



Tutorial: MySQL Setup on Docker and using it with Link and Domino Volt

Martin Lechleider

Contents

MySQL Setup.....	2
Download MySQL Docker Image.....	2
Install / Configure.....	2
User and Database Configuration.....	3
Create Sample Database with 2 Tables.....	3
Useful Database Commands.....	4
Useful Docker Commands.....	4
Technote for JDBC Driver.....	5
Create a MySQL Connection and Actions in Link.....	5
Create Connection.....	5
Create an Action to Add a Record to the Customer Table.....	7
Create an Action to Update a Record to the Customer Table.....	8
Create an Action to Read all Records from the Customer Table.....	9
Use the Actions in Domino Volt.....	10
Using the List Customers Action.....	10
Using the Add Customer Action.....	11
Using the Update Customer Action.....	13
Add Actions for the Orders Table.....	16
Use the Orders Actions in Domino Volt.....	17
Add Action to Join Customers and Orders.....	18
Use Join Action in Domino Volt.....	18

MySQL Setup

Download MySQL Docker Image

```
docker pull mysql/mysql-server
```

```
root@hip-VirtualBox:/home/hip# docker pull mysql/mysql-server
Using default tag: latest
latest: Pulling from mysql/mysql-server
c7127dfa6d78: Pull complete
530b30ab10d9: Pull complete
59c6388c2493: Pull complete
cca3f8362bb0: Pull complete
Digest: sha256:7cd104d6ff11f7e6a16087f88b1ce538bcb0126c048a60cd28632e7cf3dbe1b7
Status: Downloaded newer image for mysql/mysql-server:latest
docker.io/mysql/mysql-server:latest
root@hip-VirtualBox:/home/hip#
```

Install / Configure

```
docker run --name=mysql1 -p 3306:3306 -d mysql/mysql-server
```

Get the initial Root Password:

```
docker logs mysql1 2>&1 | grep GENERATED
```

```
docker exec -it mysql1 mysql -uroot -p <Hit Enter>
```

Type in initial root password

```
root@hip-VirtualBox:/home/hip# docker run --name=mysql1 -p 3306:3306 -d mysql/mysql-server
27409452ea57ef8be4b61d83cf51973611ee340095bc034c86c6dbe37276df89
root@hip-VirtualBox:/home/hip# docker logs mysql1 2>&1 | grep GENERATED
[Entrypoint] GENERATED ROOT PASSWORD: ic$4H94n0M70wxyfAJubmUp1eDA
root@hip-VirtualBox:/home/hip# docker exec -it mysql1 mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.19

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

User and Database Configuration

Change root password:

```
ALTER USER 'root'@'localhost' IDENTIFIED BY 'DBAdmin#1';
```

Create Demo database:

```
CREATE DATABASE Demo;
```

Create DB user:

```
CREATE USER 'dbuser'@'%' IDENTIFIED BY 'dbuser';
```

```
GRANT ALL PRIVILEGES ON *.* TO 'dbuser'@'%';
```

```
ALTER USER 'dbuser'@'%' IDENTIFIED WITH mysql_native_password BY 'dbuser';
```

Type exit to get to the command prompt

Type docker restart mysql1

To bring up the MySQL CLI with new user created - dbuser

```
docker exec -it mysql1 mysql -u dbuser -p
```

type the password - dbuser

Create Sample Database with 2 Tables

```
CREATE DATABASE Demo; ← Already created from above
```

Set Demo as the database to use:

```
Use Demo;
```

Create Customer Table:

```
CREATE TABLE `Customers` (  
  `CustomerID` int NOT NULL AUTO_INCREMENT,  
  `CustomerName` varchar(255),  
  `ContactName` varchar(255),  
  `Address` varchar(255),  
  `City` varchar(255),  
  `PostalCode` varchar(255),  
  `Country` varchar(255),  
  PRIMARY KEY (`CustomerID`)  
);
```

Add a record to Customers

```
INSERT INTO Customers (CustomerName, ContactName, Address, City, PostalCode, Country) VALUES ("Cardinal", "Tom B. Erichsen", "Skagen 21", "Stavanger", "4006", "Norway");
```

Create Orders Table:

```
CREATE TABLE `Orders` (  
  `OrderID` int NOT NULL AUTO_INCREMENT,  
  `CustomerID` int DEFAULT NULL,  
  `EmployeeID` int DEFAULT NULL,  
  `OrderDate` date DEFAULT NULL,  
  `ShipperID` int DEFAULT NULL,  
  PRIMARY KEY (`OrderID`)  
);
```

Add a record to Orders

```
INSERT INTO Orders (CustomerID, EmployeeID, OrderDate, ShipperID) VALUES ("92", "1234", "2021-09-20", "5678");
```

Useful Database Commands

To Set db user PW:

```
ALTER USER 'dbuser'@'%' IDENTIFIED WITH mysql_native_password BY 'dbuser';
```

To check the users:

```
select host, user from mysql.user;
```

Drop a user:

```
DROP USER 'test'@'localhost';
```

Useful Docker Commands

```
docker start mysql1  
docker restart mysql1  
docker stop mysql1  
docker rm mysql1  
docker logs mysql
```

To bring up MySQL CLI with dbuser:

```
docker exec -it mysql1 mysql -u dbuser -p
```

Get IP of the database:

```
docker inspect --format '{{ .NetworkSettings.IPAddress }}' mysql1
```

Technote for JDBC Driver

[download the JDBC driver \(mysql-connector-java-8.0.19.jar\)](#)

The JDBC driver should be placed in the /hiplib directory.

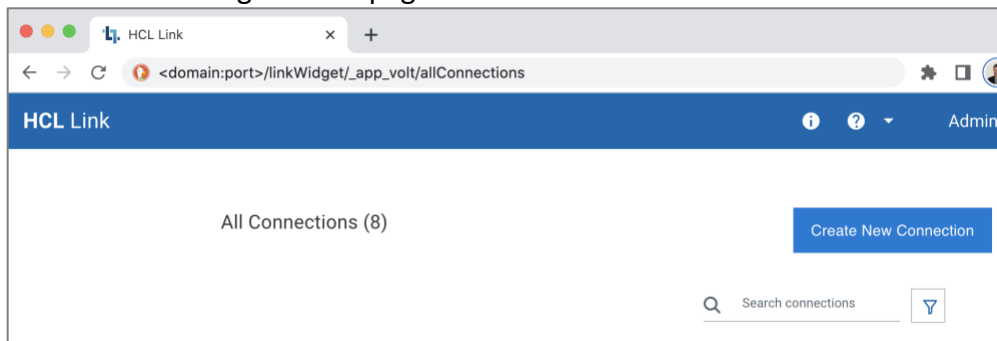
You may find that you still receive the following error message at runtime "No suitable driver detected for the specified connection URL". This is a known issue and the workaround can be found here:

https://support.hcltechsw.com/csm/en?id=kb_article&sysparm_article=KB0093913

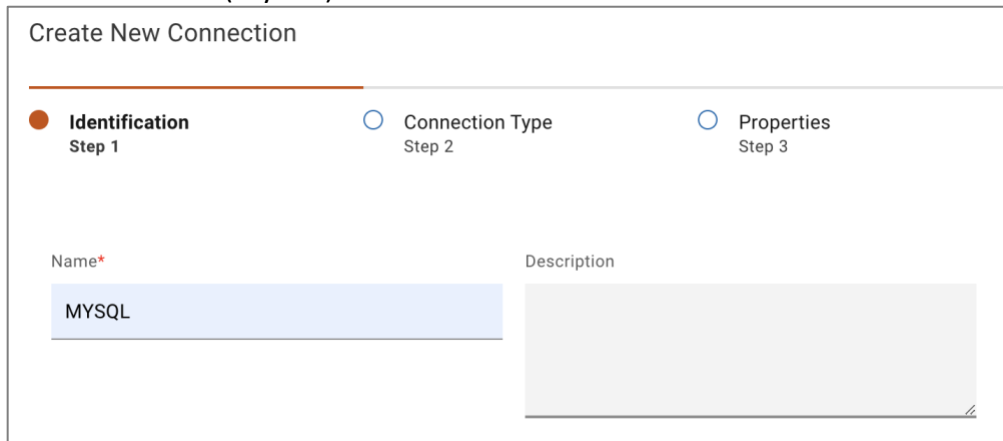
Create a MySQL Connection and Actions in Link

Create Connection

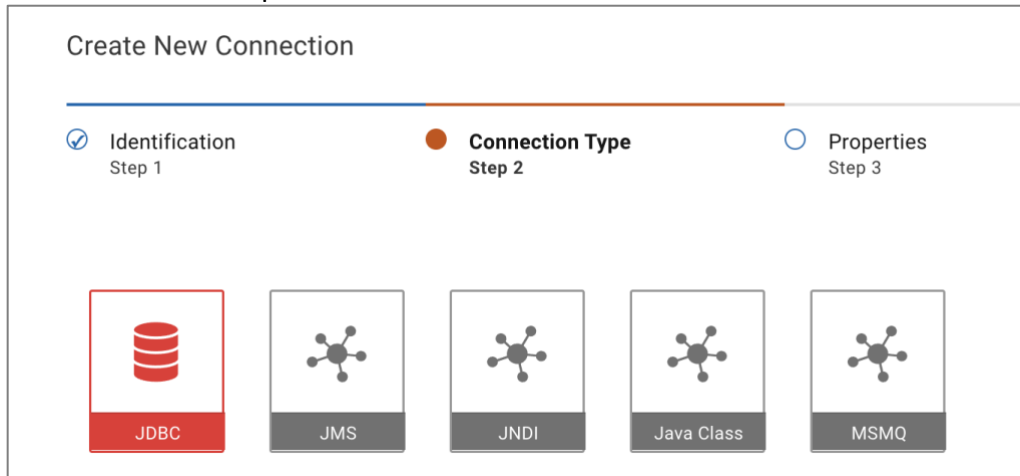
From the Link Widget home page click Create Connection



Give it a name – (MySQL)

A screenshot of the 'Create New Connection' form in HCL Link. The form is titled 'Create New Connection' and has three steps: 'Identification Step 1' (selected), 'Connection Type Step 2', and 'Properties Step 3'. Under 'Identification Step 1', there are two input fields: 'Name*' with the value 'MYSQL' and 'Description' which is empty.

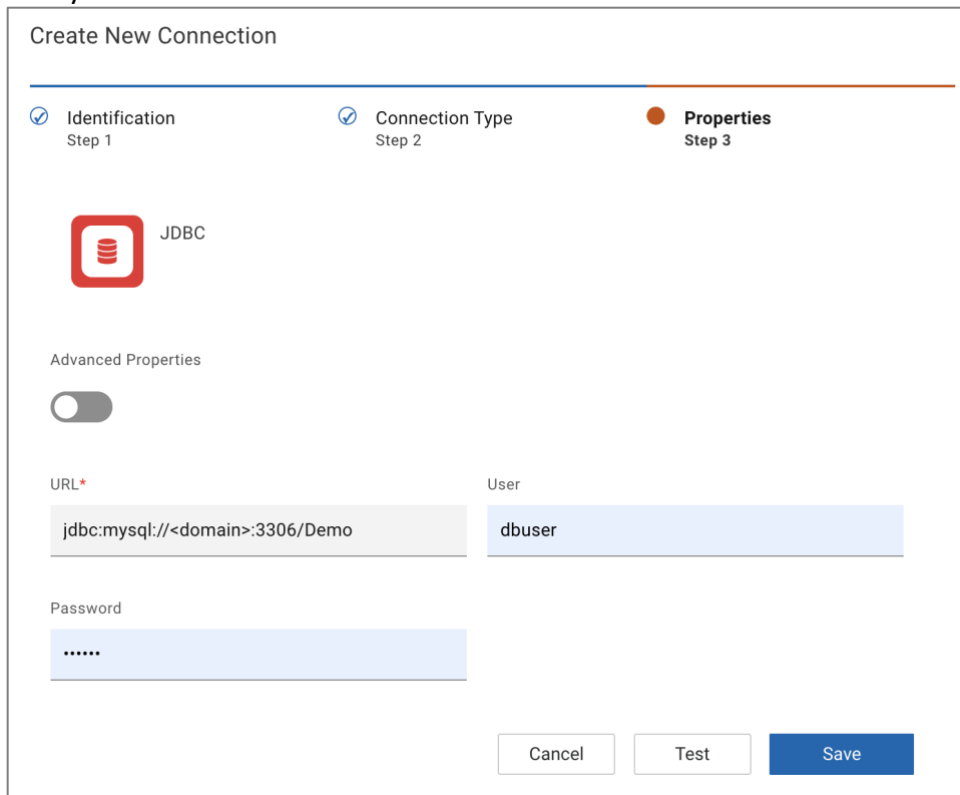
Select the JDBC adapter



Enter the MySQL URL, Username and Password – use the name/password previously setup (*dbuser/dbuser*)

The format of the URL is *jdbc:mysql://<domain>:3306/Demo* – where <domain> is the ip address or domain of your MySQL server. Demo is the name of the database previously setup. The port is always 3306. Make sure port 3306 is exposed outside the firewall if accessing it remotely.

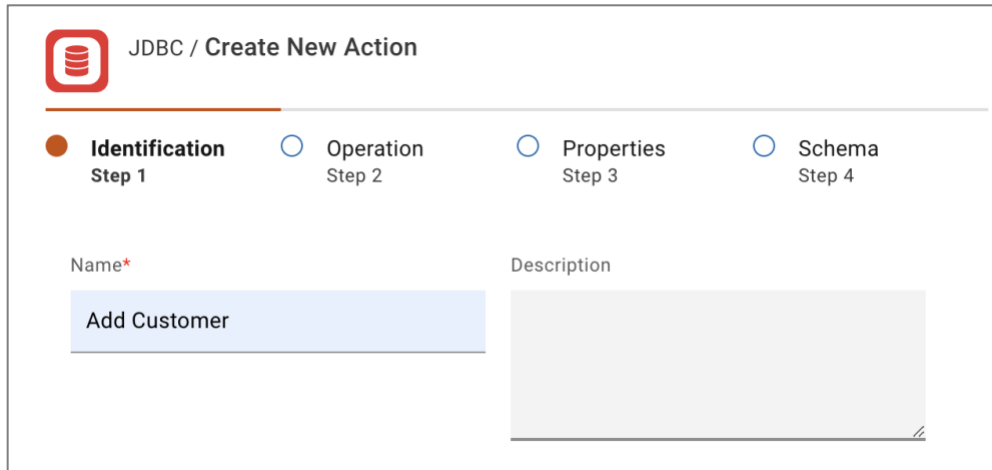
Test your connection and Save.



Create an Action to Add a Record to the Customer Table

On the Link Widget home page, click in the MySQL Connection and then click on Create New Action.

Give it a name – Add Customer.



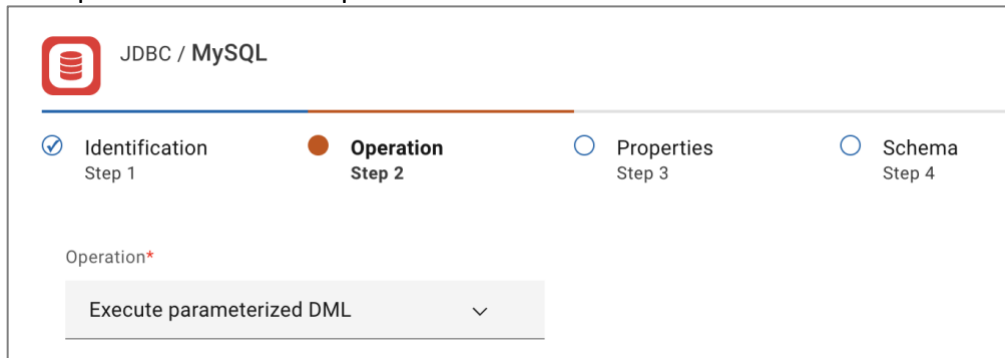
JDBC / Create New Action

● Identification Step 1 ○ Operation Step 2 ○ Properties Step 3 ○ Schema Step 4

Name*
Add Customer

Description

In step 2 select – Execute parameterized DML



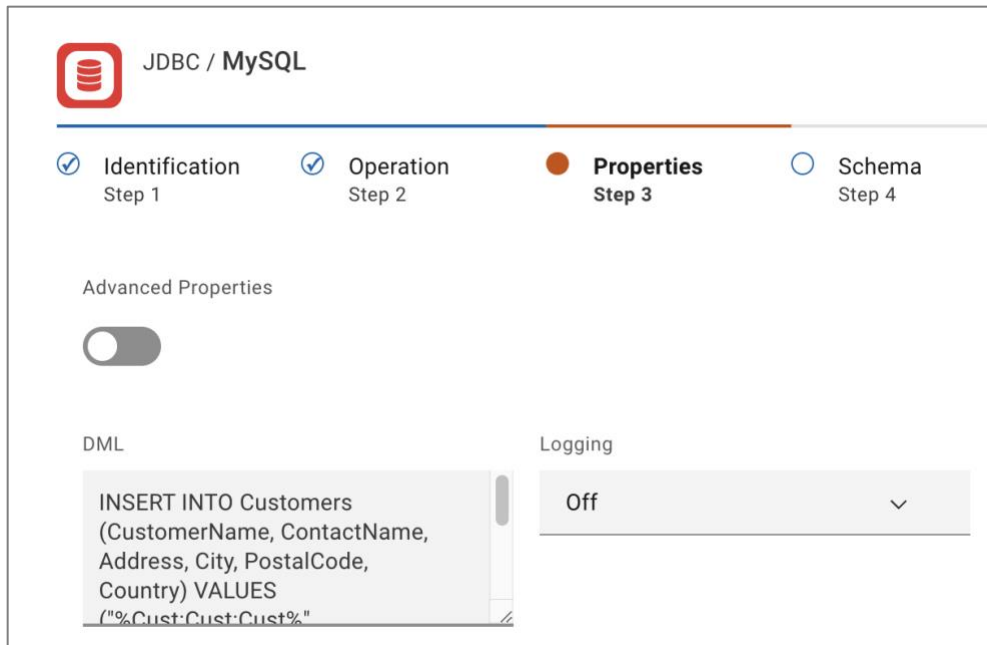
JDBC / MySQL

✔ Identification Step 1 ● Operation Step 2 ○ Properties Step 3 ○ Schema Step 4

Operation*
Execute parameterized DML

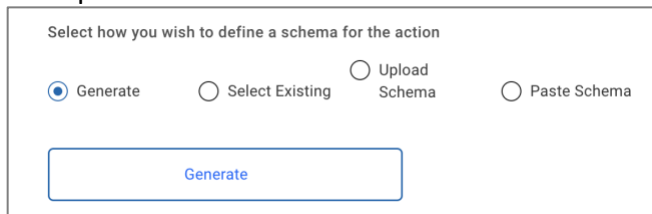
In step 3 – add the following SQL to the DML box:

```
INSERT INTO Demo.Customers (CustomerName, ContactName, Address, City, PostalCode, Country) VALUES ("%Cust:Cust:Cust%", "%Name:Name:Name%", "%Address:Address:Address%", "%City:City:City%", "%Postal:Postal:Postal%", "%Country:Country:Country%");
```

The values inside the % marks are parameters which you will pass to the Action. The format of the parameter is %name:default:description% - where name is input label, default is the default value, and description is the long name.

In step 4 – click Generate and then Save.

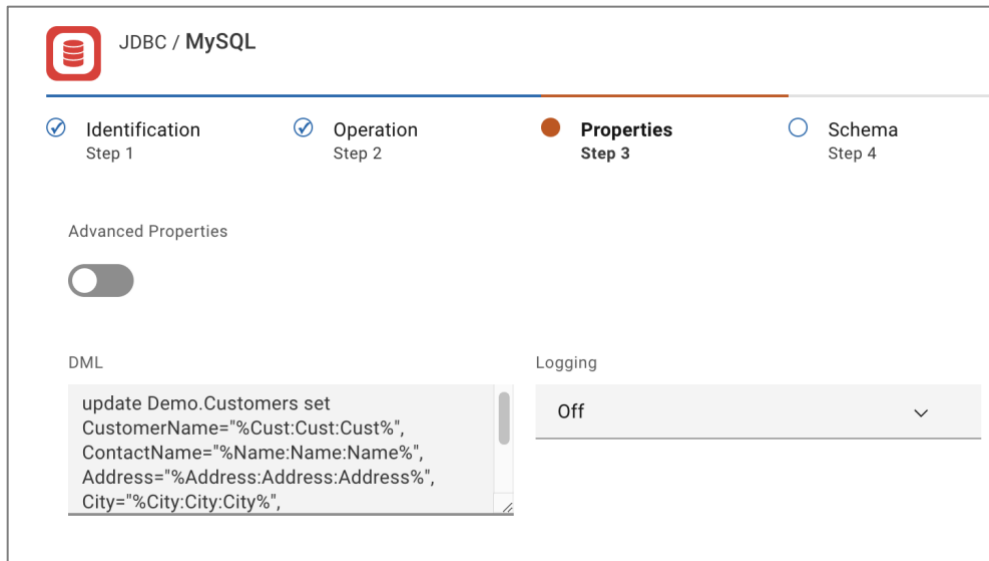


Create an Action to Update a Record to the Customer Table

Follow all the same step as the first Action, but in this case give it a name Update Customer in step 1, and use the following SQL in step 3:

```
UPDATE Demo.Customers SET CustomerName="%Cust:Cust:Cust%",
ContactName="%Name:Name:Name%", Address="%Address:Address:Address%",
City="%City:City:City%", PostalCode="%Postal:Postal:Postal%",
Country="%Country:Country:Country%" WHERE CustomerID="%Id:Id:Id%";
```

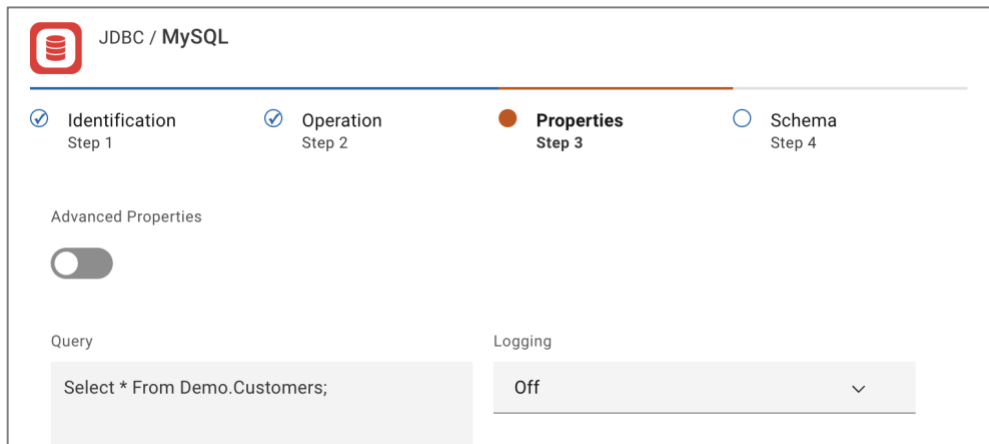
This SQL statement updates the record based on the customer id.



Create an Action to Read all Records from the Customer Table

Follow all the same step as the first Action, but in this case give it a name List Customers in step 1, select Execute query in step 2 and use the following SQL in step 3:

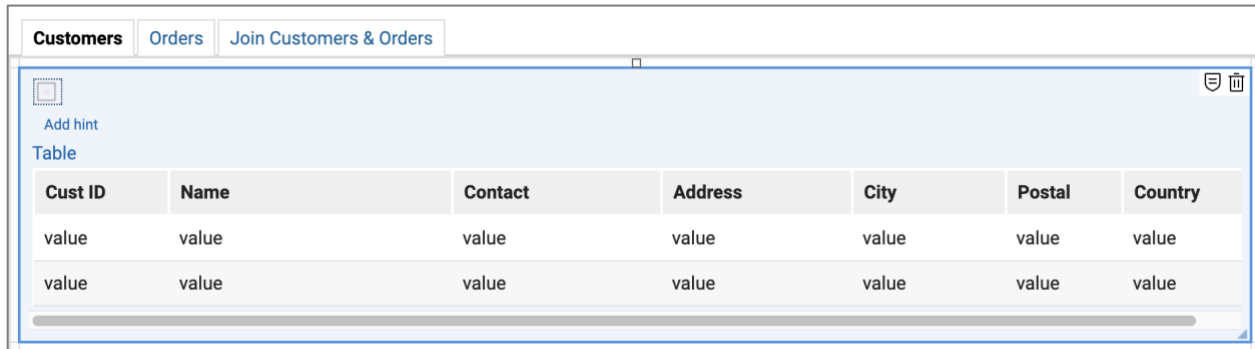
Select * From Demo.Customers;



Use the Actions in Domino Volt

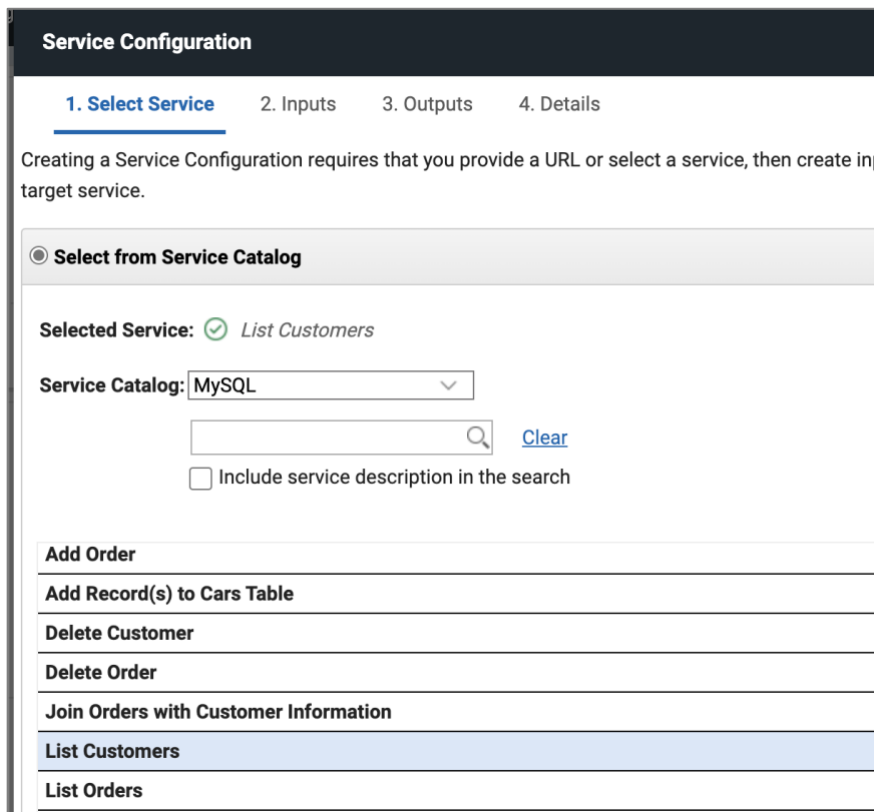
Using the List Customers Action

Create a Table with the Customer Fields



Cust ID	Name	Contact	Address	City	Postal	Country
value	value	value	value	value	value	value
value	value	value	value	value	value	value

Configure the List Customers services from the Service Catalog to populate the table on its onShow event.



Service Configuration

1. **Select Service** 2. Inputs 3. Outputs 4. Details

Creating a Service Configuration requires that you provide a URL or select a service, then create input and output target service.

Select from Service Catalog

Selected Service: *List Customers*

Service Catalog: MySQL

Include service description in the search

Add Order

Add Record(s) to Cars Table

Delete Customer

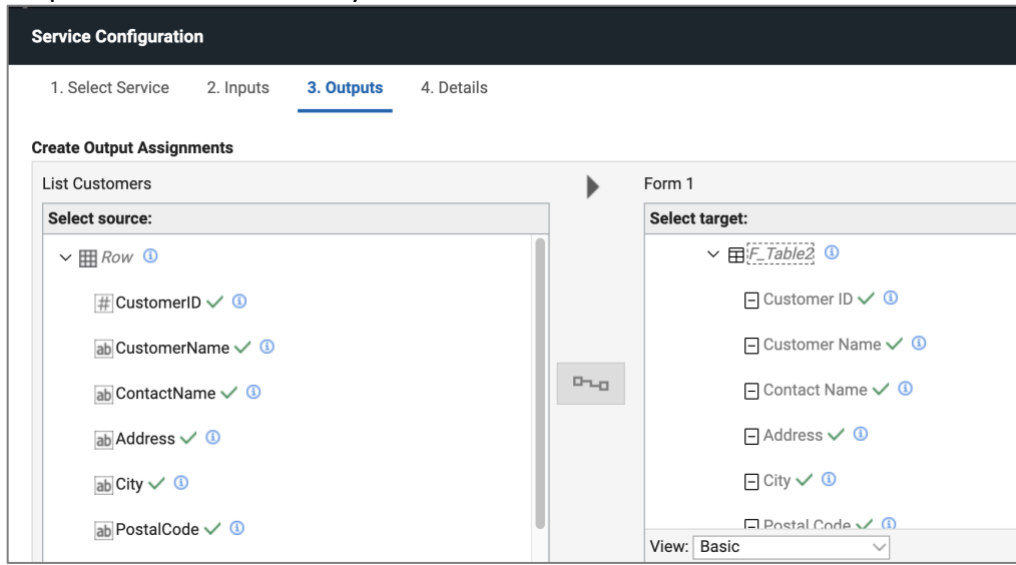
Delete Order

Join Orders with Customer Information

List Customers

List Orders

Map the fields from the MySQL table to the table in Domino Volt.



Preview or Save and launch the app to test

Customers						
Cust ID	Name	Contact	Address	City	Postal	Country
92	Cardinal	Tom B. Erichsen	Skagen 21	Stavanger	4006	Norway
93	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
94	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2223	México D.F.	05021	Mexico
95	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

Using the Add Customer Action

Create a form in Domino Volt to add a new record

[Update Customer](#) [Clear](#) [Add Customer](#) [Delete Customer](#)

Cust ID

Name

Address

Postal Code

Contact

City

Country

Add a Link or Button and use its onClick event to invoke the Add Customer service. Configure the Add Customer service from the Service Catalog to add the record.

Service Configuration

1. **Select Service** 2. Inputs 3. Details

Creating a Service Configuration requires that you provide a URL to a target service.

Select from Service Catalog

Selected Service: ✔ Add Customer

Service Catalog: ▼

🔍 [Clear](#)

Include service description in the search

Add Customer

Add Order

Map the form fields to the inputs for the MySQL record.

Note: you must map a table value to the DummyParameter which you will see when you expand Row. This is a temporary workaround to an issue in the software which will be fixed

Service Configuration

1. Select Service 2. **Inputs** 3. Details

Create Input Assignments

Form 1

Select source:

- Name ✔ ⓘ
- Contact ✔ ⓘ
- Address ✔ ⓘ
- City ✔ ⓘ
- Postal Code ✔ ⓘ
- Country ✔ ⓘ

View: ▼

Add Customer

Select target:

- > Row ⓘ
- ab Cust ✔ ⓘ
- ab Name ✔ ⓘ
- ab Address ✔ ⓘ
- ab City ✔ ⓘ
- ab Postal ✔ ⓘ
- ab Country ✔ ⓘ

Save test the app by adding some additional records.

Name	Contact	Address	City	Postal	Country
Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2223	México D.F.	5021	Mexico
Taquería	Antonio Moreno	Mataderos 2312	México D.F.	5023	Mexico
Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

Optional:

You may want to add a listener to the Add Customer onClick event to refresh the table when the Add Customer service has successfully completed. An example of the code to do so is below. The service and field IDs need to match those in your application.

Note: the code snippets to add a listener and refresh the table can be added by pressing Ctrl+Space twice in the JavaScript editor. Use Ctrl+Space once to pick the services to use in the snippet.

```
var srv = form.getServiceConfiguration("SC_AddCustomer");

srv.connectEvent("onCallFinished", function(success, errorObj) {
  if (success) {
    form.getServiceConfiguration("SC_ListCustomers").callService();
  }
});
```

Optional:

You may want to add a 'clear form' link or button to clear the form after a record is added. An example of the code to do this is below. The IDs need to match those in your app.

```
BO.F_CustID.setValue('');
BO.F_Name.setValue('');
BO.F_Contact.setValue('');
BO.F_Address.setValue('');
BO.F_City.setValue('');
BO.F_PostalCode.setValue('');
BO.F_Country.setValue('');
```

Using the Update Customer Action

The first step is to get the CustID and the values of the record to update in the form. To do this you can add some JavaScript to the table onClick event to pass the values from the selected table record to the form fields. An example of the code to do this is below. The IDs need to match those in your app.

```

var selected = item.getSelection();
if(selected === null)
    return;

BO.F_CustID.setValue(selected.F_Tbl_CustID.getValue());
BO.F_Name.setValue(selected.F_Tbl_Name.getValue());
BO.F_Contact.setValue(selected.F_Tbl_Contact.getValue());
BO.F_Address.setValue(selected.F_Tbl_Address.getValue());
BO.F_City.setValue(selected.F_Tbl_City.getValue());
BO.F_PostalCode.setValue(selected.F_Tbl_Postal.getValue());
BO.F_Country.setValue(selected.F_Tbl_Country.getValue());

```

Add a Link or Button and use its onClick event to invoke the Update Customer service. Configure the Update Customer service from the Service Catalog to update the record.

Service Configuration

1. **Select Service** 2. Inputs 3. Details

Creating a Service Configuration requires that you provide a URL or select target service.

Select from Service Catalog

Selected Service: Update Customer

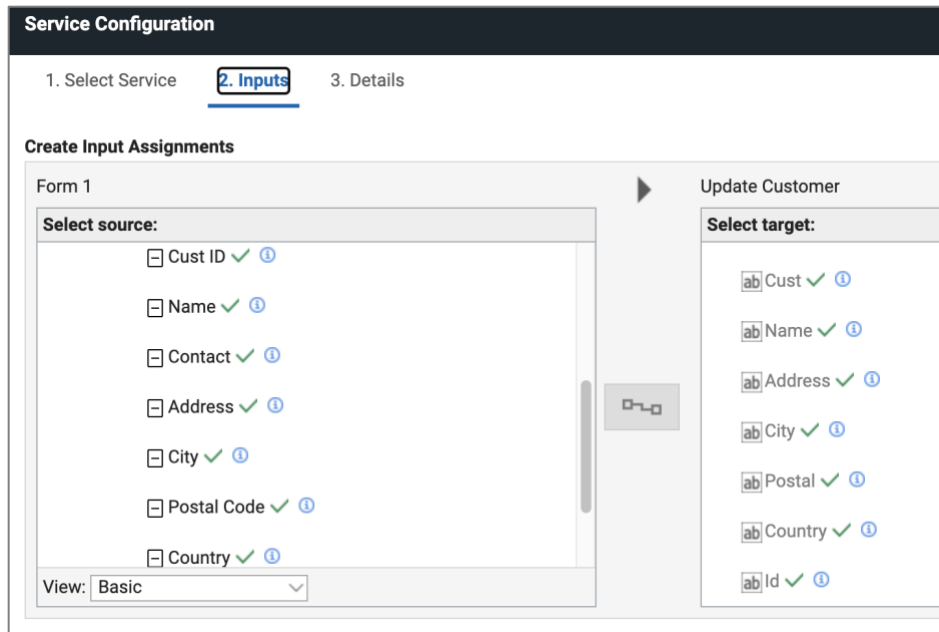
Service Catalog: MySQL

Include service description in the search

Update Cars Table
Update Customer
Update Order

Map the form fields to the inputs for the MySQL record. Make sure the CustID is included since tells the service which records to update.

Note: you must map a table value to the DummyParameter which you will see when you expand Row. This is a temporary workaround to an issue in the software which will be fixed



Optional:

You may want to add a listener to the Update Customer onClick event to refresh the table when the Update Customer service has successfully completed. An example of the code to do so is below. The service and field IDs need to match those in your application.

Note: the code snippets to add a listener and refresh the table can be added by pressing Ctrl+Space twice in the JavaScript editor. Use Ctrl+Space once to pick the services to use in the snippet.

```
var srv = form.getServiceConfiguration("SC_UpdateCustomer");

srv.connectEvent("onCallFinished", function(success, errorObj) {
  if (success) {
    form.getServiceConfiguration("SC_ListCustomers").callService();
  }
});
```


Save and test the app by selecting and then updating a record.

Customers						
Cust ID	Name	Contact	Address	City	Postal	Country
92	Cardinal Update	Tom B. Erichsen	Skagen 21	Stavanger	4006	Norway
93	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
94	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2223	México D.F.	05021	Mexico
95	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
96	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK

[Update Customer](#) [Clear](#) [Add Customer](#) [Delete Customer](#)

Cust ID

Name

Address

Contact

City

Add Actions for the Orders Table

Repeat the same steps use to add Link actions for the customers table. Create action to list the orders, add an order and update an order. Below are the SQL commands for each of the actions.

List Orders

Select * From Demo.Orders;

Add Order

```
INSERT INTO Demo.Orders (CustomerID, EmployeeID, OrderDate, ShipperID)
VALUES ("%CustID:CustID:CustID%", "%EmpID:EmpID:EmpID%", "%Date:Date:Date%",
"%ShipID:ShipID:ShipID%");
```

Update Order

```
Update Demo.Orders set CustomerID="%CustID:CustID:CustID%",
EmployeeID="%EmpID:EmpID:EmpID%", OrderDate="%Date:Date:Date%",
ShipperID="%ShipID:ShipID:ShipID%"
where OrderID="%OrderID:OrderID:OrderID%";
```

Use the Orders Actions in Domino Volt

Add a new page or tab to your Domino Volt app and repeat the same steps used to build the customers part of the app.

Order ID	Cust ID	Employee ID	Order Date	Shipper ID
11078	92	1234	2021-09-19	5678
11079	96	8235	2022-03-18	566
11080	93	36456	2022-03-21	4332
11081	94	22223	2022-03-15	4443
11082	97	22223	2022-01-10	4243

Update Order Clear Add Order Delete Order

Order ID	Cust ID
<input type="text" value="11079"/>	<input type="text" value="96"/>
Employee ID	Order Date
<input type="text" value="8235"/>	<input type="text" value="2022-03-18"/>
Shipper ID	
<input type="text" value="566"/>	

Save test the app by adding some additional records.

Order ID	Customer ID	Employee ID	Order Date	Shipper ID
11079	96	8235	2022-03-18	566
11080	93	36456	2022-03-21	4332
11081	94	22223	2022-03-15	4443
11082	97	22223	2022-01-10	4243

Add Action to Join Customers and Orders

This action will be a query and use the following SQL.

```
SELECT Orders.OrderID,Orders.OrderDate, Customers.CustomerName,  
Customers.ContactName, Customers.Country  
FROM Orders INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID;
```

This SQL will join the order id and order date with the customer name, contact and country.

Use Join Action in Domino Volt

Add a table to a new tab or page and add the fields depicted below to it.

Order ID	Order Date	Customer	Contact	Country
11078	2021-09-19	Cardinal	Tom B. Erichsen	Norway
11079	2022-03-18	Around the Horn	Thomas Hardy	UK
11080	2022-03-21	Alfreds Futterkiste	Maria Anders	Germany
11081	2022-03-15	Ana Trujillo Emparedados y helados	Ana Trujillo	Mexico
11082	2022-01-10	Berglunds snabbköp	Christina Berglund	Sweden

Add a service to the table onShow which invokes the Join service.

Service Configuration

1. **Select Service** 2. Inputs 3. Outputs 4. Details

Creating a Service Configuration requires that you provide a URL or select a service target service.

Select from Service Catalog

Selected Service: List Customer Orders

Service Catalog: MySQL

Include service description in the search

Add Order

Add Record(s) to Cars Table

Delete Customer

Delete Order

Join Orders with Customer Information

List Customers

Map the fields from the SQL join to the fields in the table.

