   
 DYNAMO DB - AMAZONS NOSQL DATABASE  
  
1. What is DynamoDB : It is a FAST and FLEXIBLE nosql database. It is for applications that need single digit milli second latency.

2. Supports : It supports both document ( json/html/xml) as well as key-value data models.

3. Reliable performance : dynamodb databases are always backed by **ssd’s**.

4. Serverless , fully managed and configured to autoscale.

5. Availability : The underlying hardware supporting the database instance is always spread across **3** geographically distinct data centers. Thus avoiding any single point failure/unavailability.

6. CONSISTENCY MODELS FOR DYNAMO DB READ OPERATIONS.  
Eventually consistent reads : In this model Performance is given importance over consistency. Consistency across all the copies of the data across the 3 locations is usually reached with in a second (if a read is performed within 100 or 200 milliseconds after you write, there is a chance that you could get old or new data).   
Note : Whenever a data write occurs DynamoDB stores this data 3 times i.e in 3 locations. As soon as it is written in 2 centers as 200 OK status code is returned to the client.  
i.e you will receive a commit OK message after only two of those replicas have been updated. And there is a small chance that a read immediately after a write, will retrieve out of data information from an un-updated replica, unless we want to specify we want a strongly consistent read.

Strongly consistent reads : A strongly consistent read will read from multiple locations at the same time to ensure it receives the most up to date copy of the data. A Strongly consistent read will always return a result that reflects all the writes that received a successful response prior to the read.

7. Tables.  
Items : similar to a row – key value pairs.  
Attributes : columns..  
key value and document data structures are supported.

8. Primary keys : How do we query data in the table.  
DynamoDB stores and retrieves data based on the primary key.  
The primary key is used when we query the data.

There are 2 types of primary key.  
Partition Key : It is a unique attribute [The value of a key which has to be unique]  
The value of the partition key is input into a internal hash function, The output of the hash function actually determines the partition or the **physical location** where that data is going to be stored.  
If you are using the partition key as the primary key then no two items with in your table can have the same partition key. As we cannot store two items in the same physical location.

Composite key : This is the second type of primary key. This is made up of a combination of the partition key and a sort key.  
composite key = partition key + sort key.  
If the partition key is not necessarily unique then you will use this key.  
Example : Consider a forum posts. We might be storing attributes like the UserID, The message typed in by the user, The timestamp of the post, The name of the Forum. In this example the Primary key could be a composite key which is made up of userID (partition key) and timestamp of the post(sort key).  
In this case two items could have the same partition key, because we could have the same user posting multiple times in the same forum or even different forums. In order to identify the record uniquely we need to use a combination of the ‘userID’ and the ‘time’ that posted the post in a forum.  
Now all items with the same partition key are stored together and they are sorted with a value of the sort key.

9. Access Control to DynamoDB.

- It is all managed by IAM

10. Dynamo DB lab.

Install SDK for PHP : get composer. And use composer to install the latest version of sdk.