

[返回首页](#)

## 创建 Cube

当成功连接到多维数据库服务器上，您便可以创建 Cube。  
本文将给出简单的示例，让您能在您的 Java 程序中创建一个完整的 Cube。

### 创建 Cube

类似在关系数据库中创建一张表，您可以在 Shrek 创建一个 Cube。

```
createCube(olapConn, cubeName);
```

其中createCube函数实现如下：

```
private static void createCube(OlapConnection olapConn, String cubeName) {
    MetadataCommandInfo commandInfo = new MetadataCommandInfo();
    commandInfo.setMetadataType(MetadataTypes.Cube);
    commandInfo.setName(cubeName);
    commandInfo.setAction(CommandTypes.create);

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

### 创建维度

现在您已创建一个空白的 Cube，下面示例如何在 Cube 中创建名称为“accounts”的维度。

```
createDimension(olapConn, cubeName, "accounts"); //创建名称为“accounts”的维度
```

其中createDimension函数实现如下：

```
private static void createDimension(OlapConnection olapConn, String cubeName, String dimension)
{
    MetadataCommandInfo commandInfo = new MetadataCommandInfo();
    commandInfo.setAction(CommandTypes.create);
    commandInfo.setMetadataType(MetadataTypes.Dimension);
    commandInfo.setName(dimension);
    commandInfo.setOwnerUniqueName(cubeName);

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

通过多次调用 createDimension 方法，我们继续创建维度“years”，“terms”，“orgs”。

## 创建成员

您可以为每个维度创建多个成员，下面演示为维度“accounts”创建名称为“account1”，“account2”，“account3”三个成员：

```
createMember(olapConn, cubeName, "accounts", "account1"); //为维度 “accounts” 创建名称为
“account1” 的成员
createMember(olapConn, cubeName, "accounts", "account2"); //为维度 “accounts” 创建名称为
“account2” 的成员
createMember(olapConn, cubeName, "accounts", "account3"); //为维度 “accounts” 创建名称为
“account2” 的成员
```

其中createMember函数实现如下：

```
private static void createMember(OlapConnection olapConn, String cubeName, String dimension,
String member) {
    MetadataCommandInfo commandInfo = new MetadataCommandInfo();
    commandInfo.setAction(CommandTypes.create);
    commandInfo.setMetadataType(MetadataTypes.Member);
    commandInfo.setName(member);
    commandInfo.setOwnerUniqueName(cubeName + "." + dimension);
    OlapCommand command = new OlapCommand(olapConn, commandInfo);
    command.executeNonQuery();
}
```

## 创建度量值

您可以在Cube 中创建度量值,下面演示创建名为“FMONEY”的度量值：

```
createMeasure(olapConn, cubeName, "FMONEY");
```

其中createMeasure函数实现如下：

```
private static void createMeasure(OlapConnection olapConn, String cubeName, String measure) {
    MetadataCommandInfo commandInfo = new MetadataCommandInfo();
    commandInfo.setAction(CommandTypes.create);
    commandInfo.setMetadataType(MetadataTypes.Measure);
    commandInfo.setName(measure);
    commandInfo.setOwnerUniqueName(cubeName);
    commandInfo.setDataTypes(OlapDataTypes.object);

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

## 效果

通过以上一系列的创建操作，我们创建了一个完整的 Cube，可在浏览器的交互页面中看到如下效果：

The screenshot shows a web interface for a cube named 'YourFirstCube'. It has two tabs: '追踪' (Track) and '管理' (Manage). Under the '追踪' tab, there are four dimension groups: 'accounts' with values 'account1', 'account2', and 'account3'; 'years' with values 'year2017', 'year2018', and 'year2019'; 'terms' with values 'term1' and 'term2'; and 'orgs' with values 'org1' and 'org2'. Each group has a '清空' (Clear) button. Below the dimensions, there is a '度量值' (Measures) section with a checked 'FMONEY' measure and a '0 数据量' (0 Data Volume) indicator.

现在您学会了如何创建 Cube 及维度，成员，度量值，后续教程将以该 Cube 为示例进行演示。

## 删除 Cube

您可以删除一个已存在的 Cube,函数实现如下:

```
private static void dropCube(OlapConnection olapConn, String cubeName) {  
    MetadataCommandInfo commandInfo = new MetadataCommandInfo();  
    commandInfo.setMetadataType(MetadataTypes.Cube);  
    commandInfo.setName(cubeName);  
    commandInfo.setAction(CommandTypes.drop);  
    OlapCommand command = new OlapCommand(olapConn, commandInfo);  
    command.executeNonQuery();  
}
```

上一篇：[连接到多维数据库服务器](#)

下一篇：[数据处理](#)