



Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Фізико-технічний інститут

Комп'ютерний практикум №3 Використання AWS DynamoDB

предмет «Хмарні технології обробки даних»

Роботу виконав:

Студент 3 курсу ФТІ, групи ФІ-91
Цибульник Антон Владиславович

Приймали:

Шелестов Андрій Юрійович
Колотій Андрій Всеволодович

Мета

Ознайомитись з використанням AWS DynamoDB (serverless database).

Завдання

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

Хід виконання роботи

1. Робота з таблицею засобами AWS Management Console

Створення таблиці

Для створення таблиці засобами AWS Management Console слід увійти в свій акаунт й віднайти розділ «DynamoDB» самостійно або перейшовши за [посиланням](#).

Далі шукаємо у правій частині екрану помаранчеву кнопку «Create table». Після цього вводимо назву таблиці та назву стовпця, який буде відігравати роль Primary key для рядків таблиці. Інакше кажучи, слід придумати стовпець, елементи якого будуть унікальними для кожного рядка таблиці.

Наприклад, для структури даних студентів групи роль такого стовпця може відігравати номер залікової книги (у кожного студента він свій, а відтак цей номер однозначно характеризуватиме кожного конкретного студента). Колонка Primary key може складатися й з двох частин. Наприклад, у випадку тих же студентів можна ввести складений ключ: назва групи й порядковий номер студента в групі. Така комбінація теж буде унікальною для кожного студента.

У цій лабораторній буде відбуватися робота з даними про курс гривні відносно інших валют. З цими даними ми вже працювали у Лабораторній роботі №2. Таблиця, яку ми намагаємося створити, матиме п'ять колонок:

r030	номер запису
txt	назва валюти українською
rate	поточне значення курсу
cc	міжнародний шифр валюти
exchangedate	поточна дата

Тож на завершення діалогового вікна створення таблиці крім її імені варто зазначити й Primary key. У випадку структури даних курсу валют я обрав складений Primary key: колонку шифра валют cc й поточної дати exchangedate (Рис. 1).

Create table

Table details [Info](#)

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name

This will be used to identify your table.

Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.).

Partition key

The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.

String ▼

1 to 255 characters and case sensitive.

Sort key - optional

You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.

String ▼

1 to 255 characters and case sensitive.

Рис. 1: Створення таблиці засобами AWS Management Console

Додавання рядків до таблиці

Поступово додамо 5 рядків до новоствореної таблиці «Currency_UAH». Для цього спершу натиснемо на ім'я таблиці в каталозі усіх таблиць DynamoDB. Потім оберемо помаранчеву кнопку «Explore table items» (Рис. 2). І наостанок знайдемо «Create item». Введемо значення для вже існуючих двох колонок нашого складеного Primary key, а решта інформації додамо інструментом «Add new attribute» (Рис. 3). Почергово додавши 5 рядків, отримаємо фінальну заповнену структуру даних (як це зображено на Рис. 4).

Отже, алгоритм додавання рядків до новоствореної таблиці виглядатиме так:

- «Currency_UAH»
- Explore table items
 - Create item
 - Fill the data for a Primary key
 - Add new attribute
 - Fill the data for an additional attribute

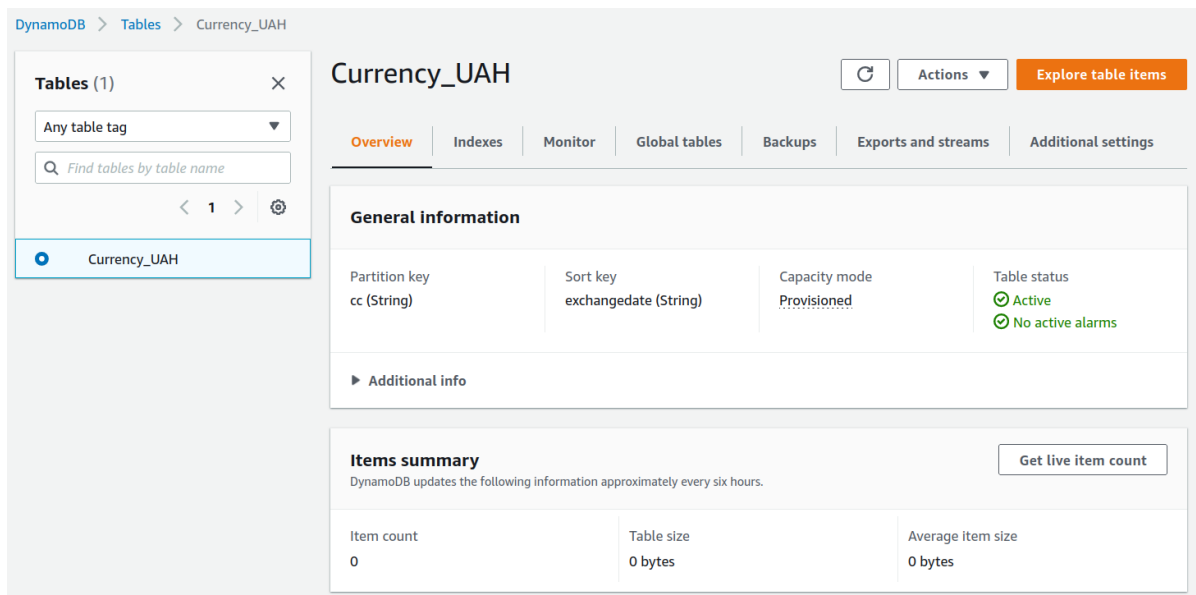


Рис. 2: Інструменти новоствореної таблиці

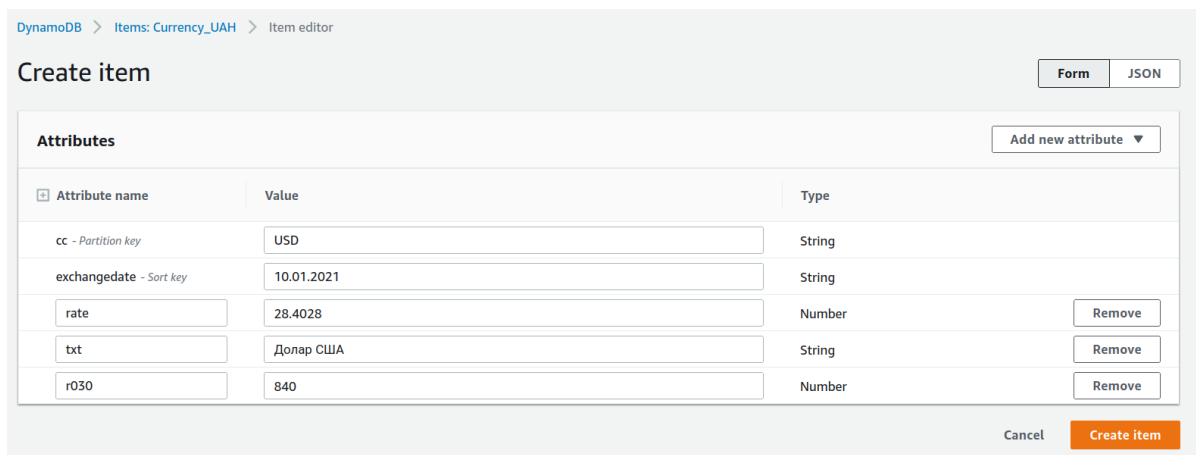


Рис. 3: Додавання рядка до таблиці

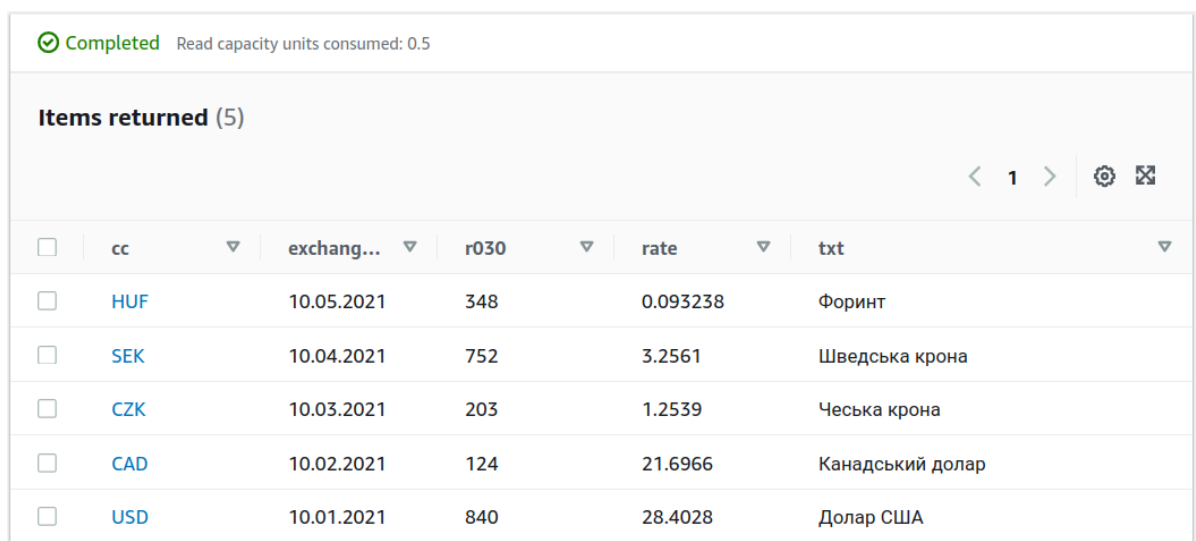


Рис. 4: Наповнена даними таблиця

Пошук елементів таблиці

Зайшовши у вкладку «Explore table items» в категорію «Scan/Query items» можна здійснити пошук елементів в таблиці. Послідовно виконаємо три запити, які зображено на Рис. 5-7.

▼ Scan/Query items

ScanQuery

Table or index

Currency_UAH

▼ Filters

Attribute name	Type	Condition	Value	
r030	Number	Greater than	100	Remove
r030	Number	Less than	500	Remove
Add filter				

Run

Reset

✔ Completed Read capacity units consumed: 0.5

Items returned (3)

< 1 > ⚙️

<input type="checkbox"/>	cc	exchang...	r030	rate	txt
<input type="checkbox"/>	HUF	10.05.2021	348	0.093238	Форинт
<input type="checkbox"/>	CZK	10.03.2021	203	1.2539	Чеська крона
<input type="checkbox"/>	CAD	10.02.2021	124	21.6966	Канадський долар

Рис. 5: Перший запит

▼ Scan/Query items

ScanQuery

Table or index

Currency_UAH

▼ Filters

Attribute name	Type	Condition	Value	
r030	Number	Equal to	752	Remove
Add filter				

Run

Reset

✔ Completed Read capacity units consumed: 0.5

Items returned (1)

< 1 > ⚙️ ✖️

<input type="checkbox"/>	cc	exchang...	r030	rate	txt
<input type="checkbox"/>	SEK	10.04.2021	752	3.2561	Шведська крона

Рис. 6: Другий запит

▼ Scan/Query items

Scan Query

Table or index
Currency_UAH

▼ Filters

Attribute name	Type	Condition	Value	
rate	Number	Greater than...	3	Remove

Add filter

Run Reset

✔ Completed Read capacity units consumed: 0.5

Items returned (3)

< 1 > ⚙️ ✖️

<input type="checkbox"/>	cc	exchang...	r030	rate	txt
<input type="checkbox"/>	USD	10.01.2021	840	28.4028	Долар США
<input type="checkbox"/>	SEK	10.04.2021	752	3.2561	Шведська крона
<input type="checkbox"/>	CAD	10.02.2021	124	21.6966	Канадський долар

Рис. 7: Третій запит

Видалення рядків таблиці

Видалення рядків можна легко виконати, зробивши декілька кроків. Перш за все слід бути у розділі «Explore table items». Навпроти рядка, який бажаємо видалити, слід поставити відмітку. Далі обираємо категорію «Actions», де шукаємо варіант «Delete items». Остаточного підтверджуємо операцію. Усі кроки показано на Рис. 8.

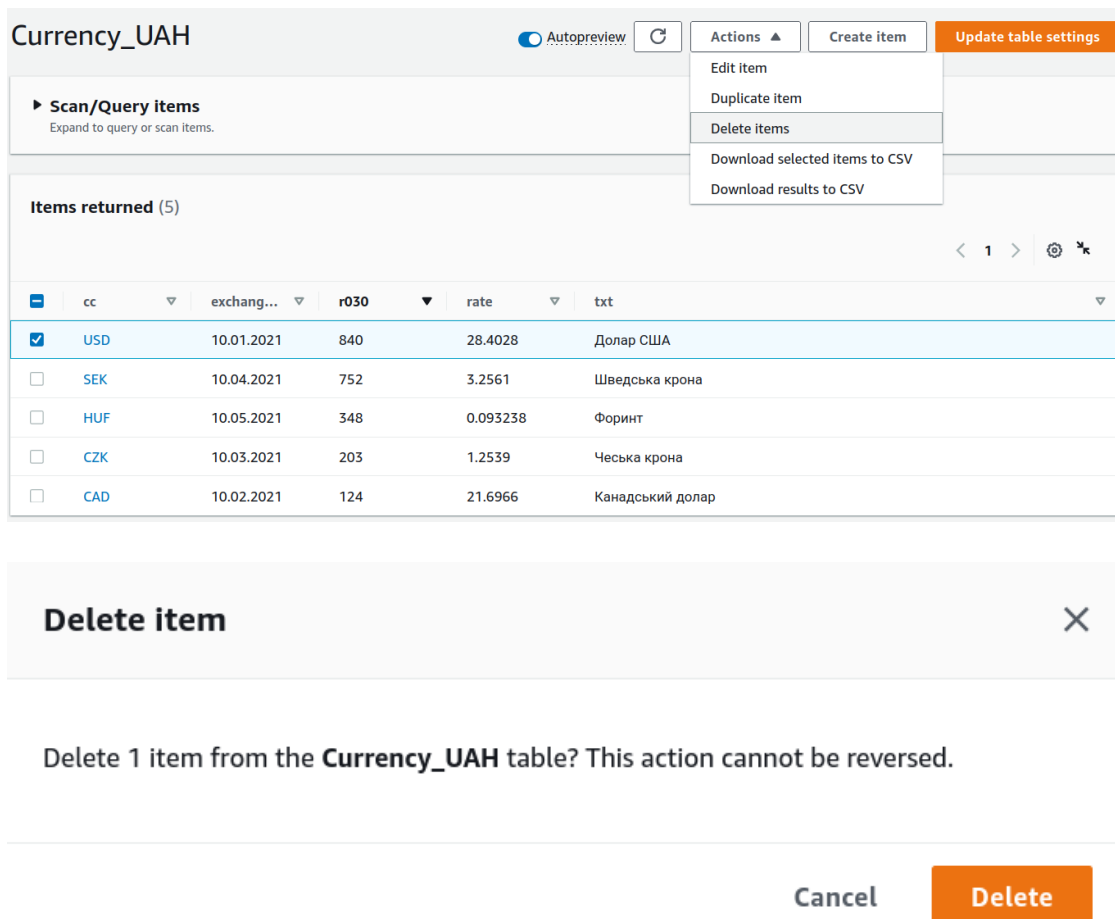


Рис. 8: Видалення елементу з таблиці

2. Робота з таблицею засобами AWS CLI

Створення таблиці

Створимо ще одну таблицю, проте цього разу скористаємося засобами AWS Command Line Interface (AWS CLI). Для цього спершу підключимося з консолі до інстансу (як це робилося у Лабораторній роботі №1). Надалі введемо таку команду:

```

1  aws dynamodb create-table \
2  --table-name Currency_UAH_aws \
3  --attribute-definitions \
4      AttributeName=cc,AttributeType=S \
5      AttributeName=exchangedate,AttributeType=S \
6  --key-schema \
7      AttributeName=cc,KeyType=HASH \
8      AttributeName=exchangedate,KeyType=RANGE \
9  --provisioned-throughput \
10     ReadCapacityUnits=10,WriteCapacityUnits=5

```

Рядком 2 оголосимо ім'я таблиці, рядками 3-5, 6-8 створимо дві колонки й оголосимо їх складеним Primary key (складові HASH та RANGE відповідно). Останнім атрибутом `--provisioned-throughput` вказуємо кількість зчитувань/записів за секунду для нашої таблиці. Детальніше про параметри цього атрибуту за [посиланням](#).

```
ubuntu@ip-172-31-81-119: ~  
ubuntu@ip-172-31-81-119:~$ aws dynamodb create-table \  
> --table-name Currency_UAH_aws \  
> --attribute-definitions \  
>   AttributeName=cc,AttributeType=S \  
>   AttributeName=exchangedate,AttributeType=S \  
> --key-schema \  
>   AttributeName=cc,KeyType=HASH \  
>   AttributeName=exchangedate,KeyType=RANGE \  
> --provisioned-throughput \  
>   ReadCapacityUnits=10,WriteCapacityUnits=5  
{  
  "TableDescription": {  
    "AttributeDefinitions": [  
      {  
        "AttributeName": "cc",  
        "AttributeType": "S"  
      },  
      {  
        "AttributeName": "exchangedate",  
        "AttributeType": "S"  
      }  
    ],  
    "TableName": "Currency_UAH_aws",  
    "KeySchema": [  

```

Tables (2) Info									
<input type="text" value="Find tables by table name"/>					Any table tag		< 1 > ⚙		
<input type="checkbox"/>	Name	Status	Partition key	Sort key	Indexes	Read capacity mode	Write capacity mode		
<input type="checkbox"/>	Currency_UAH	Active	cc (S)	exchangedate (S)	0	Provisioned with auto scaling (1)	Provisioned with auto scaling (1)	2	
<input type="checkbox"/>	Currency_UAH_aws	Active	cc (S)	exchangedate (S)	0	Provisioned (10)	Provisioned (5)	0	

Рис. 9: Створення таблиці засобами AWS CLI

Додавання рядків до таблиці

Поступово додамо 4 рядки до новоствореної таблиці «Currency_UAH_aws». Зауважимо, що інформація про рядок має бути подана у json-форматі. Тож виконаємо команду виду

```
1 aws dynamodb put-item \  
2 --table-name Currency_UAH_aws \  
3 --item \  
4   '{"r030": {"N": "124"}, "txt": {"S": "Канадський" доллар}, "rate": {"N":  
5     "21.6966"}, "cc": {"S": "CAD"}, "exchangedate": {"S": "10.02.2021"}}' \  
--return-consumed-capacity TOTAL
```

На Рис. 10 та Рис. 11 виконано додавання двох елементів. Фінальний вигляд таблиця отримає на Рис. 12.


```
ubuntu@ip-172-31-81-119: ~$ aws dynamodb put-item \
> --table-name Currency_UAH_aws \
> --item \
> '{"r030": {"N": "124"}, "txt": {"S": "Канадський долар"}, "rate": {"N": "21.6966"}, "cc": {"S": "CAD"}, "exchangedate": {"S": "10.02.2021"}}' \
> --return-consumed-capacity TOTAL
{
  "ConsumedCapacity": {
    "TableName": "Currency_UAH_aws",
    "CapacityUnits": 1.0
  }
}
```

✔ Completed Read capacity units consumed: 0.5

Items returned (1)

<input type="checkbox"/>	cc	exchang...	r030	rate	txt
<input type="checkbox"/>	CAD	10.02.2021	124	21.6966	Канадський долар

Рис. 10: Додавання першого рядка до таблиці

```
ubuntu@ip-172-31-81-119: ~$ aws dynamodb put-item \
> --table-name Currency_UAH_aws \
> --item \
> '{"r030": {"N": "348"}, "txt": {"S": "Форинт"}, "rate": {"N": "0.093238"}, "cc": {"S": "HUF"}, "exchangedate": {"S": "10.05.2021"}}' \
> --return-consumed-capacity TOTAL
{
  "ConsumedCapacity": {
    "TableName": "Currency_UAH_aws",
    "CapacityUnits": 1.0
  }
}
```

Items returned (2)

<input type="checkbox"/>	cc	exchang...	r030	rate	txt
<input type="checkbox"/>	CAD	10.02.2021	124	21.6966	Канадський долар
<input type="checkbox"/>	HUF	10.05.2021	348	0.093238	Форинт

Рис. 11: Додавання другого рядка до таблиці

DynamoDB > Items > Currency_UAH_aws

Currency_UAH_aws

Autopreview Actions Create item Update table settings

► Scan/Query items
Expand to query or scan items.

Items returned (4)

<input type="checkbox"/>	cc	exchang...	r030	rate	txt
<input type="checkbox"/>	CAD	10.02.2021	124	21.6966	Канадський долар
<input type="checkbox"/>	HUF	10.05.2021	348	0.093238	Форинт
<input type="checkbox"/>	CZK	10.03.2021	203	1.2539	Чеська крона
<input type="checkbox"/>	SEK	10.04.2021	752	3.2561	Шведська крона

Рис. 12: Наповнена даними таблиця

Пошук елементів таблиці

Для пошуку певних рядків виконаємо команду, яка зображена на Рис. 13.

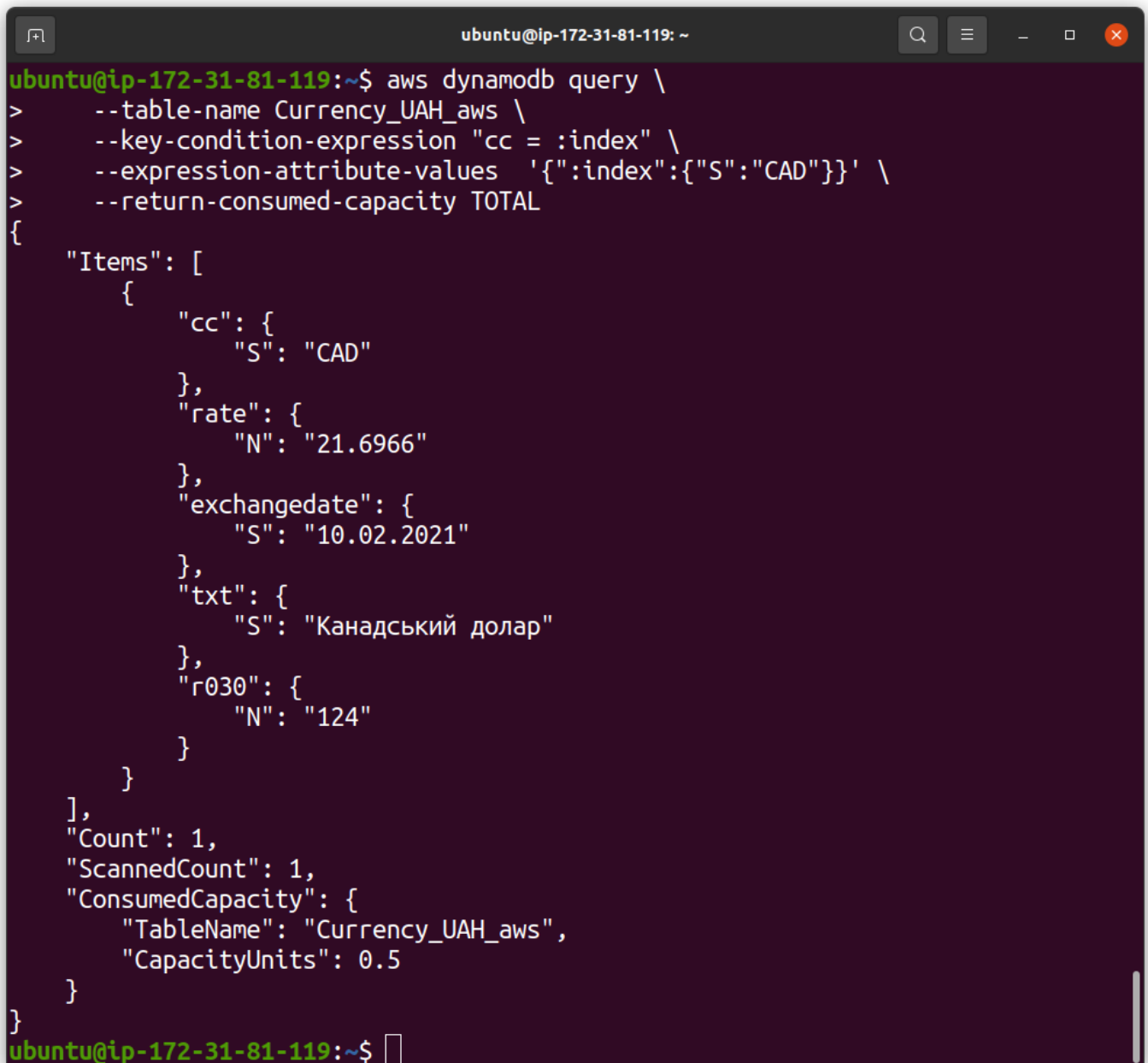
```
ubuntu@ip-172-31-81-119: ~$ aws dynamodb query \
> --table-name Currency_UAH_aws \
> --projection-expression "rate" \
> --key-condition-expression "cc = :index" \
> --expression-attribute-values '{":index":{"S":"SEK"}}' \
> --return-consumed-capacity TOTAL
{
  "Items": [
    {
      "rate": {
        "N": "3.2561"
      }
    }
  ],
  "Count": 1,
  "ScannedCount": 1,
  "ConsumedCapacity": {
    "TableName": "Currency_UAH_aws",
    "CapacityUnits": 0.5
  }
}
```

Рис. 13: Перший запит

Як бачимо, команда має вид

```
1 aws dynamodb query \  
2 --table-name Currency_UAH_aws \  
3 --projection-expression "rate" \  
4 --key-condition-expression "cc = :index" \  
5 --expression-attribute-values '{":index":{"S":"SEK"}}' \  
6 --return-consumed-capacity TOTAL
```

У рядках 4-5 через проміжний параметр `index` вказують рівність шуканої валюти: `"cc"="SEK"`, а у рядку номер 3 ми додатково вказуємо, що нас цікавить виключно значення `"rate"` обраної валюти (на противгу запиту на Рис. 14, де на вихід отримуємо весь рядок з шуканою валютою).



```
ubuntu@ip-172-31-81-119: ~  
ubuntu@ip-172-31-81-119:~$ aws dynamodb query \  
> --table-name Currency_UAH_aws \  
> --key-condition-expression "cc = :index" \  
> --expression-attribute-values '{":index":{"S":"CAD"}}' \  
> --return-consumed-capacity TOTAL  
{  
  "Items": [  
    {  
      "cc": {  
        "S": "CAD"  
      },  
      "rate": {  
        "N": "21.6966"  
      },  
      "exchangedate": {  
        "S": "10.02.2021"  
      },  
      "txt": {  
        "S": "Канадський долар"  
      },  
      "r030": {  
        "N": "124"  
      }  
    }  
  ],  
  "Count": 1,  
  "ScannedCount": 1,  
  "ConsumedCapacity": {  
    "TableName": "Currency_UAH_aws",  
    "CapacityUnits": 0.5  
  }  
}
```

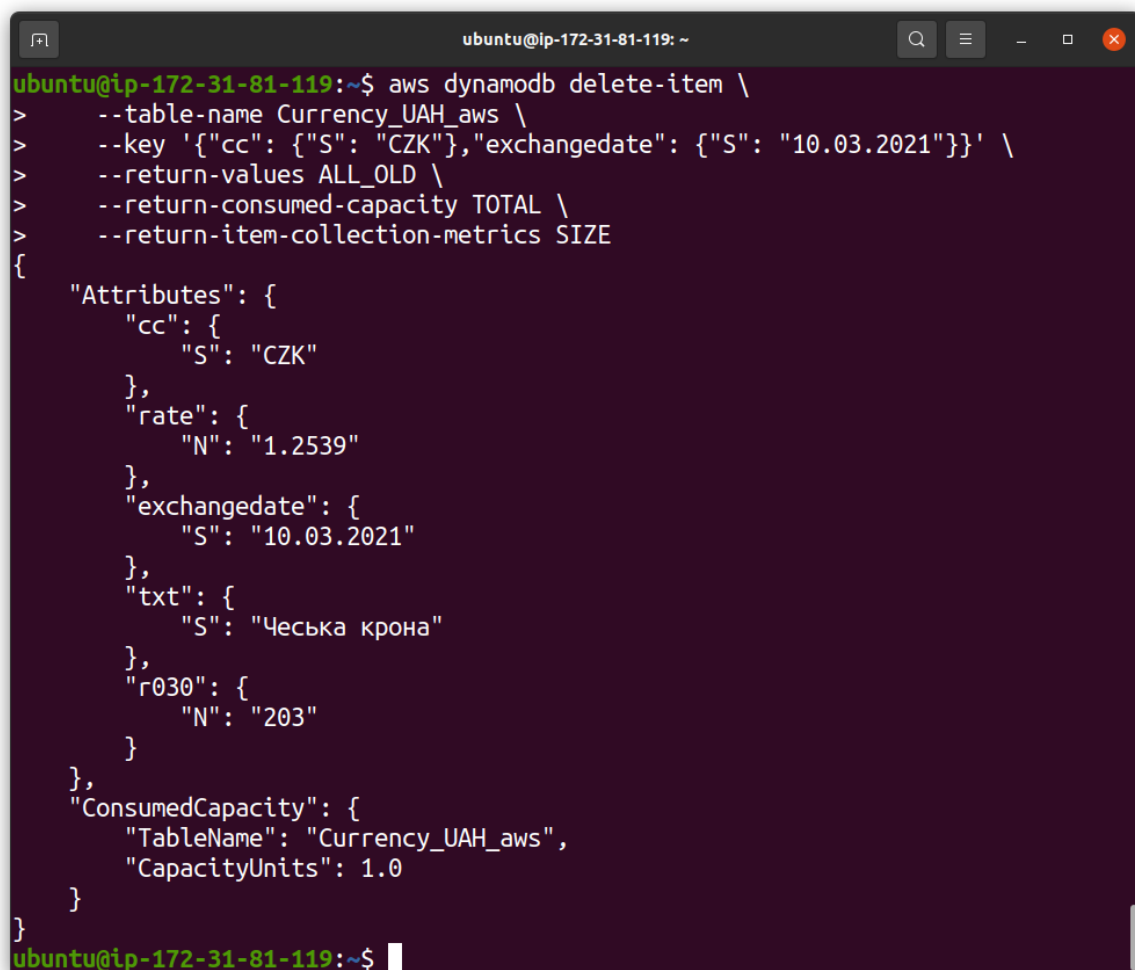
Рис. 14: Другий запит

Видалення рядків таблиці

Команда видалення рядка має просту структуру: необхідно лише вказати повне значення складеного Primary key.

```
1 aws dynamodb delete-item \  
2 --table-name Currency_UAH_aws \  
3 --key '{"cc": {"S": "CZK"},"exchangedate": {"S": "10.03.2021"}}' \  
4 --return-values ALL_OLD \  
5 --return-consumed-capacity TOTAL \  
6 --return-item-collection-metrics SIZE
```

Порівнявши таблиці на Рис. 12 та Рис. 15 переконуємося, що видалення успішне.



```
ubuntu@ip-172-31-81-119: ~  
ubuntu@ip-172-31-81-119:~$ aws dynamodb delete-item \  
> --table-name Currency_UAH_aws \  
> --key '{"cc": {"S": "CZK"},"exchangedate": {"S": "10.03.2021"}}' \  
> --return-values ALL_OLD \  
> --return-consumed-capacity TOTAL \  
> --return-item-collection-metrics SIZE  
{  
  "Attributes": {  
    "cc": {  
      "S": "CZK"  
    },  
    "rate": {  
      "N": "1.2539"  
    },  
    "exchangedate": {  
      "S": "10.03.2021"  
    },  
    "txt": {  
      "S": "Чеська крона"  
    },  
    "r030": {  
      "N": "203"  
    }  
  },  
  "ConsumedCapacity": {  
    "TableName": "Currency_UAH_aws",  
    "CapacityUnits": 1.0  
  }  
}  
ubuntu@ip-172-31-81-119:~$
```

Items returned (3)						
<div>< 1 > ⚙️ 🔍</div>						
<input type="checkbox"/>	cc	exchang...	r030	rate	txt	
<input type="checkbox"/>	CAD	10.02.2021	124	21.6966	Канадський долар	
<input type="checkbox"/>	HUF	10.05.2021	348	0.093238	Форинт	
<input type="checkbox"/>	SEK	10.04.2021	752	3.2561	Шведська крона	

Рис. 15: Видалення рядка з таблиці

3. Робота з таблицею через додаток NoSQL Workbench

Створення таблиці

Перш за все слід встановити додаток NoSQL Workbench для необхідної операційної системи, перейшовши за [посиланням](#). Конкретні кроки, які слід виконати після завантаження файлу на Linux, описані на цьому [сайті](#). В цілому, робота з файлом розширення .AppImage не викликала труднощів. Тож здійснюємо вхід у додаток й опиняємося на головні сторінці (Рис. 16).

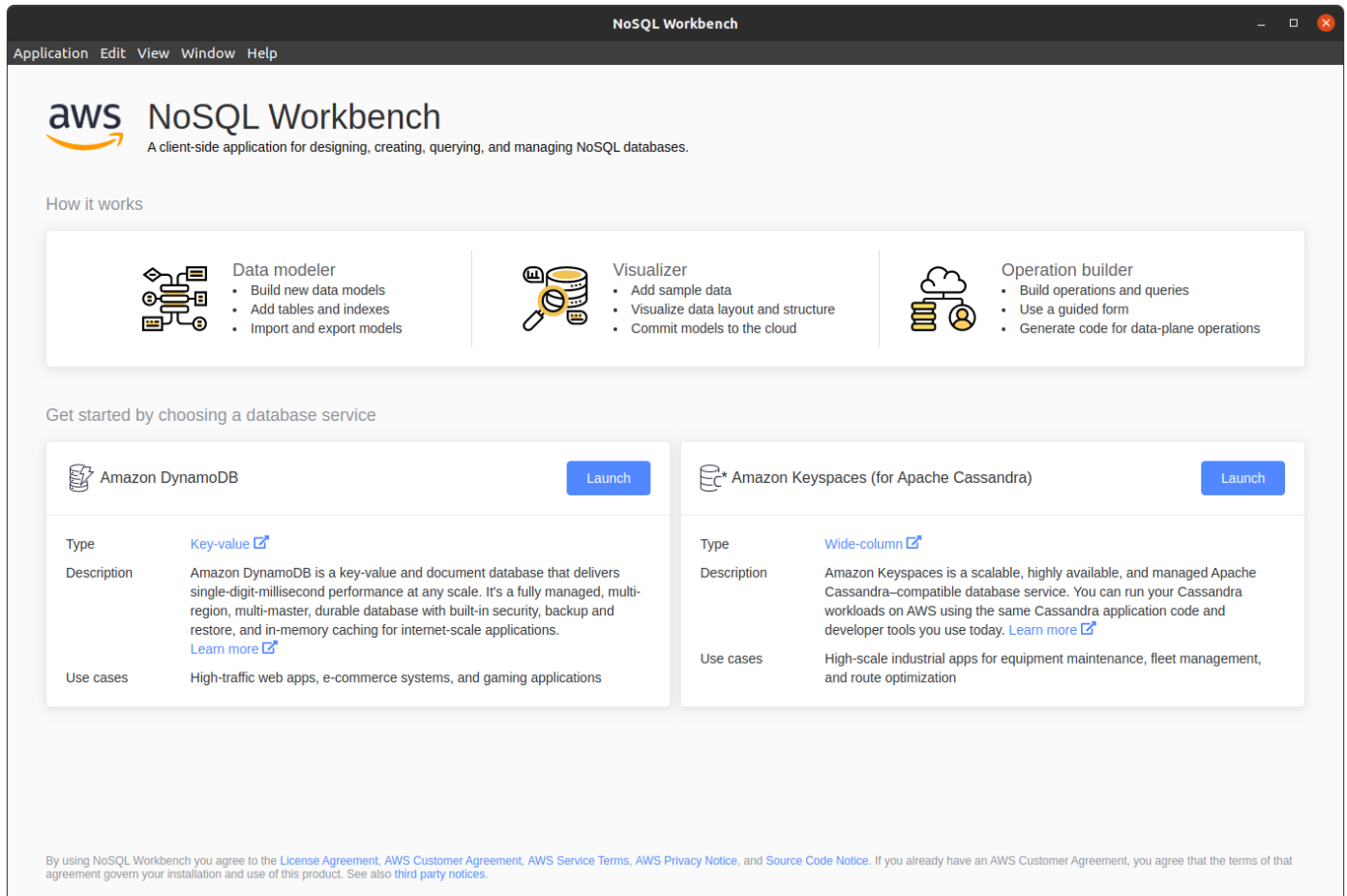


Рис. 16: Головна сторінка додатку NoSQL Workbench

Getting started

Get started with NoSQL Workbench for Amazon DynamoDB by creating your first data model or importing one. You can also start with one of the sample models below.

+ Create new data model

↓ Import data model

Create data model for Amazon DynamoDB

* Name

newproject

Author

anton

Description

Describe this data model

Cancel

Create

Рис. 17: Створення нової моделі даних

Отже, обираємо «Launch» для Amazon DynamoDB (Рис. 16). Далі створюємо так звану нову модель даних (Рис. 17). Тож у новоствореній моделі вже маємо можливість створити таблицю із потрібною нам назвою та потрібними для нас колонками. Процес створення та отриманий результат показаний на Рис. 18.

The screenshot displays the AWS DynamoDB console. On the left, the 'Add DynamoDB table' form is visible. It includes fields for 'Table name' (Currency_UAH_noSQL), 'Primary key attributes' (cc, String, Partition key; exchangedate, String, Sort key), and 'Other attributes' (r030, Number; text, String; rate, Number). On the right, the 'Data modeler' interface shows the 'Currency_UA...' table with the same attributes. The 'JSON view of data model' is also visible, showing the table structure in JSON format.

Рис. 18: Створення таблиці даних

Підключення таблиці до існуючого DynamoDB

Здійснимо підключення створеної таблиці до DynamoDB на акаунті Amazon. Оберемо розділ «Visualizer» та введемо усі необхідні параметри користувача сервісу Amazon (Рис. 19). Переконаємося в успішності підключення (Рис. 20).

The screenshot shows the AWS NoSQL Workbench Visualizer interface. A 'Commit to Amazon DynamoDB' dialog box is open, prompting the user to provide AWS credentials. The dialog includes fields for 'Connection name' (Connection 1), 'AWS Region' (us-east-1), 'Access key ID' (AWS access key ID), 'Secret access key' (AWS secret access key), 'Session token' (AWS session token), and 'IAM role ARN' (IAM role ARN). The background shows the 'Visualizer' tab with the 'Currency_UA...' table selected.

Рис. 19: Підключення таблиці до DynamoDB

DynamoDB > Tables

Tables (3) Info Refresh Actions Delete Create table

Find tables by table name Any table tag < 1 > Settings

<input type="checkbox"/>	Name	Status	Partition key	Sort key	Indexes	Read capacity mode	Write capacity mode
<input type="checkbox"/>	Currency_UAH	Active	cc (S)	exchangedate (S)	0	Provisioned with auto scaling (1)	Provisioned with auto scaling (1)
<input type="checkbox"/>	Currency_UAH_aws	Active	cc (S)	exchangedate (S)	0	Provisioned (10)	Provisioned (5)
<input type="checkbox"/>	Currency_UAH_noSQL	Active	cc (S)	exchangedate (S)	0	Provisioned with auto scaling (1)	Provisioned with auto scaling (1)

Рис. 20: Таблиця «Currency_UAH_noSQL» успішно підключена

4. Робота з таблицею засобами Python (додаткове завдання)

Створення таблиці

Перш за все запустимо вже встановлений у Лабораторній роботі №2 jupyter notebook на інстансі та працюватимемо з ним з локального робочого місця. Підключимося з консолі до інстансу й знову виконаємо команди:

```
jupyter notebook --no-browser --port 8889          на інстансі
ssh -i lab.pem -N -f -L localhost:8888:localhost:8889  локально
```

Створивши файл (наприклад, з назвою Lab2.ipynb), напишемо рядки коду, які зображено на Лістингу 1. Придивившись, легко впізнаємо абсолютно аналогічну схему команд, як це було у випадку роботи з таблицями засобами AWS CLI (Рис. 9).

Лістинг 1: Створення таблиці

```
1 import boto3
2 dynamoDB = boto3.resource("dynamodb")
3
4 table = dynamoDB.create_table(
5     TableName = "Currency_UAH_Python",
6
7     KeySchema = [
8         {
9             "AttributeName": "cc",
10            "KeyType": "HASH"
11        },
12        {
13            "AttributeName": "exchangedate",
14            "KeyType": "RANGE"
15        }
16    ],
17
18    AttributeDefinitions = [
19        {
20            "AttributeName": "cc",
21            "AttributeType": "S"
```

```

22     {
23         "AttributeName": "exchangedate",
24         "AttributeType": "S"
25     }],
26
27     ProvisionedThroughput={
28         "ReadCapacityUnits": 10,
29         "WriteCapacityUnits": 5
30     }
31 )

```

DynamoDB > Tables

Tables (4) Info Refresh Actions ▼ Delete Create table

Any table tag ▼ < 1 > ⚙

<input type="checkbox"/>	Name ▲	Status	Partition key	Sort key	Indexes	Read capacity mode	Write capacity mode
<input type="checkbox"/>	Currency_UAH	✔ Active	cc (S)	exchangedate (S)	0	Provisioned with auto scaling (1)	Provisioned with auto scaling (1)
<input type="checkbox"/>	Currency_UAH_aws	✔ Active	cc (S)	exchangedate (S)	0	Provisioned (10)	Provisioned (5)
<input type="checkbox"/>	Currency_UAH_noSQL	✔ Active	cc (S)	exchangedate (S)	0	Provisioned with auto scaling (1)	Provisioned with auto scaling (1)
<input type="checkbox"/>	Currency_UAH_Python	✔ Active	cc (S)	exchangedate (S)	0	Provisioned (10)	Provisioned (5)

Рис. 21: Таблиця «Currency_UAH_Python» успішно створена

Додавання рядків до таблиці

Послідовно додамо 4 рядки, виконуючи відповідну команду (Лістинг 2). Таким чином отримаємо заповнену даними й остаточно сформовану таблицю (Рис. 22).

Лістинг 2: Додавання двох рядків

```

33 table = dynamoDB.Table("Currency_UAH_Python")
34
35 response = table.put_item(
36     Item = {
37         "r030": 124,
38         "txt": "Канадський долар",
39         "rate": "21.6966",
40         "cc": "CAD",
41         "exchangedate": "10.02.2021"
42     }
43 )
44 response = table.put_item(
45     Item = {
46         "r030": 348,
47         "txt": "Форинт",
48         "rate": "0.093238",
49         "cc": "HUF",
50         "exchangedate": "10.05.2021"
51     }
52 )

```


DynamoDB > Items > Currency_UAH_Python

Tables (4)

Any table tag

Find tables by table name

< 1 > ⚙

- ☐ Currency_UAH
- ☐ Currency_UAH_aws
- ☐ Currency_UAH_noSQL
- ☒ Currency_UAH_Python

Currency_UAH_Python

☒ Autopreview

► **Scan/Query items**
Expand to query or scan items.

Items returned (4)

<input type="checkbox"/>	cc	exchang...	r030	rate	txt
<input type="checkbox"/>	CAD	10.02.2021	124	21.6966	Канадський долар
<input type="checkbox"/>	HUF	10.05.2021	348	0.093238	Форинт
<input type="checkbox"/>	CZK	10.03.2021	203	1.2539	Чеська крона
<input type="checkbox"/>	SEK	10.04.2021	752	3.2561	Шведська крона

Рис. 22: Наповнена даними таблиця

Пошук елементів таблиці

Лістинг 3: Виконання пошукових запитів

```

72 from boto3.dynamodb.conditions import Key
73
74 dynamoDB = boto3.resource("dynamodb")
75 table = dynamoDB.Table("Currency_UAH_Python")
76
77 response1 = table.query(KeyConditionExpression=Key("cc").eq("SEK"))
78 response2 = table.query(KeyConditionExpression=Key("cc").eq("CAD"))
79 response3 = table.query(KeyConditionExpression=Key("cc").eq("HUF") &
80                        Key("exchangedate").eq("10.05.2021"))
81
82 print(response1["Items"])
83 print(response2["Items"])
84 print(response3["Items"])

```

```

import boto3
from boto3.dynamodb.conditions import Key

dynamoDB = boto3.resource("dynamodb")
table = dynamoDB.Table("Currency_UAH_Python")

response1 = table.query(KeyConditionExpression=Key("cc").eq("SEK"))
response2 = table.query(KeyConditionExpression=Key("cc").eq("CAD"))
response3 = table.query(KeyConditionExpression=Key("cc").eq("HUF") & Key("exchangedate").eq("10.05.2021"))

print(response1["Items"])
print(response2["Items"])
print(response3["Items"])

[{'cc': 'SEK', 'rate': '3.2561', 'exchangedate': '10.04.2021', 'txt': 'Шведська крона', 'r030': Decimal('752')}]
[{'cc': 'CAD', 'rate': '21.6966', 'exchangedate': '10.02.2021', 'txt': 'Канадський долар', 'r030': Decimal('124')}]
[{'cc': 'HUF', 'rate': '0.093238', 'exchangedate': '10.05.2021', 'txt': 'Форинт', 'r030': Decimal('348')}]

```

Рис. 23: Результати запитів

Видалення рядків таблиці

Команда видалення рядка має просту структуру: необхідно лише вказати повне значення складеного Primary key (Лістинг 4).

Лістинг 4: Видалення рядка

```
85 dynamoDB = boto3.resource("dynamodb")
86 table = dynamoDB.Table("Currency_UAH_Python")
87
88 response = table.delete_item(Key = {"cc": "HUF", "exchangedate": "10.05.2021"})
```

Порівнявши таблиці на Рис. 22 та Рис. 24 переконаємося, що видалення рядка з таблиці пройшло успішно.

Items returned (3)							< 1 >		⚙	✕
<input type="checkbox"/>	cc	▼	exchang...	▼	r030	▼	rate	▼	txt	▼
<input type="checkbox"/>	CAD		10.02.2021		124		21.6966		Канадський долар	
<input type="checkbox"/>	CZK		10.03.2021		203		1.2539		Чеська крона	
<input type="checkbox"/>	SEK		10.04.2021		752		3.2561		Шведська крона	

Рис. 24: Рядок успішно видалений