[**MySQL 5.7原生JSON格式支持**](https://www.cnblogs.com/zoucaitou/p/4424575.html)

在MySQL与PostgreSQL的对比中，PG的JSON格式支持优势总是不断被拿来比较。其实早先MariaDB也有对非结构化的数据进行存储的方案，称为dynamic column，但是方案是通过BLOB类型的方式来存储。这样导致的问题是查询性能不高，不能有效建立索引，与一些文档数据库对比，优势并不大，故在社区的反应其实比较一般。当然，MariaDB的dynamic column功能还不仅限于非结构化数据的存储，但不在本文进行展开。

MySQL 5.7.7 labs版本开始InnoDB存储引擎已经原生支持JSON格式，该格式不是简单的BLOB类似的替换。原生的JSON格式支持有以下的优势：

* JSON数据有效性检查：BLOB类型无法在数据库层做这样的约束性检查
* 查询性能的提升：查询不需要遍历所有字符串才能找到数据
* 支持索引：通过虚拟列的功能可以对JSON中的部分数据进行索引

首先我们来看如何在MySQL中使用原生的JSON格式：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | mysql> create table user ( uid int auto\_increment,      -> data json,primary key(uid))engine=innodb;  Query OK, 0 rows affected (0.01 sec)    mysql> insert into user values (NULL,      -> '{"name":"David","mail":"jiangchengyao@gmail.com","address":"Shangahai"}');  Query OK, 1 row affected (0.00 sec)    mysql> insert into user values (NULL,'{"name":"Amy","mail":"amy@gmail.com"}');  Query OK, 1 row affected (0.00 sec) |

可以看到我们新建了表user，并且将列data定义为了JSON类型。这意味着我们可以对插入的数据做JSON格式检查，确保其符合JSON格式的约束，如插入一条不合法的JSON数据会报如下错误：

|  |  |
| --- | --- |
| 1  2 | mysql> insert into user values (NULL,"test");  ERROR 3130 (22032): Invalid JSON text: "Invalid value" at position 2 in value (or column) 'test'. |

此外，正如前面所说的，MySQL 5.7提供了一系列函数来高效地处理JSON字符，而不是需要遍历所有字符来查找，这不得不说是对MariaDB dynamic column的巨大改进：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | mysql> select jsn\_extract(data, '$.name'),jsn\_extract(data,'$.address') from user;  +-----------------------------+-------------------------------+  | jsn\_extract(data, '$.name') | jsn\_extract(data,'$.address') |  +-----------------------------+-------------------------------+  | "David" | "Shangahai" |  | "Amy" | NULL |  +-----------------------------+-------------------------------+  2 rows in set (0.00 sec) |

**当然，最令人的激动的功能应该是MySQL 5.7的虚拟列功能**，通过传统的B+树索引即可实现对JSON格式部分属性的快速查询。使用方法是首先创建该虚拟列，然后在该虚拟列上**创建索引**：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | mysql> ALTER TABLE user ADD user\_name varchar(128)      -> GENERATED ALWAYS AS (jsn\_extract(data,'$.name')) VIRTUAL;  Query OK, 0 rows affected (0.01 sec)  Records: 0 Duplicates: 0 Warnings: 0    mysql> select user\_name from user;  +-----------+  | user\_name |  +-----------+  | "Amy"     |  | "David"   |  +-----------+  2 rows in set (0.00 sec)    mysql> alter table user add index idx\_username (user\_name);  Query OK, 2 rows affected (0.01 sec)  Records: 2  Duplicates: 0  Warnings: 0 |

然后可以通过添加的索引对用户名进行快速的查询，这和普通类型的列查询一样。而通过explain可以验证优化器已经选择了在虚拟列上创建的新索引：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | mysql> explain select \* from user where user\_name='"Amy"'\G  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*             id: 1    select\_type: SIMPLE          table: user     partitions: NULL           type: ref  possible\_keys: idx\_username            key: idx\_username        key\_len: 131            ref: const           rows: 1       filtered: 100.00          Extra: NULL  1 row in set, 1 warning (0.00 sec) |

可以发现MySQL 5.7对于JSON格式堪称完美，相信PostgreSQL阵营需要寻找新的策略来“攻击”MySQL了吧。如无意外，还是会停留在优化器这块，毕竟这块是目前MySQL必须要克服的最大问题，好在MySQL团队已经在重构优化器代码，相信更好的优化器将会在下一个版本中全面爆发。而一大堆文档数据库们已经哭晕在厕所了吧。