

CREATION OF TABLE 'bsm_agg_data'

The screenshot shows the AWS DynamoDB console interface. The left sidebar contains navigation links for DynamoDB, Tables, Backups, Reserved capacity, Exports to S3, PartiQL editor, and Preferences. The main content area displays a message about the new DynamoDB console preview and a notification for Kinesis Data Streams for DynamoDB. Below these messages, there are buttons for 'Create table' and 'Delete table'. A table list is shown with columns: Name, Status, Partition key, Sort key, Indexes, Total read capacity, and Total write capacity. The table 'bsm_agg_data' is listed with a status of 'Active', a partition key of 'sensorType (String)', and a sort key of 'timestamp (String)'. The table has 0 indexes, 5 total read capacity, and 5 total write capacity.

console.aws.amazon.com/dynamodb/home?region=us-east-1&table:selected=bsm_agg_data&tab=overview

The preview of the new DynamoDB console is now available
We are redesigning the DynamoDB console. The preview of the new console is a work in progress, but we encourage you to try it and let us know what you think.

Kinesis Data Streams for DynamoDB is now available
You now can capture item-level changes in your DynamoDB tables as a Kinesis data stream and start taking advantage of Kinesis services to build advanced streaming applications.

Create table Delete table

Filter by table name X Choose a table group Actions 1 to 1 of 1 tables

Name	Status	Partition key	Sort key	Indexes	Total read capacity	Total write capacity
bsm_agg_data	Active	sensorType (String)	timestamp (String)	0	5	5

console.aws.amazon.com/dynamodb/home?region=us-east-1&table:

The preview of the new DynamoDB console is now available
We are redesigning the DynamoDB console. The preview of the new console is a work in progress, but we encourage you to try it and let us know what you think.

Kinesis Data Streams for DynamoDB is now available
You now can capture item-level changes in your DynamoDB tables as a Kinesis data stream and start taking advantage of Kinesis services to build advanced streaming applications.

Create table Delete table

Filter by table name X Choose a table group Actions 1 to 2 of 2 tables

Name	Status	Partition key	Sort key	Indexes	Total read capacity	Total write capacity
bsm_agg_data	Active	sensorType (String)	timestamp (String)	0	5	5
bsm_agg_data	Active	sensorType (String)	timestamp (String)	0	5	5

DEMONSTRATION OF FETCHING ALL THE DATA FROM TABLE

```
def fetch_all():  
    print ('Fetching all data from the DynamoDB table')  
    table = dynamodb.Table('BedSideMonitorData')  
    response = table.scan()  
    for item in response['Items']:  
        print (item)  
    print('Total items in the table are ', response['Count'])
```

```
def update_data(deviceid, timestamp, val):  
    print ('Update data in DynamoDB table')  
    table = dynamodb.Table('BedSideMonitorData')
```

```
if __name__ == '__main__':
```

ManageDynamoDB

```
{'value': Decimal('81'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:48.398854'}  
{'value': Decimal('66'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:49.398852'}  
{'value': Decimal('100'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:50.399165'}  
{'value': Decimal('69'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:51.399458'}  
{'value': Decimal('89'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:52.399613'}  
{'value': Decimal('86'), 'datatype': 'SPO2', 'deviceid': 'HC_101', 'timestamp': '2021-06-01 12:29:54.399688'}  
{'value': Decimal('90'), 'deviceid': 'HC_101', 'datatype': 'SPO2', 'timestamp': '2021-06-01 12:29:55.399140'}  
{'value': Decimal('96'), 'datatype': 'Temp', 'deviceid': 'HC_101', 'timestamp': '6/2/2021'}  
Total items in the table are 14032
```

```
Process finished with exit code 0
```

DEMONSTRATION OF UPDATE FUNCTION

(i) Before update

```
98
99 def update_data(deviceid, timestamp, val):
100     print ('Update data in DynamoDB table')
101     table = dynamodb.Table('BedSideMonitorData')
102     table.update_item(
103         Key={
104             'deviceid': deviceid,
105             'timestamp': timestamp
106         },
107         UpdateExpression='SET datatype = :val1',
108         ExpressionAttributeValues={
109             ':val1': val
110         }
111     )
112     print ('Value updated')
113
114 if __name__ == '__main__':
115
Run: ManageDynamoDB
```

['value': Decimal('51'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:46.606636']
['value': Decimal('63'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:47.399032']
['value': Decimal('81'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:48.398854']
['value': Decimal('66'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:49.398852']
['value': Decimal('100'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:50.399165']
['value': Decimal('69'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:51.399458']
['value': Decimal('89'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:52.399613']
['value': Decimal('79'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:53.399744']
['value': Decimal('86'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:54.398688']
['value': Decimal('90'), 'deviceid': 'HC_101', 'datatype': 'SPO2', 'timestamp': '2021-06-01 12:29:55.398148']
Sensor data from HC_101
06/01/2021
06/02/2021 06:00:00
<class 'pandas.core.frame.DataFrame'>

(ii) After update

```
99
100 def update_data(deviceid, timestamp, val):
101     print ('Update data in DynamoDB table')
102     table = dynamodb.Table('BedSideMonitorData')
103     table.update_item(
104         Key={
105             'deviceid': deviceid,
106             'timestamp': timestamp
107         },
108         UpdateExpression='SET datatype = :val1',
109         ExpressionAttributeValues={
110             ':val1': val
111         }
112     )
113
114 if __name__ == '__main__':
115
Run: ManageDynamoDB
```

['value': Decimal('56.9'), 'deviceid': 'HC_101', 'datatype': 'Temperature', 'timestamp': '2021-06-01 12:29:45.399058']
['value': Decimal('65'), 'deviceid': 'HC_101', 'datatype': 'SPO2', 'timestamp': '2021-06-01 12:29:45.697606']
['value': Decimal('77'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:46.007622']
['value': Decimal('51'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:46.606636']
['value': Decimal('63'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:47.399032']
['value': Decimal('81'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:48.398854']
['value': Decimal('66'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:49.398852']
['value': Decimal('100'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:50.399165']
['value': Decimal('69'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:51.399458']
['value': Decimal('89'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:52.399613']
['value': Decimal('79'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:53.399744']
['value': Decimal('86'), 'deviceid': 'HC_101', 'datatype': 'SPO2', 'timestamp': '2021-06-01 12:29:54.398688']
['value': Decimal('90'), 'deviceid': 'HC_101', 'datatype': 'SPO2', 'timestamp': '2021-06-01 12:29:55.398148']
Sensor data from HC_101

DEMONSTRATION OF DELETE FUNCTIONALITY

(i) Before delete

```
def delete_data(deviceid, timestamp):
    print ('Delete data in DynamoDB table')
    table = dynamodb.Table('BedSideMonitorData')
    table.delete_item(
        Key={
            'deviceid': deviceid,
            'timestamp': timestamp
        },
    )
    print ('Item deleted left in the table are ', table.item_count)

if __name__ == '__main__':
    deviceid = 'HC_101'
    insert_BSM_data('Temp', '6/21/2021', deviceid, 96)
    fetch_all()
    update_data('HC_101', '2021-06-01 12:29:54.398688', 'SPO2')
    delete_data('HC_101', '2021-06-01 12:29:53.398744')

if __name__ == '__main__':
    ManageDynamoDB =
```

```
{
  'value': Decimal('66'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:49.398852'
}, {
  'value': Decimal('100'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:50.399165'
}, {
  'value': Decimal('69'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:51.399458'
}, {
  'value': Decimal('89'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:52.399613'
}, {
  'value': Decimal('79'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:53.398744'
}, {
  'value': Decimal('86'), 'datatype': 'SPO2', 'deviceid': 'HC_101', 'timestamp': '2021-06-01 12:29:54.398688'
}, {
  'value': Decimal('90'), 'deviceid': 'HC_101', 'datatype': 'SPO2', 'timestamp': '2021-06-01 12:29:55.398148'
}
Sensor data from HC_101
06/01/2021
```

(ii) After delete

```
def delete_data(deviceid, timestamp):
    print ('Delete data in DynamoDB table')
    table = dynamodb.Table('BedSideMonitorData')
    table.delete_item(
        Key={
            'deviceid': deviceid,
            'timestamp': timestamp
        },
    )
    print ('Item deleted left in the table are ', table.item_count)

if __name__ == '__main__':
    deviceid = 'HC_101'
    insert_BSM_data('Temp', '6/21/2021', deviceid, 96)
    fetch_all()
    update_data('HC_101', '2021-06-01 12:29:54.398688', 'SPO2')
    delete_data('HC_101', '2021-06-01 12:29:53.398744')

if __name__ == '__main__':
    ManageDynamoDB =
```

```
{
  'value': Decimal('63'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:47.399032'
}, {
  'value': Decimal('81'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:48.398854'
}, {
  'value': Decimal('66'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:49.398852'
}, {
  'value': Decimal('100'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:50.399165'
}, {
  'value': Decimal('69'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:51.399458'
}, {
  'value': Decimal('89'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:52.399613'
}, {
  'value': Decimal('86'), 'datatype': 'SPO2', 'deviceid': 'HC_101', 'timestamp': '2021-06-01 12:29:54.398688'
}, {
  'value': Decimal('90'), 'deviceid': 'HC_101', 'datatype': 'SPO2', 'timestamp': '2021-06-01 12:29:55.398148'
}
Sensor data from HC_101
06/01/2021
```

DEMONSTRATION OF INSERT FUNCTION

(i) Before insert

```
def insert_BSM_data(datatype, timestamp, deviceid, value):
    print ('Data Insertion')
    table = dynamodb.Table('BedSideMonitorData')
    table.put_item(
        Item={
            'datatype': datatype,
            'timestamp': timestamp,
            'deviceid': deviceid,
            'value': value
        }
    )
    print ('Total items in the table are ', table.item_count)

if __name__ == '__main__':
    ManageDynamoDB

{'value': Decimal('63'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:47.399032'}
{'value': Decimal('81'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:48.398854'}
{'value': Decimal('66'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:49.398852'}
{'value': Decimal('100'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:50.399165'}
{'value': Decimal('69'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:51.399458'}
{'value': Decimal('69'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:52.399613'}
{'value': Decimal('86'), 'datatype': 'SPO2', 'deviceid': 'HC_101', 'timestamp': '2021-06-01 12:29:54.398688'}
{'value': Decimal('90'), 'deviceid': 'HC_101', 'datatype': 'SPO2', 'timestamp': '2021-06-01 12:29:55.398148'}
Sensor data from HC_101
06/01/2021
.....
```

(ii) After insert

```
def insert_BSM_data(datatype, timestamp, deviceid, value):
    print ('Data Insertion')
    table = dynamodb.Table('BedSideMonitorData')
    table.put_item(
        Item={
            'datatype': datatype,
            'timestamp': timestamp,
            'deviceid': deviceid,
            'value': value
        }
    )
    print ('Total items in the table are ', table.item_count)

insert_BSM_data()

ManageDynamoDB

{'value': Decimal('81'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:48.398854'}
{'value': Decimal('66'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:49.398852'}
{'value': Decimal('100'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:50.399165'}
{'value': Decimal('69'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:51.399458'}
{'value': Decimal('89'), 'deviceid': 'HC_101', 'datatype': 'HeartRate', 'timestamp': '2021-06-01 12:29:52.399613'}
{'value': Decimal('86'), 'datatype': 'SPO2', 'deviceid': 'HC_101', 'timestamp': '2021-06-01 12:29:54.398688'}
{'value': Decimal('90'), 'deviceid': 'HC_101', 'datatype': 'SPO2', 'timestamp': '2021-06-01 12:29:55.398148'}
{'value': Decimal('96'), 'datatype': 'Temp', 'deviceid': 'HC_101', 'timestamp': '6/2/2021'}
Sensor data from HC_101
06/01/2021
```

