

```

+-----+
| col->"$.mascot" |
+-----+
| "Our mascot is a dolphin named \"Sakila\"." |
+-----+
1 row in set (0.00 sec)

```

This leaves the backslashes intact, along with the surrounding quote marks. To display the desired value using `mascot` as the key, but without including the surrounding quote marks or any escapes, use the inline path operator `->>`, like this:

```

mysql> SELECT sentence->>"$.mascot" FROM facts;
+-----+
| sentence->>"$.mascot" |
+-----+
| Our mascot is a dolphin named "Sakila". |
+-----+

```



Note

The previous example does not work as shown if the `NO_BACKSLASH_ESCAPES` server SQL mode is enabled. If this mode is set, a single backslash instead of double backslashes can be used to insert the JSON object literal, and the backslashes are preserved. If you use the `JSON_OBJECT()` function when performing the insert and this mode is set, you must alternate single and double quotes, like this:

```

mysql> INSERT INTO facts VALUES
> (JSON_OBJECT('mascot', 'Our mascot is a dolphin named "Sakila".'));

```

See the description of the `JSON_UNQUOTE()` function for more information about the effects of this mode on escaped characters in JSON values.

Normalization, Merging, and Autowrapping of JSON Values

When a string is parsed and found to be a valid JSON document, it is also normalized. This means that members with keys that duplicate a key found later in the document, reading from left to right, are discarded. The object value produced by the following `JSON_OBJECT()` call includes only the second `key1` element because that key name occurs earlier in the value, as shown here:

```

mysql> SELECT JSON_OBJECT('key1', 1, 'key2', 'abc', 'key1', 'def');
+-----+
| JSON_OBJECT('key1', 1, 'key2', 'abc', 'key1', 'def') |
+-----+
| {"key1": "def", "key2": "abc"} |
+-----+

```

Normalization is also performed when values are inserted into JSON columns, as shown here:

```

mysql> CREATE TABLE t1 (c1 JSON);

mysql> INSERT INTO t1 VALUES
> ('{"x": 17, "x": "red"}'),
> ('{"x": 17, "x": "red", "x": [3, 5, 7]}');

mysql> SELECT c1 FROM t1;
+-----+
| c1 |
+-----+
| {"x": "red"} |
| {"x": [3, 5, 7]} |
+-----+

```