



МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»  
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ  
Кафедра Інформаційної Безпеки

## Хмарні технології обробки даних

---

### Лабораторна робота №3

### Використання AWS DynamoDB (Serverless Database)

**Мета:** ознайомитись з використанням AWS DynamoDB (Serverless Database)

Перевірив:

\_\_\_\_\_

Виконав:

студент III курсу

групи ФБ-01

Сахній Н.Р.

Київ 2023

ФБ-01 Сахній Назар

## Завдання:

- Спроекувати структуру даних (таблицю)
- Вивчити способи роботи з даними засобами DynamoDB
- Виконати завдання відповідно до варіанту (в GUI, засобами AWS CLI та Python – останнє за бажанням)

## Хід виконання роботи

1. Для даних, отриманих в лабораторній роботі №2, спроектуємо таблицю DynamoDB за допомогою наступних засобів:

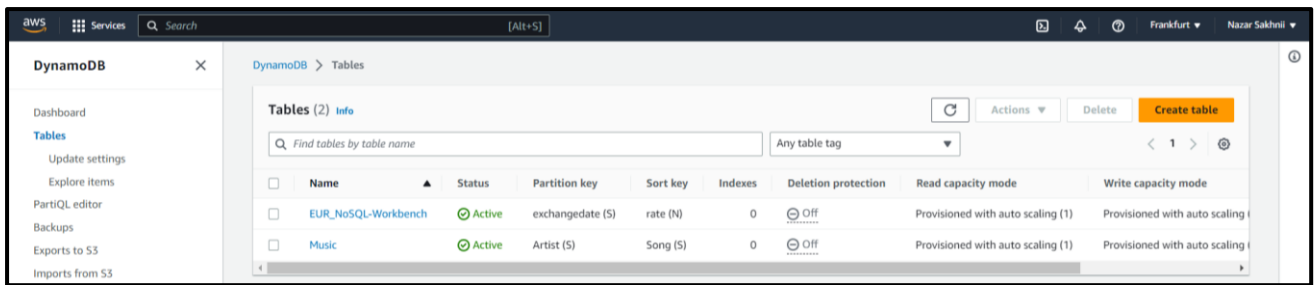
- NoSQL Workbench

The screenshot shows the AWS NoSQL Workbench Visualizer interface. On the left is a sidebar with navigation options: AWS database catalog, Amazon DynamoDB, Data modeler, Visualizer (selected), Operation builder, Documentation, and Share feedback. The main area is titled 'Visualizer' and shows a data model for 'Exchange Rates'. Below the data model, there are buttons for 'Aggregate view' and 'Commit to Amazon DynamoDB'. The 'Aggregate view' is selected, displaying a table with columns: Partition key: exchangedate, Sort key: rate, and Attributes. The table contains 10 rows of data, including dates, rates, and various attributes like currency, text, and group.

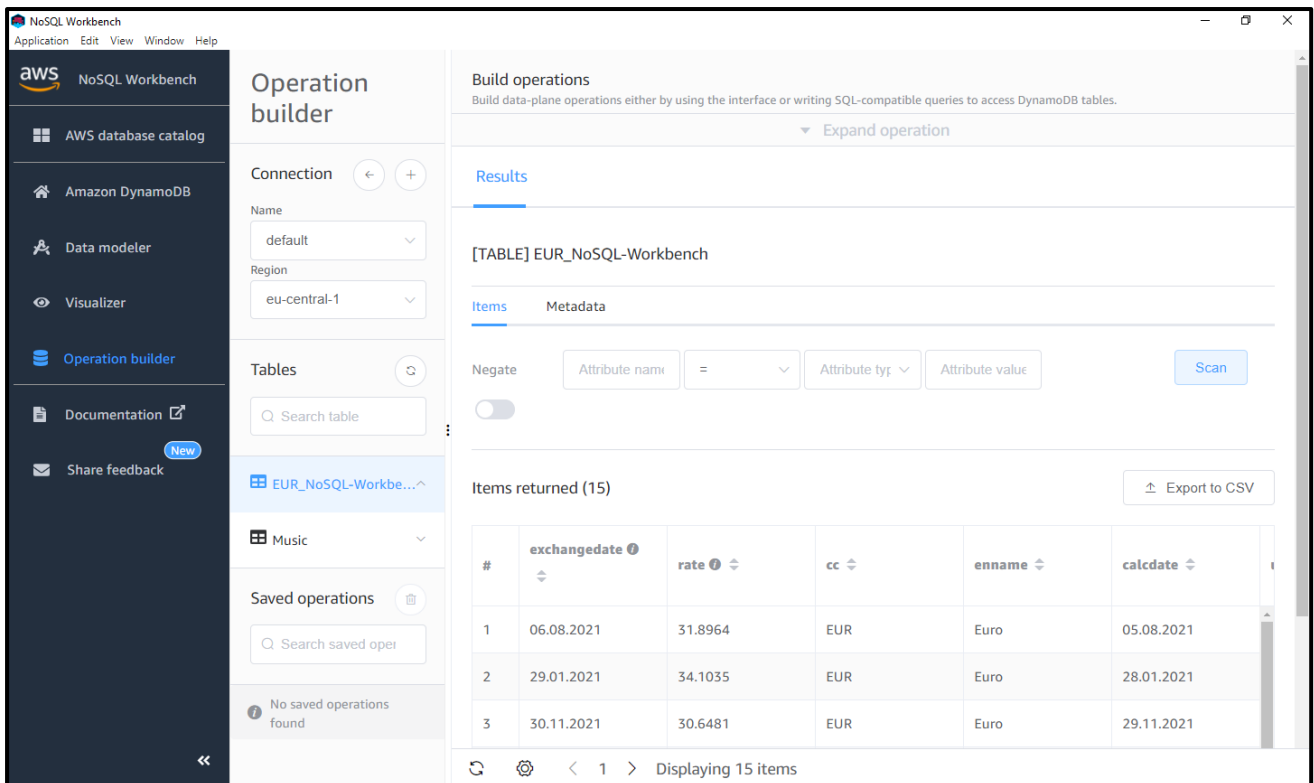
Partition key: exchangedate	Sort key: rate	Attributes							
31.12.2021	30.9226	r030	cc	txt	enname	units	rate_per_unit	group	caldate
		978	EUR	€apo	Euro	1	30.9226	1	30.12.2021
30.12.2021	30.776	r030	cc	txt	enname	units	rate_per_unit	group	caldate
		978	EUR	€apo	Euro	1	30.776	1	29.12.2021
29.12.2021	30.8401	r030	cc	txt	enname	units	rate_per_unit	group	caldate
		978	EUR	€apo	Euro	1	30.8401	1	28.12.2021
28.12.2021	30.8196	r030	cc	txt	enname	units	rate_per_unit	group	caldate
		978	EUR	€apo	Euro	1	30.8196	1	24.12.2021
27.12.2021	30.845	r030	cc	txt	enname	units	rate_per_unit	group	caldate
		978	EUR	€apo	Euro	1	30.845	1	23.12.2021
26.12.2021	30.845	r030	cc	txt	enname	units	rate_per_unit	group	caldate
		978	EUR	€apo	Euro	1	30.845	1	23.12.2021
25.12.2021	30.845	r030	cc	txt	enname	units	rate_per_unit	group	caldate
		978	EUR	€apo	Euro	1	30.845	1	23.12.2021

↑ Отже, щоб синхронізувати (завантажити) новостворену базу даних в онлайн-середовище “AWS DynamoDB”, необхідно натиснути кнопку “Commit to Amazon DynamoDB”, і далі обрати попередньо сконфігуроване відповідне з’єднання ↓

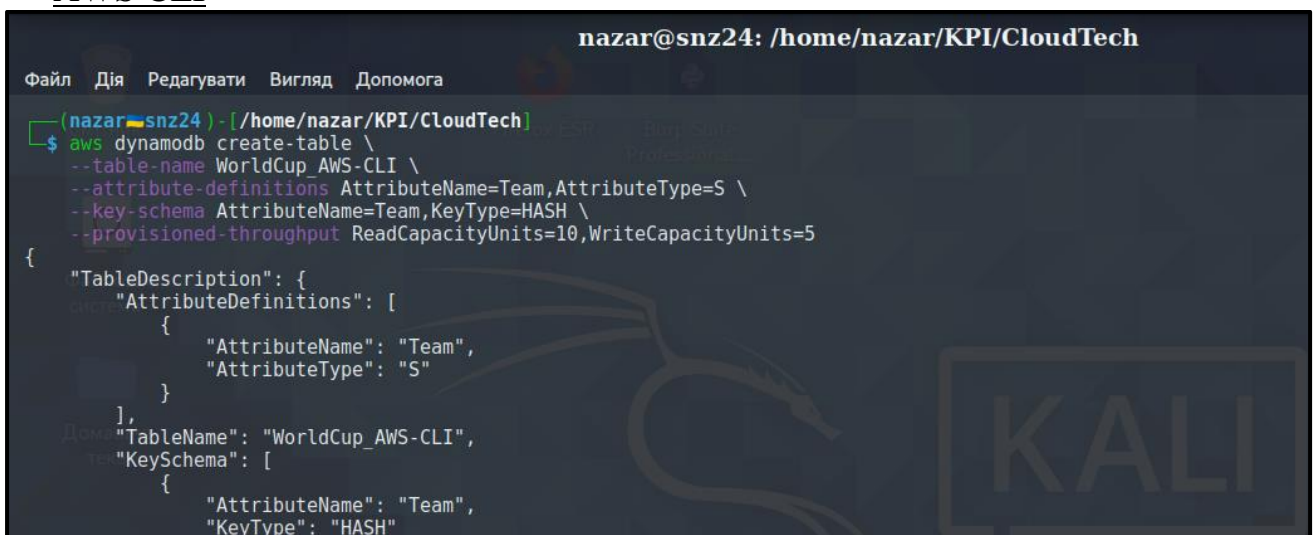
The screenshot shows the 'Commit to Amazon DynamoDB' dialog box. It contains a warning message about auto scaling and table creation. Below the message, there are three tabs: 'Use saved connections' (selected), 'Add a new remote connection', and 'Add a new DynamoDB local connection'. Under the 'Use saved connections' tab, there is a dropdown menu labeled 'Saved connections' with 'default' selected. At the bottom, there are three buttons: 'Cancel', 'Reset', and 'Commit'.



↑ Демонстрація вдалого процесу синхронізації DynamoDB таблиці ↓



## • AWS CLI



```

    },
    "TableStatus": "CREATING",
    "CreationDateTime": "2023-04-12T22:48:57.601000+03:00",
    "ProvisionedThroughput": {
      "NumberOfDecreasesToday": 0,
      "ReadCapacityUnits": 10,
      "WriteCapacityUnits": 5
    },
    "TableSizeBytes": 0,
    "ItemCount": 0,
    "TableArn": "arn:aws:dynamodb:eu-central-1:334248859612:table/WorldCup_AWS-CLI",
    "TableId": "19827ebb-2cf6-491e-9c86-7ce66f5dbeb6",
    "DeletionProtectionEnabled": false
  }
}

```

↑ Створення таблиці бази даних й внесення деякого набору даних із json-файлу ↓

```

(nazar@snz24) - [~/home/nazar/KPI/CloudTech]
$ aws dynamodb batch-write-item \
  --request-items file://football.json
{
  "UnprocessedItems": {}
}

```

The screenshot shows the AWS Management Console interface for a DynamoDB table named 'WorldCup\_AWS-CLI'. The table is in the 'CREATING' state. The 'Scan or query items' section shows the table selected. The 'Items returned (5)' section displays a table with 5 items, each representing a football team with attributes like Team, CurrentCoach, Stadium, WorldCupTrophies, and YearFounded.

Team	CurrentCoach	Stadium	WorldCupTrophies	YearFounded
Spain	Luis Enrique	Santiago Be...	1	1902
Germany	Hans-Dieter Flick	Allianz Arena	4	1900
Brazil	Tite	Maracanã	5	1914
France	Didier Descha...	Stade de Fr...	2	1904
Ukraine	Ruslan Rotan	NSC Olimpi...	0	1991

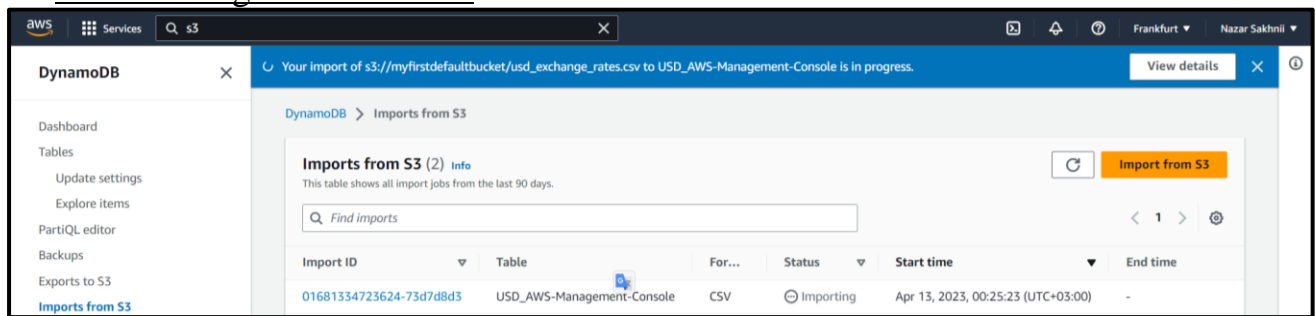
↑ Знов таки, ось зображено результат того, що таблицю було успішно створено ↓

```

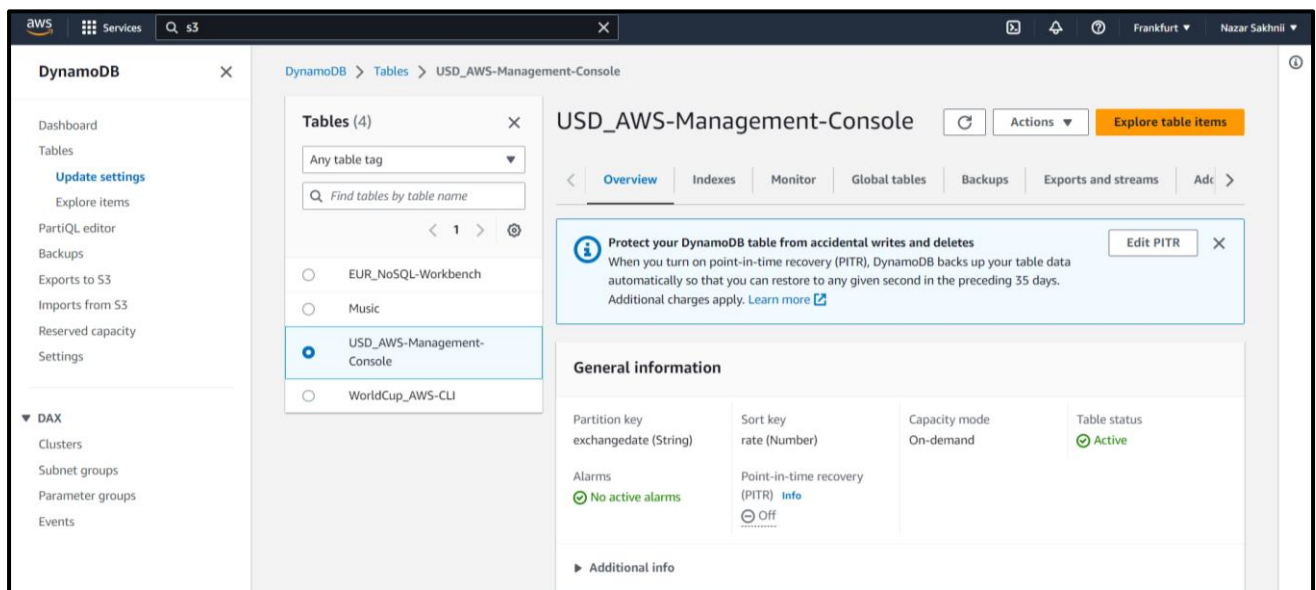
(nazar@snz24) - [~/home/nazar/KPI/CloudTech]
$ aws dynamodb list-tables
{
  "TableNames": [
    "EUR_NoSQL-Workbench",
    "Music",
    "WorldCup_AWS-CLI"
  ]
}

```

- AWS Management Console



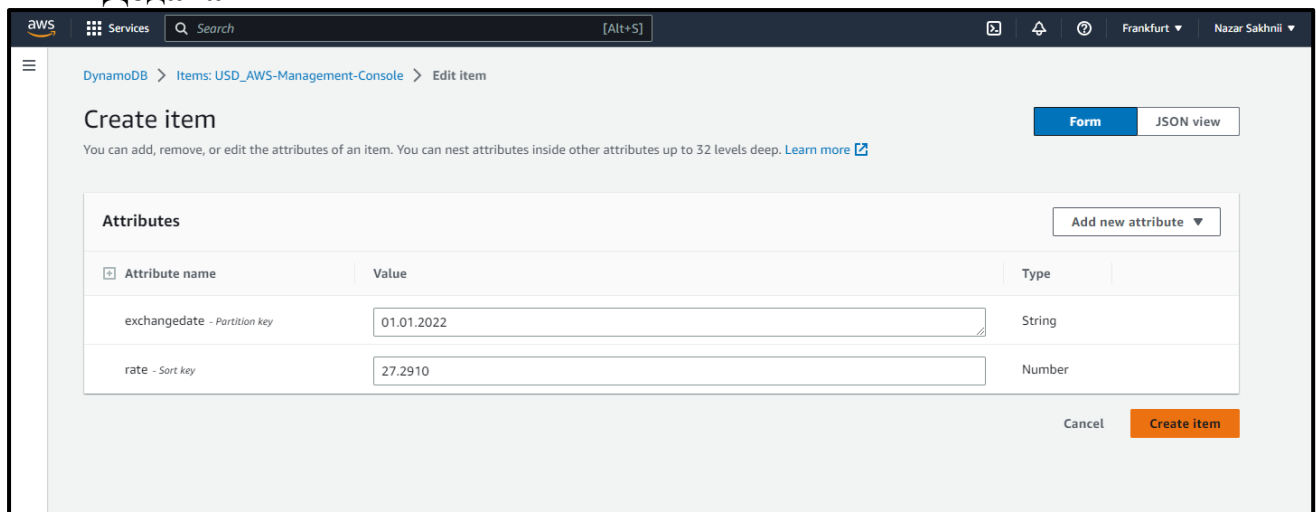
↑Імпортування .csv файлу із контейнера та перегляд інформації про нову таблицю↓

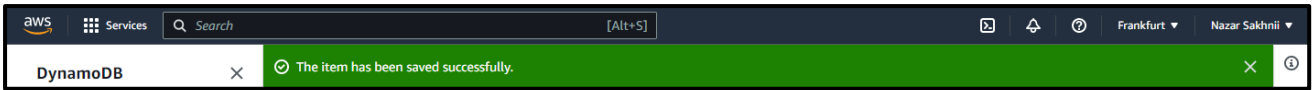


2. Спробуємо відпрацювати додавання, видалення та пошук даних засобами:

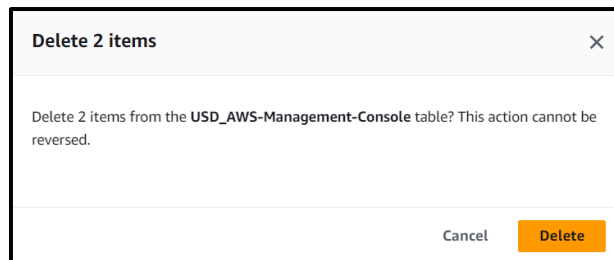
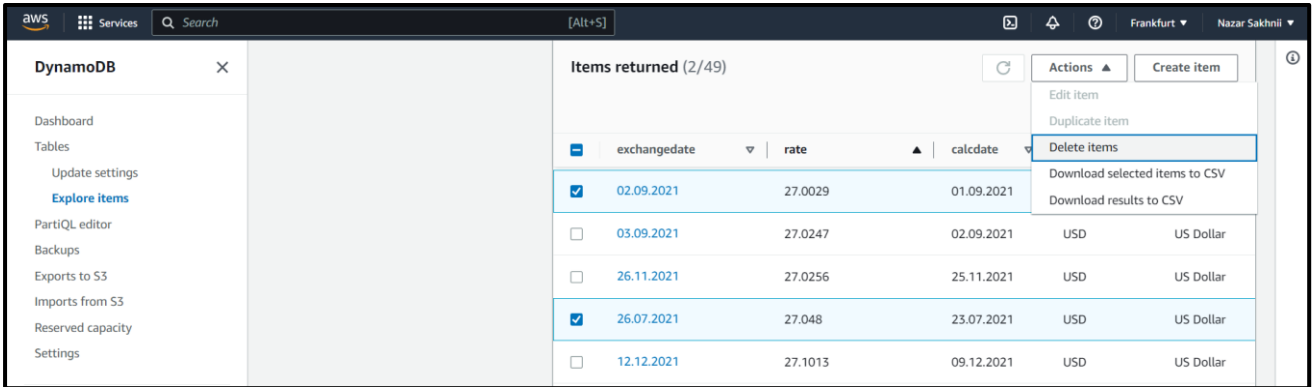
- AWS Management Console

- ❖ Додавання

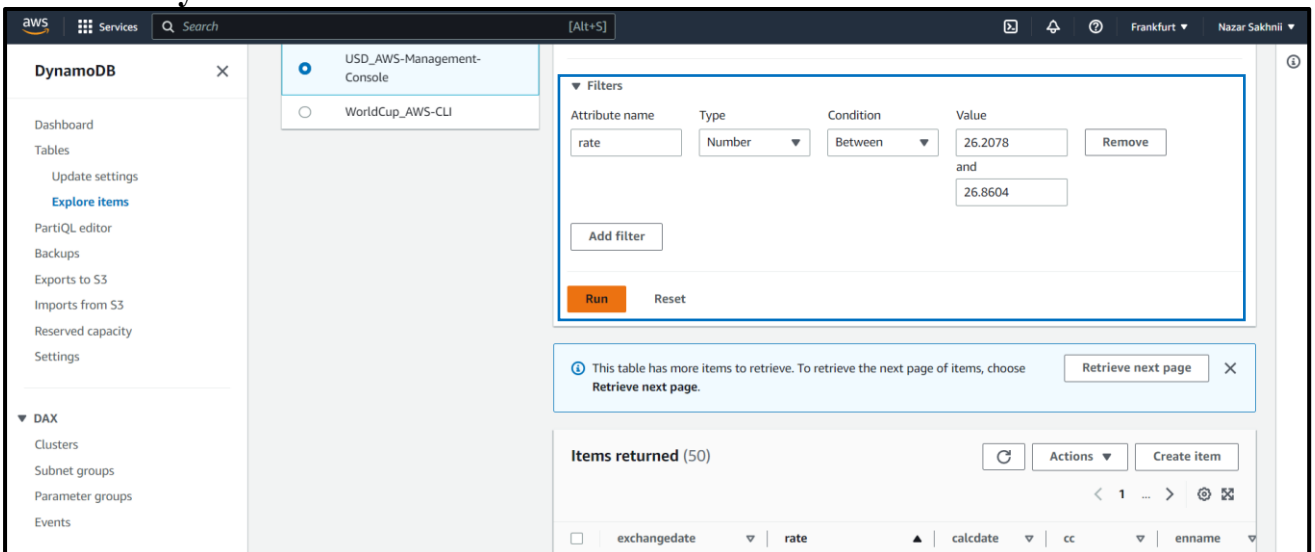




## ❖ Видалення



## ❖ Пошук

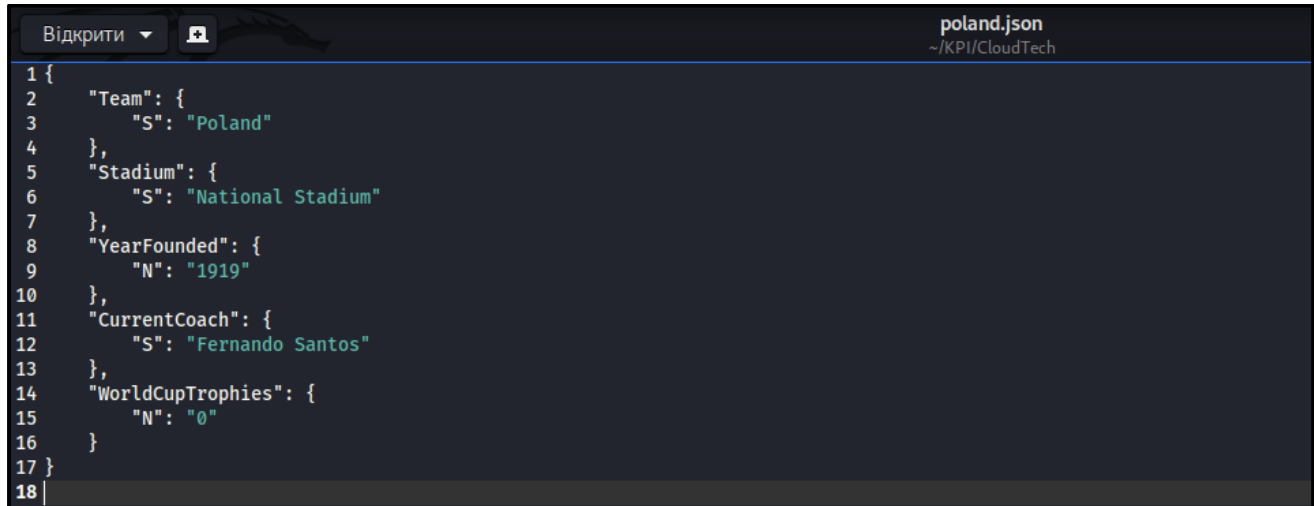


<input type="checkbox"/>	24.10.2021	26.2918	22.10.2021	USD	US Dollar
<input type="checkbox"/>	30.10.2021	26.3265	28.10.2021	USD	US Dollar

- AWS CLI

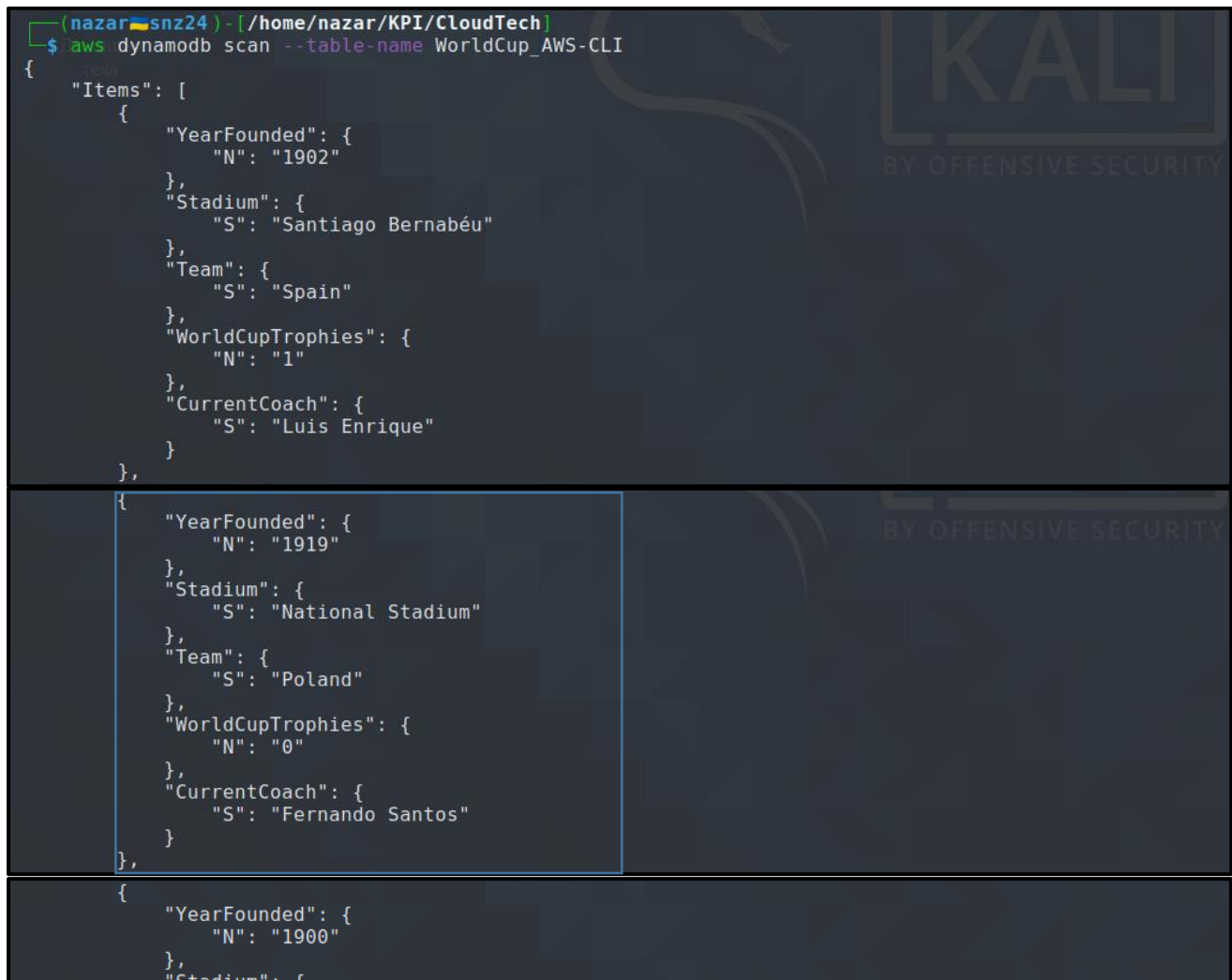
- ❖ Додавання

```
(nazar@snz24) - [/home/nazar/KPI/CloudTech]  
$ aws dynamodb put-item --table-name WorldCup_AWS-CLI --item file:///poland.json
```



The screenshot shows a text editor window titled "poland.json" with the file path "~ / KPI / CloudTech". The editor contains a JSON object representing the Poland national football team. The JSON is formatted with line numbers from 1 to 18 on the left margin.

```
1 {  
2   "Team": {  
3     "S": "Poland"  
4   },  
5   "Stadium": {  
6     "S": "National Stadium"  
7   },  
8   "YearFounded": {  
9     "N": "1919"  
10  },  
11  "CurrentCoach": {  
12    "S": "Fernando Santos"  
13  },  
14  "WorldCupTrophies": {  
15    "N": "0"  
16  }  
17 }  
18 |
```



The screenshot shows the output of the AWS CLI command `aws dynamodb scan --table-name WorldCup_AWS-CLI`. The output is a JSON array of items. The first item is for Spain, and the second item (highlighted with a blue box) is for Poland, matching the content of the `poland.json` file shown in the previous screenshot.

```
(nazar@snz24) - [/home/nazar/KPI/CloudTech]  
$ aws dynamodb scan --table-name WorldCup_AWS-CLI  
{  
  "Items": [  
    {  
      "YearFounded": {  
        "N": "1902"  
      },  
      "Stadium": {  
        "S": "Santiago Bernabéu"  
      },  
      "Team": {  
        "S": "Spain"  
      },  
      "WorldCupTrophies": {  
        "N": "1"  
      },  
      "CurrentCoach": {  
        "S": "Luis Enrique"  
      }  
    },  
    {  
      "YearFounded": {  
        "N": "1919"  
      },  
      "Stadium": {  
        "S": "National Stadium"  
      },  
      "Team": {  
        "S": "Poland"  
      },  
      "WorldCupTrophies": {  
        "N": "0"  
      },  
      "CurrentCoach": {  
        "S": "Fernando Santos"  
      }  
    },  
    {  
      "YearFounded": {  
        "N": "1900"  
      },  
      "Stadium": {  
        "S": "Santiago Bernabéu"  
      },  
      "Team": {  
        "S": "Spain"  
      },  
      "WorldCupTrophies": {  
        "N": "1"  
      },  
      "CurrentCoach": {  
        "S": "Luis Enrique"  
      }  
    }  
  ]  
}
```



## ❖ Видалення

```
(nazar@snz24) - [/home/nazar/KPI/CloudTech]
$ aws dynamodb delete-item --table-name WorldCup_AWS-CLI --key '{"Team": {"S": "Spain"}}'
```

```
(nazar@snz24) - [/home/nazar/KPI/CloudTech]
$ aws dynamodb scan --table-name WorldCup_AWS-CLI
{
  "Items": [
    {
      "YearFounded": {
        "N": "1919"
      },
      "Stadium": {
        "S": "National Stadium"
      },
      "Team": {
        "S": "Poland"
      },
      "WorldCupTrophies": {
        "N": "0"
      },
      "CurrentCoach": {
        "S": "Fernando Santos"
      }
    },
    {
      "YearFounded": {
        "N": "1900"
      },
      "Stadium": {

```

## ❖ Пошук

```
(nazar@snz24) - [/home/nazar/KPI/CloudTech]
$ aws dynamodb scan \
  --table-name WorldCup_AWS-CLI \
  --filter-expression "WorldCupTrophies >= :val" \
  --expression-attribute-values '{":val":{"N":"5"}}'
{
  "Items": [
    {
      "YearFounded": {
        "N": "1914"
      },
      "Stadium": {
        "S": "Maracanã"
      },
      "Team": {
        "S": "Brazil"
      },
      "WorldCupTrophies": {
        "N": "5"
      },
      "CurrentCoach": {
        "S": "Tite"
      }
    }
  ],
  "Count": 1,
  "ScannedCount": 5,
  "ConsumedCapacity": null
}
```

```
(nazar@snz24) - [/home/nazar/KPI/CloudTech]
$ aws dynamodb scan \
  --table-name WorldCup_AWS-CLI \
  --filter-expression "Stadium = :val" \
  --expression-attribute-values '{":val":{"S":"Wembley"}}'
{
  "Items": [],
  "Count": 0,
  "ScannedCount": 5,
  "ConsumedCapacity": null
}
```



## • Python SDK \*

```

jupyter Python SDK and DynamoDB Last Checkpoint: 32 minutes ago (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

In [1]: import boto3

dynamodb = boto3.resource("dynamodb")
table = dynamodb.Table("EUR_NoSQL-Workbench")

In [2]: print(table.key_schema)

[{'AttributeName': 'exchangedate', 'KeyType': 'HASH'}, {'AttributeName': 'rate', 'KeyType': 'RANGE'}]

```

## ❖ Додавання

```

In [3]: from decimal import Decimal

new_record = {
    "exchangedate": "10.04.2023",
    "r030": 840,
    "cc": "EUR",
    "txt": "Євро",
    "enname": "Euro",
    "rate": Decimal("40.568"),
    "units": 1,
    "rate_per_unit": Decimal("40.568"),
    "group": "2",
    "calcdte": "09.04.2023",
}

table.put_item(Item=new_record)

Out[3]: {'ResponseMetadata': {'RequestId': 'MUBMM36C14QFNVA559ATSH568FVV4KQNS05AEMVJF66Q9ASUAAJG',
    'HTTPStatusCode': 200,
    'HTTPHeaders': {'server': 'Server',
    'date': 'Thu, 13 Apr 2023 18:23:12 GMT',
    'content-type': 'application/x-amz-json-1.0',
    'content-length': '2',
    'connection': 'keep-alive',
    'x-amzn-requestid': 'MUBMM36C14QFNVA559ATSH568FVV4KQNS05AEMVJF66Q9ASUAAJG',
    'x-amz-crc32': '2745614147'},
    'RetryAttempts': 0}}

```

Attribute name	Type	Condition	Value	
exchangedate	String	Equal to	10.04.2023	Remove

Add filter

Run Reset

Completed. Read capacity units consumed: 5.5

Items returned (1)

	exchangedate	rate	calcdte	cc	enname	group	r030	rate_per_un
<input type="checkbox"/>	10.04.2023	40.568	09.04.2023	EUR	Euro	2	840	40.568

## ❖ Видалення

```

In [4]: table.delete_item(Key={"exchangedate": "21.06.2021", "rate": Decimal("36.626")})

```

```
Out[4]: {'ResponseMetadata': {'RequestId': 'AA1IUN6J0S09QK407G0KF49D1JVV4KQNS05AEMVJF66Q9ASUAAJG',
  'HTTPStatusCode': 200,
  'HTTPHeaders': {'server': 'Server',
    'date': 'Thu, 13 Apr 2023 18:29:36 GMT',
    'content-type': 'application/x-amz-json-1.0',
    'content-length': '2',
    'connection': 'keep-alive',
    'x-amzn-requestid': 'AA1IUN6J0S09QK407G0KF49D1JVV4KQNS05AEMVJF66Q9ASUAAJG',
    'x-amz-crc32': '2745614147'},
  'RetryAttempts': 0}}
```

Attribute name	Type	Condition	Value	
<input type="text" value="exchangedate"/>	<input type="text" value="String"/>	<input type="text" value="Equal to"/>	<input type="text" value="21.06.2021"/>	<input type="button" value="Remove"/>
<input type="button" value="Add filter"/>				
<input type="button" value="Run"/> <input type="button" value="Reset"/>				
Completed. Read capacity units consumed: 5.5				
Items returned (0) <input type="button" value="Refresh"/> <input type="button" value="Actions"/> <input type="button" value="Create item"/>				
<div>&lt; 1 &gt; <input type="button" value="Settings"/> <input type="button" value="Fullscreen"/></div>				
The query did not return any results.				

## ❖ Пошук

```
In [5]: response = table.query(KeyConditionExpression=boto3.dynamodb.conditions.Key("exchangedate").eq("10.04.2023"))
print(response['Items'])
[{'cc': 'EUR', 'rate': Decimal('40.568'), 'exchangedate': '10.04.2023', 'enname': 'Euro', 'calcddate': '09.04.2023',
  'units': Decimal('1'), 'group': '2', 'txt': 'Євро', 'rate_per_unit': Decimal('40.568'), 'r030': Decimal('840')}]
```

Attribute name	Type	Condition	Value												
<input type="text" value="exchangedate"/>	<input type="text" value="String"/>	<input type="text" value="Equal to"/>	<input type="text" value="10.04.2023"/>	<input type="button" value="Remove"/>											
<input type="button" value="Add filter"/>															
<input type="button" value="Run"/> <input type="button" value="Reset"/>															
Completed. Read capacity units consumed: 5.5															
Items returned (1) <input type="button" value="Refresh"/> <input type="button" value="Actions"/> <input type="button" value="Create item"/>															
<div>&lt; 1 &gt; <input type="button" value="Settings"/> <input type="button" value="Fullscreen"/></div>															
<input type="checkbox"/>	exchangedate	▼	rate	▼	calcddate	▼	cc	▼	enname	▼	group	▼	r030	▼	rate_per_un
<input type="checkbox"/>	10.04.2023		40.568		09.04.2023		EUR		Euro		2		840		40.568

## Висновки:

У ході виконання лабораторної роботи з проектування таблиці DynamoDB та виконання операцій додавання, видалення та пошуку даних за допомогою AWS Management Console, AWS CLI та Python SDK виникли деякі невеликі труднощі. Втім, з використанням різноманітних шляхів розв'язку проблем вдалося успішно виконати завдання. Наприклад, при використанні NoSQL Workbench виникли питання з щодо налагодження з'єднання з DynamoDB, але ці питання вдалося вирішити за допомогою прочитання відповідних розділів у документації.

Також при імплементації операцій засобами Python SDK було необхідно здійснити додаткову конфігурацію середовища та правильно використати команди з бібліотеки boto3, але з використанням документації та різноманітних матеріалів з Інтернету вдалося виконати це завдання.