



MySQL and JSON

LECTURE 3: JSON ARRAYS AND OBJECTS IN MYSQL

JavaScript Object Notation

2

- ▶ **JavaScript Object Notation** (JSON) is an open standard file format and data interchange format, that uses human-readable text to store and transmit data objects consisting of attribute-value pairs and array data types.
- ▶ JSON is a **language-independent** data format.
 - ▶ It was derived from JavaScript, but MySQL 5.7 and later includes code to generate and parse JSON-format data.
 - ▶ <https://dev.mysql.com/doc/refman/5.7/en/json.html>
- ▶ **JSON Data types:**
 - ▶ **Number:** (1 or 1.1)
 - ▶ **String:** "Gene"
 - ▶ **Boolean:** true or false
 - ▶ **Array:** [1,2,3,4] ...an ordered list of zero or more values each of which may be a any data type
 - ▶ **Object:** {"FName": "Gene", "LName": "Locklear"}
 - ▶ **null:** an empty value
- ▶ JSON allows us to store data as **'structure of values'** the data.

Defining JSON Values

3

- ▶ A **JSON array** contains a list of values separated by commas and enclosed within `[]`.
 - ▶ `["Stormtroopers",100,null,true]` ...can be a mixture of data types.
 - ▶ `[1,2,3,4,5]` ...can be a single data type.
 - ▶ `[[1,2,3],[4,5,6],[7,8,9]]` ...can nest arrays inside of arrays.
- ▶ A **JSON object** contains a set of key-value pairs separated by commas and enclosed with `{ }`.
 - ▶ `{"TrooperID": "FINN-01", "TrooperAge": 25}` ...key value pairs (Keys must be Strings).
 - ▶ `{"StartDate": "2020-01-01", "EndDate": "2020-05-01"}` ...can use MySQL date and time as values.
 - ▶ `{"TrooperID": "FINN-01", "TrooperLocation": [10,20]}` ...a value for a key may be an array.
 - ▶ `{"NewTrooper": {"TrooperID": "BLIS-11", "TrooperAge": 23}}` ...a value for a key may be another object.

The Array Concept

4

- ▶ Arrays are a sequence of values.

0	1	2	3	4
8	4	2	9	76

Index

Value

0	1	2
'A'	'Gene'	2

Index

Value

- ▶ Arrays can have a numeric index or a non-numeric index

0	1	2	3	4
8	4	2	9	76

Index

Value

Numeric Array or standard Array

JSON Array

'ID'	'Name'	'Status'
1008	'Gene'	'Online'

Index

Value

Associative Array or Map

JSON Object

Indexing an Numeric Array

5

0	1	2	3	4
8	4	2	9	76

Diagram illustrating a numeric array with indices 0 to 4 and corresponding values 8, 4, 2, 9, 76. Red arrows point from the 'Index' label to the top row and from the 'Value' label to the bottom row.

0	1	2
'A'	'Gene'	2

Diagram illustrating a string array with indices 0 to 2 and corresponding values 'A', 'Gene', 2. Red arrows point from the 'Index' label to the top row and from the 'Value' label to the bottom row.

```
C:\Users\msgth\Desktop\A_Database_PACE_2020\LocklearJSON2.sql - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
LocklearJSON.sql x LocklearJSON2.sql x
1 USE jsonTesting;
2
3 SELECT JSON_ARRAY(8,4,2,9,76);
4 SELECT JSON_ARRAY('A','Gene',2);
5 SELECT JSON_EXTRACT(JSON_ARRAY(8,4,2,9,76),'$[3]');
6 SELECT JSON_EXTRACT('["A","Gene",2]','$[1]');
```

We use the \$ sign to refer to the JSON array

Diagram illustrating the SQL code in a Sublime Text editor. Red arrows point from the 'Index' label to the '\$[3]' and '\$[1]' expressions in the SQL queries.

```
MySQL 5.7 Command Line Client
Database changed
+-----+
| JSON_ARRAY(8,4,2,9,76) |
+-----+
| [8, 4, 2, 9, 76] |
+-----+
1 row in set (0.00 sec)

+-----+
| JSON_ARRAY('A','Gene',2) |
+-----+
| ["A", "Gene", 2] |
+-----+
1 row in set (0.00 sec)

+-----+
| JSON_EXTRACT(JSON_ARRAY(8,4,2,9,76),'$[3]') |
+-----+
| 9 |
+-----+
1 row in set (0.00 sec)

+-----+
| JSON_EXTRACT('["A","Gene",2]','$[1]') |
+-----+
| "Gene" |
+-----+
```

The Nested Array Concept

6

- ▶ Arrays are a sequence of values and sometimes those values can be another sequence of values.

0	1	2	3	4
8	[1,4,3]	2	[9,1,2]	76

```
1 USE jsonTesting;
2
3 SELECT JSON_ARRAY('[8,[1,4,3],[9,1,2],76]');
4 SELECT JSON_EXTRACT('[8,[1,4,3],[9,1,2],76]','$[1][1]');
```

Line 5, Column 1 [Index][Index] Tab Size: 4

```
MySQL 5.7 Command Line Client
+-----+
| JSON_ARRAY('[8,[1,4,3],[9,1,2],76]') |
+-----+
| "[8,[1,4,3],[9,1,2],76]"           |
+-----+
1 row in set (0.00 sec)

+-----+
| JSON_EXTRACT('[8,[1,4,3],[9,1,2],76]','$[1][1]') |
+-----+
| 4 |
+-----+
1 row in set (0.00 sec)

mysql>
```

JSON Arrays

7

- ▶ MySQL contains functions that create and manipulate JSON Arrays.

```
C:\Users\msgth\Desktop\A_Database_PACE_2020\LocklearJSON.sql - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

LocklearFunctions.sql x LocklearJSON.sql

1 DROP DATABASE IF EXISTS jsonTesting;
2 CREATE DATABASE jsonTesting;
3 USE jsonTesting;
4
5
6 -- Create a JSON Array from values
7 SELECT JSON_ARRAY(1,2,3,4,5);
8
9 -- Assign a JSON Array to user-defined session variable
10 SET @MyList = JSON_ARRAY(1,2,3,4,5);
11 SELECT @MyList;
12
13 -- Select a value within a JSON Array
14 SELECT JSON_EXTRACT('[1,2,3,4,5]','$[0]');
15
16 -- Select all values within a JSON Array
17 SELECT JSON_EXTRACT('[1,2,3,4,5]','$[*]');
18
```

@MyList is just a String even though it looks like JSON

We use the \$ sign to refer to the JSON array

Within the [] we can specify an index for a specific value in the array. A * means all values. Indexing starts at 0

```
MySQL 5.7 Command Line Client

+-----+
| JSON_ARRAY(1,2,3,4,5) |
+-----+
| [1, 2, 3, 4, 5] |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

+-----+
| @MyList |
+-----+
| [1, 2, 3, 4, 5] |
+-----+
1 row in set (0.00 sec)

+-----+
| JSON_EXTRACT('[1,2,3,4,5]','$[0]') |
+-----+
| 1 |
+-----+
1 row in set (0.00 sec)

+-----+
| JSON_EXTRACT('[1,2,3,4,5]','$[*]') |
+-----+
| [1, 2, 3, 4, 5] |
+-----+
1 row in set (0.00 sec)

mysql>
```


8

- The screenshot shows a Sublime Text editor window with the file path `C:\Users\msgth\Desktop\A_Database_PACE_2020\LocklearJSON.sql`. The editor contains the following SQL code:

```
19 -- Create an Array that contains another Array
20 SELECT JSON_ARRAY(''[1,2,3,4,5]','[6,7,8,9,10]');
21
22 -- Select an Array from an Array containing other arrays
23 SELECT JSON_EXTRACT('[[1,2,3,4,5],[6,7,8,9,10]]','$[1]');
24
25 -- Select an Value from an Array containing other arrays
26 SELECT JSON_EXTRACT('[[1,2,3,4,5],[6,7,8,9,10]]','$[1][2]');
27
```

Three red arrows point from text boxes to specific parts of the code:

 - An arrow points from the text box "An array of arrays" to the second argument `'[6,7,8,9,10]'` in the `JSON_ARRAY` function on line 20.
 - An arrow points from the text box "Index [1] refers to the second array" to the `$[1]` index in the `JSON_EXTRACT` function on line 23.
 - An arrow points from the text box "Index [1][2] refers to the third value in the second array" to the `$[1][2]` index in the `JSON_EXTRACT` function on line 26.

The status bar at the bottom indicates "Line 26, Column 58" and "Tab Size: 4".

[illegible]

JSON Arrays

9

- ▶ We can assign JSON arrays to user-defined session variables but that **become a String** and **cannot be used as JSON values**.

```
C:\Users\msgth\Desktop\A_Database_PACE_2020\LocklearJSON.sql - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

LocklearFunctions.sql x LocklearJSON.sql x

28
29 -- Create an Array that contains another Array
30 SELECT JSON_ARRAY(''[1,2,3,4,5]', '[6,7,8,9,10]]'');
31 -- Select an Array from an Array containing other arrays
32 SELECT JSON_EXTRACT(''[1,2,3,4,5],[6,7,8,9,10]]', '$[1]');
33 -- Assign a JSON Array to a user-defined sessions variable
34 SELECT JSON_ARRAY(''[1,2,3,4,5]', '[5,6,7,8,9,10]]'') INTO @V;
35 -- display the contents of @V
36 SELECT @V;
37 -- Attempt to extract values from @V using JSON function
38 SELECT JSON_EXTRACT(@V, '$[0]');
39 SELECT JSON_EXTRACT(@V, '$[1]');
40 -- Attempt to convert @V to JSON Array and extract values
41 SELECT JSON_EXTRACT(JSON_ARRAY(@V), '$[1]');
```

Once we assign a JSON value to a variable it becomes a String.

The JSON_EXTRACT function just sees this as a single String value.

```
MySQL 5.7 Command Line Client

Database changed
+-----+
| JSON_ARRAY(''[1,2,3,4,5]', '[6,7,8,9,10]]'') |
+-----+
| '[[1,2,3,4,5],[6,7,8,9,10]]'                |
+-----+
1 row in set (0.00 sec)

+-----+
| JSON_EXTRACT('[[1,2,3,4,5],[6,7,8,9,10]]', '$[1]') |
+-----+
| [6, 7, 8, 9, 10]                                   |
+-----+
1 row in set (0.00 sec)

Query OK, 1 row affected (0.00 sec)

+-----+
| @V                                             |
+-----+
| '[[1,2,3,4,5],[5,6,7,8,9,10]]'                |
+-----+
1 row in set (0.00 sec)

+-----+
| JSON_EXTRACT(@V, '$[0]')                       |
+-----+
| '[[1,2,3,4,5],[5,6,7,8,9,10]]'                |
+-----+
1 row in set (0.00 sec)

+-----+
| JSON_EXTRACT(@V, '$[1]')                       |
+-----+
| NULL                                           |
+-----+
1 row in set (0.00 sec)

+-----+
| JSON_EXTRACT(JSON_ARRAY(@V), '$[1]')           |
+-----+
| NULL                                           |
+-----+
1 row in set (0.00 sec)

mysql>
```

JSON Arrays

10

- ▶ We can store JSON arrays in tables.
- ▶ We can then access the entire array or parts of the array.

```
C:\Users\msgth\Desktop\A_Database_PACE_2020\LocklearJSON.sql - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
LocklearFunctions.sql x LocklearJSON.sql
44
45 CREATE TABLE R1 (
46 InfoID VARCHAR(20),
47 InfoArray JSON,
48 CONSTRAINT pk_R1 PRIMARY KEY(InfoID)
49 );
50
51 INSERT INTO R1 VALUES('A1','[[1,2,3,4,5],[5,6,7,8,9,10]]');
52 INSERT INTO R1 VALUES('B1','[[11,2,31,4,51],[5,61,7,81,9,10]]');
53 INSERT INTO R1 VALUES('C1','[[1,22,3,42,5],[52,6,72,8,92,10]]');
54
55 SELECT * FROM R1;
56
57 SELECT JSON_EXTRACT(InfoArray,'$[0]') AS 'Array 1 of A1'
58 FROM R1
59 WHERE InfoID = 'A1';
60
61 SELECT JSON_EXTRACT(InfoArray,'$[1][2]') AS 'Value 3 in Array 2 of A1'
62 FROM R1
63 WHERE InfoID = 'A1';
64
```

```
MySQL 5.7 Command Line Client
+-----+-----+
| InfoID | InfoArray |
+-----+-----+
| A1      | [[1, 2, 3, 4, 5], [5, 6, 7, 8, 9, 10]] |
| B1      | [[11, 2, 31, 4, 51], [5, 61, 7, 81, 9, 10]] |
| C1      | [[1, 22, 3, 42, 5], [52, 6, 72, 8, 92, 10]] |
+-----+-----+
3 rows in set (0.00 sec)

+-----+
| Array 1 of A1 |
+-----+
| [1, 2, 3, 4, 5] |
+-----+
1 row in set (0.00 sec)

+-----+
| Value 3 in Array 2 of A1 |
+-----+
| 7 |
+-----+
```

JSON Arrays

11

- ▶ We can create JSON arrays from data and then store them as JSON values in tables.
- ▶ If a JSON Array consist of nested arrays each must be created separately inside the actual JSON Array value.

The screenshot shows a Sublime Text editor window with the file 'LocklearJSON.sql' open. The code defines a table R2 with a primary key InfoID and a JSON array column InfoArray. It includes three INSERT statements: one for a simple array, one for a nested array, and one for a partially nested array. A MySQL Command Line Client window shows the table structure and data, confirming the successful insertion of the nested array. A yellow callout box points to the nested array in the SQL code and the MySQL output, stating: 'Each separate array in a nested JSON array must be created inside the JSON_ARRAY function.'

```
65 CREATE TABLE R2 (  
66 InfoID VARCHAR(20),  
67 InfoArray JSON,  
68 CONSTRAINT pk_R2 PRIMARY KEY(InfoID)  
69 );  
70  
71 INSERT INTO R2 VALUES('A1',JSON_ARRAY(1,2,3,4,5,6,7,8,9,10));  
72 INSERT INTO R2 VALUES('B1',JSON_ARRAY(JSON_ARRAY(1,2,3,4,5),JSON_ARRAY(6,7,8,9,10)));  
73 INSERT INTO R2 VALUES('C1',JSON_ARRAY(1,2,3,JSON_ARRAY(6,7,8,9,10)));  
74  
75 SELECT * FROM R2;
```

InfoID	InfoArray
A1	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
B1	[[1, 2, 3, 4, 5], [6, 7, 8, 9, 10]]
C1	[1, 2, 3, [6, 7, 8, 9, 10]]

3 rows in set (0.00 sec)

mysql>

Each separate array in a nested JSON array must be created inside the JSON_ARRAY function.

JSON Functions

12

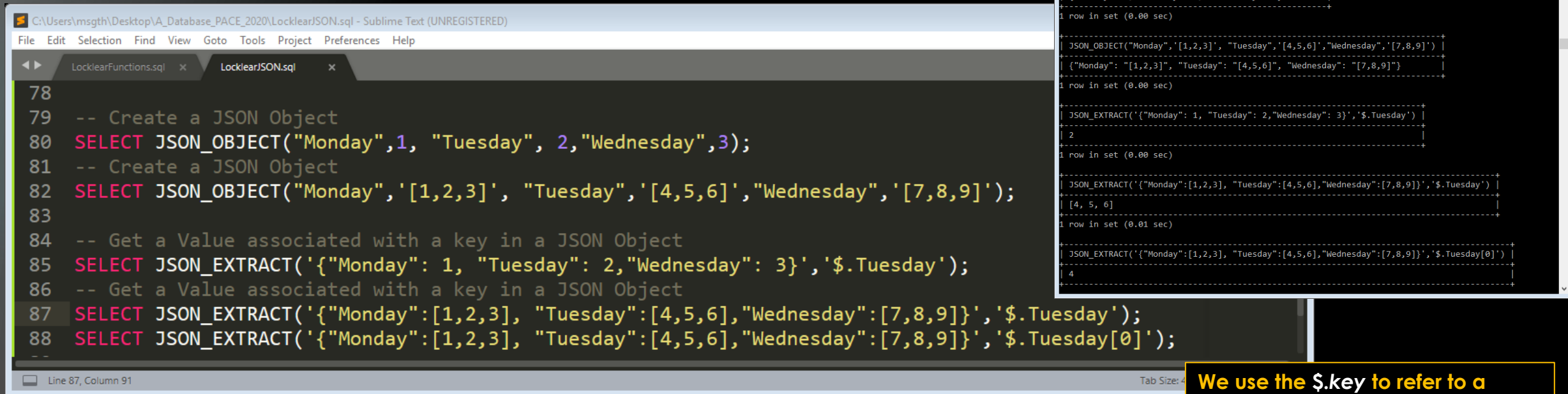
- ▶ There are many useful JSON Functions.
- ▶ <https://dev.mysql.com/doc/refman/5.7/en/json-function-reference.html>

Name	Description
->	Return value from JSON column after evaluating path; equivalent to JSON_EXTRACT().
->> (introduced 5.7.13)	Return value from JSON column after evaluating path and unquoting the result; equivalent to JSON_UNQUOTE(JSON_EXTRACT()).
<u>JSON_APPEND()</u> (deprecated)	Append data to JSON document
<u>JSON_ARRAY()</u>	Create JSON array
<u>JSON_ARRAY_APPEND()</u>	Append data to JSON document
<u>JSON_ARRAY_INSERT()</u>	Insert into JSON array
<u>JSON_CONTAINS()</u>	Whether JSON document contains specific object at path
<u>JSON_CONTAINS_PATH()</u>	Whether JSON document contains any data at path
<u>JSON_DEPTH()</u>	Maximum depth of JSON document
<u>JSON_EXTRACT()</u>	Return data from JSON document
<u>JSON_INSERT()</u>	Insert data into JSON document
<u>JSON_KEYS()</u>	Array of keys from JSON document
<u>JSON_LENGTH()</u>	Number of elements in JSON document
<u>JSON_MERGE()</u> (deprecated 5.7.22)	Merge JSON documents, preserving duplicate keys. Deprecated synonym for JSON_MERGE_PRESERVE()
<u>JSON_MERGE_PATCH()</u> (introduced 5.7.22)	Merge JSON documents, replacing values of duplicate keys
<u>JSON_MERGE_PRESERVE()</u> (introduced 5.7.22)	Merge JSON documents, preserving duplicate keys
<u>JSON_OBJECT()</u>	Create JSON object
<u>JSON_PRETTY()</u> (introduced 5.7.22)	Print a JSON document in human-readable format
<u>JSON_QUOTE()</u>	Quote JSON document
<u>JSON_REMOVE()</u>	Remove data from JSON document
<u>JSON_REPLACE()</u>	Replace values in JSON document
<u>JSON_SEARCH()</u>	Path to value within JSON document
<u>JSON_SET()</u>	Insert data into JSON document
<u>JSON_STORAGE_SIZE()</u> (introduced 5.7.22)	Space used for storage of binary representation of a JSON document
<u>JSON_TYPE()</u>	Type of JSON value
<u>JSON_UNQUOTE()</u>	Unquote JSON value
<u>JSON_VALID()</u>	Whether JSON value is valid

JSON Objects

13

- ▶ MySQL contains functions that create and manipulate JSON Objects.
- ▶ JSON Objects consist of key-value pairs and are referred to as **Associative Arrays**.
- ▶ Unlike a standard array a JSON Object (associative array) uses keys rather than a numerical index to reference its values.



```
C:\Users\msgth\Desktop\A_Database_PACE_2020\LocklearJSON.sql - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
LocklearFunctions.sql x LocklearJSON.sql x
78
79 -- Create a JSON Object
80 SELECT JSON_OBJECT("Monday",1, "Tuesday", 2,"Wednesday",3);
81 -- Create a JSON Object
82 SELECT JSON_OBJECT("Monday",'[1,2,3]', "Tuesday",'[4,5,6]','Wednesday','[7,8,9]');
83
84 -- Get a Value associated with a key in a JSON Object
85 SELECT JSON_EXTRACT('{ "Monday": 1, "Tuesday": 2,"Wednesday": 3}','$.Tuesday');
86 -- Get a Value associated with a key in a JSON Object
87 SELECT JSON_EXTRACT('{ "Monday":[1,2,3], "Tuesday":[4,5,6],"Wednesday":[7,8,9]}','$.Tuesday');
88 SELECT JSON_EXTRACT('{ "Monday":[1,2,3], "Tuesday":[4,5,6],"Wednesday":[7,8,9]}','$.Tuesday[0]');
```

```
MySQL 5.7 Command Line Client
+-----+
| JSON_OBJECT("Monday",1, "Tuesday", 2,"Wednesday",3) |
+-----+
| {"Monday": 1, "Tuesday": 2, "Wednesday": 3}           |
+-----+
1 row in set (0.00 sec)

+-----+
| JSON_OBJECT("Monday",'[1,2,3]', "Tuesday",'[4,5,6]','Wednesday','[7,8,9]') |
+-----+
| {"Monday": "[1,2,3]", "Tuesday": "[4,5,6]", "Wednesday": "[7,8,9]"}         |
+-----+
1 row in set (0.00 sec)

+-----+
| JSON_EXTRACT('{ "Monday": 1, "Tuesday": 2,"Wednesday": 3}','$.Tuesday') |
+-----+
| 2                               |
+-----+
1 row in set (0.00 sec)

+-----+
| JSON_EXTRACT('{ "Monday": "[1,2,3]", "Tuesday": "[4,5,6]","Wednesday": "[7,8,9]"}','$.Tuesday') |
+-----+
| [4, 5, 6]                       |
+-----+
1 row in set (0.01 sec)

+-----+
| JSON_EXTRACT('{ "Monday": "[1,2,3]", "Tuesday": "[4,5,6]","Wednesday": "[7,8,9]"}','$.Tuesday[0]') |
+-----+
| 4                               |
+-----+
```

We use the \$.key to refer to a specific key in the JSON Object.

JSON Objects

14

- ▶ We can store JSON Objects in tables

```
C:\Users\msgth\Desktop\A_Database_PACE_2020\LocklearJSON.sql - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

LocklearFunctions.sql x LocklearJSON.sql x

90
91 CREATE TABLE R3 (
92   InfoID VARCHAR(20),
93   InfoArray JSON,
94   CONSTRAINT pk_R3 PRIMARY KEY(InfoID)
95 );
96
97 INSERT INTO R3 VALUES('J01',JSON_OBJECT("Monday",JSON_ARRAY(1,2,3), "Tuesday",JSON_ARRAY(4,5,6)));
98 INSERT INTO R3 VALUES('J02',JSON_OBJECT("TrooperID",'ST-1', "TrooperAge",26));
99 INSERT INTO R3 VALUES('J03',JSON_OBJECT("TrooperID",JSON_ARRAY(1,2,JSON_ARRAY(4,5,6))));
100
101
102 SELECT * FROM R3;
103
104 SELECT JSON_EXTRACT(InfoArray,'$.Tuesday[0]') AS 'Value of Key Tuesday in J01'
105 FROM R3
106 WHERE InfoID = 'J01';
107
108 SELECT JSON_EXTRACT(InfoArray,'$.TrooperID') AS 'ID', JSON_EXTRACT(InfoArray,'$.TrooperAge') AS 'AGE'
109 FROM R3
110 WHERE InfoID = 'J02';
111
112 SELECT JSON_EXTRACT(InfoArray,'$.TrooperID[0]') AS 'First Element of Value'
113 , JSON_EXTRACT(InfoArray,'$.TrooperID[2][1]') AS '2nd Number of Second Element of Value'
114 FROM R3
115 WHERE InfoID = 'J03';
```

```
MySQL 5.7 Command Line Client

+-----+-----+
| InfoID | InfoArray |
+-----+-----+
| J01    | {"Monday": [1, 2, 3], "Tuesday": [4, 5, 6]} |
| J02    | {"TrooperID": "ST-1", "TrooperAge": 26} |
| J03    | {"TrooperID": [1, 2, [4, 5, 6]]} |
+-----+-----+
3 rows in set (0.00 sec)

+-----+-----+
| Value of Key Tuesday in J01 |
+-----+-----+
| 4 |
+-----+-----+
1 row in set (0.00 sec)

+-----+-----+
| ID | AGE |
+-----+-----+
| "ST-1" | 26 |
+-----+-----+
1 row in set (0.00 sec)

+-----+-----+
| First Element of Value | 2nd Number of Second Element of Value |
+-----+-----+
| 1 | 5 |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```