

The logo for Gdevops, featuring a stylized orange 'G' followed by the word 'devops' in a white, lowercase, sans-serif font. The background is a solid blue color with decorative white geometric shapes and network-like patterns in the corners.

Gdevops

全球敏捷运维峰会

JSON Support in MySQL 5.7

演讲人：赖明星

全球敏捷运维峰会 杭州站

MySQL 5.7新特性概览

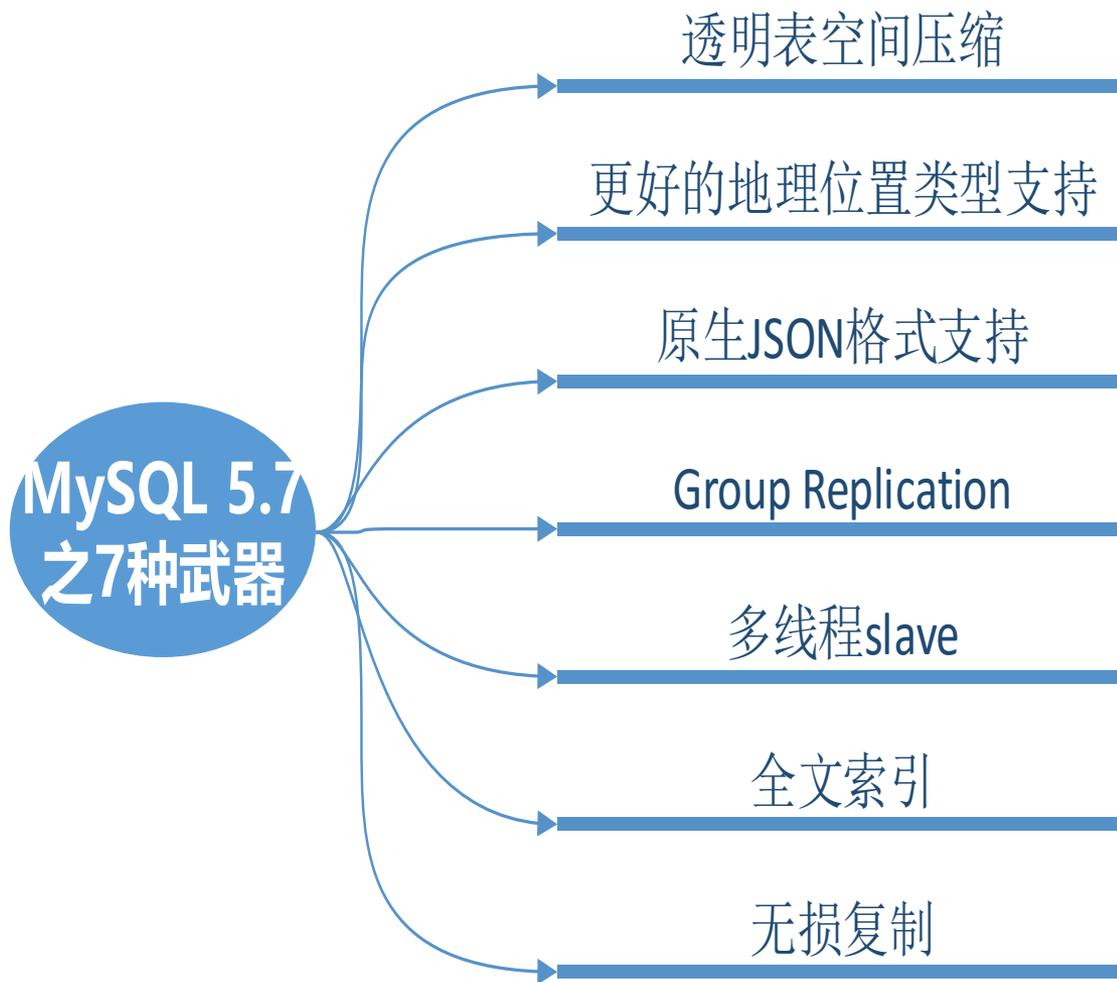
MySQL 5.7如何实现JSON存储

MySQL 5.7为了支持JSON新增的函数及使用示例

MySQL 5.7的JSON编码

如何为JSON数据类型创建索引

MySQL 5.7新特性概览



为什么MySQL需要支持JSON？

- 1、为MySQL用户提供更好的服务
- 2、随着非结构化数据需求的快速增长，各种NoSQL数据库的崛起（如mongodb）
- 3、竞争对手开始支持JSON

传统关系型数据库	支持JSON格式的版本
MySQL	5.7
Oracle	12.0.1
Microsoft SQL Server	2016
PostgreSQL	9.3

之前的MySQL如何实现JSON存储

在MySQL5.7之前，没有提供对非结构化数据的支持，但是，如果用户有这样的需求，也可以通过MySQL的blob来存储非结构化的数据。

```
mysql> create table t(json_data blob);

Query OK, 0 rows affected (0.13 sec)

mysql> insert into t values('{"key1":"data1", "key2":2, "key3":{"sub_key1":"sub_val1"}}');

Query OK, 1 row affected (0.01 sec)

mysql> select * from t;
+-----+
| json_data |
+-----+
| {"key1":"data1", "key2":2, "key3":{"sub_key1":"sub_val1"}} |
+-----+
1 row in set (0.00 sec)
```

之前的MySQL如何实现JSON存储

使用blob来存储JSON数据，需要用户保证插入的数据是一个能够转换成JSON格式的字符串，MySQL并不保证任何正确性。此外，提取JSON中的字段，也需要用户的代码中完成。

```
def get_conn():
    return pymysql.connect(user='lmx', password='123456', port=3306, host='127.0.0.1')

def test_json(n):
    sql = 'select * from t'
    with get_conn() as cursor:
        cursor.execute(sql)
        rows = cursor.fetchall()
        print json.dumps(json.loads(row[0][0]), indent=4)
```

```
root@dev1:~# python test.py
{
  "key3": {
    "sub_key1": "sub_val1"
  },
  "key2": 2,
  "key1": "data1"
}
```

用户自己使用blob存储JSON的缺点

这种方式虽然也能够实现JSON的存储，但是，有诸多缺点，最为显著的缺点有：

- 需要用户保证JSON的正确性，如果用户插入的数据并不是一个有效的JSON字符串，MySQL并不会报错
- 所有对JSON的操作，都需要在用户的代码里进行处理，可用性不高
- 即使只是提取JSON中某一个字段，也需要读出整个blob，效率较差
- 无法在JSON字段上建索引，以加快查询

全球敏捷运维峰会 杭州站

MySQL 5.7新特性概览

MySQL 5.7如何实现JSON存储

MySQL 5.7为了支持JSON新增的函数及使用示例

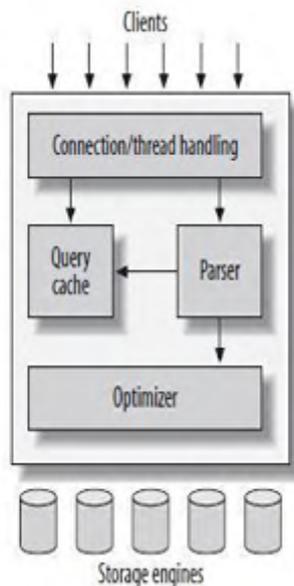
MySQL 5.7的JSON编码

如何为JSON数据类型创建索引

MySQL 5.7如何实现JSON存储

MySQL本身已经是一个比较完备的数据库系统，对于底层存储，并不适合有太大的改动，那么，MySQL是如何支持JSON格式的呢？

和我们前面的做法几乎一样——**通过blob来存储**。MySQL 5.7支持JSON的做法是，在server层提供了一堆便于操作JSON的函数，至于存储，就是简单地将JSON编码成blob，然后交由存储引擎层进行处理，也就是说，MySQL 5.7的JSON支持与存储引擎没有关系，MyISAM存储引擎也支持JSON格式



```
mysql> create table t_innodb(data json)engine=innodb;
Query OK, 0 rows affected (0.18 sec)

mysql> insert into t_innodb values('{"key":"val"}');
Query OK, 1 row affected (0.03 sec)

mysql> create table t_myisam(data json)engine=myisam;
Query OK, 0 rows affected (0.02 sec)

mysql> insert into t_myisam values('{"key":"val"}');
Query OK, 1 row affected (0.00 sec)
```

全球敏捷运维峰会 杭州站

MySQL 5.7新特性概览

MySQL 5.7如何实现JSON存储

MySQL 5.7为了支持JSON新增的函数及使用示例

MySQL 5.7的JSON编码

如何为JSON数据类型创建索引

MySQL 5.7为了支持JSON新增的函数

info	modify	create	Get data and helper
<ul style="list-style-type: none">• JSON_VALID• JSON_TYPE• JSON_KEYS• JSON_LENGTH• JSON_DEPTH	<ul style="list-style-type: none">• JSON_REMOVE• JSON_APPEND• JSON_SET• JSON_INSERT• JSON_REPLACE	<ul style="list-style-type: none">• JSON_MERGE• JSON_ARRAY• JSON_OBJECT	<ul style="list-style-type: none">• JSON_EXTRACT• JSON_SEARCH• JSON_QUOTE• JSON_UNQUOTE

函数太多记不住怎么办？在MySQL客户端，输入“? 函数名”即可获取帮助文档

```
mysql> ? json_object
Name: 'JSON_OBJECT'
Description:
JSON_OBJECT([key, val[, key, val] ...])

Evaluates a (possibly empty) list of key/value pairs and returns a JSON
object containing those pairs. An error occurs if any key name is NULL
or the number of arguments is odd.

URL: http://dev.mysql.com/doc/refman/5.7/en/json-creation-functions.html

Examples:
mysql> SELECT JSON_OBJECT('id', 87, 'name', 'carrot');
+-----+
| JSON_OBJECT('id', 87, 'name', 'carrot') |
+-----+
| {"id": 87, "name": "carrot"}           |
+-----+
```

MySQL5.7 JSON使用示例

```
mysql> create table t1(data json);
Query OK, 0 rows affected (0.13 sec)

mysql> insert into t1(data) values('{\"series\":1}'), ('{\"series\":7}');
Query OK, 2 rows affected (0.01 sec)
Records: 2 Duplicates: 0 Warnings: 0

mysql> select * from t1 where json_extract(data, \"$.series\") >= 7;
+-----+
| data          |
+-----+
| {\"series\": 7} |
+-----+
1 row in set (0.00 sec)
```

MySQL5.7 JSON使用示例：长度限制

- key最长为65535个字节
- Json的长度限制为4G(因为blob最大为4G)

```
mysql> insert into t values(JSON_OBJECT(repeat('a', 65535), 'val'));  
Query OK, 1 row affected (0.37 sec)
```

```
mysql> insert into t values(JSON_OBJECT(repeat('a', 65536), 'val'));  
ERROR 3151 (22032): The JSON object contains a key name that is too long.
```

MySQL5.7 JSON使用示例：大小写敏感

```
mysql> create table t(data json);
Query OK, 0 rows affected (0.20 sec)

mys node1 [localhost] {msandbox} ((none)) > select json_valid('null'), json_valid('NULL');
Que +-----+-----+
   | json_valid('null') | json_valid('NULL') |
   +-----+-----+
mys |                1 |                0 |
+-- +-----+-----+
   | data |
   +-----+-----+
   | {"LastName": "mingxing", "lastName": "mingxing"} |
   +-----+-----+
1 row in set (0.00 sec)
```

MySQL5.7 JSON使用示例：规范

```
node1 [localhost] {msandbox} ((none)) > select json_object('key1', 1, 'key2', 2, 'key1', 3);
+-----+
| json_object('key1', 1, 'key2', 2, 'key1', 3) |
+-----+
| {"key1": 1, "key2": 2} |
+-----+
1 row in set (0.00 sec)
```

全球敏捷运维峰会 杭州站

MySQL 5.7新特性概览

MySQL 5.7如何实现JSON存储

MySQL 5.7为了支持JSON新增的函数及使用示例

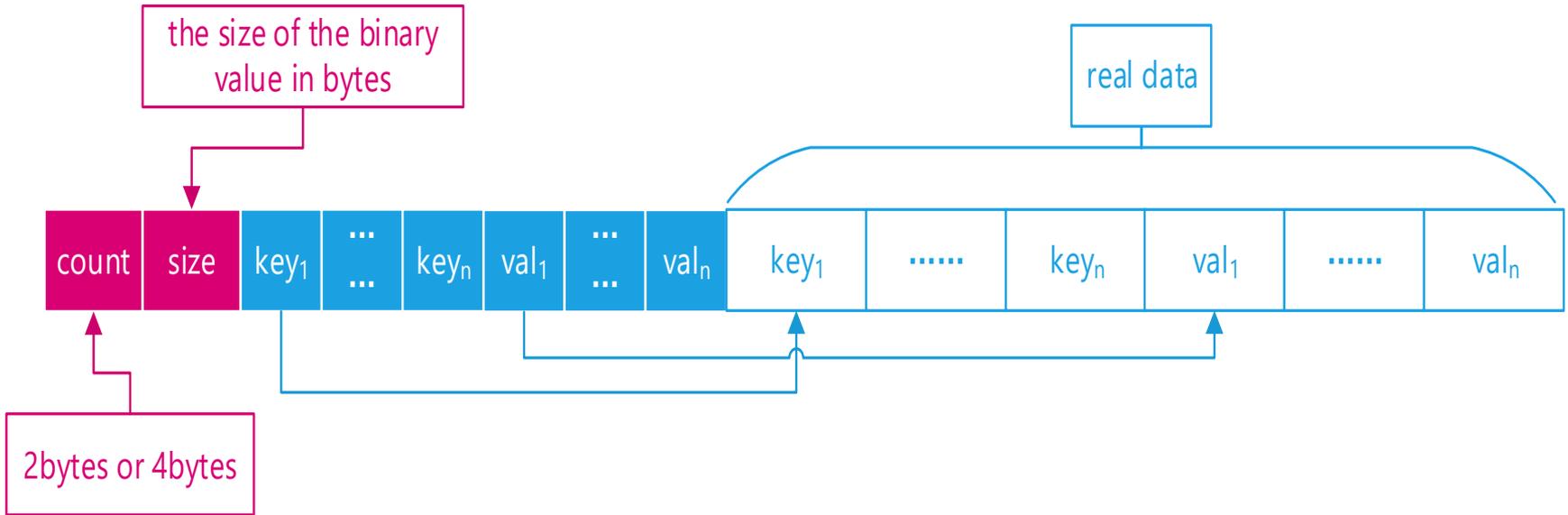
MySQL 5.7的JSON编码

如何为JSON数据类型创建索引

MySQL 5.7 JSON编码：编码格式

If the value is a JSON object, its binary representation will have a header that contains:

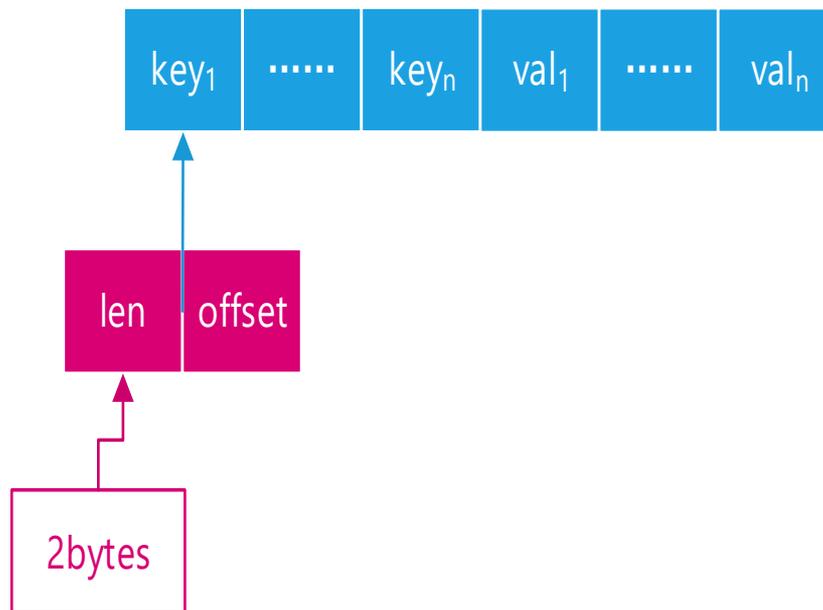
- the member count
- the size of the binary value in bytes
- a list of pointers to each key
- a list of pointers to each value



MySQL 5.7 JSON编码：编码格式

The size of key entries for objects when using the small storage format or the large storage format. In the small format it is 4bytes (2 bytes for key length and 2 bytes for key offset). In the large format it is 6 (2 bytes for length, 4 bytes for offset).

- `#define KEY_ENTRY_SIZE_SMALL(2 + SMALL_OFFSET_SIZE)`
- `#define KEY_ENTRY_SIZE_LARGE(2 + LARGE_OFFSET_SIZE)`



全球敏捷运维峰会 杭州站

MySQL 5.7新特性概览

MySQL 5.7如何实现JSON存储

MySQL 5.7为了支持JSON新增的函数及使用示例

MySQL 5.7的JSON编码

如何为JSON数据类型创建索引

如何为JSON数据类型创建索引

- generated column介绍
- generated column创建索引
- JSON column创建索引

如何为JSON数据类型创建索引

generated column是MySQL 5.7引入的新特性，所谓generated column，就是数据库中这一列由其他列计算而得，我们以参考手册中的例子予以说明。

例如，知道直角三角形的两条直角边，要求斜边的长度。很明显，可以通过两条直角边计算而得，那么，这时候就可以在数据库中只存放直角边，斜边使用generated column，如下所示：

```
CREATE TABLE triangle (  
  sidea DOUBLE,  
  sideb DOUBLE,  
  sidec DOUBLE AS (SQRT(sidea * sidea + sideb * sideb))  
);  
INSERT INTO triangle (sidea, sideb) VALUES(1,1),(3,4),(6,8);
```

如何为JSON数据类型创建索引

MySQL支持为generated column建立索引，如下所示：

```
mysql> create table t(x int primary key, y int, z int generated always as (x / y), unique key idz(z));
Query OK, 0 rows affected (0.11 sec)

mysql> show create table t\G
***** 1. row *****
      Table: t
Create Table: CREATE TABLE `t` (
  `x` int(11) NOT NULL,
  `y` int(11) DEFAULT NULL,
  `z` int(11) GENERATED ALWAYS AS (x / y) VIRTUAL,
  PRIMARY KEY (`x`),
  UNIQUE KEY `idz` (`z`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
1 row in set (0.01 sec)
```

如何为JSON数据类型创建索引

为JSON column创建索引的步骤如下：

1. 创建generated column
2. 在generated column上创建索引

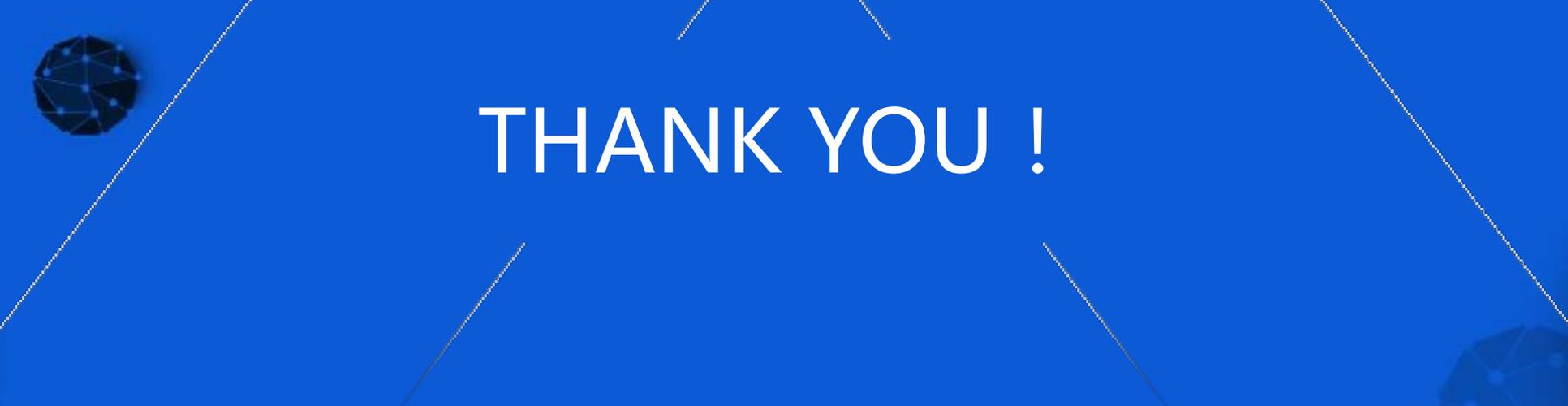
```
node1 [localhost] {msandbox} (test) > show create table UserJson\G
***** 1. row *****
      Table: UserJson
Create Table: CREATE TABLE `UserJson` (
  `uid` bigint(20) NOT NULL AUTO_INCREMENT,
  `data` json DEFAULT NULL,
  `name` varchar(128) GENERATED ALWAYS AS (json_unquote(json_extract(`data`, '$.name'))) VIRTUAL,
  PRIMARY KEY (`uid`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
1 row in set (0.00 sec)

node1 [localhost] {msandbox} (test) > alter table UserJson Add index idx_name(name);
Query OK, 0 rows affected (0.06 sec)
Records: 0 Duplicates: 0 Warnings: 0
```



G*devops*

全球敏捷运维峰会



THANK YOU !