

Apache Cassandra static column 介绍与实战

假设我们有这样的场景：我们想在 Cassandra 中使用一张表记录用户基本信息（比如 email、密码等）以及用户状态更新。我们知道，用户的基本信息一般很少会变动，但是状态会经常变化，如果每次状态更新都把用户基本信息都加进去，势必会让费大量的存储空间。为了解决这种问题，Cassandra 引入了 static column。同一个 partition key 中被声明为 static 的列只有一个值的，也就是只存储一份。

如果想及时了解 Spark、Hadoop 或者 Hbase 相关的文章，欢迎关注微信公共帐号：iteblog_hadoop

定义 static column

在表中将某个列定义为 STATIC 很简单，只需要在列的最后面加上 STATIC 关键字，具体如下：

```
CREATE TABLE "iteblog_users_with_status_updates" (  
  "username" text,  
  "id" timeuuid,  
  "email" text STATIC,  
  "encrypted_password" blob STATIC,  
  "body" text,  
  PRIMARY KEY ("username", "id")  
);
```

iteblog_users_with_status_updates 表中我们将 email 和 encrypted_password 两个字段设置为 STATIC 了，这意味着同一个 username 只会有一个 email 和 encrypted_password。

注意，不是任何表都支持给列加上 STATIC 关键字的，静态列有以下限制。

- 如果表没有定义 Clustering columns（又称 Clustering key），这种情况是不能添加静态列的。如下：

```
cqlsh:iteblog_keyspace> CREATE TABLE "iteblog_users_with_status_updates_invalid" (  
  ... "username" text,  
  ... "id" timeuuid,  
  ... "email" text STATIC,  
  ... "encrypted_password" blob STATIC,  
  ... "body" text,  
  ... PRIMARY KEY ("username")
```

```
... );
```

InvalidRequest: Error from server: code=2200 [Invalid query] message="Static columns are only useful (and thus allowed) if the table has at least one clustering column"

iteblog_users_with_status_updates_invalid 表只有 PRIMARY KEY，没有定义 clustering column，不支持创建 Static columns。这是因为静态列在同一个 partition key 存在多行的情况下才能达到最优情况，而且行数越多效果也好。但是如果没有定义 clustering column，相同 PRIMARY KEY 的数据在同一个分区里面只存在一行数据，本质上就是静态的，所以没必要支持静态列。

- 如果建表的时候指定了 COMPACT STORAGE，这时候也不允许存在静态列：

```
cqlsh:iteblog_keyspace> CREATE TABLE "iteblog_users_with_status_updates_invalid" (  
    ... "username" text,  
    ... "id" timeuuid,  
    ... "email" text STATIC,  
    ... "encrypted_password" blob STATIC,  
    ... "body" text,  
    ... PRIMARY KEY ("username", "id")  
    ... )WITH COMPACT STORAGE;
```

InvalidRequest: Error from server: code=2200 [Invalid query] message="Static columns are not supported in COMPACT STORAGE tables"

- 如果列是 partition key/Clustering columns 的一部分，那么这个列不能说明为静态列：

```
cqlsh:iteblog_keyspace> CREATE TABLE "iteblog_users_with_status_updates_invalid" (  
    ... "username" text,  
    ... "id" timeuuid STATIC,  
    ... "email" text STATIC,  
    ... "encrypted_password" blob STATIC,  
    ... "body" text,  
    ... PRIMARY KEY ("username", "id")  
    ... );
```

InvalidRequest: Error from server: code=2200 [Invalid query] message="Static column id cannot be part of the PRIMARY KEY"

```
cqlsh:iteblog_keyspace> CREATE TABLE "iteblog_users_with_status_updates_invalid" (  
    ... "username" text,  
    ... "id" timeuuid,  
    ... "email" text STATIC,  
    ... "encrypted_password" blob STATIC,  
    ... "body" text,  
    ... PRIMARY KEY (("username", "id"), email)  
    ... );
```

InvalidRequest: Error from server: code=2200 [Invalid query] message="Static column email cannot be part of the PRIMARY KEY"

给静态列的表插入数据

含有静态列的表插入数据和正常表类似，比如我们现在往 iteblog_users_with_status_updates 导入数据：

```
cqlsh:iteblog_keyspace> INSERT INTO "iteblog_users_with_status_updates"
... ("username", "id", "email", "encrypted_password", "body")
... VALUES (
... 'iteblog',
... NOW(),
... 'iteblog_hadoop@iteblog.com',
... 0x877E8C36EFA827DBD4CAFBC92DD90D76,
... 'Learning Cassandra!'
... );
```

```
cqlsh:iteblog_keyspace> select username, email, encrypted_password, body from iteblog_users_with_status_updates;
```

username	email	encrypted_password	body
iteblog	iteblog_hadoop@iteblog.com	0x877e8c36efa827dbd4cafbc92dd90d76	Learning Cassandra!

(1 rows)

我们成功的插入一条数据了。但是上面的插入语句做了两件事：

- 所有 username 为 iteblog 数据中的 email 和 encrypted_password 都被设置为 iteblog_hadoop@iteblog.com 和 0x877e8c36efa827dbd4cafbc92dd90d76 了。
- 在 iteblog 所在的分区中新增了 body 内容为 Learning Cassandra! 的记录。

现在我们在再往表中插入一条数据，如下：

```
cqlsh:iteblog_keyspace> INSERT INTO "iteblog_users_with_status_updates"
... ("username", "id", "body")
... VALUES ('iteblog', NOW(), 'I love Cassandra!');
```

```
cqlsh:iteblog_keyspace> select username, email, encrypted_password, body from iteblog_users_with_status_updates;
```

```

username | email | encrypted_password | body
-----+-----+-----+-----
iteblog | iteblog_hadoop@iteblog.com | 0x877e8c36efa827dbd4cafbc92dd90d76 | Learning
Cassandra!
iteblog | iteblog_hadoop@iteblog.com | 0x877e8c36efa827dbd4cafbc92dd90d76 | I love Ca
ssandra!

(2 rows)
cqlsh:iteblog_keyspace>

```

可以看到，这次插入数据的时候，我们并没有指定 email 和 encrypted_password，但是从查询结果可以看出，新增加的行 email 和 encrypted_password 的值和之前是一样的！

现在由于某些原因，用户修改了自己的 email，我们来看看会发生什么事：

```

cqlsh:iteblog_keyspace> UPDATE iteblog_users_with_status_updates SET email = 'iteblog@itebl
og.com'
... WHERE username = 'iteblog';
cqlsh:iteblog_keyspace> select username, email, encrypted_password, body from iteblog_user
s_with_status_updates;

```

```

username | email | encrypted_password | body
-----+-----+-----+-----
iteblog | iteblog@iteblog.com | 0x877e8c36efa827dbd4cafbc92dd90d76 | Learning Cassand
ra!
iteblog | iteblog@iteblog.com | 0x877e8c36efa827dbd4cafbc92dd90d76 | I love Cassandra!

(2 rows)

```

从上面查询这输出的结果可以看出，username 为 iteblog 的 email 全部修改成一样的了！这就是静态列的强大之处。

现在表中存在了用户的邮箱和密码等信息，如果我们前端做了个页面支持用户修改自己的邮箱和密码，这时候我们的后台系统需要获取到现有的邮箱和密码，具体如下：

```

cqlsh:iteblog_keyspace> SELECT "username", "email", "encrypted_password"
... FROM "iteblog_users_with_status_updates"
... WHERE "username" = 'iteblog';

```

```
username | email          | encrypted_password
-----+-----+-----
iteblog | iteblog@iteblog.com | 0x877e8c36efa827dbd4cafbc92dd90d76
iteblog | iteblog@iteblog.com | 0x877e8c36efa827dbd4cafbc92dd90d76
```

(2 rows)

可以看出，表中有多少行 username 为 iteblog 的数据将会输出多少行邮箱和密码，这肯定不是我们想要的。这时候我们可以在查询的时候加上 DISTINCT 关键字，如下：

```
cqlsh:iteblog_keyspace> SELECT DISTINCT "username", "email", "encrypted_password"
... FROM "iteblog_users_with_status_updates"
... WHERE "username" = 'iteblog';
```

```
username | email          | encrypted_password
-----+-----+-----
iteblog | iteblog@iteblog.com | 0x877e8c36efa827dbd4cafbc92dd90d76
```

(1 rows)

这样不管表中有多少行 username 为 iteblog 的数据，最终都会显示一行数据。注意，虽然我们加了 DISTINCT 关键字，但是 Cassandra 并不是将 username 为 iteblog 的数据全部拿出来，然后再去重的，因为静态列本来在底层就存储了一份，所以没必要去重。

静态列的意义

到这里，我们已经了解了 Cassandra 中静态列的创建、使用等。那静态列有什么意义呢？因为 Cassandra 中是不支持 join 的，静态列相当于把两张表进行了 join 操作。

那什么时候建议使用静态列呢？如果两张表关联度很大，而且我们经常需要同时查询这两张表，那这时候就可以考虑使用静态列了。

本博客文章除特别声明，全部都是原创！
 转载本文请加上：转载自过往记忆 (<https://www.iteblog.com/>)
 本文链接: **【】 ()**