English

Using IBM Cloudant Query

Last updated 2023-04-11

In this tutorial, we demonstrate how to create an index and use the index to query the database. You also learn to create different types of queries to more easily find data.

Here you run the commands from the command line, but you can also complete these tasks with the IBM® Cloudant® for IBM Cloud® Dashboard, which gives you a visual example of each task. For more information about the dashboard, see Using the IBM Cloudant Dashboard tutorial.

Before you begin

Before you begin, follow these tutorials to create an instance, and then create and populate a database.

1. Create an IBM Cloudant instance.
2. Create a database.
3. Populate the database.
4. (Optional) Create an `acurl` alias.

> **Attention:** If you decide not to set up `acurl`, use the following URL with `curl` instead of the one provided in the exercises, `curl "https://$USERNAME:$PASSWORD@$ACCOUNT.cloudant.com/databasedemo"`.

The `acurl` alias is more secure. It prevents someone from reading your password over your shoulder as you type. It also makes sure that your password isn't sent in plain text over the network by enforcing HTTPS.

Now, we're ready to learn how to run queries against the database you created in step two of Before you begin.

Step 1: Creating an index

IBM Cloudant Query uses Mongo-style query syntax to search for documents by using logical operators. IBM Cloudant Query is a combination of a view and a search index.

When you use IBM Cloudant Query, the query planner looks at the selector (your query) to determine the correct index to choose from. In memory, you filter out the documents by the selector, which is why, even without an index, you can still query with various fields.

> **Tip:** If no available defined index matches the specified query, then IBM Cloudant uses the `_all_docs` index, which looks up documents by ID. In the worst case scenario, it returns all the documents by ID (full table scan). Full table scans are expensive to process. It is recommended that you create an index.

To create an index, follow these steps:

1. Copy the following sample JSON data into a file named `query-demo-index.json`:

```
{
	"index": {
		"fields": [
			"descriptionField",
			"temperatureField"
		],
		"partial_filter_selector": {
			"descriptionField": {
				"$eq": "hot"
			},
			"temperatureField": {
				"$gt": 50
			}
		}
```

Show more

2. Run the following command to create an index:

```
acurl "https://$ACCOUNT.cloudant.com/databasedemo/_index" \
  -X POST \
  -H "Content-Type: application/json" \
  -d \@query-demo-index.json
```

3. Review the results:

```
{
      "result": "created",
      "id": "_design/query-demo-index",
      "name": "490441584f9eddb8d09ef234d636b5f3b18e4ce6"
}
```

You aren't required to create an index to run a query. However, if you don't, the following warning is included with your results as an indicator that creating an index reduces processing and makes your queries more effective. `"Warning": "No matching index found, create an index to optimize query time."`

Step 2: Running a simple query

This example demonstrates how IBM Cloudant Query finds documents based on the `descriptionField` with the value `boiling`.

To run the query, follow these steps:

1. Copy the following sample JSON into a data file named `query1.json`:

```
{
 "selector": {
   "descriptionField": "boiling"
```

```
    }
}
```

2. Run the following command to query the database:

```
acurl "https://$ACCOUNT.cloudant.com/databasedemo/_find" \
  -X POST \
  -H "Content-Type: application/json" \
  -d \@query1.json
```

3. Review the query results:

```
{
      "docs": [{
            "_id": "91d1fa833d28efe15069604f98de701d",

            "_rev": "1-f998fc7b89d4466c1e7bb204b1b00f74",
            "numberField": 1,
            "nameField": "one",
            "descriptionField": "boiling",
            "temperatureField": 100
      }],
      "bookmark":
"g1AAAABweJzLYWBgYMpgSmHgKy5JLCrJTq2MT8lPzkzJBYorWBqmGKYlWhgbpxhZpKalGpoamFmaGZikWVqkpJob
GKaA9HHA9BGlIwsAmn8eLw",
      "warning": "No matching index found, create an index to optimize query time."
}
```

Show more

Next, you find a document in the database by using two fields.

Step 3: Running a query with two fields

This example uses two fields to find a document with the values: `freezing` and `-5`.

The search is described by using a 'selector' expression that looks like the following example:

```
{
        "selector": {
                "descriptionField": "freezing",
                "temperatureField": -5
        }
}
```

You can tailor the results by adding more details within the selector expression. The `fields` parameter specifies the fields to include with the results. In our example, the results include the `nameField`, `descriptionField`, and `temperatureField`, as shown in the following example.

```
{
        ...
        "fields": [
                "nameField",
                "descriptionField",
                "temperatureField"
        ]
}
```

To run the query, follow these steps:

1. Copy the sample JSON into a data file named `query2.json`.

```
{
        "selector": {
                "descriptionField": "freezing",
                "temperatureField": -5
        },
        "fields": [
                "nameField",
```

```
                "descriptionField",
                "temperatureField"
        ]
}
```

2. Run the following command to query the database:

```
acurl "https://$ACCOUNT.cloudant.com/databasedemo/_find" \
  -X POST \
  -H "Content-Type: application/json" \
  -d \@query2.json
```

3. Review the query results:

```
{
        "docs": [{
                "nameField": "eight",
                "descriptionField": "freezing",
                "temperatureField": -5
        }],
        "bookmark":
"g1AAAABweJzLYWBgYMpgSmHgKy5JLCrJTq2MT8lPzkzJBYorpBiYGJolWaSZmaWYpFqmGBiYmKclphhappqZmRgl
G5uD9HHA9BGlIwsAms8eJw",
        "warning": "No matching index found, create an index to optimize query time."
}
```

Next, you find a document in the database by using multiple operators.

Step 4: Running a query with operators

In this example, the `$gt` (greater than) and `$eq` (equal) operators are used to search for documents that include a temperature that is greater than `20` degrees and a description that contains the value `hot`. The results include the `descriptionField` and

`temperatureField`, shown in descending order by temperature.

You use a selector expression like the following example:

```
{
        "selector": {
                "temperatureField": {
                        "$gt": 50
                },
                "descriptionField": {
                        "$eq": "hot"
                }
        },
        "fields": [
                "descriptionField",
                "temperatureField"
        ],
        "sort": [{
                "temperatureField": "desc"
```

Show more

To run the query, follow these steps:

1. Copy the following sample JSON to a file named `query3.json`.

```
{
        "selector": {
                "descriptionField": {
                        "$eq": "hot"
                },
                "temperatureField": {
                        "$gt": 50
                }
        },
        "fields": [
                "descriptionField",
```

```
                "temperatureField"
        ],
        "sort": [{                                                    Show more
```

2. Run this query:

```
acurl "https://$ACCOUNT.cloudant.com/databasedemo/_find" \
  -X POST \
  -H "Content-Type: application/json" \
  -d \@query3.json
```

3. Review the query results:

```
{
        "docs": [{
                        "descriptionField": "hot",
                        "temperatureField": 97
                },
                {
                        "descriptionField": "hot",
                        "temperatureField": 75
                }
        ],
        "bookmark":
"g1AAAABbeJzLYWBgYMpgSmHgKy5JLCrJTq2MT8lPzkzJBYorJBoYmKWaWVokpZkamJtaGJskm5slmSenGhulWpib
JhqC9HHA9OWATErUAGlkzsgvycoCAEsUF_A"
}
```

Now you know how to extract data from your database by using IBM Cloudant Query. For more information, see the IBM Cloudant documentation.

Contribute in GitHub

Open doc issue | Edit topic