

功能的背景

业务系统使用过程中，维度成员的层级结构会随着时间而进行调整，例如：
合并报表的往来组织维度中，某些组织可能在某段时间是属于内部组织，某段时间又属于外部组织。
这样的变更在业务系统使用过程中是常见的，如果在这种场景中使用 [分段公式](#)，理论是可行，
但是随着变更的频率的增加，分段公式会显得很臃肿，使用性能也会随之大幅下降。

缓慢变化

自定义属性及标签

自定义属性及标签为缓慢变化相关的内容，请参考 [自定义属性及标签](#)。

成员函数表达式及标签表达式

成员函数表达式及成员标签表达式为缓慢变化相关的内容，请参考 [成员函数表达式及成员标签表达式](#)。

整型时间值

当一个成员的属性或者标签发生变更时，可以设置其在不同时间段对应的属性值或者标签。
以星瀚合并报表应用为例，一个标准的合并报表体系包含两个与时间相关的维度：`Year` 和 `Period`，
相应的，多维库内置一个合并报表获取时间值的模型：`bcmGetDate`，该模型可通过 `Year` 和 `Period`
计算得到一个 `整型时间值`。

例如：

`Year: FY2024, Period: M_M07` 对应的整型时间值为 20240700；

`Year: FY2024, Period: Q_Q02` 对应的整型时间值为 20240400。

以 `整型时间值` 为轴心，可以设置不同轴段（时间段）的属性值或标签。

基于成员属性的缓慢变化设置

在往来组织场景中，可以使用 `intC0` 用于区分内部和外部组织，例如：

0：不是内部组织也不是外部组织

1：内部组织

2：外部组织

现有往来组织，其各个时间段 `intC0` 的属性值如下：

`before ~ 20200400`：内部组织，`intC0=1`；

`20200400 ~ 20221100`：外部组织，`intC0=2`；

`20221100 ~ now`：内部组织，`intC0=1`；

多维数据库通过倒序的方式对以上的缓慢变化进行描述：

`{ lowerLimit: 20221100, value: 1 }`：表示在 `>= 20221100` 时间段，`intC0` 的值为 1；

`{ lowerLimit: 20200400, value: 2 }`：结合上一个时间段，表示在 `[20200400, 20221100)` 时间段，`intC0` 的值为 2；

`initValue: 1`：结合上一个时间段，表示在 `(before, 20221100)` 时间段，`intC0` 的值为 1；

/**

* 设置往来组织的缓慢变化场景

* @param type 创建成员（create）和修改成员（alter）均支持

```

* @param dimension 维度名称
* @param member 成员名称
*/
public static void membersSlowEvolving(CommandTypes type, String dimension,
String member) {
    OlapConnection olapConn = OlapConnectionUtil.getConnection(cubeName);
    MetadataCommandInfo commandInfo = new MetadataCommandInfo();
    commandInfo.setMetadataType(MetadataTypes.Member);
    commandInfo.setOwnerUniqueName(cubeName + "." + dimension);
    commandInfo.setAction(type);
    // 给一个成员设置成员标签表达式，一个命令设置多个成员的缓慢变化。
    MemberMetadataItem item = new MemberMetadataItem(member);
    SlowEvolvingProfileMetadataItem slowEvolvingItem = new
SlowEvolvingProfileMetadataItem();
    slowEvolvingItem.setPropertyName("intc0");
    List<SlowEvolvingProfileMetadataItem> list = new ArrayList<>();
    DateValuesSlowEvolvingSetMetadataItem dateValuesSlowEvolvingItem = new
DateValuesSlowEvolvingSetMetadataItem();
    dateValuesSlowEvolvingItem.setDateModel(BCM_GET_DATE);
    dateValuesSlowEvolvingItem.getItems().add(new
DateValuesSlowEvolvingPairMetadataItem(20221100, 1));
    dateValuesSlowEvolvingItem.getItems().add(new
DateValuesSlowEvolvingPairMetadataItem(20200400, 2));
    dateValuesSlowEvolvingItem.setInitValue(1);
    slowEvolvingItem.setSlowEvolvingSet(dateValuesSlowEvolvingItem);
    list.add(slowEvolvingItem);
    // 一个成员可以设置多个属性的缓慢变化。
    item.setSlowEvolvingProfiles(list);
    commandInfo.getItems().add(item);

    OlapCommand command = new OlapCommand(olapConn, commandInfo);
    command.executeNonQuery();
}

```

基于成员属性的缓慢变化表达式

将 成员函数表达式 和 基于成员属性的缓慢变化 结合起来，就是 基于成员属性的缓慢变化表达式。
合并报表体系中，一般使用成员 `ICEntity` 来表示往来组织所有内部组织的汇总，
那么在存在缓慢变化的场景下，基于成员属性的缓慢变化表达式可以这样设置：

```

/**
* 设置缓慢变化成员函数表达式
* @param type 创建成员（create）和修改成员（alter）均支持
*/
public static void setAggFunSlowEvolvingExpression(CommandTypes type) {
    OlapConnection olapConn = OlapConnectionUtil.getConnection(cubeName);
    MetadataCommandInfo commandInfo = new MetadataCommandInfo();
    commandInfo.setMetadataType(MetadataTypes.Member);
    commandInfo.setOwnerUniqueName(cubeName + "." + "InternalCompany");
    commandInfo.setAction(type);

    MemberMetadataItem memberItem = new MemberMetadataItem("ICEntity");
    AggFunSlowEvolvingExpressionMetadataItem item = new
AggFunSlowEvolvingExpressionMetadataItem();
    item.setName("ICEntity"); // 设置成员名称
    item.setObservedPropertyNames("intc0"); // 设置观察的属性名称
    item.setBaseExpression(getBaseAggFunExpression());
}

```

```

memberItem.setExpression(item);

commandInfo.getItems().add(memberItem);
OlapCommand command = new OlapCommand(olapConn, commandInfo);
command.executeNonQuery();
}

private static AggFunExpressionMetadataItem getBaseAggFunExpression() {
    AggFunExpressionMetadataItem item = new AggFunExpressionMetadataItem();
    item.setAggOperatorExpression("property(currentMember,'intC0') == 1"); // 设置表达式
    // 请尽量设置预定义因子清单
    return item;
}

```

基于成员属性的缓慢变化表达式的查询

通过自定义方法 `getFactorsV2`，基于维度以及成员，使用路径查找模式查询成员属性缓慢变化表达式

路径：表示获取当前cube的维度成员因子结构，path格式为：1、dimName 2、dimName.memberName 3、dimName@memberName

管理

aggFunCube

追踪

管理

性能分析

慢语句查询

自定义方法

异步任务中心 正常

执行自定义方法

getFactorsV2

cubeName ?

aggFunCube

path ?

orgs@orgs_3

执行

执行并下载

result

```

1  [[
2    {
3      "name": "orgs_3",
4      "storageType": "dynamicCalc",
5      "aggShieldRule": "all",
6      "calcExpression": {
7        "type": "aggFunSlowEvolvingExpression.v1",
8        "baseExpression": {
9          "type": "aggFunExpression.v1",
10         "aggOperatorExpression": "property(currentMember,'intC0') == 1",
11         "predefinedFactors": [],
12         "usingPredefinedFactors": false
13       },
14       "observedPropertyNames": "intC0"
15     }
16  ]]

```

通过自定义方法 `getMemberCustomPropertyAndTag`，基于维度以及成员清单，使用成员查找模式查询缓慢变化

成员查找模式：提供成员清单（使用JSON规范，例如：["member1","member2"]），获取所提供成员的自定义属性、打上的标签、缓慢变化，如传入 [] 则获取所有成员的自定义属性和打上的标签。

管理

aggFunCube 追踪 管理 性能分析 慢语句查询 自定义方法 异步任务中心 正常

执行自定义方法

getMemberCustomPropertyAndTag

cubeName ②
aggFunCube

Dimension ②
orgs

Tags ②
Tags

Members ②
["orgs_3"]

执行 执行并下载

Result

```
1 [[
2   "member": "orgs_3",
3   "slowEvolvingProfiles": [{
4     "property": "int00",
5     "memberName": "orgs_3",
6     "slowEvolvingSet": {
7       "type": "dateValueSlowEvolvingSet.v1",
8       "dateModel": "bcmGetDate",
9       "itemsJson": "{ \"initValue\":1, \"items\": [{ \"lowerLimit\":20221100, \"value\":1}, { \"lowerLimit\":20200400, \"value\":2} ] }"
10    }
11  }]
12 ]]
```

基于成员标签的缓慢变化设置

在往来组织场景中，也可以使用标签来用于区分内部和外部组织，例如：

不是内部组织也不是外部组织：不打任何标签

内部组织：打上 parent-internal 标签

外部组织：打上 parent-external 标签

现有往来组织，其各个时间段的标签如下：

before ~ 20200400：内部组织，打上 parent-internal 标签；

20200400 ~ 20221100：外部组织，打上 parent-external 标签；

20221100 ~ now：内部组织，打上 parent-internal 标签；

多维数据库通过倒序的方式对以上的缓慢变化进行描述：

{ lowerLimit: 20221100, value: "parent-internal" }：表示在 ≥ 20221100 时间段，打上 parent-internal 标签；

{ lowerLimit: 20200400, value: "parent-external" }：结合上一个时间段，表示在 [20200400, 20221100) 时间段，打上 parent-external 标签；

initValue: "parent-internal"：结合上一个时间段，表示在 (before, 20221100) 时间段，打上 parent-internal 标签；

这里需要特别注意的是，逻辑上一个往来组织在任何一个时间点只能打上 parent-internal 或 parent-external 标签，

但是，成员标签表达式跟成员函数表达式不一样，成员标签表达式是不支持预定义因子的，其在内部只关注打上相关标签的成员，

因此这里需要同时给这个往来组织打上 `parent-internal` 或 `parent-external` 标签。

```
/**
 * 设置往来组织的缓慢变化场景
 * @param type 创建成员（create）和修改成员（alter）均支持
 * @param dimension 维度名称
 * @param member 成员名称
 */
public static void memberSlowEvolving(CommandTypes type, String dimension,
String member) {
    OlapConnection olapConn = OlapConnectionUtil.getConnection(cubeName);
    MetadataCommandInfo commandInfo = new MetadataCommandInfo();
    commandInfo.setMetadataType(MetadataTypes.Member);
    commandInfo.setOwnerUniqueName(cubeName + "." + dimension);
    commandInfo.setAction(type);

    // 给一个成员设置成员标签表达式，一个命令设置多个成员的缓慢变化。
    MemberMetadataItem item = new MemberMetadataItem(member);
    SlowEvolvingProfileMetadataItem slowEvolvingItem = new
SlowEvolvingProfileMetadataItem();
    slowEvolvingItem.setPropertyName("parent"); // 注意：这个属性可以不是成员属性
    List<SlowEvolvingProfileMetadataItem> list = new ArrayList<>();
    DateValuesSlowEvolvingSetMetadataItem dateValuesSlowEvolvingItem = new
DateValuesSlowEvolvingSetMetadataItem();
    dateValuesSlowEvolvingItem.setDateModel(BCM_GET_DATE);
    dateValuesSlowEvolvingItem.getItems().add(new
DateValuesSlowEvolvingPairMetadataItem(20221100, "parent-internal"));
    dateValuesSlowEvolvingItem.getItems().add(new
DateValuesSlowEvolvingPairMetadataItem(20200400, "parent-external"));
    dateValuesSlowEvolvingItem.setInitValue("parent-internal");
    slowEvolvingItem.add(dateValuesSlowEvolvingItem);
    // 一个成员可以设置多个属性的缓慢变化。
    slowEvolvingItem.setSlowEvolvingSet(dateValuesSlowEvolvingItem);
    list.add(slowEvolvingItem);
    item.setSlowEvolvingProfiles(list);

    commandInfo.getItems().add(item);
    OlapCommand command = new OlapCommand(olapConn, commandInfo);
    command.executeNonQuery();
}
```

基于成员标签的缓慢变化表达式

将 成员标签表达式 和 基于成员标签的缓慢变化 结合起来，就是 基于成员标签的缓慢变化表达式。合并报表体系中，一般使用成员 `ICEntity` 来表示往来组织所有内部组织的汇总，那么在存在缓慢变化的场景下，基于成员标签的缓慢变化表达式可以这样设置：

```
/**
 * 设置缓慢变化成员标签表达式
 * @param type 创建成员（create）和修改成员（alter）均支持
 */
public static void setAggTagSlowEvolvingExpression(CommandTypes type) {
    OlapConnection olapConn = OlapConnectionUtil.getConnection(cubeName);
    MetadataCommandInfo commandInfo = new MetadataCommandInfo();
    commandInfo.setMetadataType(MetadataTypes.Member);
    commandInfo.setOwnerUniqueName(cubeName + "." + "InternalCompany");
}
```

```

        commandInfo.setAction(type);

        AggTagSlowEvolvingExpressionMetadataItem item = new
        AggTagSlowEvolvingExpressionMetadataItem();
        item.setName("ICEntity");    // 设置成员名称
        item.setObservedPropertyNames("parent"); // 设置观察的属性名称
        item.setBaseExpression(getBaseAggTagExpression());

        commandInfo.getItems().add(item);
        OlapCommand command = new OlapCommand(olapConn, commandInfo);
        command.executeNonQuery();
    }

    private static AggTagExpressionMetadataItem getBaseAggTagExpression() {
        AggTagExpressionMetadataItem item = new AggTagExpressionMetadataItem();
        item.setTags("parent-internal");    // 设置标签
        return item;
    }

```

基于成员标签的缓慢变化表达式的查询

通过自定义方法 `getFactorsV2`，基于维度以及成员，使用路径查找模式查询成员属性缓慢变化表达式

路径：表示获取当前cube的维度成员因子结构，path格式为：1、dimName 2、dimName.memberName 3、dimName@memberName

管理

aggFunCube 追踪 管理 性能分析 慢语句查询 自定义方法 异步任务中心 正常

执行自定义方法

getFactorsV2

cubeName ?
aggFunCube

path ?
orgs@orgs_slow

执行 执行并下载

result

```

1  [[
2      "name": "orgs_slow",
3      "aggShieldRule": "all",
4      "calcExpression": {
5          "type": "aggTagSlowEvolvingExpression.v1",
6          "baseExpression": {
7              "type": "aggTagExpression.v1",
8              "tags": "tag1"
9          },
10         "observedPropertyNames": "parent"
11     }
12 ]]

```

通过自定义方法 `getMemberCustomPropertyAndTag`，基于维度以及成员清单，使用成员查找模式查询缓慢变化

成员查找模式：提供成员清单（使用JSON规范，例如：`["member1","member2"]`），获取所提供成员的自定义属性、打上的标签、缓慢变化，如传入 `[]` 则获取所有成员的自定义属性和打上的标签。

管理

aggFunCube

追踪

管理

性能分析

慢语句查询

自定义方法

异步任务中心 正常

执行自定义方法

getMemberCustomPropertyAndTag

×

cubeName

aggFunCube

Dimension

orgs

Tags

Tags

Members

["orgs_slow"]

执行

执行并下载

Result

```
1  [[
2
3      "member": "orgs_slow",
4      "slowEvolvingProfiles": [[
5          "property": "parent",
6          "memberName": "orgs_slow",
7          "slowEvolvingSet": {
8              "type": "dateValueSlowEvolvingSet.v1",
9              "dateModel": "bcmGetDate",
10             "itemsJson": "{ \"initValue\": \"tag2\", \"items\": [{ \"lowerLimit\": 20221100, \"value\": \"tag1\" }, { \"lowerLimit\": 20200400, \"value\": \"tag2\" } ] }"
11         }
12     ]
13 ]]
```

删除缓慢变化

```
/**
 * 删除缓慢变化
 * @param type 支持使用修改成员（alter）进行删除
 */
public static void deleteAllSlowEvolving(String cubeName, String dimension,
String Member){
    OlapConnection olapConn = getConnection(cubeName, null);
    MetadataCommandInfo commandInfo = new MetadataCommandInfo();
    commandInfo.setMetadataType(MetadataTypes.Member);
    commandInfo.setOwnerUniqueName(cubeName + "." + dimension);
    commandInfo.setAction(CommandTypes.alter);

    MemberMetadataItem memberItem = new MemberMetadataItem(Member);
    SlowEvolvingProfileMetadataItem slowEvolvingItem = new
SlowEvolvingProfileMetadataItem();
    slowEvolvingItem.setPropertyName("");
    memberItem.setSlowEvolvingProfiles(Arrays.asList(slowEvolvingItem));

    commandInfo.getItems().add(memberItem);
```

```
    OlapCommand command = new OlapCommand(olapConn, commandInfo);  
    command.executeNonQuery();  
}
```

上一篇：[成员函数表达式及成员标签表达式](#)

下一篇：[度量值过滤](#)