Trabajo Tema 4: Otros modelos de datos para SI. DynamoDB.

Nombre del grupo: “Por Asignar”

## Introducción

DynamoDB es un servicio de base de datos NoSQL en la nube ofrecido por Amazon Web Services (AWS). Se trata de una base de datos altamente escalable y disponible, diseñada para manejar grandes cantidades de datos con altas tasas de consulta y actualización.

Algunas de sus características clave incluyen el almacenamiento de datos en tablas de clave-valor o clave-atributo, el soporte para consultas y escrituras de alta velocidad a través de índices secundarios y la escalabilidad horizontal automática. En resumen, se trata de una opción de base de datos muy útil para aplicaciones que requieren altas tasas de consulta y actualización, así como una alta disponibilidad. Es especialmente útil para aplicaciones que necesitan un almacenamiento de datos distribuido y una escalabilidad horizontal automática.

## Descarga e instalación de DynamoDB.

Para comenzar a trabajar con DynamoDB nos hemos descargado NoSQL WorkBench para Amazon DynamoDB. NoSQL Workbench es una herramienta IDE que proporciona funciones de modelado de datos, visualización de datos y desarrollo de consultas para ayudarnos a diseñar, crear, consultar y administrar tablas de DynamoDB.

Para instalarnos el IDE accedemos a la siguiente página:

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/workbench.settingup.html>

y ejecutamos el archivo .run. Seguimos los pasos de instalación, introduciendo nuestro ID de usuario, el usuario y la contraseña, y finalizará la instalación.

## Descripción del DDL y DML.

El lenguaje de definición de datos (DDL) de DynamoDB se utiliza para crear y eliminar tablas y para definir el esquema de una tabla. Los comandos DDL más comunes son:

* CreateTable: utilizado para crear una nueva tabla. La sintaxis para CreateTable es la siguiente:

| {  "AttributeDefinitions": [  {  "AttributeName": "string",  "AttributeType": "string"  }  ],  "KeySchema": [  {  "AttributeName": "string",  "KeyType": "string"  }  ],  "TableName": "string",  } |
| --- |

En esta estructura, las claves de partición y ordenación son especificadas en “KeySchema”:”KeyType”. La clave de partición es el atributo que se utiliza para dividir los elementos de la tabla en grupos (o particiones) de acuerdo con su valor. La clave de ordenación es un atributo opcional que se utiliza para ordenar los elementos dentro de cada partición. También en este formato podemos especificar tamaño y capacidad de lectura y escritura, valores que afectarán al rendimiento y al costo de la tabla.

* DeleteTable: utilizado para eliminar una tabla existente. La sintaxis para DeleteTable es la siguiente:

| { "TableName": "string" } |
| --- |

* DescribeTable: utilizado para obtener información sobre una tabla, como el esquema y el estado actual de la tabla. La sintaxis para DescribeTable es la siguiente:

| { "TableName": "string" } |
| --- |

El lenguaje de manipulación de datos (DML) de DynamoDB se utiliza para insertar, actualizar y eliminar datos en una tabla. Los comandos DML más comunes son:

* PutItem: utilizado para insertar un nuevo elemento en una tabla o para actualizar un elemento existente. La sintaxis para PutItem es la siguiente:

| { "Item": "string"  "TableName": "string" } |
| --- |

* GetItem: utilizado para recuperar un conjunto de elementos de una tabla mediante su clave de partición. La sintaxis para GetItem es la siguiente:

| { "Key": "string",  "TableName": "string" } |
| --- |

* DeleteItem: utilizado para eliminar un elemento de una tabla mediante su clave de partición. La sintaxis para DeleteItem es la siguiente:

| { "Key": "string",  "TableName": "string" } |
| --- |

Nota: El número de parámetros y tipo de parámetros en cada sentencia es muy grande, por lo que en todos los ejemplos hemos incluido sólo los parámetros requeridos.

Además de estos comandos básicos, DynamoDB ofrece un conjunto completo de operaciones de consulta y escritura que permiten realizar operaciones más complejas en las tablas, como búsquedas de elementos que cumplan ciertos criterios o actualizaciones masivas de elementos. Algunos ejemplos de operaciones de consulta y escritura disponibles son:

* Query: utilizado para recuperar elementos de una tabla que cumplan ciertos criterios. La sintaxis para Query es la siguiente:

| { "AttributesToGet": [ "string" ],  "ExpressionAttributesNames": { "string" : "string" },  "ExpressionAttributesValues": { "string" : ... },  "KeyConditionExpression": "string",  "KeyConditions": "string",  "Select": "string"  "TableName": "string" } |
| --- |

* Scan: utilizado para recuperar elementos de una tabla mediante un escaneo completo de la tabla. La sintaxis para Scan es la siguiente:

| { "AttributesToGet": [ "string" ],  "ExpressionAttributesNames": { "string" : "string" },  "ExpressionAttributesValues": { "string" : ... },  "Select": "string"  "TableName": "string" } |
| --- |

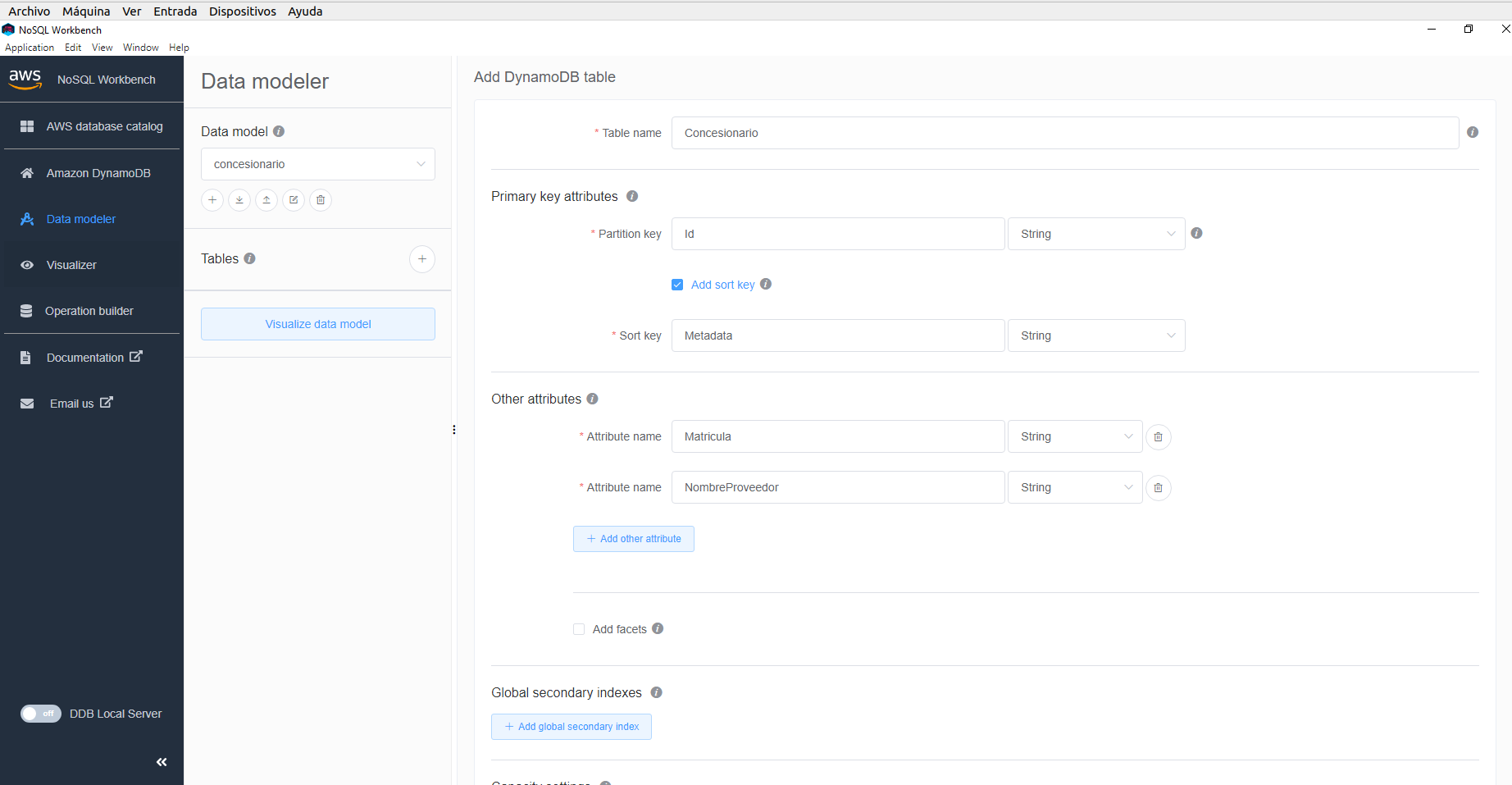
* BatchGetItem: utilizado para recuperar los atributos de uno o más elementos de una o más tablas. La sintaxis para BatchGetItem es la siguiente:

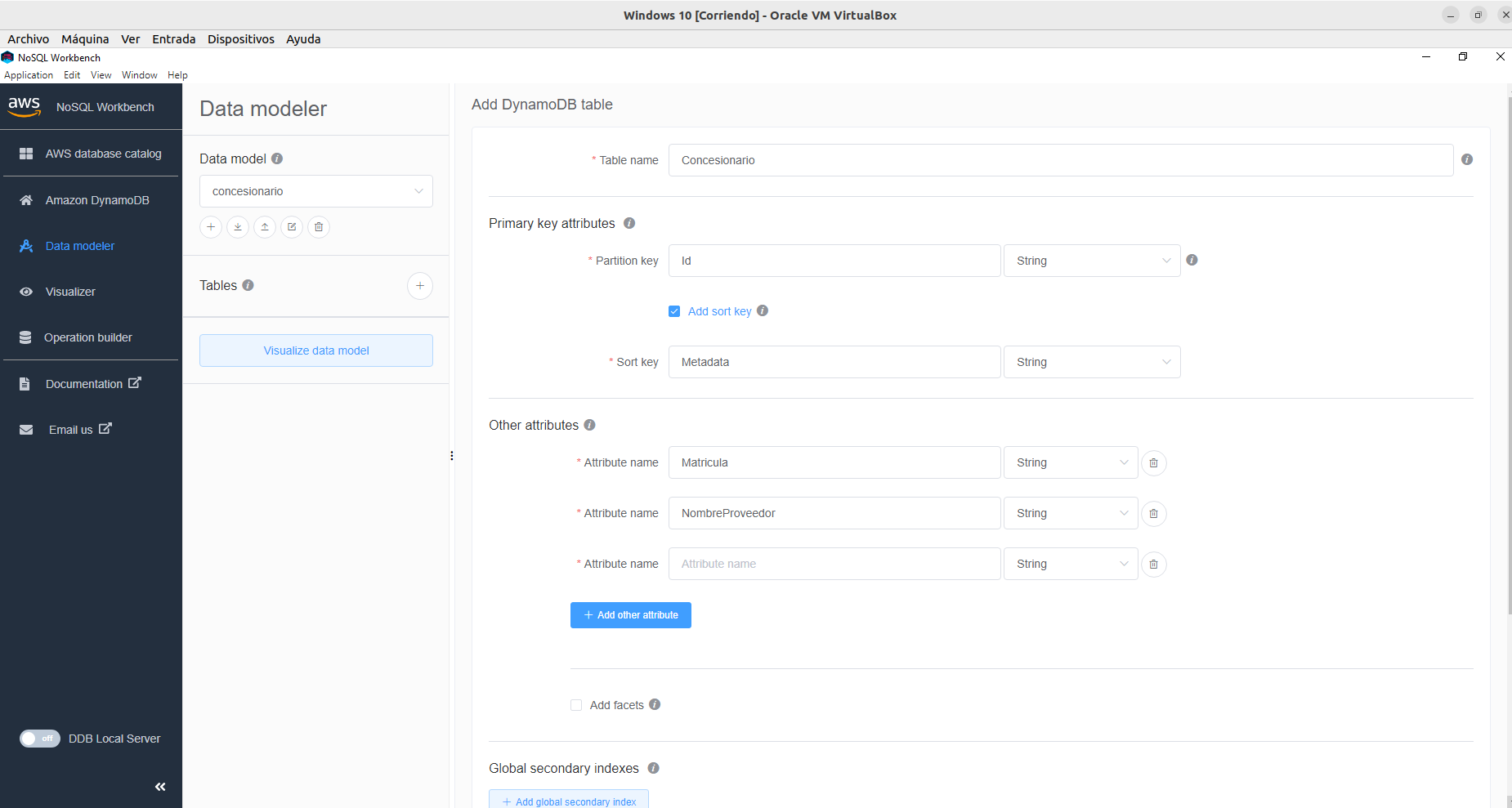
| { "RequestItems": {  "string" : {  "AttributesToGet": [ "string" ],  "ConsistentRead": boolean,  "ExpressionAttributeNames": { "string" : "string" },  "Keys": [ { "string" : ... } ],  "ProjectionExpression": "string" } } } |
| --- |

## Sentencias empleadas para la creación de estructuras, inserción/modificación/borrado de datos.

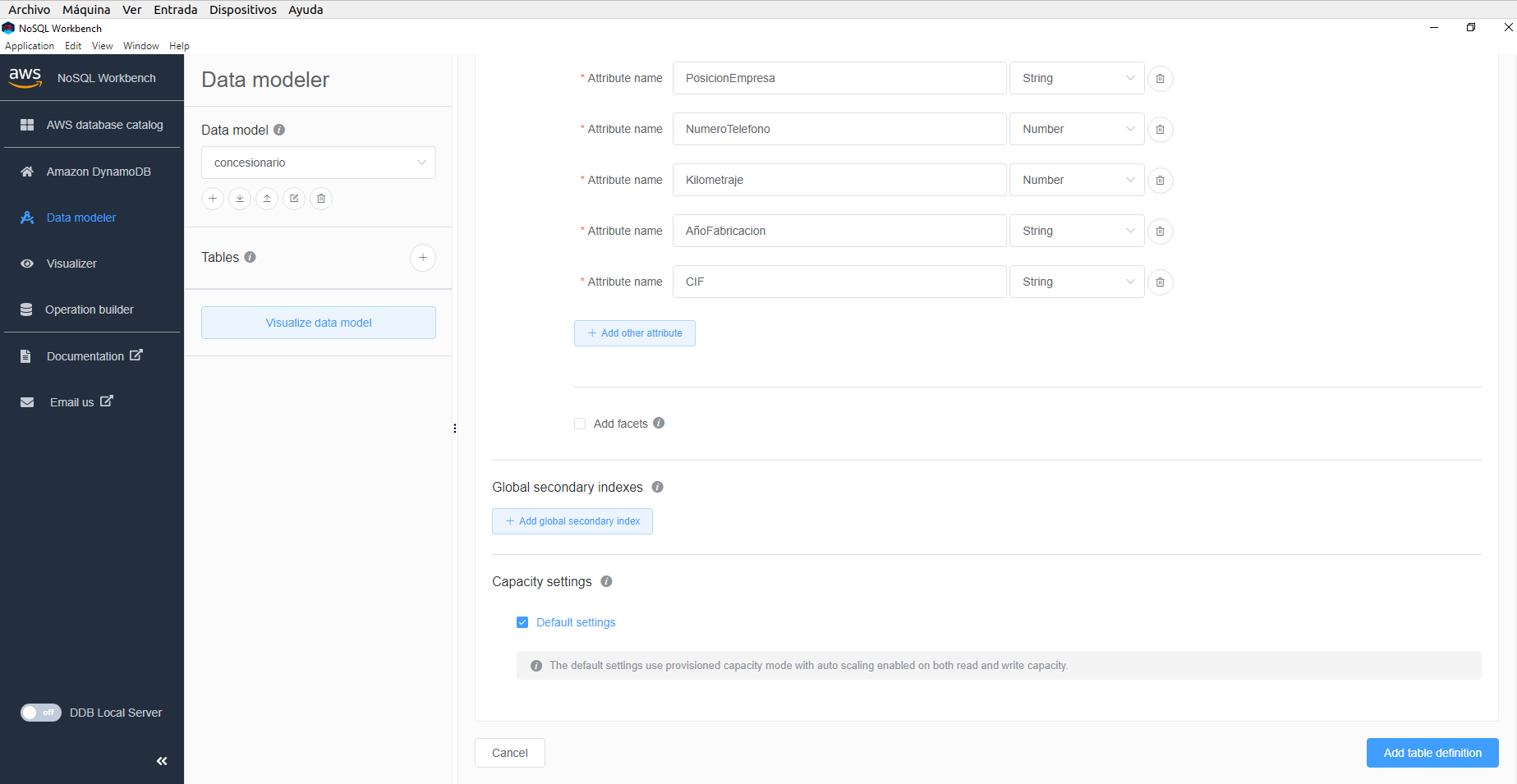
Para realizar la creación de estructuras y la inserción/modificación/borrado de datos hemos empleado la herramienta IDE NoSQL Workbench tal y como se ha mencionado anteriormente.

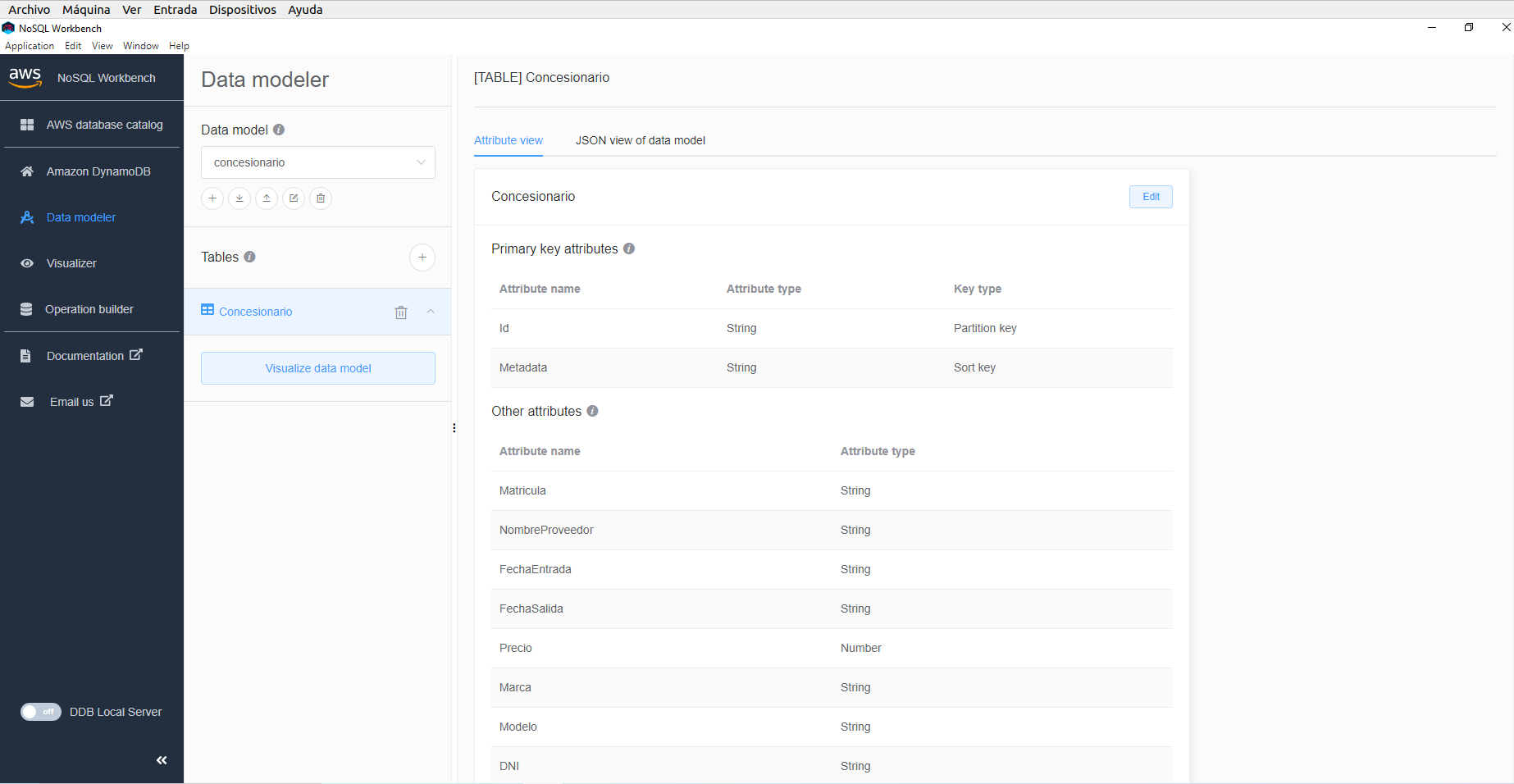
En primer lugar, para crear una nueva tabla debemos irnos al apartado Data modeler y hacer click en el símbolo + que hay a la derecha de pestaña Tables. Tras hacerlo se nos desplegará el cuadro Add DynamoDB table. También hemos rellenado los datos de la Partition key y de la Sort key que vamos a utilizar y hemos añadido algún que otro atributo tal y como se muestra a continuación:



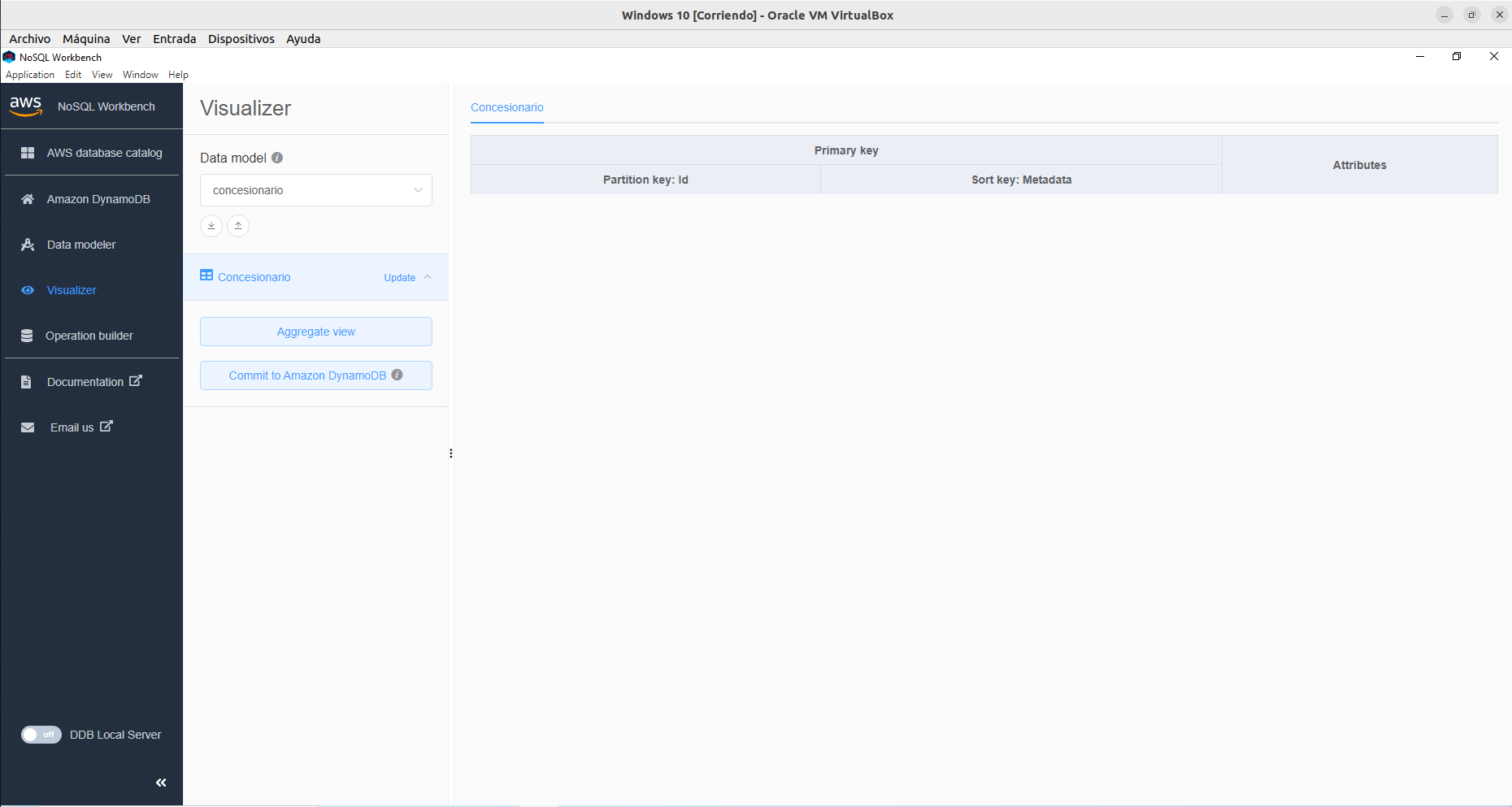
Para añadir atributos hacemos click en el botón Add other attribute y nos aparecerán dos cuadros vacíos para insertar el nombre del atributo y el tipo del mismo.

Una vez que tenemos todos los atributos insertados, solo nos queda hacer click en el botón Add table definition y hemos terminado de definir el esquema de la tabla.

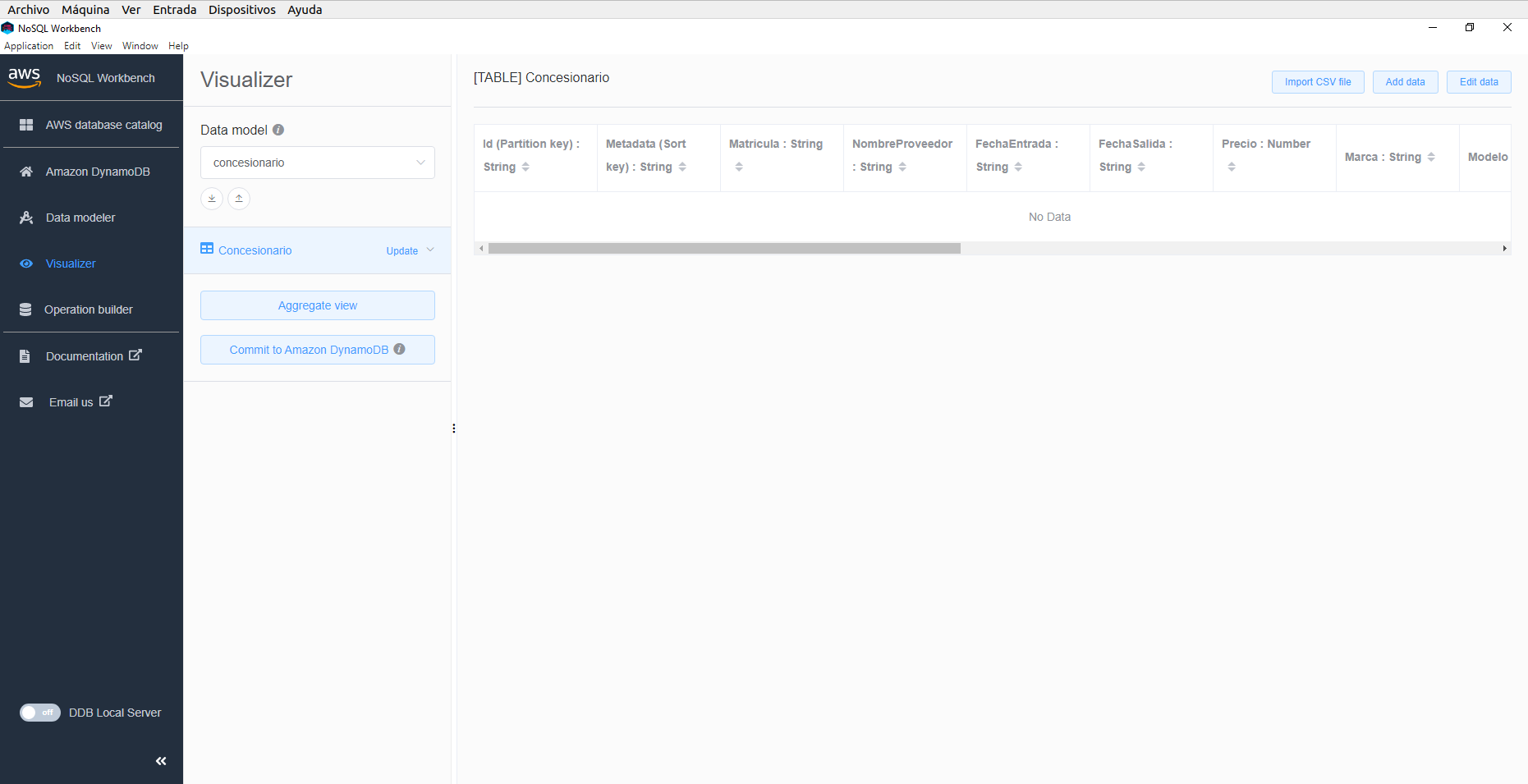
A continuación se muestra el resultado:



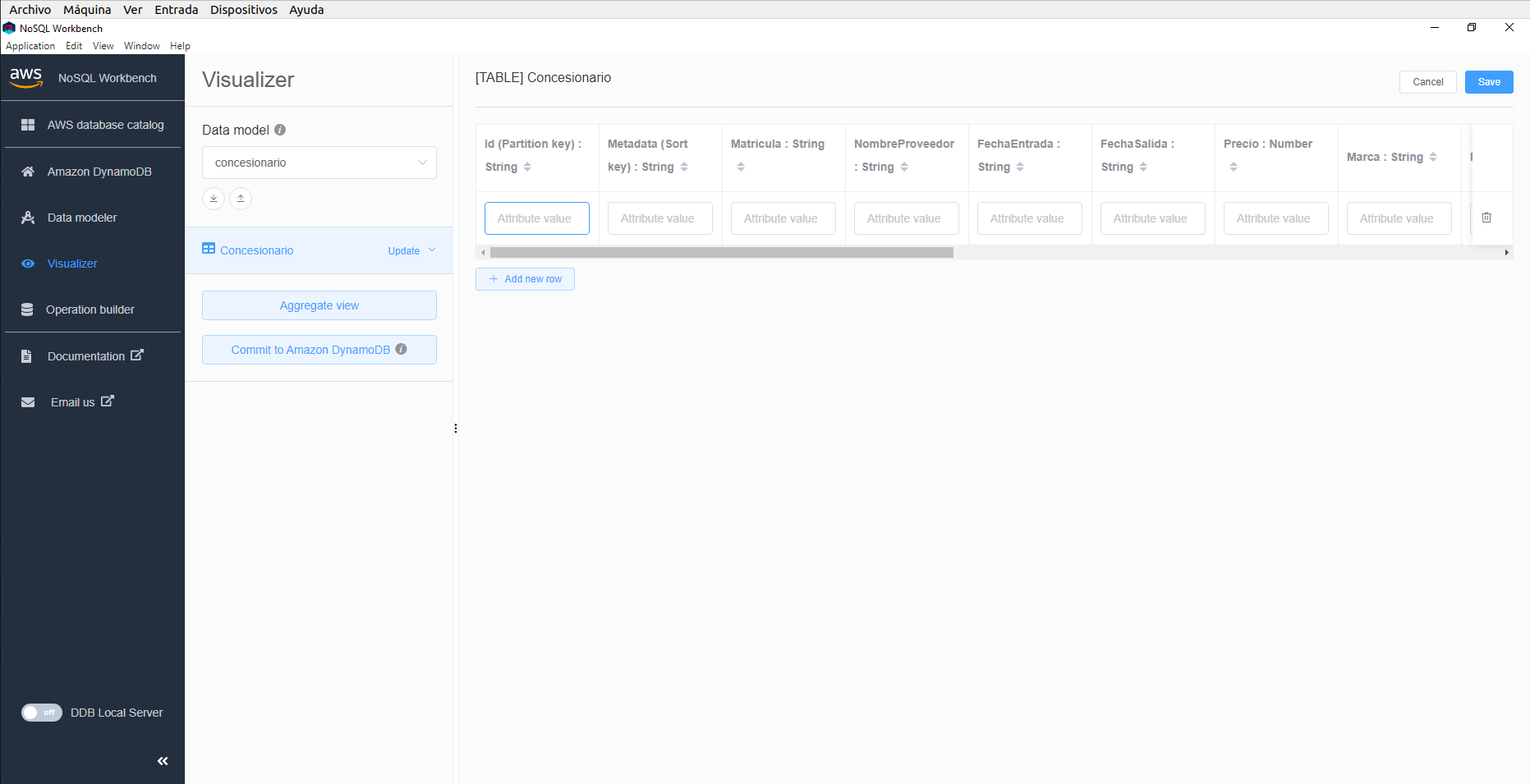
Ahora, para insertar una tupla, debemos cambiarnos a las pestaña Visualizer.

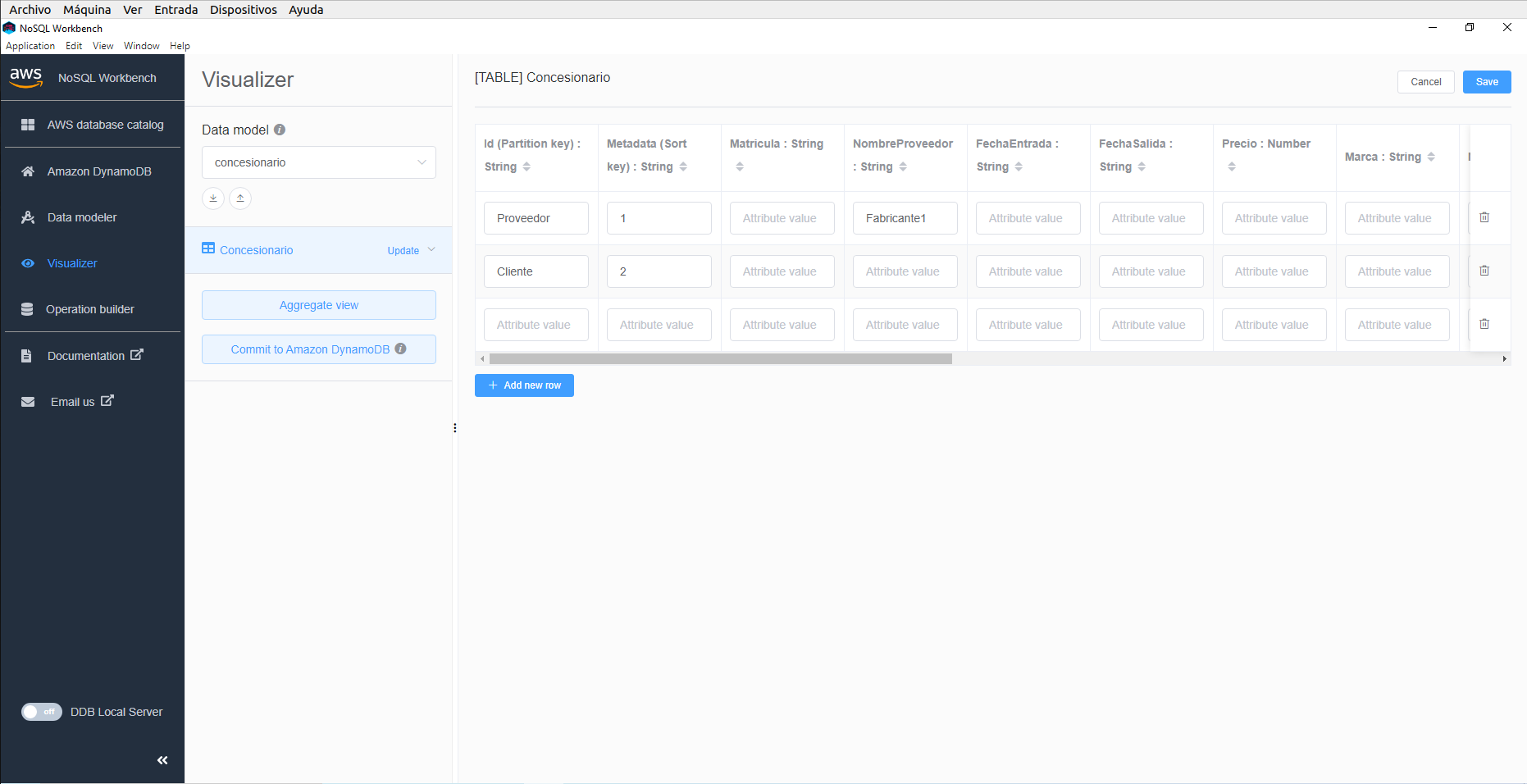


Desde allí, haremos click en Update y se nos abrirá la tabla Concesionario a la derecha, tal y como se muestra a continuación:

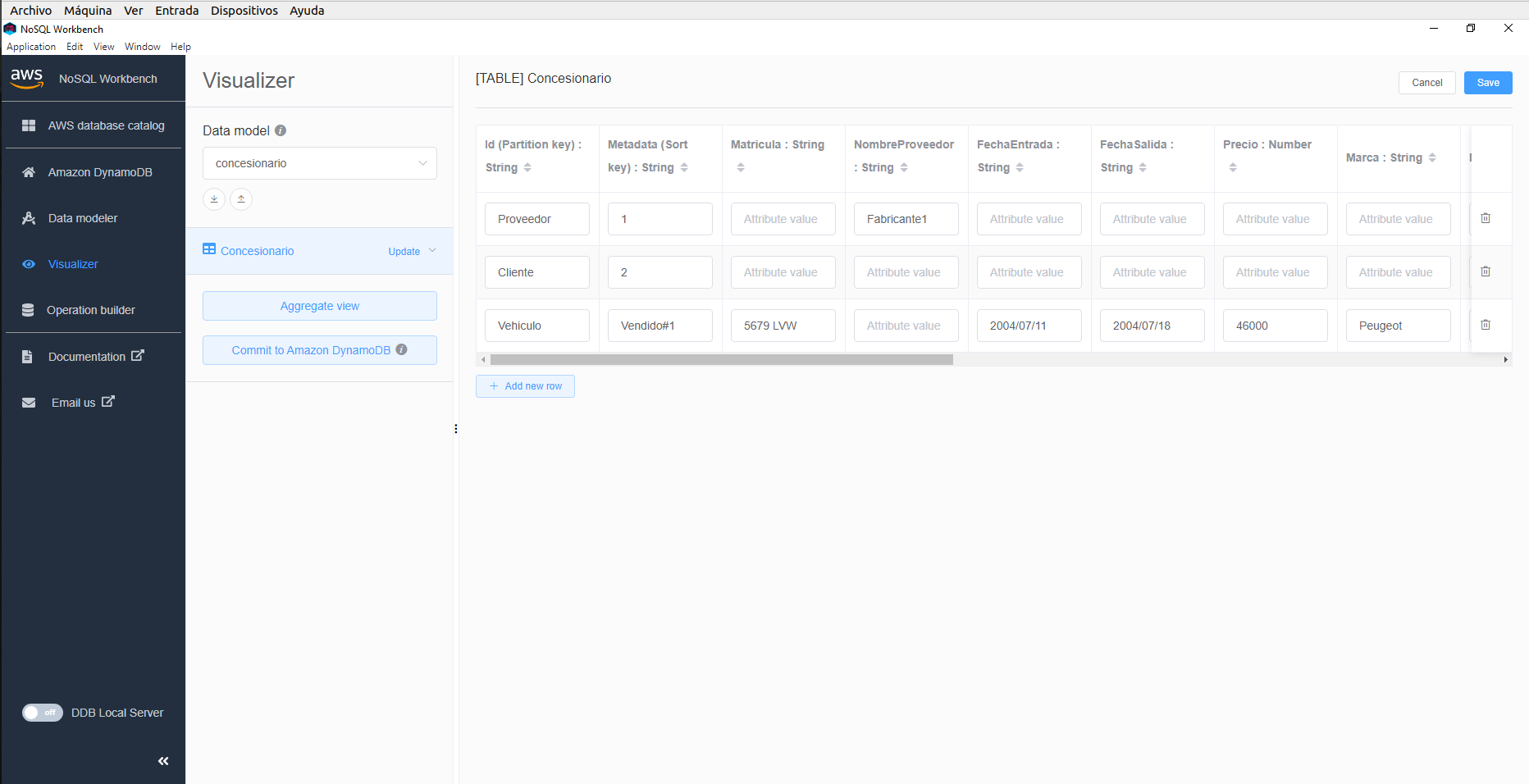


Desde allí, para insertar un registro hacemos click en Add data. Así pues, aparecerán cuadros de texto para ir rellenando los valores de los diferentes campos que hemos definido en el esquema. Nótese que no tenemos por qué rellenarlos todos.

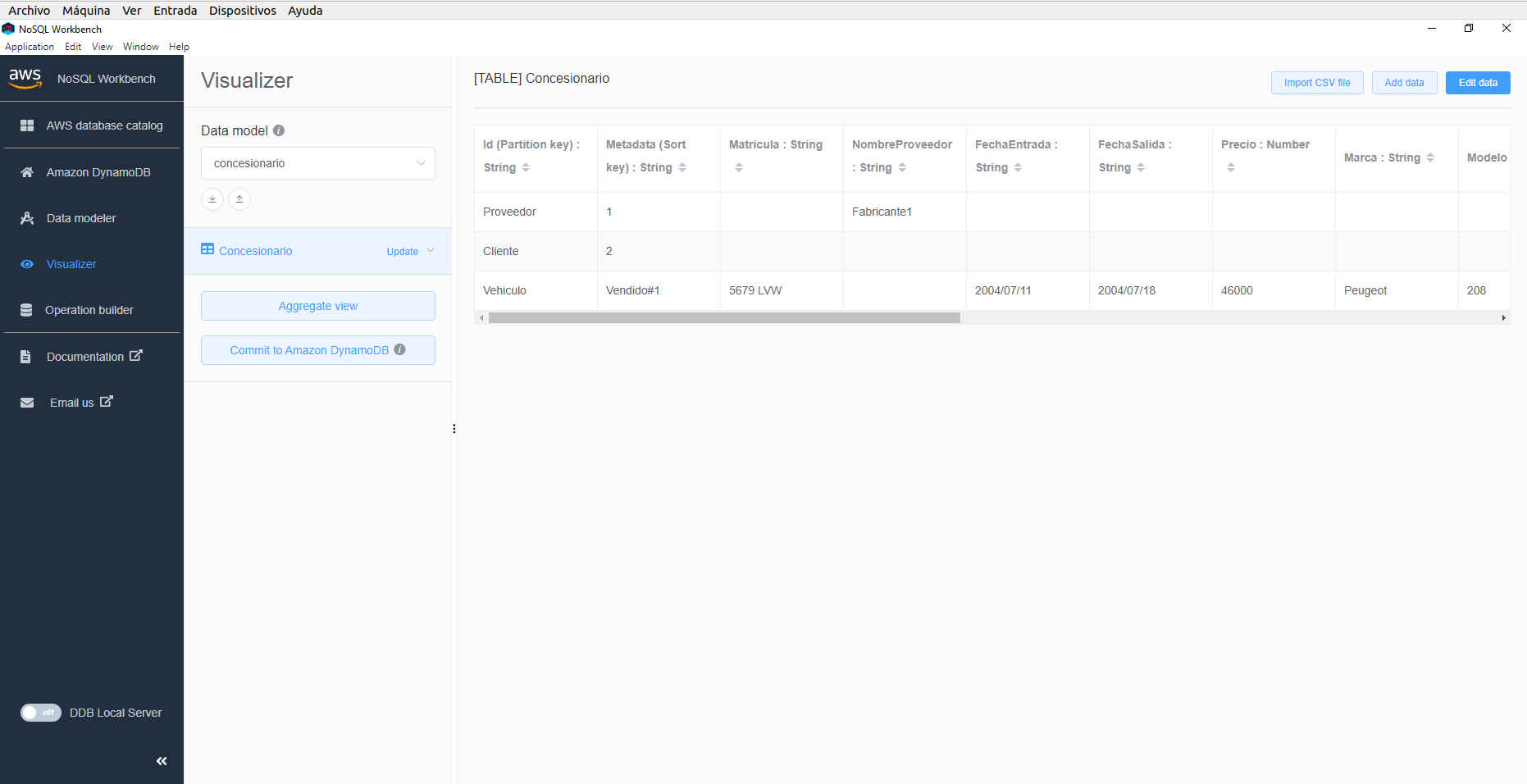
Cada vez que queramos insertar una nueva fila haremos click en Add new row.

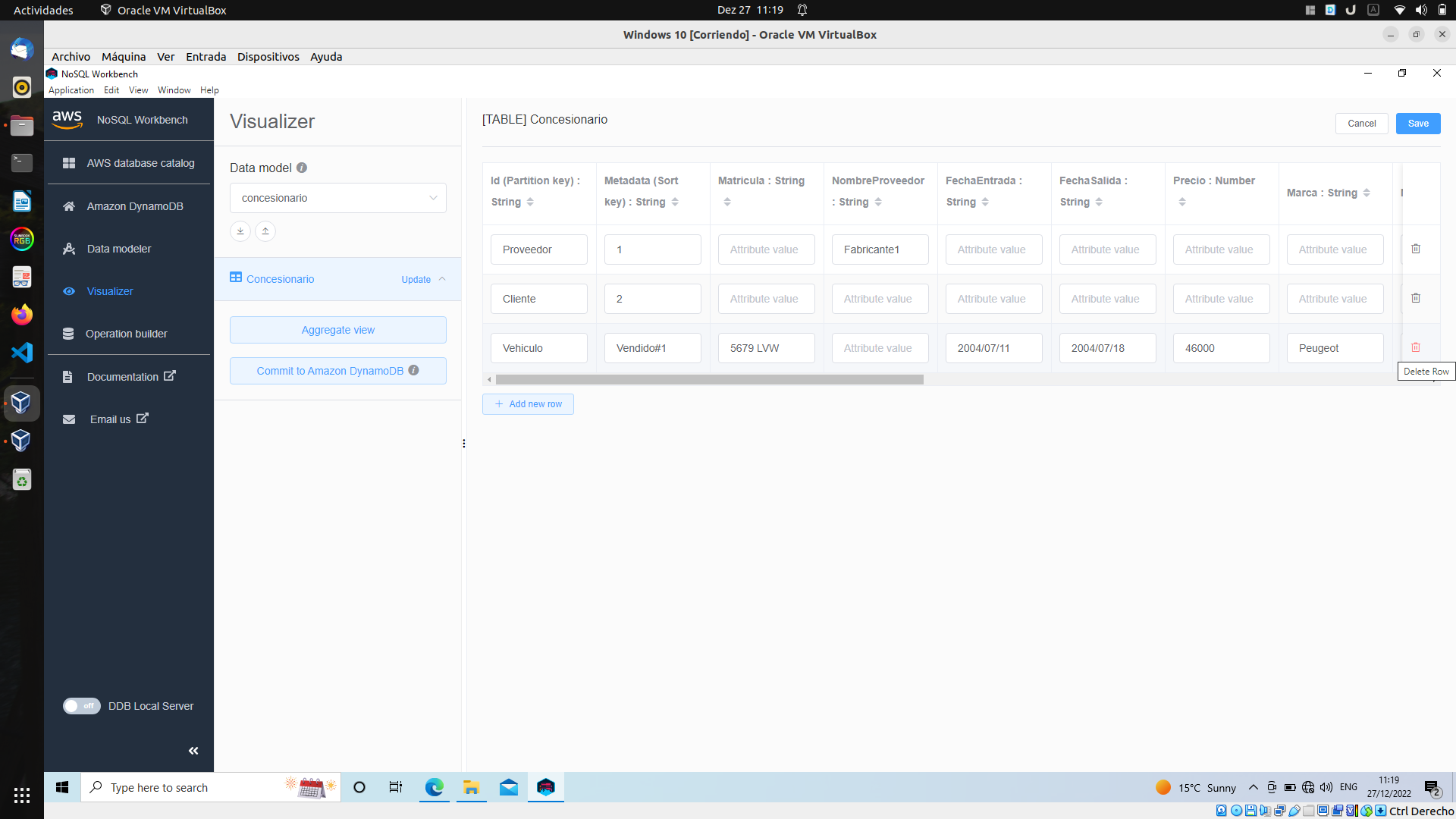


Cuando hayamos introducido todos los registros que queríamos, solo tenemos que darle al botón de Save para terminar.

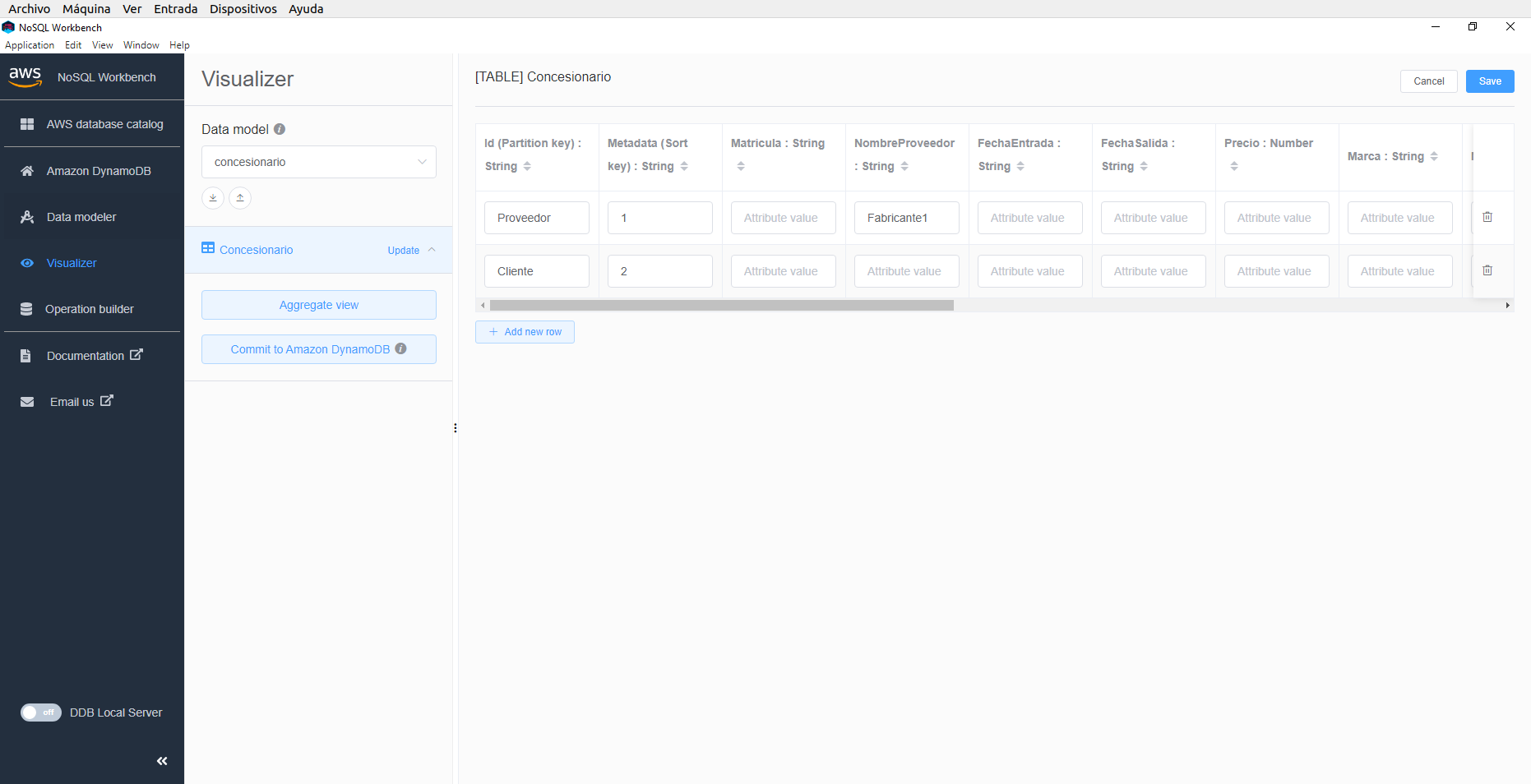


En la imagen que se muestra a continuación se ven los nuevos registros insertados tras salir del modo edición. Para modificar una tupla, simplemente le daríamos al botón Edit data y modificaríamos los cuadros de texto asociados a la tupla que queremos cambiar.

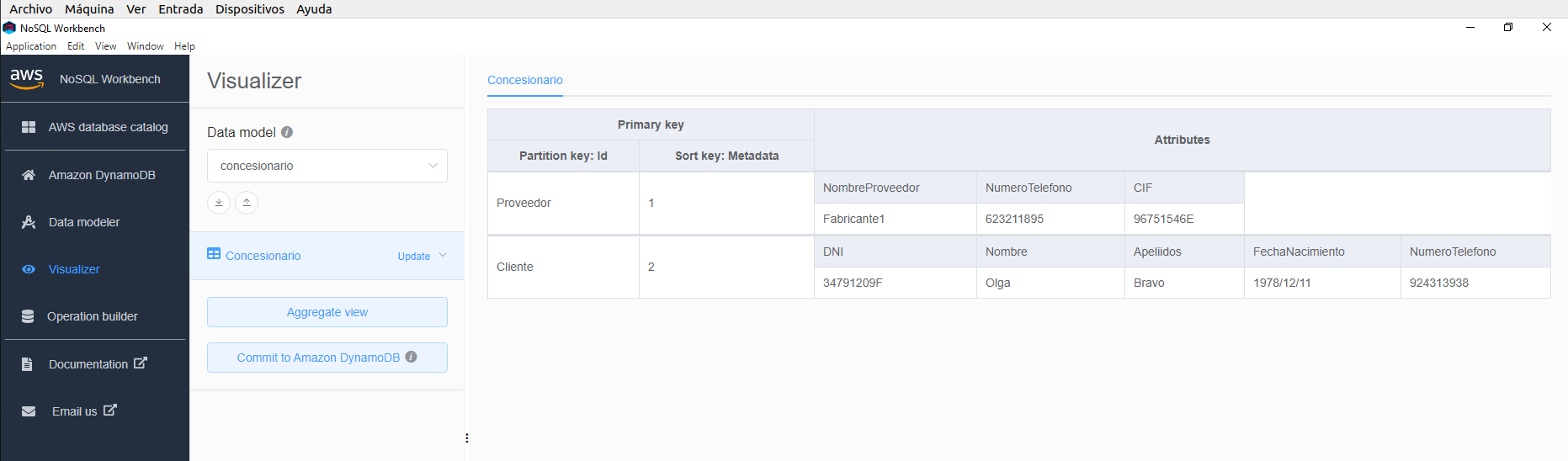
Para eliminar una tupla, en el modo edición, seleccionaremos la papelera de reciclaje que aparece a su derecha, tal y como se muestra a continuación:



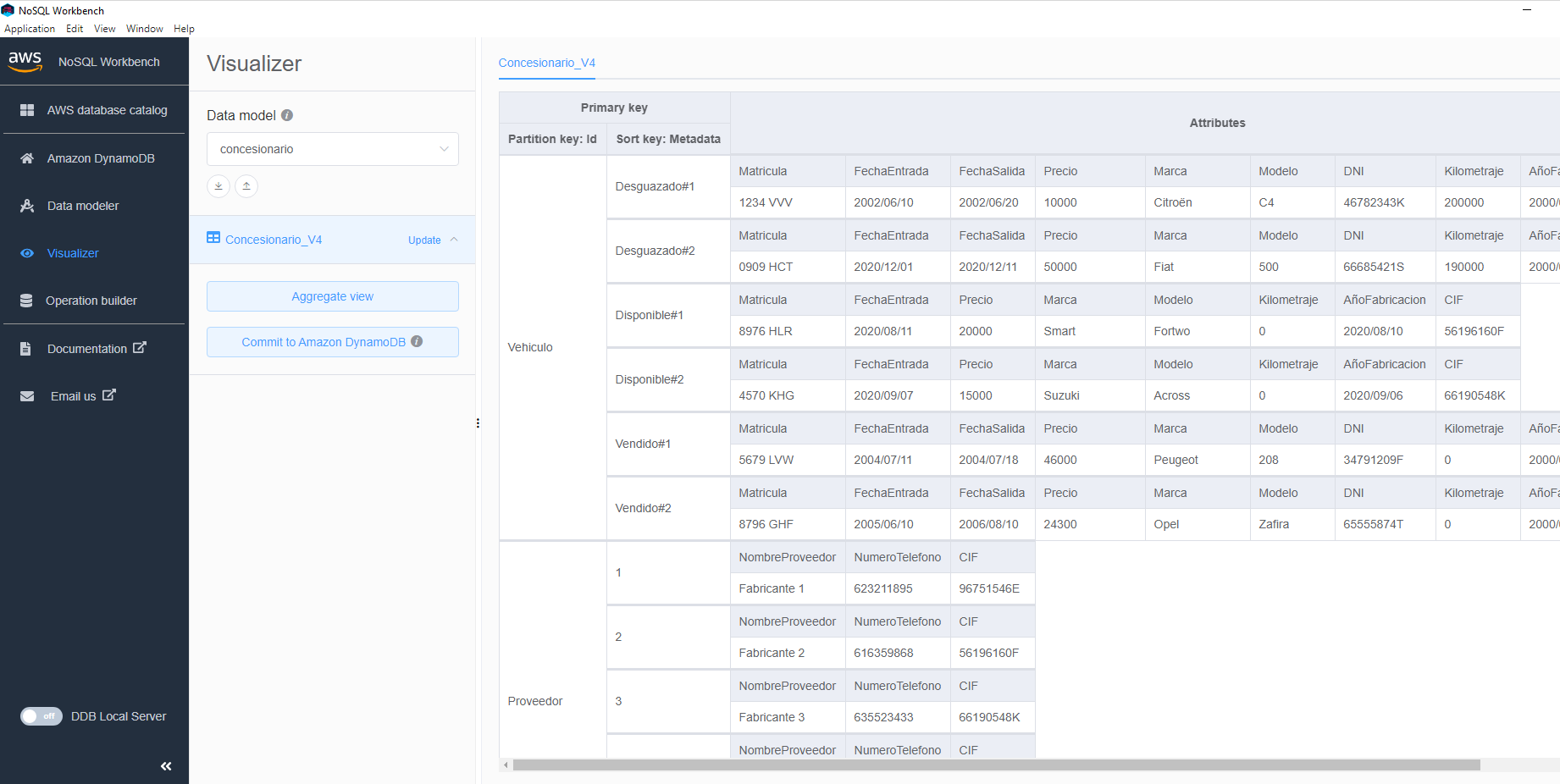
Como podemos ver a continuación, la tupla ha sido eliminada:

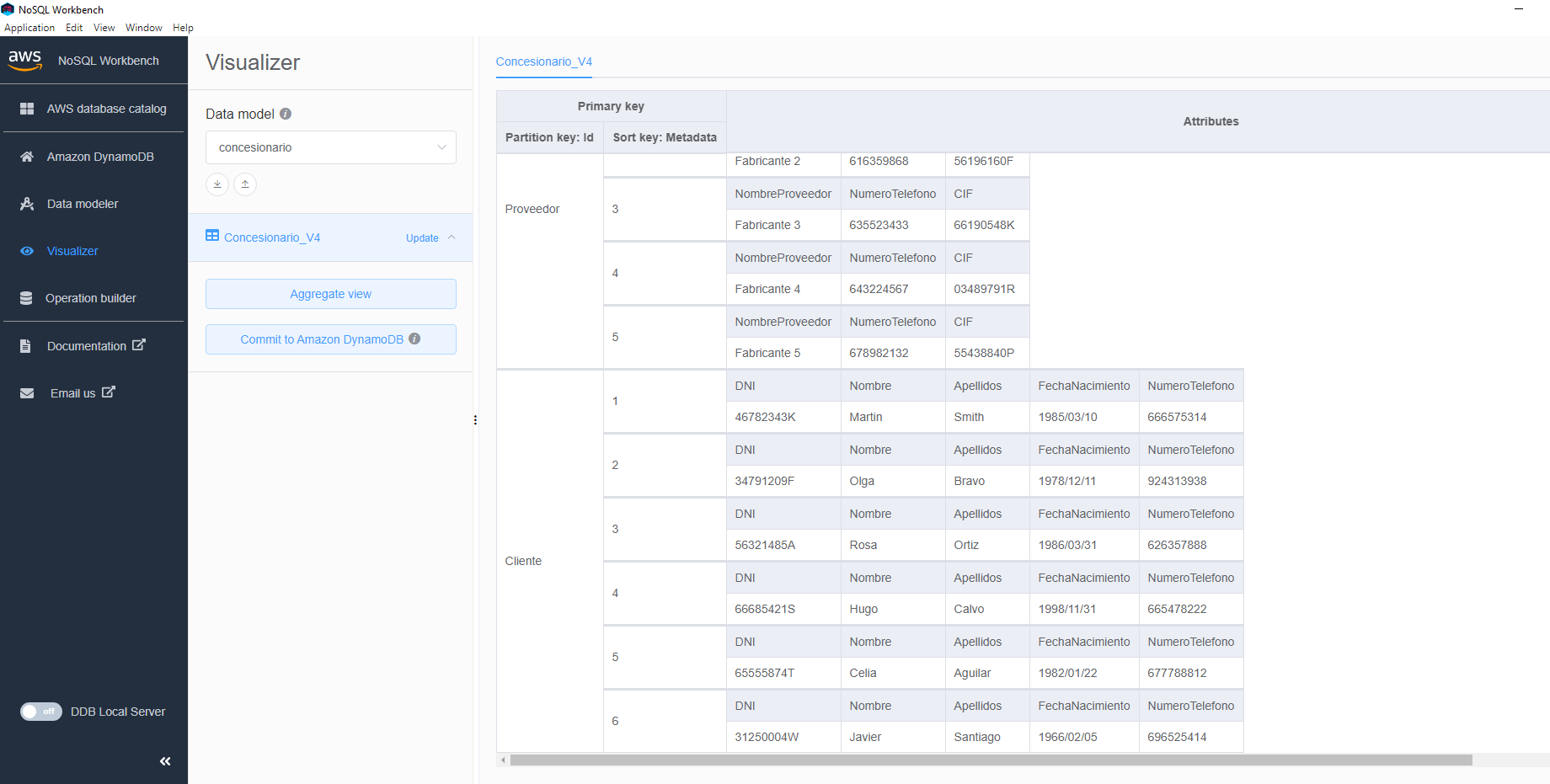


Tras salir del modo edición nos quedarían los siguientes registros:



La tabla, tras insertar unos pocos registros, nos ha quedado así:





Para guardar los cambios, tenemos que hacer click en el botón Commit to Amazon DynamoDB y seleccionar una conexión. Más adelante se detalla el mecanismo de conexión a la base de datos.

Nótese que, a pesar de que hemos podido crear y modificar esquemas con facilidad usando el IDE mencionado al principio de esta sección, no hemos podido borrar un esquema desde esta herramienta.

### Diseño del esquema implementado

El esquema anterior se ha realizado con la intención de almacenar toda la información relativa a un concesionario. Hay tres particiones: Vehiculo, Proveedor y Cliente. También tenemos una clave de ordenación.

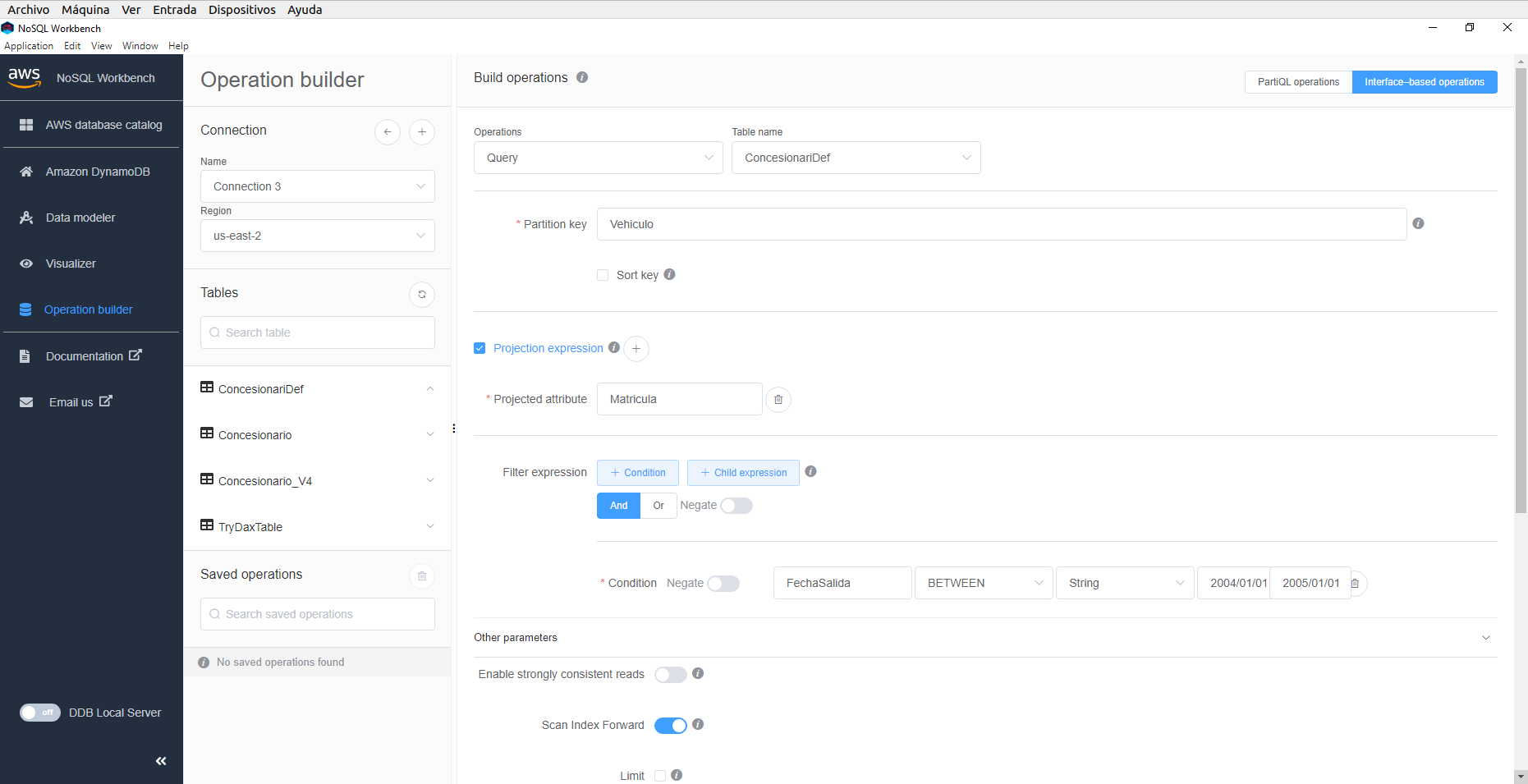
Además, se ha organizado de tal forma que se puede almacenar en un vehículo, el CIF del proveedor que nos lo suministró (relacionando así los vehículos con los proveedores). Por ejemplo, véase la tupla con Partition Key Vehiculo y Sort Key Disponible#1. De la misma forma, se puede almacenar información relativa a qué cliente compró un determinado vehículo, almacenando el DNI del cliente que lo compró en una tupla de la partición Vehiculo, relacionando así los vehículos del concesionario con los clientes del mismo. Por ejemplo, véase la tupla con Partition Key Vehiculo y Sort Key Vendido#1.

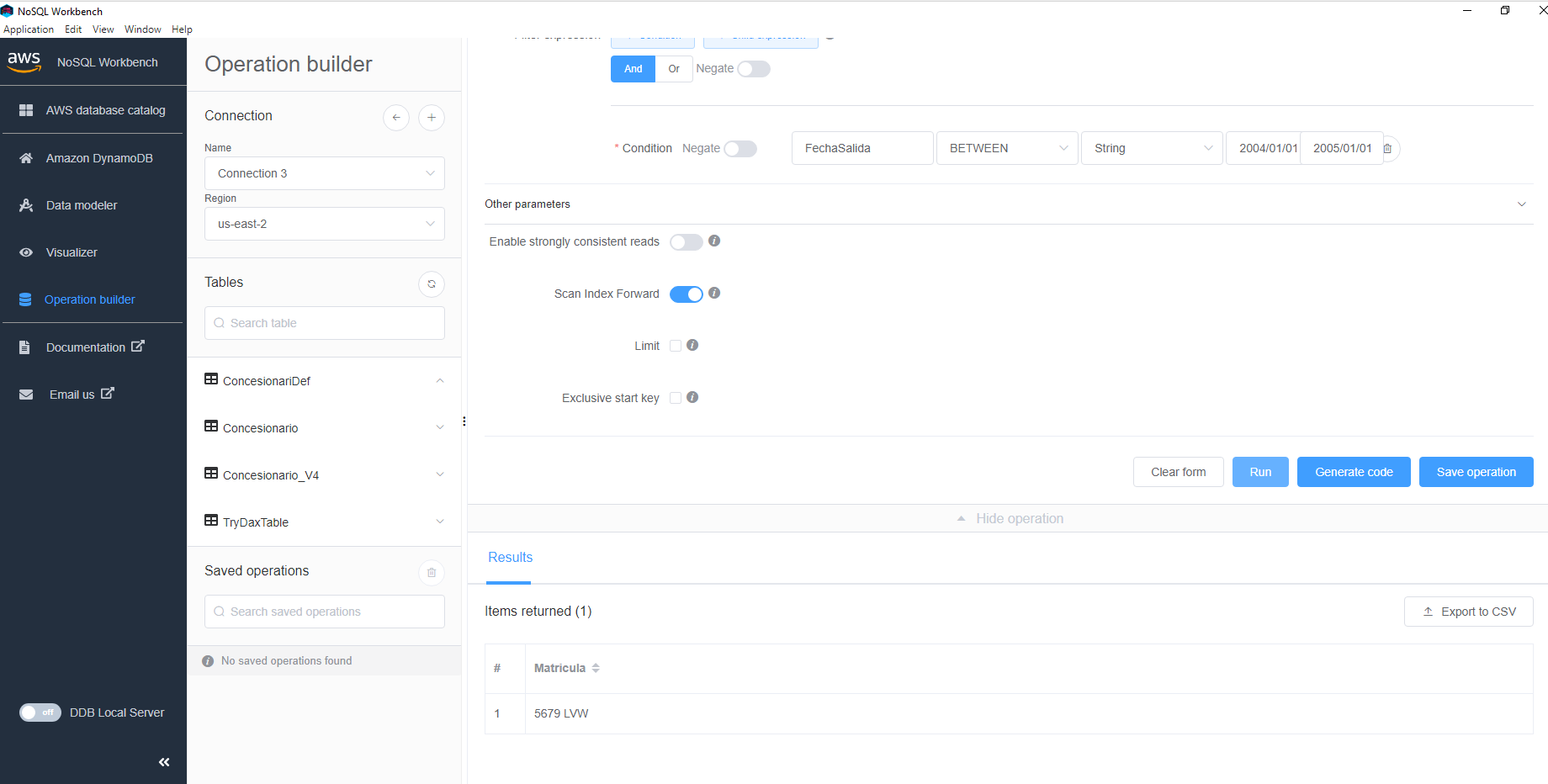
## Consultas

Con NoSQL y el SGBD NoSQL Workbench las consultas se vuelven muy sencillas.

### Coches Vendidos entre el 01/01/2004 y el 01/01/2005

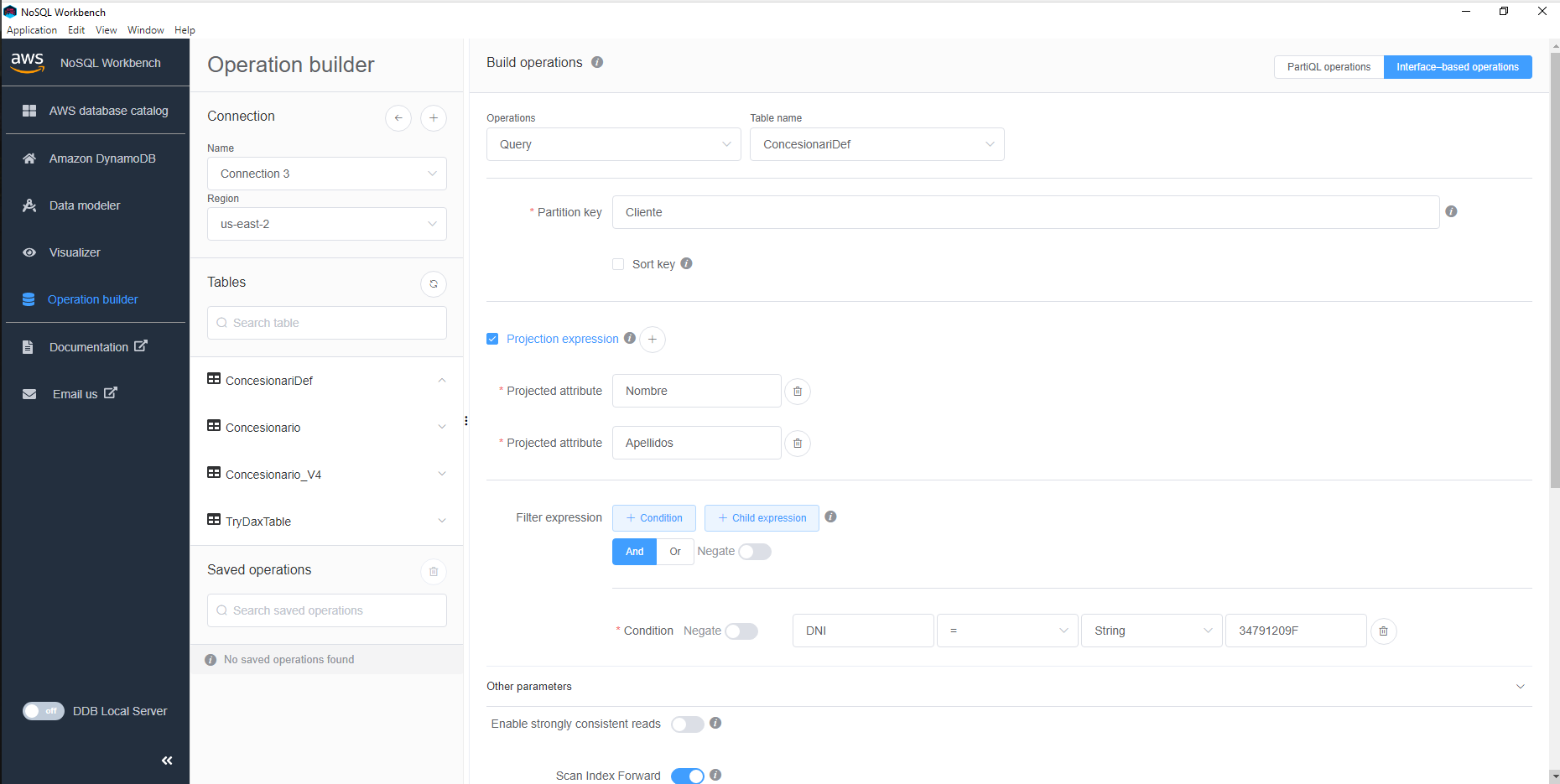
Para realizar una consulta tenemos que movernos a la pestaña Operation builder. Desde allí, solo tenemos que rellenar los diferentes campos para filtrar los datos de la tabla. En esta primera consulta hemos buscado dentro de la partición Vehiculo los coches vendidos entre el 01/01/2004 y el 01/01/2005 (hemos filtrado utilizando el atributo FechaSalida). Solo se mostrará la matrícula de los mismos (solo tenemos un Projected attribute).

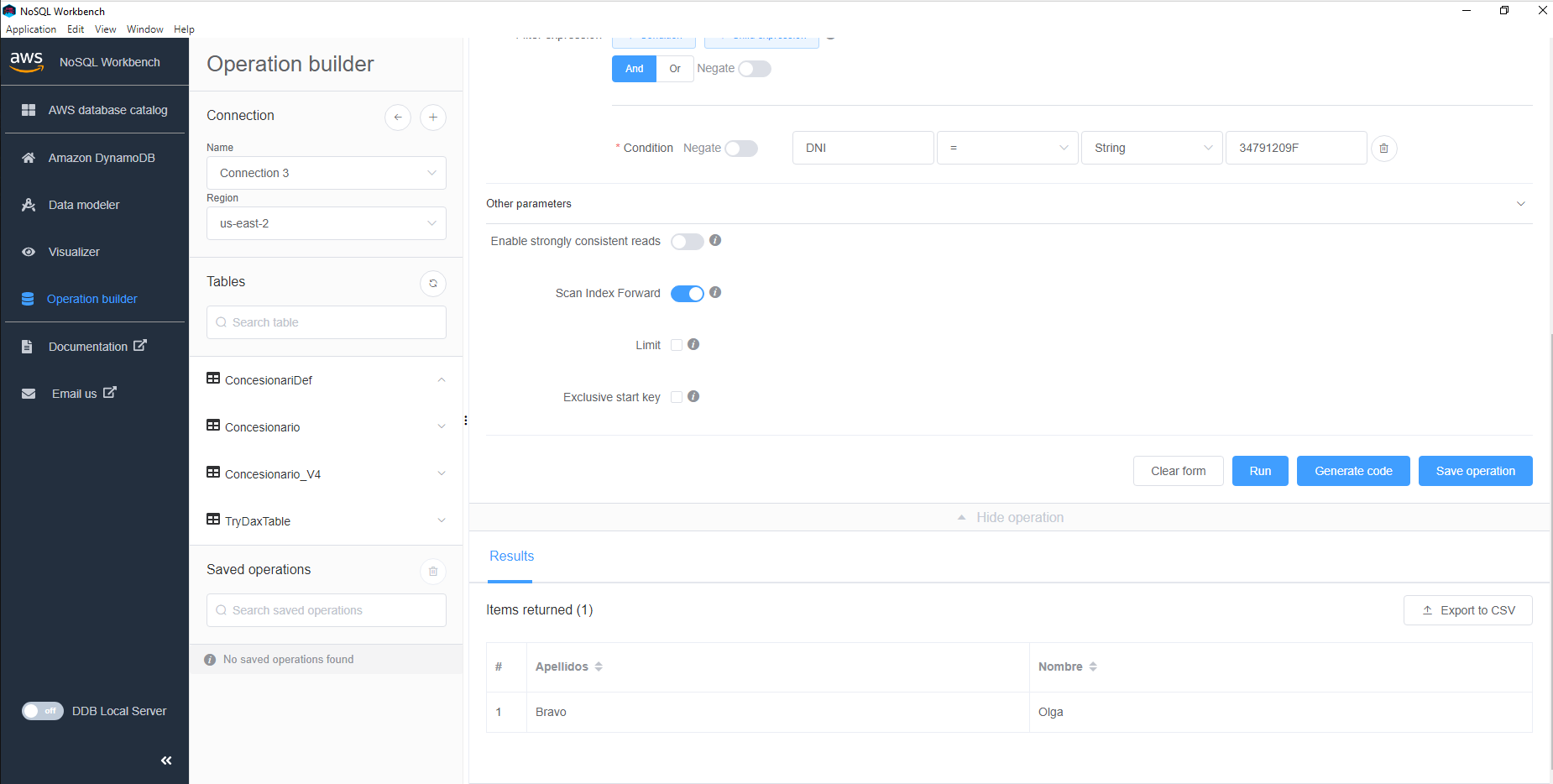




### Nombre y apellidos del cliente que con DNI ‘34791209F’

En esta segunda consulta hemos buscado dentro de la partición Cliente el nombre y los apellidos del cliente con DNI ‘34791209F’ (hemos filtrado utilizando el atributo DNI). Solo se mostrará el nombre y los apellidos del mismo (tenemos dos Projected attributes: Nombre y Apellidos).

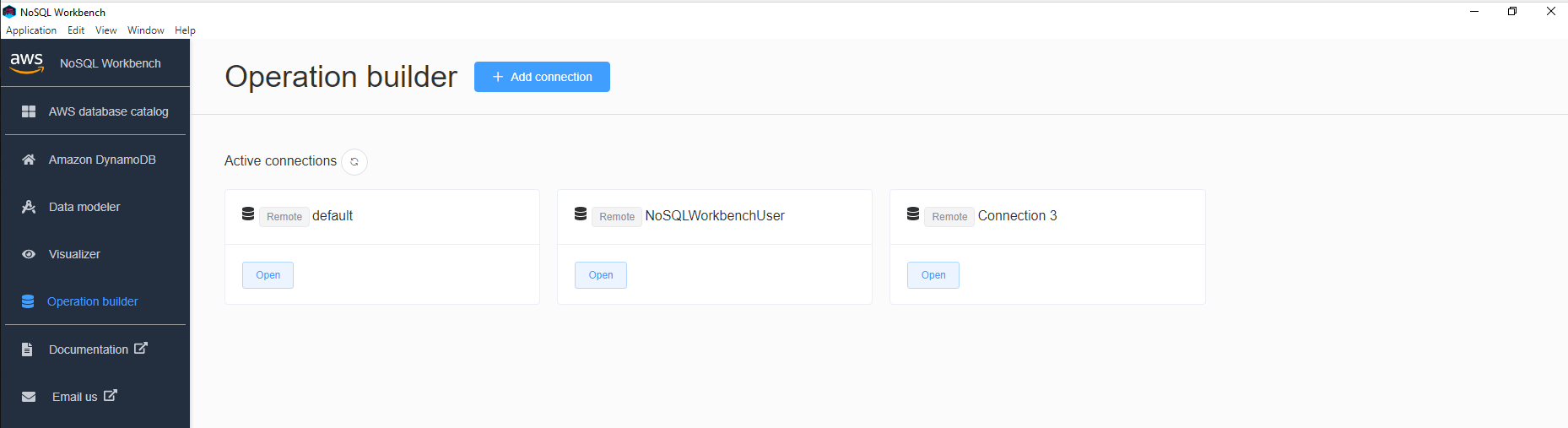




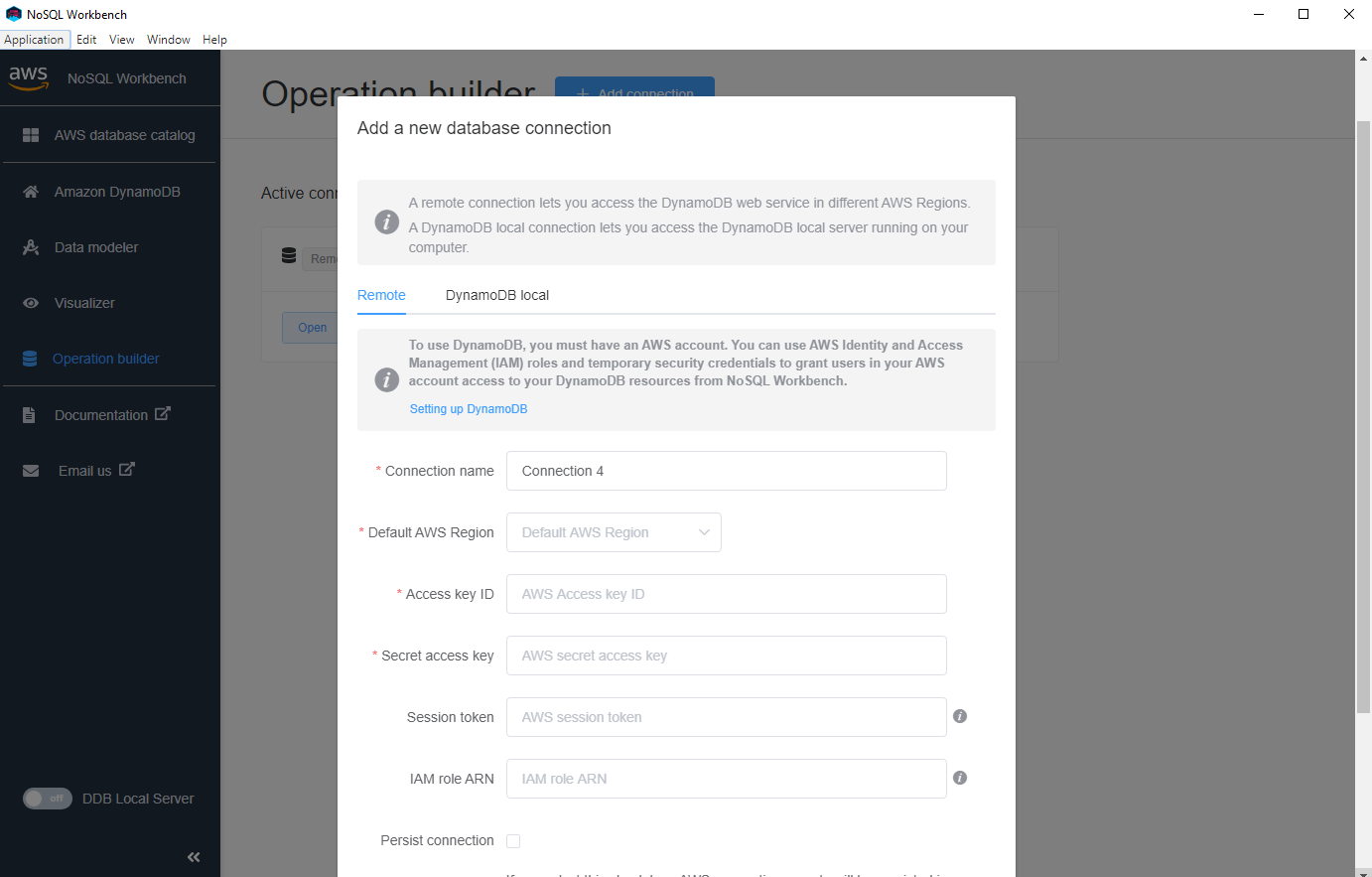
## Descripción del mecanismo de conexión.

José Luis Guidú Navas tenía cuenta en AWS, y a través de su cuenta nos concedió unas subcuentas asociadas a las suya para que pudiéramos utilizar AWS. Una vez hecho esto, introduciendo las credenciales que nos proporcionó pudimos iniciar sesión.

Para conectarnos a la base de datos dentro de DynamoDB tenemos que irnos a las pestaña Operation builder. Allí se muestran las conexiones que tenemos creadas.



Para crear una nueva conexión haremos click en el botón Add connection y se nos abrirá la siguiente ventana emergente:



Solo tendremos que rellenar los datos de la conexión que han generado desde la cuenta.

## Discusión sobre si sería adecuado para implementar el SI de la práctica

Toda la implementación del sistema de información realizada en la práctica 3 con un servicio de base de datos SQL puede implementarse usando DynamoDB.

Sin embargo, como nuestra aplicación en principio no requiere de una alta tasa de consulta y actualización, ni tampoco de una alta disponibilidad, no veo necesario el uso de un servicio de base de datos NoSQL como es DynamoDB.

Así pues, no vemos adecuado implementar el sistema de información de la práctica con DynamoDB.