Lab - Manage instances of the IBM Bluemix Data Services

Version:	1.13
Last modification date:	2016/03/22
Owner:	IBM Ecosystem Development

Lab : Manage instances of the IBM Bluemix Data Services

Lab Objectives: *This lab will show you the basics of how to manage the following Data Services in IBM Bluemix*

- *Cloudant NoSQL Database*
- *dashDB*

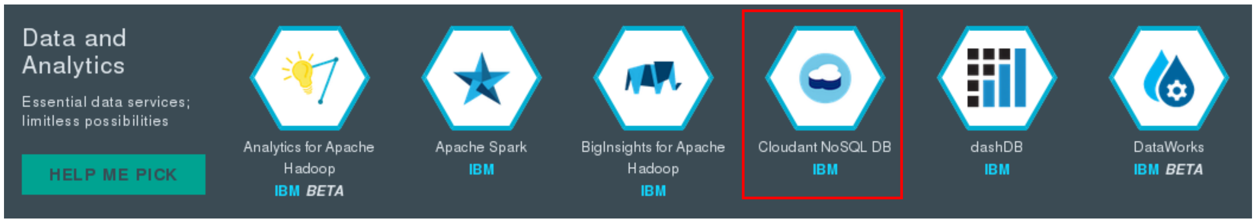
Lab Duration : 35 minutes

1. Manage instances of the Cloudant NoSQL Database Service

In this section you'll go through the basics of managing the Cloudant NoSQL Database service in IBM Bluemix.

1.1 Launch the Cloudant Dashboard

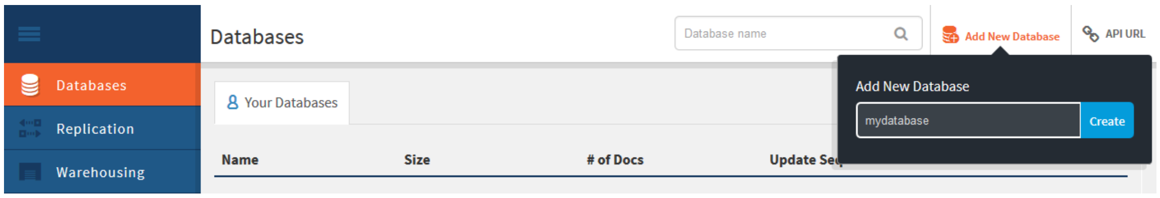
1. In your browser go to the Bluemix URL http://bluemix.net and login is necessary.
2. Make sure you're in the Dashboard tab (if not click on the Dashboard link at the top of the page to take you there)
3. Click on **USE SERVICES OR APIS**
4. Scroll down to the **Data and Analytics** section and click **Cloudant NoSQL DB.**


Data and Analytics
Essential data services; limitless possibilities
HELP ME PICK
Analytics for Apache Hadoop IBM BETA
Apache Spark IBM
BigInsights for Apache Hadoop IBM
Cloudant NoSQL DB IBM
dashDB IBM
DataWorks IBM BETA

5. Under **App** select **Leave unbound**
6. Click on **CREATE** to create a new instance of Cloudant NoSQL DB
7. Click **Launch when the service** landing page appears to launch the dashboard

1.2 Create a database

1. From the dashboard click on **Add New Database**, enter `mydatabase` as the name and click **Create**

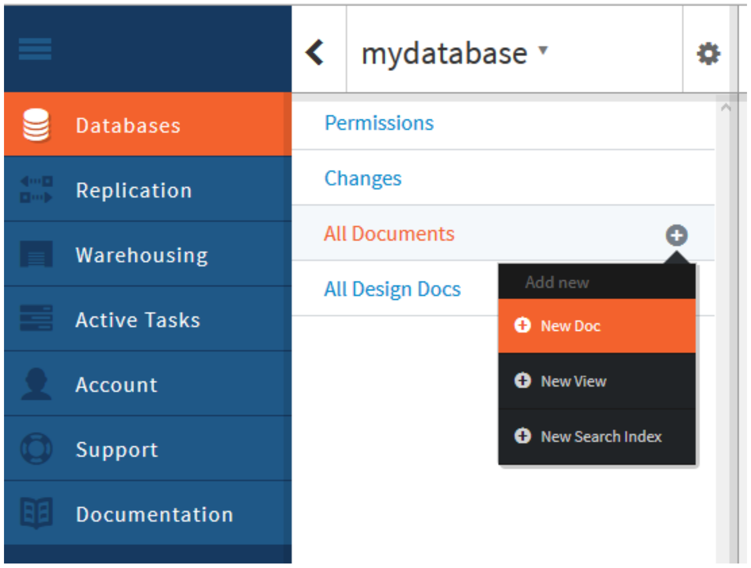


2. You'll be taken to the administration screen for the new database

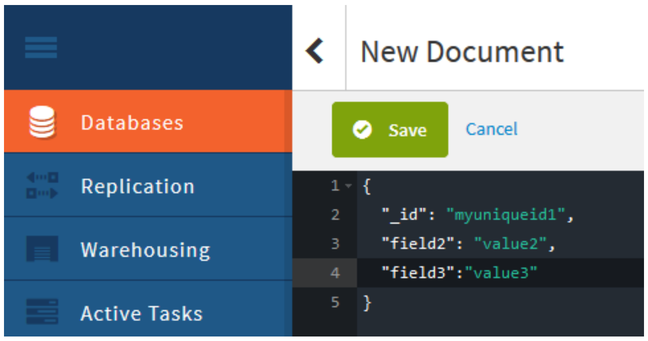
1.3 Add data to an existing database

1. Click on the + icon next to **All Documents** and select **New Doc** from the context menu

2. A new JSON document will appear with a single attribute named **_id** . This is the unique identifier



of your new document. Modify the **_id** and add the fields **field2** and **field3** as shown below

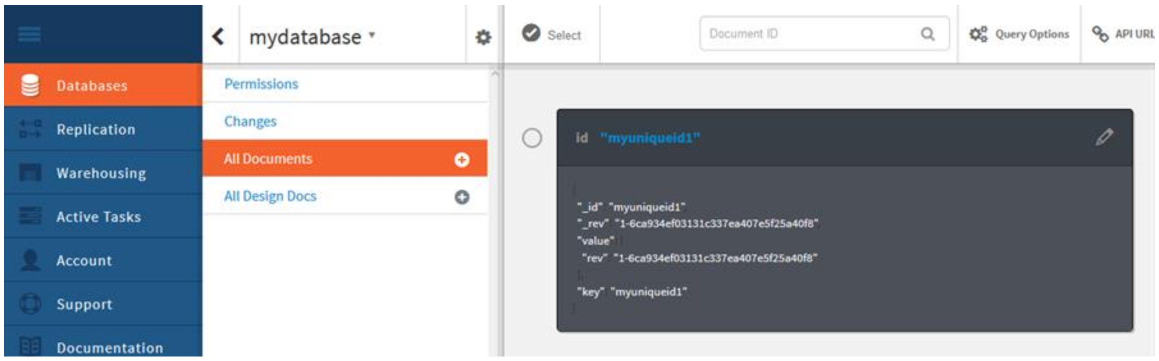


```
{
  "_id": "myuniqueid1",
  "field2": "value2",
  "field3":"value3"
}
```

3. Click **Save** to save the changes and return to the database view

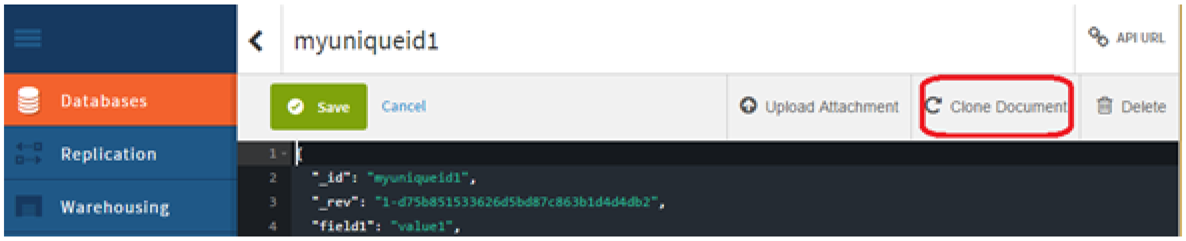
1.4 Clone documents in a database

1. From the database view in the dashboard click on **All Documents**, a summary of the documents in the database will appear at the right



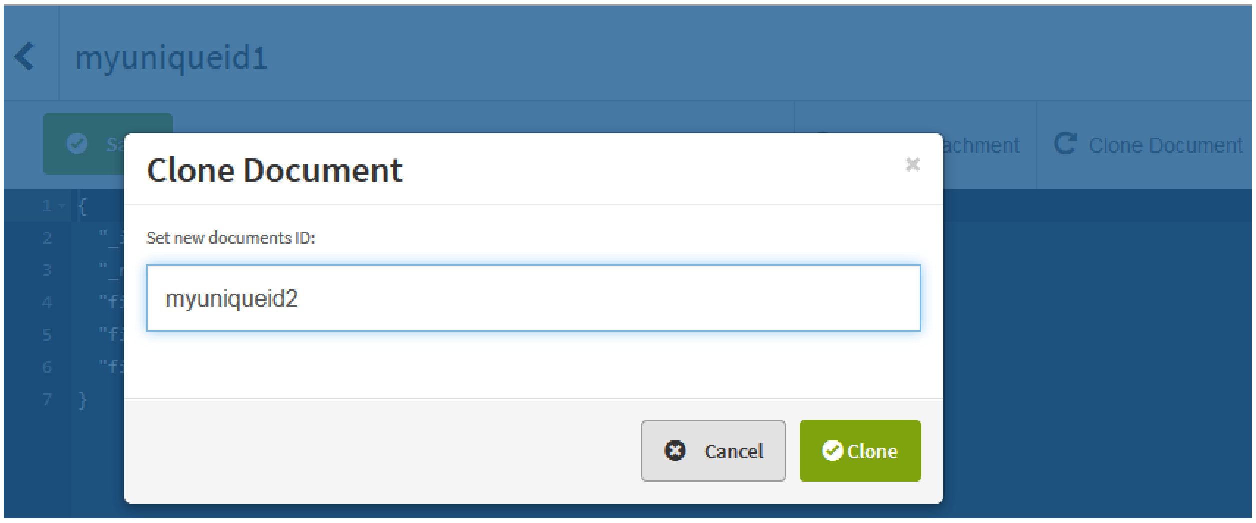
```
id "myuniqueid1"
{
"_id" "myuniqueid1"
"_rev" "1-6ca934ef03131c337ea407e5f25a40f8"
"value" {
 "rev" "1-6ca934ef03131c337ea407e5f25a40f8"
},
"key" "myuniqueid1"
}
```

2. Click on the pencil icon of your only document to edit it.
3. Click on **Clone Document** in the document editor



```
{
  "_id": "myuniqueid1",
  "_rev": "1-d75b851533626d5bd87c863b1d4d4db2",
  "field1": "value1",
```

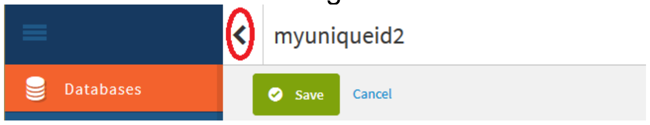
4. You'll be prompted to accept a system generated unique id for the new clone or to provide your own value. Change the id to `myuniqueid2`



Clone Document

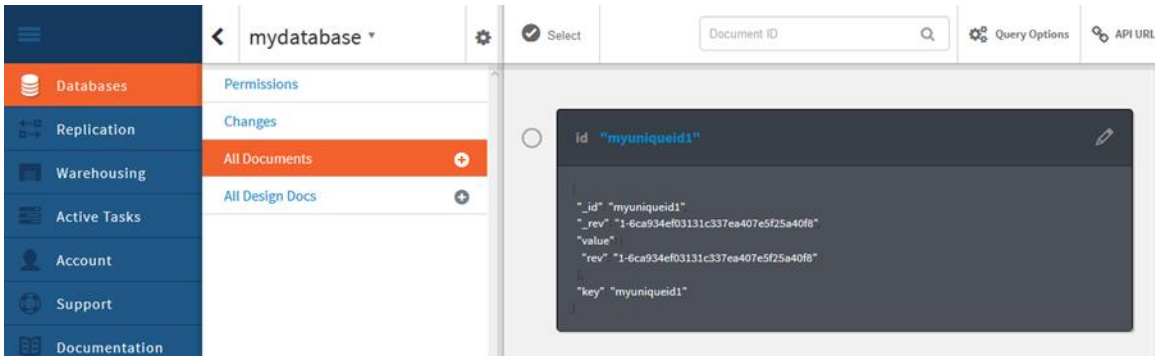
Set new documents ID: myuniqueid2

5. Click **Clone**. Your clone is added to the database.
6. Click on the arrow icon to go back to the database view



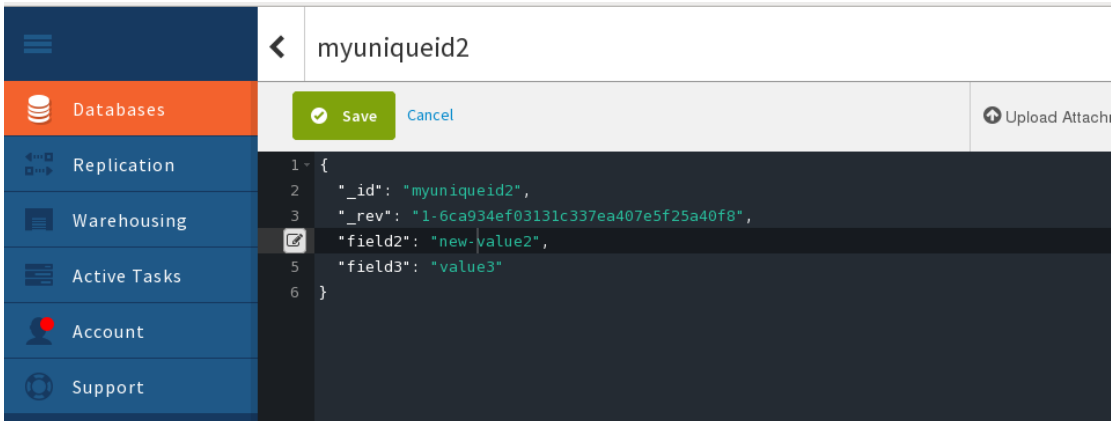
1.5 Edit documents in a database

1. From the database view in the dashboard click on **All Documents**, a summary of the documents in the database will appear at the right



```
{
"_id" "myuniqueid1"
"_rev" "1-6ca934ef03131c337ea407e5f25a40f8"
"value" {
 "rev" "1-6ca934ef03131c337ea407e5f25a40f8"
},
"key" "myuniqueid1"
}
```

2. Click on the pencil icon of your document with the **_id** of “myuniqueid2” to edit it. Update the value for field **field2** as shown below

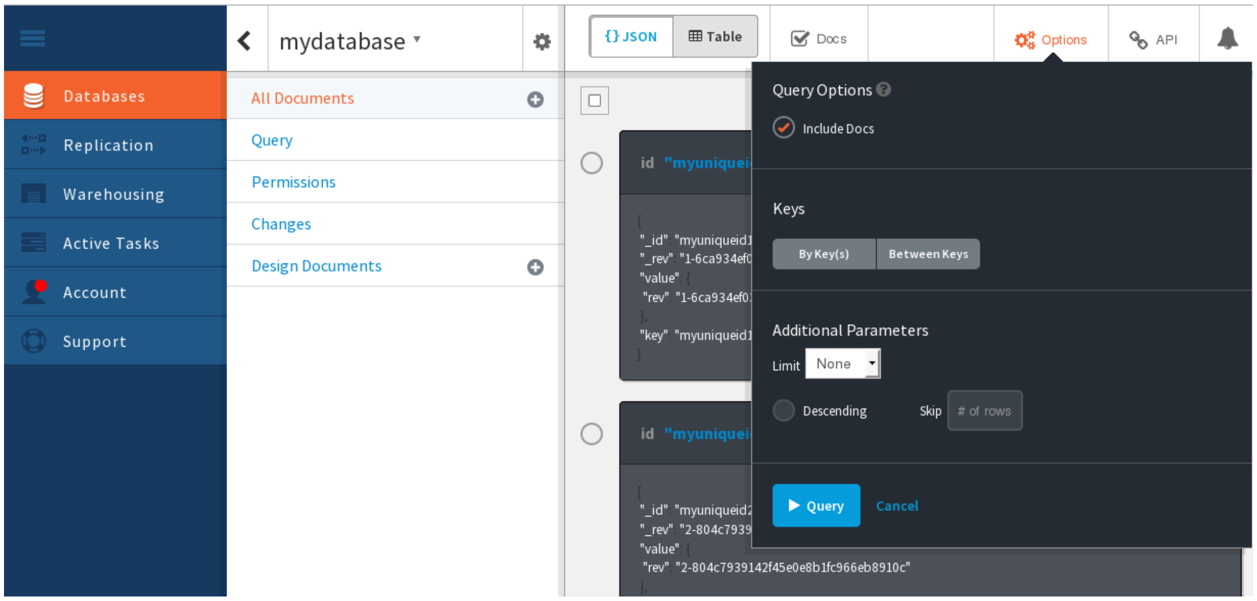


```
{
  "_id": "myuniqueid2",
  "_rev": "1-6ca934ef03131c337ea407e5f25a40f8",
  "field2": "new-value2",
  "field3": "value3"
}
```

3. Click **Save** to save your changes and return to the database view

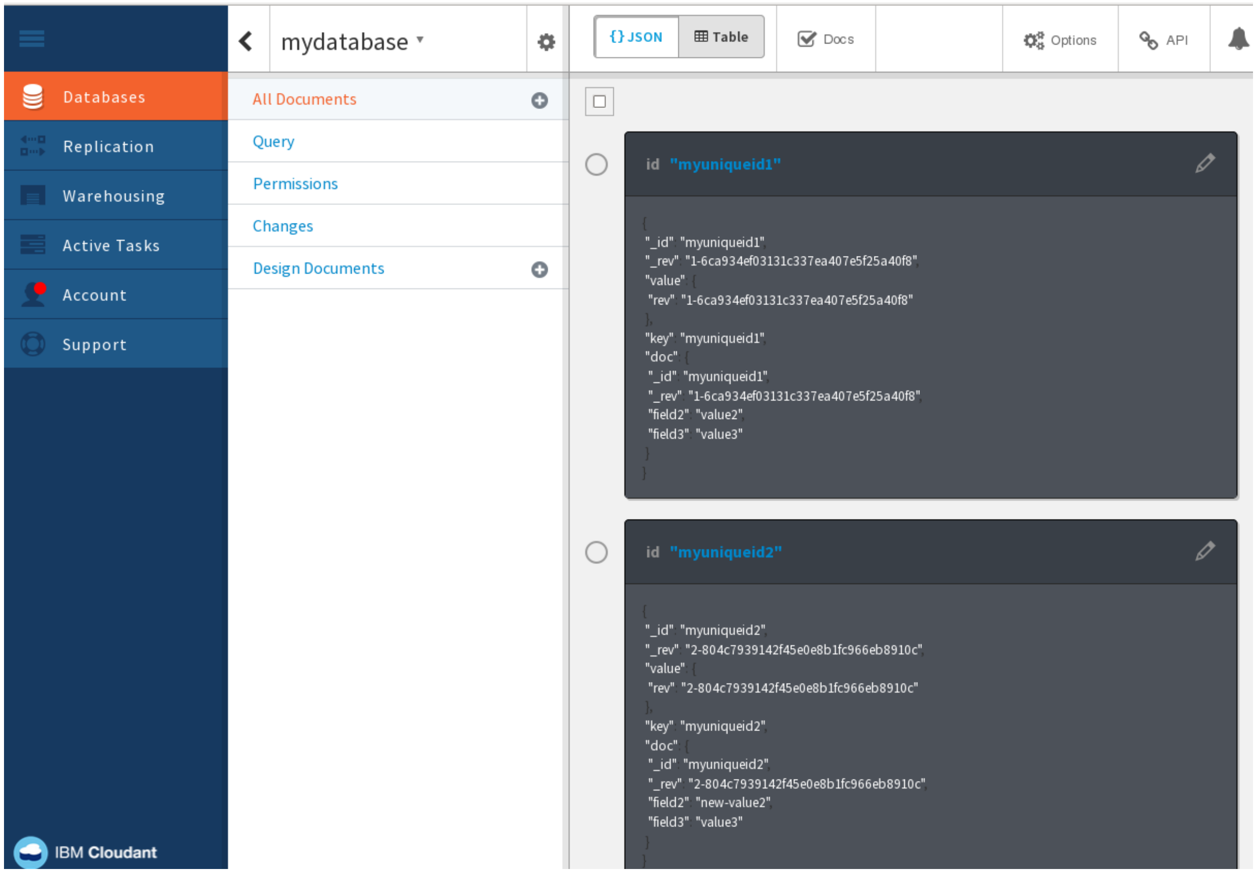
1.6 Simple query of all documents in a database

1. From the database view in the dashboard click on **All Documents**, a summary of the documents in the database will appear at the right
2. Click on **Query Options**, select **Include Docs** and click **Query**



```
id "myuniquei
{
"_id" "myuniqueid1
"_rev" "1-6ca934ef0
"value" {
"rev" "1-6ca934ef0
},
"key" "myuniqueid1
}
id "myuniquei
{
"_id" "myuniqueid2
"_rev" "2-804c7939
"value" {
"rev" "2-804c7939142f45e0e8b1fc966eb8910c"
},
```

3. Verify that all the fields in your 2 documents appear



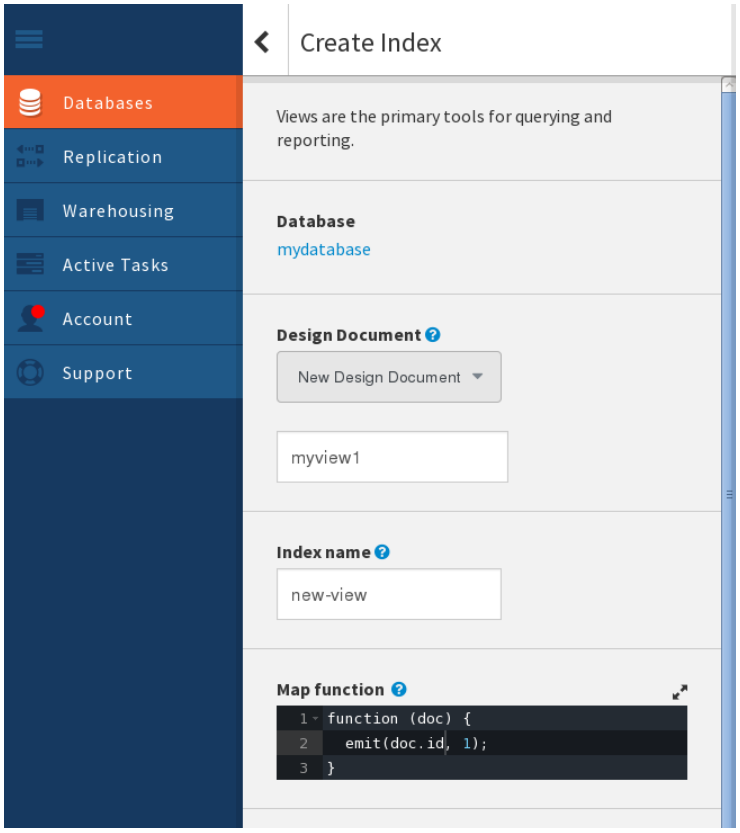
```
id "myuniqueid1"
{
"_id" "myuniqueid1",
"_rev" "1-6ca934ef03131c337ea407e5f25a40f8",
"value" {
 "rev" "1-6ca934ef03131c337ea407e5f25a40f8"
},
"key" "myuniqueid1",
"doc" {
 "_id" "myuniqueid1",
 "_rev" "1-6ca934ef03131c337ea407e5f25a40f8",
 "field2" "value2",
 "field3" "value3"
}
}

id "myuniqueid2"
{
"_id" "myuniqueid2",
"_rev" "2-804c7939142f45e0e8b1fc966eb8910c",
"value" {
 "rev" "2-804c7939142f45e0e8b1fc966eb8910c"
},
"key" "myuniqueid2",
"doc" {
 "_id" "myuniqueid2",
 "_rev" "2-804c7939142f45e0e8b1fc966eb8910c",
 "field2" "new-value2",
 "field3" "value3"
}
}
```

1.7 Create a secondary index using a view with a map function

1. From the database view in the dashboard click on the plus icon for **Design Documents** , and then select **New View**

2. Enter `myview1` as the name for the New Design Document name and click on the outward pointing arrows above the map function to expand the edit window (aka “zen mode”).



Views are the primary tools for querying and reporting.

Database
mydatabase

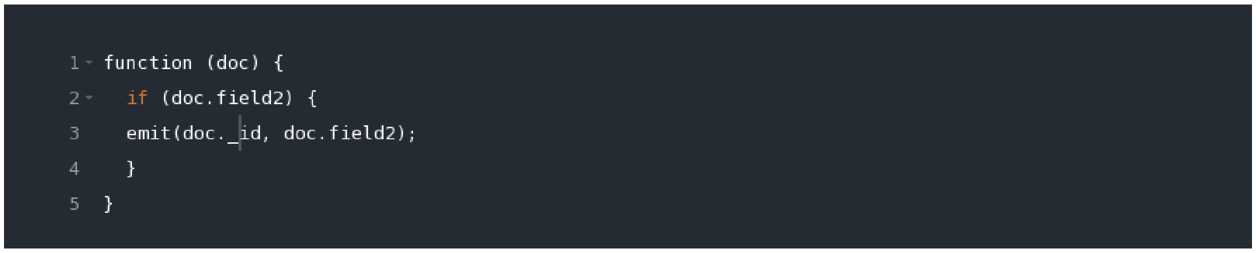
Design Document
New Design Document
myview1

Index name
new-view

Map function

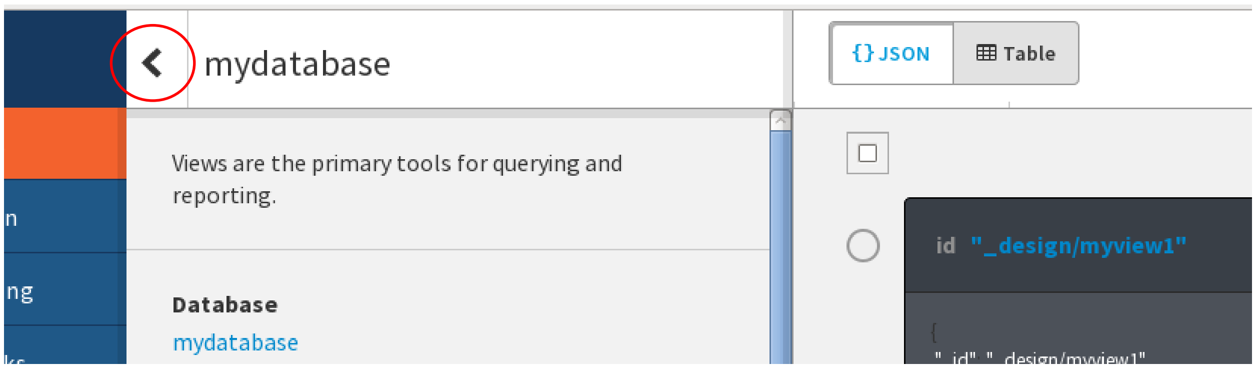
```
function (doc) {
  emit(doc.id, 1);
}
```

3. Edit the map function to check if a document has a field called **field2** and if present, to return the document id and the value of that field.



```
function (doc) {
  if (doc.field2) {
  emit(doc._id, doc.field2);
  }
}
```

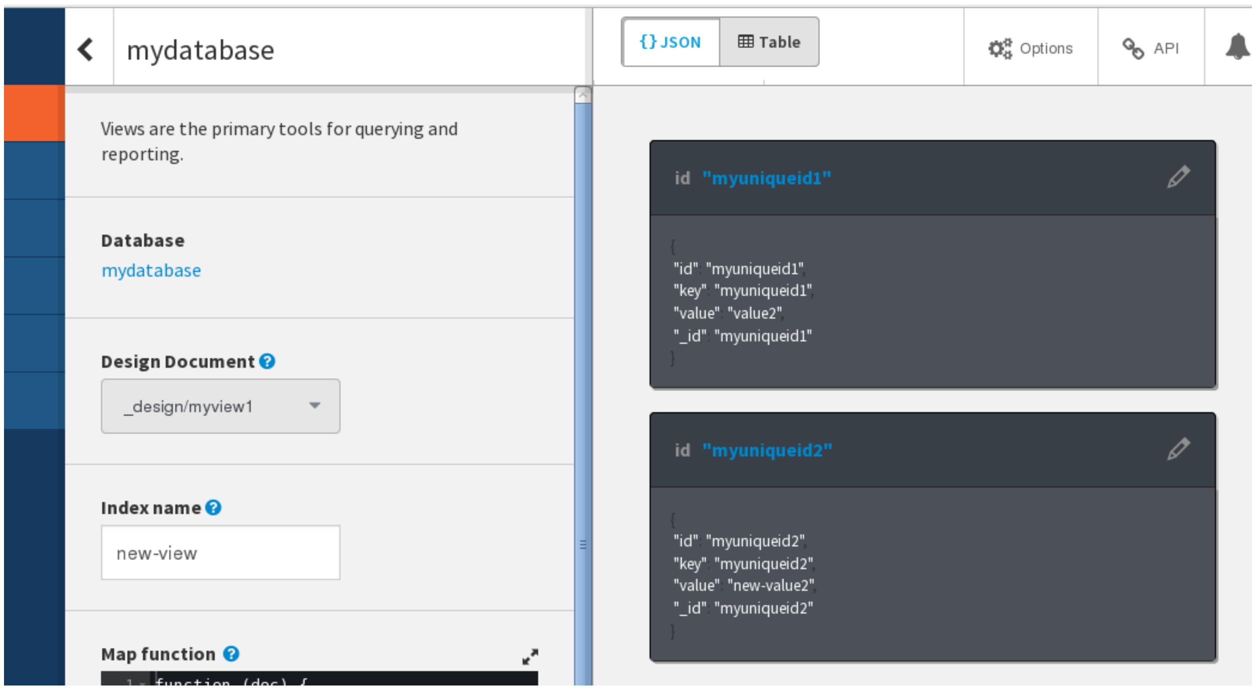
4. Click on the inward pointing arrows (or the Esc key) to go back to the view display and then scroll down and click on **Save Document and Build Index** to create the index.
5. Click on the arrow to return to the database view.



Views are the primary tools for querying and reporting.

Database
mydatabase

6. Expand **myview1**, then **Views** and select **new-view**



Views are the primary tools for querying and reporting.

Database
mydatabase

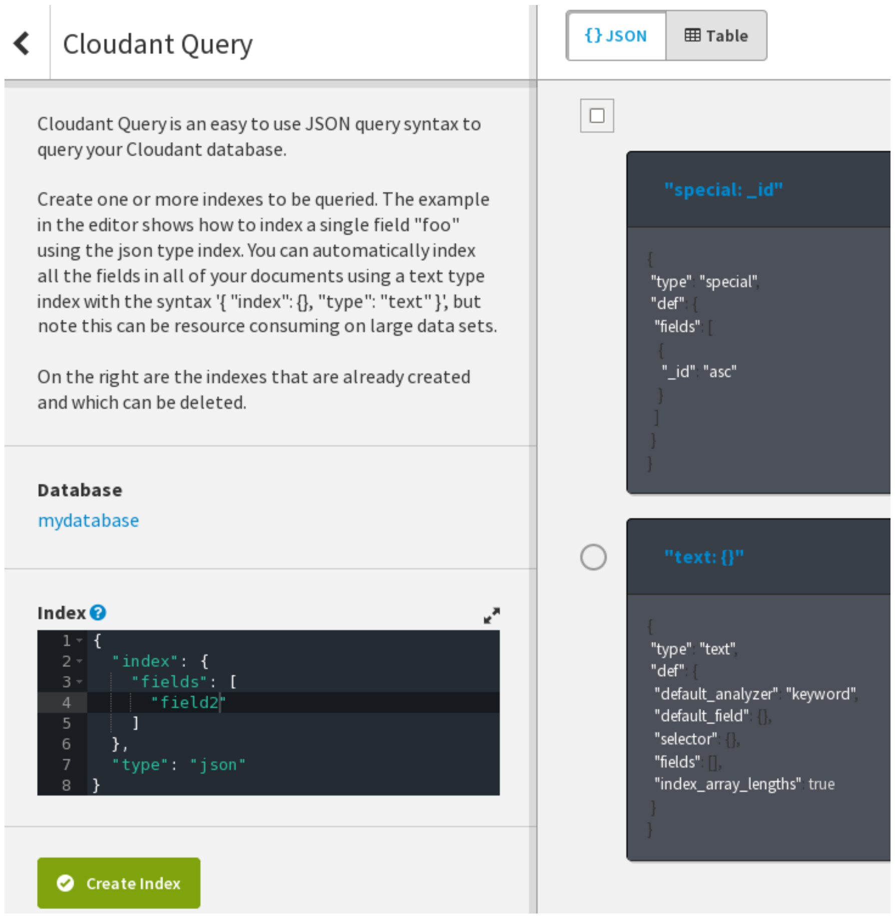
```
id "myuniqueid1"
{
"id": "myuniqueid1",
"key": "myuniqueid1",
"value": "value2",
"_id": "myuniqueid1"
}

id "myuniqueid2"
{
"id": "myuniqueid2",
"key": "myuniqueid2",
"value": "new-value2",
"_id": "myuniqueid2"
}
```

7. Note how the returned documents in the view have a key that is equal to the original document id and a value corresponding to the value that was set in the **field2** field. You may also copy the url for this view using the **API** button in the upper right and paste that into another browser tab to view the results.

1.8 Build an index for use with Cloudant Query

1. From the database view in the dashboard click on the plus icon for **Design Documents** , and then select **Query Indexes**
2. In the Cloudant Query index editor, change the default field of “foo” to “field2”, keep the index type set to json:



Cloudant Query is an easy to use JSON query syntax to query your Cloudant database.

Create one or more indexes to be queried. The example in the editor shows how to index a single field "foo" using the json type index. You can automatically index all the fields in all of your documents using a text type index with the syntax '{ "index": {}, "type": "text" }', but note this can be resource consuming on large data sets.

On the right are the indexes that are already created and which can be deleted.

Database
mydatabase

Index

```
{
  "index": {
    "fields": [
      "field2"
    ]
  },
  "type": "json"
}
```

"special: _id"

```
{
"type": "special",
"def": {
 "fields": [
  {
   "_id": "asc"
  }
 ]
}
}
```

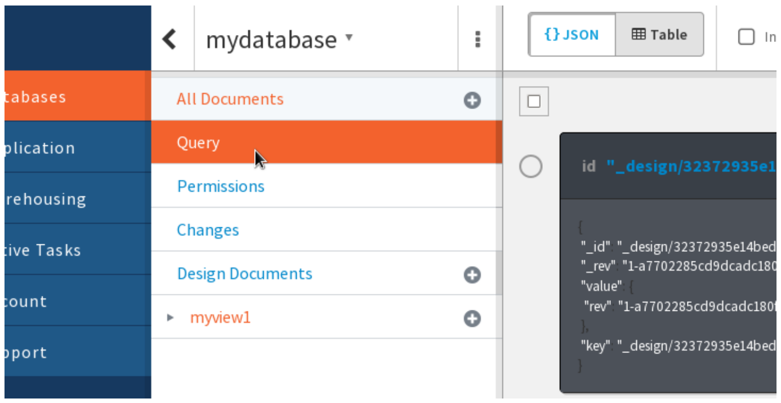
"text: {}"

```
{
"type": "text",
"def": {
 "default_analyzer": "keyword",
 "default_field": {},
 "selector": {},
 "fields": [],
 "index_array_lengths": true
}
}
```

3. Click on the **Create Index** button to create the new index.

4. Cloudant will create the index and display status updates along the top of the dashboard page. After confirmation that the index has been created, return to the database view by clicking on the **<** next to **Cloudant Query**

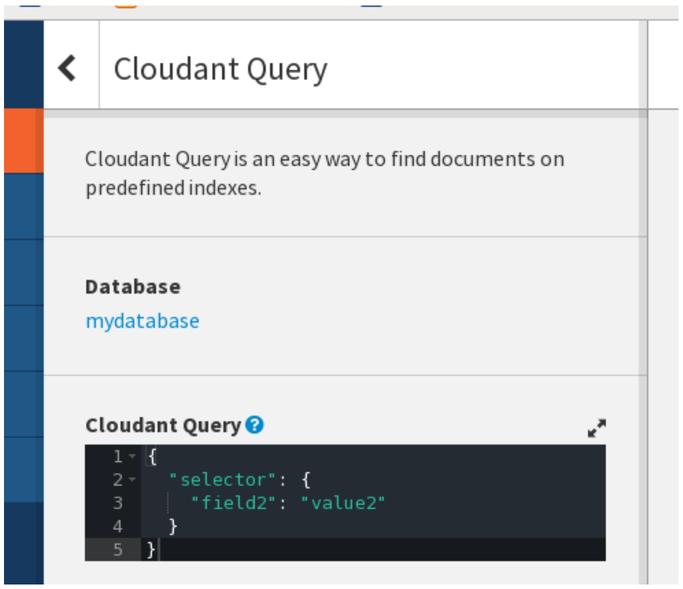
5. Click on the **Query** button



id "_design/32372935e1

```
{
"_id": "_design/32372935e14bed
"_rev": "1-a7702285cd9dcadc180
"value": {
 "rev": "1-a7702285cd9dcadc180
},
"key": "_design/32372935e14bed
}
```

6. In the Query editor update the selector to choose documents where “field2” is equal to “value2”, and simplify the query by removing the fields and sort options:


Cloudant Query

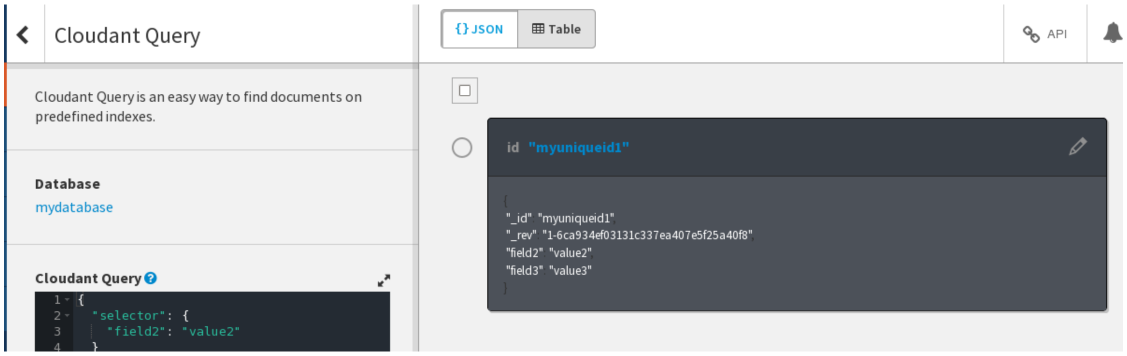
Cloudant Query is an easy way to find documents on predefined indexes.

Database
mydatabase

Cloudant Query

```
{
  "selector": {
    "field2": "value2"
  }
}
```

7. Click on the **Run Query** button to display the results of the query.



Cloudant Query

Cloudant Query is an easy way to find documents on predefined indexes.

Database
mydatabase

Cloudant Query

```
{
  "selector": {
    "field2": "value2"
  }
```

id "myuniqueid1"

```
{
"_id": "myuniqueid1",
"_rev": "1-6ca934ef03131c337ea407e5f25a40f8",
"field2": "value2",
"field3": "value3"
}
```

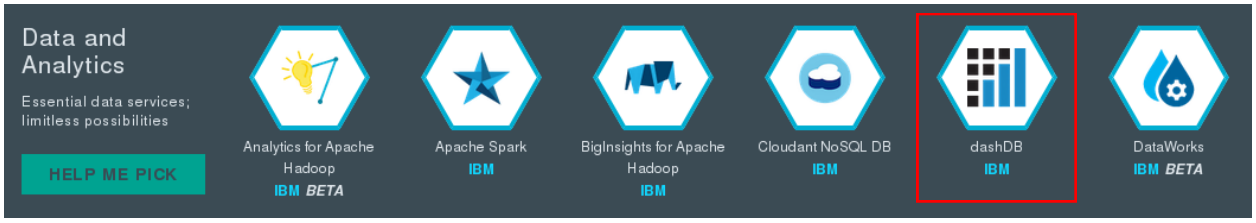
The document matched by the query is shown with all of the document fields.

2. Managing instances of dashDB

In this section you'll go through the basics of managing the dashDB service in IBM Bluemix.

2.1 Launch the dashDB Dashboard

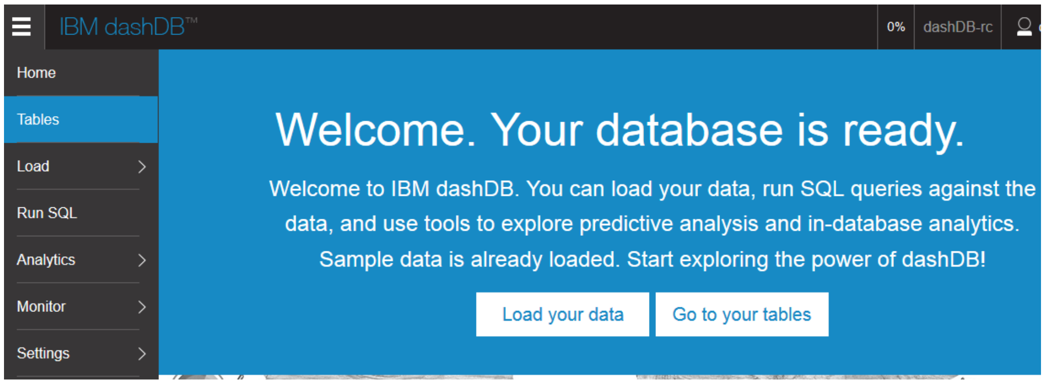
1. In your browser go to the Bluemix URL http://bluemix.net and login is necessary.
2. Make sure you're in the Dashboard tab (if not click on the Dashboard link at the top of the page to take you there)
3. Click on **USE SERVICES OR APIS**
4. Scroll down to the **Data and Analytics** section and click **dashDB.**


Data and Analytics
Essential data services; limitless possibilities
HELP ME PICK
Analytics for Apache Hadoop IBM BETA
Apache Spark IBM
BigInsights for Apache Hadoop IBM
Cloudant NoSQL DB IBM
dashDB IBM
DataWorks IBM BETA

5. Under **App** select **Leave unbound**
6. Click on **CREATE** to create a new instance of dashDB
7. Click on **Launch** when the service's landing page appears to bring up the dashDB console

2.2 Create a new table

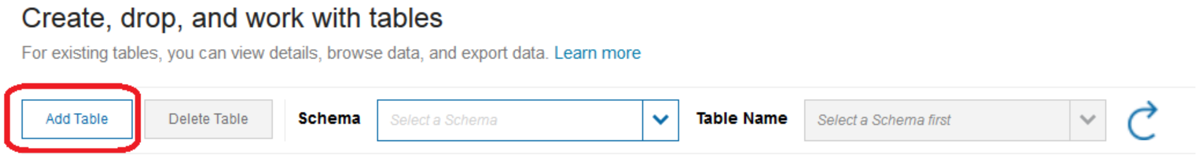
1. From the dashDB console click on **Tables**



Welcome. Your database is ready.

Welcome to IBM dashDB. You can load your data, run SQL queries against the data, and use tools to explore predictive analysis and in-database analytics. Sample data is already loaded. Start exploring the power of dashDB!

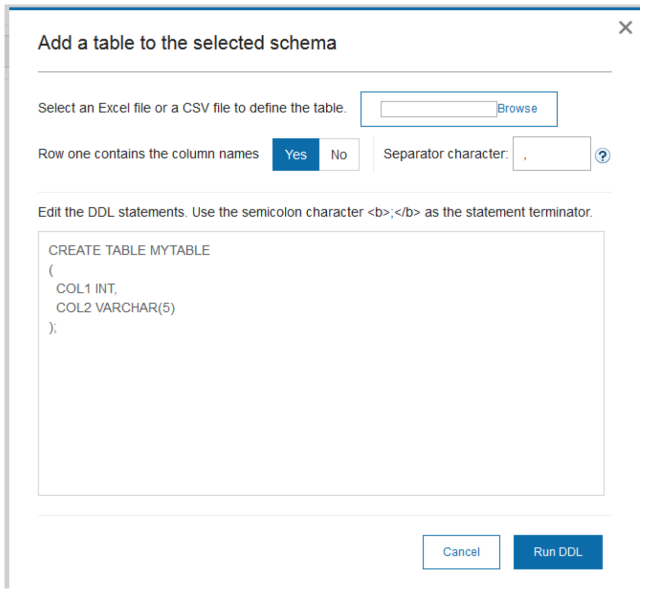
2. Click on **Add Table**.



Create, drop, and work with tables

For existing tables, you can view details, browse data, and export data. Learn more

3. Some sample DDL to create a table is generated for you.


Add a table to the selected schema

Select an Excel file or a CSV file to define the table. Browse

Row one contains the column names [✓ Yes] No Separator character: ,

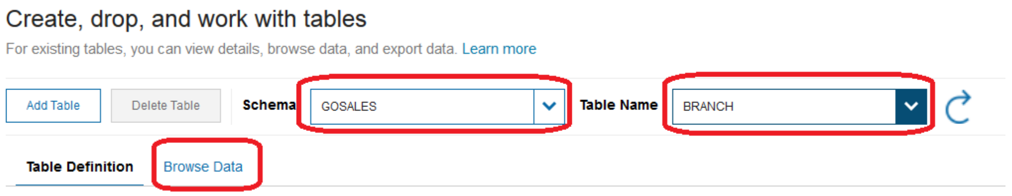
Edit the DDL statements. Use the semicolon character <b>;</b> as the statement terminator.

```
CREATE TABLE MYTABLE
(
  COL1 INT,
  COL2 VARCHAR(5)
);
```

4. Let's use the sample DDL to keep things simple. Click **Run DDL**
5. Click **OK** in the dialog that indicates the DDL ran successfully and then click **Cancel** to exit the dialog with the sample DDL.

2.3 Browse an existing table

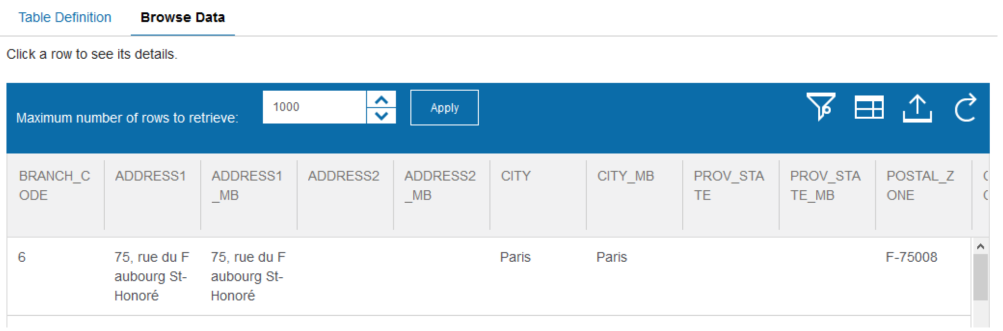
1. In the dashDB console select **GOSALES** as the **Schema** and **BRANCH** as the **Table Name** and then click on **Browse Data**


Create, drop, and work with tables

For existing tables, you can view details, browse data, and export data. Learn more

Schema: GOSALES
Table Name: BRANCH

2. Verify that the data in the BRANCH table is displayed



Click a row to see its details.

Maximum number of rows to retrieve: 1000

BRANCH_CODE	ADDRESS1	ADDRESS1_MB	ADDRESS2	ADDRESS2_MB	CITY	CITY_MB	PROV_STATE	PROV_STATE_MB	POSTAL_ZONE
6	75, rue du Faubourg St-Honoré	75, rue du Faubourg St-Honoré			Paris	Paris			F-75008

3. Verify that you can click on a row to see all the data for the row

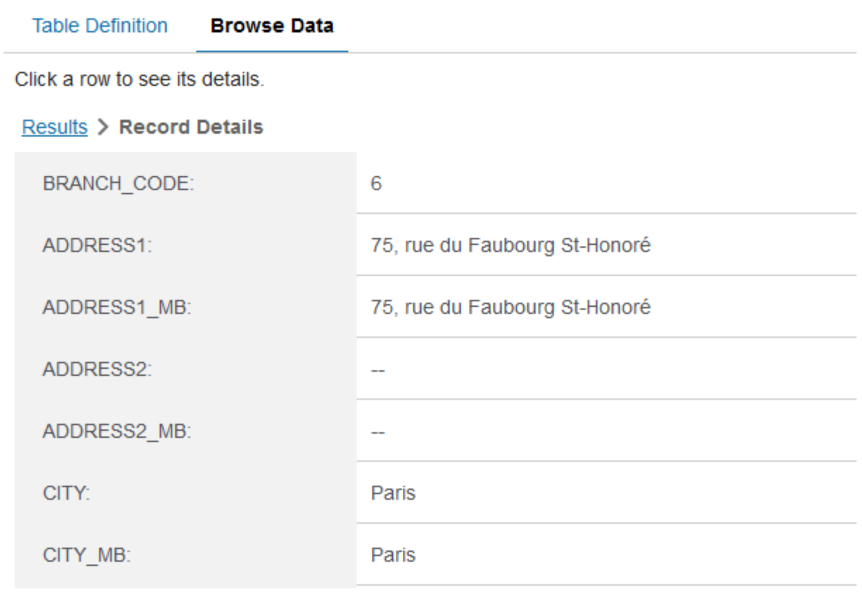


Table Definition | Browse Data

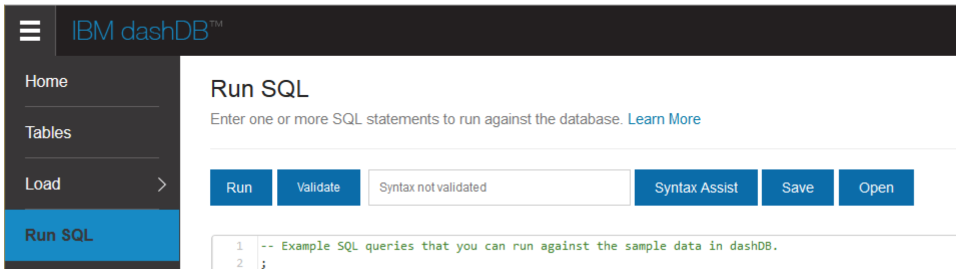
Click a row to see its details.

Results > Record Details

BRANCH_CODE:	6
ADDRESS1:	75, rue du Faubourg St-Honoré
ADDRESS1_MB:	75, rue du Faubourg St-Honoré
ADDRESS2:	--
ADDRESS2_MB:	--
CITY:	Paris
CITY_MB:	Paris

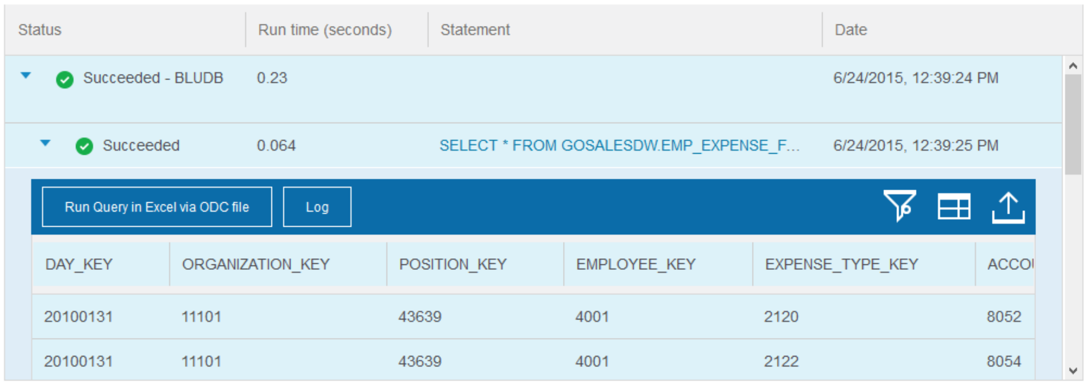
2.4 Run SQL Scripts

1. In the dashDB console click on **Run SQL** in the navigation area on the left



```
-- Example SQL queries that you can run against the sample data in dashDB.
;
```

2. A sample script is preloaded that runs against the sample data in dashDB.
3. Click **Validate** to check the syntax of the script
4. Click **Run** to run the sample script
5. Navigate through the results to see the data returned



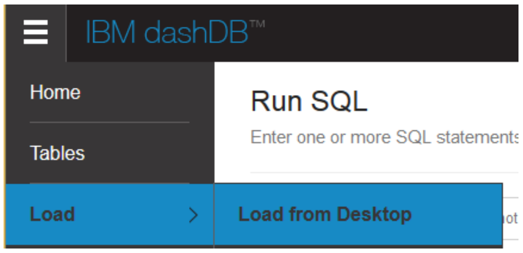
Status	Run time (seconds)	Statement	Date
Succeeded - BLUDB	0.23		6/24/2015, 12:39:24 PM
Succeeded	0.064	SELECT * FROM GOSALESDW.EMP_EXPENSE_F...	6/24/2015, 12:39:25 PM

DAY_KEY	ORGANIZATION_KEY	POSITION_KEY	EMPLOYEE_KEY	EXPENSE_TYPE_KEY	ACCO
20100131	11101	43639	4001	2120	8052
20100131	11101	43639	4001	2122	8054

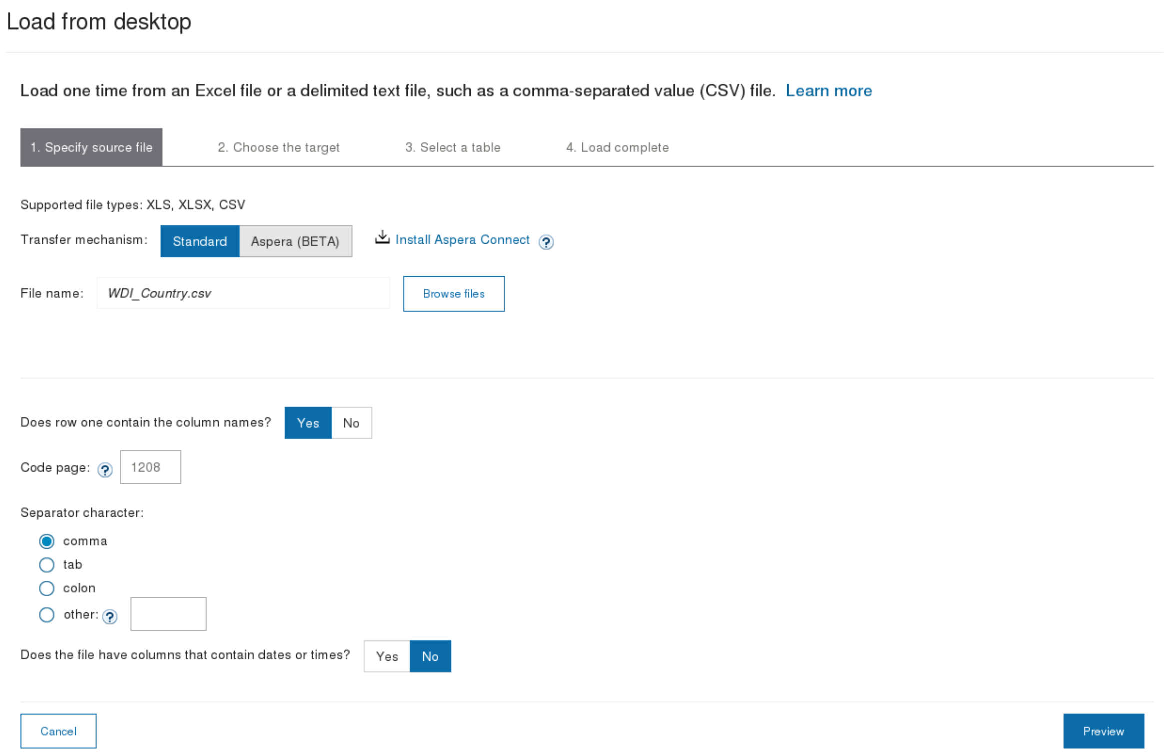
2.5 Import CSV data

1. Download the file WDI_Country.csv file to your local machine from https://ibm.biz/LabCSV-SampleFile This file is from a dataset from the contains data from the World Bank on economic data.

2. In the dashDB console select ***Load->Load from Desktop***



3. Click **Browse files**, select the **WDI_Country.csv** that was just downloaded.



Load one time from an Excel file or a delimited text file, such as a comma-separated value (CSV) file. Learn more

Supported file types: XLS, XLSX, CSV

Transfer mechanism: [✓ Standard] Aspera (BETA)

File name: WDI_Country.csv

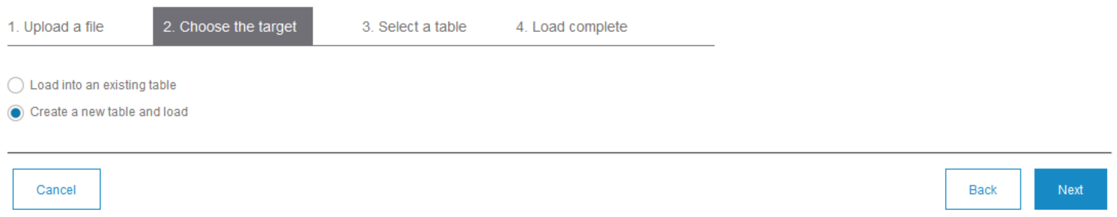
Does row one contain the column names? [✓ Yes] No

Code page: 1208

Separator character: [✓ comma] tab colon other:

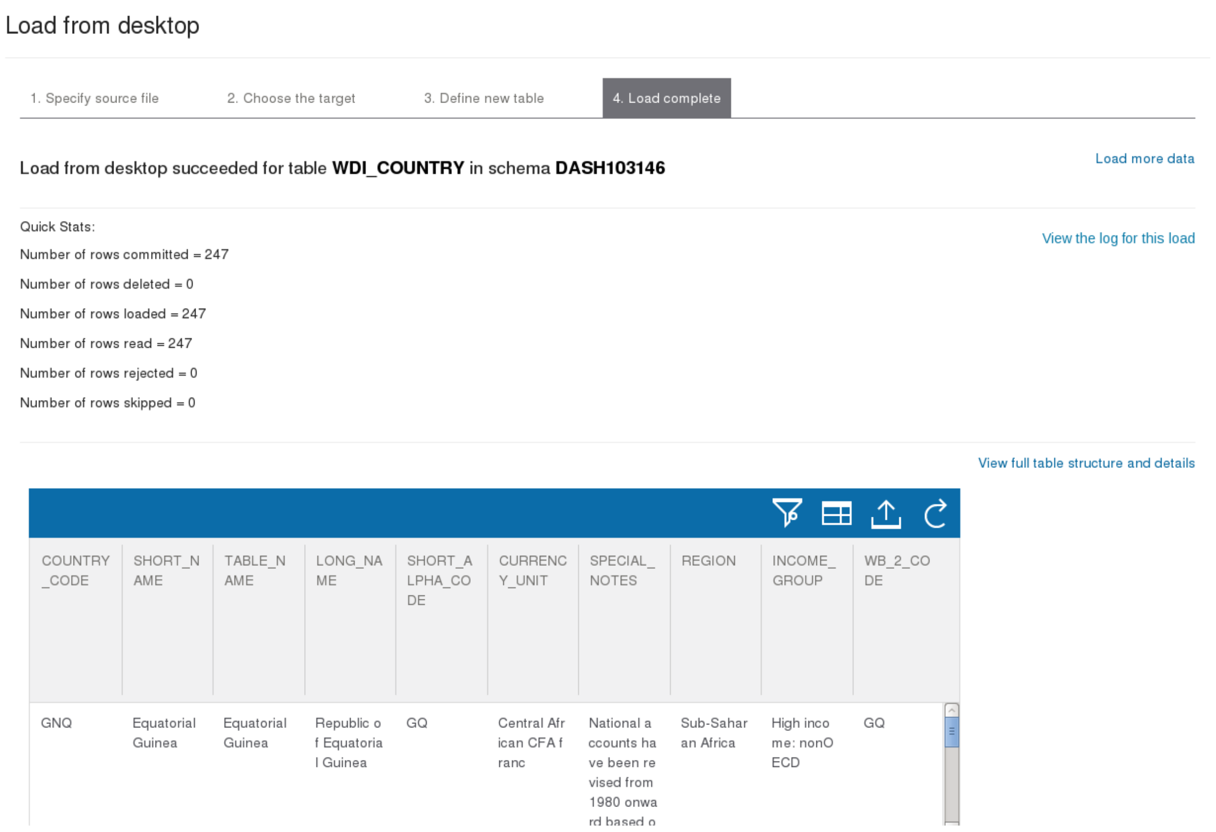
Does the file have columns that contain dates or times? Yes [✓ No]

4. Click **Preview**
5. A preview of the data is shown, click **Next**
6. Select **Create a new table and load**



Load into an existing table
[✓ Create a new table and load]

7. Click **Next**
8. Accept the defaults for the new table definition and click Finish
9. A preview of the data in the new table will appear as well as the number of rows imported



Load from desktop succeeded for table **WDI_COUNTRY** in schema **DASH103146**

Quick Stats:

Number of rows committed = 247

Number of rows deleted = 0

Number of rows loaded = 247

Number of rows read = 247

Number of rows rejected = 0

Number of rows skipped = 0

COUNTRY_CODE	SHORT_NAME	TABLE_NAME	LONG_NAME	SHORT_ALPHA_CODE	CURRENCY_UNIT	SPECIAL_NOTES	REGION	INCOME_GROUP	WB_2_CODE
GNQ	Equatorial Guinea	Equatorial Guinea	Republic of Equatorial Guinea	GQ	Central African CFA franc	National accounts have been revised from 1980 onward based o	Sub-Saharan Africa	High income: nonOECD	GQ

Congratulations. You've mastered the basic of managing the following Data Services in IBM Bluemix:

- Cloudant NoSQL Database
- dashDB