

CS2032 - Cloud Computing (Ciclo 2024-2)

Multi-tenancy

Semana 9 - Taller 1b: Base Datos No SQL Multi-tenancy

ELABORADO POR: GERALDO COLCHADO

Contenido

Base Datos No SQL Multi-tenancy

1. **Objetivo del taller 1b**
2. DynamoDB: Clave de Partición
3. DynamoDB: Clave de Ordenamiento
4. Ejercicio 1: Sistema Académico
5. Ejercicio 2: Red Social (Ejercicio propuesto)
6. Cierre

Objetivo del taller 1b:

Base Datos No SQL Multi-tenancy

- Aprender buenas prácticas para definir Clave de Partición
- Aprender buenas prácticas para definir Clave de Ordenamiento
- Diseñar tablas en DynamoDB de Microservicios Multi-tenancy

Contenido

Base Datos No SQL Multi-tenancy

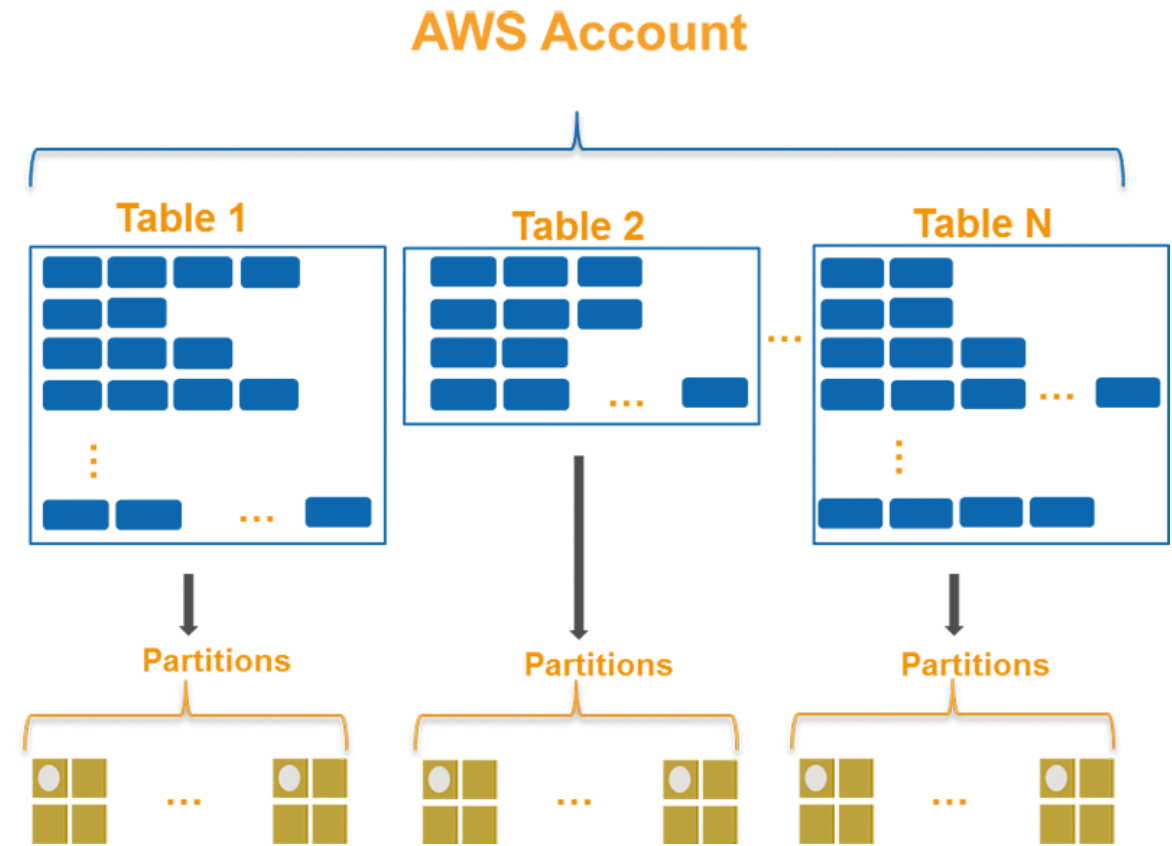
1. Objetivo del taller 1b
2. **DynamoDB: Clave de Partición**
3. DynamoDB: Clave de Ordenamiento
4. Ejercicio 1: Sistema Académico
5. Ejercicio 2: Red Social (Ejercicio propuesto)
6. Cierre

DynamoDB: Clave de Partición

Concepto

DynamoDB utiliza el valor de la *clave de partición* como parámetro de entrada para una *función hash* interna. El resultado de la función hash *determina la partición* en la que se almacena el elemento. La ubicación de cada elemento viene determinada por el valor hash de su clave de partición.

Los elementos se distribuyen entre *unidades de almacenamiento de 10 GB cada una*, denominadas *particiones* (almacenamiento físico interno de DynamoDB). *Cada tabla tiene una o más particiones*.



DynamoDB: Clave de Partición

Buena práctica

Recomendaciones para claves de partición:

Utilice atributos de cardinalidad alta (gran cantidad de valores únicos). Se trata de atributos que tienen valores distintos para cada elemento como, por ejemplo:

- *dirección de correo electrónico*
- *código de empleado*
- *código de cliente*
- *número de sesión*
- *número de orden*

The following is a comparison of the provisioned throughput efficiency of some common partition key schemas.

Partition key value	Uniformity
User ID, where the application has many users.	Good
Status code, where there are only a few possible status codes.	Bad
Item creation date, rounded to the nearest time period (for example, day, hour, or minute).	Bad
Device ID, where each device accesses data at relatively similar intervals.	Good
Device ID, where even if there are many devices being tracked, one is by far more popular than all the others.	Bad

If a single table has only a small number of partition key values, consider distributing your write operations across more distinct partition key values. In other words, structure the primary key elements to avoid one "hot" (heavily requested) partition key value that slows overall performance.

Contenido

Base Datos No SQL Multi-tenancy

1. Objetivo del taller 1b
2. DynamoDB: Clave de Partición
3. **DynamoDB: Clave de Ordenamiento**
4. Ejercicio 1: Sistema Académico
5. Ejercicio 2: Red Social (Ejercicio propuesto)
6. Cierre

DynamoDB: Clave de Ordenamiento

Concepto y Buenas Prácticas

*En DynamoDB una **clave primaria** puede ser:*

- *Clave de partición*
- *Clave de partición + Clave de ordenamiento*

*La clave primaria debe ser **única en toda la tabla**.*

*La **clave de ordenamiento** permite organizar los datos*

Best practices for using sort keys to organize data in DynamoDB

[PDF](#) | [RSS](#)

In an Amazon DynamoDB table, the primary key that uniquely identifies each item in the table can be composed of a partition key and a sort key.

Well-designed sort keys have two key benefits:

- They gather related information together in one place where it can be queried efficiently. Careful design of the sort key lets you retrieve commonly needed groups of related items using range queries with operators such as `begins_with`, `between`, `>`, `<`, and so on.
- Composite sort keys let you define hierarchical (one-to-many) relationships in your data that you can query at any level of the hierarchy.

For example, in a table listing geographical locations, you might structure the sort key as follows.

```
[country]#[region]#[state]#[county]#[city]#[neighborhood]
```

This would let you make efficient range queries for a list of locations at any one of these levels of aggregation, from `country`, to a `neighborhood`, and everything in between.

DynamoDB: Clave de Ordenamiento

Ejemplo

La **Clave de ordenamiento** es útil para ampliar las **opciones de consulta de datos**. Ejemplo **Tabla ChatMessages**:

- **Clave de Partición:** chatroomId
- **Clave de Ordenamiento:** user_name#datetime

Table name* ⓘ

Primary key*

Partition key

String ⓘ

☒ Add sort key

String ⓘ

	chatRoomId	chatMessageIdentifier	message
<input type="checkbox"/>	houston	amsg#2011-07-09:13:01:01	hi there
<input type="checkbox"/>	houston	tay#2011-07-09:13:01:02	: -)
<input type="checkbox"/>	seattle	amsg#2017-07-22:15:14:13	that was deli...
<input type="checkbox"/>	seattle	amsg#2018-08-29:12:25:32	okay! how ca...
<input type="checkbox"/>	seattle	tay#2018-08-29:12:23:32	pleaseeee ca...

DynamoDB: Clave de Ordenamiento

Ejemplo

Query [Table] ChatMessages: chatRoomId, chatMessageIdentifier

Partition key chatRoomId String = seattle

Sort key chatMessageIdentifier String Begins with amsg

+ Add filter

Sort ☒ Ascending ☐ Descending

Attributes ☒ All ☐ Projected

Start search

chatRoomId	chatMessageIdentifier
seattle	amsg#2017-07-22:15:14:13
seattle	amsg#2018-08-29:12:25:32

Query [Table] ChatMessages: chatRoomId, chatMessageIdentifier

Partition key chatRoomId String = seattle

Sort key chatMessageIdentifier String Begins with amsg#2018

+ Add filter

Sort ☒ Ascending ☐ Descending

Attributes ☒ All ☐ Projected

Start search

chatRoomId	chatMessageIdentifier
seattle	amsg#2018-08-29:12:25:32

Contenido

Base Datos No SQL Multi-tenancy

1. Objetivo del taller 1b
2. DynamoDB: Clave de Partición
3. DynamoDB: Clave de Ordenamiento
4. **Ejercicio 1: Sistema Académico**
5. Ejercicio 2: Red Social (Ejercicio propuesto)
6. Cierre

Ejercicio 1: Sistema Académico

Propuesta de diseño Multi-tenancy con DynamoDB - Alternativa 1

Microservicio **Alumnos**

Clave Partición tenant_id	Clave Ordenamiento alumno_id	Datos
UTEC	202310424	{“ape_pat”: “Castro”, “ape_mat”: “Coa”, ...}
UNIV2	F202400123	{“ape_pat”: “Seminario”, “ape_mat”: “Lezama”, ...}

Microservicio **Cursos**

Clave Partición tenant_id	Clave Ordenamiento curso_id	Datos
UTEC	CS2032	{“nombre”: “Cloud Computing”}
UTEC	CS2033	{“nombre”: “Base de Datos”}
UTEC	CS2034	{“nombre”: “Machine Learning”}

Microservicio **Matriculas** (Incluye evaluaciones con sus notas)
Patrón de acceso para consultas por **alumno_id**

Clave Partición tenant_id#alumno_id	Clave Ordenamiento ciclo#curso_id#seccion_id#eval_id	Datos
UTEC#202310424	2024-2#CS2032#1	{“fecha_matricula”: “2024-07-24”}
UTEC#202310424	2024-2#CS2032#1#PA1	{“fecha_registro”: “2024-10-13”, “registrado_por”: “gcolchado”, “nota”: 18}
UTEC#202310424	2024-2#CS2032#1#TP	{“fecha_registro”: “2024-10-14”, “registrado_por”: “gcolchado”, “nota”: 15}
UTEC#202310424	2024-2#CS2032#1#EP	{“fecha_registro”: “2024-10-10”, “registrado_por”: “gcolchado”, “nota”: 13}

Ejercicio 1: Sistema Académico

Propuesta de diseño Multi-tenancy con DynamoDB - Alternativa 2

Microservicio **Alumnos**

Clave Partición tenant_id	Clave Ordenamiento alumno_id	Datos
UTEC	202310424	{“ape_pat”: “Castro”, “ape_mat”: “Coa”, ...}
UNIV2	F202400123	{“ape_pat”: “Seminario”, “ape_mat”: “Lezama”, ...}

Microservicio **Cursos**

Clave Partición tenant_id	Clave Ordenamiento curso_id	Datos
UTEC	CS2032	{“nombre”: “Cloud Computing”}
UTEC	CS2033	{“nombre”: “Base de Datos”}
UTEC	CS2034	{“nombre”: “Machine Learning”}

Microservicio **Matriculas** (Incluye evaluaciones con sus notas)
Patrón de acceso para consultas por **curso_id**

Clave Partición tenant_id#curso_id	Clave Ordenamiento ciclo#seccion_id#alumno_id#eval_id	Datos
UTEC#CS2032	2024-2#1#202310424	{“fecha_matricula”: “2024-07-24”}
UTEC#CS2032	2024-2#1#202310424#PA1	{“fecha_registro”: “2024-10-13”, “registrado_por”: “gcolchado”, “nota”: 18}
UTEC#CS2032	2024-2#1#202310424#TP	{“fecha_registro”: “2024-10-14”, “registrado_por”: “gcolchado”, “nota”: 15}
UTEC#CS2032	2024-2#1#202310424#EP	{“fecha_registro”: “2024-10-10”, “registrado_por”: “gcolchado”, “nota”: 13}

Ejercicio 1: Sistema Académico

Propuesta de diseño Multi-tenancy con DynamoDB - Alternativa 3

Amazon DynamoDB supports two types of secondary indexes:

- **Global secondary index (GSI)** - An index with a partition key and a sort key that can be different from those on the base table.
- **Local secondary index (LSI)** - An index that has the same partition key as the base table, but a different sort key.

Microservicio **Matriculas** (Incluye evaluaciones con sus notas)
Patrón de acceso para consultas por **alumno_id** y por **curso_id** con **Global secondary index (GSI)**

Clave Partición tenant_id#alumno_id	Clave Ordenamiento ciclo#curso_id#seccion_id#eval_id	Clave Partición 2 (GSI) tenant_id#curso_id	Clave Ordenamiento 2 (GSI) ciclo#seccion_id#alumno_id#eval_id	Datos
UTEC#202310424	2024-2#CS2032#1	UTEC#CS2032	2024-2#1#202310424	{"fecha_matricula": "2024-07-24"}
UTEC#202310424	2024-2#CS2032#1#PA1	UTEC#CS2032	2024-2#1#202310424#PA1	{"fecha_registro": "2024-10-13", "registrado_por": "gcolchado", "nota": 18}
UTEC#202310424	2024-2#CS2032#1#TP	UTEC#CS2032	2024-2#1#202310424#TP	{"fecha_registro": "2024-10-14", "registrado_por": "gcolchado", "nota": 15}
UTEC#202310424	2024-2#CS2032#1#EP	UTEC#CS2032	2024-2#1#202310424#EP	{"fecha_registro": "2024-10-10", "registrado_por": "gcolchado", "nota": 13}

Contenido

Base Datos No SQL Multi-tenancy

1. Objetivo del taller 1b
2. DynamoDB: Clave de Partición
3. DynamoDB: Clave de Ordenamiento
4. Ejercicio 1: Sistema Académico
5. **Ejercicio 2: Red Social (Ejercicio propuesto)**
6. Cierre

Ejercicio 2: Red Social

Ejercicio propuesto

Elabore una propuesta de diseño Multi-tenancy con DynamoDB para una red social (Ejemplo: Instagram, TikTok)

El tenant de una Red Social es el usuario identificado por su correo.

Contenido

Base Datos No SQL Multi-tenancy

1. Objetivo del taller 1b
2. DynamoDB: Clave de Partición
3. DynamoDB: Clave de Ordenamiento
4. Ejercicio 1: Sistema Académico
5. Ejercicio 2: Red Social (Ejercicio propuesto)
6. **Cierre**

Cierre:

Base Datos No SQL Multi-tenancy - Qué aprendimos?

- Aprender buenas prácticas para definir Clave de Partición
- Aprender buenas prácticas para definir Clave de Ordenamiento
- Diseñar tablas en DynamoDB de Microservicios Multi-tenancy

Gracias

Elaborado por docente: Geraldo Colchado