DynamoDB Schema Design

1. Introduction

This document provides a schema design for a DynamoDB table intended to store metadata about crawled web pages. The schema is designed to support efficient querying based on various attributes such as URL, date range, page title, and word count.

2. Notes and considerations

- Understanding the business problems and the application use cases up front is essential.
- The first step in designing your DynamoDB application is to identify the specific query patterns that the system must satisfy

Based on the statement, the DynamoDB table should support efficient querying for the following query patterns:

1. Query by URL: The system should be able to quickly retrieve all metadata records for a given URL. This is useful for tracking how a specific web page's metadata changes over time.

2. Query by date range: The system should be able to fetch all metadata records that were crawled within a specific date range. This is important for analyzing trends or changes over a period of time.

3. Query by Specific Attribute:

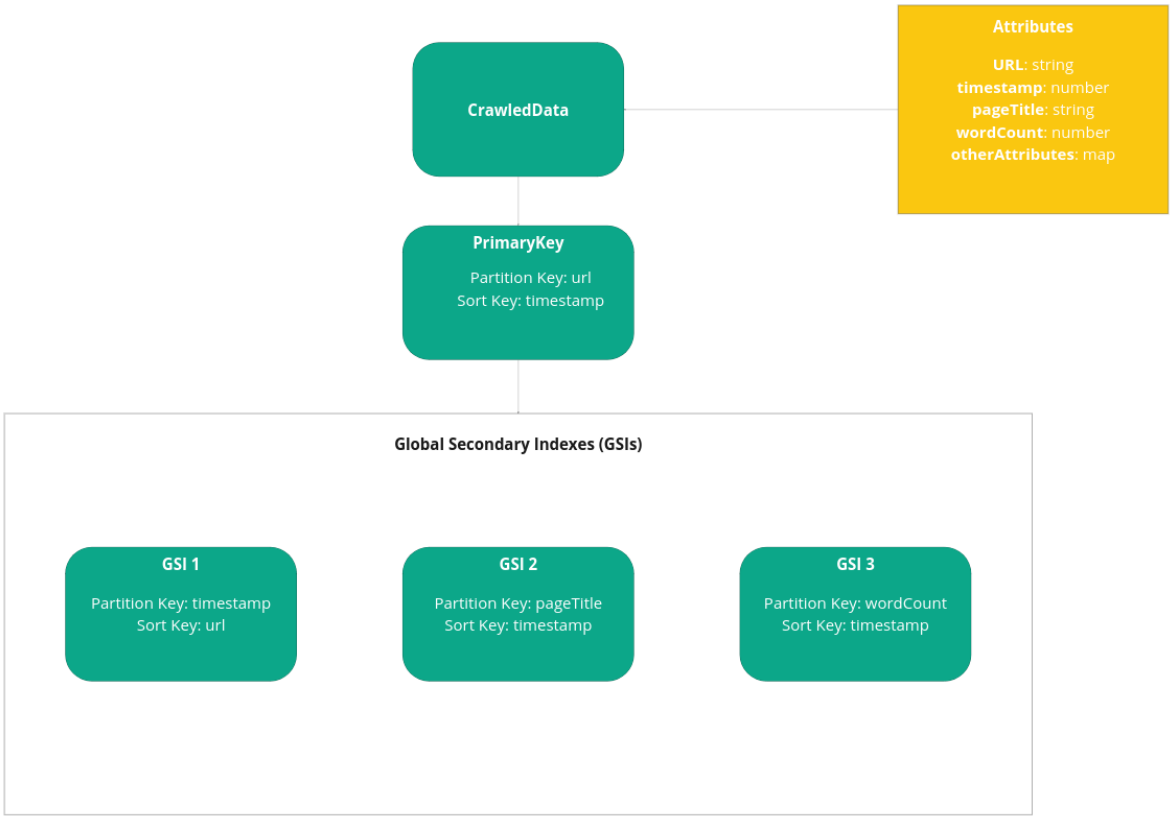
   By page title: The system should be able to find all records that have a specific page title. This could be useful for identifying duplicate or similar content across different URLs.

   By word count: The system should be able to find all records that have a specific word count. This could be useful for filtering content based on length or depth.

These query patterns guides the design of the primary key and any Global Secondary Indexes (GSIs) we might need

Global Secondary Indexes (GSIs) are additional tables that allow you to query your data using attributes other than those in the primary key. They offer a way to optimize your database for various query patterns, providing flexibility and performance improvements.

3. Schema



Miro board here

4. Alternative Solutions

Additional GSIs could be created with pageTitle and wordCount as partition keys to support queries based on these specific attributes.

5. Cost Implications

Storage Costs: Each GSI consumes additional storage.

Read/Write Costs: The cost is based on the read and write capacity units, which should be planned according to the query patterns.

6. References

- AWS DynamoDB Documentation