Querying JSON & Hierarchical Data with SQL

Apache Spark™ and Databricks® make it easy to work with hierarchical data, such as nested JSON records.

In this lesson you:

- Use SQL to query a table backed by JSON data
- Query nested structured data
- Query data containing array columns

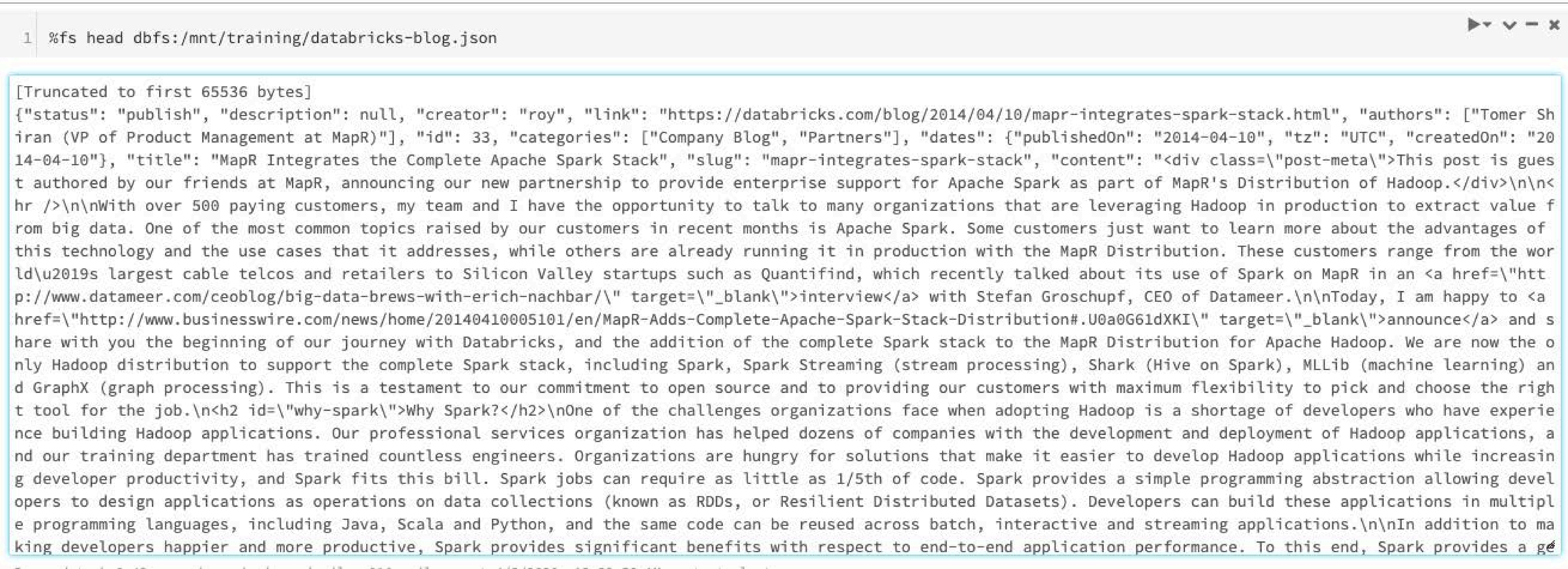
Examining the contents of a JSON file

JSON is a common file format in big data applications and in data lakes (or large stores of diverse data). Datatypes such as JSON arise out of a number of data needs. For instance, what if...

- Your schema, or the structure of your data, changes over time?
- You need nested fields like an array with many values or an array of arrays?
- You don't know how you're going use your data yet so you don't want to spend time creating relational tables?

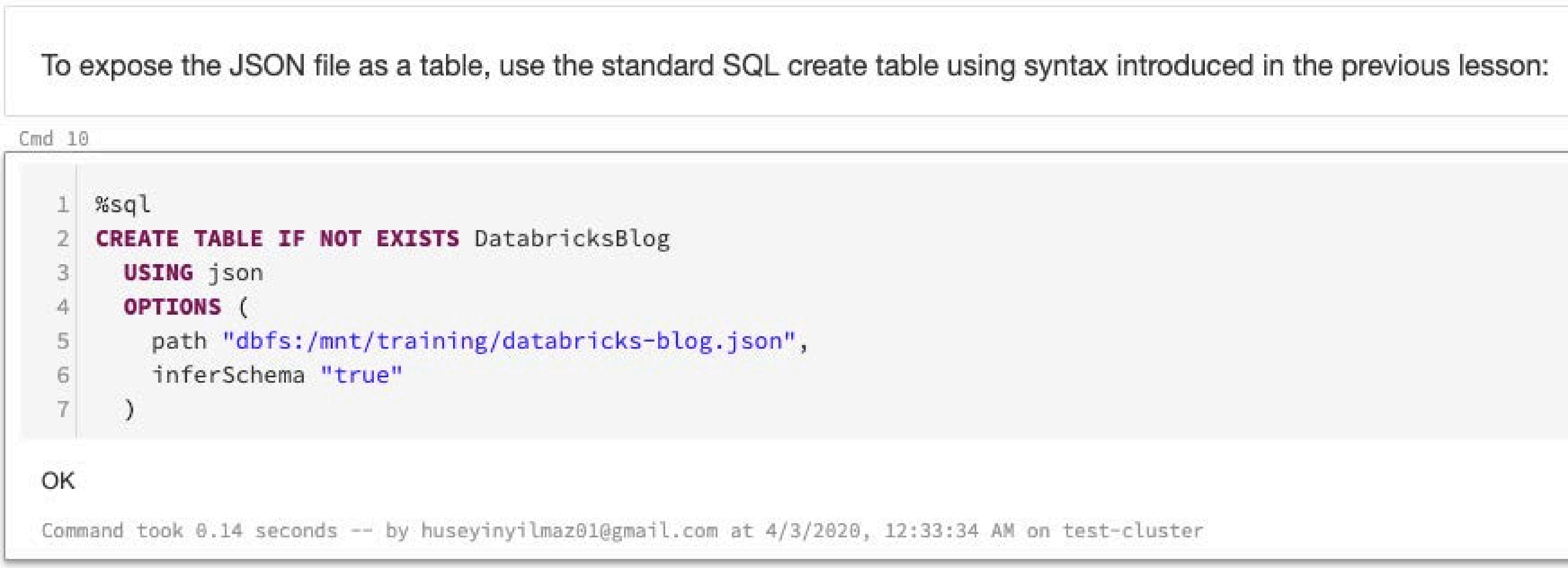
The popularity of JSON is largely due to the fact that JSON allows for nested, flexible schemas.

This lesson uses the `DatabricksBlog` table, which is backed by JSON file `dbfs:/mnt/training/databricks-blog.json`. If you examine the raw file, you can see that it contains compact JSON data. There's a single JSON object on each line of the file; each object corresponds to a row in the table. Each row represents a blog post on the Databricks blog, and the table contains all blog posts through August 9, 2017.



```
%fs head dbfs:/mnt/training/databricks-blog.json
```

```
[Truncated to first 65536 bytes]
{"status": "publish", "description": null, "creator": "roy", "link": "https://databricks.com/blog/2014/04/10/mapr-integrates-spark-stack.html", "authors": ["Tomer Sh
iran (VP of Product Management at MapR)"], "id": 33, "categories": ["Company Blog", "Partners"], "dates": {"publishedOn": "2014-04-10", "tz": "UTC", "createdOn": "20
14-04-10"}, "title": "MapR Integrates the Complete Apache Spark Stack", "slug": "mapr-integrates-spark-stack", "content": "<div class=\"post-meta\">This post is gues
t authored by our friends at MapR, announcing our new partnership to provide enterprise support for Apache Spark as part of MapR's Distribution of Hadoop.</div>\n\n<
hr />\n\nWith over 500 paying customers, my team and I have the opportunity to talk to many organizations that are leveraging Hadoop in production to extract value f
rom big data. One of the most common topics raised by our customers in recent months is Apache Spark. Some customers just want to learn more about the advantages of
this technology and the use cases that it addresses, while others are already running it in production with the MapR Distribution. These customers range from the wor
ld\u2019s largest cable telcos and retailers to Silicon Valley startups such as Quantifind, which recently talked about its use of Spark on MapR in an <a href=\"htt
p://www.datameer.com/ceoblog/big-data-brews-with-erich-nachbar/\" target=\"_blank\">interview</a> with Stefan Groschupf, CEO of Datameer.\n\nToday, I am happy to <a
href=\"http://www.businesswire.com/news/home/20140410005101/en/MapR-Adds-Complete-Apache-Spark-Stack-Distribution#.U0a0G61dXKI\" target=\"_blank\">announce</a> and s
hare with you the beginning of our journey with Databricks, and the addition of the complete Spark stack to the MapR Distribution for Apache Hadoop. We are now the o
nly Hadoop distribution to support the complete Spark stack, including Spark, Spark Streaming (stream processing), Shark (Hive on Spark), MLLib (machine learning) an
d GraphX (graph processing). This is a testament to our commitment to open source and to providing our customers with maximum flexibility to pick and choose the righ
t tool for the job.\n<h2 id=\"why-spark\">Why Spark?</h2>\nOne of the challenges organizations face when adopting Hadoop is a shortage of developers who have experie
nce building Hadoop applications. Our professional services organization has helped dozens of companies with the development and deployment of Hadoop applications, a
nd our training department has trained countless engineers. Organizations are hungry for solutions that make it easier to develop Hadoop applications while increasin
g developer productivity, and Spark fits this bill. Spark jobs can require as little as 1/5th of code. Spark provides a simple programming abstraction allowing devel
opers to design applications as operations on data collections (known as RDDs, or Resilient Distributed Datasets). Developers can build these applications in multipl
e programming languages, including Java, Scala and Python, and the same code can be reused across batch, interactive and streaming applications.\n\nIn addition to ma
king developers happier and more productive, Spark provides significant benefits with respect to end-to-end application performance. To this end, Spark provides a ge
```


To expose the JSON file as a table, use the standard SQL create table using syntax introduced in the previous lesson:

Cmd 10

```
%sql
CREATE TABLE IF NOT EXISTS DatabricksBlog
  USING json
  OPTIONS (
    path "dbfs:/mnt/training/databricks-blog.json",
    inferSchema "true"
  )
```

OK

```
Command took 0.14 seconds -- by huseyinyilmaz01@gmail.com at 4/3/2020, 12:33:34 AM on test-cluster
```

Take a look at the schema with the `DESCRIBE` function.

Cmd 12

```
%sql
DESCRIBE DatabricksBlog
```

col_name	data_type	comment
authors	array<string>	null
categories	array<string>	null
content	string	null
creator	string	null
dates	struct<createdOn:string,publishedOn:string,tz:string>	null
description	string	null
id	bigint	null
link	string	null

Run a query to view the contents of the table.

Notice:

- The `authors` column is an array containing multiple author names.
- The `categories` column is an array of multiple blog post category names.
- The `dates` column contains nested fields `createdOn` , `publishedOn` and `tz` .

Cmd 14

```
%sql
SELECT authors, categories, dates, content
FROM DatabricksBlog
```

(1) Spark Jobs

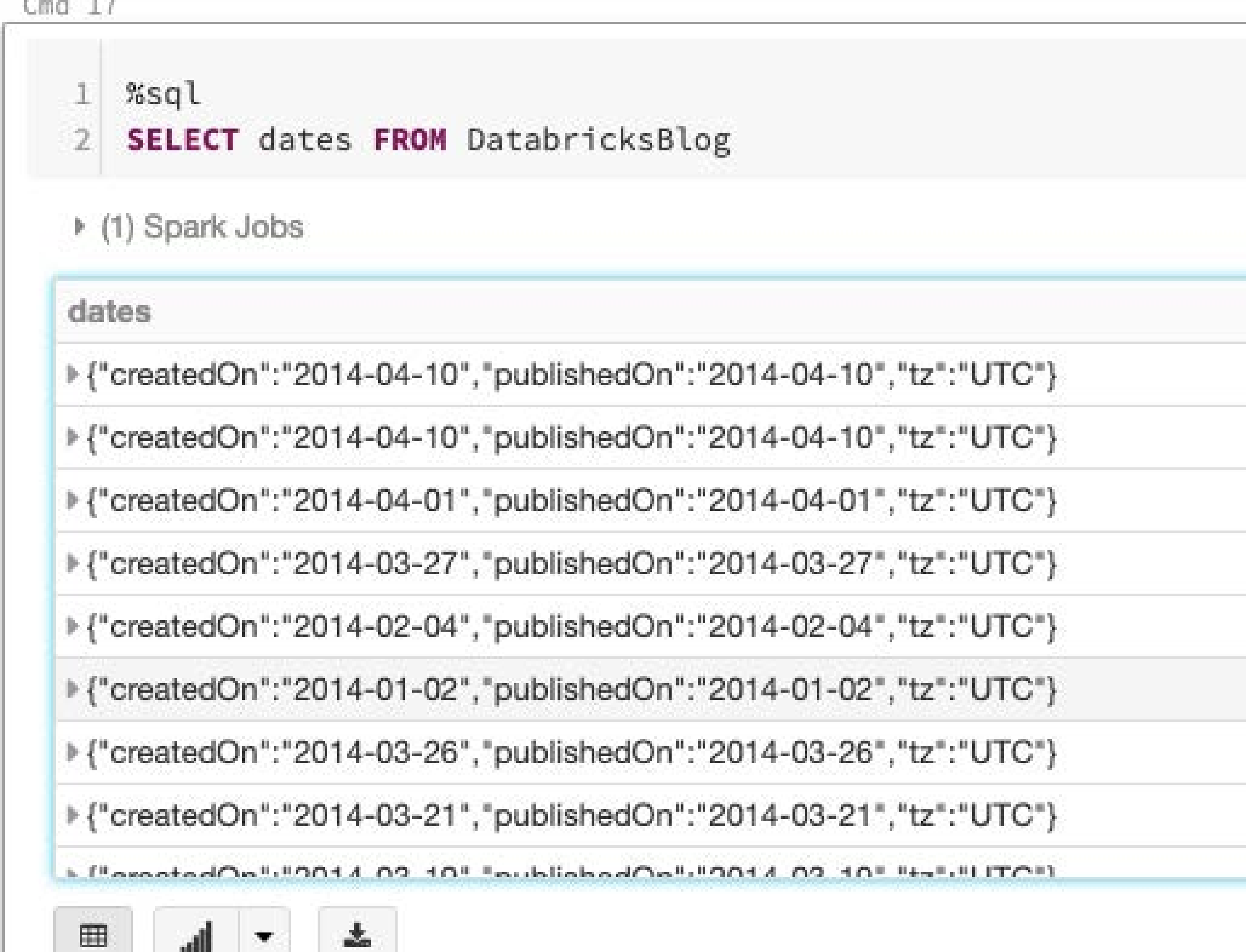
authors	categories	dates	content
["Tomer Shiran (VP of Product Management at MapR)"]	["Company Blog","Partners"]	{"createdOn":"2014-04-10","publishedOn":"2014-04-10","tz":"UTC"}	<div class="post-meta">This post is guest authored by our friends at MapR, announcing our new partnership to provide enterpris </div> <hr /> With over 500 paying customers, my team and I have the opportunity to talk to many organizations that are leveraging Hadoop in topics raised by our customers in recent months is Apache Spark. Some customers just want to learn more about the advantages already running it in production with the MapR Distribution. These customers range from the world's largest cable telcos and retail about its use of Spark on MapR in an <a href="http://www.datameer.com/ceoblog/big-data-brews-with-erich-nachbar/" target="_ Today, I a...

Cmd 15

Nested Data

Think of nested data as columns within columns.

For instance, look at the `dates` column.

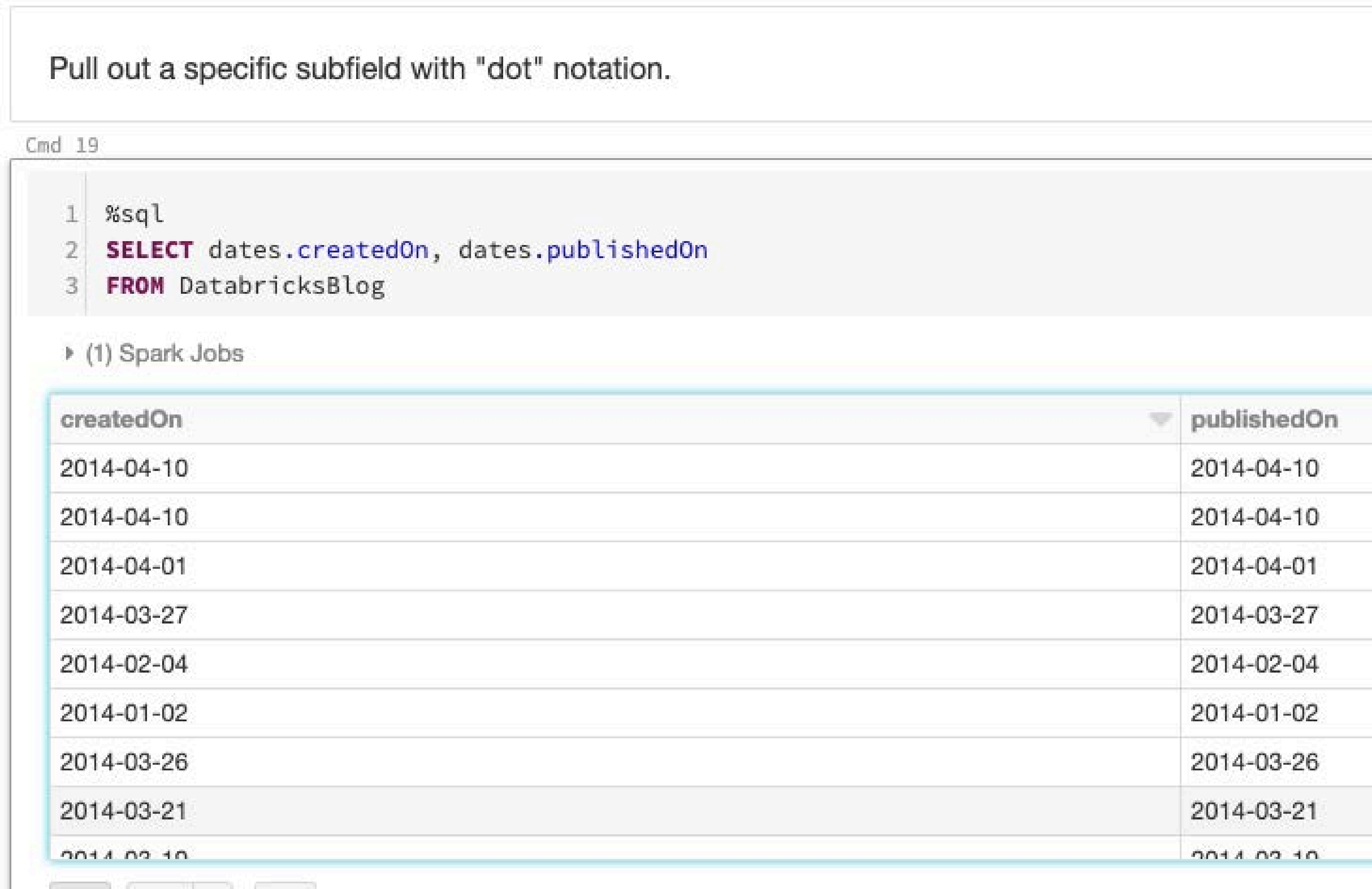


Cmd 17

```sql
%sql
SELECT dates FROM DatabricksBlog
```

(1) Spark Jobs

```
dates
{"createdOn":"2014-04-10","publishedOn":"2014-04-10","tz":"UTC"}
{"createdOn":"2014-04-10","publishedOn":"2014-04-10","tz":"UTC"}
{"createdOn":"2014-04-01","publishedOn":"2014-04-01","tz":"UTC"}
{"createdOn":"2014-03-27","publishedOn":"2014-03-27","tz":"UTC"}
{"createdOn":"2014-02-04","publishedOn":"2014-02-04","tz":"UTC"}
{"createdOn":"2014-01-02","publishedOn":"2014-01-02","tz":"UTC"}
{"createdOn":"2014-03-26","publishedOn":"2014-03-26","tz":"UTC"}
{"createdOn":"2014-03-21","publishedOn":"2014-03-21","tz":"UTC"}
{"createdOn":"2014-03-19","publishedOn":"2014-03-19","tz":"UTC"}
```



Pull out a specific subfield with "dot" notation.

Cmd 19

```sql
%sql
SELECT dates.createdOn, dates.publishedOn
FROM DatabricksBlog
```

(1) Spark Jobs

createdOn	publishedOn
2014-04-10	2014-04-10
2014-04-10	2014-04-10
2014-04-01	2014-04-01
2014-03-27	2014-03-27
2014-02-04	2014-02-04
2014-01-02	2014-01-02
2014-03-26	2014-03-26
2014-03-21	2014-03-21
[illegible]	[illegible]

Both `createdOn` and `publishedOn` are stored as strings.

Cast those values to SQL timestamps:

In this case, use a single `SELECT` statement to:

1. Cast `dates.publishedOn` to a `timestamp` data type.
2. "Flatten" the `dates.publishedOn` column to just `publishedOn`.

Cmd 21

```
%sql
SELECT title,
       cast(dates.publishedOn AS timestamp) AS publishedOn
FROM DatabricksBlog
```

▸ (1) Spark Jobs

title	publishedOn
MapR Integrates the Complete Apache Spark Stack	2014-04-10T00:00:00.000+0000
Apache Spark 0.9.1 Released	2014-04-10T00:00:00.000+0000
Application Spotlight: Alpine Data Labs	2014-04-01T00:00:00.000+0000
Spark SQL: Manipulating Structured Data Using Apache Spark	2014-03-27T00:00:00.000+0000
Apache Spark 0.9.0 Released	2014-02-04T00:00:00.000+0000
Apache Spark In MapReduce (SIMR)	2014-01-02T00:00:00.000+0000
Sharethrough Uses Apache Spark Streaming to Optimize Bidding in Real Time	2014-03-26T00:00:00.000+0000
Apache Spark: A Delight for Developers	2014-03-21T00:00:00.000+0000

Cmd 22

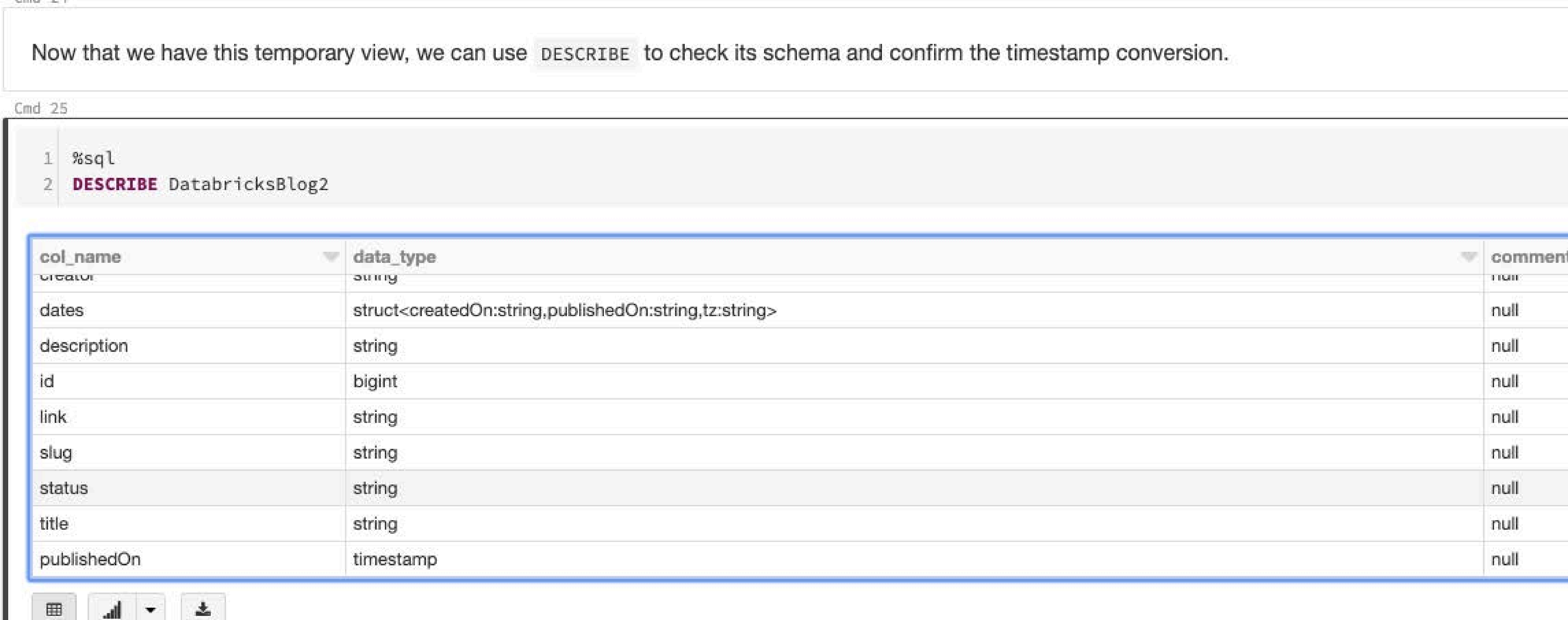
Create the temporary view `DatabricksBlog2` to capture the conversion and flattening of the `publishedOn` column.

Cmd 23

```
%sql
CREATE OR REPLACE TEMPORARY VIEW DatabricksBlog2 AS
  SELECT *,
         cast(dates.publishedOn AS timestamp) AS publishedOn
  FROM DatabricksBlog
```

OK

```
Command took 0.17 seconds -- by huseyinyilmaz01@gmail.com at 4/3/2020, 12:39:56 AM on test-cluster
```


Now that we have this temporary view, we can use `DESCRIBE` to check its schema and confirm the timestamp conversion.

Cmd 25

```
%sql
DESCRIBE DatabricksBlog2
```

col_name	data_type	comment
creator	string	null
dates	struct<createdOn:string,publishedOn:string,tz:string>	null
description	string	null
id	bigint	null
link	string	null
slug	string	null
status	string	null
title	string	null
publishedOn	timestamp	null

Now the dates are represented by a `timestamp` data type, query for articles within certain date ranges (such as getting a list of all articles published in 2013), and format the date for presentation purposes.

See the Spark documentation, built-in functions, for a long list of date-specific functions.

Cmd 27

```sql
%sql
SELECT title,
       date_format(publishedOn, "MMM dd, yyyy") AS date,
       link
FROM DatabricksBlog2
WHERE year(publishedOn) = 2013
ORDER BY publishedOn
```

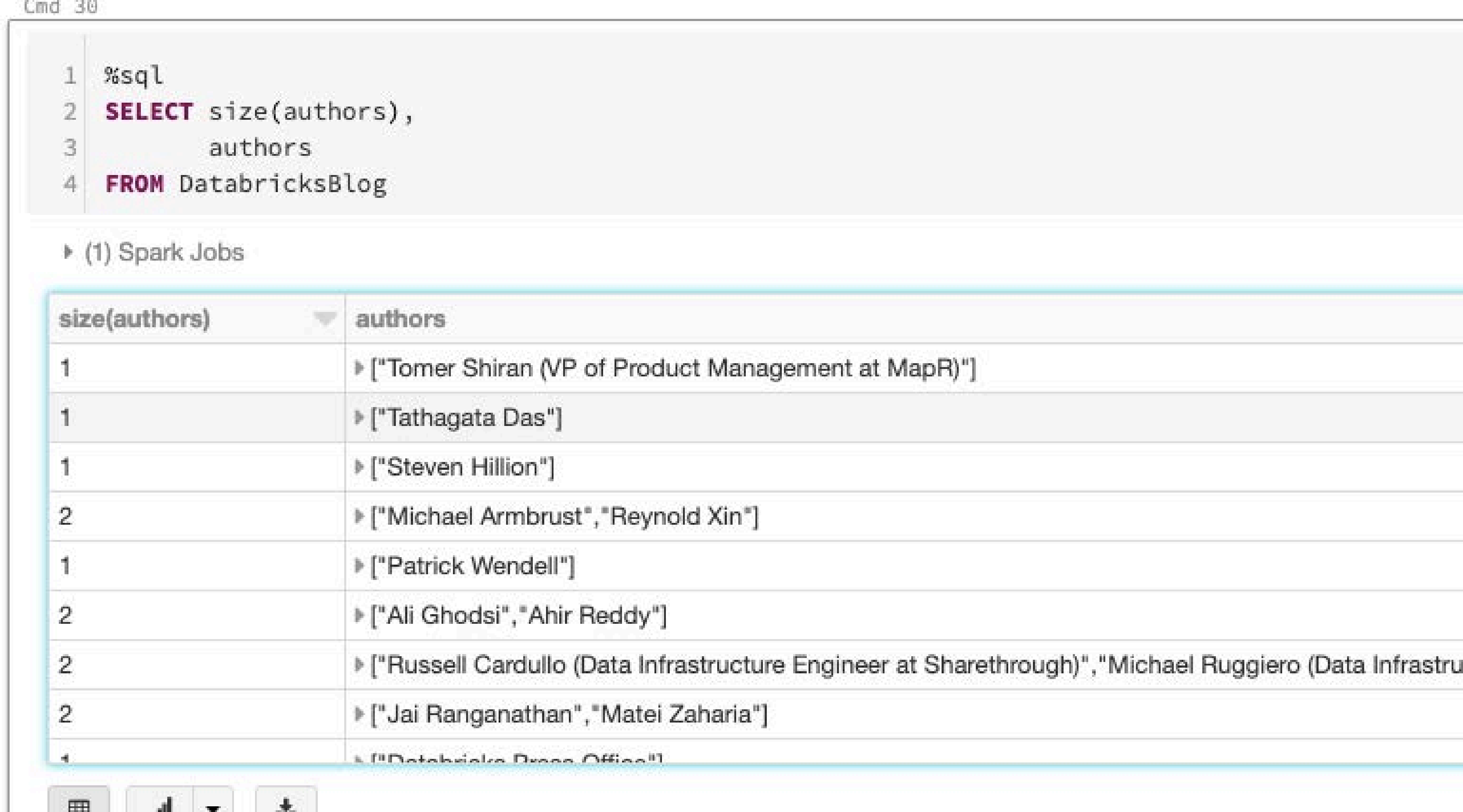
▸ (1) Spark Jobs

title	date	link
Databricks and the Apache Spark Platform	Oct 27, 2013	https://databricks.com/blog/2013/10/27/databricks-and-the-apache-spark-platform.html
The Growing Apache Spark Community	Oct 28, 2013	https://databricks.com/blog/2013/10/27/the-growing-spark-community.html
Databricks and Cloudera Partner to Support Apache Spark	Oct 29, 2013	https://databricks.com/blog/2013/10/28/databricks-and-cloudera-partner-to-support-spark.html
Putting Apache Spark to Use: Fast In-Memory Computing for Your Big Data Applications	Nov 22, 2013	https://databricks.com/blog/2013/11/21/putting-spark-to-use.html
Highlights From Spark Summit 2013	Dec 19, 2013	https://databricks.com/blog/2013/12/18/spark-summit-2013-follow-up.html
Apache Spark 0.8.1 Released	Dec 20, 2013	https://databricks.com/blog/2013/12/19/release-0_8_1.html

Array Data

The table also contains array columns.

Easily determine the size of each array using the built-in `size(..)` function with array columns.

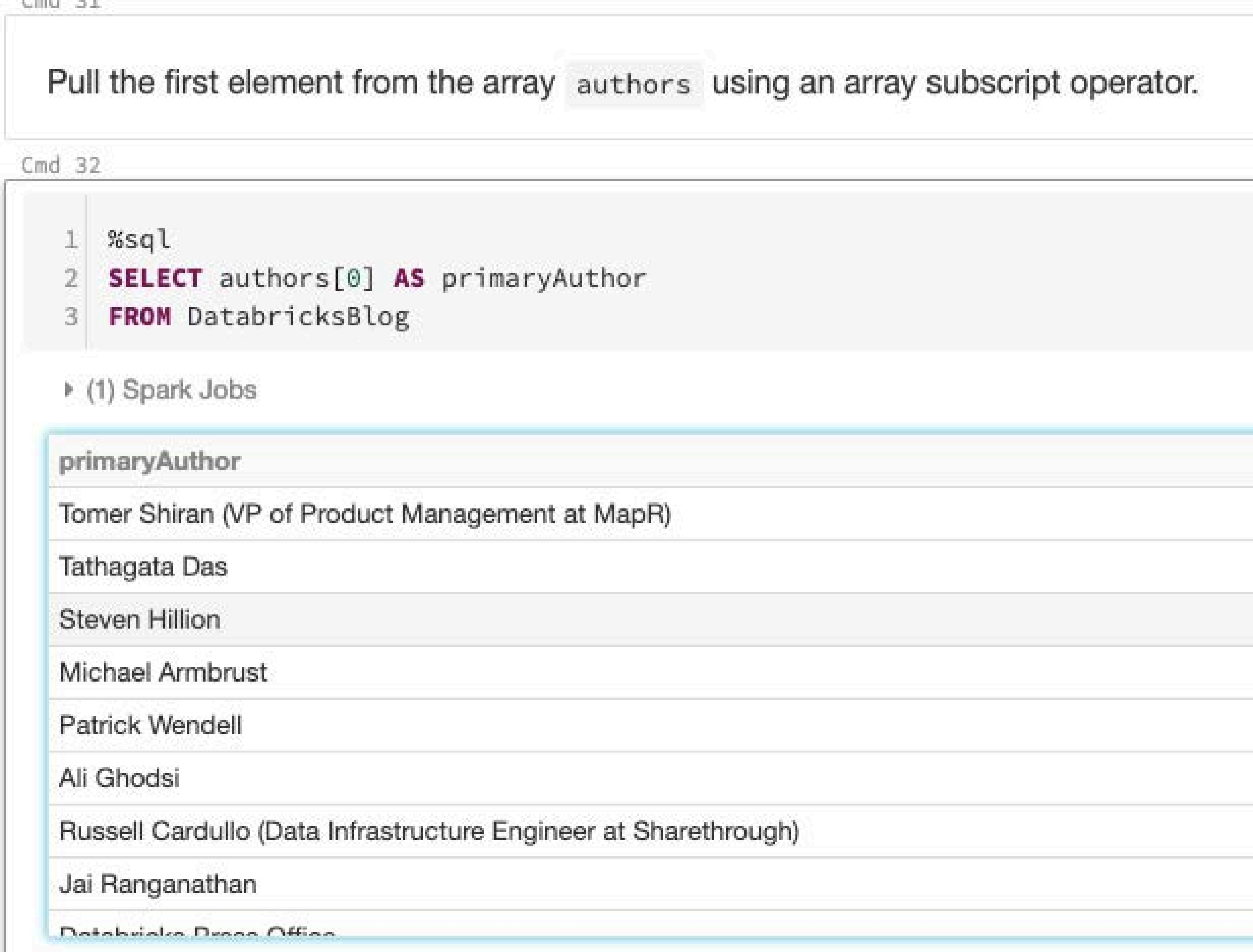


Cmd 30

```
%sql
SELECT size(authors),
       authors
FROM DatabricksBlog
```

(1) Spark Jobs

size(authors)	authors
1	["Tomer Shiran (VP of Product Management at MapR)"]
1	["Tathagata Das"]
1	["Steven Hillion"]
2	["Michael Armbrust","Reynold Xin"]
1	["Patrick Wendell"]
2	["Ali Ghodsi","Ahir Reddy"]
2	["Russell Cardullo (Data Infrastructure Engineer at Sharethrough)","Michael Ruggiero (Data Infrastru
2	["Jai Ranganathan","Matei Zaharia"]
1	["Databricks Press Office"]


Cmd 31

Pull the first element from the array `authors` using an array subscript operator.

Cmd 32

```
%sql
SELECT authors[0] AS primaryAuthor
FROM DatabricksBlog
```

▸ (1) Spark Jobs

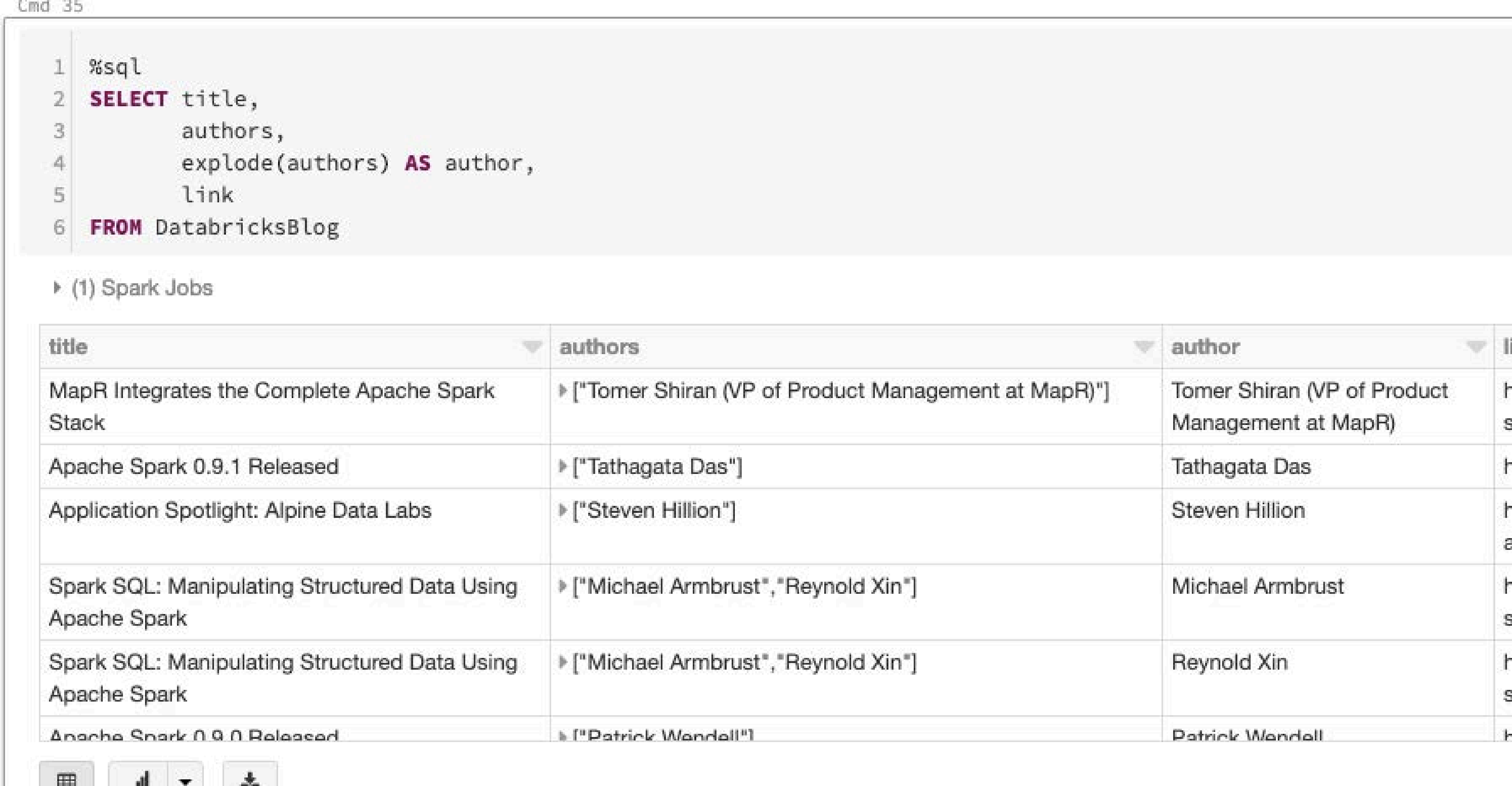
primaryAuthor
Tomer Shiran (VP of Product Management at MapR)
Tathagata Das
Steven Hillion
Michael Armbrust
Patrick Wendell
Ali Ghodsi
Russell Cardullo (Data Infrastructure Engineer at Sharethrough)
Jai Ranganathan
Databricks Press Office

Cmd 33

Explode

The `explode` function allows you to split an array column into multiple rows, copying all the other columns into each new row.

For example, you can split the column `authors` into the column `author`, with one author per row.


Cmd 35

```
%sql
SELECT title,
       authors,
       explode(authors) AS author,
       link
FROM DatabricksBlog
```

(1) Spark Jobs

title	authors	author	li
MapR Integrates the Complete Apache Spark Stack	["Tomer Shiran (VP of Product Management at MapR)"]	Tomer Shiran (VP of Product Management at MapR)	h s
Apache Spark 0.9.1 Released	["Tathagata Das"]	Tathagata Das	h
Application Spotlight: Alpine Data Labs	["Steven Hillion"]	Steven Hillion	h a
Spark SQL: Manipulating Structured Data Using Apache Spark	["Michael Armbrust","Reynold Xin"]	Michael Armbrust	h s
Spark SQL: Manipulating Structured Data Using Apache Spark	["Michael Armbrust","Reynold Xin"]	Reynold Xin	h s
Apache Spark 0.9.0 Released	["Patrick Wendell"]	Patrick Wendell	h

Cmd 36

It's more obvious to restrict the output to articles that have multiple authors, and sort by the title.

Cmd 37

```
%sql
SELECT title,
       authors,
       explode(authors) AS author,
       link
FROM DatabricksBlog
WHERE size(authors) > 1
ORDER BY title
```

▸ (1) Spark Jobs

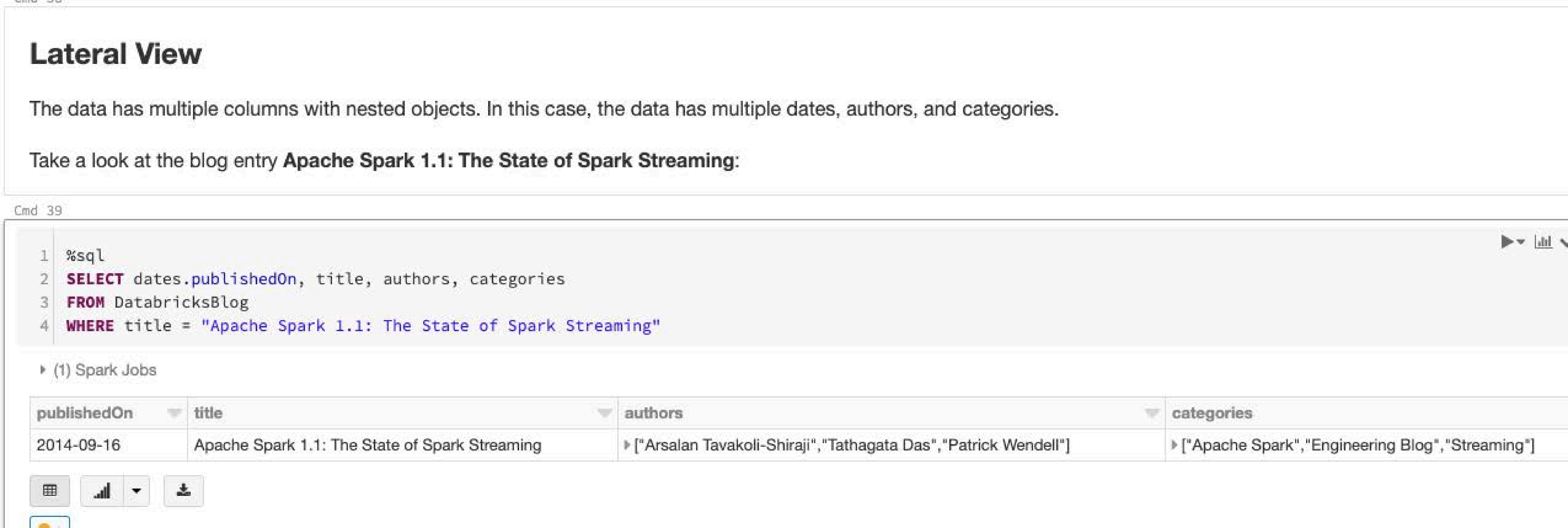
title	authors	author	lin
"Learning Spark" book available from O'Reilly	▸ ["Holden Karau","Andy Konwinski","Patrick Wendell","Matei Zaharia"]	Matei Zaharia	htt ava
"Learning Spark" book available from O'Reilly	▸ ["Holden Karau","Andy Konwinski","Patrick Wendell","Matei Zaharia"]	Holden Karau	htt ava
"Learning Spark" book available from O'Reilly	▸ ["Holden Karau","Andy Konwinski","Patrick Wendell","Matei Zaharia"]	Andy Konwinski	htt ava
"Learning Spark" book available from O'Reilly	▸ ["Holden Karau","Andy Konwinski","Patrick Wendell","Matei Zaharia"]	Patrick Wendell	htt ava
AMPLab updates the Big Data Benchmark	▸ ["Ahir Reddy","Reynold Xin"]	Ahir Reddy	htt
AMPLab updates the Big Data Benchmark	▸ ["Ahir Reddy","Reynold Xin"]	Reynold Xin	htt

Cmd 38

Lateral View

The data has multiple columns with nested objects. In this case, the data has multiple dates, authors, and categories.

Take a look at the blog entry **Apache Spark 1.1: The State of Spark Streaming**:



Lateral View

The data has multiple columns with nested objects. In this case, the data has multiple dates, authors, and categories.

Take a look at the blog entry **Apache Spark 1.1: The State of Spark Streaming**:

Cmd 39

```
%sql
SELECT dates.publishedOn, title, authors, categories
FROM DatabricksBlog
WHERE title = "Apache Spark 1.1: The State of Spark Streaming"
```

(1) Spark Jobs

publishedOn	title	authors	categories
2014-09-16	Apache Spark 1.1: The State of Spark Streaming	["Arsalan Tavakoli-Shiraji","Tathagata Das","Patrick Wendell"]	["Apache Spark","Engineering Blog","Streaming"]

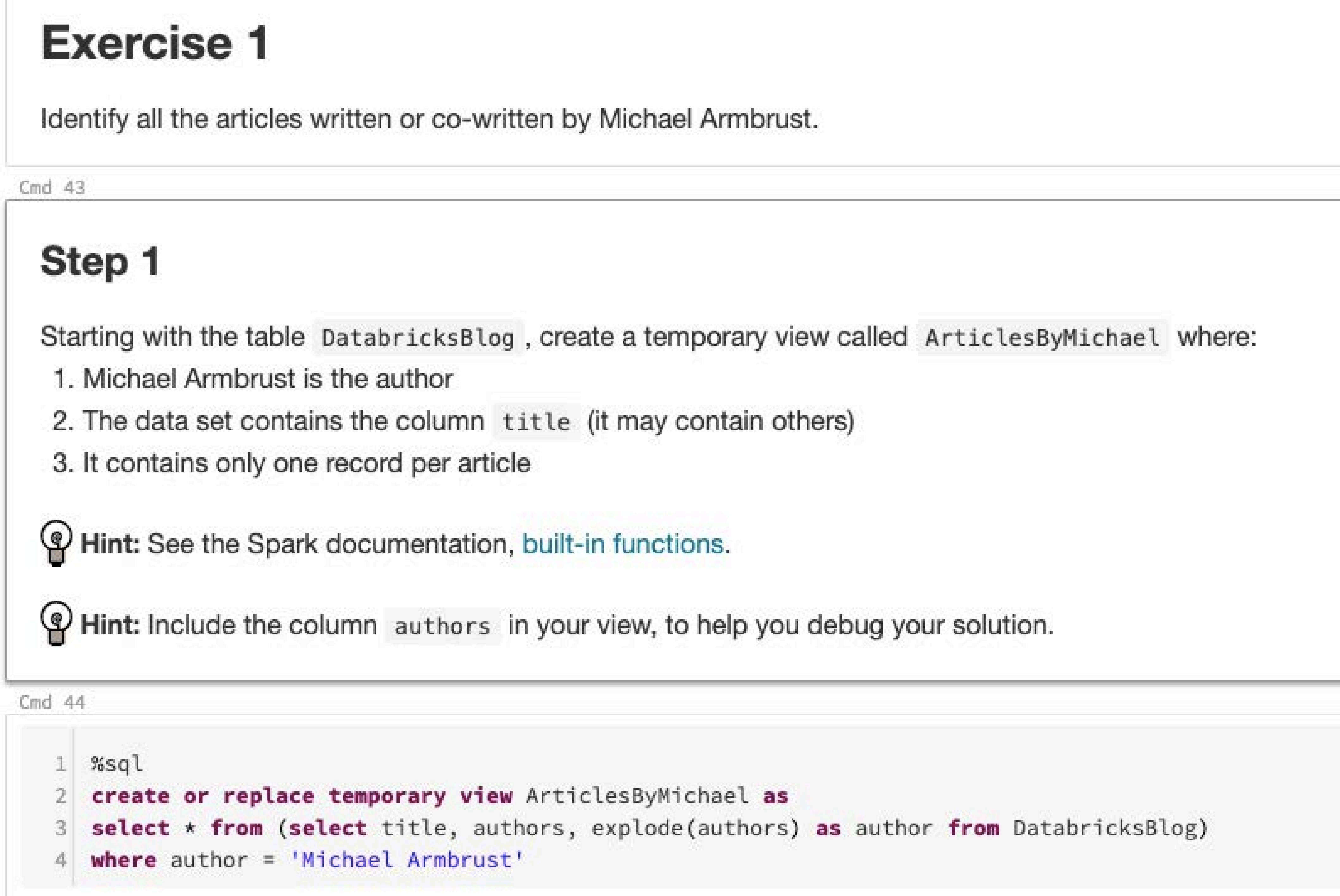
Next, use `LATERAL VIEW` to explode multiple columns at once, in this case, the columns `authors` and `categories` .

Cmd 41

```sql
%sql
SELECT dates.publishedOn, title, author, category
FROM DatabricksBlog
LATERAL VIEW explode(authors) exploded_authors_view AS author
LATERAL VIEW explode(categories) exploded_categories AS category
WHERE title = "Apache Spark 1.1: The State of Spark Streaming"
ORDER BY author, category
```

(1) Spark Jobs

publishedOn	title	author	category
2014-09-16	Apache Spark 1.1: The State of Spark Streaming	Arsalan Tavakoli-Shiraji	Apache Spark
2014-09-16	Apache Spark 1.1: The State of Spark Streaming	Arsalan Tavakoli-Shiraji	Engineering Blog
2014-09-16	Apache Spark 1.1: The State of Spark Streaming	Arsalan Tavakoli-Shiraji	Streaming
2014-09-16	Apache Spark 1.1: The State of Spark Streaming	Patrick Wendell	Apache Spark
2014-09-16	Apache Spark 1.1: The State of Spark Streaming	Patrick Wendell	Engineering Blog
2014-09-16	Apache Spark 1.1: The State of Spark Streaming	Patrick Wendell	Streaming
2014-09-16	Apache Spark 1.1: The State of Spark Streaming	Tathagata Das	Apache Spark
2014-09-16	Apache Spark 1.1: The State of Spark Streaming	Tathagata Das	Engineering Blog
2014-09-16	Apache Spark 1.1: The State of Spark Streaming	Tathagata Das	Streaming



Exercise 1

Identify all the articles written or co-written by Michael Armbrust.

Cmd 43

Step 1

Starting with the table `DatabricksBlog`, create a temporary view called `ArticlesByMichael` where:

1. Michael Armbrust is the author
2. The data set contains the column `title` (it may contain others)
3. It contains only one record per article

Hint: See the Spark documentation, built-in functions.

Hint: Include the column `authors` in your view, to help you debug your solution.

Cmd 44

```sql
%sql
create or replace temporary view ArticlesByMichael as
select * from (select title, authors, explode(authors) as author from DatabricksBlog)
where author = 'Michael Armbrust'
```

Exercise 2

Identify the complete set of categories used in the Databricks blog articles.

Cmd 49

Step 1

Starting with the table `DatabricksBlog`, create another view called `UniqueCategories` where:

1. The data set contains the one column `category` (and no others)
2. This list of categories should be unique

Cmd 50

```
%sql
create or replace temporary view UniqueCategories as
select distinct explode(categories) as category from DatabricksBlog
```

```
OK
```

Exercise 3

Count how many times each category is referenced in the Databricks blog.

Cmd 57

Step 1

Starting with the table `DatabricksBlog` , create a temporary view called `TotalArticlesByCategory` where:

1. The new table contains two columns, `category` and `total`
2. The `category` column is a single, distinct category (similar to the last exercise)
3. The `total` column is the total number of articles in that category

Hint: You need either multiple views or a `LATERAL VIEW` to solve this.

Because articles can be tagged with multiple categories, the sum of the totals adds up to more than the total number of articles.

Cmd 58

```
%sql
create or replace temporary view TotalArticlesByCategory as
select category, count(category) as total from (select explode(categories) as category from DatabricksBlog)
group by category
```

OK

```
Command took 0.41 seconds -- by huseyinyilmaz01@gmail.com at 4/3/2020, 1:53:55 PM on test-cluster
```

Summary

- Spark SQL allows you to query and manipulate structured and semi-structured data
- Spark SQL's built-in functions provide powerful primitives for querying complex schemas

Cmd 63

Review Questions

Q: What is the syntax for accessing nested columns?

A: Use the dot notation: `SELECT dates.publishedOn`

Q: What is the syntax for accessing the first element in an array?

A: Use the [subscript] notation: `SELECT authors[0]`

Q: What is the syntax for expanding an array into multiple rows?

A: Use the explode keyword, either:

`SELECT explode(authors) as Author` or

`LATERAL VIEW explode(authors) exploded_authors_view AS author`