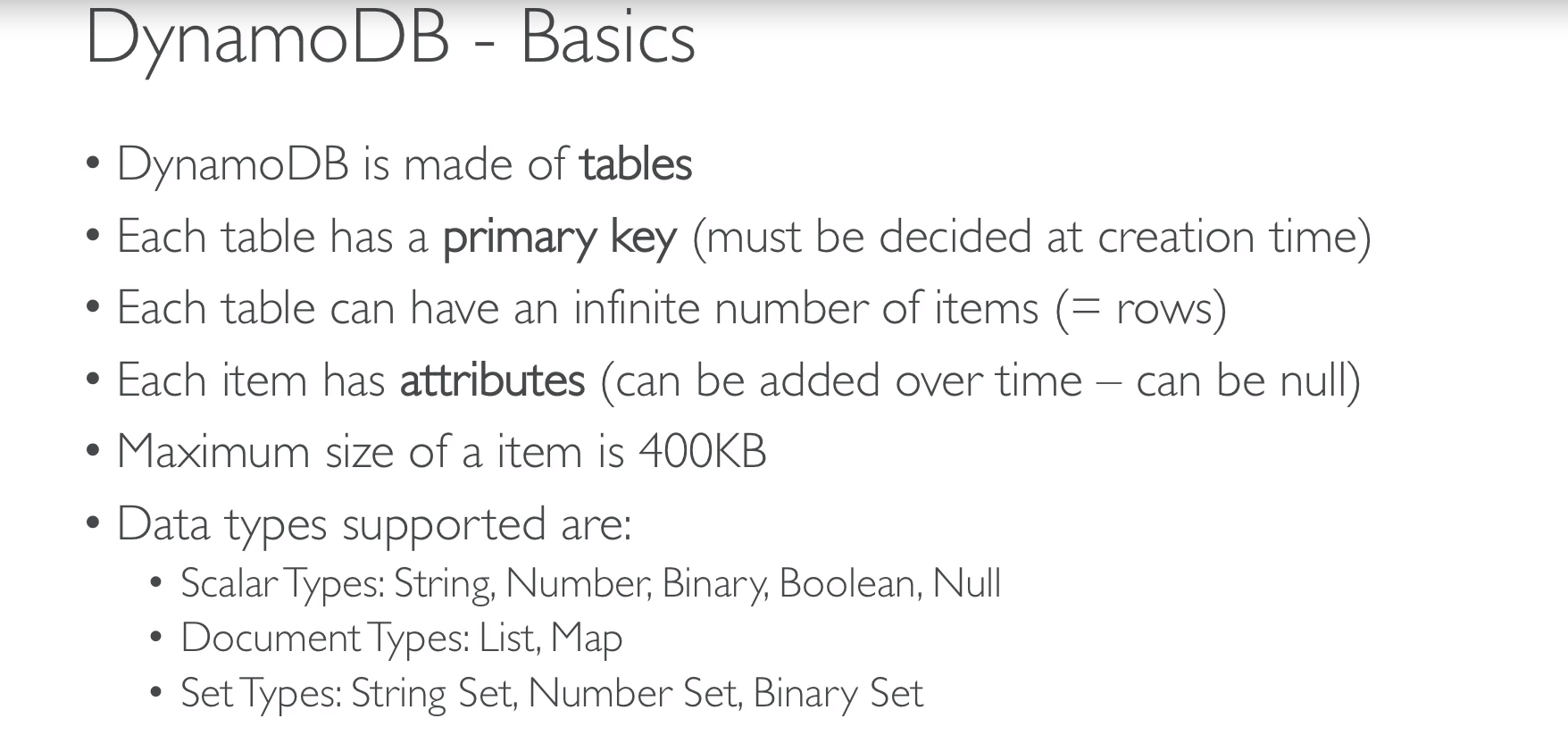
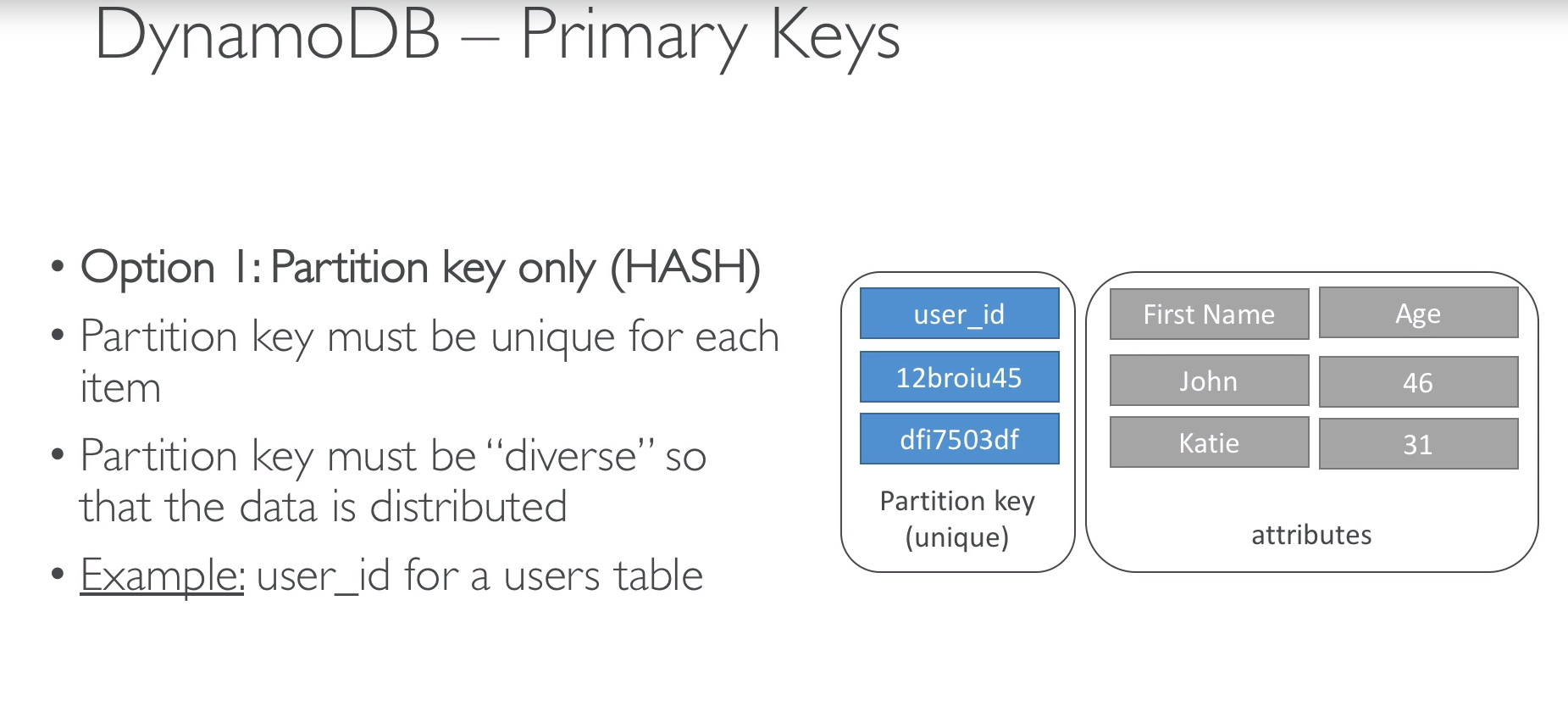


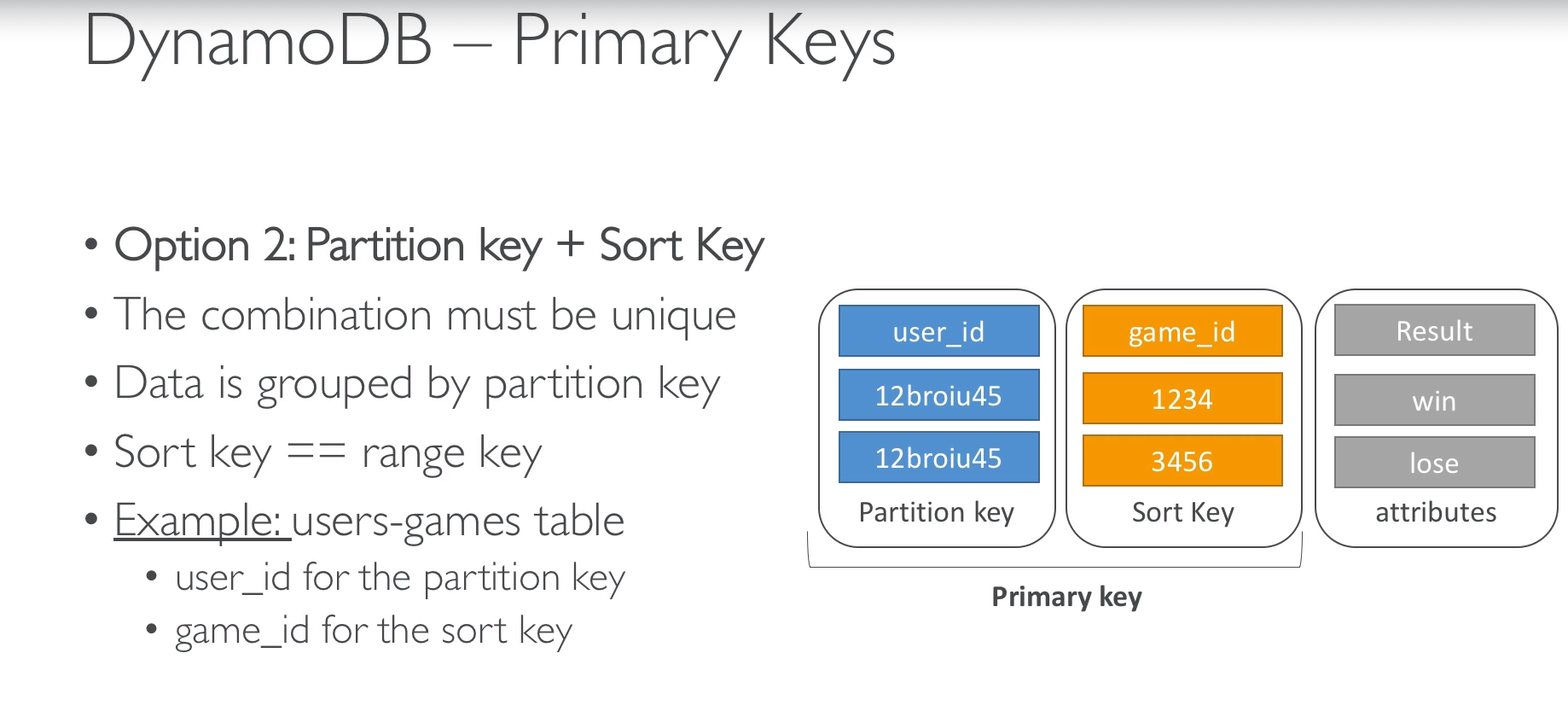
**NOSql: [Not only sql]**

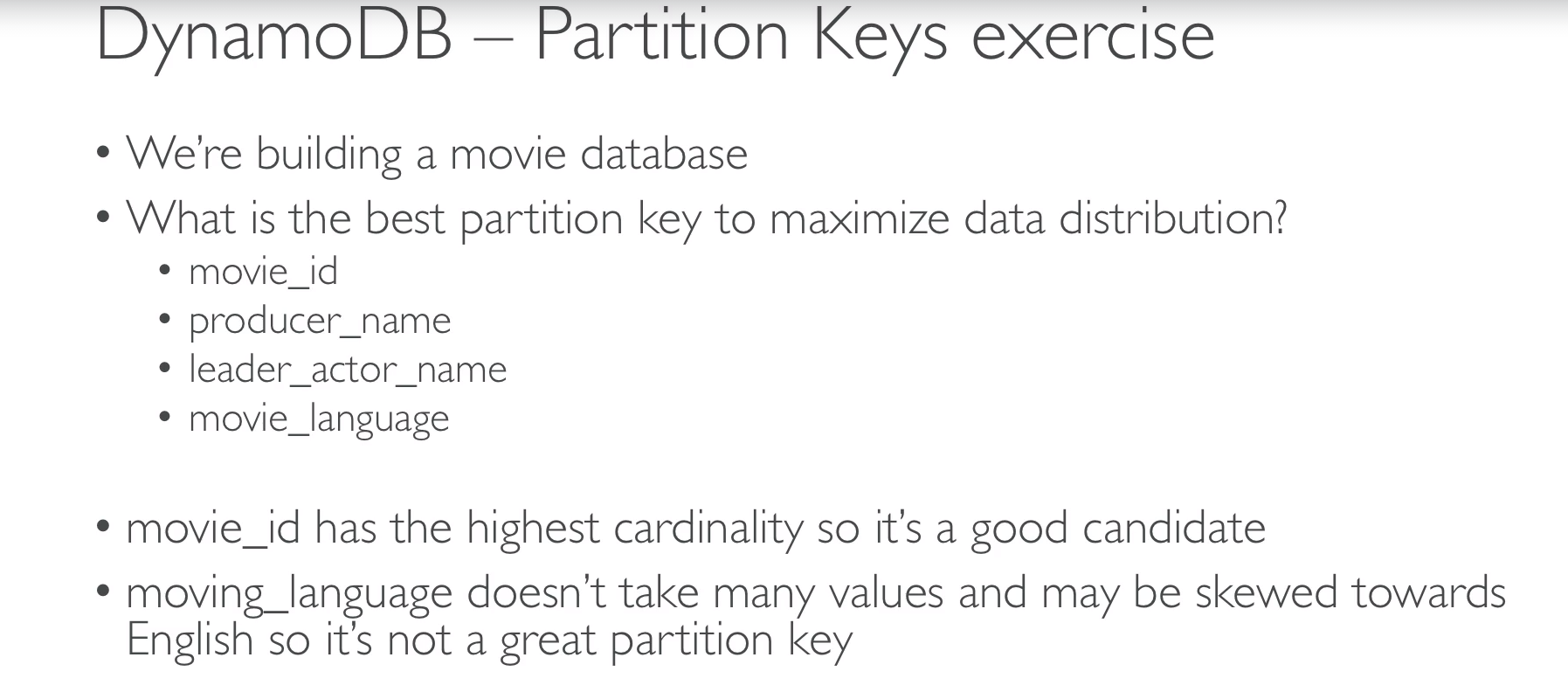




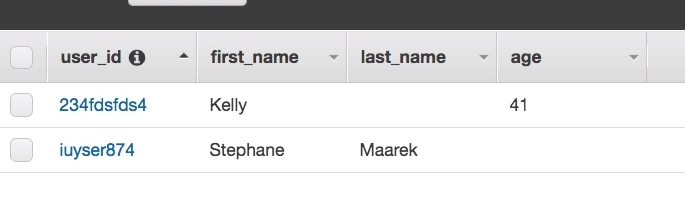




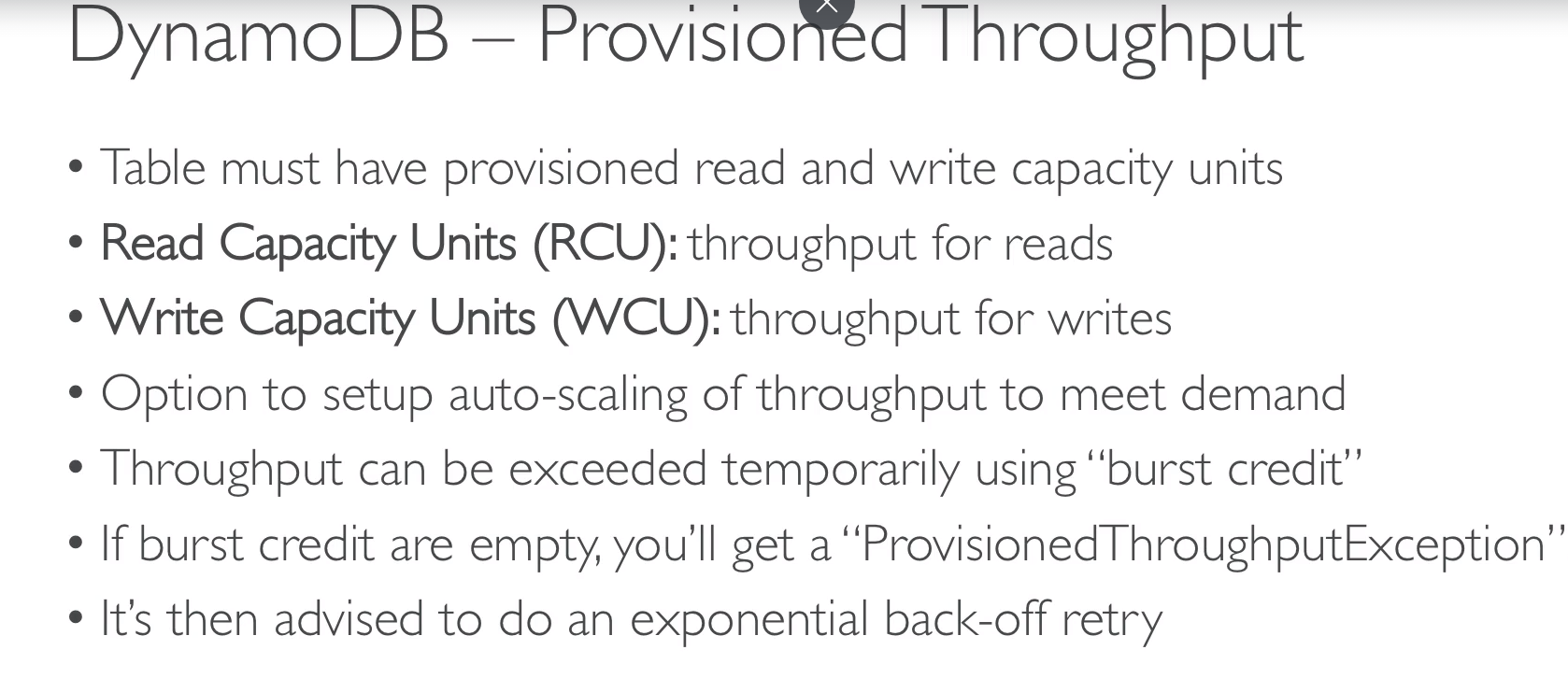




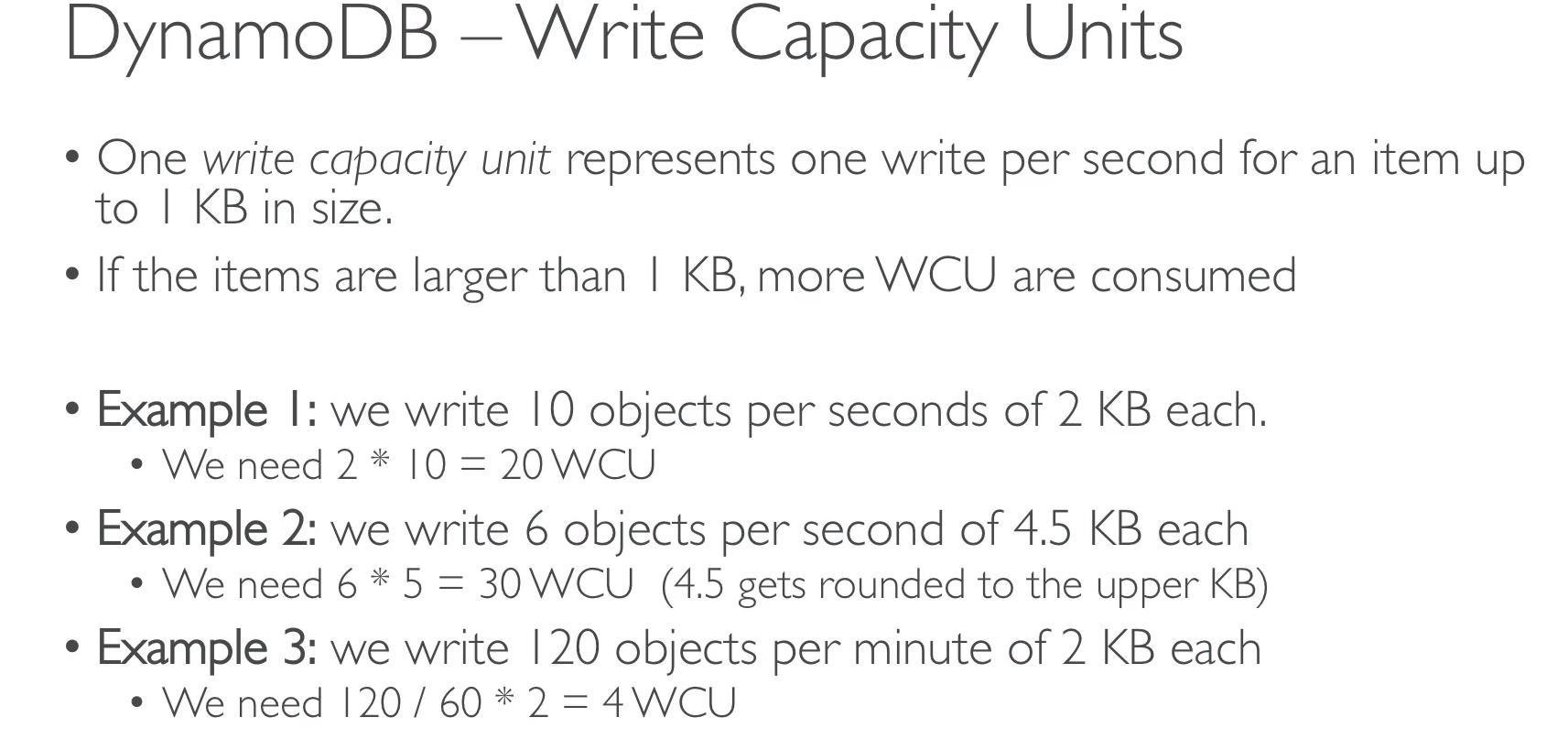
* We can’t enforce any field to be not null except partition key and sort key.



* Here autoscaling is not fully automatic means we define read and write capacity to provision throughput.



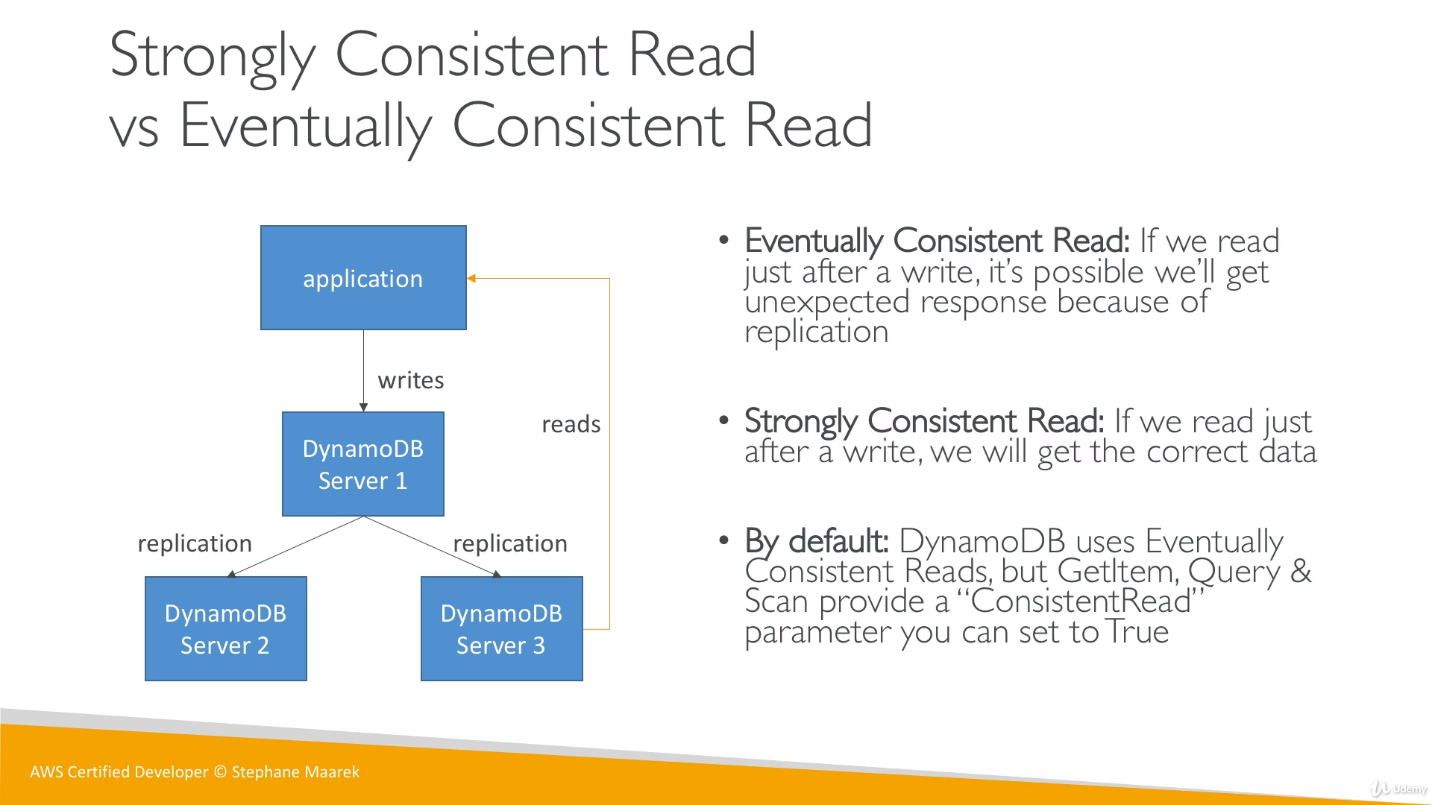
* **WCU calculation:**
* **1 write capacity = 1 write/sec [1 KB]**



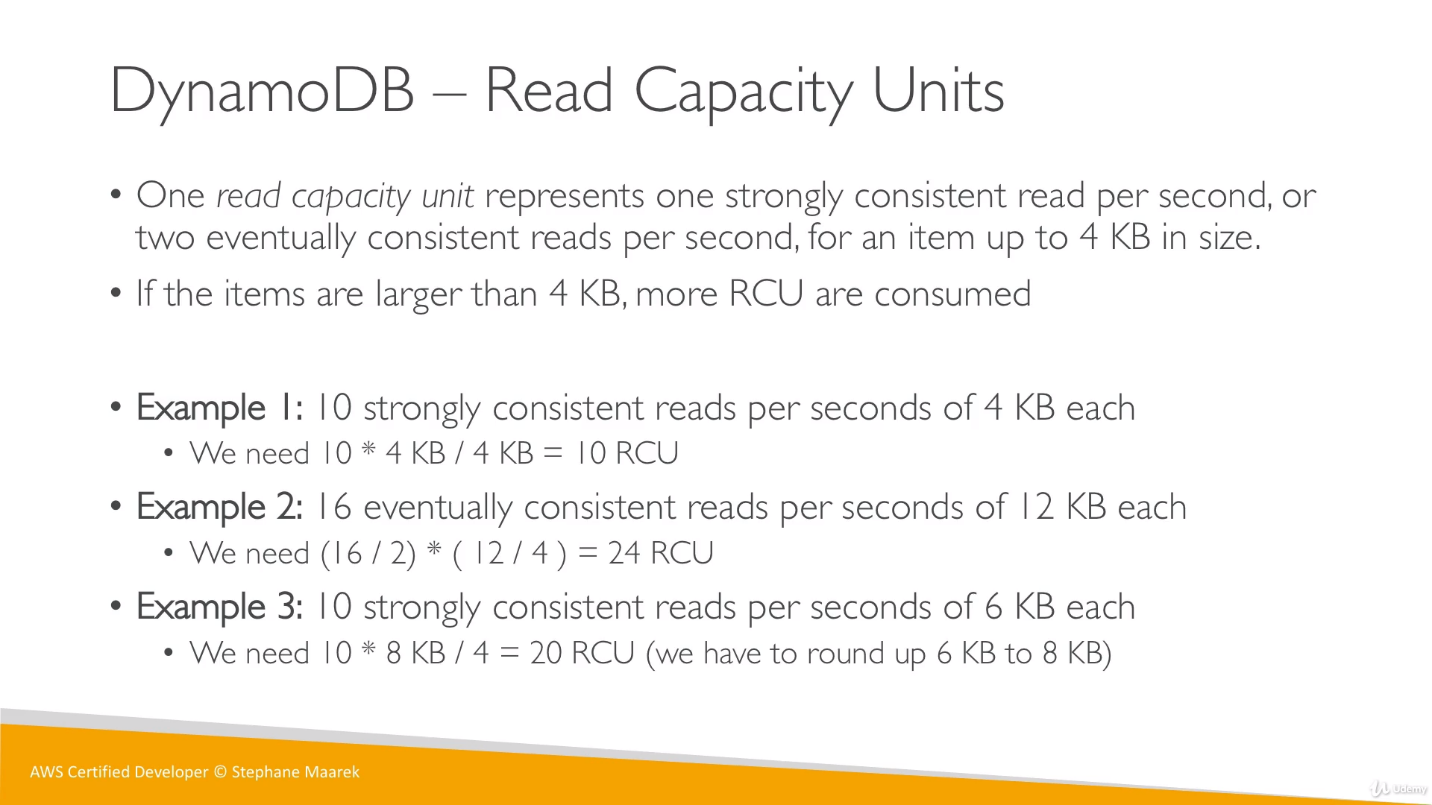
* **RCU calculation:**
* **1 Read capacity = 1 strongly consistent read/sec or 2 eventually consistent read/sec [ 4KB]**

In below example Dynamo DB is distributed across 3 AGs.

Whenever we are reading data, we actually don’t know where the request is going. It is all illustrative so per below example there might be scenario where we write data in server1 and after request, it is reading from server3 before replication happens to server3. This kind of request is known as Eventually Consistent.



If we enabled strongly consistence read, it will take bit more time but we will be able to get data for sure.



Here if size is more than 4 and less than 8 then we will consider it as 8KB and rest also in similar fashion ie in multiplies of 4.

Ex 1: 6KB, 4 read/sec, eventually

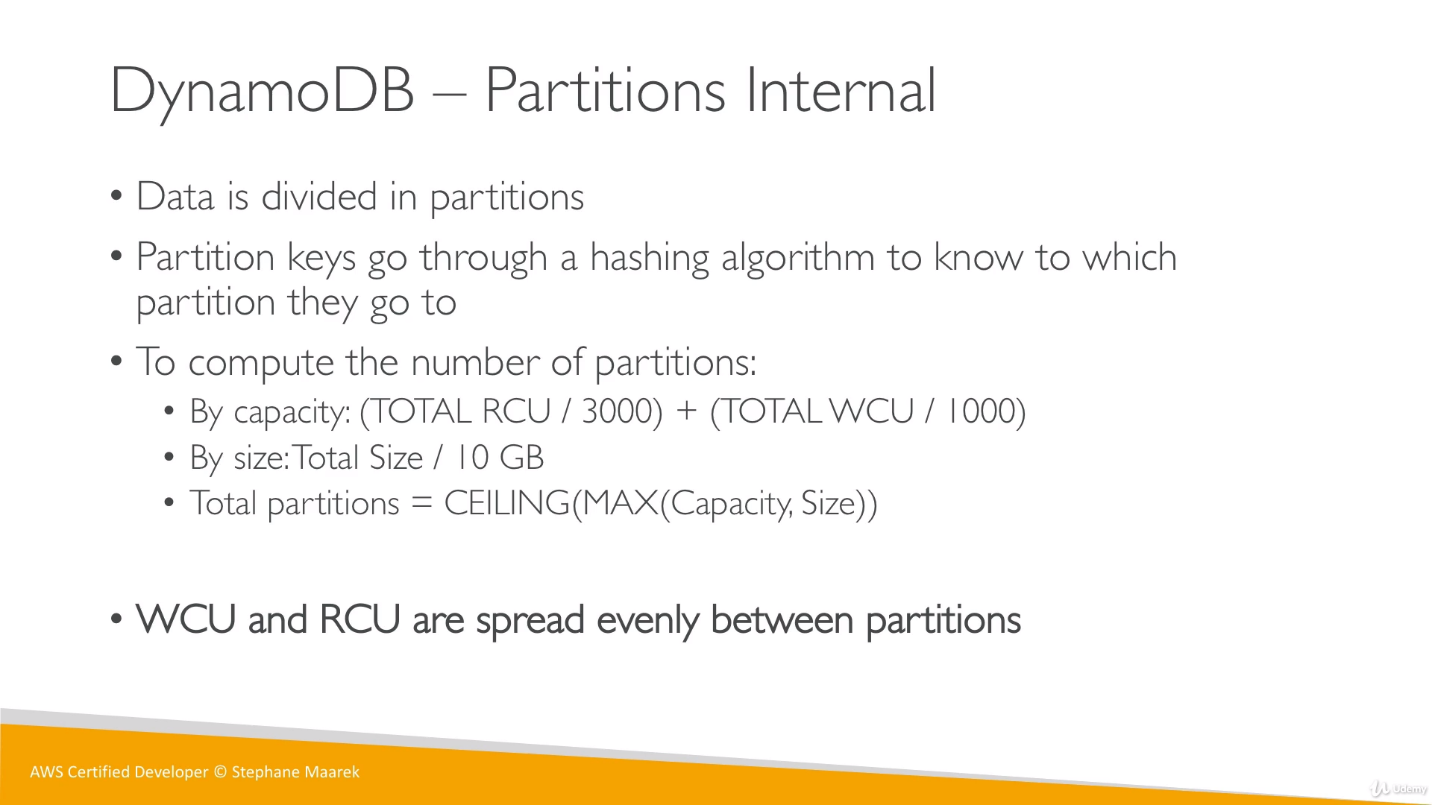
eventually: 6KB => 8 KB

(4/2) \* (8/4) => 2 \* 2 => 4

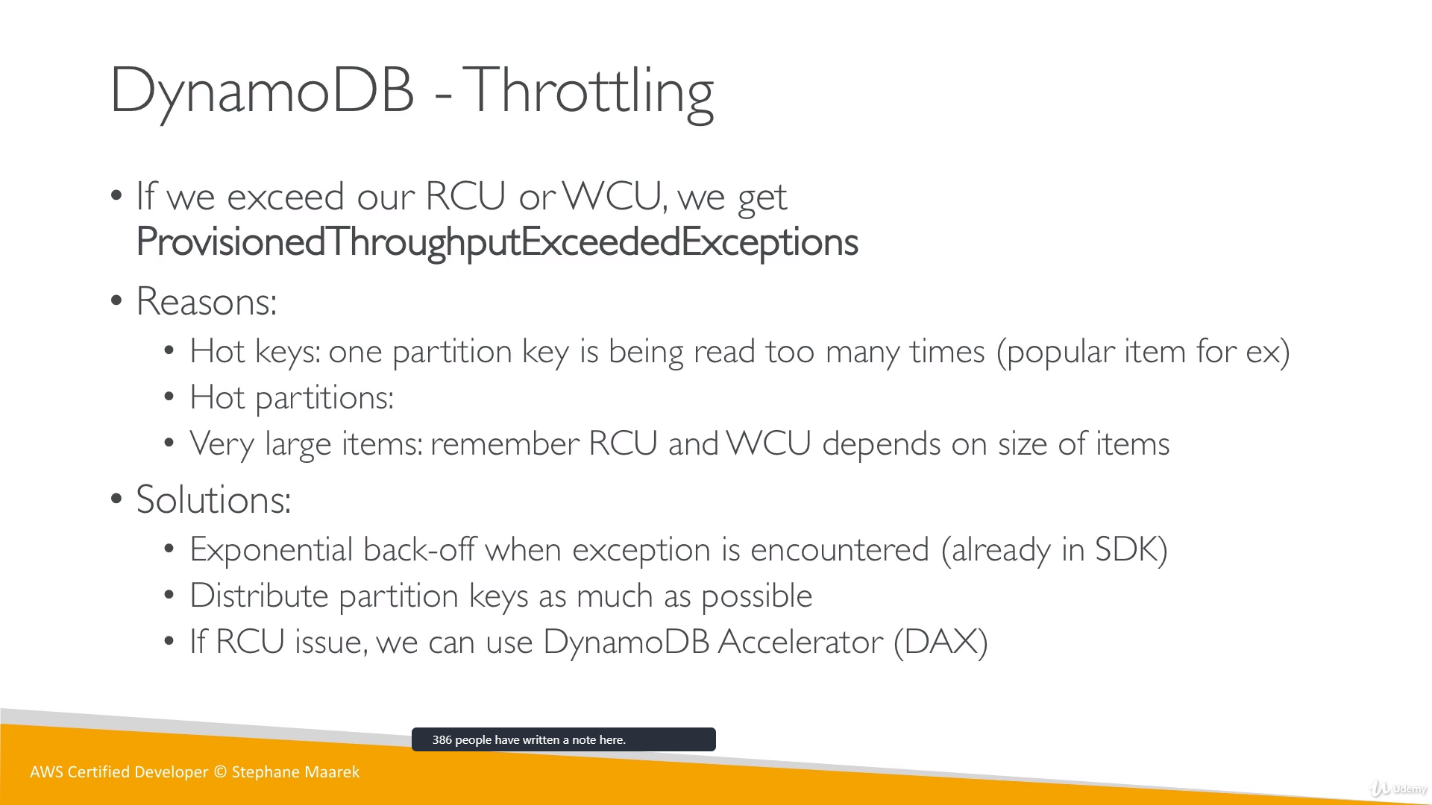
Strongly:

(4/1) \* (8/4) => 4 \* 2 => 8

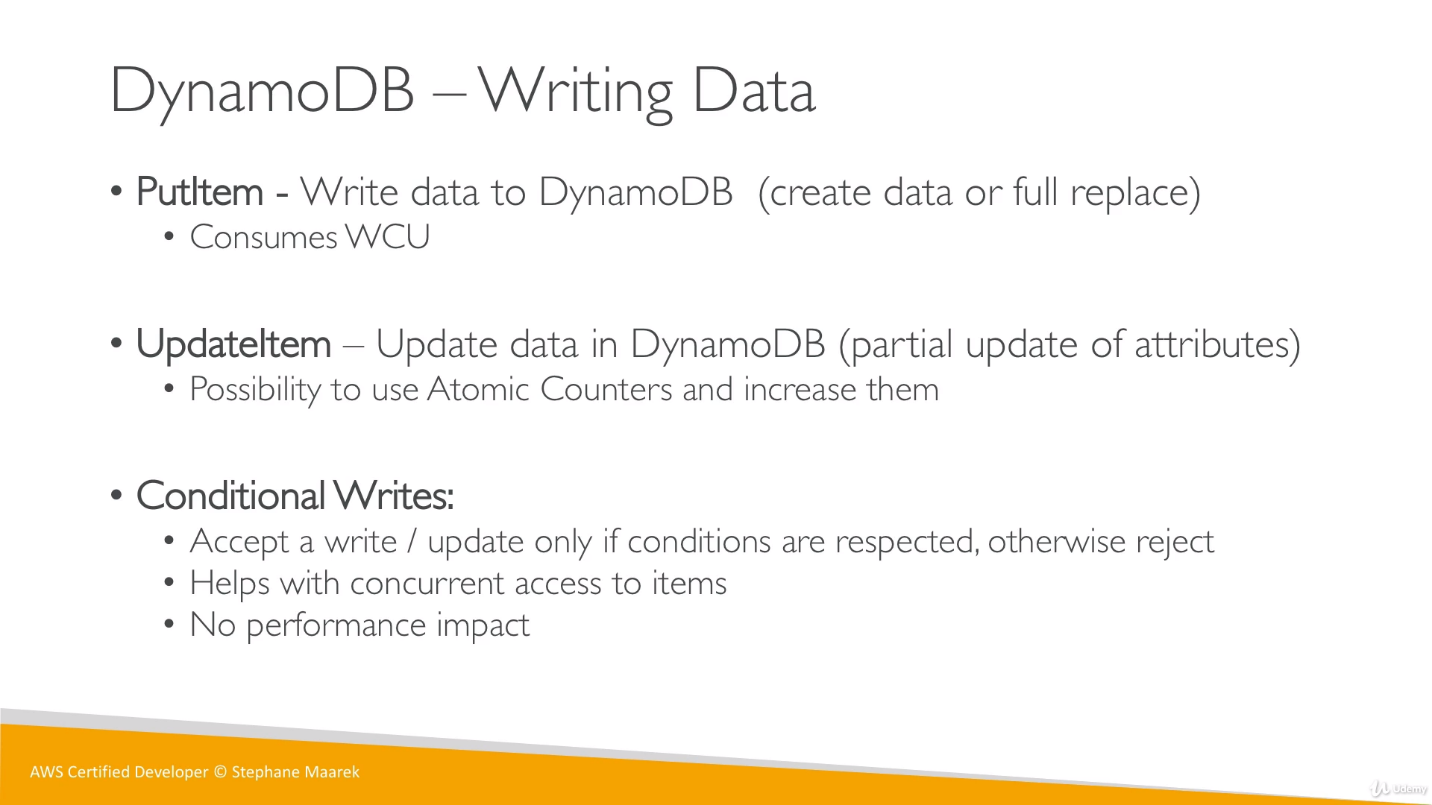
No need to remember below formula.

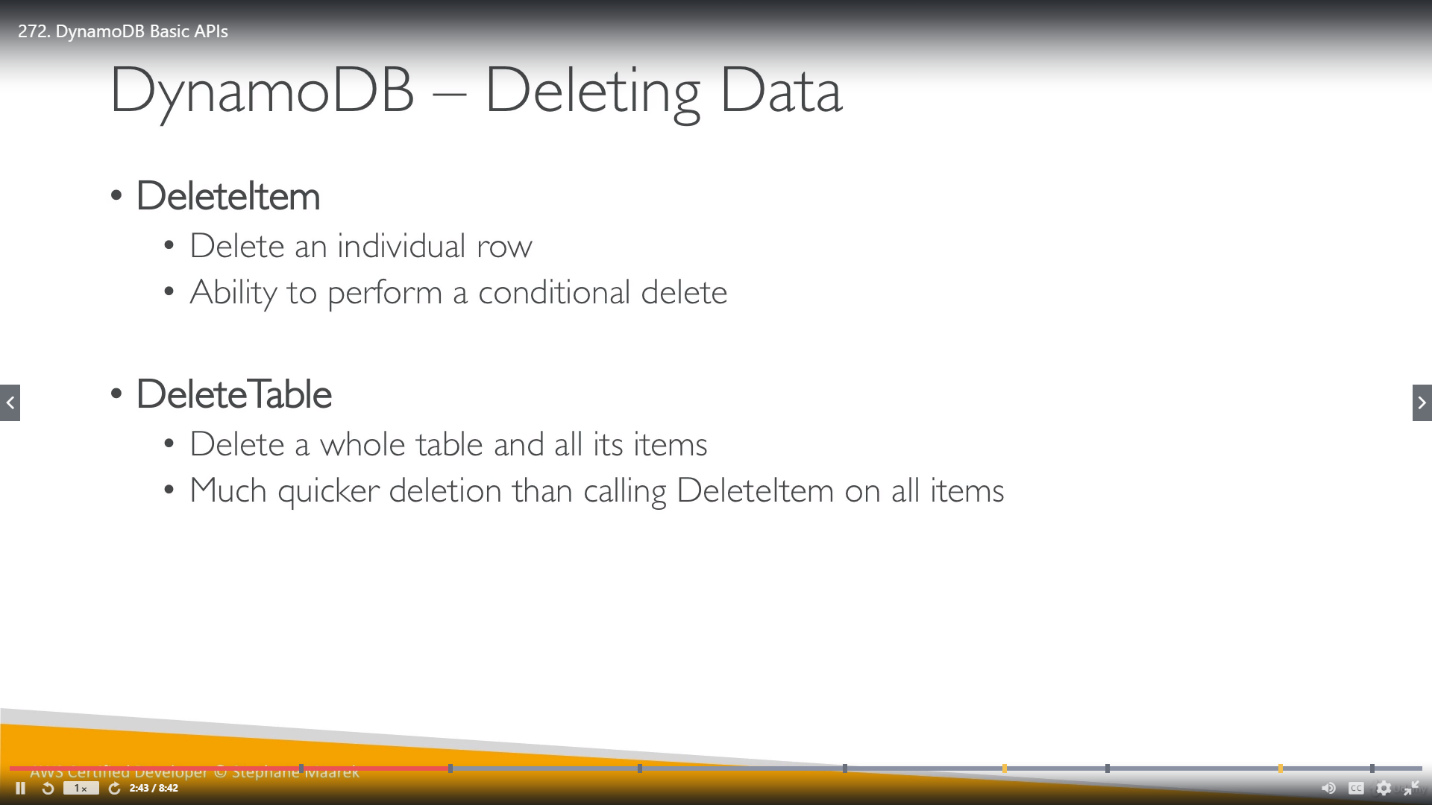


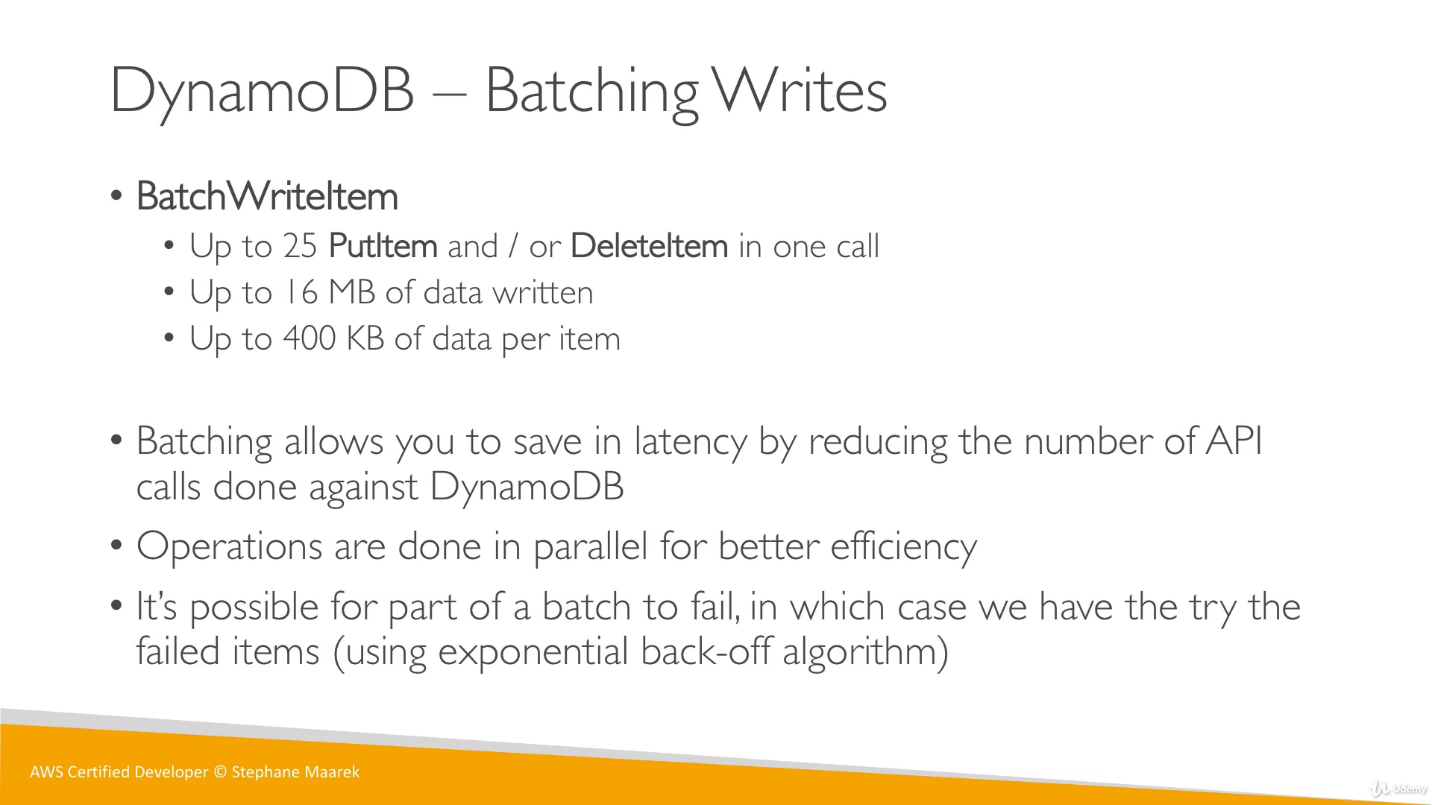
If we have 10 partitions and 100 WCU and 100 RCU, then each partition gets 10 as WCU and RCU are spread evenly between partitions.

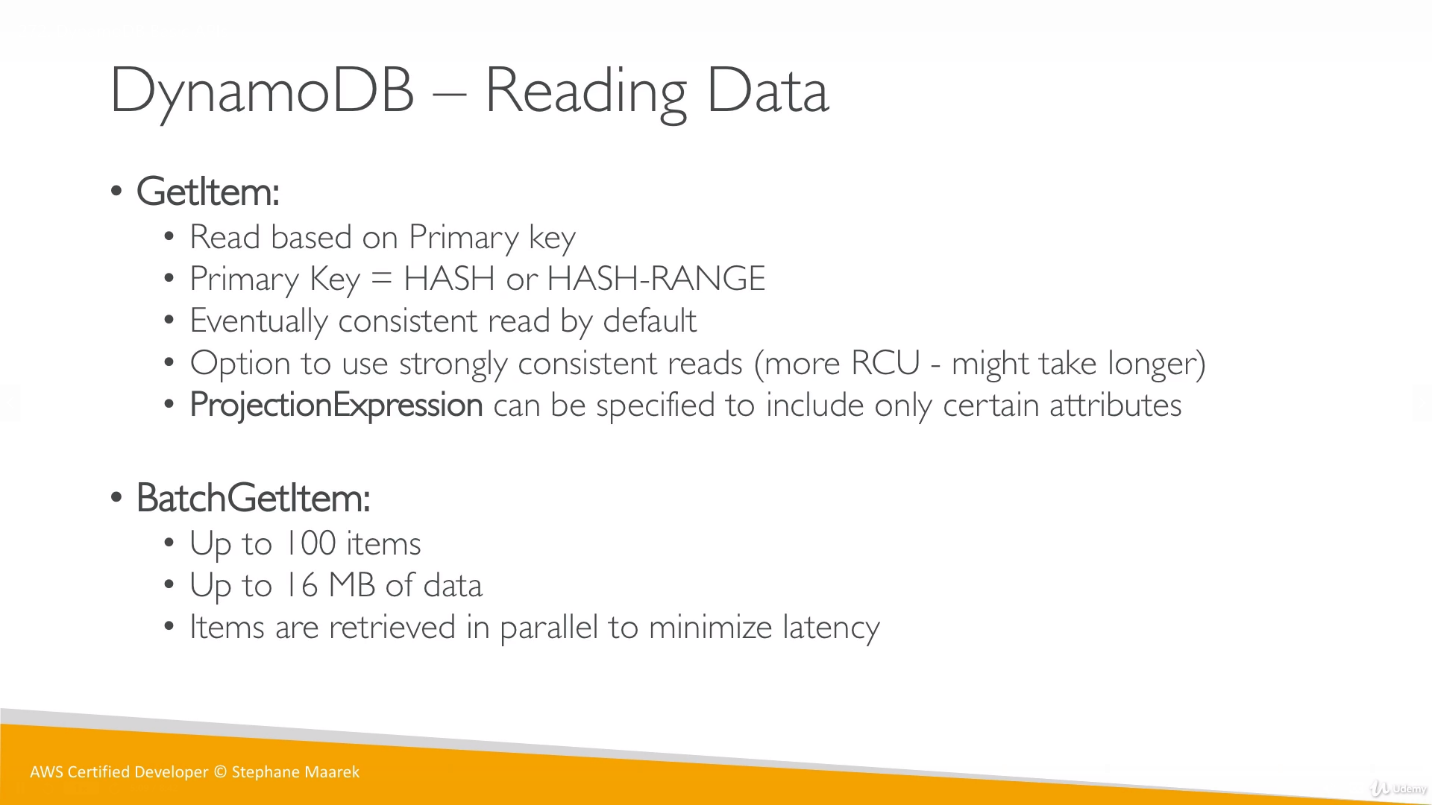


* If we enable Auto Scaling, AWS will manage WCU and RCU. Mean we wont able to choose capacity/unit for Read and Write.
* **DynamoDB APIs**:









**ProjectionExpression** is Kind a filter which is allowed only on some specific attributes.

Filtering happens at client side, DynamoDB will give whole data, client is responsible to filter the data according to filter condition and return to us.

