

# Amazon DynamoDB

- Amazon DynamoDB es un servicio de base de datos NoSQL totalmente administrado que ofrece un rendimiento rápido y predecible, así como una perfecta escalabilidad.
- Muy sencillo de utilizar usando python.
- [https://docs.aws.amazon.com/es\\_es/dynamodb/index.html](https://docs.aws.amazon.com/es_es/dynamodb/index.html)

- En DynamoDB se trabaja principalmente con tablas, elementos y atributos.
- Una tabla es una colección de ITEMS, y cada elemento es una colección de atributos.
- DynamoDB utiliza claves principales para identificar de forma exclusiva cada uno de los elementos de la tabla e índices secundarios para proporcionar mayor flexibilidad a la hora de realizar consultas.

- Tablas: Al igual que otros sistemas de bases de datos, DynamoDB almacena datos en tablas. Una tabla es una colección de datos.
- Elementos: cada tabla contiene cero o más elementos. Un elemento es un grupo de atributos que puede identificarse de forma exclusiva entre todos los demás elementos.
- Atributos: cada elemento se compone de uno o varios atributos. Un atributo es un componente fundamental de los datos

## People

```
{  
  "PersonID": 101,  
  "LastName": "Smith",  
  "FirstName": "Fred",  
  "Phone": "555-4321"  
}
```

```
{  
  "PersonID": 102,  
  "LastName": "Jones",  
  "FirstName": "Mary",  
  "Address": {  
    "Street": "123 Main",  
    "City": "Anytown",  
    "State": "OH",  
    "ZIPCode": 12345  
  }  
}
```

```
{  
  "PersonID": 103,  
  "LastName": "Stephens",  
  "FirstName": "Howard",  
  "Address": {  
    "Street": "123 Main",  
    "City": "London",  
    "PostalCode": "ER3 5K8"  
  },  
  "FavoriteColor": "Blue"  
}
```

- Cada elemento de la tabla tiene un identificador único, o clave principal, que lo distingue de todos los demás.
- Dejando a un lado la clave principal, la tabla no tiene esquema.
- La clave principal puede constar de dos atributos o más. Cada elemento de la tabla debe tener estos atributos. La combinación de atributos distingue a cada elemento de la tabla de todos los demás.

# Clave principal

- Al crear una tabla, además de asignarle un nombre, debe especificar su clave principal. La clave principal identifica de forma única a cada elemento de la tabla, de manera que no puede haber dos elementos con la misma clave.
- DynamoDB admite dos tipos distintos de clave principal:
  - Clave de partición: una clave principal simple que consta de un solo atributo denominado clave de partición. DynamoDB utiliza el valor de clave de partición como información de entrada a una función hash interna.
  - Clave de partición y clave de ordenación: Se denomina clave principal compuesta, este tipo de clave se compone de dos atributos. El primer atributo es la clave de partición y el segundo, la clave de ordenación.

- Todos los elementos con el mismo valor de clave de partición se almacenan en posiciones contiguas, ordenados según el valor de la clave de ordenación.



# Creación de una tabla

```
import boto3

dynamodb = boto3.resource('dynamodb')

table = dynamodb.create_table(
    TableName='Movies',
    KeySchema=[
        {
            'AttributeName': 'year',
            'KeyType': 'HASH' # Partition key
        },
        {
            'AttributeName': 'title',
            'KeyType': 'RANGE' # Sort key
        }
    ],
    AttributeDefinitions=[
        {
            'AttributeName': 'year',
            'AttributeType': 'N'
        },
        {
            'AttributeName': 'title',
            'AttributeType': 'S'
        }
    ],
    ProvisionedThroughput={
        'ReadCapacityUnits': 10,
        'WriteCapacityUnits': 10
    }
)
```

# Insertar

```
import boto3

dynamodb = boto3.resource('dynamodb')

table = dynamodb.Table('Movies')

response = table.put_item(
    Item={
        'year': 2015,
        'title': "The Big New Movie",
        'info': {
            'plot': "Nothing happens at all.",
            'rating': 0
        }
    }
)
```

## Lectura de un elemento

```
import boto3
from botocore.exceptions import ClientError

dynamodb = boto3.resource('dynamodb')
table = dynamodb.Table('Movies')
response = table.get_item(Key={'year': 2015, 'title': "The Big New Movie"})
print(response['Item'])
```

# Query

```
import boto3
from boto3.dynamodb.conditions import Key

table = dynamodb.Table('Movies')
response = table.query(
    KeyConditionExpression=Key('year').eq(year)
)
print(response['Items'])
```

# Ejercicio

- Crea una tabla que tenga el siguiente esquema:

```
KeySchema=[  
  {  
    'AttributeName': 'VALOR',  
    'KeyType': 'HASH'    # Partition key  
  },  
  {  
    'AttributeName': 'TIME',  
    'KeyType': 'RANGE'    # Sort key  
  }  
]
```

- Usando el fichero: market\_data\_proc.csv inserta los elementos en la tabla.
- Lee los datos del santander.