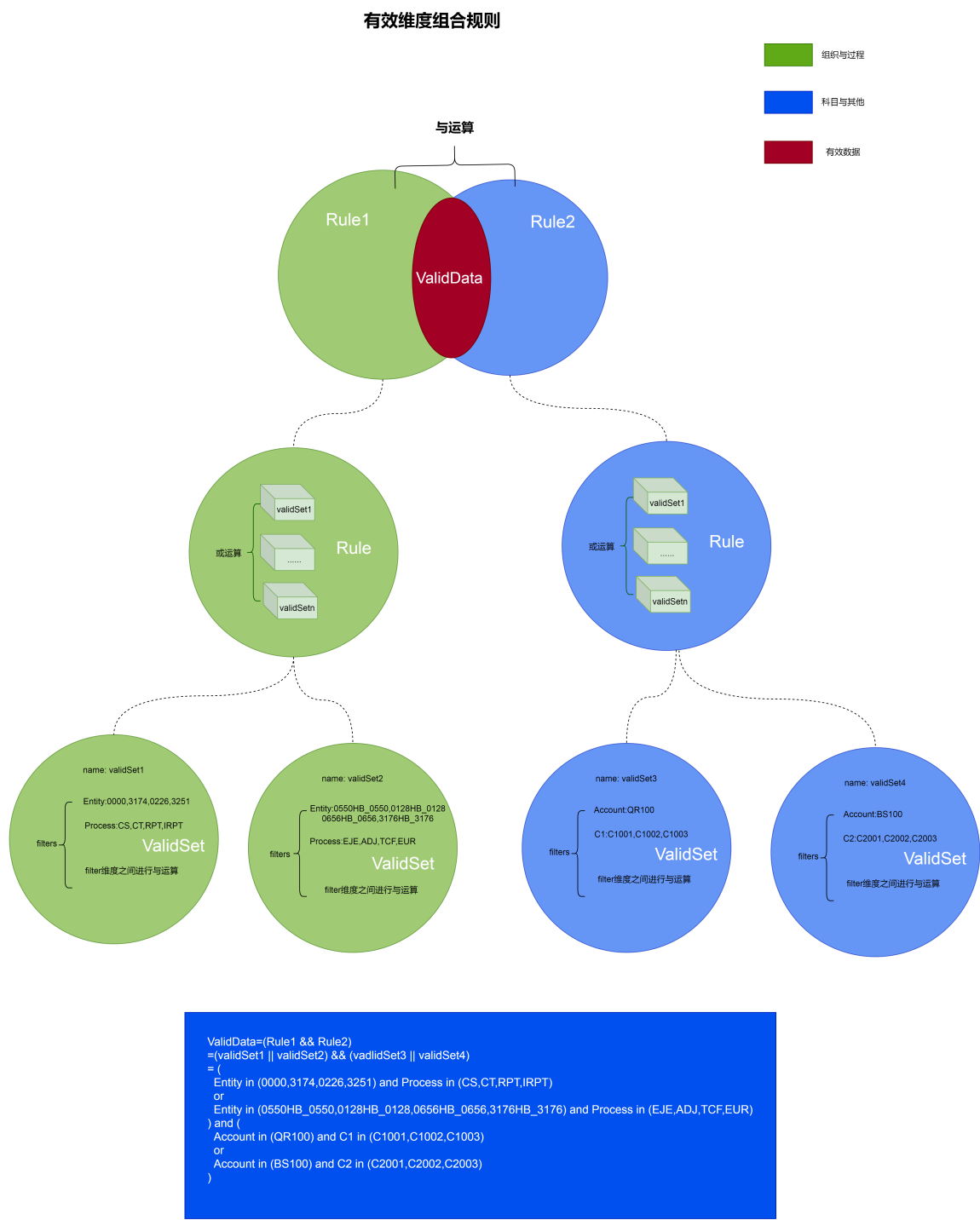


# 有效维度组合规则

本文将给出简单的示例，让您能在您的 Java 程序中对有效维度组合规则进行配置：



支持下面几种：

ValidRule/Types	create	alter	drop	repair
cubeName				√ ( 全量 )
ruleName			√	
validSetName		√ ( 增量创建或更新 )	√	√ ( 全量 )
dimName			√	
filterItem			√	

为了方便，本文指代名词：

	全称	简称
validDataRule	有效维度组合规则	规则
validSet	有效维度组合规则有效集	有效集
validSetDim	有效维度组合规则有效集维度	有效集维度
validSetFilterItem	有效维度组合规则有效集过滤项	有效集过滤项

## 全量修复规则

全量修复规则是覆盖修复，会清空原先所有规则，以发送过来的修复规则为准；

```
/**
 * 全量修复规则
 */
static void repairValidDataRule(OlapConnection olapConn,String cubeName) {
    //构建有效维度组合，有效数据如下
    //( set0 || set1 ) && ( set2 || set3)
    //( set0 && set2 ) || (set0 && set3) || (set1 && set2) || (set1 && set3)
    // years terms accounts orgs
    // y2018 t1      a1,a2  o1,o2
    // y2018 t1      a3,a4  o3,o4
    // y2019 t2,t3   a1,a2  o1,o2
    // y2019 t2,t3   a3,a4  o3,o4
    DefaultValidSetItem validSet0 = new DefaultValidSetItem("set0")
        .addFilter("years", "year2018")
        .addFilter("terms", "term1");

    DefaultValidSetItem validSet1 = new DefaultValidSetItem("set1")
        .addFilter("years", "year2019")
}
```

```

        .addFilter("terms", "term2", "term3");

DefaultValidSetItem validSet2 = new DefaultValidSetItem("set2")
    .addFilter("accounts", "account1", "account2")
    .addFilter("orgs", "org1", "org2");

DefaultValidSetItem validSet3 = new DefaultValidSetItem("set3")
    .addFilter("accounts", "account3", "account4")
    .addFilter("orgs", "org3", "org4");

MetadataCommandInfo metadataCommandInfo = new MetadataCommandInfo();
metadataCommandInfo.setAction(CommandTypes.repair);
metadataCommandInfo.setMetadataType(MetadataTypes.ValidDataRule);
metadataCommandInfo.setOwnerUniqueName(cubeName);
metadataCommandInfo.addValidDataRule("rule1", validSet0, validSet1);
metadataCommandInfo.addValidDataRule("rule2", validSet2, validSet3);
//执行元数据命令
OlapCommand olapCommand = new OlapCommand(olapConn, metadataCommandInfo);
olapCommand.executeNonQuery();
}

```

## 清空所有规则

```

/**
 * 清空所有规则
 */
static void clearValidDataRule(OlapConnection olapConn, String cubeName) {
    MetadataCommandInfo metadataCommandInfo = new MetadataCommandInfo();
    metadataCommandInfo.setAction(CommandTypes.repair);
    metadataCommandInfo.setMetadataType(MetadataTypes.ValidDataRule);
    metadataCommandInfo.setOwnerUniqueName(cubeName);

    //执行元数据命令
    OlapCommand olapCommand = new OlapCommand(olapConn, metadataCommandInfo);
    olapCommand.executeNonQuery();
}

```

## 删除指定规则

```

/**
 * 删除指定规则
 */
static void dropValidDataRule(OlapConnection olapConn, String cubeName, String ruleName) {
    MetadataCommandInfo metadataCommandInfo = new MetadataCommandInfo();
    metadataCommandInfo.setAction(CommandTypes.drop);
    metadataCommandInfo.setMetadataType(MetadataTypes.ValidDataRule);
    metadataCommandInfo.setName(ruleName);
}

```

```

        metadataCommandInfo.setOwnerUniqueName(cubeName);

        //执行元数据命令
        OlapCommand olapCommand = new OlapCommand(olapConn, metadataCommandInfo);
        olapCommand.executeNonQuery();
    }

    //删除rule1
    demo.dropValidDataRule(olapConn, cubeName, "rule1");

```

## 差量创建或更新规则

差量创建或更新有效集，差量新增对应的validSet，rule会自动创建，新增名称存在，不报错

```

/**
 * 差量创建或更新有效集，差量新增对应的有效集，规则会自动创建或更新，如果维度或成员不存在时，不报错
 */
static void alterValidSet(OlapConnection olapConn, String cubeName,
                          String ruleName, DefaultValidSetItem... validSets) {
    MetadataCommandInfo metadataCommandInfo = new MetadataCommandInfo();
    metadataCommandInfo.setAction(CommandTypes.alter);
    metadataCommandInfo.setMetadataType(MetadataTypes.ValidDataRule);
    metadataCommandInfo.setOwnerUniqueName(cubeName + "." + ruleName);
    metadataCommandInfo.getItems().addAll(Arrays.asList(validSets));
    //执行元数据命令
    OlapCommand olapCommand = new OlapCommand(olapConn, metadataCommandInfo);
    olapCommand.executeNonQuery();
}

//往rule1.set0新增year8888
DefaultValidSetItem set0 = new DefaultValidSetItem("set0");
set0.addFilter("years", "year8888");
demo.alterValidSet(olapConn, cubeName, "rule1", set0);
//往rule2.set3新增account300,org888
DefaultValidSetItem set3 = new DefaultValidSetItem("set3");
set3.addFilter("accounts", "account300");
set3.addFilter("orgs", "org888");
demo.alterValidSet(olapConn, cubeName, "rule2", set3);

```

## 删除指定有效集

```

/**
 * 删除指定有效集
 */
static void dropValidSet(OlapConnection olapConn, String cubeName, String ruleName,

```

```

        String validSetName) {
    MetadataCommandInfo metadataCommandInfo = new MetadataCommandInfo();
    metadataCommandInfo.setAction(CommandTypes.drop);
    metadataCommandInfo.setMetadataType(MetadataTypes.ValidDataRule);
    metadataCommandInfo.setName(validSetName);
    metadataCommandInfo.setOwnerUniqueName(cubeName+"."+ruleName);
    //执行元数据命令
    OlapCommand olapCommand = new OlapCommand(olapConn, metadataCommandInfo);
    olapCommand.executeNonQuery();
}

//删除rule1.set0
demo.dropValidSet(olapConn,cubeName,"rule1","set0");

```

## 全量修复指定有效集

```

/**
 * 全量修复指定有效集
 */
static void repairValidSet(OlapConnection olapConn,String cubeName,
        String ruleName,DefaultValidSetItem validSet) {
    MetadataCommandInfo metadataCommandInfo = new MetadataCommandInfo();
    metadataCommandInfo.setAction(CommandTypes.repair);
    metadataCommandInfo.setMetadataType(MetadataTypes.ValidDataRule);
    metadataCommandInfo.setName(validSet.getName());
    metadataCommandInfo.setOwnerUniqueName(cubeName+"."+ruleName);
    metadataCommandInfo.getItems().add(validSet);
    //执行元数据命令
    OlapCommand olapCommand = new OlapCommand(olapConn, metadataCommandInfo);
    olapCommand.executeNonQuery();
}

//修复rule1.set0为year8888,term1
DefaultValidSetItem validSet0 = new DefaultValidSetItem("set0")
    .addFilter("years", "year8888")
    .addFilter("terms", "term1");
demo.repairValidSet(olapConn,cubeName,"rule1",validSet0);

```

## 删除指定有效集维度

```

/**
 * 删除指定有效集维度
 */
static void dropValidSetDim(OlapConnection olapConn,String cubeName,String ruleName,
        String validSetName,String dimName) {
    MetadataCommandInfo metadataCommandInfo = new MetadataCommandInfo();

```

```

        metadataCommandInfo.setAction(CommandTypes.drop);
        metadataCommandInfo.setMetadataType(MetadataTypes.ValidDataRule);
        metadataCommandInfo.setName(dimName);
        metadataCommandInfo.setOwnerUniqueName(cubeName+"."+ruleName+"."+validSetName);
        //执行元数据命令
        OlapCommand olapCommand = new OlapCommand(olapConn, metadataCommandInfo);
        olapCommand.executeNonQuery();
    }

    //删除rule1.set0的terms维度
    demo.dropValidSetDim(olapConn, cubeName, "rule1", "set0", "terms");

```

## 删除指定有效集过滤项

```

/**
 * 删除指定有效集过滤项
 */
static void dropValidSetFilterItem(OlapConnection olapConn, String cubeName,
                                   String ruleName, DefaultValidSetItem validSet) {
    MetadataCommandInfo metadataCommandInfo = new MetadataCommandInfo();
    metadataCommandInfo.setAction(CommandTypes.drop);
    metadataCommandInfo.setMetadataType(MetadataTypes.ValidDataRule);
    metadataCommandInfo.setOwnerUniqueName(cubeName+"."+ruleName+"."+validSet.getName());
    metadataCommandInfo.getItems().add(validSet);
    //执行元数据命令
    OlapCommand olapCommand = new OlapCommand(olapConn, metadataCommandInfo);
    olapCommand.executeNonQuery();
}

//删除rule1.set1的 year2019 term2,term3
DefaultValidSetItem validSet0 = new DefaultValidSetItem("set1")
    .addFilter("years", "year2019")
    .addFilter("terms", "term2", "term3");
demo.dropValidSetFilterItem(olapConn, cubeName, "rule1", validSet0);

```

## 查询接口

默认通过function接口查询，下面两个方法：

getValidDataRules

getValidDataRulesMetadata

function接口客户端驱动包默认可以使用，业务端增加对应方法添加使用即可。

一个全量查询，一个层级查询；

CUBECDM\_zs1386833854433620110

已分区

追踪

管理

性能分析

慢语句查询

自定义方法

## 执行自定义方法

getValidDataRules

cubeName ?

CUBECDM\_zs1386833854433620110

执行

CUBECDM\_zs1386833854433620110

已分区

追踪

管理

性能分析

慢语句查询

自定义方法

## 执行自定义方法

getValidDataRulesMetadata

cubeName ?

CUBECDM\_zs1386833854433620110

path ?

path

children ?

☐

执行

上一篇：[有效维度组合规则](#)

下一篇：[tcp连接使用](#)