IS664 Database Programming
Fall 2022

Database Programming

LECTURE 6: USING JSON

Professor HG Locklear
hlocklear@pace.edu

JavaScript Object Notation

- 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/.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

- 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

- 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 a Numeric Array

0	1	2	3	4
8	4	2	9	76

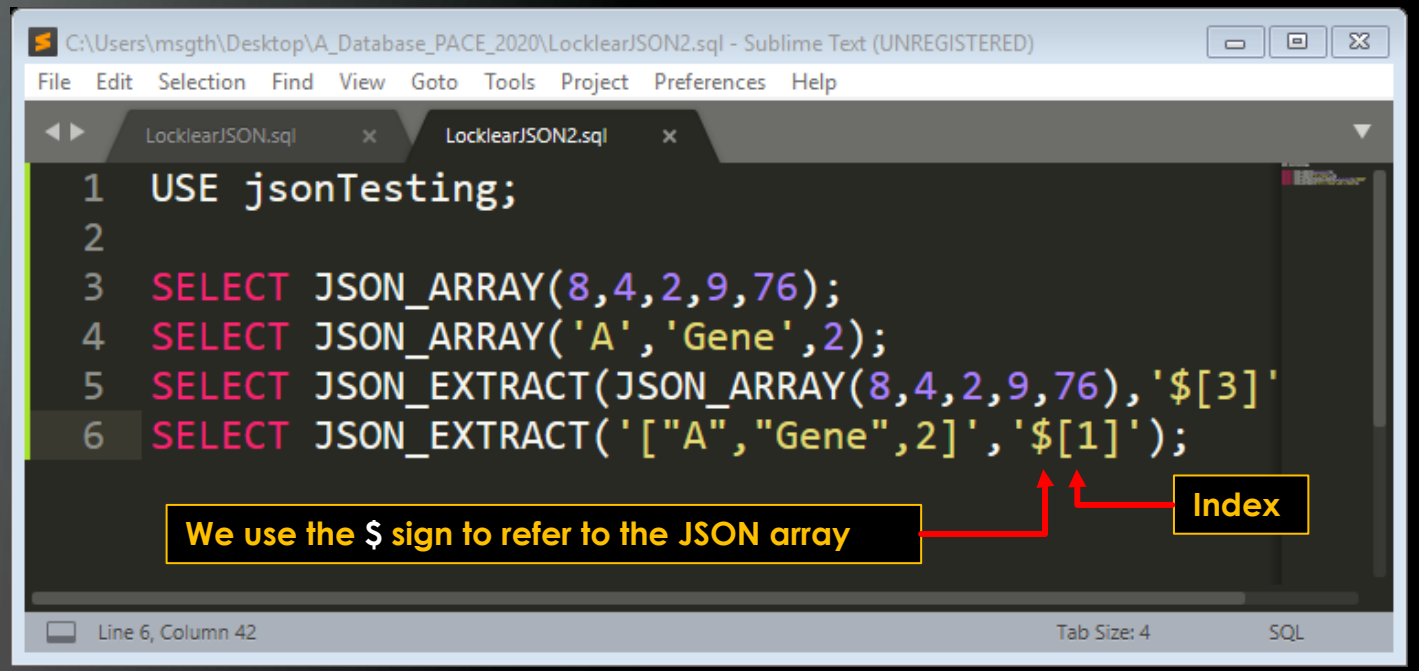
Index

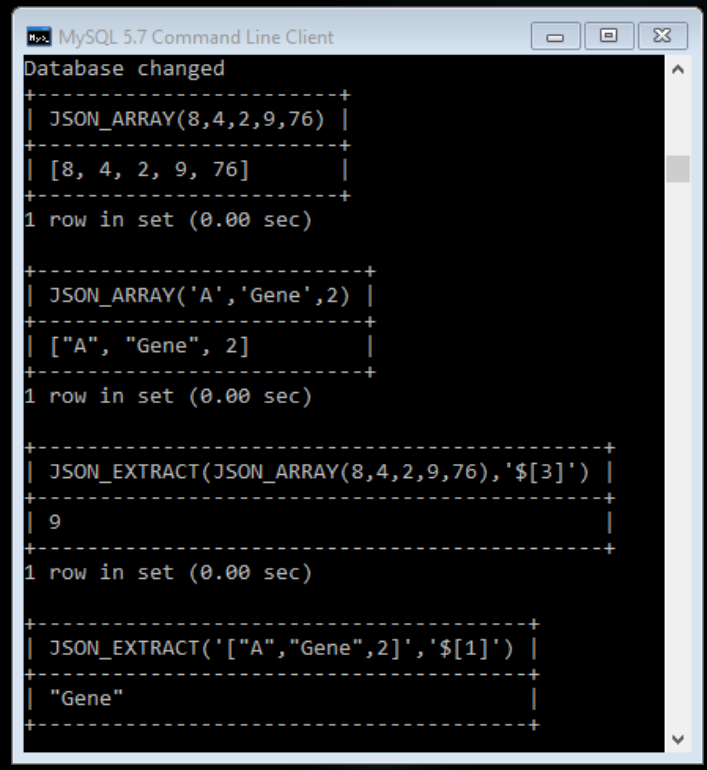
Value

0	1	2
‘A’	‘Gene’	2

Index

Value

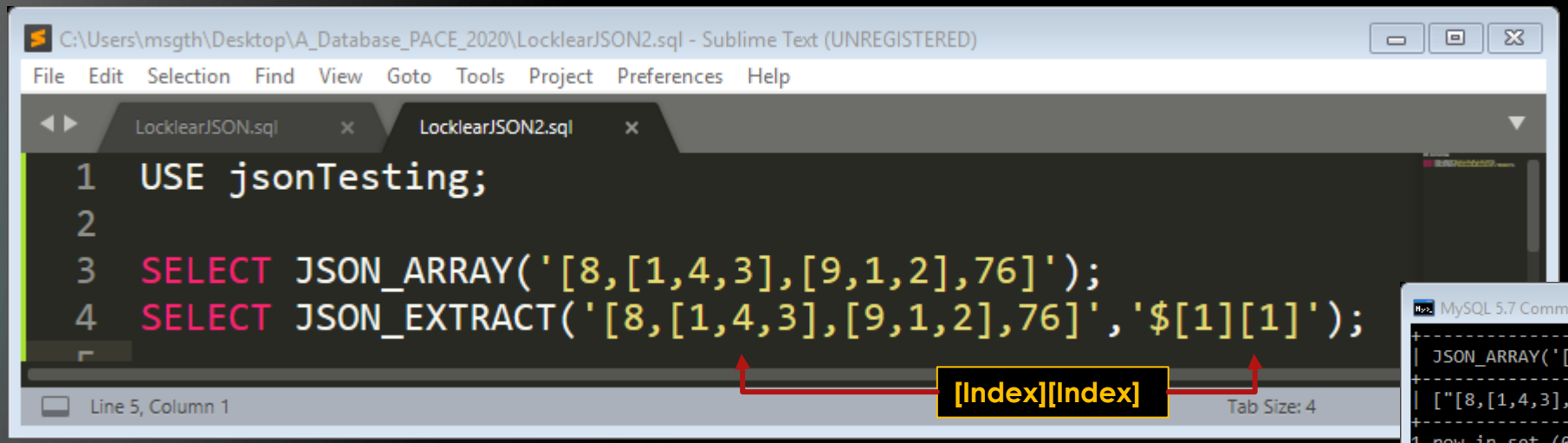


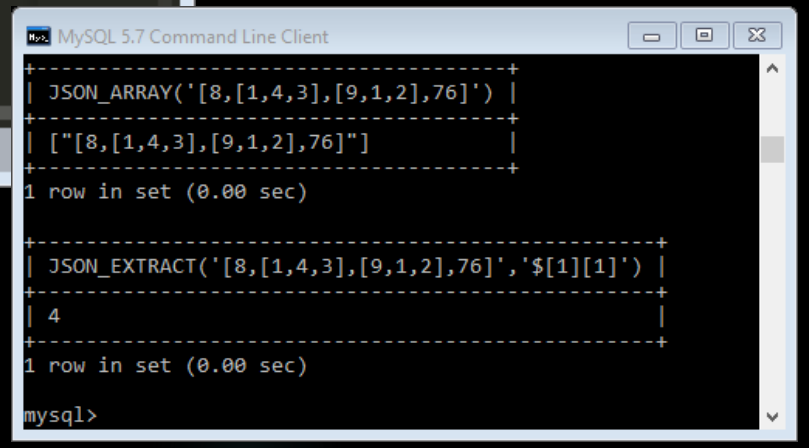


The Nested Array Concept

- 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





JSON Arrays

- 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 | 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
```

Line 17, Column 43

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

We use the $ sign to refer to the JSON array

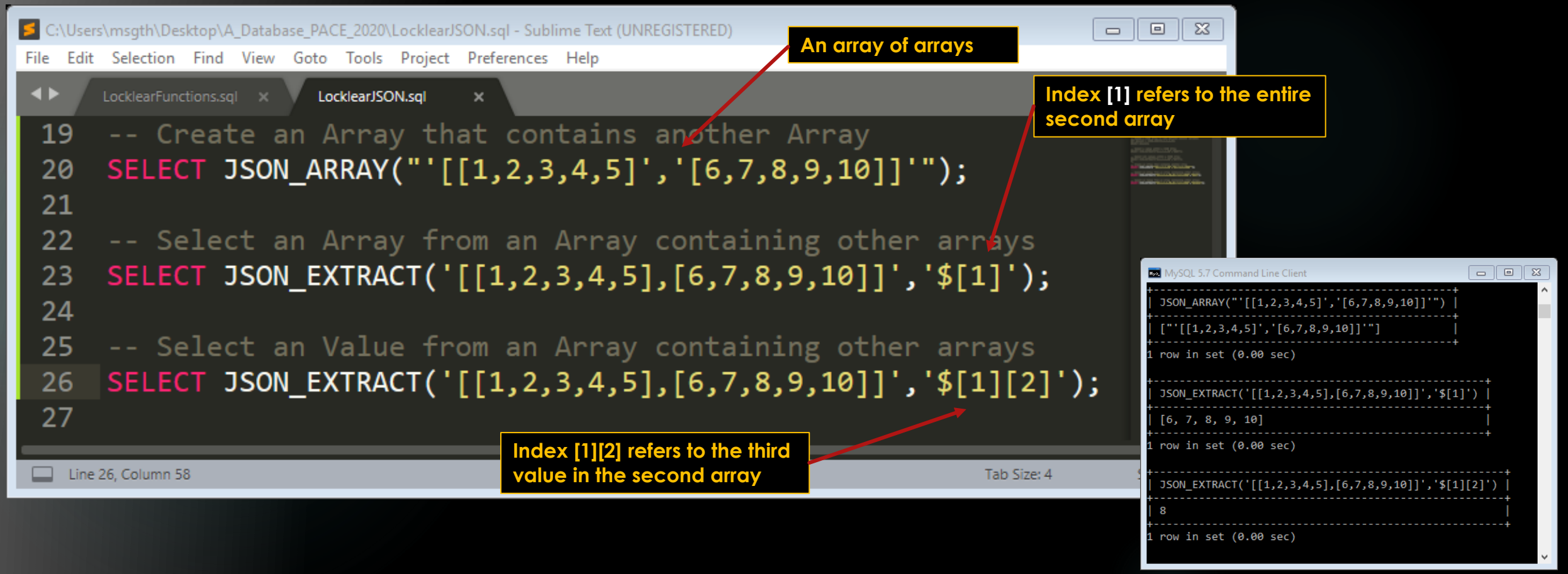
Within the [ ] we can specify an index for a specific value in the array. A * means all values. Indexing starts at 0

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>
```

JSON Arrays

- We can create JSON arrays that contain other JSON arrays.
- We utilizing indexing to access all the values within JSON arrays.



JSON Arrays

- 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 | LocklearJSON.sql

```
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]');
```

Line 39, Column 26

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

- 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 | 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';
```

Line 43, Column 1 Tab Size: 4 SQL

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

- 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.

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 | LocklearJSON.sql

```
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;
```

Line 75, Column 18 Tab Size: 4

MySQL 5.7 Command Line Client

```
+--------+--------------------------------------+
| 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

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

JSON Objects

- 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 | LocklearJSON.sql

```
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]');
```

Line 87, Column 91 Tab Size: 4

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

- 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 | LocklearJSON.sql

```
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('JO1',JSON_OBJECT("Monday",JSON_ARRAY(1,2,3), "Tuesday",JSON_ARRAY(4,5,6)));
98  INSERT INTO R3 VALUES('JO2',JSON_OBJECT("TrooperID",'ST-1', "TrooperAge",26));
99  INSERT INTO R3 VALUES('JO3',JSON_OBJECT("TrooperID",JSON_ARRAY(1,2,JSON_ARRAY(4,5,6))));
100
101
102 SELECT * FROM R3;
103
104 SELECT JSON_EXTRACT(InfoArray,'$.Tuesday[0]') AS 'Value of Key Tuesday in JO1'
105 FROM R3
106 WHERE InfoID = 'JO1';
107
108 SELECT JSON_EXTRACT(InfoArray,'$.TrooperID') AS 'ID', JSON_EXTRACT(InfoArray,'$.TrooperAge') AS 'AGE'
109 FROM R3
110 WHERE InfoID = 'JO2';
111
112 SELECT JSON_EXTRACT(InfoArray,'$.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 = 'JO3';
```

Line 90, Column 1 Tab Size: 4 SQL

MySQL 5.7 Command Line Client

```
+--------+-----------------------------------------------+
| InfoID | InfoArray                                     |
+--------+-----------------------------------------------+
| JO1    | {"Monday": [1, 2, 3], "Tuesday": [4, 5, 6]} |
| JO2    | {"TrooperID": "ST-1", "TrooperAge": 26}     |
| JO3    | {"TrooperID": [1, 2, [4, 5, 6]]}            |
+--------+-----------------------------------------------+
3 rows in set (0.00 sec)

+-----------------------------+
| Value of Key Tuesday in JO1 |
+-----------------------------+
| 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>
```