[ENGLISH]

Hello Cloud Gurus, and welcome to this lecture.

And this lecture is going to be all about DynamoDB indexes.

So let's get started. So what is an index? Well,

a secondary index allows you to perform more flexible querying

within DynamoDB, and it allows you to query based on an attribute,

which is not the primary key.

So DynamoDB allows you to run a query on non-primary key

attributes using global secondary indexes and local

secondary indexes.

And a secondary index allows you to perform fast queries on specific

columns in a table.

And you select the columns that you want included in the index,

and then run your searches on the index,

rather than on the entire dataset.

And because you're only running the search on the required columns,

it means your queries are a lot faster and more efficient than running the query

on the entire dataset. So let's take a look at these in little bit more detail.

Now,

a local secondary index is an index that can only be created when you are

creating your table.

And it has the same partition key as your original table,

but it does have a different sort key.

And that's what gives you the different view on your data,

because your data is going to be organized according to an alternative sort

key. Any queries that are based on this sort key that you run against the local

secondary index are then much faster than they would be if you were using the

main table. So, for example,

in a customer table where you're storing all information about your customers,

their email address, all of their purchases they've made,

all their contact details, et cetera, you might have a local secondary index,

which uses the main partition key of user ID,

and then uses a different sort key to your main table.

So you might use a sort key like account creation date.

And then the limitation with local secondary indexes is that the local

secondary index must be added at creation time.

So it can only be created when you're creating the table and you cannot add,

remove, or modify it later on.

Now for global secondary indexes, these are much, much, more flexible.

So you can create the global secondary index whenever you like.

So you can either create it when you create your table,

or you can add it later on,

and it allows you to choose a different partition key,

as well as a different sort key to your main table.

So that gives you a completely different view of the data. And of course,

it speeds up any queries that are relating to this alternative partition key and

sort key. So for example, in our customer table,

we might want to create a global secondary index using a different partition key

of email address, and then use a sort key of last login date.

And you can create a global secondary index whenever you like.

So you can create it when you create your table itself,

or you can add it later on.

So what are our exam tips for the DynamoDB indexes? Well firstly,

indexes overall enable much faster queries,

but only on very specific data columns.

They give you a different view of your data based on an alternative partition

key or an alternative partition key and an alternative sort key.

And it's really important to understand the differences between these 2 types of

index. So just remember with a local secondary index,

it uses the same partition key and a different sort key to your

original table,

and it must be created when you first create your table.

And then with a global secondary index,

it's going to use a different partition key and a different sort key to

your original table. And with a global secondary index,

you can create this at any time. So it's got that added flexibility.

So that is the end of this lecture. If you have any questions,

please let me know. Otherwise, feel free to move on to the next lecture.

Thank you.

[SPANISH]

Hola Cloud Gurus, y bienvenidos a esta conferencia.

Y esta conferencia tratará sobre los índices de DynamoDB .

Entonces empecemos. Entonces, ¿qué es un índice? Bien,

un índice secundario le permite realizar consultas más flexibles

dentro DynamoDB , y le permite consultar en función de un atributo,

que no es la clave principal.

Entonces, DynamoDB le permite ejecutar una consulta en una clave no principal

atributos usando índices secundarios globales y locales

secundarios .

Y un índice secundario le permite realizar consultas rápidas sobre determinados

columnas en una tabla.

Y seleccionas las columnas que quieres incluir en el índice,

y luego ejecute sus búsquedas en el índice,

en lugar de en todo el conjunto de datos.

Y debido a que solo ejecuta la búsqueda en las columnas requeridas,

significa que sus consultas son mucho más rápidas y eficientes que ejecutar la consulta

en todo el conjunto de datos. Así que echemos un vistazo a estos con un poco más de detalle.

Ahora,

un índice secundario local es un índice que solo se puede crear cuando está

creando tu mesa.

Y tiene la misma clave de partición que su tabla original,

pero tiene una clave de clasificación diferente.

Y eso es lo que le da una visión diferente de sus datos,

porque sus datos se organizarán de acuerdo con un tipo alternativo

llave \_ Cualquier consulta que se base en esta clave de ordenación que ejecute en el local

secundario son mucho más rápidos de lo que serían si estuviera usando el

mesa principal Así por ejemplo,

en una tabla de clientes donde está almacenando toda la información sobre sus clientes,

su dirección de correo electrónico, todas sus compras que han hecho,

todos sus datos de contacto, etcétera, es posible que tenga un índice secundario local,

que utiliza la clave de partición principal de ID de usuario,

y luego usa una clave de clasificación diferente a su tabla principal.

Por lo tanto, puede usar una clave de clasificación como la fecha de creación de la cuenta.

Y luego la limitación con los índices secundarios locales es que el local

secundario debe agregarse en el momento de la creación.

Por lo tanto, solo se puede crear cuando está creando la tabla y no puede agregar,

elimínelo o modifíquelo más adelante.

Ahora, para los índices secundarios globales, estos son mucho, mucho más flexibles.

Entonces puede crear el índice secundario global cuando lo desee.

Entonces puede crearlo cuando crea su tabla,

o puedes agregarlo más tarde,

y le permite elegir una clave de partición diferente,

así como una clave de clasificación diferente a su tabla principal.

Eso le da una visión completamente diferente de los datos. Y por supuesto,

acelera cualquier consulta relacionada con esta clave de partición alternativa y

de clasificación . Entonces, por ejemplo, en nuestra tabla de clientes,

podríamos querer crear un índice secundario global usando una clave de partición diferente

de la dirección de correo electrónico y, a continuación, utilice una clave de clasificación de la última fecha de inicio de sesión.

Y puede crear un índice secundario global cuando lo desee.

Entonces puede crearlo cuando crea su propia tabla,

o puede agregarlo más adelante.

Entonces, ¿cuáles son nuestros consejos de examen para los índices de DynamoDB ? Bueno, en primer lugar,

los índices en general permiten consultas mucho más rápidas,

pero solo en columnas de datos muy específicas.

Le brindan una vista diferente de sus datos basada en una partición alternativa

clave de partición alternativa y una clave de ordenación alternativa.

Y es muy importante entender las diferencias entre estos 2 tipos de

índice \_ Así que recuerda con un índice secundario local,

utiliza la misma clave de partición y una clave de clasificación diferente a su

mesa original ,

y debe crearse cuando crea su tabla por primera vez.

Y luego con un índice secundario global,

va a utilizar una clave de partición diferente y una clave de clasificación diferente para

tu mesa original. Y con un índice secundario global,

puede crear esto en cualquier momento. Así que tiene esa flexibilidad adicional.

Así que ese es el final de esta conferencia. Si tienes alguna pregunta,

por favor hágamelo saber De lo contrario, siéntase libre de pasar a la siguiente lección.

Gracias.