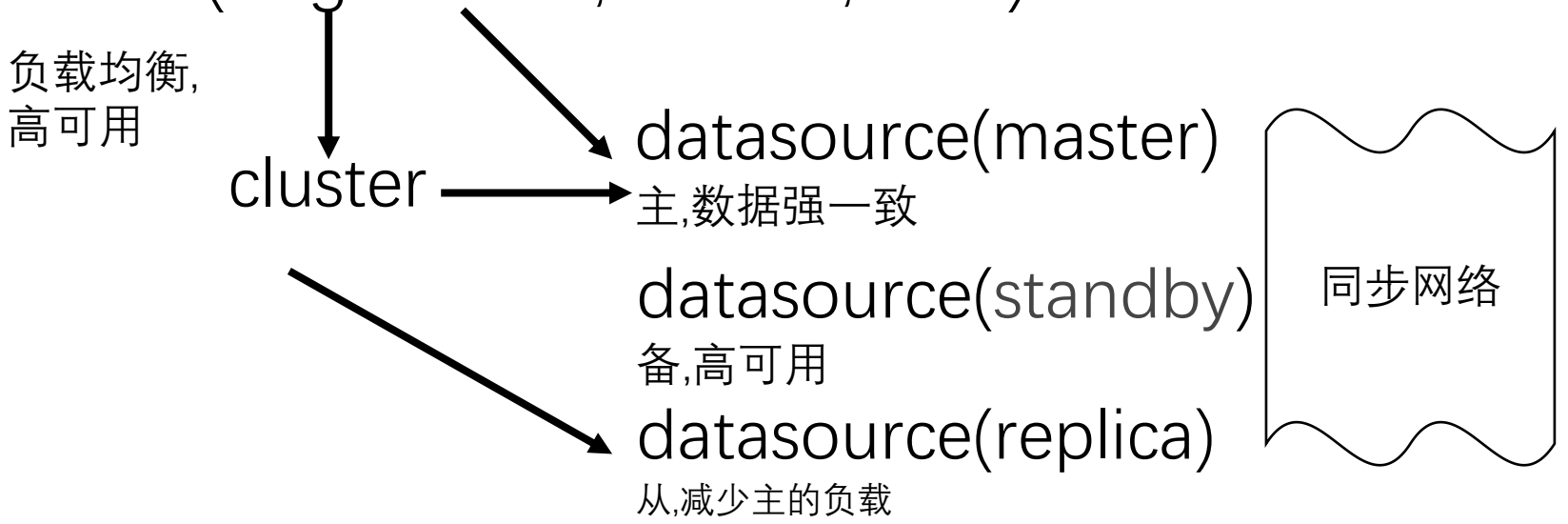
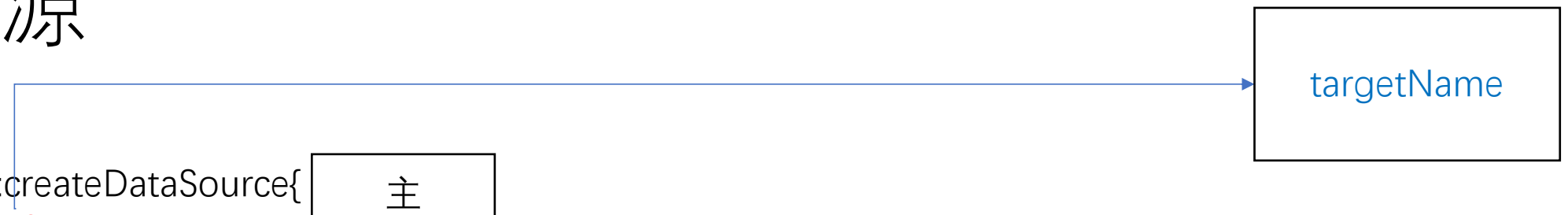
映射关系

Mycat2

logical table->partition(targetName,schema,table)

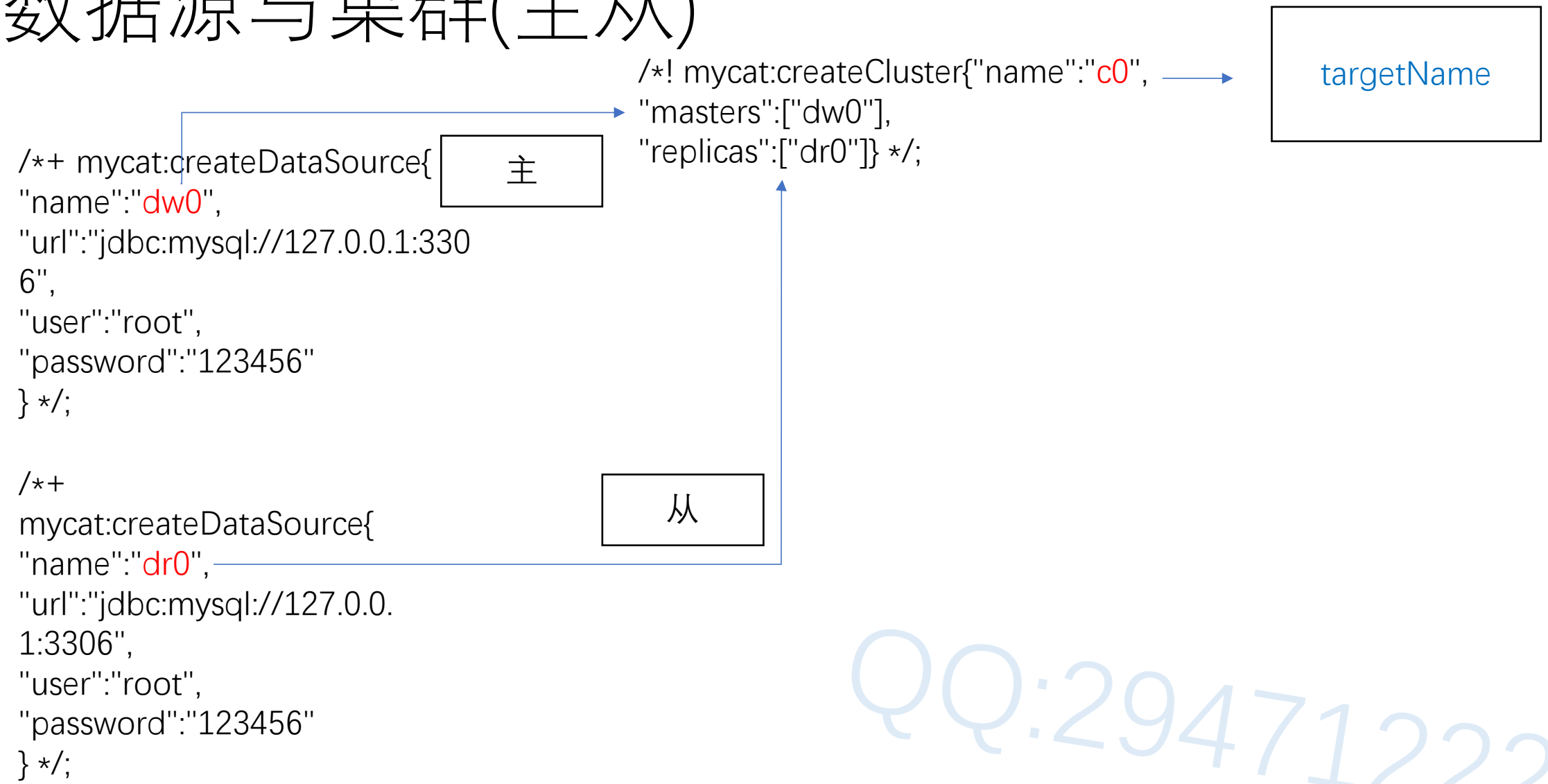


数据源

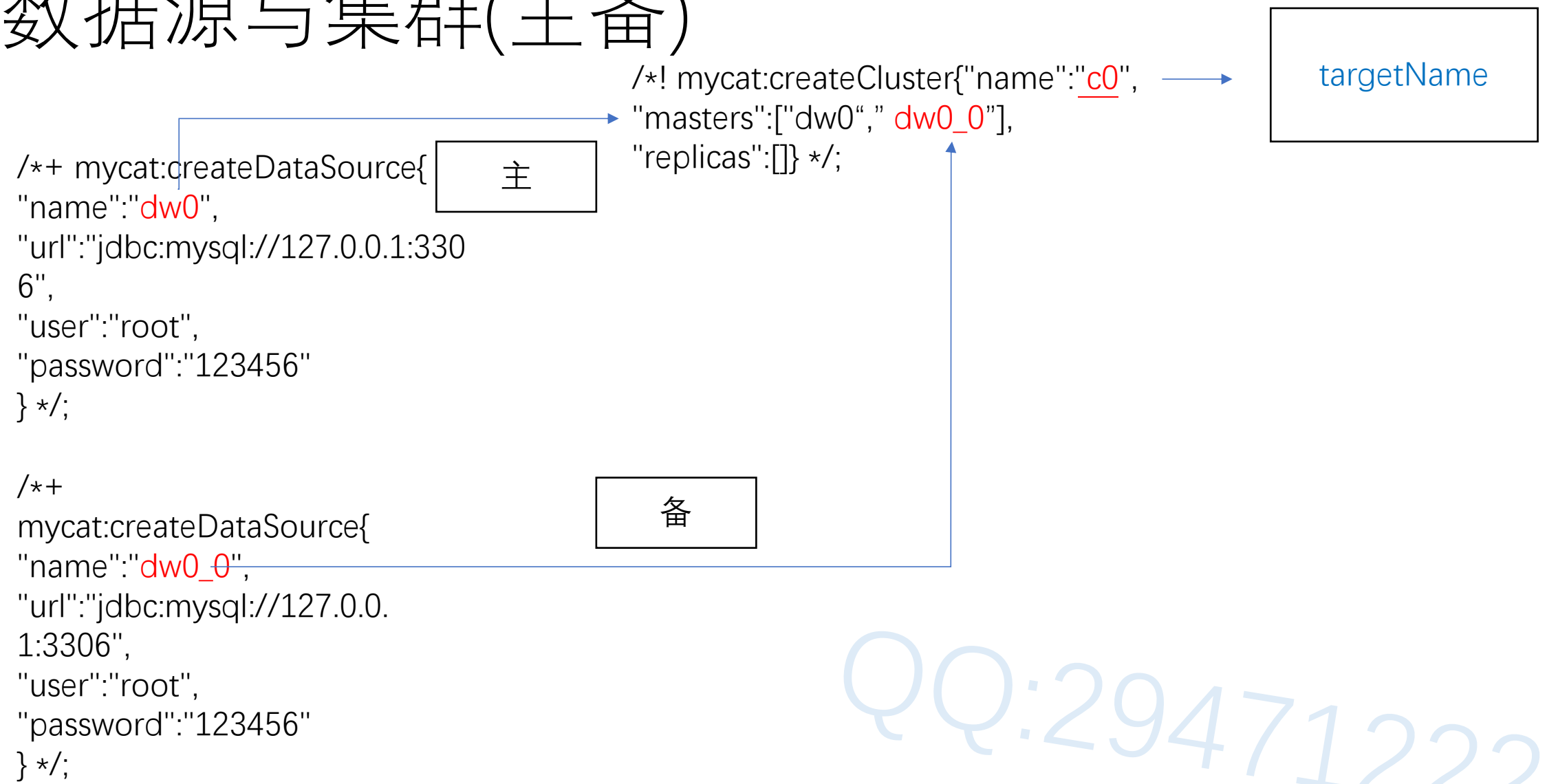



```
/*+ mycat:createDataSource{
"name":"dw0",
"url":"jdbc:mysql://127.0.0.1:330
6",
"user":"root",
"password":"123456"
} */;
```

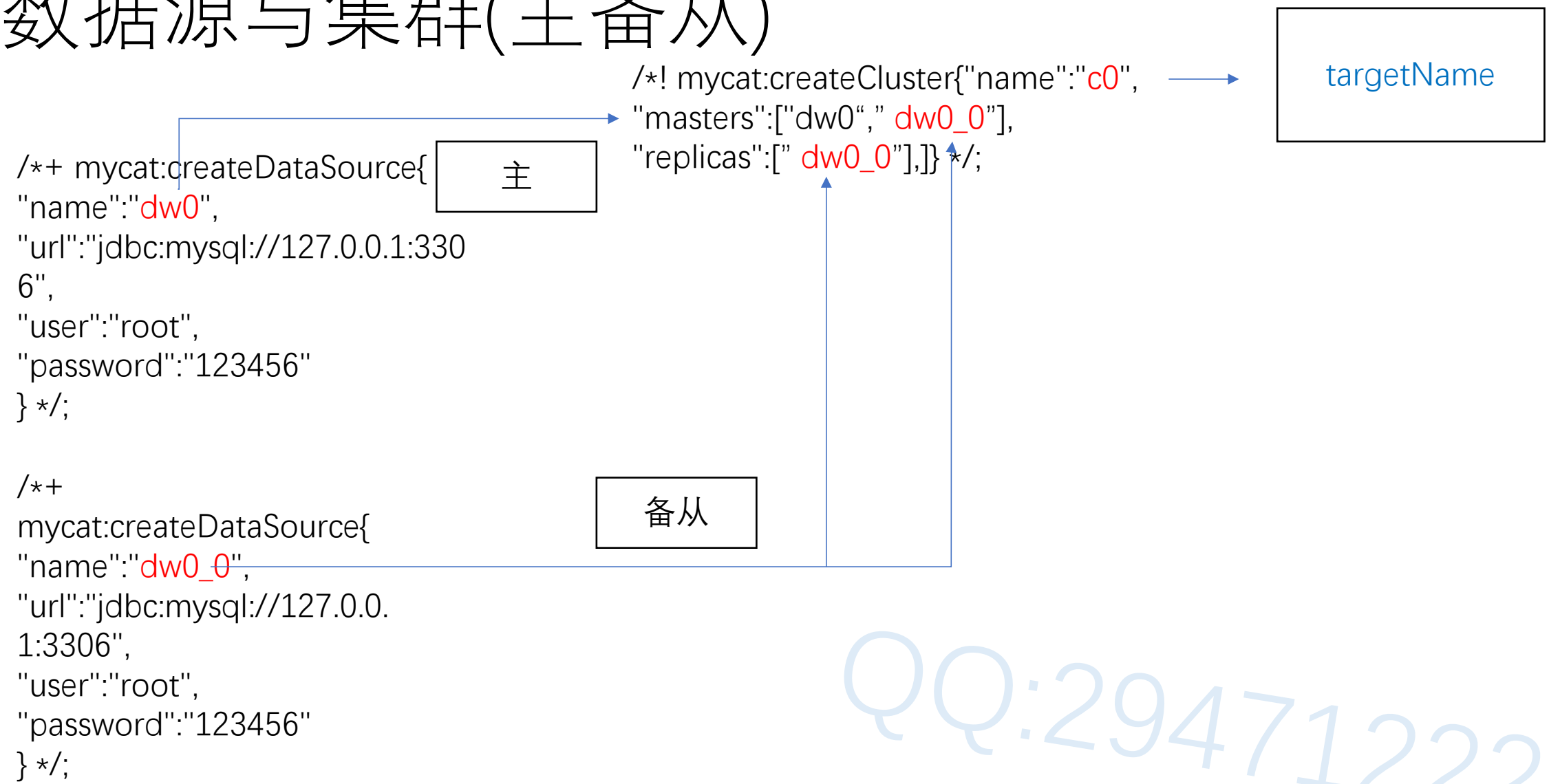
数据源与集群(主从)



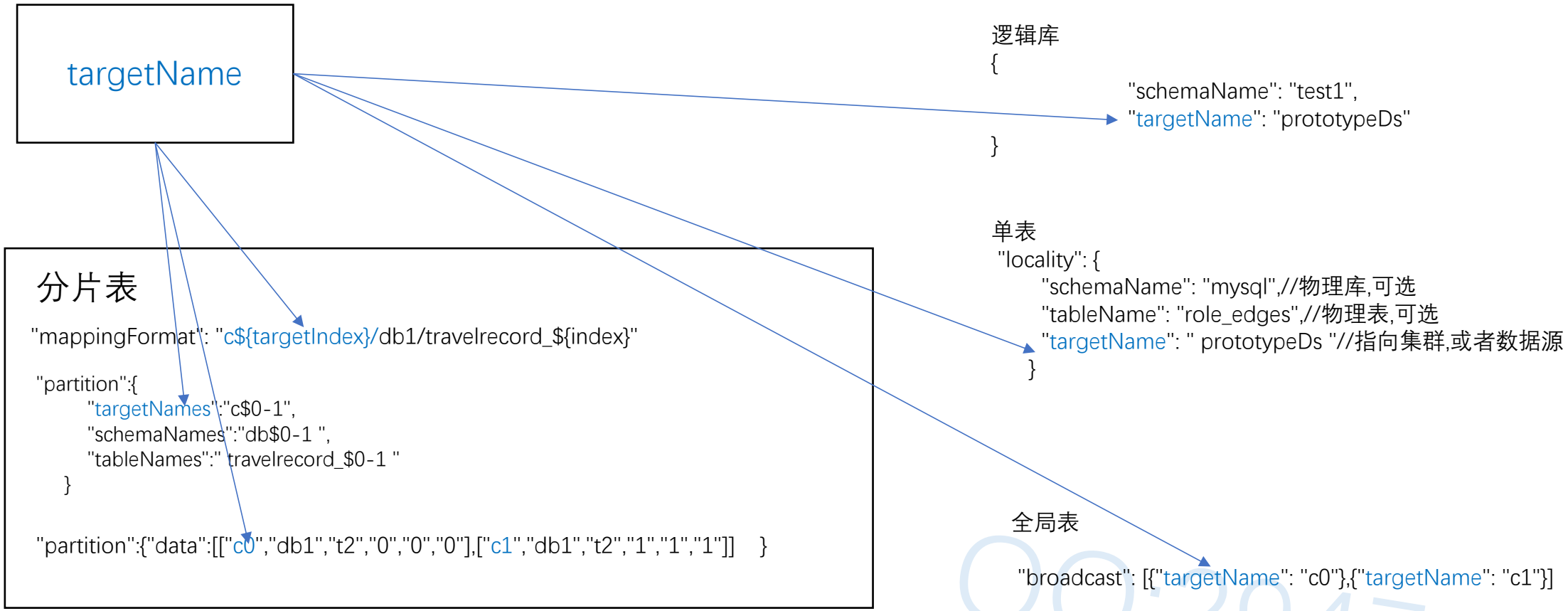
数据源与集群(主备)



数据源与集群(主备从)



数据源与逻辑表



以下说明不区分集群或数据源,统一称为数据源

Mycat2

```
//test1.schema.json
{
        "schemaName": "test1",
        "targetName": "prototypeDs"
}
```

逻辑库: test1
数据源: prototypeDs

MySQL

物理库: test1
URL:jdbc:mysql://localhost:3306/test1

以下说明不区分集群或数据源,统一称为数据源

Mycat2-单表

```
{
  "schemaName": “test1 ",
  "normalTables": {
    "role_edges": {
      "createTableSQL":null,//可选
      "locality": {
        "schemaName": "mysql",//物理库,可选
        "tableName": "role_edges",//物理表,可选
        "targetName": " prototypeDs "//指向集群,或者数据源
      }
    }
    ......
```

逻辑库: test1
逻辑表: role_edges
数据源: prototypeDs

MySQL

物理库: mysql
物理表: role_edges
URL:jdbc:mysql://localhost:3306/mysql

以下说明不区分集群或数据源,统一称为数据源

Mycat2-全局表

```
{
  "schemaName": "test1",
  "globalTables": {
    "role_edges": {
      "broadcast": [{"targetName": "c0"},{"targetName": "c1"}]
    }
    ......
```

逻辑库: test1
逻辑表: role_edges
数据源: c0,c1

MySQL

物理库: test1
物理表: role_edges
URL:jdbc:mysql://localhost:3306/test1

物理库: test1
物理表: role_edges
URL:jdbc:mysql://localhost:3307/test1

Mycat2在处理test1. role_edges的insert/update语句会把SQL广播到3306,3307上面
只查询test1. role_edges表的时候,会随机选一个MySQL查询

以下说明不区分集群或数据源,统一称为数据源

Mycat2-分片表-分库分表下标

```
"shardingTables": {
  "travelrecord": {
    "function": {
      "properties": {
        "dbNum": "2",//分库数量
        "tableNum": "2",//分表数量
        "tableMethod": "hash(id)",//分表分片函数
        "storeNum": 2,//实际存储节点数量
        "dbMethod": "hash(id)",//分库分片函数
        "mappingFormat":
"c${targetIndex}/db1_${dbIndex}/travelrecord_${tableIndex}"
      }
    }
  }
}
```

逻辑库: test1
逻辑表: travelrecord
数据源: c0,c1

MySQL

目标:c0
物理库:db1_0
物理表:
db1_0. travelrecord_0
db1_0. travelrecord_1
URL:jdbc:mysql://localhost:3306/db1_0

目标:c1
物理库:db1_1
物理表:
db1_1. travelrecord_0
db1_1. travelrecord_1
URL:jdbc:mysql://localhost:3307/db1_1

以下说明不区分集群或数据源,统一称为数据源

Mycat2-分片表-全局分区(分表)下标

```
"shardingTables": {
  "travelrecord": {
    "function": {
      "properties": {
        "dbNum": "2",//分库数量
        "tableNum": "2",//分表数量
        "tableMethod": "hash(id)",//分表分片函数
        "storeNum": 2,//实际存储节点数量
        "dbMethod": "hash(id)",//分库分片函数
        "mappingFormat":
"c${targetIndex}/db1/travelrecord_${index}"
      }
    }
  }
}
```

逻辑库: test1
逻辑表: travelrecord
数据源: c0,c1

MySQL

目标:c0
物理库:db1
物理表:
db1. travelrecord_0
db1. travelrecord_1
URL:jdbc:mysql://localhost:3306/db1

目标:c1
物理库:db1
物理表:
db1. travelrecord_2
db1. travelrecord_3
URL:jdbc:mysql://localhost:3307/db1

以下说明不区分集群或数据源,统一称为数据源

Mycat2-分片表-分区枚举-多实例分库

```
"shardingTables": {
  "travelrecord": {
   "function": {
     "clazz": ,//具体自定义分片算法
     "properties": {

       ...分片算法参数
     }
   },
   "partition":{
      "targetNames":"c$0-1",
      "schemaNames":"db1",
      "tableNames":" travelrecord "
   }
  }
}
```

逻辑库: test1
逻辑表: travelrecord
数据源: c0,c1

```
for (String target : targets) {
    for (String schema : schemas) {
        for (String table : tables) {
            ....生成存储节点
        }
    }
}
```

MySQL

目标:c0
物理库:db1
物理表:
db1. travelrecord
URL:jdbc:mysql://localhost:3306/db1

目标:c1
物理库:db1
物理表:
db1. travelrecord
URL:jdbc:mysql://localhost:3307/db1

以下说明不区分集群或数据源,统一称为数据源

Mycat2-分片表-分区枚举-单实例单库分表

```
"shardingTables": {
  "travelrecord": {
   "function": {
     "clazz": ,//具体自定义分片算法
     "properties": {

       ...分片算法参数
     }
   },
   "partition":{
      "targetNames":"c0",
      "schemaNames":"db1",
      "tableNames":" travelrecord_$0-1 "
  }
 }
}
```

逻辑库: test1
逻辑表: travelrecord
数据源: c0

```
for (String target : targets) {
    for (String schema : schemas) {
        for (String table : tables) {
            ....生成存储节点
        }
    }
}
```

MySQL

目标:c0
物理库:db1
物理表:
db1. travelrecord_0
db1. travelrecord_1
URL:jdbc:mysql://localhost:3306/db1

以下说明不区分集群或数据源,统一称为数据源

Mycat2-分片表-分区枚举-层次化分库分表(与分片算法结合)

```
"shardingTables": {
  "travelrecord": {
   "function": {
     "clazz": ,//具体自定义分片算法
     "properties": {

       ...分片算法参数
     }
   },
  "partition":{
     "targetNames":"c$0-1",
     "schemaNames":"db$0-1 ",
     "tableNames":" travelrecord_$0-1 "
  }
 }
}
```

逻辑库: test1
逻辑表: travelrecord
数据源: c0,c1

```
for (String target : targets) {
    for (String schema : schemas) {
        for (String table : tables) {
            ....生成存储节点
        }
    }
}
```

MySQL

目标:c0
物理库: db0,db1
物理表:
db0. travelrecord_0, db0. travelrecord_1
db1. travelrecord_0, db1. travelrecord_1
URL:jdbc:mysql://localhost:3306/db1

目标:c1
物理库: db0 ,db1
物理表:
db0. travelrecord_0, db0. travelrecord_1
db1. travelrecord_0, db1. travelrecord_1
URL:jdbc:mysql://localhost:3307/db1

以下说明不区分集群或数据源,统一称为数据源

Mycat2-分片表-自定义分区(结合分片算法)

```
"shardingTable":{
   "createTableSQL":"….",
   "function":{},
   "partition":{
"data":[["c0","db1","t2","0","0","0"],["c1","db1","t2","1","1","1"]]   }
 }
 }
```

物理分库下标
物理分表下标
全局分区(分表)下标

逻辑库: test1
逻辑表: travelrecord
数据源: c0,c1

MySQL

目标:c0
物理库: db1
物理表:
db1. t2
URL:jdbc:mysql://localhost:3306/db1

目标:c1
物理库: db1
物理表:
db1. t2
URL:jdbc:mysql://localhost:3307/db1