

Implementación y consulta de una tabla de DynamoDB

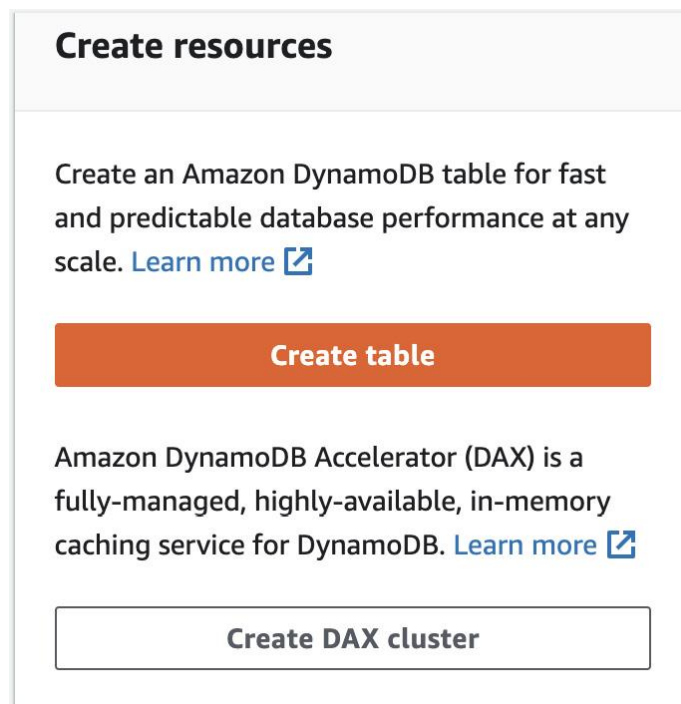
Ahora que hemos aprendido sobre DynamoDB y sus características, vamos a implementar nuestra propia tabla de DynamoDB para practicar el uso de la consola y la API. Usaremos la tabla GameScores para construir una base de datos de clasificación simple. Usaremos tanto la consola como la CLI de AWS para estos pasos.

Aprovisionamiento de una tabla de DynamoDB

Comenzaremos por aprovisionar una tabla de DynamoDB. Usaremos la región **de Oregon**:

Abra la consola de AWS en un navegador de Internet e inicie sesión con una cuenta que tenga privilegios para crear y modificar una tabla de DynamoDB.

Navigate a la sección DynamoDB. Haga clic en el botón naranja **Crear tabla** en el lado derecho de la pantalla:



Esto abrirá la página **Crear tabla**, permitiéndonos ingresar los detalles de nuestra tabla. Elegirlas siguientes opciones. Cualquier opción que no se mencione aquí, déjela como predeterminada:

- **Nombre de la tabla:** GameScores.
- **Clave de partición:** PlayerID.
- **Ordenar clave:** GameName.
- **Configuración:** personaliza la configuración.
- **Ajustes de capacidad de lectura/escritura:** on-demand.
- **Índices secundarios:** Dejar en blanco.
- **Cifrado en reposo:** Propiedad de Amazon DynamoDB.
- Haz clic en **Crear tabla**.

Tenga en cuenta que no agregamos ningún atributo excepto las claves de partición y clasificación, ya que solo se agregan cuando se crean nuevos registros en la tabla debido a la capacidad de DynamoDB. para manejar datos no estructurados.

La tabla se crea inmediatamente. Podemos usar la CLI de AWS para verificar el estado de la tabla:

```
aws dynamodb describe-table --table-name GameScore
```

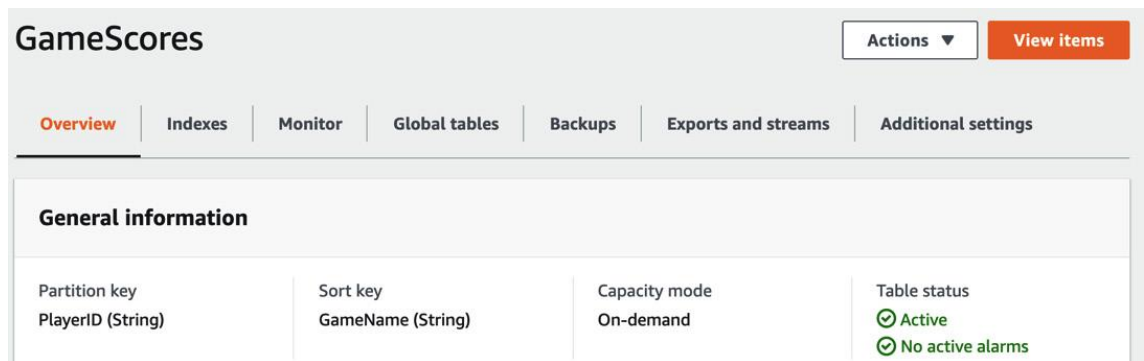
Ahora hemos creado nuestra tabla llamada GameScores y estamos listos para agregar algunos elementos.

Agregar elementos

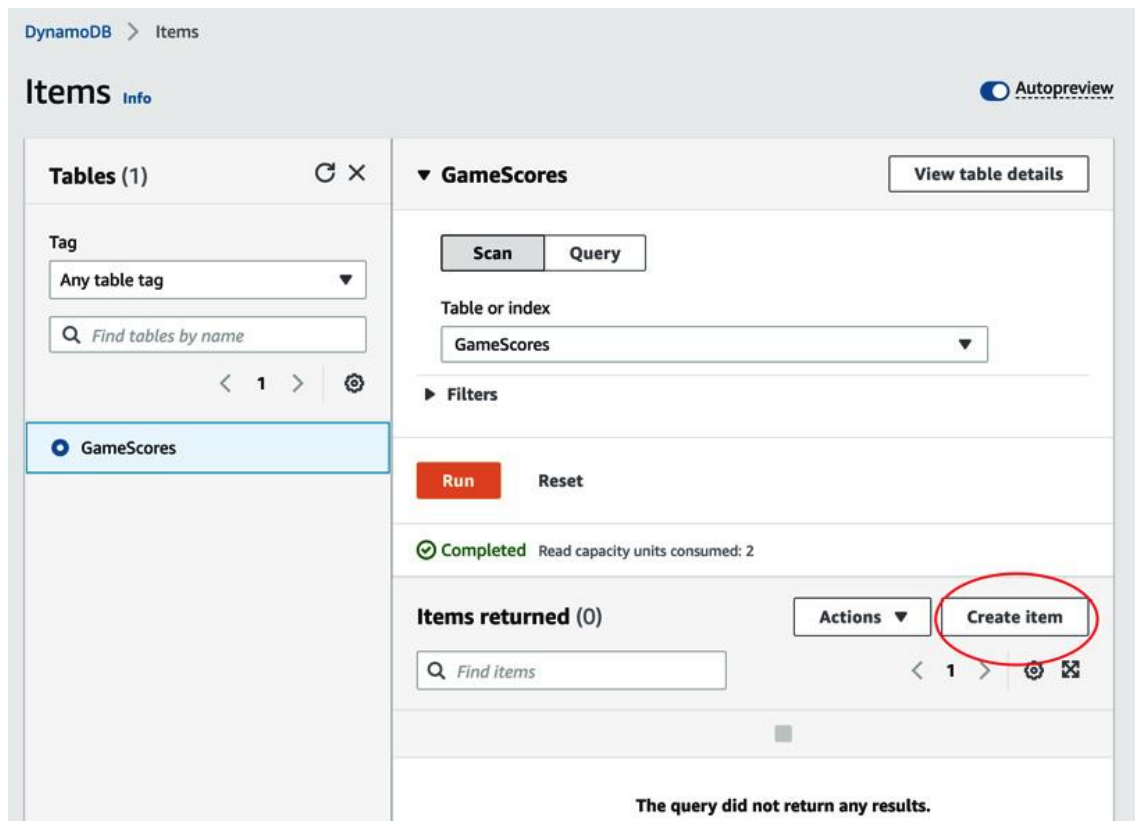
Ahora tenemos nuestra tabla vacía, podemos agregar algunos elementos. Podemos usar la API para hacer esto o podemos usar la consola. Comencemos con la consola:

Desde el panel de DynamoDB, haga clic en **Tablas** en el lado izquierdo y haga clic en la tabla GameScores.

Haz clic en **Ver elementos** en la parte superior izquierda de la pantalla:



Esto abre una página que permite ejecutar consultas en la tabla de DynamoDB, pero primero debemos agregar algunos elementos. Haga clic en **Crear elemento** como se muestra en la siguiente captura de pantalla:



Agregue algunos artículos a la tabla; necesitaremos agregar ambos PlayerID y GameName. También agregaremos un atributo en este punto. Haga clic en el menú desplegable **Agregar nuevo atributo** y **seleccione Número**. Llamaremos a esto HighScore. Complete los tres campos y luego haga clic en **Agregar elemento**. Si lo deseamos, podemos repetir este paso para añadir más registros.

Una vez hecho esto, podemos ver los elementos creados en la tabla. Tenga en cuenta el mensaje en el medio de la página que dice lo siguiente:

Consultar y escanear la tabla

Ahora que tenemos un par de registros en nuestra tabla, podemos comenzar a ejecutar una consulta y escanear para ver cómo se comportan.

Vamos a usar la API para ejecutar la consulta. Aquí, vamos a consultar la puntuación de *JohnS* en el juego *Bug Hunt* cambiarlo según los registros que se han agregado en el paso anterior:

```
aws dynamodb query \
  --table-name GameScores \
  --key-condition-expression "PlayerID = :playerid AND GameName = :gamename" \
  --expression-attribute-values '{
    ":playerid": { "S": "JohnS" },
    ":gamename": { "S": "Bug Hunt" }
  }' \
  --projection-expression "PlayerID, GameName, HighScore"
```

Tenga en cuenta que solo podemos consultar elementos que son parte de la clave, pero podemos filtrar por otros atributos si se devuelven en el conjunto de datos original. Entonces, por ejemplo, podemos agregar un filtro adicional para obtener solo los puntajes mayores a 1,000:

```
aws dynamodb query \
  --table-name GameScores \
  --key-condition-expression "PlayerID = :playerid AND GameName = :gamename" \
  --expression-attribute-values '{
    ":playerid": { "S": "JohnS" },
```

```
":gamename": { "S": "Bug Hunt" }  
}  
  
--filter-expression "HighScore > :score"  
  
--projection-expression "PlayerID, GameName, HighScore"
```

Ahora hemos aprendido cómo usar consultas para recuperar rápidamente registros usando la clave de partición, así que ahora aprendamos cómo usar comandos de escaneo para recuperar registros usando otras columnas que no sean la clave de partición.

Queremos conocer a todos los jugadores. quien obtuvo más de 100, pero como el atributo HighScore no es parte de la clave, no podemos obtenerlo a través de una consulta, por lo que necesitamos usar un escaneo. Recuerde que los escaneos implican verificar cada elemento de la tabla, por lo que puede ser lento y costoso. y debe evitarse si es posible:

```
aws dynamodb scan \  
--table-name GameScores \  
--filter-expression "HighScore > :score" \  
--projection-expression "PlayerID, GameName" \  
--expression-attribute-values '{":score":{"N":"1000"}}'
```

Una mejor manera de hacer este tipo de consulta sería agregar un GSI y usarlo en lugar de confiar en un escaneo. Hagamos eso ahora.

En la sección anterior, teníamos que usar un escaneo para obtener los datos que queríamos ya que HighScore no era parte de nuestra clave. Si agregamos un nuevo GSI a la tabla, podemos convertirlo en una clave.

Primero, piense en cómo queremos que se devuelvan los datos. Como estamos almacenando datos para varios juegos, es poco probable que queramos que nuestra consulta arroje todos los puntajes, sino los puntajes de un juego específico, por lo que debemos crear un GSI con la clave de partición GameName y HighScore como la clave de ordenación.

El beneficio de hacer esto es que las puntuaciones se almacenan ordenadas, por lo que si queremos obtener una instancia de HighScore, simplemente podemos obtener la primera clave:

Use la API (o la consola si lo prefiere) para crear un nuevo GSI:

```
aws dynamodb update-table \
  --table-name GameScore \
  --attribute-definitions '[
    {AttributeName=GameName,AttributeType=S},
    {AttributeName=HighScore,AttributeType=N}
  ]'
  --global-secondary-index-updates '[
    {"Create": {"IndexName": "GameTitleScores","KeySchema":[
      {"AttributeName":"GameName","KeyType":"HASH"}, {"AttributeName":"HighScore","KeyType":"RANGE"}],
      "Projection":{"ProjectionType":"ALL"}}]
  ]'
```

Ahora que tenemos el nuevo índice, podemos ejecutar una consulta para obtener la puntuación más alta para nuestro juego; a diferencia del caso con un SQLbase de datos, necesitamos decirle específicamente a DynamoDB qué índice usar. Recuerde que el atributo HighScore está ordenado, por lo que solo podemos obtener el primer valor devuelto usando la no-scan-index-forward configuración para invertir el orden:

```
aws dynamodb query \  
  --table-name GameScores \  
  --index-name GameTitleScores \  
  --key-condition-expression "GameName = :gameName" \  
  --expression-attribute-values '{  
    ":gameName": { "S": "Bug Hunt" }  
  }' \  
  --projection-expression "PlayerID, HighScore" \  
  --no-scan-index-forward \  
  --max-items 1 \  
'
```

Ahora hemos aprendido cómo crear una tabla de DynamoDB, insertar elementos, consultarlos y usar un escaneo.