Sprint 1 – DynamoDB Indexing & Searching Documentation

As a developer, I want to document how indexing and searching could work in DynamoDB so stories can be retrieved efficiently.

This document provides the Sprint 1 deliverable for DynamoDB indexing and searching in the AMW project.”

Why Indexing Matters

DynamoDB is a NoSQL database. Unlike traditional SQL databases, it cannot run broad “find everything” queries. Instead, data must be organised in advance using partition keys, sort keys, and indexes. Indexes provide shortcuts so that common queries can be performed quickly and without scanning the whole table.

Access Patterns for AMW

The A Moment With (AMW) web app requires:

1. Get a user by email (login).
2. List all stories by a user (dashboard).
3. Get a story by ID (view story).
4. Search stories by title (basic search).
5. List chapters by story (ordered sequence).
6. List media (photos, audio, video) by story or chapter.
7. (Optional) Show the latest N stories sitewide (recent feed).

Table Keys (Single-Table Design)

- **Partition Key (PK):** groups related items.
- **Sort Key (SK):** orders items within the group.

Example:

- `PK = STORY#123`
- `SK = CHAPTER#001`
  This means all chapters and media for story 123 are grouped together.

Global Secondary Indexes (GSIs)

Index	Purpose	Example Use
GSI1	User by email	Login lookup
GSI2	Stories by user	Dashboard
GSI3	Stories by title prefix	Search
GSI4	Media by story	Story gallery
GSI5	Media by chapter	Per-chapter uploads
GSI6 (optional)	Latest stories	Global feed

Search Strategy

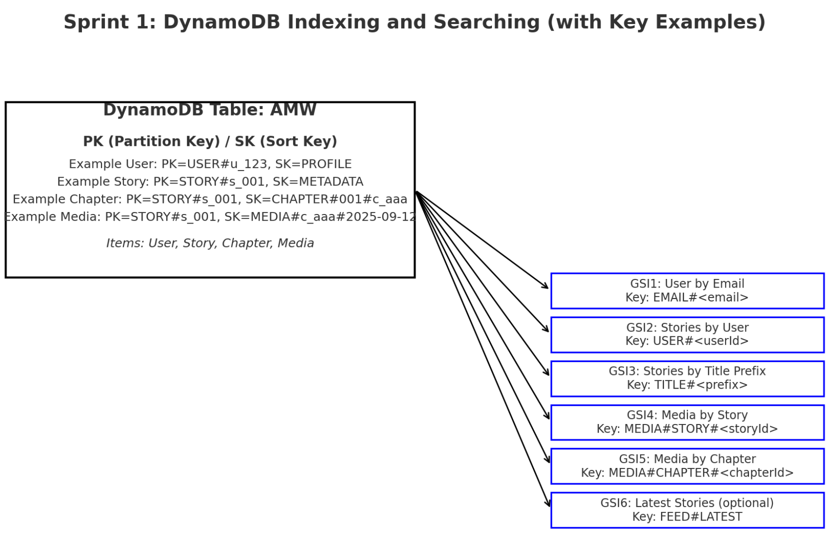
- **MVP (Prefix Search):** Titles are normalised (lowercase, no punctuation). A `titlePrefix` value is stored and queried through GSI3.
- **Future Upgrade (Full Text):** Use Amazon OpenSearch with DynamoDB Streams for advanced search, typo tolerance, and relevancy ranking.

Example Query – List Stories by User

```
aws dynamodb query \
  --table-name AMW \
  --index-name GSI2 \
  --key-condition-expression "GSI2PK = :user" \
  --expression-attribute-values '{":user":{"S":"USER#u_123"}}' \
  --scan-index-forward false
```

This query retrieves all stories owned by user `u_123`, ordered with the most recent first.

Figure 1: DynamoDB table design for AMW showing partition/sort keys and the GSIs used for efficient queries.


Sprint 1: DynamoDB Indexing and Searching (with Key Examples)
DynamoDB Table: AMW
PK (Partition Key) / SK (Sort Key)
Example User: PK=USER#u_123, SK=PROFILE
Example Story: PK=STORY#s_001, SK=METADATA
Example Chapter: PK=STORY#s_001, SK=CHAPTER#001#c_aaa
Example Media: PK=STORY#s_001, SK=MEDIA#c_aaa#2025-09-12
Items: User, Story, Chapter, Media
GSI1: User by Email
Key: EMAIL#<email>
GSI2: Stories by User
Key: USER#<userId>
GSI3: Stories by Title Prefix
Key: TITLE#<prefix>
GSI4: Media by Story
Key: MEDIA#STORY#<storyId>
GSI5: Media by Chapter
Key: MEDIA#CHAPTER#<chapterId>
GSI6: Latest Stories (optional)
Key: FEED#LATEST

7. DynamoDB as a Library (Simple Analogy)

DynamoDB can be thought of as a library. Each item of data, such as a story, chapter, or photo, is like a book, chapter, or picture in the library. To stay organised, every item is placed on a shelf. The shelf number is the **Partition Key**, and the position on the shelf is the **Sort Key**.

For example, a story is like a complete book, stored as `STORY#123` with its position set as `METADATA` for basic information. Chapters are stored on the same shelf in different positions such as `CHAPTER#001`. Photos and videos are also stored on the same shelf with positions like `MEDIA#001`. This way, all the content that belongs to one story is kept together.

Indexes in DynamoDB act like catalogs in a library. Instead of searching through every shelf, you can go to the catalog to find what you need. The **By Author** catalog shows all stories written by one user. The **By Title Prefix** catalog lets you find stories that start with certain words. The **By Date** catalog lists the most recent stories. These indexes save time and make searching efficient.

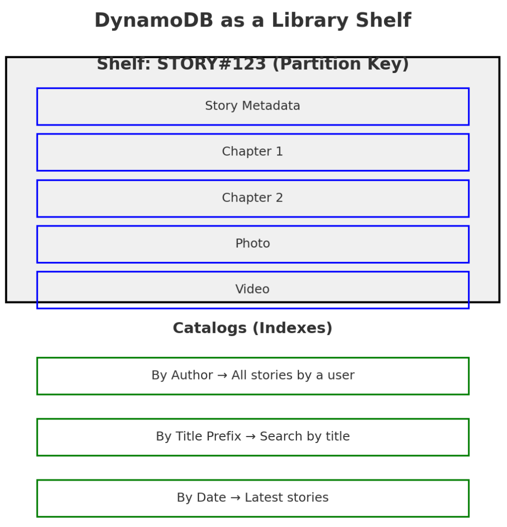

DynamoDB as a Library Shelf
Shelf: STORY#123 (Partition Key)
Story Metadata
Chapter 1
Chapter 2
Photo
Video
Catalogs (Indexes)
By Author → All stories by a user
By Title Prefix → Search by title
By Date → Latest stories

Figure 2 : ***DynamoDB as a library. A shelf groups one story’s items together, while catalogs (indexes) make it easy to search and find stories quickly.***

Outcome

This document explains

: why indexing is necessary,lists AMW’s access patterns, outlines table keys, defines required GSIs, and provides a simple search strategy. The library analogy helps illustrate the concept in an easy-to-understand way, while the technical details provide a foundation for Sprint 2 schema design and implementation.