

Introduction à DynamoDB

Description du jeu de données :

Chaque élément de données représente une chanson et contient plusieurs attributs, dont :

- `artistName` : le nom de l'artiste de la chanson.
- `trackName` : le titre de la chanson.
- `collectionName` : le nom de l'album auquel appartient la chanson.
- `releaseDate` : la date de sortie de la chanson.
- `collectionPrice` : le prix de l'album.
- `trackPrice` : le prix de la chanson.
- `primaryGenreName` : le genre principal de la chanson.
- `artworkUrl30`, `artworkUrl60`, `artworkUrl100` : les URL des illustrations de l'album en différentes tailles.
- `previewUrl` : l'URL de l'extrait audio de la chanson.
- `trackTimeMillis` : la durée de la chanson en millisecondes.
- `country` : le pays de l'artiste.
- `currency` : la devise dans laquelle les prix sont indiqués.
- `isStreamable` : un booléen indiquant si la chanson peut être diffusée en streaming

Métadonnées de la base créée :

- `wrapperType` : le type d'élément, expl "track".
- `kind` : le type de média, expl "song".
- `artistId` : l'identifiant de l'artiste.
- `collectionId` : l'identifiant de l'album.
- `trackId` : l'identifiant de la chanson.
- `artistName` : le nom de l'artiste, expl "Depeche Mode".
- `collectionName` : le nom de l'album, expl "The Best of Depeche Mode, Vol. 1 (Deluxe Version)".
- `trackName` : le titre de la chanson, expl "Shake the Disease".
- `collectionCensoredName` : le nom censuré de l'album.
- `trackCensoredName` : le titre censuré de la chanson.
- `artistViewUrl` : l'URL de la page de l'artiste sur iTunes.
- `collectionViewUrl` : l'URL de la page de l'album sur iTunes.
- `trackViewUrl` : l'URL de la page de la chanson sur iTunes.
- `previewUrl` : l'URL de l'extrait audio de la chanson.
- `artworkUrl30`, `artworkUrl60`, `artworkUrl100` : les URL des illustrations de l'album en différentes tailles.
- `collectionPrice` : le prix de l'album.
- `trackPrice` : le prix de la chanson.
- `releaseDate` : la date de sortie de la chanson.
- `collectionExplicitness` : l'explicité de l'album.
- `trackExplicitness` : l'explicité de la chanson.

- discCount : le nombre de disques dans l'album.
- discNumber : le numéro du disque contenant la chanson.
- trackCount : le nombre de chansons dans l'album.
- trackNumber : le numéro de la chanson dans l'album.
- trackTimeMillis : la durée de la chanson en millisecondes.
- country : le pays de l'artiste.
- currency : la devise dans laquelle les prix sont indiqués.
- primaryGenreName : le genre principal de la chanson.
- isStreamable : un booléen indiquant si la chanson peut être diffusée en streaming

Nom, région où elle a été stockée etc :

Les métadonnées de la base de données créée à partir de l'API de recherche iTunes d'Apple sont les suivantes :

- Nom de la table : iTunes
- Région : us-west-2
- Endpoint : http://localhost:8000
- Clé de partition : Artist (type de données : chaîne)
- Clé de tri : SongTitle (type de données : chaîne)
- Autres attributs :
 - Album (type de données : chaîne)
 - Year (type de données : chaîne)
- Capacité provisionnée :
 - ReadCapacityUnits : 5
 - WriteCapacityUnits : 5
- Index secondaire global :
 - IndexName : Album-index
 - KeySchema : Album (type de données : chaîne)

La table :

```
table = dynamodb.create_table(
    TableName='iTunes',
    KeySchema=[
        {
            'AttributeName': 'Artist',
            'KeyType': 'HASH'
        },
        {
            'AttributeName': 'SongTitle',
            'KeyType': 'RANGE'
        }
    ],
    AttributeDefinitions=[
        {
            'AttributeName': 'Artist',
            'AttributeType': 'S'
        },
        {
            'AttributeName': 'SongTitle',
            'AttributeType': 'S'
        },
        {
            'AttributeName': 'Album',
            'AttributeType': 'S'
        }
    ],
    GlobalSecondaryIndexes=[
        {
            'IndexName': 'Album-index',
            'KeySchema': [
                {
                    'AttributeName': 'Album',
                    'KeyType': 'HASH'
                }
            ],
            'Projection': {
                'ProjectionType': 'ALL'
            },
            'ProvisionedThroughput': {
                'ReadCapacityUnits': 5,
                'WriteCapacityUnits': 5
            }
        }
    ],
    ProvisionedThroughput={
        'ReadCapacityUnits': 5,
        'WriteCapacityUnits': 5
    }
)
```

Les requêtes :

Obtenir toutes les chansons d'un artiste spécifique :

```
response = table.query(
    KeyConditionExpression=Key('Artist').eq('Coldplay')
)
items = response['Items']
print(items)
```

```
[{'Artist': 'Coldplay', 'Album': 'A Head Full of Dreams', 'Year': '2015', 'SongTitle': 'A Head Full of Dreams'}, {'Artist': 'Coldplay', 'Album': 'Ghost Stories', 'Year': '2014', 'SongTitle': 'A Sky Full of Stars'}, {'Artist': 'Coldplay', 'Album': 'A Head Full of Dreams', 'Year': '2015', 'SongTitle': 'Adventure of a Lifetime'}, {'Artist': 'Coldplay', 'Album': 'Ghost Stories', 'Year': '2014', 'SongTitle': 'Always In My Head'}, {'Artist': 'Coldplay', 'Album': 'Ghost Stories', 'Year': '2014', 'SongTitle': 'A Nother's Arms'}, {'Artist': 'Coldplay', 'Album': 'A Head Full of Dreams', 'Year': '2015', 'SongTitle': 'Everglow'}, {'Artist': 'Coldplay', 'Album': 'A Head Full of Dreams', 'Year': '2015', 'SongTitle': 'Hymn for the Weekend'}, {'Artist': 'Coldplay', 'Album': 'Ghost Stories', 'Year': '2014', 'SongTitle': 'Ink'}, {'Artist': 'Coldplay', 'Album': 'Ghost Stories', 'Year': '2014', 'SongTitle': 'Magic'}, {'Artist': 'Coldplay', 'Album': 'Ghost Stories', 'Year': '2014', 'SongTitle': 'Midnight'}, {'Artist': 'Coldplay', 'Album': 'Ghost Stories', 'Year': '2014', 'SongTitle': 'O'}, {'Artist': 'Coldplay', 'Album': 'Ghost Stories', 'Year': '2014', 'SongTitle': 'Oceans'}, {'Artist': 'Coldplay', 'Album': 'Mylo Xyloto', 'Year': '2011', 'SongTitle': 'Paradise'}, {'Artist': 'Coldplay', 'Album': 'Ghost Stories', 'Year': '2014', 'SongTitle': 'True Love'}, {'Artist': 'Coldplay', 'Album': 'A Head Full of Dreams', 'Year': '2015', 'SongTitle': 'Up&Up'}]
```

Obtenir toutes les chansons d'un album spécifique :

```
response = table.query(
    IndexName='Album-index',
    KeyConditionExpression=Key('Album').eq('The Best of Depeche Mode, Vol. 1 (Deluxe Version)')
)
items = response['Items']
print(items)
```

```
[{'Artist': 'Depeche Mode', 'Album': 'The Best of Depeche Mode, Vol. 1 (Deluxe Version)', 'Year': '1993', 'SongTitle': 'I Feel You'}, {'Artist': 'Depeche Mode', 'Album': 'The Best of Depeche Mode, Vol. 1 (Deluxe Version)', 'Year': '1984', 'SongTitle': 'Mister and Servant'}, {'Artist': 'Depeche Mode', 'Album': 'The Best of Depeche Mode, Vol. 1 (Deluxe Version)', 'Year': '1997', 'SongTitle': 'It's No Good'}, {'Artist': 'Depeche Mode', 'Album': 'The Best of Depeche Mode, Vol. 1 (Deluxe Version)', 'Year': '1981', 'SongTitle': 'New Life'}, {'Artist': 'Depeche Mode', 'Album': 'The Best of Depeche Mode, Vol. 1 (Deluxe Version)', 'Year': '1985', 'SongTitle': 'Shake the Disease'}, {'Artist': 'Depeche Mode', 'Album': 'The Best of Depeche Mode, Vol. 1 (Deluxe Version)', 'Year': '2005', 'SongTitle': 'Precious'}, {'Artist': 'Depeche Mode', 'Album': 'The Best of Depeche Mode, Vol. 1 (Deluxe Version)', 'Year': '2006', 'SongTitle': 'Martyr'}, {'Artist': 'Depeche Mode', 'Album': 'The Best of Depeche Mode, Vol. 1 (Deluxe Version)', 'Year': '1999', 'SongTitle': 'Dream On'}, {'Artist': 'Depeche Mode', 'Album': 'The Best of Depeche Mode, Vol. 1 (Deluxe Version)', 'Year': '1993', 'SongTitle': 'Walking In My Shoes'}, {'Artist': 'Depeche Mode', 'Album': 'The Best of Depeche Mode, Vol. 1 (Deluxe Version)', 'Year': '1983', 'SongTitle': 'Everything Counts'}]
```

Obtenir toutes les chansons d'un artiste spécifique sorties après une certaine date :

```
from boto3.dynamodb.conditions import Key

response = table.query(
    KeyConditionExpression=Key('Artist').eq('Dido')
)
filtered_items = [item for item in response['Items'] if 'Year' in item and item['Year'] > '1987']
print(filtered_items)
```

```
[{'Artist': 'Dido', 'Album': 'Love Actually Soundtrack', 'Year': '1999', 'SongTitle': 'Here With Me'}, {'Artist': 'Dido', 'Album': 'No Angel', 'Year': '1999', 'SongTitle': 'Thank You'}, {'Artist': 'Dido', 'Album': 'Life for Rent', 'Year': '1999', 'SongTitle': 'White Flag'}]
```

Supprimer une chanson spécifique d'un artiste :

```
response = table.delete_item(
    Key={
        'Artist': 'Depeche Mode',
        'SongTitle': 'Enjoy the Silence'
    }
)
```

```
response = table.get_item(
    Key={
        'Artist': 'Depeche Mode',
        'SongTitle': 'Enjoy the Silence'
    }
)

if 'Item' in response:
    print("L'élément existe toujours :", response['Item'])
else:
    print("L'élément a été supprimé avec succès.")
```

L'élément a été supprimé avec succès.

Mettre à jour le prix d'une chanson spécifique :

```
from decimal import Decimal

response = table.update_item(
    Key={
        'Artist': 'Dido',
        'SongTitle': 'Thank You'
    },
    UpdateExpression="set Price = :p",
    ExpressionAttributeValues={
        ':p': Decimal('0.99'), # Convertir Le float en Decimal
    },
    ReturnValues="UPDATED_NEW"
)
```

```
response = table.get_item(
    Key={
        'Artist': 'Dido',
        'SongTitle': 'Thank You'
    }
)

if 'Item' in response:
    updated_item = response['Item']
    print("L'élément mis à jour :", updated_item)
else:
    print("L'élément n'a pas été trouvé.")
```

L'élément mis à jour : {'Artist': 'Dido', 'Album': 'No Angel', 'Year': '1999', 'Price': Decimal('0.99'), 'SongTitle': 'Thank You'}