

U8-Tema8.4-Servicios de bases de datos en AWS. Amazon Dynamo DB

Fundamentos de Computación en la Nube (1ºDAM/1ºDAW)

Álvaro García Sánchez



Introducción a Amazon Dynamo DB	2
Características	2
Componentes (tablas, elementos y atributos)	3
Particiones y consultas	5
Tipos de datos	5
Consultas	6
Costes	6
Referencias	6

Introducción a Amazon Dynamo DB

DynamoDB (<https://aws.amazon.com/es/dynamodb/>) es un **servicio administrado** de base de datos **NoSQL clave-valor y documental**, ágil y flexible para todas las aplicaciones que necesiten una latencia constante de milisegundos de un solo dígito a cualquier escala y una capacidad de almacenamiento prácticamente ilimitado.

Ofrece un modelo clave/valor (similar a **Redis** y **MongoDB** a la vez), flexible y sin estructura fija (los elementos pueden tener atributos diferentes), diseñado para garantizar un determinado rendimiento y disponibilidad para cada tabla, es decir, se definen elementos por tabla y se paga según lo exigido en cada una.

Características

- Amazon administra toda la infraestructura de datos subyacente para este servicio y almacena los datos de forma redundante en varias instalaciones de una región como parte de la arquitectura tolerante a fallos.
- Permite crear **tablas** y **elementos**. Es posible agregar elementos a una tabla.
 - El sistema particiona automáticamente los datos y proporciona almacenamiento en tablas para satisfacer los requisitos de la carga de trabajo.
 - No existe prácticamente ningún límite con respecto al número de elementos que puede almacenarse en una tabla. Por ejemplo, algunos clientes tienen tablas de producción con miles de millones de elementos.
- Los elementos en la misma tabla pueden tener diferentes atributos. Ofrece flexibilidad para añadir atributos. Los elementos de nuevo formato se pueden almacenar en paralelo con los elementos anteriores de la misma tabla, sin necesidad de realizar migraciones de esquemas.
- Todos los datos de DynamoDB se almacenan en unidades de estado sólido (SSD) y su sencillo lenguaje de consulta permite un rendimiento de consulta coherente y de baja latencia.
- Proporciona escalado del almacenamiento y permite provisionar el volumen del rendimiento de lectura o escritura que se necesita para cada tabla.
 - Las tablas de DynamoDB se pueden escalar para admitir el incremento de solicitudes de escritura y lectura mediante aprovisionamiento manual.
 - También permite escalado automático para que DynamoDB supervise la carga en la tabla e incremente o disminuya el rendimiento de aprovisionamiento de manera automática.
- Algunas funciones clave de diferenciación adicionales incluyen **las tablas globales** que permiten generar réplicas de manera automática en una selección de regiones de AWS, cifrado en reposo y visibilidad del periodo de vida (TTL) de los elementos.

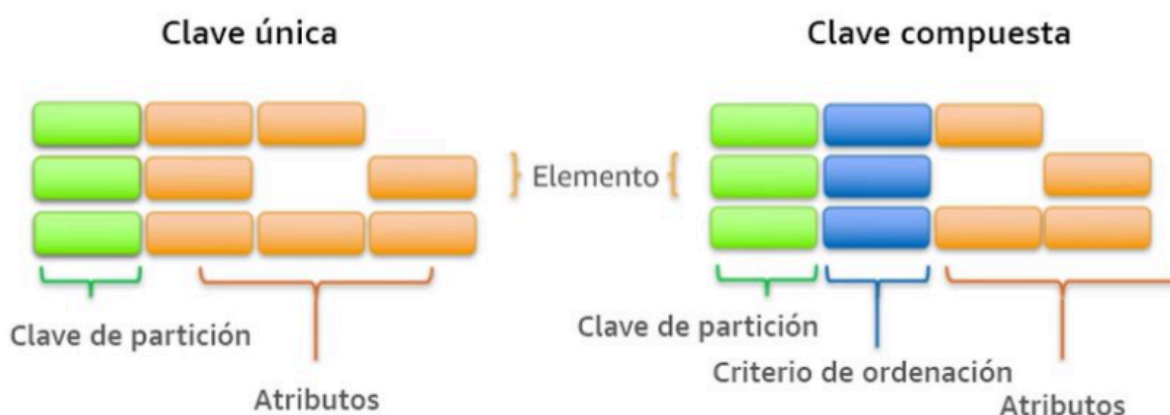
Componentes (tablas, elementos y atributos)

Los componentes principales de son **DynamoDB**:

- **Tablas:** son conjuntos de datos, formada por los elementos.
- **Elementos:** grupo de atributos que pueden identificar de forma exclusiva a un registro.
- **Atributos:** elementos de datos fundamentales que no es preciso seguir dividiendo.No todos los elementos tienen que tener todos los atributos.

DynamoDB soporta dos tipos de claves principales:

- La **clave de partición (*Partition Key*)** es una clave principal simple.
 - Para aprovechar al máximo las operaciones de consulta, es importante que la clave utilizada identifique de forma unívoca los elementos de la tabla de DynamoDB, ya que se crea automáticamente un índice desordenado sobre dicha clave.
 - Por ejemplo, un ID de producto en una tabla de producto.
- La **clave de partición y de ordenación (*Partition Key and Sort Key*)**, también conocida como clave principal compuesta, ya que está formada por dos atributos.Partition Key and Sort Key .
 - Por ejemplo: en una tabla con libros, se podría utilizar la combinación de autor y título para identificar de forma exclusiva los elementos de la tabla.
 - AWS crea dos índices, uno desordenado para la clave principal y otro índice ordenado para la clave secundaria.



Fuente: Curso *AWS Cloud Foundations*

Ejemplo:

- Tabla **usuarios** (en formato JSON)

```
{
  "user_id": "u001",
  "email": "juan@example.com",
  "nombre": "Juan Pérez",
  "edad": 30,
  "fecha_registro": "2024-05-01",
  "dirección": "Calle 1, Ciudad A"
}

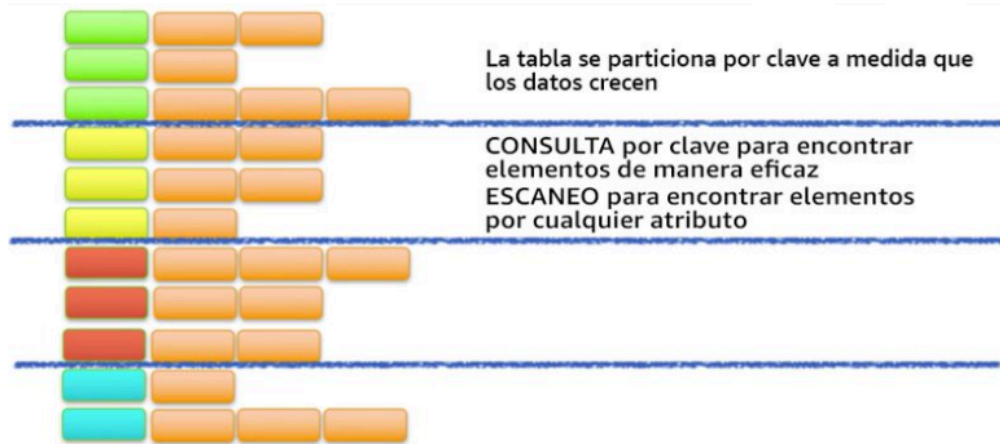
{
  "user_id": "u002",
  "email": "ana@example.com",
  "nombre": "Ana Torres",
  "edad": 32,
  "fecha_registro": "2024-08-05",
  "teléfono": "555-1111"
}
```

- La tabla tiene dos **elementos**.
- **Los atributos** son los pares **clave-valor** dentro de un elemento. En el ejemplo los atributos son:
 - **user_id**: **identificador único del usuario (clave primaria)**.
 - **email**
 - **nombre**
 - **edad**
 - **fecha_registro**
 - **dirección**
 - **teléfono**

No es obligatorio que todos los elementos tengan los mismos atributos. Puede tener un usuario con un atributo "teléfono", y otro sin él. Lo mismo pasa con "dirección".

Particiones y consultas

A medida que aumenta el volumen de datos, la clave principal particiona e indexa los datos de la tabla.



Fuente: Curso AWS Cloud Foundations

Es posible recuperar los datos de una tabla de DynamoDB de dos formas distintas

- Por la **clave haciendo una consulta directa**.
- Utilizar un **escaneo de todos los elementos** en busca de aquello que coincida con el parámetro de búsqueda. Permite localizar los elementos en la tabla a partir de las coincidencias con las condiciones en atributos que no son clave. Este método le da flexibilidad para localizar elementos por medio de otros atributos, sin embargo, esta operación es menos eficiente, debido a que DynamoDB escaneará todos los elementos en la tabla para encontrar los que coinciden con sus parámetros.

DynamoDB permite crear índices para optimizar las consultas que realicemos sobre atributos que no forman parte de la clave de partición u ordenamiento.

Tipos de datos

Respecto a los esquemas, como sistema NoSQL, cada elemento de una tabla puede contener un número arbitrario de atributos (con un nombre y un valor) con diferentes tipos de valores, pudiendo ser:

- Escalares: numérico (N), cadena (S), binario (B), booleano (BOOL) y nulo (NULL)
- Compuestos: conjunto de cadenas (SS), conjuntos numéricos (NS) y conjuntos binarios (BS)
- Basados en documentos: listas (L y mediante corchetes []) y mapas (mediante llaves {})

Consultas

Los datos de DynamoDB se almacenan en unidades SSD, y su lenguaje de consulta simple (**PartiQL**) permite un rendimiento de las consultas uniforme y de baja latencia.

Ejemplo: `SELECT * FROM "Usuarios" WHERE "user_id" = 'u001';`

Costes

Con DynamoDB se cobran las operaciones de lectura, escritura y almacenamiento de datos en sus tablas, junto con las características opcionales que decidamos habilitar. Ofrece dos modos de capacidad con opciones de facturación:

- **Bajo demanda:** se cobran las operaciones de lectura y escritura de datos realizada en las tablas. No necesitamos especificar el rendimiento de lectura y escritura que se espera de nuestras aplicaciones. Apropiado cuando:
 - Se crean nuevas tablas con cargas de trabajo desconocidas.
 - El tráfico de la aplicación es impredecible.
- **Aprovisionada:** se configura el número de operaciones de lectura y escritura por segundo que consideramos que necesitará nuestra aplicación. Permite usar el escalado automático para ajustar automáticamente la capacidad de la tabla en función de la tasa de uso especificada. Apropiado cuando:
 - El tráfico de la aplicación es predecible.
 - Las aplicaciones tienen un tráfico uniforme o aumenta gradualmente.

Los requisitos de capacidad se pueden predecir para controlar los costos.

Se pueden consultar todos los detalles en <https://aws.amazon.com/es/dynamodb/pricing/>

Referencias

- Curso Academy Cloud Foundation de Amazon Web Services.
- Aitor Medrano. Especialización en Inteligencia Artificial y Big Data (IABD) <https://aitor-medrano.github.io/iabd/>. Licencia CC BY-NC-SA 4.0.
- OpenAI. (2025). Chat GPT.
- <https://aprendiendoaws.com/>