在kunDB的分shard库中，每个表都必须指定一个shard key，然后这个表的全量数据根据shard key会被分到各个shard中独立存储，如果一个表有主键唯一性约束的话，为了保证主键唯一性，可以采用将主键指定为shard key的办法。

但是在实际的使用过程中发现，如果所有表都按照主键来做shard key的话，很多join都变成了cross shard的join，这点给kunDB的query性能造成了很大的影响，经过实验，发现对于主键唯一性约束，除了将主键指定为shard key之外，还有另外一种方法，即利用secondary vindex。

在kunDB中，vindex分为primary vindex（即shard key）与secondary vindex，secondary vindex中有一类为look up index，下面将介绍如何利用unique的look up vindex来保证主键的唯一性：

首先介绍一下表的schema：

CREATE TABLE `ry\_zzbdxx` (  
`BDBH` varchar(32) NOT NULL,  
`RYBH` varchar(23) DEFAULT NULL,  
`ZZBH` varchar(32) DEFAULT NULL,  
`GMSFHM` char(18) DEFAULT NULL,  
`XM` varchar(50) DEFAULT NULL,  
`XMHYPY` varchar(150) DEFAULT NULL,  
`CSRQ` char(8) DEFAULT '0',  
`JUZRQ` char(8) DEFAULT '0',  
`LKRQ` char(8) DEFAULT '0',  
`ZXBSDM` char(1) DEFAULT NULL,  
`ZXDW\_GAJGJGDM` char(12) DEFAULT NULL,  
`ZXR\_GAJGJYDM` char(10) DEFAULT NULL,  
`JZLBDM` char(1) DEFAULT NULL,  
`ZZ\_XZQHDM` char(6) DEFAULT NULL,  
`ZZ\_QHNXXDZ` varchar(100) DEFAULT NULL,  
`ZYFW\_SFXXDM` char(1) DEFAULT NULL,  
`DJDW\_GAJGJGDM` char(12) DEFAULT NULL,  
`DJR\_GAJGJYDM` char(10) DEFAULT NULL,  
`DJSJ` varchar(14) DEFAULT NULL,  
`CZDW` char(12) DEFAULT NULL,  
`CZR` char(10) DEFAULT NULL,  
`CZSJ` varchar(17) DEFAULT NULL,  
`CZBS` varchar(1) DEFAULT NULL,  
PRIMARY KEY (`BDBH`),  
KEY `idx1` (`CZBS`),  
KEY `idx2` (`ZZBH`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8

其中BDBH为主键，但是实际业务中，需要按RYBH来分shard

这样的话，为了保证主键唯一性约束，现在需要在RYBH列上建立一个unique的look up index

kunDB中，look up index的实现原理是在kunDB中建立一个look up表，然后将相关信息写入look up表中即可。

在这个例子中，我们需要建立BDBH到keyspaceID的look up index，look up表的schema信息如下：

CREATE TABLE `ry\_zzbdxx\_bdbh\_rybh\_idx` (  
`BDBH` varchar(32) DEFAULT NULL,  
`RYBH` varbinary(16) DEFAULT NULL，

PRIMARY KEY (`BDBH`)  
) ENGINE=InnoDB DEFAULT

注意look up表中，第一列即secondary vindex所在的列，其类型必须与原表保持一致，第二列实际上是指向keyspaceID的列，该列的类型与原表（即RY\_ZZBDXX）的primary vindex的类型有关，在本例中，ry\_zzbdxx的primary vindex为binary\_md5类型，其对应的keyspaceID类型即为varbinary（16），如果primary vindex类型不同该列类型也需要做相应的改变，比如如果primary vindex类型为hash，则该列类型即为bigint。

secondary vindex保证主键唯一性的原理就在于index表是按照原表的主键做shard key的，而且这一列在index表中也被声明成主键，这样就可以通过secondary vindex来间接实现原表主键的唯一性约束。

最新的master分支中，新开发了分区表支持的feature, 可以直接用sql语句创建secondary vindex，不需要手动去apply vschema和创建lookup 表。

在本例中的sql语句为：create global unique index ry\_zzbdxx\_bdbh\_rybh\_idx on ry\_zzbdxx(BDBH);

这样vschema中就会包含名为ry\_zzbdxx\_bdbh\_rybh\_idx的look up index的vindex 和table表

{

"sharded": true,

"vindexes": {

"binary\_md5": {

"type": "binary\_md5"

},

"unicode\_loose\_md5": {

"type": "unicode\_loose\_md5"

},

"ry\_zzbdxx\_bdbh\_rybh\_idx": {

"type": "lookup\_unique",

"params": {

"from": "bdbh",

"table": "ry\_zzbdxx\_bdbh\_rybh\_idx",

"to": "rybh"

},

"owner": "ry\_zzbdxx"

}

},

"tables": {

"ry\_zzbdxx": {

"column\_vindexes": [

{

"column": "rybh",

"name": "binary\_md5"

},

{

"column": "bdbh",

"name": "ry\_zzbdxx\_bdbh\_rybh\_idx"

}

]

},

"ry\_zzbdxx\_bdbh\_rybh\_idx": {

"column\_vindexes": [

{

"column": "bdbh",

"name": "unicode\_loose\_md5"

}

]

}

}

}