

Part 5

Data Schema & Data Model

As discussed in part 1, we are using MongoDB, which means that our NoSQL database has a schema-less design. Nonetheless, our data does follow a schema based on its structure. As a reminder, the following is the structure of our data:

1. Playlists.json

```
{
  "playlists": [
    {
      "name": "String",
      "lastModifiedDate": "ISO 8601",
      "items": [
        {
          "track": {
            "trackName": "String",
            "artistName": "String",
            "albumName": "String",
            "trackUri": "URI"
          },
          "episode": null,
          "localTrack": null
        }
      ],
      "description": "String",
      "numberOfFollowers": "int"
    }
  ]
}
```

1. StreamingHistory.json

```
[
  {
    "endTime" : "ISO 8601",
    "artistName" : "String",
    "trackName" : "String",
    "msPlayed" : 179026
  }
]
```

1. SearchQueries.json

```
[
  {
    "platform" : "String",
    "searchTime" : "ISO 8601",
    "searchQuery" : "String",
    "searchInteractionURIs" : [
      "URI"
    ]
  }
]
```

Implementing CRUD operations

This section provides an overview of the basic Create, Read, Update, and Delete (CRUD) operations implemented in a Python script for interacting with a MongoDB database. The operations are tailored for a specific use case involving a music streaming service, with collections like playlists, search_queries, and streaming_history.

Create

- Function Name: create
- Purpose: Inserts documents into a specified MongoDB collection.
- Specialized Inner Functions: create_search_query for search_queries (Validates the presence of searchTime; Checks for existing documents with the same searchTime to avoid duplicates; Inserts a single document if it passes the checks), create_streaming_history for streaming_history (Validates the presence of required fields: endTime, artistName, trackName, msPlayed; Ensures no duplicate records based on these fields; Inserts a single document if it is unique), create_playlist for playlists (Checks for a unique name; Validates the items field to ensure it is a list; Checks for duplicate tracks within the same playlist; Inserts a single playlist document if all checks pass).
- Usage: The function can handle both single document insertion and bulk insertion of multiple documents. Based on the collection.name, the function determines which specialized insertion logic to apply. This allows for collection-specific validation and insertion rules. If the collection is not one of the specified types (playlists, search_queries, streaming_history), the function defaults to either insert_one (for single documents) or insert_many (for lists). The function is versatile and can be used for different types of data insertion tasks in the specified collections, ensuring data integrity through validation and duplication checks.

- Limitations: While the function handles duplicates and basic validations, it might not cover every edge case or complex data structure, particularly for deeply nested documents or arrays. Additionally, the function is designed specifically for the collections and fields mentioned. For other collections or different data structures, additional customization would be required.

Read

- Function Name: read
- Purpose: Fetches documents from a specified MongoDB collection based on a query.
- Usage: The function allows querying documents with specified conditions and limits the number of documents returned. It uses the find method with optional query parameters.
- Limitations: The read function is designed for basic queries. It does not handle complex queries involving nested arrays, subdocuments, or aggregation operations, which are common in more intricate data models.

Update

- Function Name: update
- Purpose: Modifies existing documents in a MongoDB collection.
- Usage: The function can update a single document or multiple documents based on a query condition. It uses either update_one or update_many depending on the update_many flag.
- Limitations: The generic nature of the update function means it is not suited for complex updates, especially those involving nested arrays or the need for atomic operations. It's primarily for straightforward field updates.

Delete

- Function Name: delete
- Purpose: Removes documents from a specified MongoDB collection.
- Usage: The function can delete a single document or multiple documents based on a query condition. It uses either delete_one or delete_many depending on the delete_many flag.
- Limitations: Similar to update, the delete function is designed for straightforward use cases. It does not cater to scenarios where conditional logic is needed to

determine which elements of an array within a document should be removed.

While the basic CRUD operations cover a wide range of typical database interactions, there are scenarios where more complex and intricate queries are needed, particularly in sophisticated data models:

Nested Documents and Arrays:

- Operations involving nested documents or arrays, such as updating a specific item within an array or filtering an array based on certain conditions, require more complex queries using operators like \$elemMatch, \$push, \$pull, or aggregation pipelines.

Aggregation and Data Analysis:

- For data analysis or operations that involve grouping, filtering, and processing data across multiple stages, MongoDB's aggregation framework is needed. This goes beyond basic CRUD and involves building pipelines of data processing stages.

Sample Query Operations

CRUD operations were performed on the following sample playlist, along with their respective query results & outputs:

```
sample_playlist = {
  "name": "Chill Vibes",
  "lastModifiedDate": "2023-03-15",
  "items": [
    {
      "track": {
        "trackName": "COPYCAT",
        "artistName": "Billie Eilish",
        "albumName": "dont smile at me",
        "trackUri": "spotify:track:5w7wuzMzsDer96KqxaferK"
      }
    },
    {
      "track": {
        "trackName": "bloodline",
        "artistName": "Ariana Grande",
        "albumName": "thank u, next",
        "trackUri": "spotify:track:2hloaUoRonYssMuqLCBLTX"
      }
    },
    {
      "track": {
        "trackName": "Boyfriend",
        "artistName": "Dove Cameron",
        "albumName": "Boyfriend",

```

```

        "trackUri": "spotify:track:59CfNbKERJ3NoTXDvoURjj"
      }
    },
    {
      "track": {
        "trackName": "needy",
        "artistName": "Ariana Grande",
        "albumName": "thank u, next",
        "trackUri": "spotify:track:1TEL6MlSSVLSdh0Sddid1J"
      }
    }
  ],
  "description": "A playlist for relaxing and unwinding.",
  "numberOfFollowers": 5
}

```

CREATE:

Data to be Inserted:

```

{'name': 'Chill Vibes', 'lastModifiedDate': '2023-03-15', 'items': [{'track': {'trackName': 'COPYCAT', 'artistName': 'Billie Eilish', 'albumName': 'dont smile at me', 'trackUri': 'spotify:track:5w7wuzMzsDer96KqxaferK'}}, {'track': {'trackName': 'bloodline', 'artistName': 'Ariana Grande', 'albumName': 'thank u, next', 'trackUri': 'spotify:track:2hloaUoRonYssMuqLCBLTX'}}, {'track': {'trackName': 'Boyfriend', 'artistName': 'Dove Cameron', 'albumName': 'Boyfriend', 'trackUri': 'spotify:track:59CfNbKERJ3NoTXDvoURjj'}}, {'track': {'trackName': 'needy', 'artistName': 'Ariana Grande', 'albumName': 'thank u, next', 'trackUri': 'spotify:track:1TEL6MlSSVLSdh0Sddid1J'}}], 'description': 'A playlist for relaxing and unwinding.', 'numberOfFollowers': 5}

```

Query Data Inserted:

```

[{'_id': ObjectId('6564d7f8bd9b2c34357c08f6'), 'name': 'Chill Vibes', 'lastModifiedDate': '2023-03-15', 'items': [{'track': {'trackName': 'COPYCAT', 'artistName': 'Billie Eilish', 'albumName': 'dont smile at me', 'trackUri': 'spotify:track:5w7wuzMzsDer96KqxaferK'}}, {'track': {'trackName': 'bloodline', 'artistName': 'Ariana Grande', 'albumName': 'thank u, next', 'trackUri': 'spotify:track:2hloaUoRonYssMuqLCBLTX'}}, {'track': {'trackName': 'Boyfriend', 'artistName': 'Dove Cameron', 'albumName': 'Boyfriend', 'trackUri': 'spotify:track:59CfNbKERJ3NoTXDvoURjj'}}, {'track': {'trackName': 'needy', 'artistName': 'Ariana Grande', 'albumName': 'thank u, next', 'trackUri': 'spotify:track:1TEL6MlSSVLSdh0Sddid1J'}}], 'description': 'A playlist for relaxing and unwinding.', 'numberOfFollowers': 5}]

```

READ:

Query First Track in Playlist:

```
{'_id': ObjectId('6564d7f8bd9b2c34357c08f6'), 'name': 'Chill Vibes', 'lastModifiedDate': '2023-03-15', 'items': [{'track': {'trackName': 'COPYCAT', 'artistName': 'Billie Eilish', 'albumName': 'dont smile at me', 'trackUri': 'spotify:track:5w7wuzMzsDer96KqxaFeRK'}}], 'description': 'A playlist for relaxing and unwinding.', 'numberOfFollowers': 5}
```

Query All Ariana Grande Tracks:

```
[{'_id': ObjectId('6564d7f8bd9b2c34357c08f6'), 'items': [{'track': {'trackName': 'bloodline', 'artistName': 'Ariana Grande', 'albumName': 'thank u, next', 'trackUri': 'spotify:track:2hloaUoRonYssMuqLCBLTX'}}, {'track': {'trackName': 'needy', 'artistName': 'Ariana Grande', 'albumName': 'thank u, next', 'trackUri': 'spotify:track:1TEL6MlSSVLSdhOSddidlJ'}}]]]
```

UPDATE:

```
Update "COPYCAT" to "bellyache" with trackUri "spotify:track:51NFXnQvaosfDDutk0tams":  
[{'_id': ObjectId('6564d7f8bd9b2c34357c08f6'), 'name': 'Chill Vibes', 'lastModifiedDate': '2023-03-15', 'items': [{'track': {'trackName': 'bellyache', 'artistName': 'Billie Eilish', 'albumName': 'Bellyache', 'trackUri': 'spotify:track:51NFXnQvaosfDDutk0tams'}}, {'track': {'trackName': 'bloodline', 'artistName': 'Ariana Grande', 'albumName': 'thank u, next', 'trackUri': 'spotify:track:2hloaUoRonYssMuqLCBLTX'}}, {'track': {'trackName': 'Boyfriend', 'artistName': 'Dove Cameron', 'albumName': 'Boyfriend', 'trackUri': 'spotify:track:59CfNbKERJ3NoTXDvoURjj'}}, {'track': {'trackName': 'needy', 'artistName': 'Ariana Grande', 'albumName': 'thank u, next', 'trackUri': 'spotify:track:1TEL6MlSSVLSdhOSddidlJ'}}], 'description': 'A playlist for relaxing and unwinding.', 'numberOfFollowers': 5}]
```

Update the description of the playlist to say "My favorite songs.":

```
[{'_id': ObjectId('6564d7f8bd9b2c34357c08f6'), 'name': 'Chill Vibes', 'lastModifiedDate': '2023-03-15', 'items': [{'track': {'trackName': 'bellyache', 'artistName': 'Billie Eilish', 'albumName': 'Bellyache', 'trackUri': 'spotify:track:51NFXnQvaosfDDutk0tams'}}, {'track': {'trackName': 'bloodline', 'artistName': 'Ariana Grande', 'albumName': 'thank u, next', 'trackUri': 'spotify:track:2hloaUoRonYssMuqLCBLTX'}}, {'track': {'trackName': 'Boyfriend', 'artistName': 'Dove Cameron', 'albumName': 'Boyfriend', 'trackUri': 'spotify:track:59CfNbKERJ3NoTXDvoURjj'}}, {'track': {'trackName': 'needy', 'artistName': 'Ariana Grande', 'albumName': 'thank u, next', 'trackUri': 'spotify:track:1TEL6MlSSVLSdhOSddidlJ'}}], 'description': 'My favorite songs.', 'numberOfFollowers': 5}]
```

DELETE:

Delete "needy":

```
[{'_id': ObjectId('6564d7f8bd9b2c34357c08f6'), 'name': 'Chill Vibes', 'lastModifiedDate': '2023-03-15', 'items': [{'track': {'trackName': 'bellyache', 'artistName': 'Billie Eilish', 'albumName': 'Bellyache', 'trackUri': 'spotify:track:51NFXnQvaosfDDutk0tams'}}, {'track': {'trackName': 'bloodline', 'artistName': 'Ariana Grande', 'albumName': 'thank u, next', 'trackUri': 'spotify:track:2hloaUoRonYssMuqLCBLTX'}}, {'track': {'trackName': 'Boyfriend', 'artistName': 'Dove Cameron', 'albumName': 'Boyfriend', 'trackUri': 'spotify:track:59CfNbkerJ3NoTXDvoURjj'}}], 'description': 'My favorite songs.', 'numberOfFollowers': 5}]
```

Delete playlist "Chill Vibes":

Playlist not found